8362751

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

April 05, 2023

THIS IS TO CERTIFY THAT ANNEXED IS A TRUE COPY FROM THE RECORDS OF THIS OFFICE OF THE FILE WRAPPER AND CONTENTS OF:

APPLICATION NUMBER: *10/181,564*
FILING DATE: *July 18, 2002*
PATENT NUMBER: *7266682*
ISSUE DATE: *September 04, 2007*

Certified by

Performing the Functions and Duties of the
Under Secretary of Commerce
for Intellectual Property
and Director of the United States
Patent and Trademark Office

533 Rec'd PCT/PTO 18 JUL 2002

S&H Form: FORM PTO-1390 (2/01)

U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE

TRANSMITTAL LETTER TO THE UNITED STATES DESIGNATED/ELECTED OFFICE (DO/EO/US) CONCERNING A FILING UNDER 35 U.S.C. 371

ATTORNEY'S DOCKET NUMBER
1454.1302

10/181564

INTERNATIONAL APPLICATION NO.	INTERNATIONAL FILING DATE	PRIORITY DATE CLAIMED
PCT/DE01/00021	5 January 2001	18 January 2000

TITLE OF INVENTION
METHOD, SYSTEM FOR TRANSMITTING DATA FROM A TRANSMITTER TO A RECEIVER, AND TRANSMITTER OR RECEIVER

APPLICANT(S) FOR DO/EO/US
Martin EUCHNER

Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. ☒ This is a FIRST submission of items concerning a filing under 35 U.S.C. 371.
2. ☒ This is an express request to immediately begin national examination procedures (35 U.S.C. 371(f)).
3. ☒ The US has been elected by the expiration of 19 months from the priority date (PCT Article 31).
4. ☒ A copy of the International Application as filed (35 U.S.C. 371(c)(2))
   a. ☒ is transmitted herewith (required only if not transmitted by the International Bureau).
   b. ☐ has been transmitted by the International Bureau.
   c. ☐ is not required, as the application was filed in the United States Receiving Office (RO/US).
5. ☒ A translation of the International Application into English (35 U.S.C. 371(c)(2)).
6. ☐ Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371(c)(3))
   a. ☐ are transmitted herewith (required only if not transmitted by the International Bureau).
   b. ☐ have been transmitted by the International Bureau.
   c. ☐ is not required, as the application was filed in the United States Receiving Office (RO/US)
7. ☐ A translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)).
8. ☒ An oath or declaration of the inventor (35 U.S.C. 371(c)(4)).
9. ☐ A translation of the Annexes to the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371(c)(5)).

Items 10-15 below concern document(s) or information included:

10. ☒ An Information Disclosure Statement Under 37 CFR 1.97 and 1.98.
11. ☒ An assignment document for recording.
    Please mail the recorded assignment document to:
    a. ☒ the person whose signature, name & address appears at the bottom of this document.
    b. ☐ the following:
12. ☒ A preliminary amendment.
13. ☒ A substitute specification
14. ☐ A change of power of attorney and/or address letter.
15. ☒ Other items or information:

Copies of PCT EASY forms filed with International Application, cover page of International Application as published, International Search Report, and International Preliminary Examination Report.

©2001 Staas & Halsey LLP

☒ The U.S. National Fee (35 U.S.C. 371(c)(1)) and other fees as follows:

CLAIMS	(1) FOR	(2) NUMBER FILED	(3) NUMBER EXTRA	(4) RATE	(5) CALCULATIONS
	TOTAL CLAIMS	27 -20=	7	x $ 18.00	126.00
	INDEPENDENT CLAIMS	4 -3=	1	x $ 84.00	84.00
	MULTIPLE DEPENDENT CLAIM(S) (if applicable)			+$280.00	0.00
	BASIC NATIONAL FEE (37 CFR 1.492(a)(1)-(4)): ☐ Neither international preliminary examination fee (37 CFR 1.482) nor international search fee (37 CFR 1.445(a)(2)) paid to USPTO .....$1,040 ☐ International preliminary examination fee (37 C.F.R. 1.482) not paid to USPTO but International Search Report prepared by the EPO or JPO.....$ 890 ☐ International preliminary examination fee (37 C.F.R. 1.482) not paid to USPTO but international search fee (37 C.F.R. 1.445(a)(2) paid to USPTO...$ 740 ☐ International preliminary examination fee paid to USPTO (37 CFR 1.482) but all claims did not satisfy provision of PCT Article 33(1)-(4).....$ 710 ☐ International preliminary examination fee paid to USPTO (37 CFR 1.482) and all claims satisfied provisions of PCT Article 33(2) to (4) .....$ 100				890.00
	Surcharge of $130 for furnishing the National fee or oath or declaration later than ☐ 20 ☐ 30 mos. from the earliest claimed priority date (37 CFR 1.482(e)).				0.00
			TOTAL OF ABOVE CALCULATIONS		1,100.00
	Reduction by 1/2 for filing by small entity, if applicable. Affidavit must be filed also. (Note 37 CFR 1.9, 1.27, 1.28.)				
			SUBTOTAL		1,100.00
	Processing fee of $130 for furnishing the English Translation later than [ ] 20 [ ] 30 mos. from the earliest claimed priority date (37 CFR 1.482(f)).				
			TOTAL NATIONAL FEE		1,100.00
	Fee for recording the enclosed assignment (37 CFR 1.21(h)).				+ 40.00
			TOTAL FEES ENCLOSED		1,140.00

a. ☒ A check in the amount of $1,140.00 to cover the above fees is enclosed.
b. ☐ Please charge my Deposit Account No. 19-3935 in the Amount of $ to cover the above fees. A duplicate copy of this sheet is enclosed.
c. ☒ The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account No. 19-3935. A duplicate copy of this sheet is enclosed.

21171
PATENT TRADEMARK OFFICE

SUBMITTED BY: STAAS & HALSEY LLP

Type Name	Richard A. Gollhofer	Reg. No.	31,106
Signature	Richard A. Gollhofer	Date	7/18/02

©2001 Staas & Halsey LLP

10/181564
512 Rec'd PCT/PTO 16 JUL 2002

Docket No.: 1454.1302

#4/a

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of:

Martin EUCHNER

Serial No. Group Art Unit:

Confirmation No.

Filed: (concurrently) Examiner:

For: METHOD AND SYSTEM FOR TRANSMITTING DATA FROM A TRANSMITTER TO A RECEIVER, AND TRANSMITTER AND RECEIVER THEREFOR (as amended)

PRELIMINARY AMENDMENT

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

Before examination of the above-identified application, please amend the application as follows:

IN THE TITLE:

Please change "METHOD," to --METHOD AND-- and "OR RECEIVER" to --AND RECEIVER THEREFOR--.

IN THE SPECIFICATION:

Please REPLACE the pending specification with the substitute specification attached hereto.

IN THE CLAIMS:

Please cancel without prejudice or disclaimer claims 1-16 and ADD new claims 17-43 in accordance with the following:

16. (NEW) A method for transmitting data from a transmitter to a receiver, comprising:

extending the data in the transmitter by authentication data using an application protocol on an application layer;

ascertaining, by the receiver, whether the receiver knows the transmitter based on the authentication data; and

accepting the data at the receiver, if the receiver knows the transmitter, and otherwise rejecting the data.

17. (NEW) The method as claimed in claim 16, wherein the data are transmitted on a packet-oriented basis.

18. (NEW) The method as claimed in claim 16, wherein the authentication data are determined based on at least part of a protocol data unit available on the application layer, and the application protocol being linked to a secret by the transmitter.

19. (NEW) The method as claimed in claim 18, wherein the linking to the secret is carried out by at least one of cryptographical linking to at least one further parameter, particularly by virtue of an encryption operation and a cryptographic checksum and key information.

20. (NEW) The method as claimed in claim 19, wherein said ascertaining includes verification at the receiver to determine whether the receiver knows the transmitter.

21. (NEW) The method as claimed in claim 20, wherein the verification is cryptographic verification using parameters and including at least one of a decryption operation and a cryptographic checksum check.

22. (NEW) The method as claimed in claim 21, wherein the secret between the transmitter and the receiver is one of a key for symmetric encryption and a key pair for asymmetric encryption.

23. (NEW) The method as claimed in claim 22, wherein the part of the protocol data unit available on the application layer includes at least one of a sequence number and a timestamp.

24. (NEW) The method as claimed in claim 23, wherein the encryption operation is performed using one of a one-way hash function and a message authentication code with a key known only to the receiver and the transmitter to be authenticated.

25. (NEW) The method as claimed in claim 24, wherein the authentication is carried out between the transmitter and the receiver before the data are transmitted from the transmitter to the receiver.

26. (NEW) The method as claimed in claim 25, wherein the receiver and transmitter provide packet-switched telephone services.

27. (NEW) The method as claimed in claim 26, wherein the receiver and transmitter are each disposed in at least one of a switching node and a switching installation.

28. (NEW) The method as claimed in claim 25, wherein the receiver and transmitter are used in Internet telephony.

29. (NEW) The method as claimed in claim 28, wherein the receiver and transmitter are each disposed in at least one of a switching node and a switching installation.

30. (NEW) The method as claimed in claim 20, wherein the secret between the transmitter and the receiver is one of a key for symmetric encryption and a key pair for asymmetric encryption.

31. (NEW) The method as claimed in claim 20, wherein the part of the protocol data unit available on the application layer includes at least one of a sequence number and a timestamp.

32. (NEW) The method as claimed in claim 20, wherein the encryption operation is performed using one of a one-way hash function and a message authentication code with a key known only to the receiver and the transmitter to be authenticated.

33. (NEW) The method as claimed in claim 19, wherein the secret between the transmitter and the receiver is one of a key for symmetric encryption and a key pair for asymmetric encryption.

34. (NEW) The method as claimed in claim 19, wherein the part of the protocol data unit available on the application layer includes at least one of a sequence number and a timestamp.

35. (NEW) The method as claimed in claim 19, wherein the encryption operation is performed using one of a one-way hash function and a message authentication code with a key known only to the receiver and the transmitter to be authenticated.

36. (NEW) The method as claimed in claim 18, wherein the secret between the transmitter and the receiver is one of a key for symmetric encryption and a key pair for asymmetric encryption.

37. (NEW) The method as claimed in claim 18, wherein the part of the protocol data unit available on the application layer includes at least one of a sequence number and a timestamp.

38. (NEW) The method as claimed in claim 18, wherein the secret is an encryption operation performed using one of a one-way hash function and a message authentication code with a key known only to the receiver and the transmitter to be authenticated.

39. (NEW) The method as claimed in claim 18, wherein said ascertaining includes cryptographic verification using parameters and including at least one of a decryption operation and a cryptographic checksum check.

40. (NEW) The method as claimed in claim 18, wherein said ascertaining includes cryptographic verification using parameters and including at least one of a decryption operation and a cryptographic checksum check.

41. (NEW) A system for transmitting data, comprising:

a transmitter to extend the data by authentication data using an application protocol on an application layer; and

a receiver to determine whether the transmitter is known based on the authentication data, to accept the data if the transmitter is known and to reject the data if the transmitter is not known.

42. (NEW) A transmitter for sending data to a receiver, comprising:

a processor to extend the data by authentication data using an application protocol on an application layer; and

a transmission unit to send the data and authentication data to the receiver.

43. (NEW) A receiver for receiving data from a transmitter, comprising:

a processing unit, coupled to the transmitter, to determine whether the transmitter is known based on authentication data added by the transmitter according to an application protocol on an application layer, to accept the data if the transmitter is known, and to reject the data if the transmitter is not known.

IN THE ABSTRACT:

Please DELETE the Abstract in its entirety and replace with the attached Substitute Abstract.

REMARKS

This Preliminary Amendment is submitted to improve the form of the English translation as filed. It is respectfully requested that this Preliminary Amendment be entered in the above-referenced application.

In accordance with the foregoing, claims 1-16 have been canceled and claims 17-43 have been added. Thus, claims 17-43 are pending and are under consideration.

A substitute specification is also being filed herewith. The substitute specification is accompanied by a marked-up copy of the original specification.

If there are any questions regarding these matters, such questions can be addressed by telephone to the undersigned. Otherwise, an early action on the merits is respectfully solicited.

If any further fees are required in connection with the filing of this Preliminary Amendment, please charge same to our Deposit Account No. 19-3935.

Respectfully submitted,

STAAS & HALSEY LLP

Date: 7/18/02

By: Richard A. Gollhofer
Richard A. Gollhofer
Registration No. 31,106

700 Eleventh Street, NW, Suite 500
Washington, D.C. 20001
(202) 434-1500

ABSTRACT

METHOD AND SYSTEM FOR TRANSMITTING DATA FROM A TRANSMITTER TO A RECEIVER, AND TRANSMITTER AND RECEIVER THEREFOR

Data from a transmitter is extended to include authentication data on the application level by an application protocol. The authentication data is used by the receiver to determine whether the transmitter is known by the receiver. If the transmitter is known by the receiver, the data is accepted. If not, the data is rejected.

10/181564
18 JUL 2002

MARKED-UP COPY OF SUBSTITUTE SPECIFICATION

[Description] TITLE OF THE INVENTION

METHOD[,] AND SYSTEM FOR TRANSMITTING DATA FROM A TRANSMITTER TO A RECEIVER, AND TRANSMITTER AND RECEIVER THEREFOR

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application is based on and hereby claims priority to German Application No. 10001855.6 filed on 18 January 2000, the contents of which are hereby incorporated by reference.

BACKGROUND OF THE INVENTION

[0002] The invention relates to a method and a system and for transmitting data from a transmitter to a receiver, and to the transmitter and the receiver therefor.

[0003] The transmission of data is known, by way of example, on the basis of the OSI reference model [1] as described in Andrew S. Tanenbaum, Computer-Netzwerke; Wolfram's Fachverlag, Attenkirchen 1992, pages 17 - 32. The OSI reference model (OSI model for short) from the International Standards Organization (ISO) [comprises] has seven layers, each of which has a different functionality (in terms of abstraction). In the OSI model, layer 1 corresponds to a physical layer, where data and messages are transmitted from the transmitter to the receiver using a physical protocol. In layer 2, a data link layer is provided, and a data link protocol is used for transmission from layer 2 of the transmitter to layer 2 of the receiver. In a similar manner, layer 3, a network layer, uses a network protocol between transmitter and receiver, and layer 4, a transport layer, uses a transport protocol. In the OSI model, layer 5 corresponds to a session layer using a session protocol, layer 6 corresponds to a presentation layer using a presentation protocol, and layer 7 corresponds to an application layer using an application protocol. In practice, many applications do not always permit exact nomination of strict boundaries specifically between the upper protocol layers, particularly layers 5 to 7. By way of example, within the context of Internet telephony, i.e. use of the telephone service using the medium of the Internet, the three layers above the transport layer 4 combine to form an "application layer" for which, in a similar way to in the implementation above, an "application protocol" is used.

[0004] A special feature of the OSI model and hence of all [communications] communication models following the OSI model is that, as a result of the division into layers, in each layer the bottom layers perform functions fully transparently with regard to the currently considered layer and provide this upper layer with a service which is determined by all of the functionalities of the bottom layers. In this context, "transparent" means that currently considered layer does not need to concern itself with the functionalities of the bottom layers. In the case discussed here, the application protocol can thus be used between an application layer transmitter and an application layer receiver. This may cover numerous services, for example for switching, for protection or for actual transmission via a physical channel; these need no longer be a concern from the point of view of the application protocol, particularly as a user of this protocol.

[0005] Similarly with respect to this consideration, the individual layers have "protocol data units" (PDUs) between them which can be designed specifically for each protocol of the respective layer. Thus, for each layer, the respective protocol can comprise a dedicated header containing administrative information for the respective layer's protocol, with this header also being able to be seen and used only by the respective protocol's layer in accordance with the OSI model. Details about the design of the OSI model can be found in numerous pieces of literature, inter alia in those cited above.

[0006] When referring to application layer below, this denotes the layers above the transport layer (layers above 4). The application protocol also denotes the protocol for communication between a transmitter application layer situated above the transport layer and a receiver application layer situated above the transport layer.

[0007] A message authentication code (MAC) is known generally and denotes a cryptographic checksum which is intended to be used to identify an alteration in a message or in data (see [2] Christoph Ruland, Informationssicherheit in Datennetzen (which translates as Information Security in Data Networks), DATACOM-Verlag; Bergheim, 1993, pp.61[ff] - 63 and 68 - 79).

[0008] A one-way hash function is known from [2] Ruland, pp. 68[ff.] - 79, for example. Such a one-way hash function cannot be used to calculate the correct input value for a given function value. Another feature is the absence of collision, i.e. it must not be possible to use the one-way hash function to obtain the same output value for two different input values.

[0009] One difference between the hash code and the MAC is that the MAC requires a secret key in order to calculate it, whereas the hash function can be independent of a key and can be known publicly.

[0010] In addition, an asymmetric cryptographic method (also: public key method) is known, e.g. from [2] Ruland, pp. 73 [ff] - 79. Each party involved in an asymmetric cryptographic method receives two keys, a public key and a secret (or private) key. In principle, the secret key can be derived from the public key, with this task needing to be as complex as possible. The asymmetric method can also be used to produce an electronic signature (authentication) and/or to encrypt the content of a message (using the receiver's public key) such that only the receiver can decrypt the message again using its secret key.

[0011] Finally, there is also a symmetric method, which requires a (secret) key used both for encryption and decryption. An example of a symmetric encryption method is the DES algorithm.

[0012] A [communications] communication system is subject to a multiplicity of possible attacks which can target, inter alia, the content of the messages interchanged in the [communications] communication system or the availability of the [communications] communication system. Taking the example of Internet telephony, then it is firstly important for the content of the telephone conversation not to be able to be monitored by an unauthorized third party, and secondly it is also necessary for the third party not to be able to initiate countless calls, and hence for him not to interfere with the receiver and unnecessarily encumber the [communications] communication system. Such attacks are also called denial-of-service attacks. Possible examples of such attacks are mass data or mass messages which are automatically generated and sent to one or more receivers, where they significantly encumber availability and performance. In the Internet telephony example, it is at least unwanted for an Internet telephone to ring incessantly and hence not only to load the network unnecessarily but also to distract the receiver's attention. It is likewise perceived to be disruptive if an attacker interferes with the flow of communication between transmitter and receiver by sending unauthorized voice data.

[0013] To fend off attacks on a private [communications] communication network, "firewalls" are often used which ensure, in particular, that the private [communications] communication network is separated from public networks. However, it is a simple matter for an attacker to send his meaningless (mass) data to addressees within the private network as well, that is to

say behind the firewall. There, these data are decoded and possibly reproduced. [By this means] Thus, the effect achieved by the attacker is that, by way of example, the reproduction of such meaningless data produces nothing but annoying audio interference or noise and hence restricts the bandwidth and availability of the private network. In the extreme case, part or all of the private [communications] communication network can even crash.

[0014] In addition, an RTP protocol for transmitting media data ("payload"), i.e. video or audio data, is known [3] from H. Schulzrinne: RTP: A Transport Protocol for Real-Time Applications; Internet Engineering Task Force, RFC1889, 1996.

[0015] One approach to holding off unwanted data is provided by the IPSEC protocol [5] described S. Kent, R. Atkinson; IP Authentication Header, Internet Engineering Task Force, RFC2402, 1998 and [6] S. Kent, R. Atkinson; IP Encapsulating Security Payload (ESP), Internet Engineering Task Force, RFC2406, 1998. In this case, data packets of the Internet protocol (IP data packets) can be encapsulated and can be protected in terms of confidentiality and/or integrity (implicit sender authentication). In addition, the IPSEC protocol affords a key management method using a "cookie" mechanism (see [4] D. Harkins, D. Carrel, The Internet Key Exchange (IKE), Internet Engineering Task Force, RFC2409, 1998), which can be used to fend off the mass data attacks (denial-of-service attacks) discussed above during the key management phase. The cookie mechanism involves linking fast one-way hash functions (e.g. SHA-1), random numbers and IP addresses to one another. However, the first "cookie" transmitted from the transmitter to the receiver is not protected, which results in a security gap. In addition, the IPSEC cookie method is not suitable for ensuring protection against such unwanted mass data on the application layer (for the application protocol) under real-time conditions as well. However, the OSI layer structure means that a drawback for IPSEC is, among other things, that it is not possible to link IPSEC to the security functions of an application layer in this manner.

SUMMARY OF THE INVENTION

[0016] [The] An object of the invention is to transmit data from a transmitter to a receiver, with the receiver ensuring that the data are not unwanted data arising from a denial-of-service attack, for example. [This object is achieved in accordance with the features of the independent patent claims.]

[0017] First, the object is achieved by specifying a method for transmitting data from a transmitter to a receiver, in which the transmitter extends the data by authentication data using an application protocol on the application layer. The authentication data are used by the receiver to ascertain whether it knows the transmitter. If the receiver knows the transmitter, the data are accepted, otherwise the data are rejected.

[0018] As stated above, depending on the protocol structure (cf. OSI model), data in each protocol layer have their own transparency with regard to this protocol layer, i.e. the services in protocol layers beneath are performed (without being seen by the current protocol layer). Advantageously, data packets on the application layer, in particular, are free of the administrative data associated with layers beneath, i.e. the data packets in the application layer have, besides the administrative information for the application layer, only the data which are actually to be communicated. Accordingly, authentication in the application layer is particularly advantageous, because the data packets themselves, in contrast to layers beneath, are of significantly reduced size. This small size has an advantageous effect on real time and availability of the entire [communications] communication system. If the receiver knows the transmitter, i.e. particularly if the transmitter is successfully authenticated with respect to the receiver, the message from the transmitter is accepted, in particular the data are stored. Otherwise, i.e. if the receiver does not know the transmitter, the data are rejected, i.e. buffering does not take place. This is particularly advantageous if such a decision is made automatically.

[0019] In particular, available security functions and keys on the application layer can advantageously be allocated to the individual application users. This can advantageously be linked to the application data in a filter function outlined above. Special functions at the application level, such as picture and/or sound compression methods, can advantageously be combined with security functions in the application layer, which makes it possible to increase the end systems' performance and reduces the implementation complexity; this also applies to the aforementioned filtering mechanism.

[0020] The described authentication mechanism in the application layer allows transmitters and receivers which know one another to be separated from transmitters which are not known by the receiver and from which the receiver does not accept any messages either. This allows an attack of the type described in the introduction ("denial-of-service" attack) to be effectively prevented, i.e. unwanted mass data are actually rejected when arriving at the transmitter.

[0021] In this context, special note will be made that the receiver in no way has to be an end receiver or addressee. Instead, the receiver itself can be a switching entity, e.g. a switching node or a firewall with switching functionality, and hence can act with regard to authentication of the transmitter for the end receiver. Thus, in relation to the "firewall" example, it is possible to prevent superfluous network load from being produced by rejecting the unwanted mass data before they even enter the private [communications] communication network.

[0022] In this scenario (authentication in a switching entity), it is advantageous, in particular, if the authentication data are generated using an asymmetric method, since it is also possible for the switching entity, in the example the firewall, to ascertain from whom the data are coming, without the decryption secret already needing to be known in the firewall (as in the case of symmetric encryption) for this purpose. The firewall rejects unwanted mass data if it does not know the transmitter; in the other case, that is to say if the switching entity knows the transmitter and the data are by no means unwanted, these data are forwarded to the receiver after the transmitter has been verified. The receiver can still use its private key to decrypt the data and process them further locally (display them, store them, etc.) irrespective of the check on the data's origin, which has already taken place in the firewall. It will be stressed once again that the decryption (in terms of confidentiality) of the data is performed by the addressee using its private key in the asymmetric method described, whereas the origin of the message (which is determined for this addressee) can also be checked in the switching entity using the public key of the sender – irrespective of the content of the message. Thus, an upstream filter functionality can be provided successfully by the switching entity. This results in a clear functional division between the filter function and the application function; the complexity of the terminals can be simplified as a result and the network traffic in the private [communications] communication network can be reduced.

[0023] One development is that the data are transmitted on a packet-oriented basis. Another development is that the authentication data are determined by virtue of at least part of a protocol data unit, available on the application layer, of the application protocol being encrypted by the transmitter using a secret. In this case, the secret between the transmitter and the receiver can be a key for symmetric encryption or a key pair for asymmetric encryption (public key method, see description with switching entity, for example). Decryption of the secret determines, particularly at the receiver end, whether or not the receiver knows the transmitter. This is advantageous if data encryption is available on the application level anyway and can also be used for this purpose.

[0024] One refinement is that the authentication data are determined using part of the protocol data unit available on the application layer, in particular a sequence number [and/or] or a timestamp [lacuna] [comprises] [lacuna].

[0025] In particular, one development is that a one-way hash function is used for protection. Protection can also be effected using a message authentication code (MAC), with a key which is known only to the transmitter and to the receiver additionally being necessary.

[0026] Another refinement is that, before the data are transmitted from the transmitter to the receiver, authentication is carried out between the transmitter and the receiver.

[0027] Another refinement is that the described method is used in packet-switching telephone services, particularly within the context of Internet telephony. Alternatively, the method can be used in switching nodes and switching installations.

[0028] The object is achieved in another way by specifying a system for transmitting data, in which a transmitter and a receiver are provided, the transmitter extending the data by authentication data using an application protocol on the application layer. The receiver uses the authentication data to check whether it knows the transmitter. If the receiver knows the transmitter, the data from the transmitter are used, otherwise the receiver rejects the data from the transmitter which is unknown to it.

[0029] The object is also achieved by specifying a transmitter for sending data to a receiver, which extends the data by authentication data using an application protocol on the application layer and sends them to the receiver.

[0030] Finally, the object is achieved by specifying a receiver for receiving data from a transmitter designed on the basis of the statements above, in particular, the receiver using authentication data within the application protocol on the application layer to determine whether it knows the transmitter. If it knows the transmitter, it uses the data, otherwise the receiver rejects the data. The system or the transmitter and the receiver are particularly suitable for carrying out the inventive method or one of its developments explained above.

[0031] [Exemplary embodiments are illustrated and explained below with reference to the drawings, in which:] Reference will now be made in detail to the preferred embodiments of the present invention, examples of which are illustrated in the accompanying drawings, wherein like reference numerals refer to like elements throughout.

BRIEF DESCRIPTION OF THE DRAWINGS

[0032] These and other objects and advantages of the present invention will become more apparent and more readily appreciated from the following description of the preferred embodiments, taken in conjunction with the accompanying drawings of which:

[figure] Figure 1 [shows] is a block diagram of a protocol architecture (protocol stack) for a [communications] communication system in the layer model;

[figure] Figure 2 [shows] is a block diagram of an application layer with possible protocols;

[figure] Figure 3 [shows] is a flowchart between the transmitter and the receiver for transmitting data from the transmitter to the receiver;

[figure] Figure 4 [shows] is a protocol data unit (PDU) of the application protocol (layer > 5) with authentication data.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0033] Figure 1 [shows] is a block diagram of a protocol architecture (protocol stack) for a [communications] communication system in the layer model, following the OSI model mentioned [in the introduction] above. [To this end, it shows a] A transmitter 104 and a receiver 105 [which] interchange messages via a physical connection 103. Both the transmitter 104 and the receiver 105 have the same protocol architecture: layer 1 (110 and 111) represents a physical layer, layer 2 (109, 112) represents a data link layer, layer 3 (108, 113) represents a network layer, layer 4 (107, 114) represents a transport layer and layer 5 (101 106) represents an application layer. Between the individual layers there is a respective dedicated protocol from a transmitter 104 to a receiver 105; for the application layer 101 or 106, this is the application protocol 102, for example. From the point of view of layer 5 (101, 106), protocol data units having the format prescribed in the application protocol 102 are transmitted from the transmitter 104 to the receiver 105. The layers situated below the application layer represent functionalities which are provided transparently for the application layer. On the basis of the OSI model, the functionalities have expediently been chosen to be largely independent of one another.

[0034] Figure 2 [shows] is a block diagram of an application layer with possible protocols, based on an application concerning the transmission of media data (audio and video data). Thus, 101 and 106 represent the application layer (layer 5) [already known from figure] shown in Fig. 1 for the transmitter 104 and the receiver 105. This application layer is in turn divided into a

plurality of sublayers. Thus, directly above the layer 4 is the real time transport protocol (RTP) 201. This protocol allows voice and/or picture data to be transmitted in real time from one to possibly a plurality of points (addressees). A layer 206 contains a security application which compiles security functions of the application layer; associated security protocols provide for services such as

- User authentication,
- Access control
- Confidentiality,
- Integrity
- Binding nature of the application data, and
- Accounting.

[0035] Depending on the instance of application, audio data 203 and/or video data 202 are used. For both alternatives, there are different compression standards, a few of which are shown by way of example in [figure] Fig. 2. For [the] audio data, these are the standards G.711, G.722, G.723.1 and G.729. For [the] video data, these are the two picture compression standards H.261 and H.263. In the next abstraction stage (similarly with respect to the OSI model) come the different possibilities for an application of audio or video data after compression and decompression (see block 204). One more abstraction level above, there is, by way of example, a user interface 205, which fully transparently provides a user with the services beneath which have just been described. By way of example, a user, taking into account both the audio and the video data, can thus make use of video telephony, e.g. over the Internet, without specifically needing to concern himself with any details of the services provided by the layers beneath. He thus uses the video telephony service, visible to him, over the Internet transparently. In this context, various alternative implementation forms can be situated below his service.

[0036] Figure 3 [shows] is a flowchart [between the transmitter and the receiver] for transmitting data from the transmitter 308 to the receiver 307. In [a step] block 301, authentication data are produced on the application layer, and the protocol data unit (PDU) relevant to the application protocol is appended. In accordance with the arrangements of the application protocol, the data, including the authentication data, are transmitted to the receiver 307 (see connection 302). This is done using the functionalities or services available under the

application protocol. In [a] block 303, the authentication data are checked in the receiver 307 and there is a test (in [a] block 304) to determine whether the transmitter 308 is known to the receiver 307. If this is not the case, the procedure branches to [a] block 305 and the data are rejected. If the transmitter 308 is known to the receiver 307, the procedure branches to [a] block 306 and the data are processed further. The authentication between transmitter and receiver was successful, and the transmitted data are not unwanted mass data.

[0037] Figure 4 shows a protocol data unit (PDU) of the application protocol with authentication data. What is shown in [figure] Fig. 4 is a protocol data unit (PDU) of the real time transport protocol (RTP, see 201 in [figure] Fig. 2) 401. Such an RTP packet 401 comprises an RTP header 406, encrypted media data 404 and authentication data 405. The RTP header 406 comprises, inter alia, a sequence number 402 and a timestamp 403. Both the transmitter and the receiver know a shared secret K, in this case indicated as a key, which is used to generate a message authentication code (MAC), on the basis of the sequence number 402 and the timestamp 403 (see field 405). An advantage with respect to encryption using a DES algorithm, for example, is a field length of 64 bits.

[0038] As already described, such a packet sent by the transmitter is authenticated at the receiver end such that (in relation to the application layer) a message authentication code is generated by field 402 (sequence number) and field 403 (timestamp) on the basis of the key (K) known to the receiver. If this message authentication code is the same as the field 405, then the arriving RTP packet 401 is a data packet coming from a known transmitter, and the data are processed, for example displayed or stored. In the other case, unauthenticated data are involved, the transmitter is unknown to the receiver and the entire RTP packet is rejected.

[0039] [Literature references]

[1] [Andrew S. Tanenbaum: Computer-Netzwerke;] [Computer networks] [Wolfram's Fachverlag; Attenkirchen 1992; ISBN 3-925328-79-3; pages 17 – 32.]

[2] [Christoph Ruland: Informationssicherheit in Datennetzen;] [Information security in data networks] [DATACOM-Verlag; Bergheim 1993; ISBN 3-89238-081-3; pages 61 – 63 and 68 – 79.]

[3] [RFC1889] [H. Schulzrinne: RTP: A Transport Protocol for Real-Time Applications; Internet Engineering Task Force, 1996]

[4] [RFC2409] [The Internet Key Exchange (IKE), D. Harkins, D. Carrel,; Internet Engineering Task Force, 1998.]

[5] [RFC2402] [IP Authentication Header, S. Kent, R. Atkinson; Internet Engineering Task Force, 1998.]

[6] [RFC2406] [IP Encapsulating Security Payload (ESP), S.Kent, R. Atkinson; Internet Engineering Task Force, 1998.]

[0040] The invention has been described in detail with particular reference to preferred embodiments thereof and examples, but it will be understood that variations and modifications can be effected within the spirit and scope of the invention.

10181564 071802
10/181564
512 Rec'd PCT/PTO 18 JUL 2002

SUBSTITUTE SPECIFICATION

TITLE OF THE INVENTION

METHOD AND SYSTEM FOR TRANSMITTING DATA FROM A TRANSMITTER TO A RECEIVER, AND TRANSMITTER AND RECEIVER THEREFOR

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application is based on and hereby claims priority to German Application No. 10001855.6 filed on 18 January 2000, the contents of which are hereby incorporated by reference.

BACKGROUND OF THE INVENTION

[0002] The invention relates to a method and a system and for transmitting data from a transmitter to a receiver, and to the transmitter and the receiver therefor.

[0003] The transmission of data is known, by way of example, on the basis of the OSI reference model as described in Andrew S. Tanenbaum, Computer-Netzwerke; Wolfram's Fachverlag, Attenkirchen 1992, pages 17 - 32. The OSI reference model (OSI model for short) from the International Standards Organization (ISO) has seven layers, each of which has a different functionality (in terms of abstraction). In the OSI model, layer 1 corresponds to a physical layer, where data and messages are transmitted from the transmitter to the receiver using a physical protocol. In layer 2, a data link layer is provided, and a data link protocol is used for transmission from layer 2 of the transmitter to layer 2 of the receiver. In a similar manner, layer 3, a network layer, uses a network protocol between transmitter and receiver, and layer 4, a transport layer, uses a transport protocol. In the OSI model, layer 5 corresponds to a session layer using a session protocol, layer 6 corresponds to a presentation layer using a presentation protocol, and layer 7 corresponds to an application layer using an application protocol. In practice, many applications do not always permit exact nomination of strict boundaries specifically between the upper protocol layers, particularly layers 5 to 7. By way of example, within the context of Internet telephony, i.e. use of the telephone service using the medium of the Internet, the three layers above the transport layer 4 combine to form an "application layer" for which, in a similar way to in the implementation above, an "application protocol" is used.

[0004] A special feature of the OSI model and hence of all communication models following the OSI model is that, as a result of the division into layers, in each layer the bottom layers perform functions fully transparently with regard to the currently considered layer and provide this upper layer with a service which is determined by all of the functionalities of the bottom layers. In this context, "transparent" means that currently considered layer does not need to concern itself with the functionalities of the bottom layers. In the case discussed here, the application protocol can thus be used between an application layer transmitter and an application layer receiver. This may cover numerous services, for example for switching, for protection or for actual transmission via a physical channel; these need no longer be a concern from the point of view of the application protocol, particularly as a user of this protocol.

[0005] Similarly with respect to this consideration, the individual layers have "protocol data units" (PDUs) between them which can be designed specifically for each protocol of the respective layer. Thus, for each layer, the respective protocol can comprise a dedicated header containing administrative information for the respective layer's protocol, with this header also being able to be seen and used only by the respective protocol's layer in accordance with the OSI model. Details about the design of the OSI model can be found in numerous pieces of literature, inter alia in those cited above.

[0006] When referring to application layer below, this denotes the layers above the transport layer (layers above 4). The application protocol also denotes the protocol for communication between a transmitter application layer situated above the transport layer and a receiver application layer situated above the transport layer.

[0007] A message authentication code (MAC) is known generally and denotes a cryptographic checksum which is intended to be used to identify an alteration in a message or in data (see Christoph Ruland, Informationssicherheit in Datennetzen (which translates as Information Security in Data Networks), DATACOM-Verlag; Bergheim, 1993, pp.61 - 63 and 68 - 79).

[0008] A one-way hash function is known from Ruland, pp. 68 - 79, for example. Such a one-way hash function cannot be used to calculate the correct input value for a given function value. Another feature is the absence of collision, i.e. it must not be possible to use the one-way hash function to obtain the same output value for two different input values.

[0009] One difference between the hash code and the MAC is that the MAC requires a secret key in order to calculate it, whereas the hash function can be independent of a key and can be known publicly.

[0010] In addition, an asymmetric cryptographic method (also: public key method) is known, e.g. from Ruland, pp. 73 - 79. Each party involved in an asymmetric cryptographic method receives two keys, a public key and a secret (or private) key. In principle, the secret key can be derived from the public key, with this task needing to be as complex as possible. The asymmetric method can also be used to produce an electronic signature (authentication) and/or to encrypt the content of a message (using the receiver's public key) such that only the receiver can decrypt the message again using its secret key.

[0011] Finally, there is also a symmetric method, which requires a (secret) key used both for encryption and decryption. An example of a symmetric encryption method is the DES algorithm.

[0012] A communication system is subject to a multiplicity of possible attacks which can target, inter alia, the content of the messages interchanged in the communication system or the availability of the communication system. Taking the example of Internet telephony, then it is firstly important for the content of the telephone conversation not to be able to be monitored by an unauthorized third party, and secondly it is also necessary for the third party not to be able to initiate countless calls, and hence for him not to interfere with the receiver and unnecessarily encumber the communication system. Such attacks are also called denial-of-service attacks. Possible examples of such attacks are mass data or mass messages which are automatically generated and sent to one or more receivers, where they significantly encumber availability and performance. In the Internet telephony example, it is at least unwanted for an Internet telephone to ring incessantly and hence not only to load the network unnecessarily but also to distract the receiver's attention. It is likewise perceived to be disruptive if an attacker interferes with the flow of communication between transmitter and receiver by sending unauthorized voice data.

[0013] To fend off attacks on a private communication network, "firewalls" are often used which ensure, in particular, that the private communication network is separated from public networks. However, it is a simple matter for an attacker to send his meaningless (mass) data to addressees within the private network as well, that is to say behind the firewall. There, these data are decoded and possibly reproduced. Thus, the effect achieved by the attacker is that, by

way of example, the reproduction of such meaningless data produces nothing but annoying audio interference or noise and hence restricts the bandwidth and availability of the private network. In the extreme case, part or all of the private communication network can even crash.

[0014] In addition, an RTP protocol for transmitting media data ("payload"), i.e. video or audio data, is known from H. Schulzrinne: RTP: A Transport Protocol for Real-Time Applications; Internet Engineering Task Force, RFC1889, 1996.

[0015] One approach to holding off unwanted data is provided by the IPSEC protocol described S. Kent, R. Atkinson; IP Authentication Header, Internet Engineering Task Force, RFC2402, 1998 and S. Kent, R. Atkinson; IP Encapsulating Security Payload (ESP), Internet Engineering Task Force, RFC2406, 1998. In this case, data packets of the Internet protocol (IP data packets) can be encapsulated and can be protected in terms of confidentiality and/or integrity (implicit sender authentication). In addition, the IPSEC protocol affords a key management method using a "cookie" mechanism (see D. Harkins, D. Carrel, The Internet Key Exchange (IKE), Internet Engineering Task Force, RFC2409, 1998), which can be used to fend off the mass data attacks (denial-of-service attacks) discussed above during the key management phase. The cookie mechanism involves linking fast one-way hash functions (e.g. SHA-1), random numbers and IP addresses to one another. However, the first "cookie" transmitted from the transmitter to the receiver is not protected, which results in a security gap. In addition, the IPSEC cookie method is not suitable for ensuring protection against such unwanted mass data on the application layer (for the application protocol) under real-time conditions as well. However, the OSI layer structure means that a drawback for IPSEC is, among other things, that it is not possible to link IPSEC to the security functions of an application layer in this manner.

SUMMARY OF THE INVENTION

[0016] An object of the invention is to transmit data from a transmitter to a receiver, with the receiver ensuring that the data are not unwanted data arising from a denial-of-service attack, for example.

[0017] First, the object is achieved by specifying a method for transmitting data from a transmitter to a receiver, in which the transmitter extends the data by authentication data using an application protocol on the application layer. The authentication data are used by the

receiver to ascertain whether it knows the transmitter. If the receiver knows the transmitter, the data are accepted, otherwise the data are rejected.

[0018] As stated above, depending on the protocol structure (cf. OSI model), data in each protocol layer have their own transparency with regard to this protocol layer, i.e. the services in protocol layers beneath are performed (without being seen by the current protocol layer). Advantageously, data packets on the application layer, in particular, are free of the administrative data associated with layers beneath, i.e. the data packets in the application layer have, besides the administrative information for the application layer, only the data which are actually to be communicated. Accordingly, authentication in the application layer is particularly advantageous, because the data packets themselves, in contrast to layers beneath, are of significantly reduced size. This small size has an advantageous effect on real time and availability of the entire communication system. If the receiver knows the transmitter, i.e. particularly if the transmitter is successfully authenticated with respect to the receiver, the message from the transmitter is accepted, in particular the data are stored. Otherwise, i.e. if the receiver does not know the transmitter, the data are rejected, i.e. buffering does not take place. This is particularly advantageous if such a decision is made automatically.

[0019] In particular, available security functions and keys on the application layer can advantageously be allocated to the individual application users. This can advantageously be linked to the application data in a filter function outlined above. Special functions at the application level, such as picture and/or sound compression methods, can advantageously be combined with security functions in the application layer, which makes it possible to increase the end systems' performance and reduces the implementation complexity; this also applies to the aforementioned filtering mechanism.

[0020] The described authentication mechanism in the application layer allows transmitters and receivers which know one another to be separated from transmitters which are not known by the receiver and from which the receiver does not accept any messages either. This allows an attack of the type described in the introduction ("denial-of-service" attack) to be effectively prevented, i.e. unwanted mass data are actually rejected when arriving at the transmitter.

[0021] In this context, special note will be made that the receiver in no way has to be an end receiver or addressee. Instead, the receiver itself can be a switching entity, e.g. a switching node or a firewall with switching functionality, and hence can act with regard to authentication of

the transmitter for the end receiver. Thus, in relation to the "firewall" example, it is possible to prevent superfluous network load from being produced by rejecting the unwanted mass data before they even enter the private communication network.

[0022] In this scenario (authentication in a switching entity), it is advantageous, in particular, if the authentication data are generated using an asymmetric method, since it is also possible for the switching entity, in the example the firewall, to ascertain from whom the data are coming, without the decryption secret already needing to be known in the firewall (as in the case of symmetric encryption) for this purpose. The firewall rejects unwanted mass data if it does not know the transmitter; in the other case, that is to say if the switching entity knows the transmitter and the data are by no means unwanted, these data are forwarded to the receiver after the transmitter has been verified. The receiver can still use its private key to decrypt the data and process them further locally (display them, store them, etc.) irrespective of the check on the data's origin, which has already taken place in the firewall. It will be stressed once again that the decryption (in terms of confidentiality) of the data is performed by the addressee using its private key in the asymmetric method described, whereas the origin of the message (which is determined for this addressee) can also be checked in the switching entity using the public key of the sender – irrespective of the content of the message. Thus, an upstream filter functionality can be provided successfully by the switching entity. This results in a clear functional division between the filter function and the application function; the complexity of the terminals can be simplified as a result and the network traffic in the private communication network can be reduced.

[0023] One development is that the data are transmitted on a packet-oriented basis. Another development is that the authentication data are determined by virtue of at least part of a protocol data unit, available on the application layer, of the application protocol being encrypted by the transmitter using a secret. In this case, the secret between the transmitter and the receiver can be a key for symmetric encryption or a key pair for asymmetric encryption (public key method, see description with switching entity, for example). Decryption of the secret determines, particularly at the receiver end, whether or not the receiver knows the transmitter. This is advantageous if data encryption is available on the application level anyway and can also be used for this purpose.

[0024] One refinement is that the authentication data are determined using part of the protocol data unit available on the application layer, in particular a sequence number or a timestamp .

[0025] In particular, one development is that a one-way hash function is used for protection. Protection can also be effected using a message authentication code (MAC), with a key which is known only to the transmitter and to the receiver additionally being necessary.

[0026] Another refinement is that, before the data are transmitted from the transmitter to the receiver, authentication is carried out between the transmitter and the receiver.

[0027] Another refinement is that the described method is used in packet-switching telephone services, particularly within the context of Internet telephony. Alternatively, the method can be used in switching nodes and switching installations.

[0028] The object is achieved in another way by specifying a system for transmitting data, in which a transmitter and a receiver are provided, the transmitter extending the data by authentication data using an application protocol on the application layer. The receiver uses the authentication data to check whether it knows the transmitter. If the receiver knows the transmitter, the data from the transmitter are used, otherwise the receiver rejects the data from the transmitter which is unknown to it.

[0029] The object is also achieved by specifying a transmitter for sending data to a receiver, which extends the data by authentication data using an application protocol on the application layer and sends them to the receiver.

[0030] Finally, the object is achieved by specifying a receiver for receiving data from a transmitter designed on the basis of the statements above, in particular, the receiver using authentication data within the application protocol on the application layer to determine whether it knows the transmitter. If it knows the transmitter, it uses the data, otherwise the receiver rejects the data. The system or the transmitter and the receiver are particularly suitable for carrying out the inventive method or one of its developments explained above.

[0031] Reference will now be made in detail to the preferred embodiments of the present invention, examples of which are illustrated in the accompanying drawings, wherein like reference numerals refer to like elements throughout.

BRIEF DESCRIPTION OF THE DRAWINGS

[0032] These and other objects and advantages of the present invention will become more apparent and more readily appreciated from the following description of the preferred embodiments, taken in conjunction with the accompanying drawings of which:

Figure 1 is a block diagram of a protocol architecture (protocol stack) for a communication system in the layer model;

Figure 2 is a block diagram of an application layer with possible protocols;

Figure 3 is a flowchart between the transmitter and the receiver for transmitting data from the transmitter to the receiver;

Figure 4 is a protocol data unit (PDU) of the application protocol (layer > 5) with authentication data.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0033] Figure 1 is a block diagram of a protocol architecture (protocol stack) for a communication system in the layer model, following the OSI model mentioned above. A transmitter 104 and a receiver 105 interchange messages via a physical connection 103. Both the transmitter 104 and the receiver 105 have the same protocol architecture: layer 1 (110 and 111) represents a physical layer, layer 2 (109, 112) represents a data link layer, layer 3 (108, 113) represents a network layer, layer 4 (107, 114) represents a transport layer and layer 5 (101 106) represents an application layer. Between the individual layers there is a respective dedicated protocol from a transmitter 104 to a receiver 105; for the application layer 101 or 106, this is the application protocol 102, for example. From the point of view of layer 5 (101, 106), protocol data units having the format prescribed in the application protocol 102 are transmitted from the transmitter 104 to the receiver 105. The layers situated below the application layer represent functionalities which are provided transparently for the application layer. On the basis of the OSI model, the functionalities have expediently been chosen to be largely independent of one another.

[0034] Figure 2 is a block diagram of an application layer with possible protocols, based on an application concerning the transmission of media data (audio and video data). Thus, 101 and 106 represent the application layer (layer 5) shown in Fig. 1 for the transmitter 104 and the receiver 105. This application layer is in turn divided into a plurality of sublayers. Thus, directly above the layer 4 is the real time transport protocol (RTP) 201. This protocol allows voice

and/or picture data to be transmitted in real time from one to possibly a plurality of points (addressees). A layer 206 contains a security application which compiles security functions of the application layer; associated security protocols provide for services such as

- User authentication,
- Access control
- Confidentiality,
- Integrity
- Binding nature of the application data, and
- Accounting.

[0035] Depending on the instance of application, audio data 203 and/or video data 202 are used. For both alternatives, there are different compression standards, a few of which are shown by way of example in Fig. 2. For audio data, these are the standards G.711, G.722, G.723.1 and G.729. For video data, these are the two picture compression standards H.261 and H.263. In the next abstraction stage (similarly with respect to the OSI model) come the different possibilities for an application of audio or video data after compression and decompression (see block 204). One more abstraction level above, there is, by way of example, a user interface 205, which fully transparently provides a user with the services beneath which have just been described. By way of example, a user, taking into account both the audio and the video data, can thus make use of video telephony, e.g. over the Internet, without specifically needing to concern himself with any details of the services provided by the layers beneath. He thus uses the video telephony service, visible to him, over the Internet transparently. In this context, various alternative implementation forms can be situated below his service.

[0036] Figure 3 is a flowchart for transmitting data from the transmitter 308 to the receiver 307. In block 301, authentication data are produced on the application layer, and the protocol data unit (PDU) relevant to the application protocol is appended. In accordance with the arrangements of the application protocol, the data, including the authentication data, are transmitted to the receiver 307 (see connection 302). This is done using the functionalities or services available under the application protocol. In block 303, the authentication data are checked in the receiver 307 and there is a test (in block 304) to determine whether the transmitter 308 is known to the receiver 307. If this is not the case, the procedure branches to

block 305 and the data are rejected. If the transmitter 308 is known to the receiver 307, the procedure branches to block 306 and the data are processed further. The authentication between transmitter and receiver was successful, and the transmitted data are not unwanted mass data.

[0037] Figure 4 shows a protocol data unit (PDU) of the application protocol with authentication data. What is shown in Fig. 4 is a protocol data unit (PDU) of the real time transport protocol (RTP, see 201 in Fig. 2) 401. Such an RTP packet 401 comprises an RTP header 406, encrypted media data 404 and authentication data 405. The RTP header 406 comprises, inter alia, a sequence number 402 and a timestamp 403. Both the transmitter and the receiver know a shared secret K, in this case indicated as a key, which is used to generate a message authentication code (MAC), on the basis of the sequence number 402 and the timestamp 403 (see field 405). An advantage with respect to encryption using a DES algorithm, for example, is a field length of 64 bits.

[0038] As already described, such a packet sent by the transmitter is authenticated at the receiver end such that (in relation to the application layer) a message authentication code is generated by field 402 (sequence number) and field 403 (timestamp) on the basis of the key (K) known to the receiver. If this message authentication code is the same as the field 405, then the arriving RTP packet 401 is a data packet coming from a known transmitter, and the data are processed, for example displayed or stored. In the other case, unauthenticated data are involved, the transmitter is unknown to the receiver and the entire RTP packet is rejected.

[0039] The invention has been described in detail with particular reference to preferred embodiments thereof and examples, but it will be understood that variations and modifications can be effected within the spirit and scope of the invention.

10/181564
512 Rec'd PCT/PTO 18 JUL 2002

3/prts

Description

Method, system for transmitting data from a transmitter to a receiver, and transmitter and receiver therefor

The invention relates to a method and a system and for transmitting data from a transmitter to a receiver, and to the transmitter and the receiver therefor.

The transmission of data is known, by way of example, on the basis of the OSI reference model [1]. The OSI reference model (OSI model for short) from the International Standards Organization (ISO) comprises seven layers, each of which has a different functionality (in terms of abstraction). In the OSI model, layer 1 corresponds to a physical layer, where data and messages are transmitted from the transmitter to the receiver using a physical protocol. In layer 2, a data link layer is provided, and a data link protocol is used for transmission from layer 2 of the transmitter to layer 2 of the receiver. In a similar manner, layer 3, a network layer, uses a network protocol between transmitter and receiver, and layer 4, a transport layer, uses a transport protocol. In the OSI model, layer 5 corresponds to a session layer using a session protocol, layer 6 corresponds to a presentation layer using a presentation protocol, and layer 7 corresponds to an application layer using an application protocol. In practice, many applications do not always permit exact nomination of strict boundaries specifically between the upper protocol layers, particularly layers 5 to 7. By way of example, within the context of Internet telephony, i.e. use of the telephone service using the medium of the Internet, the three layers above the transport layer 4 combine to form an "application layer" for which, in a similar way to in the implementation above, an "application protocol" is used.

A special feature of the OSI model and hence of all communications models following the OSI model is that, as a result of the division into layers, in each layer the bottom layers perform functions fully transparently with regard to the currently considered layer and provide this upper layer with a service which is determined by all of the functionalities of the bottom layers. In this context, "transparent" means that currently considered layer does not need to concern itself with the functionalities of the bottom layers. In the case discussed here, the application protocol can thus be used between an application layer transmitter and an application layer receiver. This may cover numerous services, for example for switching, for protection or for actual transmission via a physical channel; these need no longer be a concern from the point of view of the application protocol, particularly as a user of this protocol.

Similarly with respect to this consideration, the individual layers have "protocol data units" (PDUs) between them which can be designed specifically for each protocol of the respective layer. Thus, for each layer, the respective protocol can comprise a dedicated header containing administrative information for the respective layer's protocol, with this header also being able to be seen and used only by the respective protocol's layer in accordance with the OSI model. Details about the design of the OSI model can be found in numerous pieces of literature, inter alia in those cited above.

When referring to application layer below, this denotes the layers above the transport layer (layers above 4). The application protocol also denotes the protocol for communication between a transmitter application layer situated above the

transport layer and a receiver application layer situated above the transport layer.

A message authentication code (MAC) is known generally and denotes a cryptographic checksum which is intended to be used to identify an alteration in a message or in data (see [2], pp.61ff.).

A one-way hash function is known from [2], pp. 68ff., for example. Such a one-way hash function cannot be used to calculate the correct input value for a given function value. Another feature is the absence of collision, i.e. it must not be possible to use the one-way hash function to obtain the same output value for two different input values.

One difference between the hash code and the MAC is that the MAC requires a secret key in order to calculate it, whereas the hash function can be independent of a key and can be known publicly.

In addition, an asymmetric cryptographic method (also: public key method) is known, e.g. from [2], pp. 73 ff. Each party involved in an asymmetric cryptographic method receives two keys, a public key and a secret key. In principle, the secret key can be derived from the public key, with this task needing to be as complex as possible. The asymmetric method can also be used to produce an electronic signature (authentication) and/or to encrypt the content of a message (using the receiver's public key) such that only the receiver can decrypt the message again using its secret key.

Finally, there is also a symmetric method, which requires a (secret) key used both for encryption and decryption. An example of a

symmetric encryption method is the DES algorithm.

A communications system is subject to a multiplicity of possible attacks which can target, inter alia, the content of the messages interchanged in the communications system or the availability of the communications system. Taking the example of Internet telephony, then it is firstly important for the content of the telephone conversation not to be able to be monitored by an unauthorized third party, and secondly it is also necessary for the third party not to be able to initiate countless calls, and hence for him not to interfere with the receiver and unnecessarily encumber the communications system. Such attacks are also called denial-of-service attacks. Possible examples of such attacks are mass data or mass messages which are automatically generated and sent to one or more receivers, where they significantly encumber availability and performance. In the Internet telephony example, it is at least unwanted for an Internet telephone to ring incessantly and hence not only to load the network unnecessarily but also to distract the receiver's attention. It is likewise perceived to be disruptive if an attacker interferes with the flow of communication between transmitter and receiver by sending unauthorized voice data.

To fend off attacks on a private communications network, "firewalls" are often used which ensure, in particular, that the private communications network is separated from public networks. However, it is a simple matter for an attacker to send his meaningless (mass) data to addressees within the private network as well, that is to say behind the firewall. There, these data are decoded and possibly reproduced. By this means, the effect achieved by the attacker is that, by way of example, the reproduction of such meaningless data produces nothing but annoying audio interference or

noise and hence restricts the

bandwidth and availability of the private network. In the extreme case, part or all of the private communications network can even crash.

In addition, an RTP protocol for transmitting media data ("payload"), i.e. video or audio data, is known [3].

One approach to holding off unwanted data is provided by the IPSEC protocol [5] and [6]. In this case, data packets of the Internet protocol (IP data packets) can be encapsulated and can be protected in terms of confidentiality and/or integrity (implicit sender authentication). In addition, the IPSEC protocol affords a key management method using a "cookie" mechanism (see [4]), which can be used to fend off the mass data attacks (denial-of-service attacks) discussed above during the key management phase. The cookie mechanism involves linking fast one-way hash functions (e.g. SHA-1), random numbers and IP addresses to one another. However, the first "cookie" transmitted from the transmitter to the receiver is not protected, which results in a security gap. In addition, the IPSEC cookie method is not suitable for ensuring protection against such unwanted mass data on the application layer (for the application protocol) under real-time conditions as well. However, the OSI layer structure means that a drawback for IPSEC is, among other things, that it is not possible to link IPSEC to the security functions of an application layer in this manner.

The object of the invention is to transmit data from a transmitter to a receiver, with the receiver ensuring that the data are not unwanted data arising from a denial-of-service attack, for example.

This object is achieved in accordance with the features of the independent patent claims.

First, the object is achieved by specifying a method for transmitting data from a transmitter to a receiver, in which the transmitter extends the data by authentication data using an application protocol on the application layer. The authentication data are used by the receiver to ascertain whether it knows the transmitter. If the receiver knows the transmitter, the data are accepted, otherwise the data are rejected.

As stated above, depending on the protocol structure (cf. OSI model), data in each protocol layer have their own transparency with regard to this protocol layer, i.e. the services in protocol layers beneath are performed (without being seen by the current protocol layer). Advantageously, data packets on the application layer, in particular, are free of the administrative data associated with layers beneath, i.e. the data packets in the application layer have, besides the administrative information for the application layer, only the data which are actually to be communicated. Accordingly, authentication in the application layer is particularly advantageous, because the data packets themselves, in contrast to layers beneath, are of significantly reduced size. This small size has an advantageous effect on real time and availability of the entire communications system. If the receiver knows the transmitter, i.e. particularly if the transmitter is successfully authenticated with respect to the receiver, the message from the transmitter is accepted, in particular the data are stored. Otherwise, i.e. if the receiver does not know the transmitter, the data are rejected, i.e. buffering does not take place. This is particularly advantageous if such a decision is made automatically.

In particular, available security functions and keys on the application layer can advantageously be allocated to the individual application users. This can advantageously be linked to the application data in a filter function outlined above. Special functions at application level, such as picture and/or sound compression methods, can advantageously be combined with security functions in the application layer, which makes it possible to increase the end systems' performance and reduces the implementation complexity; this also applies to the aforementioned filtering mechanism.

The described authentication mechanism in the application layer allows transmitters and receivers which know one another to be separated from transmitters which are not known by the receiver and from which the receiver does not accept any messages either. This allows an attack of the type described in the introduction ("denial-of-service" attack) to be effectively prevented, i.e. unwanted mass data are actually rejected when arriving at the transmitter.

In this context, special note will be made that the receiver in no way has to be an end receiver or addressee. Instead, the receiver itself can be a switching entity, e.g. a switching node or a firewall with switching functionality, and hence can act with regard to authentication of the transmitter for the end receiver. Thus, in relation to the "firewall" example, it is possible to prevent superfluous network load from being produced by rejecting the unwanted mass data before they even enter the private communications network.

In this scenario (authentication in a switching entity), it is advantageous, in particular, if the authentication data are generated using an asymmetric method, since it is also possible for the switching

entity, in the example the firewall, to ascertain from whom the

data are coming, without the decryption secret already needing to be known in the firewall (as in the case of symmetric encryption) for this purpose. The firewall rejects unwanted mass data if it does not know the transmitter; in the other case, that is to say if the switching entity knows the transmitter and the data are by no means unwanted, these data are forwarded to the receiver after the transmitter has been verified. The receiver can still use its private key to decrypt the data and process them further locally (display them, store them, etc.) irrespective of the check on the data's origin, which has already taken place in the firewall. It will be stressed once again that the decryption (in terms of confidentiality) of the data is performed by the addressee using its private key in the asymmetric method described, whereas the origin of the message (which is determined for this addressee) can also be checked in the switching entity using the public key of the sender - irrespective of the content of the message. Thus, an upstream filter functionality can be provided successfully by the switching entity. This results in a clear functional division between the filter function and the application function; the complexity of the terminals can be simplified as a result and the network traffic in the private communications network can be reduced.

One development is that the data are transmitted on a packet-oriented basis.

Another development is that the authentication data are determined by virtue of at least part of a protocol data unit, available on the application layer, of the application protocol being encrypted by the transmitter using a secret. In this case, the secret between the transmitter and the receiver can be a key for symmetric encryption or a key pair for

asymmetric encryption (public key method, see description with switching entity, for example). Decryption of the secret determines, particularly at the receiver end, whether or not the receiver knows the transmitter. This is advantageous if data encryption is available on the application level anyway and can also be used for this purpose.

One refinement is that the authentication data are determined using part of the protocol data unit available on the application layer, in particular a sequence number and/or a timestamp [lacuna] comprises [lacuna].

In particular, one development is that a one-way hash function is used for protection. Protection can also be effected using a message authentication code (MAC), with a key which is known only to the transmitter and to the receiver additionally being necessary.

Another refinement is that, before the data are transmitted from the transmitter to the receiver, authentication is carried out between the transmitter and the receiver.

Another refinement is that the described method is used in packet-switching telephone services, particularly within the context of Internet telephony. Alternatively, the method can be used in switching nodes and switching installations.

The object is achieved in another way by specifying a system for transmitting data, in which a transmitter and a receiver are provided, the transmitter extending the data by authentication data using an application protocol on the application layer. The receiver uses the authentication data to check whether it knows

the transmitter. If the receiver knows the transmitter, the data from the transmitter are used, otherwise the receiver rejects the data from the transmitter which is unknown to it.

The object is also achieved by specifying a transmitter for sending data to a receiver, which extends the data by authentication data using an application protocol on the application layer and sends them to the receiver.

Finally, the object is achieved by specifying a receiver for receiving data from a transmitter designed on the basis of the statements above, in particular, the receiver using authentication data within the application protocol on the application layer to determine whether it knows the transmitter. If it knows the transmitter, it uses the data, otherwise the receiver rejects the data.

The system or the transmitter and the receiver are particularly suitable for carrying out the inventive method or one of its developments explained above.

Exemplary embodiments are illustrated and explained below with reference to the drawings, in which

figure 1 shows a protocol architecture (protocol stack) for a communications system in the layer model;

figure 2 shows an application layer with possible protocols;

figure 3 shows a flowchart between the transmitter and the receiver for transmitting data from the transmitter to the receiver;

figure 4 shows a protocol data unit (PDU) of the application protocol (layer > 5) with authentication data.

Figure 1 shows a protocol architecture (protocol stack) for a communications system in the layer model, following the OSI model mentioned in the introduction. To this end, it shows a transmitter 104 and a receiver 105 which interchange messages via a physical connection 103. Both the transmitter 104 and the receiver 105 have the same protocol architecture: layer 1 110 and 111 represents a physical layer, layer 2 109, 112 represents a data link layer, layer 3 108, 113 represents a network layer, layer 4 107, 114 represents a transport layer and layer 5 101 106 represents an application layer. Between the individual layers there is a respective dedicated protocol from a transmitter 104 to a receiver 105; for the application layer 101 or 106, this is the application protocol 102, for example. From the point of view of layer 5 101, 106, protocol data units having the format prescribed in the application protocol 102 are transmitted from the transmitter 104 to the receiver 105. The layers situated below the application layer represent functionalities which are provided transparently for the application layer. On the basis of the OSI model, the functionalities have expediently been chosen to be largely independent of one another.

Figure 2 shows an application layer with possible protocols, based on an application concerning the transmission of media data (audio and video data). Thus, 101 and 106 represent the application layer (layer 5) already known from figure 1 for the transmitter 104 and the receiver 105. This application layer is in turn divided into a plurality of sublayers. Thus, directly above the layer 4 is the real time transport protocol (RTP) 201. This protocol allows

voice and/or picture data to be transmitted in real time from

one to possibly a plurality of points (addressees). A layer 206 contains a security application which compiles security functions of the application layer; associated security protocols provide for services such as

- User authentication,
- Access control
- Confidentiality,
- Integrity
- Binding nature of the application data, and
- Accounting.

Depending on the instance of application, audio data 203 and/or video data 202 are used. For both alternatives, there are different compression standards, a few of which are shown by way of example in figure 2. For the audio data, these are the standards G.711, G.722, G.723.1 and G.729. For the video data, these are the two picture compression standards H.261 and H.263. In the next abstraction stage (similarly with respect to the OSI model) come the different possibilities for an application of audio or video data after compression and decompression (see block 204). One more abstraction level above, there is, by way of example, a user interface 205, which fully transparently provides a user with the services beneath which have just been described. By way of example, a user, taking into account both the audio and the video data, can thus make use of video telephony, e.g. over the Internet, without specifically needing to concern himself with any details of the services provided by the layers beneath. He thus uses the video telephony service, visible to him, over the Internet transparently. In this context, various alternative implementation forms can be situated below his service.

Figure 3 shows a flowchart between the transmitter and

the receiver for transmitting data from the transmitter 308 to the receiver

307. In a step 301, authentication data are produced on the application layer, and the protocol data unit (PDU) relevant to the application protocol is appended. In accordance with the arrangements of the application protocol, the data, including the authentication data, are transmitted to the receiver 307 (see connection 302). This is done using the functionalities or services available under the application protocol. In a block 303, the authentication data are checked in the receiver 307 and there is a test (in a block 304) to determine whether the transmitter 308 is known to the receiver 307. If this is not the case, the procedure branches to a block 305 and the data are rejected. If the transmitter 308 is known to the receiver 307, the procedure branches to a block 306 and the data are processed further. The authentication between transmitter and receiver was successful, and the transmitted data are not unwanted mass data.

Figure 4 shows a protocol data unit (PDU) of the application protocol with authentication data. What is shown in figure 4 is a protocol data unit (PDU) of the real time transport protocol (RTP, see 201 in figure 2) 401. Such an RTP packet 401 comprises an RTP header 406, encrypted media data 404 and authentication data 405. The RTP header 406 comprises, inter alia, a sequence number 402 and a timestamp 403. Both the transmitter and the receiver know a shared secret K, in this case indicated as a key, which is used to generate a message authentication code (MAC), on the basis of the sequence number 402 and the timestamp 403 (see field 405). An advantage with respect to encryption using a DES algorithm, for example, is a field length of 64 bits.

As already described, such a packet sent by the transmitter is authenticated at the receiver end

such that (in relation to the application layer) a message authentication code is generated by field 402 (sequence number) and field 403 (timestamp) on the basis of the key (K) known to the receiver. If this message authentication code is the same as the field 405, then the arriving RTP packet 401 is a data packet coming from a known transmitter, and the data are processed, for example displayed or stored. In the other case, unauthenticated data are involved, the transmitter is unknown to the receiver and the entire RTP packet is rejected.

Literature references

[1] Andrew S. Tanenbaum: Computer-Netzwerke; [Computer networks] Wolfram's Fachverlag; Attenkirchen 1992; ISBN 3-925328-79-3; pages 17 - 32.

[2] Christoph Ruland: Informationssicherheit in Datennetzen; [Information security in data networks] DATACOM-Verlag; Bergheim 1993; ISBN 3-89238-081-3; pages 61 - 63 and 68 - 79.

[3] [RFC1889] H. Schulzrinne: RTP: A Transport Protocol for Real-Time Applications; Internet Engineering Task Force, 1996

[4] [RFC2409] The Internet Key Exchange (IKE), D. Harkins, D. Carrel,; Internet Engineering Task Force, 1998.

[5] [RFC2402] IP Authentication Header, S. Kent, R. Atkinson; Internet Engineering Task Force, 1998.

[6] RFC2406] IP Encapsulating Security Payload (ESP), S.Kent, R. Atkinson; Internet Engineering Task Force, 1998.

Patent Claims

1. A method for transmitting data from a transmitter to a receiver,
   a) in which the transmitter extends the data by authentication data using an application protocol on the application layer;
   b) in which the authentication data are used by the receiver to ascertain whether it knows the transmitter;
   c) in which, if the receiver knows the transmitter, the data are accepted, otherwise the data are rejected.

2. The method as claimed in claim 1,
   in which the data are transmitted on a packet-oriented basis.

3. The method as claimed in one of the preceding claims,
   in which the authentication data are determined by virtue of at least part of a protocol data unit, available on the application layer, of the application protocol being linked to a secret by the transmitter.

4. The method as claimed in claim 3,
   in which the linking to the secret is carried out by virtue of
   a) cryptographical linking to at least one further parameter, particularly by virtue of an encryption operation and/or
   b) a cryptographic checksum and key information.

5. The method as claimed in one of the preceding claims,
   in which verification at the receiver end determines whether or not the receiver knows the transmitter.

6. The method as claimed in claim 5, in which the verification is carried out in the form of cryptographic verification using further parameters, particularly a decryption operation and/or a cryptographic checksum check.

7. The method as claimed in one of claims 3 to 6, in which the secret between the transmitter and the receiver is a key for symmetric encryption or a key pair for asymmetric encryption.

8. The method as claimed in one of claims 3 to 7, in which the part of the protocol data unit available on the application layer comprises a sequence number and/or a timestamp.

9. The method as claimed in one of claims 3 to 8, in which the encryption is performed using a one-way hash function or a message authentication code with a key which is known only to the receiver and to the transmitter which is to be authenticated.

10. The method as claimed in one of the preceding claims, in which, before the data are transmitted from the transmitter to the receiver, authentication is carried out between the transmitter and the receiver.

11. The method as claimed in one of the preceding claims for use in packet-switched telephone services.

12. The method as claimed in claim 11, used in Internet telephony.

13. The method as claimed in one of the preceding claims,
used in switching nodes and switching installations.

14. A system for transmitting data, in which a transmitter and a receiver are provided
a) in which the transmitter extends the data by authentication data using an application protocol on the application layer;
b) in which the receiver uses the authentication data to determine whether it knows the transmitter;
c) in which, if the receiver knows the transmitter, the receiver uses the data, otherwise it rejects the data.

15. A transmitter for sending data to a receiver,
which extends the data by authentication data using an application protocol on the application layer and sends them to the receiver.

16. A receiver for receiving data from a transmitter, particularly as claimed in claim 15,
a) which uses authentication data within the application protocol on the application layer to determine whether it knows the transmitter;
b) which, if it knows the transmitter, uses the data, otherwise it rejects the data.

10154564 - 07 Jan 2002

(12) NACH DEM VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS (PCT) VERÖFFENTLICHTE INTERNATIONALE ANMELDUNG

(19) Weltorganisation für geistiges Eigentum
Internationales Büro

(43) Internationales Veröffentlichungsdatum
26. Juli 2001 (26.07.2001)

PCT

(10) Internationale Veröffentlichungsnummer
WO 01/54371 A2

(51) Internationale Patentklassifikation[7]: H04L 29/00

(21) Internationales Aktenzeichen: PCT/DE01/00021

(22) Internationales Anmeldedatum:
5. Januar 2001 (05.01.2001)

(25) Einreichungssprache: Deutsch

(26) Veröffentlichungssprache: Deutsch

(30) Angaben zur Priorität:
100 01 855.6 18. Januar 2000 (18.01.2000) DE

(71) Anmelder *(für alle Bestimmungsstaaten mit Ausnahme von US)*: **SIEMENS AKTIENGESELLSCHAFT** [DE/DE]; Wittelsbacherplatz 2, 80333 München (DE).

(72) Erfinder; und
(75) Erfinder/Anmelder *(nur für US)*: **EUCHNER, Martin** [DE/DE]; Lorenzstrasse 2, 81737 München (DE).

(74) Gemeinsamer Vertreter: **SIEMENS AKTIENGESELLSCHAFT**; Postfach 22 16 34, 80506 München (DE).

(81) Bestimmungsstaaten *(national)*: CA, JP, US.

(84) Bestimmungsstaaten *(regional)*: europäisches Patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Veröffentlicht:
— *ohne internationalen Recherchenbericht und erneut zu veröffentlichen nach Erhalt des Berichts*

Zur Erklärung der Zweibuchstaben-Codes, und der anderen Abkürzungen wird auf die Erklärungen ("Guidance Notes on Codes and Abbreviations") am Anfang jeder regulären Ausgabe der PCT-Gazette verwiesen.

(54) Title: METHOD, SYSTEM FOR TRANSMITTING DATA FROM A TRANSMITTER TO A RECEIVER AND TRANSMITTER OR RECEIVER

(54) Bezeichnung: VERFAHREN, SYSTEM ZUR ÜBERMITTLUNG VON DATEN VON EINEM SENDER ZU EINEM EMPFÄNGER UND SENDER BZW. EMPFÄNGER HIERZU

(57) Abstract: A method and system for transmitting data from a transmitter to a receiver, wherein data from a transmitter is extended to include authentication data on the application level by means an application protocol. Said authentication data is used by the receiver to determine whether the transmitter is known by the receiver. If the transmitter is known by the receiver, the data is accepted. If not, the data is rejected.

(57) Zusammenfassung: Es wird ein Verfahren zur Übermittlung von Daten von einem Sender zu einem Empfänger angegeben, bei dem von dem Sender die Daten mittels eines Anwendungsprotokolls auf der Anwendungsschicht um Authentisierungsdaten erweitert werden. Anhand der Authentisierungsdaten wird von dem Empfänger ermittelt, ob er den Sender kennt. Falls der Empfänger den Sender kennt, werden die Daten entgegengenommen, ansonsten werden die Daten verworfen.

WO 01/54371 A2

FIG 1

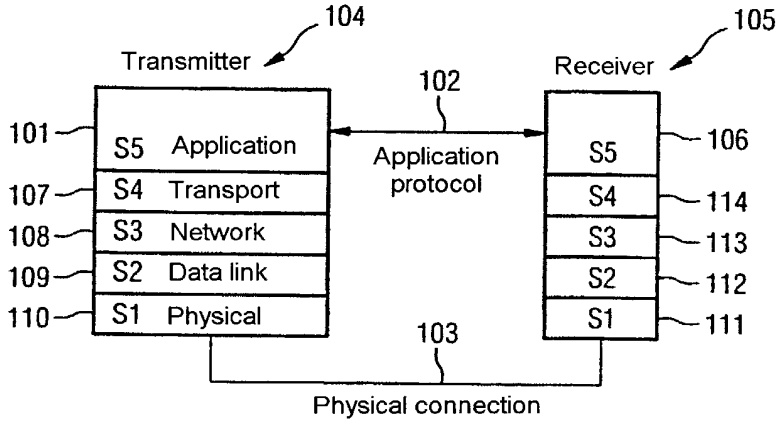

104
Transmitter
105
Receiver
102
101
S5 Application
Application protocol
S5
106
107
S4 Transport
S4
114
108
S3 Network
S3
113
109
S2 Data link
S2
112
110
S1 Physical
S1
111
103
Physical connection

FIG 2

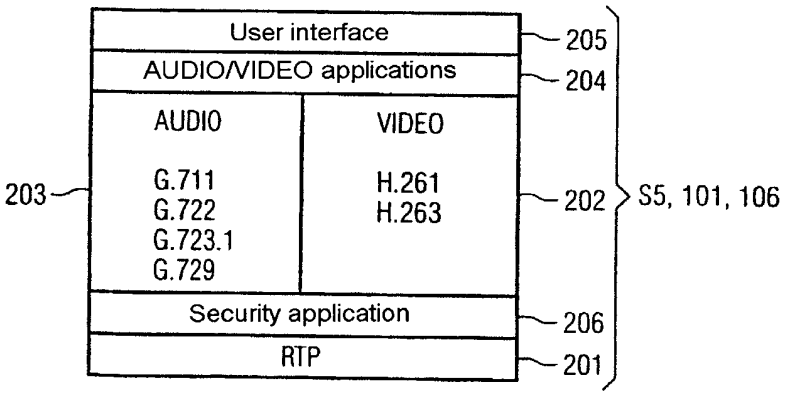

User interface
205
AUDIO/VIDEO applications
204
AUDIO
VIDEO
203
G.711
G.722
G.723.1
G.729
H.261
H.263
202
S5, 101, 106
Security application
206
RTP
201

FIG 3

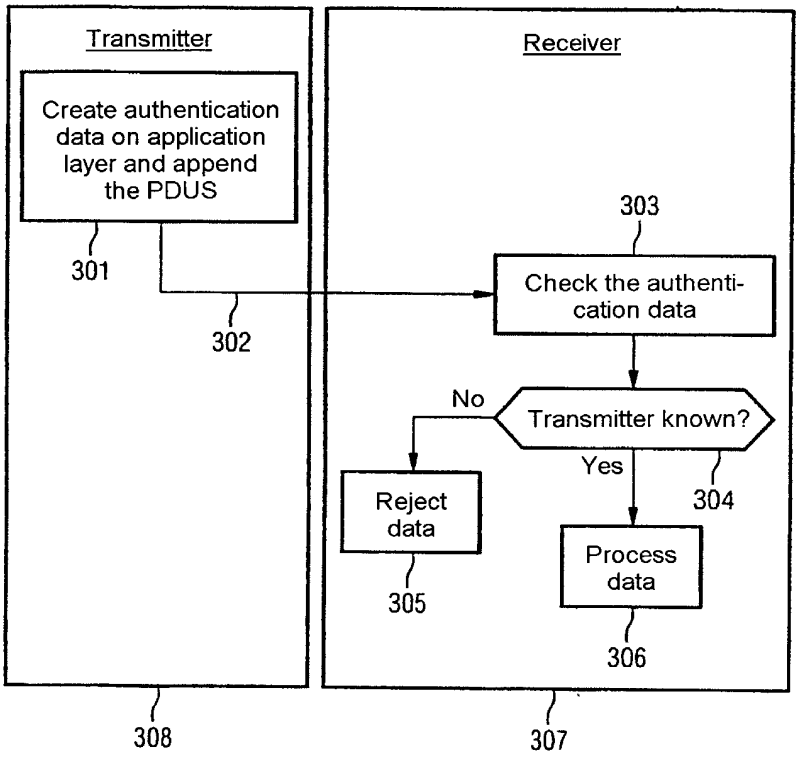
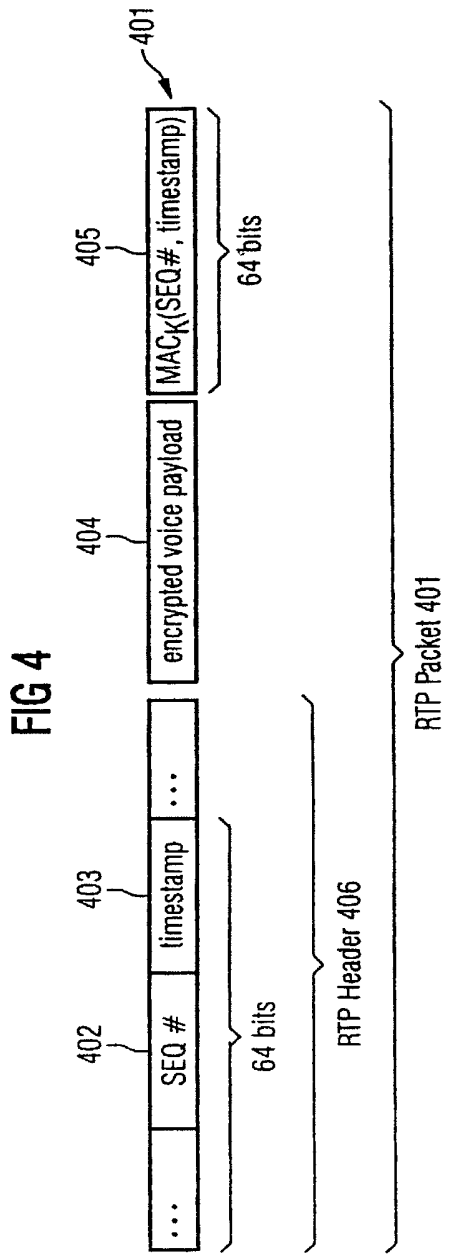

Transmitter
Receiver
Create authentication data on application layer and append the PDUS
301
302
303
Check the authenti-cation data
No
Transmitter known?
Yes
304
Reject data
305
Process data
306
308
307

FIG 4


401
405
MACK(SEQ#, timestamp)
64 bits
404
encrypted voice payload
403
timestamp
402
SEQ #
64 bits
RTP Header 406
RTP Packet 401

Declaration and Power of Attorney For Patent Application

Erklärung Für Patentanmeldungen Mit Vollmacht

German Language Declaration

Als nachstehend benannter Erfinder erkläre ich hiermit an Eides Statt:

dass mein Wohnsitz, meine Postanschrift, und meine Staatsangehörigkeit den im Nachstehenden nach meinem Namen aufgeführten Angaben entsprechen,

dass ich, nach bestem Wissen der ursprüngliche, erste und alleinige Erfinder (falls nachstehend nur ein Name angegeben ist) oder ein ursprünglicher, erster und Miterfinder (falls nachstehend mehrere Namen aufgeführt sind) des Gegenstandes bin, für den dieser Antrag gestellt wird und für den ein Patent beantragt wird für die Erfindung mit dem Titel:

Verfahren, System zur Uebermittlung von Daten von einem Sender zu einem Empfaenger und Sender bzw. Empfaenger hierzu

deren Beschreibung

(zutreffendes ankreuzen)
☐ hier beigefügt ist.
☒ am 05.01.2001 als
PCT internationale Anmeldung
PCT Anmeldungsnummer PCT/DE01/00021
eingereicht wurde und am _____
abgeändert wurde (falls tatsachlich abgeändert).

Ich bestätige hiermit, dass ich den Inhalt der obigen Patentanmeldung einschliesslich der Ansprüche durchgesehen und verstanden habe, die eventuell durch einen Zusatzantrag wie oben erwähnt abgeändert wurde.

Ich erkenne meine Pflicht zur Offenbarung irgendwelcher Informationen, die fur die Prüfung der vorliegenden Anmeldung in Einklang mit Absatz 37, Bundesgesetzbuch, Paragraph 1.56(a) von Wichtigkeit sind, an.

Ich beanspruche hiermit ausländische Prioritätsvorteile gemäss Abschnitt 35 der Zivilprozessordnung der Vereinigten Staaten, Paragraph 119 aller unten angegebenen Auslandsanmeldungen für ein Patent oder eine Erfindersurkunde, und habe auch alle Auslandsanmeldungen für ein Patent oder eine Erfindersurkunde nachstehend gekennzeichnet, die ein Anmeldedatum haben, das vor dem Anmeldedatum der Anmeldung liegt, für die Priorität beansprucht wird.

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name,

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

Method, system for transmitting data from a transmitter to a receiver and transmitter or receiver

the specification of which

(check one)
☐ is attached hereto.
☒ was filed on 05.01.2001 as
PCT international application
PCT Application No. PCT/DE01/00021
and was amended on _____
(if applicable)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, §1.56(a).

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed

IDNR: 2590 / V: 99-1.00 / B:Val

Form PTO-FB-240 (8-83) Patent and Trademark Office-U.S. DEPARTMENT OF COMMERCE

German Language Declaration

Prior foreign appplications Priorität beansprucht			Priority Claimed
10001855.6 (Number) (Nummer)	DE (Country) (Land)	18.01.2000 (Day Month Year Filed) (Tag Monat Jahr eingereicht)	☒ Yes Ja ☐ No Nein
(Number) (Nummer)	(Country) (Land)	(Day Month Year Filed) (Tag Monat Jahr eingereicht)	☐ Yes Ja ☐ No Nein
(Number) (Nummer)	(Country) (Land)	(Day Month Year Filed) (Tag Monat Jahr eingereicht)	☐ Yes Ja ☐ No Nein

Ich beanspruche hiermit gemäss Absatz 35 der Zivilprozessordnung der Vereinigten Staaten, Paragraph 120, den Vorzug aller unten aufgeführten Anmeldungen und falls der Gegenstand aus jedem Anspruch dieser Anmeldung nicht in einer früheren amerikanischen Patentanmeldung laut dem ersten Paragraphen des Absatzes 35 der Zivilprozeßordnung der Vereinigten Staaten, Paragraph 122 offenbart ist, erkenne ich gemäss Absatz 37, Bundesgesetzbuch, Paragraph 1.56(a) meine Pflicht zur Offenbarung von Informationen an, die zwischen dem Anmeldedatum der früheren Anmeldung und dem nationalen oder PCT internationalen Anmeldedatum dieser Anmeldung bekannt geworden sind.

I hereby claim the benefit under Title 35. United States Code. §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §122, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56(a) which occured between the filing date of the prior application and the national or PCT international filing date of this application.

(Application Serial No.) (Anmeldeseriennummer)	(Filing Date D, M, Y) (Anmeldedatum T, M, J)	(Status) (patentiert, anhängig, aufgegeben)	(Status) (patented, pending, abandoned)
PCT/DE01/00021	05 01.2001	anhängig	pending
(Application Serial No.) (Anmeldeseriennummer)	(Filing Date D,M,Y) (Anmeldedatum T, M; J)	(Status) (patentiert, anhangig, aufgeben)	(Status) (patented, pending, abandoned)

Ich erkläre hiermit, dass alle von mir in der vorliegenden Erklarung gemachten Angaben nach meinem besten Wissen und Gewissen der vollen Wahrheit entsprechen, und dass ich diese eidesstattliche Erklärung in Kenntnis dessen abgebe, dass wissentlich und vorsätzlich falsche Angaben gemäss Paragraph 1001, Absatz 18 der Zivilprozessordnung der Vereinigten Staaten von Amerika mit Geldstrafe belegt und/oder Gefängnis bestraft werden koennen, und dass derartig wissentlich und vorsätzlich falsche Angaben die Gültigkeit der vorliegenden Patentanmeldung oder eines darauf erteilten Patentes gefährden können.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

German Language Declaration

VERTRETUNGSVOLLMACHT: Als benannter Erfinder beauftrage ich hiermit den nachstehend benannten Patentanwalt (oder die nachstehend benannten Patentanwälte) und/oder Patent-Agenten mit der Verfolgung der vorliegenden Patentanmeldung sowie mit der Abwicklung aller damit verbundenen Geschäfte vor dem Patent- und Warenzeichenamt: *(Name und Registrationsnummer anführen)*

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. *(list name and registration number)*

And I hereby appoint

Customer No. 21171

Telefongespräche bitte richten an: *(Name und Telefonnummer)*

Direct Telephone Calls to: *(name and telephone number)*

Ext. _____

Postanschrift:

Send Correspondence to:

Staas & Halsey LLP Intellectual Property Attorneys
700 Eleventh Street NW, Suite 500 20001 Washington
Telephone: (001) 202 434 1500 and Facsimile (001) 202 434 1501
or
Customer No. 21171

1-00

Voller Name des einzigen oder ursprünglichen Erfinders:	Full name of sole or first inventor:
Martin Euchner	Martin Euchner
Unterschrift des Erfinders Datum: Euchner 29.05.2002	*Inventor's signature* Date
Wohnsitz: Muenchen, Deutschland DEK	Residence: Muenchen, Germany
Staatsangehörigkeit: DE	Citizenship: DE
Postanschrift: Lorenzstr. 2, 81737 Muenchen	Post Office Addess: Lorenzstr. 2, 81737 Muenchen
Voller Name des zweiten Miterfinders (falls zutreffend):	Full name of second joint inventor, if any:
Unterschrift des Erfinders Datum	*Second Inventor's signature* Date
Wohnsitz	Residence
Staatsangehorigkeit	Citizenship
Postanschrift	Post Office Address

(Bitte entsprechende Informationen und Unterschriften im Falle von dritten und weiteren Miterfindern angeben).

(Supply similar information and signature for third and subsequent joint inventors).

S&H Form: FORM PTO-1390 (2/01)

533 Rec'd PCT/PTO 18 JUL 2002

U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE

TRANSMITTAL LETTER TO THE UNITED STATES DESIGNATED/ELECTED OFFICE (DO/EO/US) CONCERNING A FILING UNDER 35 U.S.C. 371

ATTORNEY'S DOCKET NUMBER
1454.1302

10/181564

INTERNATIONAL APPLICATION NO.	INTERNATIONAL FILING DATE	PRIORITY DATE CLAIMED
PCT/DE01/00021	5 January 2001	18 January 2000

TITLE OF INVENTION
METHOD, SYSTEM FOR TRANSMITTING DATA FROM A TRANSMITTER TO A RECEIVER, AND TRANSMITTER OR RECEIVER

APPLICANT(S) FOR DO/EO/US
Martin EUCHNER

Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. ☒ This is a FIRST submission of items concerning a filing under 35 U.S.C. 371.
2. ☒ This is an express request to immediately begin national examination procedures (35 U.S.C. 371(f)).
3. ☒ The US has been elected by the expiration of 19 months from the priority date (PCT Article 31).
4. ☒ A copy of the International Application as filed (35 U.S.C. 371(c)(2))
   a. ☒ is transmitted herewith (required only if not transmitted by the International Bureau).
   b. ☐ has been transmitted by the International Bureau.
   c. ☐ is not required, as the application was filed in the United States Receiving Office (RO/US).
5. ☒ A translation of the International Application into English (35 U.S.C. 371(c)(2)).
6. ☐ Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371(c)(3))
   a. ☐ are transmitted herewith (required only if not transmitted by the International Bureau).
   b. ☐ have been transmitted by the International Bureau.
   c. ☐ is not required, as the application was filed in the United States Receiving Office (RO/US)
7. ☐ A translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)).
8. ☒ An oath or declaration of the inventor (35 U.S.C. 371(c)(4)).
9. ☐ A translation of the Annexes to the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371(c)(5)).

Items 10-15 below concern document(s) or information included:

10. ☒ An Information Disclosure Statement Under 37 CFR 1.97 and 1.98.
11. ☒ An assignment document for recording.
    Please mail the recorded assignment document to:
    a. ☒ the person whose signature, name & address appears at the bottom of this document.
    b. ☐ the following:
12. ☒ A preliminary amendment.
13. ☒ A substitute specification
14. ☐ A change of power of attorney and/or address letter.
15. ☒ Other items or information:

Copies of PCT EASY forms filed with International Application, cover page of International Application as published, International Search Report, and International Preliminary Examination Report.

©2001 Staas & Halsey LLP

☒ The U.S. National Fee (35 U.S.C. 371(c)(1)) and other fees as follows:

CLAIMS	(1) FOR	(2) NUMBER FILED	(3) NUMBER EXTRA	(4) RATE	(5) CALCULATIONS
	TOTAL CLAIMS	27 -20=	7	x $ 18.00	126.00
	INDEPENDENT CLAIMS	4 -3=	1	x $ 84.00	84.00
	MULTIPLE DEPENDENT CLAIM(S) (if applicable)			+$280.00	0.00
	BASIC NATIONAL FEE (37 CFR 1.492(a)(1)-(4)): ☐ Neither international preliminary examination fee (37 CFR 1.482) nor international search fee (37 CFR 1.445(a)(2)) paid to USPTO .....$1,040 ☐ International preliminary examination fee (37 C.F.R. 1.482) not paid to USPTO but International Search Report prepared by the EPO or JPO.....$ 890 ☐ International preliminary examination fee (37 C.F.R. 1.482) not paid to USPTO but international search fee (37 C.F.R. 1.445(a)(2) paid to USPTO...$ 740 ☐ International preliminary examination fee paid to USPTO (37 CFR 1.482) but all claims did not satisfy provision of PCT Article 33(1)-(4).....$ 710 ☐ International preliminary examination fee paid to USPTO (37 CFR 1.482) and all claims satisfied provisions of PCT Article 33(2) to (4) .....$ 100				890.00
	Surcharge of $130 for furnishing the National fee or oath or declaration later than ☐ 20 ☐ 30 mos. from the earliest claimed priority date (37 CFR 1.482(e)).				0.00
			TOTAL OF ABOVE CALCULATIONS		1,100.00
	Reduction by 1/2 for filing by small entity, if applicable. Affidavit must be filed also. (Note 37 CFR 1.9, 1.27, 1.28.)				
			SUBTOTAL		1,100.00
	Processing fee of $130 for furnishing the English Translation later than [ ] 20 [ ] 30 mos. from the earliest claimed priority date (37 CFR 1.482(f)).				
			TOTAL NATIONAL FEE		1,100.00
	Fee for recording the enclosed assignment (37 CFR 1.21(h)).				+ 40.00
			TOTAL FEES ENCLOSED		1,140.00

a. ☒ A check in the amount of $1,140.00 to cover the above fees is enclosed.
b. ☐ Please charge my Deposit Account No. 19-3935 in the Amount of $ to cover the above fees. A duplicate copy of this sheet is enclosed.
c. ☒ The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account No. 19-3935. A duplicate copy of this sheet is enclosed.

21171
PATENT TRADEMARK OFFICE

SUBMITTED BY: STAAS & HALSEY LLP

Type Name	Richard A. Gollhofer	Reg. No.	31,106
Signature	Richard A. Gollhofer	Date	7/18/02

©2001 Staas & Halsey LLP

Special Instructions for Evidence Copy Box Identification

Documents in this patent application scanned prior to the scan date of this document may not have a box number present in the database. The documents are in the same box as this paper. If the patent application documents that do not have a box number are stored in more than one box, a copy of this form is placed in each box. Check the database box number for each copy of this form to identify all of the evidence copy box numbers for documents that do not have a box number.

X	**The documents stored in this box are original application papers scanned and endorsed by PACR and imported into IFW.**
	The documents stored in this box were scanned into the IFW prototype for GAU 1634, 2827,or 2834.

Indexer, place and **X** in only one box above to indicate the documents placed in this box that were previously scanned in PACR or IFW and will not be scanned again.

UNITED STATES PATENT AND TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON, D.C. 20231
www.uspto.gov

Bib Data Sheet

CONFIRMATION NO. 4619

SERIAL NUMBER	FILING DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
10/181,564	07/18/2002 RULE	713	2131	1454.1302

APPLICANTS

Martin Euchner, Muenchen, GERMANY;

**** CONTINUING DATA ***************************

THIS APPLICATION IS A 371 OF PCT/DE01/00021 01/05/2001

**** FOREIGN APPLICATIONS ***********************

GERMANY 100 01 855.6 01/18/2000

Foreign Priority claimed	STATE OR COUNTRY	SHEETS DRAWING	TOTAL CLAIMS	INDEPENDENT CLAIMS
☐ yes ☐ no	GERMANY	3	28	4
35 USC 119 (a-d) conditions met ☐ yes ☐ no ☐ Met after Allowance				
Verified and Acknowledged Examiner's Signature Initials				

ADDRESS

21171

TITLE

Method, system for transmitting data from a transmitter to a receiver and transmitter or receiver

FILING FEE RECEIVED	FEES: Authority has been given in Paper	
1118	No. _____ to charge/credit DEPOSIT ACCOUNT No. _____ for following:	☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other _____ ☐ Credit

PATENT APPLICATION SERIAL NO. 10/181564

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

07/23/2002 SNAJARRO 00000143 10181564

01 FC:970 890.00 OP
02 FC:966 126.00 OP
03 FC:964 84.00 OP

09/16/2002 BCAMPBEL 00000002 193935 10181564
01 FC:966 18.00 CH

PTO-1556
(5/87)

*U.S. Government Printing Office: 2001 — 481-697/59173

PATENT APPLICATION FEE DETERMINATION RECORD
Effective October 1, 2001

Application or Docket Number: 10/181564

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS		
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	28 minus 20=	* 08
INDEPENDENT CLAIMS	4 minus 3 =	* 1
MULTIPLE DEPENDENT CLAIM PRESENT		☐

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE ☐		OR	OTHER THAN SMALL ENTITY	
RATE	FEE		RATE	FEE
BASIC FEE	370.00	OR	BASIC FEE	~~740.00~~ 890
X$ 9=		OR	X$18=	144
X42=		OR	X84=	84
+140=		OR	+280=	
TOTAL		OR	TOTAL	

CLAIMS AS AMENDED - PART II

AMENDMENT A

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT B

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
***If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Meta 002
p. 66

MULTIPLE DEPENDENT CLAIM FEE CALCULATION SHEET

(FOR USE WITH FORM PTO-875)

SERIAL NO. 10/181564

FILING DATE

APPLICANT(S)

CLAIMS

	AS FILED		AFTER 1st AMENDMENT		AFTER 2nd AMENDMENT	
	IND.	DEP.	IND.	DEP.	IND.	DEP.
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17				1		
18				1		
19				1		
20				1		
21				1		
22				1		
23				1		
24				1		
25				1		
26				1		
27				1		
28				1		
29				1		
30				1		
31				1		
32				1		
33				1		
34				1		
35				1		
36				1		
37				1		
38				1		
39				1		
40				1		
41				1		
42			1			
43			1			
44			1			
45						
46						
47						
48						
49						
50						
TOTAL IND.			4			
TOTAL DEP.			24			
TOTAL CLAIMS			28			

	*		*		*	
	IND.	DEP.	IND.	DEP.	IND.	DEP.
51						
52						
53						
54						
55						
56						
57						
58						
59						
60						
61						
62						
63						
64						
65						
66						
67						
68						
69						
70						
71						
72						
73						
74						
75						
76						
77						
78						
79						
80						
81						
82						
83						
84						
85						
86						
87						
88						
89						
90						
91						
92						
93						
94						
95						
96						
97						
98						
99						
100						
TOTAL IND.						
TOTAL DEP.						
TOTAL CLAIMS						

* MAY BE USED FOR ADDITIONAL CLAIMS OR ADMENDMENTS

Barbara Campbell
National Stage Processing
(703) 305-3631

FORM PTO-1360 (REV. 3-78)

U.S.DEPARTMENT OF COMMERCE
Patent and Trademark Office

EP

UNITED STATES NATIONAL STAGE SHEET (DO/EO)
Barbara Campbell, Paralegal

PUBLICATION NO. WO 01/54371

PUBLICATION DATE 7-26-01

U. S. APPL. NO. 10/181564

INTERNATIONAL APPL. DE01/00021

Application Filed By: 20 MOS. _____ 30 MOS. ✓

Copy of International Application ✓ Missing _____ PCT/RO/101 ✓

371 Filing fees ✓ paid _____ insufficient

Total Claims 28 Chargeable 28 Independent 4 Multiple N (Claims renumbered)

Total Drawing Sheets 3

Oath/Declaration missing _____ signed ✓ unsigned _____ defective _____

Small entity _____ large entity ✓

Biochemical Seq. Diskette _____ missing _____ damaged ____ entered ___ not entered ___

Biochemical Sequence listing ____missing ____ statement ____ no statement _____

Copy of ISR ✓ with references _____without references

Copy of IPRE ✓ annexes _____ entered _____not entered _____PCT/IB/331

Preliminary Amendments ✓ entered _____not entered

IDS ✓

Request for Immediate Examination ✓

Substitute Specification _____

Assignment yes ✓ no ____

Priority Document (s) yes ✓ no _____

Application fees owed/paid _____declaration _____ claims _____ multiple
_____ Translation ____ extension of time

35 USC Receipt of Request 7-18-02
Date completion USC 371 Requirements 11
Notice of Missing Requirements _____
Notice of Defective _____ Response _____ Translation _____ Declaration
Notice of Acceptance 9-13-02
Notice of Abandonment _____

10/181564
512 Rec'd PCT/PTO 18 JUL 2002
#3

Attorney Docket No. 1454.1302

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Martin EUCHNER

Application No.: Group Art Unit:

Filed: (concurrently) Examiner:

For: METHOD, SYSTEM FOR TRANSMITTING DATA FROM A TRANSMITTER TO A RECEIVER, AND TRANSMITTER OR RECEIVER

INFORMATION DISCLOSURE STATEMENT

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

In accordance with the duty of disclosure provisions of 37 CFR § 1.56, there is hereby provided certain information which the Examiner may consider material to the examination of the subject U.S. patent application. It is requested that the Examiner make this information of record if it is deemed material to the examination of the subject application.

1. Enclosures accompanying this Information Disclosure Statement are:

1a. ☒ Form PTO-1449.
1b. ☒ Copies of IDS citations.
1c. ☒ An English language copy of search report(s) from a counterpart foreign application or a PCT International Search Report.
1d. ☐ English language translation (complete or relevant portion(s)) attached to each non-English language publication.
1e. ☐ Explanations of Relevancy of References (ATTACHMENT 1(e), hereto) for providing a concise explanation of each non-English publication.

2. ☒ In accordance with 37 CFR § 1.98, a concise explanation of what is presently understood to be the relevance of each non-English language publication is

(Check appropriate Items 2a, 2b, 2c and/or 2d)

2a. ☐ satisfied because all non-English language publications were cited on the enclosed "English-language version of the search report or action which indicates the degree of relevance found by the foreign office". (See MPEP 609, Minimum Requirements for an Information Disclosure Statement, Part A(3): Concise Explanation of Relevance, pp. 600-100 to 600-101, Rev. 1, Feb. 2000.)
2b. ☒ set forth in the application.

©2001 Staas & Halsey LLP

2c. ☐ satisfied because an English language translation (complete or relevant portion(s)) is attached to each non-English language publication.

2d. ☐ enclosed as Attachment 1(e), hereto.

3. No admission is made that the information cited in this Statement is, or is considered to be, material to patentability nor a representation that a search has been made (other than search report(s) from a counterpart foreign application or a PCT International Search Report, if submitted herewith). 37 CFR §§ 1.97(g) and (h).

Respectfully submitted,

STAAS & HALSEY LLP

Dated: 7/18/02

By: Richard A. Gollhofer

Richard A. Gollhofer
Registration No. 31,106
700 Eleventh Street, N.W., Suite 500
Washington, D.C. 20001
Telephone: (202) 434-1500
Facsimile: (202) 434-1501

©2001 Staas & Halsey LLP

10/181564

5 Rec'd PCT 18 JUL 2002

#3

FORM PTO-1449 U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE	ATTORNEY DOCKET NO. 1454.1302	APPLICATION NO.
LIST OF REFERENCES CITED BY APPLICANT	FIRST NAMED INVENTOR Martin EUCHNER	
(Use several sheets if necessary)	FILING DATE	GROUP ART UNIT

U.S. PATENT DOCUMENTS

*EXAMINER INITIAL		DOCUMENT NO.	DATE	NAME	CLASS	SUB-CLASS	FILING DATE
	AA						
	AB						
	AC						

FOREIGN PATENT DOCUMENTS

		DOCUMENT NO.	DATE	COUNTRY	CLASS	SUB-CLASS	TRANSLATION YES	TRANSLATION NO
	AD	99/11019	3/4/99	WIPO				
	AE							

OTHER REFERENCES (INCLUDING AUTHOR, TITLE, DATE, PERTINENT PAGES, ETC.)

	AF	Ray Hunt, "Internet/Intranet Firewall Security-Policy, Architecture and Transaction Services, Computer Communications, vol. 21, No. 13, September 1998, pages 1107-1123
	AG	Andrew S. Tanenbaum, "Computer-Netzwerke", Wolfram's Fachverlag, Attenkirchen 1992, pages 17-32
	AH	Christoph Ruland, "Informationssicherheit in Datennetzen", DATACOM-VERLAG, Bergheim 1993, pages 61-63 and 68-79
	AI	H. Schulzrinne: "RTP: A Transport Protocol for Real-Time Applications; Internet Engineering Task Force", RFC 1889, 1996, pages 1-75
	AJ	D. Harkins, et al., "The Internet Key Exchange (IKE)" Internet Engineering Task Force, RFC 2409, 1998, pages 1-41
	AK	S. Kent et al. "Internet Engineering Task Force", RFC 2402, 1998, pages 1-22
	AL	S. Kent et al. "Internet Task Force", RFC 2406, 1998, pages 1-22

EXAMINER	DATE CONSIDERED

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

©2001 Staas & Halsey LLP

Andrew S. Tanenbaum

Computer-Netzwerke

BEST AVAILABLE COPY

Wolfram's Fachverlag

1 Einleitung 1

1.1 Einsatzgebiete von Computer-Netzwerken 3

1.1.1 Netzwerkziele 3

1.1.2 Netzwerk-Anwendungen 4

1.2 Netzwerk-Strukturen 6

1.3 Netzwerk-Architekturen 11

1.3.1 Protokollhierarchien 11

1.3.2 Entwicklungsaspekte der Schichten 15

1.4 Das OSI-Referenzmodell 17

1.4.1 Die Bitübertragungsschicht 18

1.4.2 Die Sicherungsschicht 19

1.4.3 Die Vermittlungsschicht 19

1.4.4 Die Transportschicht 20

1.4.5 Die Sitzungsschicht 21

1.4.6 Die Darstellungsschicht 22

1.4.7 Die Anwendungsschicht 23

1.4.8 Die Datenübertragung im OSI-Modell 23

1.5 Dienste 24

1.5.1 OSI-Terminologie 24

1.5.2 Verbindungsorientierte und verbindungsunabhängige Dienste 27

1.5.3 Dienstelemente 29

1.5.4 Die Beziehung zwischen Diensten und Protokollen 32

1.6 Die Netzwerk-Normung 33

1.6.1 Who's who in der Welt der Telekommunikation 33

1.6.2 Who's who in der Normen- und Standardwelt 35

Gibt es zwischen einer Quelle und einem Ziel mehrere Verbindungswege, so muß eine Route gewählt werden. Manchmal verteilt sich diese Entscheidung auf zwei oder noch mehr Schichten. Um zum Beispiel Daten von London nach Rom zu schicken, müßte auf hoher Ebene aufgrund des jeweilig geltenden Rechts entschieden werden, ob die Daten über Frankreich oder Deutschland geschickt werden, und eine Entscheidung auf niedriger Ebene müßte nach der gegenwärtigen Verkehrssituation die entsprechende Leitung unter den vielen vorhandenen festlegen.

1.4 Das OSI-Referenzmodell

Nachdem wir Schichtnetzwerke abstrakt besprochen haben, ist es an der Zeit, die Schichten zu betrachten, die wir durch dieses ganze Buch hindurch verwenden werden. Abbildung 1.7 zeigt das entsprechende Modell. Dieses Modell basiert auf einem Vorschlag, der von der International Standards Organization (ISO) entwickelt wurde und den ersten Schritt auf dem Weg zur internationalen Standardisierung der verschiedenen Protokolle (Day und Zimmermann, 1983) darstellt. Dieses Modell trägt den Namen **ISO-OSI-(Open Systems Interconnection, Kommunikation offener Systeme)-Referenzmodell**, weil es sich damit beschäftigt, offene Systeme miteinander zu verbinden - das heißt, Systeme, die für die Kommunikation mit anderen Systemen offen sind. Wir werden es der Einfachheit halber mit OSI-Modell abkürzen.

Das OSI-Modell hat sieben Schichten. Folgende Prinzipien haben zu der Siebenschichtigkeit geführt:

1. Eine neue Schicht sollte dort entstehen, wo ein neuer Abstraktionsgrad benötigt wird.
2. Jede Schicht sollte eine genau definierte Funktion erfüllen.
3. Bei der Funktionswahl sollte man die Definition international genormter Protokolle im Auge behalten.
4. Die Grenzen zwischen den einzelnen Schichten sollten so gewählt werden, daß der Informationsfluß über die Schnittstellen möglichst gering ist.
5. Die Anzahl an Schichten sollte so groß sein, daß keine Notwendigkeit dafür besteht, verschiedene Funktionen auf dieselbe Schicht zu packen, und so klein, daß die gesamte Architektur nicht unhandlich wird.

In den Abschnitten 1.4.1 bis 1.4.7 werden wir, angefangen mit der untersten Schicht, jeweils eine Schicht dieses Modells behandeln. Man beachte, daß das OSI-Modell keine Netzwerkarchitektur ist, da die genauen Dienste und Protokolle, die in jeder Schicht verwendet werden sollen, nicht festgelegt werden. Das OSI-Modell sagt lediglich aus, was die jeweilige Schicht können soll. Es sind jedoch von der ISO auch Normen für alle Schichten erstellt worden, obwohl diese genau genommen nicht zum

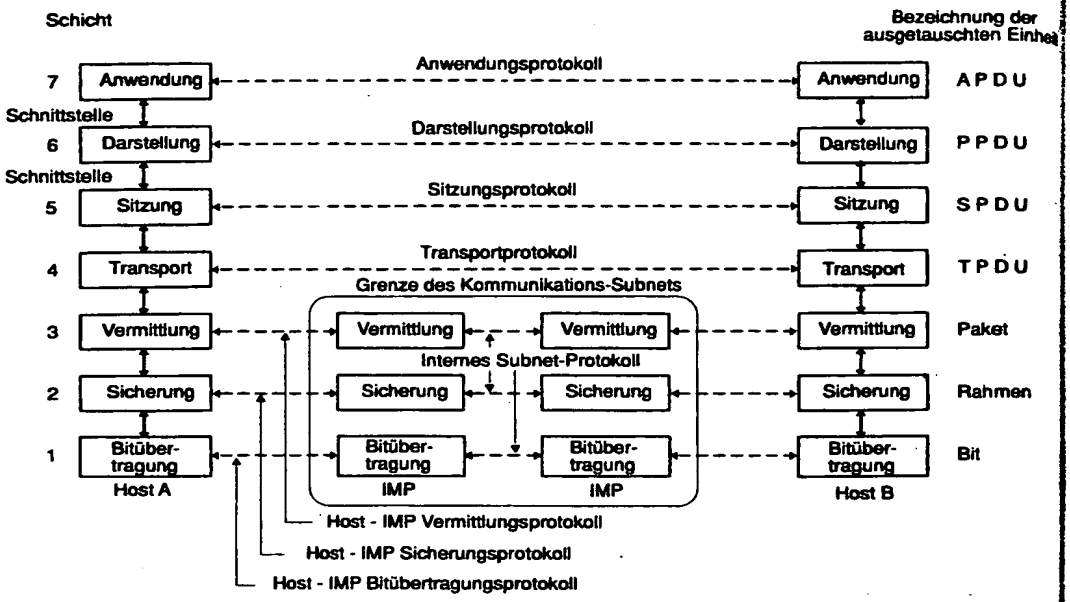


Abbildung 1.7: Die in diesem Buch verwendete Netzwerkarchitektur. Sie stützt sich auf das OSI-Modell

Modell gehören. Jede von ihnen ist als eigene internationale Norm veröffentlicht worden.

1.4.1 Die Bitübertragungsschicht

Die **Bitübertragungsschicht** (physical layer) beschäftigt sich mit der Übertragung von rohen Bits über einen Kommunikationskanal. Entwicklungstechnisch muß hier sichergestellt werden, daß, wenn die eine Seite ein Bit mit der Wertigkeit 1 schickt, dieses Bit von der anderen Seite auch als Bit mit der Wertigkeit 1 empfangen wird, und nicht als Bit mit der Wertigkeit 0. Typische Fragen, die hierbei auftauchen, sind: wieviel Volt sollten einer logischen 1 entsprechen und wieviel einer 0, wieviele Mikrosekunden soll ein Bit dauern, soll die Übertragung in beide Richtungen gleichzeitig erfolgen, wie kommt die erste Verbindung zustande und wie wird sie aufgelöst, wenn beide Seiten fertig sind, wieviele Pins hat die Endsystemverbindung und wofür werden diese verwendet. Die Fragen der Entwicklung beschäftigen sich hier weitgehend mit mechanischen, elektrischen und prozeduralen Schnittstellen und mit dem physikalischen Übertragungsmedium, das sich unterhalb der Bitübertragungsschicht befindet. Die

wicklung dieser Schicht kann man ruhigen Gewissens als Aufgabe eines Elektroin-
ieurs bezeichnen.

.4.2 Die Sicherungsschicht

wichtigste Aufgabe der **Sicherungsschicht** (data link layer) ist es, die übertrage-
n Rohdaten in eine Datenreihe zu verwandeln, die ohne Übertragungsfehler an die
rmittlungsschicht weitergegeben wird. Dies wird dadurch erreicht, daß der Sender
e Eingangsdaten in **Datenübertragungsrahmen** (data frames) - typisch sind einige
ndert Bytes - aufteilt, diese sequentiell überträgt und die vom Empfänger erzeugten
ittungsrahmen verarbeiten muß. Da die Bitübertragungsschicht einfach nur einen
om von Bits annimmt und absendet, ohne sich um die Bedeutung oder Struktur zu
mmern, ist es Sache der Sicherungsschicht, Rahmengrenzen einzufügen und zu
kennen. Dies kann dadurch erreicht werden, daß an Anfang und Ende eines Rah-
ens spezielle Bitmuster gesetzt werden. Könnten diese Bitmuster zufällig auch im
tenteil des Rahmens vorkommen, muß man besondere Sorgfalt darauf verwenden,
aß keine Verwirrung entsteht.

ine Bündelstörung in der Übertragungsleitung kann einen Rahmen gänzlich zerstö-
n. In diesem Fall muß die Software auf der Sicherungsschicht der Quellmaschine
n Rahmen erneut übertragen. Wenn ein Rahmen allerdings mehrmals übertragen
rd, entsteht die Gefahr der Rahmenduplizierung. Ein Rahmenduplikat könnte zum
ispiel dann gesendet werden, wenn der Quittungsrahmen des Empfängers zerstört
rd. Diese Schicht muß sich also mit den durch beschädigte, verlorengegangene und
uplizierte Rahmen entstehenden Probleme herumschlagen. Die Sicherungsschicht
önnte der Vermittlungsschicht mehrere verschiedene Dienstgruppen - jede davon in
nderer Qualität und zu einem anderen Preis - anbieten.

in weiteres Problem, das in der Sicherungssicht (und in den meisten der höheren
chichten ebenfalls) auftaucht, ist die Datenüberschwemmung, die entsteht, wenn der
ender schneller ist als der Empfänger. Es muß eine Art Verkehrsregelung installiert
erden, die dem Sender mitteilt, wieviel Pufferbereich beim Empfänger gerade noch
erfügbar ist. Häufig werden sowohl diese Flußregelung als auch die Fehlerbehand-
ng integriert, um die Sache bequemer zu machen.

enn die Übertragungsleitung in beide Richtungen zu verwenden ist, wirft dies ein
eues Problem für die Software der Sicherungsschicht auf, und zwar, daß die Quit-
ngsrahmen für den Verkehr von *A* nach *B* mit den DÜ-Rahmen von *B* nach *A*
onkurrieren. Hier wurde eine clevere Lösung (die Huckepackverschickung, piggybak-
ing) entwickelt, die wir später noch genauer durchleuchten werden.

.4.3 Die Vermittlungsschicht

ie **Vermittlungsschicht** (network layer) beschäftigt sich mit der Steuerung des
ubnet-Betriebs. Eine der wichtigsten Aufgaben bei der Entwicklung ist die Auswahl

der Paketleitwege vom Ursprungs- zum Bestimmungsort. Die Leitwege könnten auf statischen Tabellen aufgebaut sein, die im Netzwerk »verdrahtet« sind und sich nur selten ändern. Sie könnten auch vor Beginn jeden Gesprächs, zum Beispiel einer Terminalsitzung, festgelegt werden. Schließlich könnten sie auch hochdynamisch sein und für jedes Paket neu bestimmt werden, um so die optimale Netzwerkauslastung zu gewährleisten.

Wenn sich zu viele Pakete gleichzeitig im Subnet befinden, stehen sie sich gegenseitig im Weg, und es kommt zu Engpässen. Auch die Steuerung solcher Stauungen ist Aufgabe der Vermittlungsschicht.

Da die Betreiber des Subnets sehr wahrscheinlich für ihre Mühe etwas sehen wollen, beinhaltet die Vermittlungsschicht sehr oft eine Abrechnungsfunktion. Für die zur Rechnungsstellung nötigen Informationen muß die Software mindestens die Anzahl an Paketen, Zeichen oder Bits zählen, die jeder Kunde sendet. Überschreitet ein Paket eine Landesgrenze und bestehen auf beiden Seiten unterschiedliche Preise, kann die Buchhaltung schwierig werden.

Es können auch viele Probleme entstehen, wenn ein Paket auf seinem Weg zum Bestimmungsort von einem Netzwerk zu einem anderen reisen muß. Es könnte zum Beispiel sein, daß die Adressierung beim zweiten Netzwerk anders ist als beim ersten. Das zweite Netzwerk könnte das Paket überhaupt ablehnen, weil es zu groß ist. Die Protokolle könnten verschieden sein, und so weiter. All diese Probleme müssen in der Vermittlungsschicht bewältigt werden, damit heterogene Netzwerke miteinander in Verbindung treten können.

Bei Rundsendenetzwerken ist das Problem der Leitwegbestimmung einfach, und deshalb ist dort die Vermittlungsschicht oft auch sehr dünn oder überhaupt nicht vorhanden.

1.4.4 Die Transportschicht

Grundsätzlich ist es die Aufgabe der **Transportschicht** (transport layer), die Daten von der Sitzungsschicht zu übernehmen, sie wenn nötig in kleinere Einheiten zu zerlegen, sie danach an die Vermittlungsschicht zu übergeben und dafür zu sorgen, daß alle Teile richtig am anderen Ende ankommen. All dies muß außerdem effizient und auf eine Weise gemacht werden, daß Sitzungsschicht auf alle Fälle vor den unvermeidlichen Änderungen in der Hardware-Technologie geschützt wird.

Normalerweise baut die Transportschicht für jede von der Sitzungsschicht benötigte Transportverbindung eine eigene Netzwerkverbindung auf. Wenn die Transportverbindung jedoch einen hohen Durchsatz erfordert, könnte die Transportschicht auch mehrere Netzwerkverbindungen aufbauen und die Daten zur Erhöhung des Durchsatzes auf diese Verbindungen verteilen. Die Transportschicht könnte auf der anderen Seite mehrere Transportverbindungen auf eine Netzwerkverbindung zusammenfassen, wenn der Aufbau oder die Aufrechterhaltung einer Netzwerkverbindung eine Kosten-

lage darstellt. In jedem dieser Fälle ist die Transportschicht dafür verantwortlich, die Mehrfachnutzung für die Sitzungsschicht transparent zu machen.

Die Transportschicht bestimmt auch die Art des Dienstes, welcher der Sitzungsschicht und letztlich dem Netzwerkanwender - zur Verfügung gestellt wird. Die gebräuchlichste Art von Transportverbindung ist eine fehlerfreie Standleitung (point-to-point channel), in der die Nachrichten in ihrer Sendereihenfolge übertragen werden. Es gibt aber auch andere Möglichkeiten für den Transportdienst, zum Beispiel den Transport einzelner Nachrichten ohne Gewähr für die Einhaltung der ursprünglichen Reihenfolge oder die Rundsendung einer Nachricht an eine Vielzahl von Empfängern. Die Art des Dienstes wird beim Aufbauen der Verbindung festgelegt.

Die Transportschicht ist eine echte Ursprungs-zu-Ziel- oder **Ende-zu-Ende**-Schicht. Anders gesagt führt ein Programm auf der Quellmaschine ein Gespräch mit einem ähnlichen Programm auf der Zielmaschine und verwendet dabei die Nachrichtenköpfe und die Steuernachrichten. In den niedrigeren Schichten bestehen die Protokolle zwischen jeder Maschine und ihren unmittelbaren Nachbarn und nicht zwischen den eigentlichen Quell- und Zielmaschinen, die durch viele IMPs voneinander getrennt sein könnten. Den Unterschied zwischen den Schichten 1 bis 3, die gekettet sind, und den Schichten 4 bis 7, die Ende-zu-Ende sind, veranschaulicht Abbildung 1.7.

Viele Hostrechner arbeiten im Mehrprogrammbetrieb, wodurch viele Verbindungen zum und vom Host entstehen. Es wird dadurch erforderlich, zu bestimmen, welche Nachricht zu welcher Verbindung gehört. Der Transport-Nachrichtenkopf (H_4 in Abbildung 1.6) ist eine Stelle, an der diese Information gut aufgehoben ist.

Außer der Regelung der Mehrfachnutzung eines Kanals durch mehrere Nachrichtenströme muß die Transportschicht sich auch noch um den Aufbau und die Auflösung von Verbindungen durch das Netzwerk kümmern. Eine Art Benennungsmechanismus ist nötig, damit ein Prozeß auf einer Maschine die Möglichkeit hat, zu beschreiben, mit wem er kommunizieren will. Es muß weiterhin ein Regulativ für den Informationsfluß vorhanden sein, damit ein schneller Host einen langsamen nicht überholen kann. Flußregulierung zwischen Hosts ist anders als Flußregulierung zwischen IMPs, obwohl wir später noch sehen werden, daß für beide ähnliche Prinzipien gelten.

1.4.5 Die Sitzungsschicht

Die **Sitzungsschicht** (session layer) ermöglicht es Anwendern an verschiedenen Maschinen, zu **Sitzungen** zusammenzukommen. Eine Sitzung gestattet den ganz normalen Datentransport - genauso wie die Transportschicht -, zusätzlich dazu leistet sie aber auch noch bestimmte gehobene Dienste, die bei manchen Anwendungen nützlich sind. Eine Sitzung könnte den Zweck haben, einem Anwender Zugang zu einem rechnerfernen Teilnehmerrechensystem (time-sharing system) zu verschaffen oder eine Datei von einer Maschine zu einer anderen zu übertragen.

Meta 1002
p. 78

Einer der Dienste der Sitzungsschicht ist die Dialogsteuerung. Es gibt Sitzungen, be denen der Verkehr in beide Richtungen gleichzeitig möglich ist und auch solche, be der Verkehr nur in jeweils eine Richtung möglich ist. Wenn der Verkehr nur in eine Richtung möglich ist (wie bei einspurigen Eisenbahnstrecken), so hilft die Sitzungsschicht bei der Festlegung, wer gerade an der Reihe ist.

Ein ähnlicher Dienst ist das **Token-Management**. Bei manchen Protokollen ist es vor grundlegender Bedeutung, daß nicht beide Seiten gleichzeitig dieselbe Operation einleiten. Von der Sitzungsschicht bereitgestellte Tokens, die ausgetauscht werden können, regeln diese Problemstelle, denn nur die Seite, die gerade über das Token verfügt, darf die fragliche Operation ausführen.

Ein weiterer Sitzungsdienst ist die **Synchronisierung**. Man denke an die Probleme, die entstehen könnten, wenn man versuchte, eine Datei mit zwei Stunden Übertragungszeit an eine Maschine in einem Netzwerk zu schicken, dessen mittlere Absturzzeit bei einer Stunde liegt. Nach jedem Abbruch der Übertragung müßte wieder ganz von vorne angefangen werden, und man würde wahrscheinlich wieder nur bis zum nächsten Netzwerkabsturz kommen. Um diesem Problem aus dem Weg zu gehen, können in der Sitzungsschicht Fixpunkte (checkpoints) in den Datenfluß eingefügt werden, damit nach einem Netzwerkabsturz nur noch die Daten nach dem letzten Fixpunkt erneut übertragen werden müssen.

1.4.6 Die Darstellungsschicht

Die **Darstellungsschicht** (presentation layer) führt gewisse Funktionen durch, die durch ihr häufiges Auftreten eine allgemeine Lösung rechtfertigen, anstatt es jedem Anwender selbst zu überlassen, die beinhalteten Probleme zu lösen. Anders als alle anderen Schichten, die nur daran interessiert sind, Bits zuverlässig von hier nach da zu schaufeln, hat die Darstellungsschicht die Besonderheit, daß für sie auch die Syntax und Semantik der übertragenen Information von Interesse ist.

Ein typisches Beispiel für einen Dienst der Darstellungsschicht ist die Kodierung von Daten auf eine standardisierte und vereinbarte Art und Weise. Die meisten Anwenderprogramme tauschen keine beliebigen Binärbitketten aus. Sie tauschen Dinge wie Menschennamen, Datumsangaben, Geldsummen oder Rechnungen aus. Diese Begriffe werden dargestellt durch Zeichenketten, ganze Zahlen, Gleitpunktzahlen oder Datenstrukturen, die aus mehreren einfacheren Bestandteilen aufgebaut sind. Nun haben verschiedene Computer auch verschiedene Darstellungscodes für Zeichenketten (zum Beispiel ASCII und EBCDIC), ganze Zahlen (zum Beispiel das Komplement von eins oder das Komplement von zwei), und so weiter. Damit Computer mit verschiedenen Darstellungsformen miteinander kommunizieren können, kann man die auszutauschenden Datenstrukturen auf abstrakte Art definieren und ihnen »für unterwegs« noch eine Standardkodierung mitgeben. Die Darstellungsschicht übernimmt die Aufgabe, diese abstrakten Datenstrukturen handzuhaben und die interne Darstellungsform des Computers in die Standarddarstellung des Netzwerks zu konvertieren.

Außerdem beschäftigt sich die Darstellungsschicht auch mit anderen Aspekten der Informationsdarstellung. Hier kann man zum Beispiel über die Datenkompression die zu übertragende Anzahl an Bits reduzieren und über die Kryptographie für Vertraulichkeit und Authentizität sorgen.

1.4.7 Die Anwendungsschicht

Die **Anwendungsschicht** (application layer) enthält eine Vielzahl von häufig benötigten Protokollen. Es gibt zum Beispiel hunderte von nicht kompatiblen Terminalarten auf der ganzen Welt. Man stelle sich die Mühsal eines Ganzseiteneditors vor, der in einem Netzwerk mit vielen verschiedenen Terminalarten - jede mit anderem Bildschirmlayout, anderen Steuer-Sequenzen zum Einfügen und Löschen von Text, zum Bewegen des Cursors und so weiter - vernünftig arbeiten soll.

Eine mögliche Lösung dieses Problems ist die Definition eines abstrakten **virtuellen Netzwerkterminals**, auf das man Editoren und andere Programme anpassen kann. Damit dann jede Terminalart mitspielen kann, muß ein Stück Software geschrieben werden, das die Funktionen des virtuellen Netzwerkterminals auf das echte Terminal abbildet. Wenn zum Beispiel der Editor den Cursor des virtuellen Terminals in die obere linke Ecke des Bildschirms bewegt, so muß das Programm die passende Sequenz ausgeben, damit das echte Terminal seinen Cursor ebenfalls dorthin setzt. Die gesamte Software des virtuellen Terminals befindet sich in der Anwendungsschicht.

Eine weitere Funktion der Anwendungsschicht ist der Dateitransfer. Verschiedene Dateisysteme haben auch verschiedene Konventionen für Dateinamen, verschiedene Arten, Textzeilen darzustellen und so weiter. Wenn man eine Datei zwischen zwei verschiedenen Systemen hin- und hertransferieren will, muß man sich mit diesen und noch einigen anderen Inkompatibilitäten herumschlagen. Dies ist ebenso die Aufgabe der Anwendungsschicht wie die elektronische Post, die Auftragsferneingabe, die Verzeichnisabfrage und einige weitere Einrichtungen allgemeiner und spezieller Art.

1.4.8 Die Datenübertragung im OSI-Modell

In Abbildung 1.8 sehen wir an einem Beispiel, wie über das OSI-Modell Daten übertragen werden können. Der Senderprozeß hat ein paar Daten, die er dem Empfängerprozeß zukommen lassen will. Er gibt die Daten an die Anwendungsschicht weiter, welche den Anwendungsnachrichtenkopf *AH* (der Null sein kann) vorne an die Nachricht klebt und die entstehende Einheit an die Darstellungsschicht weitergibt.

Die Darstellungsschicht wandelt diese Einheit möglicherweise ein paar Mal um, setzt vielleicht einen weiteren Nachrichtenkopf vorne dran und gibt das Ergebnis an die Sitzungsschicht weiter. Es ist hierbei wichtig zu wissen, daß die Darstellungsschicht sich nicht darüber klar ist, welcher Teil der von der Anwendungsschicht gesendeten

Daten jetzt *AH* ist, sofern *AH* vorhanden ist, und welcher Teil richtige Anwenderdater sind. Soll sie auch nicht.

Dieser Vorgang wiederholt sich solange, bis die Daten die Bitübertragungsschich erreichen, von wo aus sie tatsächlich auf die Empfängermaschine übertragen werden Auf dieser Maschine werden die verschiedenen Nachrichtenköpfe nun einer nach der anderen wieder entfernt, während sich die Nachricht durch die verschiedenen Schichten nach oben bewegt, bis sie schließlich beim Empfängerprozeß ankommt.

Der maßgebliche Grundgedanke an der ganzen Sache ist, daß - obwohl die Datenübertragung in Abbildung 1.8 vertikaler Natur ist - jede Schicht so programmiert wird, als sei die Datenübertragung horizontal. Wenn zum Beispiel die Transportschicht des Senders eine Nachricht von der Sitzungsschicht erhält, fügt sie dieser einen Transportnachrichtenkopf an und schickt sie an die Transportschicht des Empfängers. Aus ihrer Sicht ist die Tatsache, daß sie die Nachricht tatsächlich an die Vermittlungsschicht auf ihrer eigenen Maschine weiterleiten muß, unwichtiges technisches Drumherum. Analog dazu wendet sich ein Uighur-sprechender Diplomat vor den Vereinten Nationen ja auch direkt an die anderen Diplomaten, die versammelt sind. Daß er eigentlich nur mit seinem Dolmetscher spricht, wird nur als technisches Detail gesehen.

1.5 Dienste

Die letztendliche Funktion jeder Schicht des OSI-Modells ist es, der nächsthöheren Schicht bestimmte Dienste anzubieten. In diesem Abschnitt werden wir uns genauer ansehen, was ein Dienst wirklich ist - vorher aber noch ein bißchen OSI-Terminologie.

1.5.1 OSI-Terminologie

Die aktiven Elemente jeder Schicht nennt man **Instanzen** (entities). Eine Instanz kann eine Software-Instanz (wie zum Beispiel ein Prozeß) oder eine Hardware-Instanz (wie zum Beispiel ein intelligenter I/O-Chip) sein. Instanzen derselben Schicht auf verschiedenen Maschinen heißen **Partnerinstanzen** (peer entities). Die Instanzen von Schicht 7 heißen **Anwendungsinstanzen**, die von Schicht 6 heißen **Darstellungsinstanzen**, und so weiter.

Über die Instanzen in Schicht *N* wird ein Dienst für die Schicht *N+1* implementiert. In diesem Fall ist Schicht *N* der **Dienstleister** (service provider) und Schicht *N+1* der **Dienstbenutzer** (service user). Für die Bereitstellung ihrer Dienste kann sich Schicht *N* die Dienste der Schicht *N-1* zu Nutze machen. Sie könnte weiterhin verschiedene Dienstklassen anbieten, zum Beispiel schnelle, teure Kommunikation und langsame, billige Kommunikation.

Dienste sind an den **Dienstzugangspunkten**, den sogenannten **SAPs** (service access points) verfügbar. Die SAPs von Schicht *N* sind die Stellen, an denen Schicht *N+1* die angebotenen Dienste in Anspruch nehmen kann. Jeder SAP hat eine Adresse, die ihn unverwechselbar kennzeichnet. Zur Verdeutlichung sind die SAPs des

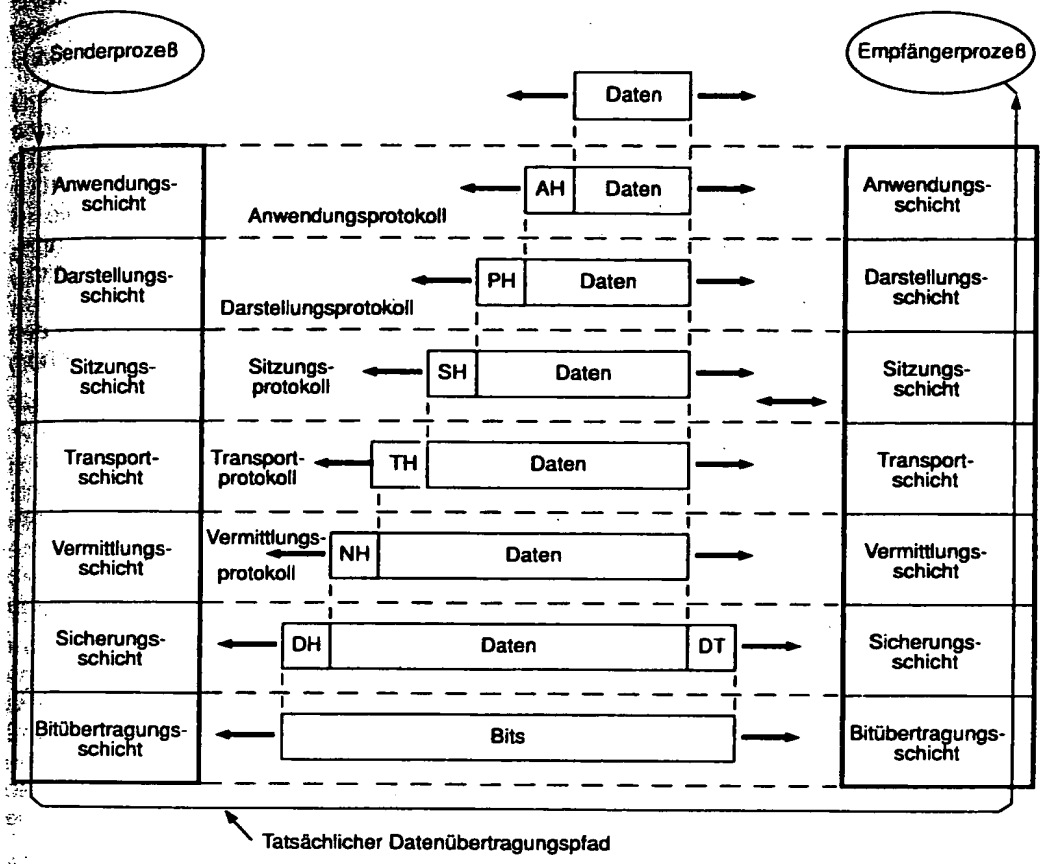


Abbildung 1.8: Ein Beispiel dafür, wie das OSI-Modell eingesetzt wird. Einige der Nachrichtenköpfe könnten leer sein. (verwendet mit frdl. Genehmigung von H.C. Folts)

Telefonsystems die Telefondosen, an die man modulare Telefone anschließen kann, und die SAP-Adressen sind die Telefonnummern, die zu je einer dieser Dosen gehören. Wenn Sie jemanden anrufen wollen, müssen Sie die SAP-Adresse wissen. Ähnlich ist es beim Postsystem, wo die SAP-Adressen die jeweiligen Straßennamen und

Hausnummern oder die Postfächer sind. Wenn Sie einen Brief schicken wollen, müssen Sie die SAP-Adresse des Empfängers kennen. Beim Berkeley-UNIX[1] sind die SAPs die Steckplätze und die SAP-Adressen die Steckplatznummern. Das SAP-Konzept wurde von Tomas u. a. (1987) genau abgehandelt.

Damit zwei Schichten Informationen austauschen können, muß ein genau vereinbarter Satz an Regeln die **Schnittstelle** umschreiben. Bei einer typischen Schnittstelle gibt die Instanz von Schicht *N+1* eine **Schnittstellendateneinheit**, die sogenannte **IDU** (interface data unit) über den SAP an die Instanz von Schicht *N* weiter, wie in Abbildung 1.9 gezeigt. Die IDU besteht aus einer **Dienstdateneinheit**, der sogenannten **SDU** (service data unit) und etwas Steuerinformation. Die SDU ist die Information, die durch das Netzwerk an die Partnerinstanz weitergegeben wird und dann bei Schicht *N+1* landet. Die Steuerinformation (zum Beispiel die Anzahl an Bytes der SDUs) wird benötigt, damit es die untere Schicht bei ihrer Arbeit leichter hat; sie gehört aber nicht zu den eigentlichen Daten.

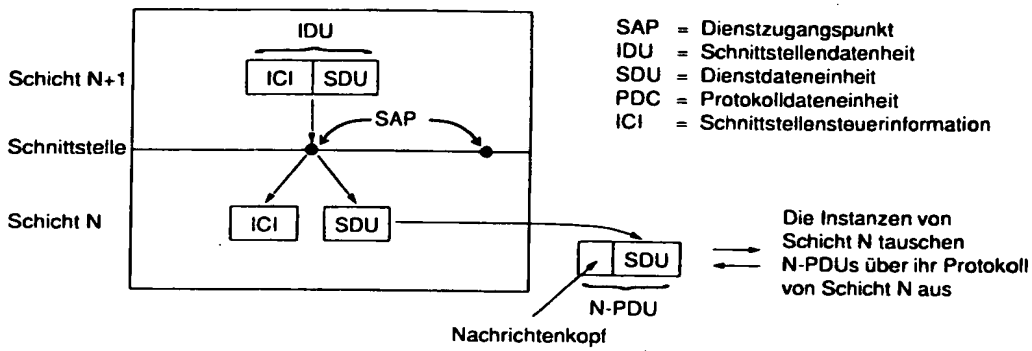


Abbildung 1.9: Die Beziehungen der Schichten an einer Schnittstelle

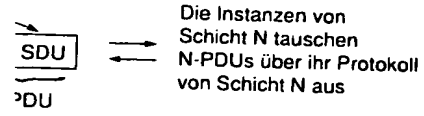
Für die Übertragung wird die Instanz von Schicht *N* die SDU möglicherweise in mehrere Teile zerstückeln müssen, von denen dann jeder einen Nachrichtenkopf bekommt und als separate **Protokolldateneinheit** oder **PDU** (protocol data unit) in einem Paket übertragen wird. Die Nachrichtenköpfe der PDU werden von den Partnerinstanzen für die Ausführung ihres Partnerprotokolls verwendet. Sie sagen aus, welche PDUs Daten enthalten und welche Steuerinformation, beinhalten Folgenummern

1 UNIX ist ein eingetragenes Warenzeichen von AT&T

...n Sie einen Brief schicken wolle...
...ennen. Beim Berkeley-UNIX[1] sind di...
...Steckplatznummern. Das SAP-Kon...
...andelt.

...können, muß ein genau vereinbarte...
...Bei einer typischen Schnittstelle gib...
...ndateneinheit, die sogenannte **IDU**...
...stanz von Schicht *N* weiter, wie in...
Dienstdateneinheit, der sogenann...
...rmation. Die SDU ist die Information...
...weitergegeben wird und dann be...
...n Beispiel die Anzahl an Bytes der...
...bei ihrer Arbeit leichter hat; sie gehört...

SAP = Dienstzugangspunkt
IDU = Schnittstellendatenheit
SDU = Dienstdateneinheit
PDC = Protokolldateneinheit
ICI = Schnittstellensteuerinformation



Die Instanzen von Schicht N tauschen N-PDUs über ihr Protokoll von Schicht N aus

...r Schichten an einer

...cht *N* die SDU möglicherweise in ...dann jeder einen Nachrichtenkopf ...t oder **PDU** (protocol data unit) in ...e der PDU werden von den Partner...lls verwendet. Sie sagen aus, wel...rmation, beinhalten Folgenummern

...und Zähler und so weiter. Die PDUs von Transport-, Sitzungs- und Anwendungsschicht heißen oft TPDUs, SPDUs oder APDUs. Über die anderen PDUs redet eigentlich kein Mensch.

Diese Sprache wird oft als **Internationalesbürokratenkauderwelsch** bezeichnet. Wir werden sie so gut es geht zugunsten der wirklich von den Computerwissenschaftlern und Informatikern verwendeten Fachausdrücke vermeiden.

1.5.2 Verbindungsorientierte und verbindungsunabhängige Dienste

Jede Schicht kann den Schichten darüber zwei verschiedene Dienstarten anbieten: verbindungsorientierte und verbindungsunabhängige Dienste. In diesem Abschnitt werden wir uns diese beiden Typen genauer ansehen und die Unterschiede zwischen ihnen analysieren. Weiterführendes Material findet sich in Bucciarelli und Caneschi (1985) und Chapin (1983).

Der **verbindungsorientierte Dienst** (connection-oriented service) wurde nach dem Telefonsystem entwickelt. Wenn Sie mit jemandem sprechen wollen, nehmen Sie den Hörer ab, wählen eine Nummer, sprechen und hängen dann wieder auf. Genauso muß ein Dienstbenutzer, der einen verbindungsorientierten Netzwerkdienst in Anspruch nehmen will, erst eine Verbindung aufbauen, diese Verbindung dann benutzen und schließlich wieder auflösen. Der Grundaspekt bei einer solchen Verbindung ist, daß sie wie ein Rohr arbeitet: Der Sender stopft auf der einen Seite etwas hinein, und der Empfänger holt es auf der anderen Seite in derselben Reihenfolge wieder heraus.

Verbindungsunabhängige Dienste (connectionless services) hingegen arbeiten nach dem Postprinzip. Jede Nachricht (jeder Brief) trägt die volle Bestimmungsadresse und wird unabhängig von allen anderen durch das System geschleust. Wenn zwei Boten zum selben Empfänger geschickt werden, so kommt normalerweise derjenige, der zuerst geschickt wurde, auch zuerst an. Es kann allerdings auch vorkommen, daß der erste irgendwo aufgehalten wird und deshalb der zweite zuerst ankommt. Bei einem verbindungsorientierten Dienst ist dies unmöglich.

Jeder Dienst kann qualitativ charakterisiert werden. Manche Dienste sind im Hinblick auf Datenverluste recht zuverlässig. Gewöhnlich wird dies dadurch erreicht, daß der Empfänger den Empfang jeder Nachricht bestätigen muß, damit der Sender weiß, daß sie angekommen ist. Dieses Rückmeldeverfahren verursacht Gemeinkosten und Verzögerungen, die oft durch die Sache gerechtfertigt, manchmal aber auch nicht erwünscht sind.

Eine typische Situation, in der ein zuverlässiger, verbindungsorientierter Dienst gefragt ist, ist der Dateitransfer. Der Besitzer der Datei möchte gern genau wissen, daß alle Bits wohlbehalten und in der richtigen Reihenfolge ankommen. Es gibt wahrscheinlich sehr wenige Dateitransferkunden, die einen Dienst vorziehen würden, der gelegentlich ein paar Bits verhunzt oder verliert, auch wenn er wesentlich schneller wäre.

Meta 1002

Beim zuverlässigen verbindungsorientierten Dienst gibt es zwei kleinere Spielarten: Nachrichtenfolgen und Byte-Ströme. Beim ersteren werden die Nachrichtengrenzen beibehalten. Wenn zwei Nachrichten mit 1K gesendet werden, so kommen sie auch als zwei separate Nachrichten mit 1K an und nie als eine Nachricht mit 2K. Beim letzteren ist die Verbindung einfach ein Strom von Bytes ohne Nachrichtengrenzen. Wenn beim Empfänger 2K Bytes ankommen, kann man einfach nicht feststellen, ob sie als eine Nachricht mit 2K, zwei mit 1K oder 2048 mit 1Byte abgeschickt wurden. Würden die Seiten eines Buches über ein Netzwerk an ein Fotosatzgerät geschickt werden, könnte es wichtig sein, die Nachrichtengrenzen beizubehalten. Wenn andrerseits ein Terminal mit einem rechnerfernen Teilnehmerrechensystem verbunden wird, so ist dafür lediglich ein Byte-Strom vom Terminal zum Computer nötig.

Wie bereits vorher erwähnt sind bei manchen Anwendungen die durch Rückmeldungen hervorgerufenen Verzögerungen nicht akzeptabel. So eine Anwendung ist zum Beispiel die Kommunikation von digitalisierten Stimmen. Ein Telefonbenutzer hört tausendmal lieber von Zeit zu Zeit ein Rauschen oder ein verstümmeltes Wort, als daß er eine Verzögerung für eine Empfangsbestätigung hinnimmt.

Nicht alle Anwendungen machen Verbindungen erforderlich. Jetzt, wo zum Beispiel die elektronische Post immer weiter ins Alltagsleben eindringt, ist da der Schritt zur elektronischen Wurfsendung noch weit? Der Versender der elektronischen Wurfsendung möchte wahrscheinlich nicht für jedes einzelne Stück eine Verbindung aufbauen und sie dann wieder auflösen. Es ist auch nicht nötig, eine hundertprozentige Empfangsgarantie zu geben, besonders, wenn sie mehr kostet. Es soll lediglich eine Einzelnachricht geschickt werden, deren Empfangschancen relativ hoch sind. Nicht zuverlässige (also nicht bestätigte) verbindungsunabhängige Dienste nennt man oft **Datagrammdienste**, analog zum Telegrammdienst, bei dem der Sender auch keine Rückmeldung erhält.

In anderen Situationen ist es zwar wünschenswert, für eine kurze Nachricht nicht extra eine eigene Verbindung aufbauen zu müssen, dafür ist aber die Zuverlässigkeit von grundlegender Bedeutung. Für Anwendungen dieser Art kommt der **bestätigte Datagrammdienst** (acknowledged datagram service) in Frage. Das ist genauso, als würde man einen eingeschriebenen Brief mit Rückschein verschicken. Wenn der Empfänger den Rückschein erhält, so kann er absolut sicher sein, daß der gewünschte Empfänger den Brief erhalten hat.

Ein weiterer Dienst ist der Anfrage-Antwort-Dienst (request-reply service). Hier schickt der Sender ein einfaches Datagramm, das eine Anfrage enthält; die Rückmeldung enthält die Antwort. Eine Anfrage bei der städtischen Bücherei, in der man sich beispielsweise danach erkundigt, wo Uighur gesprochen wird, würde in diese Kategorie fallen. In Abbildung 1.10 werden die oben genannten Dienste noch einmal zusammengefaßt.

	Beispiel	
zuverlässiger Nachrichtenstrom	Folge von Seiten	verbindungsorientiert
zuverlässiger Bytestrom	Rechner-Fernverbindung	
unzuverlässige Verbindung	digitalisierte Stimme	
unzuverlässiges Datagramm	elektronische Wurfsendung	verbindungsunabhängig
bestätigtes Datagramm	eingeschriebener Brief	
Anfrage-Antwort	Datenbankabfrage	

Abbildung 1.10: Sechs verschiedene Dienstarten

1.5.3 Dienstelemente

Ein Dienst wird formal durch eine Reihe von **Dienstelementen** (primitives) oder Operationen gekennzeichnet, über die es dem Anwender oder der Instanz möglich ist, den Dienst in Anspruch zu nehmen. Diese Dienstelemente sorgen dafür, daß der Dienst eine bestimmte Handlung durchführt oder daß er auf eine Handlung der Partnerinstanz in bestimmter Weise reagiert. Innerhalb des OSI-Modells kann man die Dienstelemente in vier Klassen aufteilen. Diese zeigt Abbildung 1.11.

Dienstelement	Bedeutung
Anforderung (request)	Eine Instanz möchte einen Dienst dazu veranlassen, etwas bestimmtes zu tun
Anzeige (indication)	Eine Instanz soll über ein Ereignis informiert werden
Antwort (response)	Eine Instanz möchte auf ein Ereignis reagieren
Bestätigung (confirm)	Eine Instanz soll über eine Anforderung informiert werden

Abbildung 1.11: Vier Klassen von Dienstelementen

Die erste Klasse von Dienstelementen ist das *Anforderungs*element (request). Damit erledigt man eine bestimmte Arbeit, zum Beispiel den Aufbau einer Verbindung oder das Senden von Daten. Wenn diese Arbeit erledigt worden ist, bekommt die Partner-

instanz über eine *Anzeige* (indication) ein Signal. So erhält zum Beispiel nach einem *CONNECT.request* (in OSI-Schreibweise) die angesprochene Instanz eine *CONNECT.indication*, in der angekündigt wird, daß jemand mit ihr in Verbindung treten will. Die Instanz, welche die *CONNECT.indication* bekommt, verwendet dann eine *CONNECT.response* (eine Antwort), um mitzuteilen, ob sie die Verbindungsaufforderung annehmen oder ablehnen will. Die Instanz, die das ursprüngliche *CONNECT.request* ausgegeben hat, erfährt in jedem Fall über das *CONNECT.confirm* (Bestätigung), was passiert ist.

Dienstelemente können Parameter haben, und die meisten haben auch welche. Die Parameter für ein *CONNECT.request* könnten die Maschine, zu der Verbindung aufgenommen werden soll, die Art des Dienstes und die maximale Nachrichtengröße für diese Verbindung festlegen. Die Parameter für eine *CONNECT.indication* könnten die Kennung des Anrufenden, die Art des gewünschten Dienstes und die vorgeschlagene maximale Nachrichtengröße enthalten. Sollte die angerufene Instanz nicht mit der vorgeschlagenen Nachrichtengröße einverstanden sein, so könnte sie in ihrem *Antwort*element (response) einen Gegenvorschlag machen. Die Einzelheiten dieser **Absprache** (negotiation) sind Teil des Protokolls. Für den Fall, daß es zwei verschiedene Vorschläge über die maximale Nachrichtengröße gibt, könnte das Protokoll zum Beispiel vorschreiben, daß immer der kleinere Wert gewählt wird.

Als kleinen Ausflug in die Terminologie möchte ich erwähnen, daß das OSI-Modell peinlichst die Begriffe »eine Verbindung öffnen (open a connection)« und »eine Verbindung schließen (close a connection)« vermeidet, denn für Elektroingenieure hat ein »offener« Schaltkreis eine Lücke oder Unterbrechung. Die Elektrizität kann nur über »geschlossene« Schaltkreise fließen. Informatiker könnten sich nie damit abfinden, daß ihre Information über einen geschlossenen Schaltkreis fließen soll. Damit beide Lager ihren Frieden finden, lauten die offiziellen Ausdrücke »eine Verbindung aufbauen« und »eine Verbindung auflösen«.

Dienste können entweder **bestätigt** (confirmed) oder **unbestätigt** (unconfirmed) sein. Bei einem bestätigten Dienst gibt es eine *Anforderung*, eine *Anzeige*, eine *Antwort* und eine *Bestätigung*. Bei einem unbestätigten Dienst gibt es lediglich eine *Anforderung* und eine *Anzeige*. *CONNECT* ist immer ein bestätigter Dienst, weil der Partner am anderen Ende seine Zustimmung zum Aufbau der Verbindung geben muß. Der Datentransfer hingegen kann bestätigt oder unbestätigt ablaufen, je nach dem, ob der Sender eine Rückmeldung braucht. Beide Dienstarten werden in Netzwerken eingesetzt.

Damit das Dienstkonzept etwas durchsichtiger wird, wollen wir uns als Beispiel einen einfachen verbindungsorientierten Dienst mit acht Dienstelementen ansehen:

1. *CONNECT.request* - den Aufbau einer Verbindung anfordern
2. *CONNECT.indication* - die angerufene Partei verständigen

3. *CONNECT.res*
   anzunehmen o
4. *CONNECTION*
   angenommen v
5. *DATA.request* -
6. *DATA.indicatio*
7. *DISCONNECT.*
8. *DISCONNECT.*
   Auflösung mitte

In diesem Beispiel ist *CONN*
erforderlich), *DISCONNECT*

Eine Analogie aus dem Tel
verdeutlichen. Man nehme
anzurufen und sie zum Tee z

1. *CONNECT.requ*
2. *CONNECT.indic*
3. *CONNECT.resp*
4. *CONNECTION.*
5. *DATA.request* -
6. *DATA.indication*
7. *DATA.request* -
8. *DATA.indication*
9. *DISCONNECT.r*
10. *DISCONNECT.i*

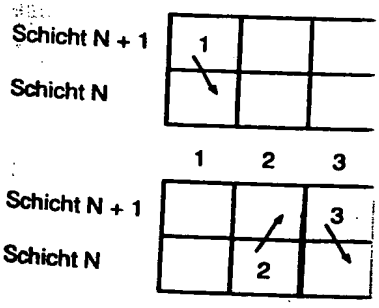


Abbildung 1.12.
einladen würde. Die Z
Dienstelemente,

3. *CONNECT.response* - wird vom Angerufenen verwendet, um Anrufe anzunehmen oder abzulehnen
4. *CONNECTION.confirm* - dem Anrufer mitteilen, ob der Anruf angenommen wurde
5. *DATA.request* - die Sendung der Daten anfordern
6. *DATA.indication* - die Ankunft der Daten signalisieren
7. *DISCONNECT.request* - die Auflösung einer Verbindung anfordern
8. *DISCONNECT.indication* - dem Partner die Anforderung der Auflösung mitteilen

In diesem Beispiel ist *CONNECT* ein bestätigter Dienst (eine ausdrückliche Antwort ist erforderlich), *DISCONNECT* hingegen ist unbestätigt (keine Antwort).

Eine Analogie aus dem Telefonwesen könnte den Einsatz dieser Dienstelemente verdeutlichen. Man nehme hierfür die Schritte, die dafür nötig sind, Tante Gertrud anzurufen und sie zum Tee zu sich nach Hause einzuladen.

1. *CONNECT.request* - Tante Gertruds Telefonnummer wählen
2. *CONNECT.indication* - ihr Telefon läutet
3. *CONNECT.response* - sie nimmt den Hörer ab
4. *CONNECTION.confirm* - Sie hören, daß das Läuten aufhört
5. *DATA.request* - Sie laden sie zum Tee ein
6. *DATA.indication* - sie hört Ihre Einladung
7. *DATA.request* - sie sagt, daß sie gerne kommt
8. *DATA.indication* - Sie hören ihr Einverständnis
9. *DISCONNECT.request* - Sie legen auf
10. *DISCONNECT.indication* - sie hört dies und legt ebenfalls auf

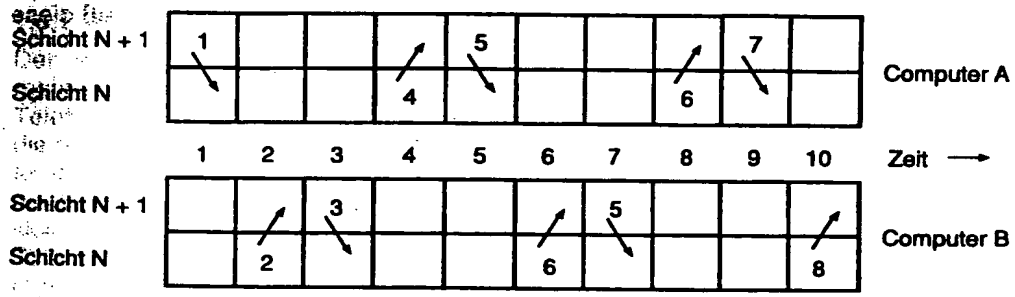


Abbildung 1.12: Wie ein Computer Tante Gertrud zum Tee einladen würde. Die Zahlen am Ende der Pfeile repräsentieren die acht Dienstelemente, die in diesem Abschnitt besprochen wurden.

Abbildung 1.12 zeigt dieselbe Folge von Schritten als Reihe von Dienstelem inklusive der abschließenden Bestätigung der Verbindungsauflösung. Jeder beinhaltet einen Dialog zwischen zwei Schichten auf einem der Computer. *Anforderung* oder *Antwort* erzeugt wenig später eine *Anzeige* oder *Bestätigu* anderen Seite. In diesem Beispiel befinden sich die Dienstbenutzer (Sie und Gertrud) in Schicht *N+1*, und der Dienstleister (die Post) in Schicht *N*.

1.5.4 Die Beziehung zwischen Diensten und Protokollen

Dienste und Protokolle haben verschiedene Konzepte, auch wenn sie oft verwe werden. Ihre Unterscheidung ist jedoch so wichtig, daß wir sie hier nochmals be wollen. Ein *Dienst* ist ein Satz von Dienstelementen (Operationen), die von Schicht der über ihr liegenden Schicht zur Verfügung gestellt werden. Der Diens aus, welche Operationen diese Schicht für ihre Anwender ausführen will, aber e nicht das geringste darüber aus, wie diese Operationen implementiert sind. Ein D läuft über die Schnittstelle zwischen zwei Schichten, wobei die untere Schicl Dienstleister und die obere Schicht der Dienstbenutzer ist.

Ein *Protokoll* hingegen ist das Regelgefüge, welches das Format und die Bede der von den Partnerinstanzen innerhalb einer Schicht ausgetauschten Rahmen, ! te oder Nachrichten festlegt. Mit Protokollen führen die Instanzen ihre defin Dienste aus. Sie können ihre Protokolle nach Belieben verändern, solange sie ihre für den Dienstbenutzer sichtbaren Dienste verändern. Ein Protokoll bezieh auf die *Ausführung* des Dienstes und ist als solches für den Dienstbenutzer sichtbar.

Viele der vor dem OSI-Modell entwickelten Protokolle unterschieden den Dienst vom Protokoll. Daraus konnte eine typische Schicht ein Dienstelement na *SEND_PACKET* haben, bei dem der Dienstbenutzer einen Zeiger auf ein komp Paket setzte. Das wiederum bedeutete, daß jede Änderung des Protokolls sofc den Dienstbenutzer erkennbar war. Es ist inzwischen allgemein anerkannt, daß Art der Entwicklung gehobener Bockmist ist.

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☑ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☑ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

BEST AVAILABLE COPY

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Ruland Christoph:
Informationssicherheit in Datennetzen/Christoph Ruland. –
Bergheim: DATACOM-Verl., 1993
(DATACOM-Fachbuchreihe)
ISBN 3-89238-081-3

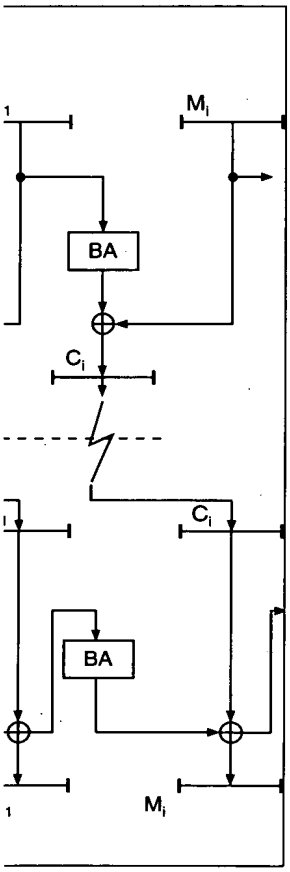
Copyright 1993 by DATACOM-Verlag
Nachdruck, auch auszugsweise, ist genehmigungspflichtig.

DATACOM-VERLAG
Klaus Lipinski
Postfach 15 02
D-50105 Bergheim
Tel. (0 22 71) 6 08-0
Fax (0 22 71) 6 08-2 90

selungsseite die 0-Folge als
hron, und die Entschlüsse-
m. Der Datentransfer erfolgt
ations verlust bemerkt wird.
sation.

triebsarten

sarten gibt es noch weitere
Modus.



Cipher Block Chaining with Error Propagation (PCBC-Modus)

Dieser Modus soll eine unbegrenzte Fehlerfortpflanzung unterstützen und somit zur Erkennung der Datenunversehrtheit dienen. Dieser Modus ist ähnlich dem CBC-Modus, allerdings wird der aktuelle Klartextblock nicht mit dem vorhergehenden Schlüsseltextblock, sondern der Schlüsseltextblock wird mit dem nachfolgenden Klartextblock verknüpft.

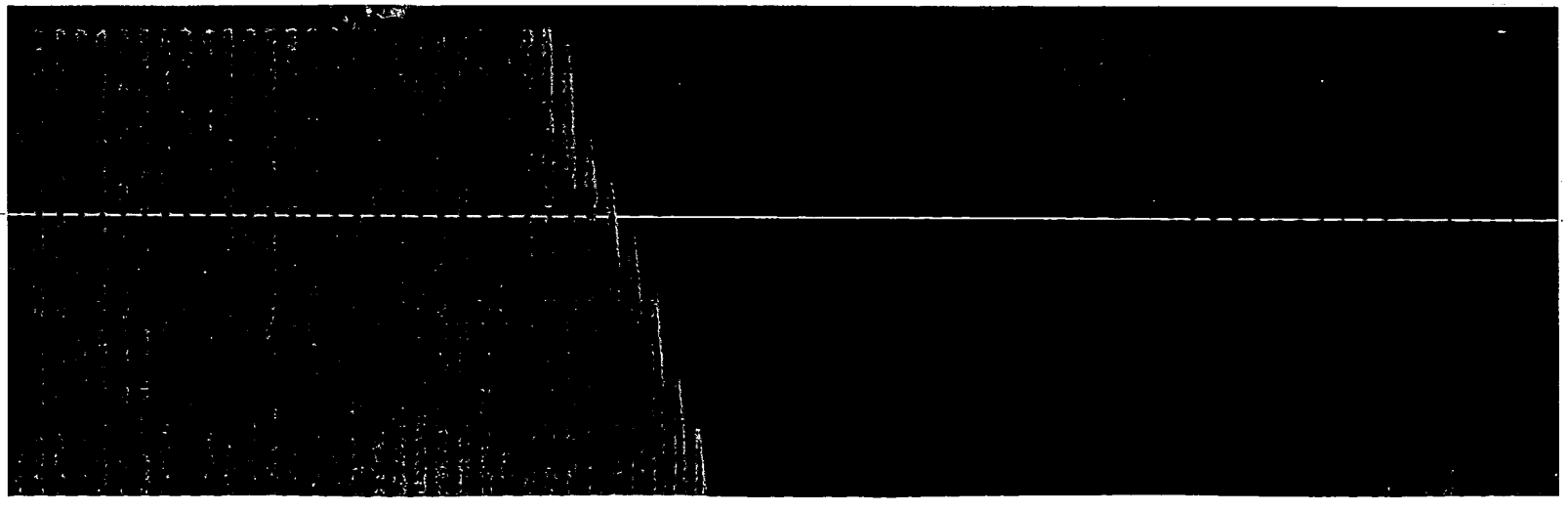
Durch diese Maßnahme wird auf der Empfangsseite eine unbegrenzte Fehlerfortpflanzung erreicht, wenn der Schlüsseltext gestört oder verändert wird (siehe Fehlerfortpflanzungseigenschaften des CBC-Modus').

Auf diese Weise können Vertraulichkeit und Erkennung der Datenunversehrtheit in einem Arbeitsschritt erledigt werden, wenn auf Grund von Redundanz im letzten Block eine erfolgreiche Entschlüsselung festgestellt wird.

5.6. Der Message Authentication Code (MAC)

5.6.1. Die Berechnung des MAC's

Bei der MAC-Berechnung wird ein Message Authentication Code unter Verwendung eines geheimen Schlüssels und eines kryptographischen N-Bit-Blockalgorithmus' berechnet. Der MAC ist eine kryptographische Prüfsumme, mit der Veränderungen der Nachricht erkannt werden sollen. Daher wird der MAC auch als MDC (Message Error Detetection Code) verwendet. Eine Authentikation der Nachricht, die die Bezeichnung MAC suggeriert, ist nur eingeschränkt möglich (siehe Kap. 5.6.2). Die MAC-Berechnung wurde in der ISO 9797 (siehe Kap. 21) festgelegt. Der bekannte, weitverbreitete Standard ANSI X9.9 (siehe Kap. 21) ist ein Spezialfall dieses Verfahrens: die Blocklänge beträgt 64 Bit, und es wird der DES-Algorithmus eingesetzt. Die Prüfsumme wird der Nachricht angehängt. Die Nachricht selber bleibt im Klartext erhalten. Die "Nachricht" können auch ausführbare Programme sein, die mit dieser



Prüfsumme (z.B. gegen eine Verseuchung mit Viren) geschützt werden und natürlich trotz des Schutzes lesbar, d.h. ausführbar bleiben müssen (vgl. auch Hashfunktionen in Kap. 5.9.).

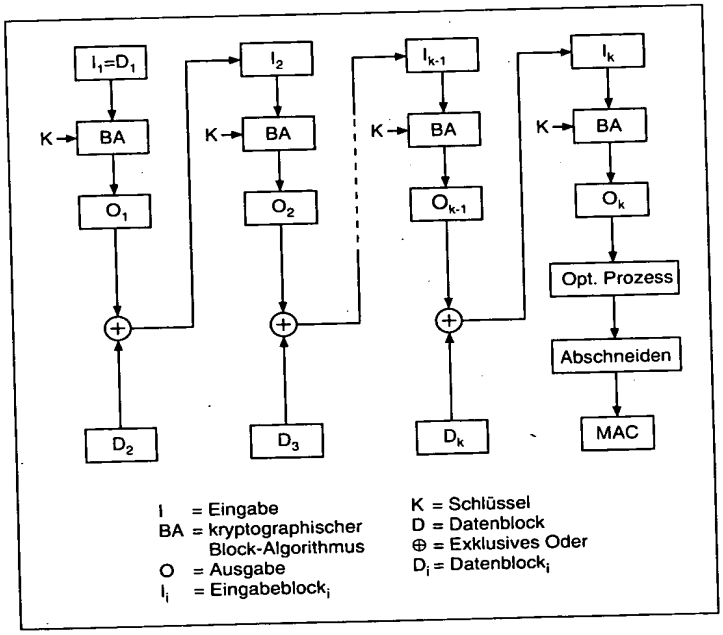
Die Nachricht D wird in die N-Bit-Blöcke D_1, D_2,...D_k aufgeteilt, wobei der letzte Datenblock evtl. durch Padding aufgefüllt werden muß (siehe Kap.5.4.).

Dieses Padding wird aber nur temporär für die Berechnung der Prüfsumme auf Sender- und Empfängerseite benötigt, muß also nicht der Nachricht für die Übertragung oder Speicherung tatsächlich hinzugefügt werden.

Wenn derselbe Algorithmus zur MAC-Berechnung und zur Verschlüsselung benutzt wird, sollten für beide Sicherheitsmechanismen verschiedene Schlüssel gewählt werden.

Nach dem letzten Schritt kann eine zusätzliche optionale Operation folgen. In ANSI X9.9 wird O_k (optional) (siehe Bild 5.13.) mit einem weiteren geheimen Schlüssel entschlüsselt und anschließend mit dem in den anderen Schritten verwendeten Schlüssel K nochmals verschlüsselt.

Bild 5.13
MAC-Berechnung



Von den erzeugten N Bits werden die linken (höchstwertigen) m Bit als MAC genommen. Dabei sollte nach Möglichkeit m so groß wie möglich, d.h. am besten N sein.

Der Sender berechnet den MAC über die zu sendende Nachricht, der Empfänger über die empfangene Nachricht. Der Empfänger vergleicht die empfangene Prüfsumme mit der von ihm selbst berechneten. Bei Gleichheit ist die Nachricht ungestört

; mit Viren) geschützt wer-
ıar, d.h. ausführbar bleiben
ı. 5.9.).

ke D_1, D_2,...D_k aufgeteilt,
Padding aufgefüllt werden

ir die Berechnung der Prüf-
ıenötigt, muß also nicht der
herung tatsächlich hinzuge-

Berechnung und zur Ver-
le Sicherheitsmechanismen

tzliche optionale Operation
siehe Bild 5.13.) mit einem
t und anschließend mit dem
ichlüssel K nochmals ver-

Von den erzeugten N Bits werden die linken (höchstwertigen) m Bit als MAC genommen. Dabei sollte nach Möglichkeit m so groß wie möglich, d.h. am besten N sein.

Der Sender berechnet den MAC über die zu sendende Nachricht, der Empfänger über die empfangene Nachricht. Der Empfänger vergleicht die empfangene Prüfsumme mit der von ihm selbst berechneten. Bei Gleichheit ist die Nachricht ungestört

tragen worden und stammt von demjenigen, der über denselben geheimen Schlüssel K verfügt.

5.6.2. Die Subjektivität des MAC's

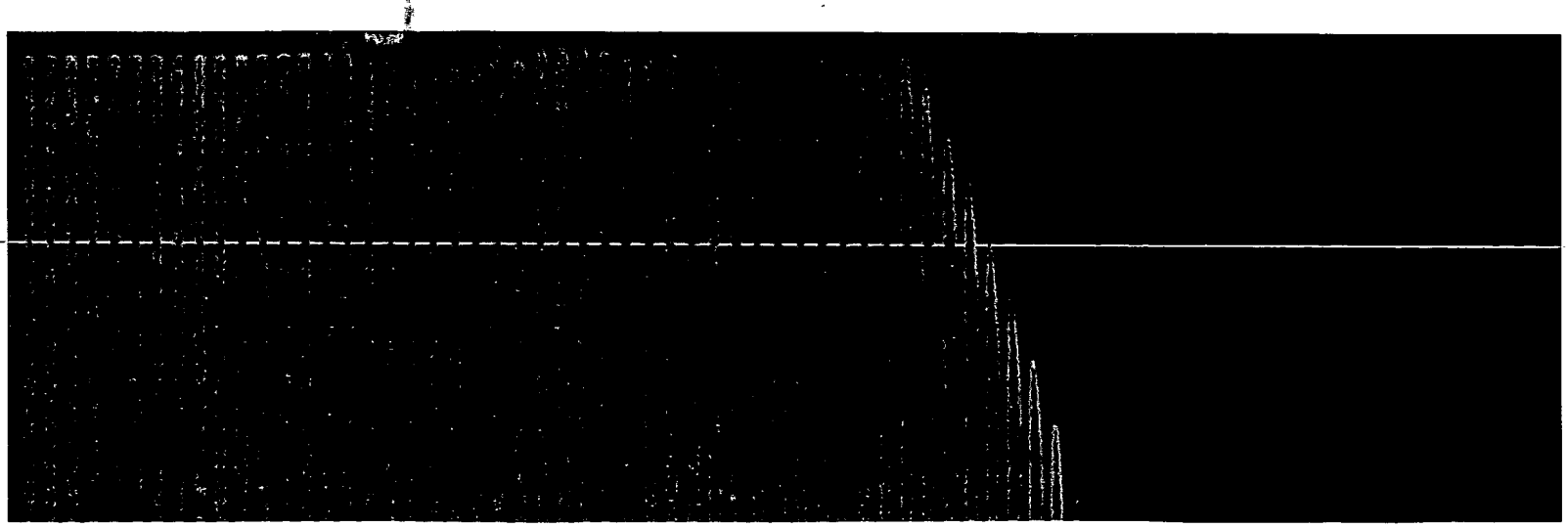
Wenn der Empfänger eine Nachricht erhält, der ein MAC hinzugefügt wurde, so kann er – bei positivem Überprüfungsergebnis – sicher sein, daß die Nachricht von jemandem stammt, der über den gemeinsamen geheimen Schlüssel verfügt, der für die MAC-Berechnung verwendet wurde. Aus Sicht des Empfängers ist die Nachricht authentisch. Handelt es sich um zwei Kommunikationspartner, von denen einer nicht fair arbeitet, so ist es möglich, daß ein Kommunikationspartner behauptet, eine bestimmte Nachricht mit MAC gesendet zu haben, während der andere Empfänger behauptet, eine andere Nachricht – ebenfalls mit MAC gesichert – empfangen zu haben. Ein objektiver Dritter, z.B. ein Richter, der den Streitfall klären soll, kann nicht entscheiden, wer von beiden Recht hat, da beide in der Lage waren, eine beliebige Nachricht mit dem passenden MAC zu versehen.

Wenn sich zwei Teilnehmer einen Schlüssel teilen, ist dies für eine objektive Authentikation bereits einer zuviel, d.h. eine MAC-Berechnung zur Gewährleistung der Authentikation ist nur geeignet, wenn sich die beteiligten Partner gegenseitig vertrauen, bzw. in geschlossenen Systemen. In offenen Systemen ist diese Methode nicht geeignet, um eine Authentikation auch zwischen beliebigen Partnern, d. h. auch konkurrierenden, zu gewährleisten. In diesem Fall werden elektronische Unterschriften auf der Basis asymmetrischer Verfahren (siehe Kap. 5.10.) oder zusätzliche Urheber- oder Empfängernachweise (siehe Kap. 9.7., Kap. 9.8.) benötigt.

Da der Message Authentication Code keine objektive Authentikation einer Nachricht gewährleistet, verdient er eigentlich nicht diesen Titel. Andererseits ist diese Bezeichnung für die nach Kap. 5.6.1. berechnete Prüfsumme seit langer Zeit weit verbreitet.

5.7. Zufallszahlengenerierung

Ein wesentliches kryptographisches Grundelement sind Zufallszahlen. Aus ihnen werden Ereignisse gebildet, die für den zu bearbeiten-



Nachricht sein, und die anderen an die Sicherheit von Pseudo-Zufallszahlengeneratoren gestellten Forderungen (siehe Kap. 5.7.) müssen ebenfalls erfüllt sein.

Diese Output-Folge wird auf der Verschlüsselungsseite mit dem Klartext verknüpft, i.a. durch XOR.

Auf der Entschlüsselungsseite wird der Pseudo-Zufallszahlengenerator mit demselben geheimen Schlüssel vorbelegt, und die Output-Folge mit dem Schlüsseltext in der Weise verknüpft, daß man den Klartext erhält, i.a. wiederum durch XOR.

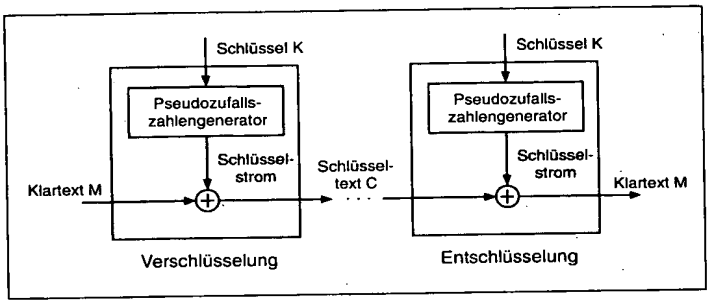


Bild 5.17 Bitstrom-Verschlüsselung

5.9. Hashfunktionen

Hashfunktionen werden in vielfältigem Zusammenhang von Sicherheitsverfahren angewendet, z.B. zur Unterstützung der Authentikation, der Erkennung der Datenunversehrtheit oder dem Urheber- und Empfängernachweis.

Dabei wird einem beliebig langen Inputstring, z. B. einer Nachricht, einem File oder einem Programm ein Outputstring fester Länge zugeordnet. Die hier betrachteten Hashfunktionen sollen gleichzeitig Einwegfunktionen sein: es darf nicht möglich sein, umgekehrt zu einem gegebenen Funktionswert einen passenden Inputwert zu berechnen.

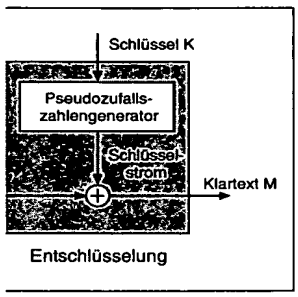
Außerdem wird Kollisionsfreiheit gefordert: es darf ebenfalls nicht möglich sein, zwei verschiedene Inputstrings zu finden, die denselben Outputstring ergeben, oder anders ausgedrückt:

wenn eine Nachricht mit einer Prüfsumme gegeben ist, darf es nicht möglich sein, auf systematische Weise in einem vernünftigen Zeit-

lie Sicherheit von Pseudo-
derungen (siehe Kap. 5.7.)

schlüsselungsseite mit dem

Pseudo-Zufallszahlengenera-
vorbelegt, und die Output-
ise verknüpft, daß man den
R.



Zusammenhang von Sicher-
nterstützung der Authentika-
theit oder dem Urheber- und

string, z. B. einer Nachricht,
tputstring fester Länge zuge-
onen sollen gleichzeitig Ein-
ch sein, umgekehrt zu einem
en Inputwert zu berechnen.

dert: es darf ebenfalls nicht
trings zu finden, die densel-
usgedrückt:

ne gegeben ist, darf es nicht
in einem vernünftigen Zeit-

kaum eine andere Nachricht zu finden, die dieselbe Prüfsumme ergibt. Sonst könnte ja ein Angreifer unbemerkt die originale Nachricht gegen die neugefundene Nachricht austauschen. Natürlich gibt es sehr viele Nachrichten, die alle dieselbe Prüfsumme ergeben, aber es darf trotzdem nicht ein derartiges Pärchen gefunden werden.

Solche Hashfunktionen werden z.B. benötigt,

- um eine Nachricht auf einen kurzen "Fingerabdruck" zu komprimieren, der als Input für elektronische Unterschriftsverfahren dient (siehe Kap. 5.10.2.), oder um einen gegebenen Wert zu bestätigen, ohne den Wert selber offenzulegen,
- oder sogar als Pseudozufallszahlengenerator, da der Hashwert ja wegen der geforderten Kollisionsfreiheit zufälligen Charakter haben muß.

"Es darf nicht möglich sein", einen passenden Inputwert oder zu einem Inputwert einen zweiten passenden Inputwert zu finden, bedeutet, daß dies mit vertretbarem Aufwand nicht möglich sein darf. Dieser "vertretbare Aufwand" hängt vom Entwicklungsstand der Ressourcen und den Sicherheitsanforderungen des Anwenders ab.

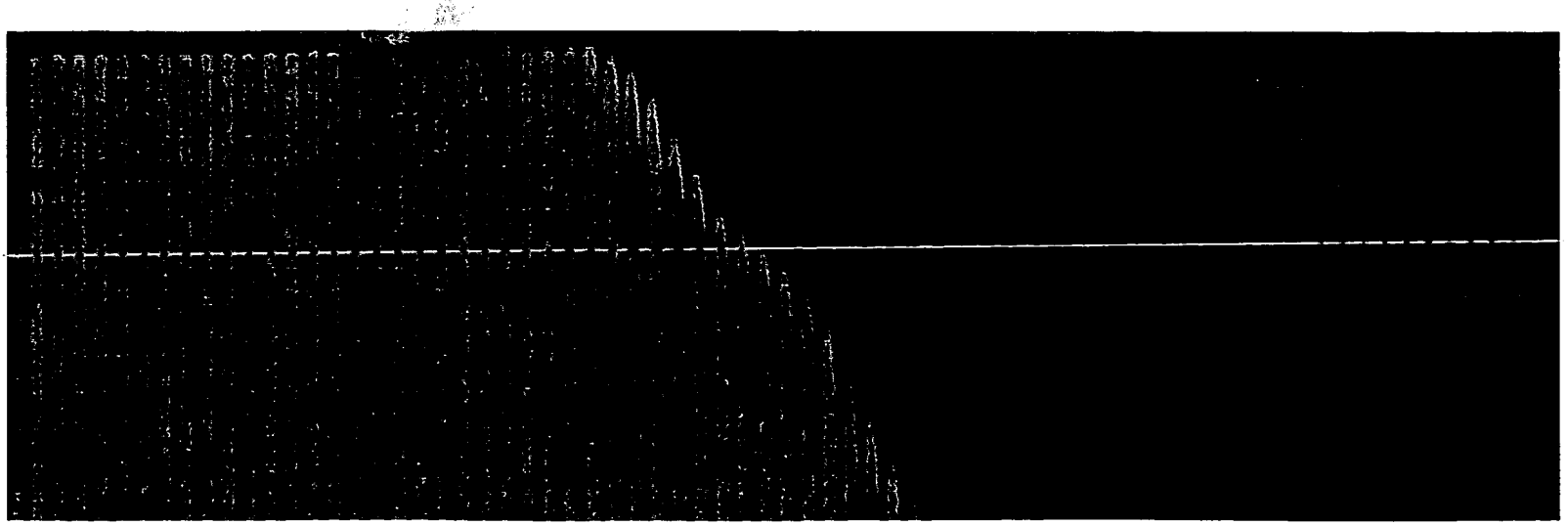
Der Output der Hashfunktion heißt Hashcode. Manchmal wird der Hashcode auch Hashergebnis oder Hashwert genannt und z.B. als Modification- oder Manipulation Detection Code (MDC), Checksumme oder Prüfsumme verwendet.

Der Hashcode darf nicht mit dem Message Authentication Code verwechselt werden (MAC), dessen Ziel die (subjektive) Authentikation einer Nachricht ist, und zu dessen Berechnung ein geheimer Schlüssel nötig ist (siehe Kap.5.6.).

Die Hashfunktion dagegen ist i.a. schlüsselunabhängig und öffentlich bekannt.

Oft ist ein Initialisierungsvektor für die Hashfunktion notwendig, z.B. ein fester Initialisierungsvektor, eine Reihenfolgenummer, Datum/Uhrzeit, eine Zufallszahl oder ein Wert, der aus den zu "hashenden" Daten abgeleitet wird.

Natürlich müssen derjenige, der den Hashcode berechnet, und derjenige, der den Hashcode überprüft, mit demselben Initialisierungsvektor starten, d.h. es müssen entsprechende Absprachen getroffen werden.



Die einzigen Hashfunktionen, die bisher standardisiert wurden, sind die im folgenden Kapitel 5.9.1. beschriebenen. In Bearbeitung der Normungsgremien ist zur Zeit eine "dedicated" Hashfunktion (Secure Hash Function SHS, siehe Kap. 21) passend zum NIST-Standardisierungsvorschlag für Elektronische Unterschriften (siehe Kap. 5.12.2.) sowie eine "Hash Function Using Modular Arithmetic" (square mod n) (siehe Kap. 21). Die "square mod n"-Funktion ist eine verbesserte Version der Hashfunktion aus CCITT X.509 (siehe Kap. 17.2.).

Weitere übliche Hashfunktionen sind die FEAL-Hashfunktion [MIYA90], der MD4 [RIVE90] und MD5 von R. Rivest [RIVE91], sowie die AR Fingerprint Function [ARL90], die ähnlich wie die folgenden Hashfunktionen (siehe Kap. 5.9.1.) auf dem DES-Algorithmus basiert und einen Hashwert von 128 Bits liefert.

5.9.1. Hashfunktionen mit blockorientierten Verschlüsselungsalgorithmen

Die Hashfunktionen, die in der ISO 10118-2 (siehe Kap. 21) definiert werden, beruhen auf der Verwendung von symmetrischen Verschlüsselungsalgorithmen, die jeweils N Bit Klartext in N Bit Schlüsseltext überführen, also sog. N-Bit-Blockalgorithmen (beim DES ist N = 64).

Diese Hashfunktionen sind besonders dann geeignet, wenn ein solcher Algorithmus auch aus anderen Gründen bereits implementiert ist. Außerdem ist es sehr schwierig, die Güte und Kollisionsfreiheit von Hashfunktionen nachzuweisen. Wird aber ein Verschlüsselungsalgorithmus verwendet, dessen Güte bekannt ist, spricht das auch für die Güte der Hashfunktion, da Hashfunktionen eine ähnliche Aufgabe wie die Verschlüsselung haben, nämlich einem Inputwert einen quasi zufälligen Outputwert zuzuordnen.

In der ISO 10118-2 werden zwei Typen von Hashfunktionen spezifiziert:

1. Erzeugung eines Hashcodes, dessen Länge kleiner oder gleich N ist.
2. Erzeugung eines Hashcodes, dessen Länge kleiner oder gleich 2 * N ist.

Hashcodes einfacher Länge

Es sei E der Verschlüsselungsalgorithmus, IV der Initialisierungsvektor. Der Hashcode H wird in drei Schritten berechnet:

1. die Nachricht D wird in die Blöcke D_1, D_2, ...D_k geteilt, wobei jeder Block aus genau N Bit besteht.
   Evtl. muß der letzte Block aufgefüllt werden (siehe Kap. 5.4.).

2. Es sei H_0 der Initialisierungsvektor IV.
   Die Output-Blöcke H_1, H_2,...H_k werden wie folgt berechnet, i = 1,...k:
   $K_i = u(H_{i-1})$
   $H_i = E_{K_i}(D_i)$ XOR D_i
   (E_{K_i}: Verschlüsselung mit dem Schlüssel K_i)

3. Als Hashcode H werden soviele höchstwertige Bits wie benötigt des letzten Output-Blockes H_k genommen.

Die Funktion u, die den Schlüssel K_i aus dem Vorgänger-Outputblock H_{i-1} ableitet, wird nicht an dieser Stelle definiert, sondern sollte Bestandteil der Beschreibung des Algorithmus' im ISO-Algorithmen-Register sein (siehe Kap. 5.2.).

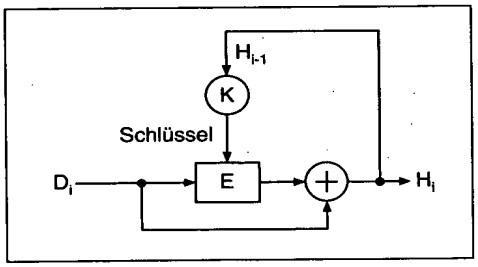


Bild 5.18 Hashcodes einfacher Länge

Wird der DES-Algorithmus als N-Bit-Blockalgorithmus verwendet, kann u folgendermaßen definiert werden - es dürfen keine schwachen oder halbschwachen Schlüssel entstehen -:

es sei $X = x_1x_2....x_{64}$ die binäre Darstellung des 64-Bit-Blockes X. Dann ist Y = u(X) der Block, der entsteht, indem die Bit x_8, x_{16}, x_{24},....x_{64} entfernt werden – die Paritätsbits, wenn X als DES-Schlüssel verwendet wird –, und x_2x_3 auf '10' gesetzt wird, d. h. $Y = x_1 10 x_4x_5x_6x_7x_9X_{10}...x_{62}X_{63}$.

Hashcodes doppelter Länge

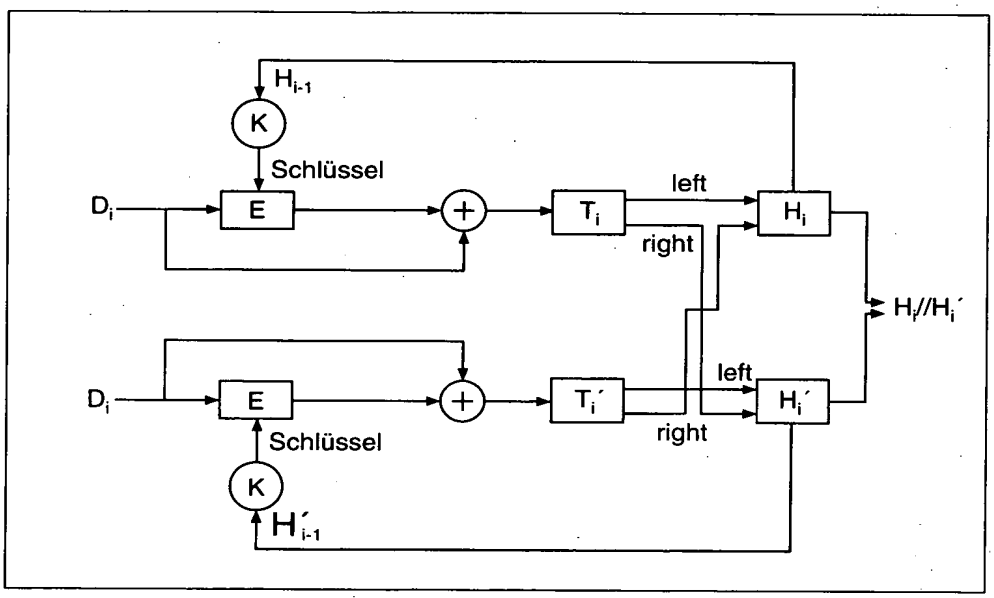
Es sei E wiederum der Verschlüsselungsalgorithmus, IV und IV' seien zwei verschiedene Initialisierungsvektoren. Der Hashcode wird ebenfalls in 3 Schritten berechnet:

1. Die Nachricht D wird in die Blöcke $D_1, D_2,..D_k$ jeweils der Länge N Bit geteilt. Evt. muß dabei der letzte Block temporär (für die Berechnung) aufgefüllt werden (siehe Kap.5.4.).

2. Es seien H_0 und H'_0 die Initialisierungsvekoren IV und IV'.
   Die Output-Blöcke $H_1, H_2,..H_k$, bzw. $H'_1, H'_2,..H'_k$ werden wie folgt berechnet, i = 1,...k
   $K_i = u(H_{i-1})$
   $K'_i = u'(H'_{i-1})$
   $T_i = e_{K_i}(D_i)$ XOR D_i
   $T'_i = e_{K'_i}(Di)$ XOR D_i
   $H_i = T_{i\ left}//T'_{i\ right}$
   $H'_i = T'_{i\ left}//T_{i\ right}$
   "//" bedeutet Konkatenation, also Aneinananderkettung der beiden Bitstrings, dabei sind X_{left} die N/2 linken Bits des N-Bit-Blockes X, falls N gerade ist, und die (N+1)/2 linken Bits, falls N ungerade ist, und X_{right} sind die N/2 rechten Bits des N-Bit-Blockes X, falls N gerade ist, und die (N-1)/2 rechten Bits, falls N ungerade ist.

3. Als Hashcode H werden soviele höchstwertige Bits des letzten Outputblockes $H_k//H'_k$ wie benötigt genommen.

Bild 5.19 Hashcodes doppelter Länge

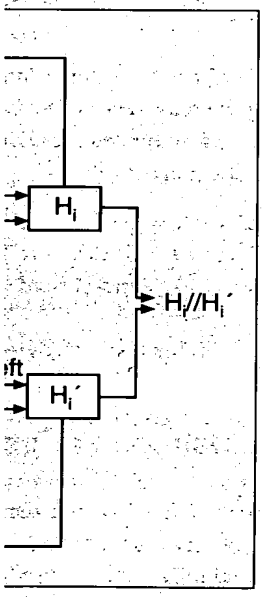


D_2,..D_k jeweils der Länge
Block temporär (für die
ap.5.4.).

vekoren IV und IV'.
H'_1, H'_2,..H'_k werden wie

nananderkettung der bei-
'2 linken Bits des N-Bit-
[+1)/2 linken Bits, falls N
rechten Bits des N-Bit-
-1)/2 rechten Bits, falls N

twertige Bits des letzten
ommen.



Zuletzt müssen u und u' definiert werden.

Falls man als N-Bit-Blockalgorithmus den DES-Algorithmus verwendet, ist eine mögliche Definition für u' (Beispiel für u s.o.):
es sei $X = x_1x_2....x_{64}$ die binäre Darstellung des 64-Bit-Blockes X.
Dann ist $Y = u'(X)$ der Block, der entsteht, indem die Bits x_8, x_{16}, x_{24},....x_{64} entfernt werden, und x_2x_3 auf '01' gesetzt wird, d.h.
$Y = x_101x_4x_5x_6x_7x_9x_{10}....x_{62}x_{63}$.

5.10. Asymmetrische kryptographische Verfahren

Es wurden asymmetrische kryptographische Verfahren entwickelt [DIFF76], bei denen die beiden Schlüssel, die ein Paar von kryptographischen Operationen steuern, z.B. die Ver- und Entschlüsselung oder Message Authentication Codes berechnen und verifizieren, verschieden sind. Daher ist einem Teilnehmer nicht ein Schlüssel zugeordnet, sondern ein Schlüsselsystem, das aus zwei Schlüsseln, z.B. p und s besteht: mit dem einen Schlüssel wird die Abbildung vom Klartext in eine transformierte Form gesteuert, der andere Schlüssel steuert die inverse Operation und überführt den transformierten Text wieder in Klartext. Diese Verfahren heißen asymmetrisch, weil beide Seiten, die an einer kryptographischen Operation beteiligt sind, verschiedene Schlüssel (eines Schlüsselsystems) verwenden. Zur Erinnerung: bei symmetrischen Verfahren benutzen beide Seiten denselben Schlüssel, z.B. zur Ver- und Entschlüsselung.

Einer von den beiden Schlüsseln, z.B. p kann öffentlich bekanntgegeben werden, wenn folgende Voraussetzungen erfüllt werden [POHL85]:

1. Es ist nicht möglich, aus dem Schlüssel p den zur inversen Operation notwendigen Schlüssel s abzuleiten.
2. Selbst wenn jemand ausgewählten Klartext mit dem Schlüssel p transformiert, ist es nicht möglich, daraus den Schlüssel s abzuleiten.

Daher heißen diese Verfahren auch Public Key-Verfahren. Jeder Teilnehmer erhält ein persönliches Schlüsselsystem. Der Schlüssel des Schlüsselsystems, der öffentlich bekanntgegeben werden kann, heißt auch der öffentliche Schlüssel (p = public key), der andere ist der geheime Schlüssel (s = secret key).

Vom Grundsatz her ist der geheime Schlüssel immer aus dem öffentlichen Schlüssel ableitbar. Damit dies aber möglichst schwierig ist, werden Algorithmen gewählt, die auf der Lösung von Problemen der Komplexitätstheorie beruhen. Man bezeichnet derartige Funktionen auch als one-way trapdoor Funktionen [POHL90].

Mit der oben zitierten Arbeit von Diffie und Hellman [DIFF76] ist die Grundlage der modernen Kryptographie gelegt worden. Sie nimmt eine Schlüsselrolle in der gesamten kryptographischen Literatur ein.

One-way Funktionen (Einwegfunktionen)

Dieser Typ von Funktionen hat die Eigenschaft, daß zu einem gegebenen Inputwert der Funktionswert "leicht" zu berechnen ist. Es ist mit vertretbarem Aufwand nicht möglich, zu einem gegebenen Funktionswert ein passendes Argument zu finden.

Die Hashfunktionen aus Kap. 5.9. sind spezielle One-Way-Funktionen.

Gibt es aber zu einer one-way Funktion einen Parameter, z.B. einen geheimen Schlüssel, mit dem die inverse Transformation "leicht" zu berechnen ist, handelt es sich um eine **One-way trapdoor Funktion.**

Ausgehend von diesen Überlegungen, die Diffie und Hellman im Jahre 1976 veröffentlichten [DIFF76], wurden verschiedene Vorschläge für Public-Key-Verfahren gemacht.

Vertreter dieser Klasse von Verfahren sind z.B. das Pohlig-Hellman-Verfahren [POHL77], das Merkle-Hellman-Verfahren [MERK78] und das RSA-Verfahren [RIVE78].

5.10.1. Verschlüsselung mit asymmetrischen Verfahren

Es gibt asymmetrische Verfahren, die sich zur Verschlüsselung von Nachrichten eignen. Dabei wird der öffentliche Schlüssel zur Verschlüsselung und der geheime Schlüssel zur Entschlüsselung verwendet.

Teilnehmer A will eine Nachricht an Teilnehmer B senden:

1. Er entnimmt den Verschlüsselungs-Schlüssel von B einem öffentlichen Verzeichnis, dem Briefkopf von B, einem Telefonbucheintrag, o.ä. (Anmerkung: es wird später gezeigt, wie A den öffentlichen Schlüssel in der Praxis erhält, wenn in Kap.6 die Zertifikate eingeführt wurden).
2. A verschlüsselt die Nachricht mit dem öffentlichen Schlüssel von B und sendet sie an Teilnehmer B.
3. Da nur Teilnehmer B den zugehörigen geheimen Entschlüsselungs-Schlüssel kennt (die "Trapdoor"), und da dieser Schlüssel weder aus dem öffentlichen Schlüssel noch aus der verschlüsselten Nachricht bestimmt werden kann, ist der Teilnehmer B tatsächlich der einzige, der in der Lage ist, die Nachricht wieder zu entschlüsseln.

Eine vertrauliche Kommunikation ist somit möglich, ohne daß dazu vorher eine geheime Schlüsselübermittlung zwischen A und B oder von dritter Seite an beide notwendig wäre.

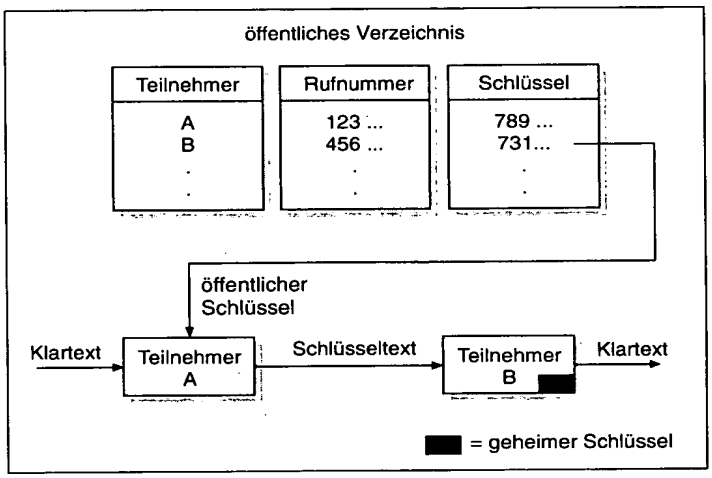


Bild 5.20 Asymmetrische Verschlüsselung

Manchmal wird in der Praxis der Fehler gemacht, öffentliche Schlüssel ungeschützt über das Netz auszutauschen, da es sich ja schließlich um öffentliche Schlüssel handelt, die jeder kennen darf. Werden die Nachrichten jeweils mit dem öffentlichen Schlüssel des Empfängers verschlüsselt, könnte angenommen werden, daß eine vertrauliche Kommunikation stattfindet.

Hat sich ein Angreifer aber bereits während des Schlüsselaustausches aktiv in den Übertragungskanal geschaltet und beiden Partnern sei-

nen eigenen öffentlichen Schlüssel mitgeteilt und nicht die Schlüssel der Partner durchgereicht, so kann er alle Daten entschlüsseln und umschlüsseln (sog. Back-to-Back-Angriff).

Daher ist Vorsicht beim Austausch von öffentlichen Schlüsseln geboten, wenn nicht gleichzeitig der Ursprung der Daten bzw. des Schlüssels nachgewiesen wird (siehe Kap. 11, Schlüsselverteilung und -vereinbarung)!

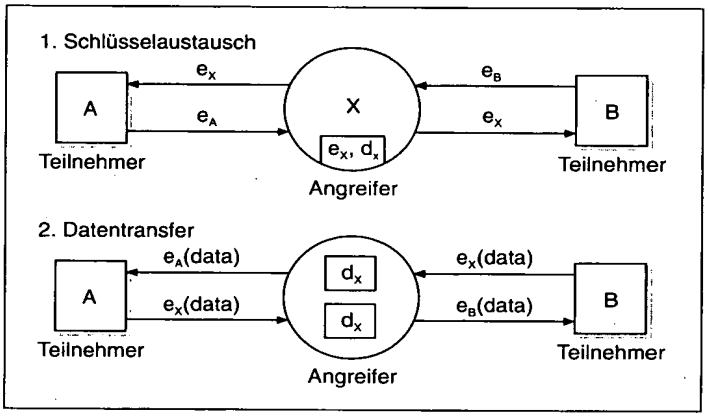


Bild 5.21
Angriff auf Austausch der öffentlichen Schlüssel

5.10.2. Elektronische Unterschrift

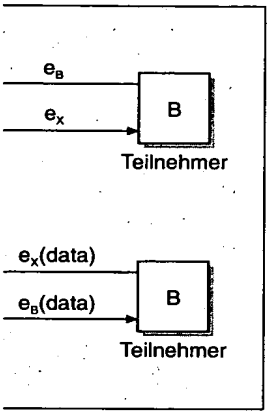
Einige asymmetrische Verfahren haben eine ganz besondere Anwendungsmöglichkeit: erfolgt zuerst die Transformation der Klartext-Nachricht mit dem geheimen Schlüssel, und die inverse Operation wird mit dem öffentlichen Schlüssel durchgeführt, ist der eindeutige Nachweis möglich, von wem eine Nachricht stammt.

Will der Urheber einer Nachricht sicherstellen, daß eine Nachricht nur ihm zugeordnet werden kann, so "verschlüsselt" er sie mit seinem geheimen Schlüssel. Mit dem Wort "verschlüsselt" ist hier nur "transformiert" gemeint, da gerade keine Verschlüsselung stattfindet, die diese Nachricht vor anderen Teilnehmern schützt, sondern sie kann von jedem wieder in Klartext zurücktransformiert werden.

Wegen der Äquivalenz zur Unterschrift spricht man hier von elektronischer Unterschrift. Die elektronische Unterschrift wurde früher in Deutschland und wird heute auch noch im englischen Sprachraum als

ilt und nicht die Schlüssel
: Daten entschlüsseln und

ntlichen Schlüsseln gebo-
ler Daten bzw. des Schlüs-
lüsselverteilung und -ver-



erschrift

e ganz besondere Anwen-
nsformation der Klartext-
nd die inverse Operation
geführt, ist der eindeutige
t stammt.

ellen, daß eine Nachricht
lüsselt" er sie mit seinem
üsselt" ist hier nur "trans-
lüsselung stattfindet, die
schützt, sondern sie kann
rmiert werden.

cht man hier von elektro-
erschrift wurde früher in
nglischen Sprachraum als

digitale Unterschrift bezeichnet (digital signature). Da diese jedoch von vielen mit einer digitalisierten Unterschrift verwechselt wurde, geht zur Zeit im deutschen Sprachraum der Trend zur Bezeichnung "elektronische Unterschrift".

Es gibt zwei Möglichkeiten der elektronischen Unterschrift: entweder wird die Nachricht selber transformiert, d. h. unterschrieben, oder es wird ein Authentikator gebildet, mit dessen Hilfe die Unversehrtheit und Urheberschaft der Nachricht überprüft werden kann. Ein solcher Authentikator kann ein Hashwert der Nachricht sein, der elektronisch unterschrieben wird, oder er wird durch das verwendete Verfahren direkt gebildet. In der internationalen Standardisierung wird daher zwischen "Digital Signature Schemes giving Message Recovery", bei der die Verifikation der elektronischen Unterschrift die Nachricht wieder liefert, und "Digital Signature Schemes with Appendix", die einen Authentikator liefern, unterschieden. In der Praxis ist die Nachricht, die mit dem Verfahren des "Digital Signature Scheme giving Message Recovery" elektronisch unterschrieben wird, aber auch nicht die Benutzernachricht selber, sondern ein Hashwert, der über diese Nachricht berechnet wurde, d.h. auch in diesem Fall wird ein Authentikator gebildet. Nur bei sehr kurzen Benutzernachrichten kann das Verfahren des "Digital Signature Scheme giving Message Recovery" direkt auf die Nachricht angewendet werden. Dieses Verfahren wird in Kap. 5.15. näher erläutert.

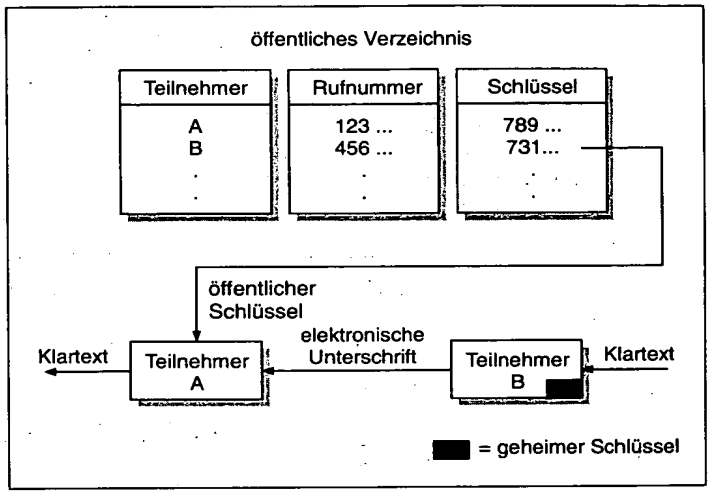


Bild 5.22 Elektronische Unterschrift

Der Wert einer elektronischen Unterschrift geht über den einer handschriftlichen Unterschrift hinaus, da sie sogar den Text miteinschließt: an Hand der Nachricht kann festgestellt werden, von wem die Nachricht stammt, und daß der Inhalt seit der Unterschrift nicht geändert wurde.

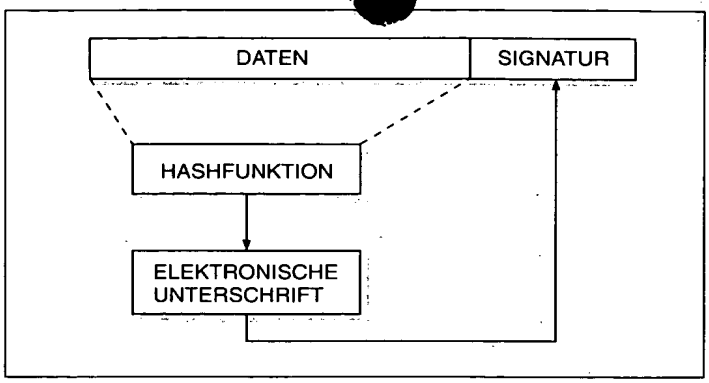


Bild 5.23
Elektronische Unterschrift unter Daten

Soll die elektronische Unterschrift mit Vertraulichkeit kombiniert werden, so ist eine zusätzliche Verschlüsselung mit dem öffentlichen Schlüssel des Empfängers notwendig. Auf diese Weise lassen sich elektronisch unterschriebene Nachrichten vertraulich über ein unsicheres Netz übertragen.

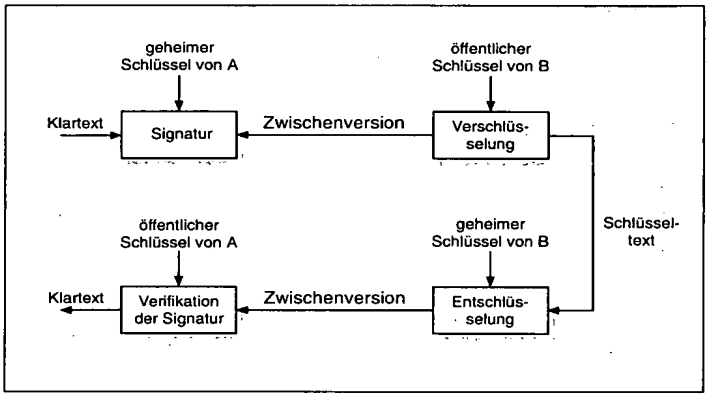
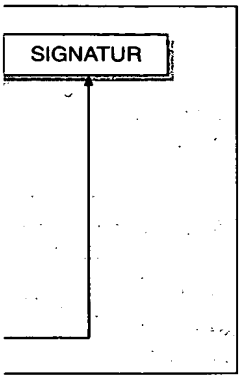


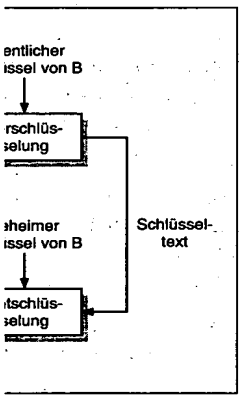
Bild 5.24
Elektronische Unterschrift kombiniert mit Vertraulichkeit

Kommutative asymmetrische Verfahren

Das (im nächsten Kapitel beschriebene) RSA-Verfahren eignet sich zur Verschlüsselung und elektronischen Unterschrift, da es kommutativ ist: wird die Reihenfolge "Zuerst Operation mit dem öffentlichen Schlüssel - entspricht Verschlüsselung -, dann Operation mit dem geheimen Schlüssel - entspricht der Entschlüsselung -" vertauscht zu der Reihenfolge "zuerst Operation mit dem geheimen Schlüssel - entspricht der elektronischen Unterschrift -, dann Operation mit dem



raulichkeit kombiniert
g mit dem öffentlichen
liese Weise lassen sich
traulich über ein unsi-



-Verfahren eignet sich
schrift, da es kommuta-
n mit dem öffentlichen
an Operation mit dem
selung -" vertauscht zu
heimen Schlüssel - ent-
an Operation mit dem

öffentlichen Schlüssel - entspricht der Verifikation -", so ergibt sich in jedem Fall wieder die Klartext-Nachricht, auf die die erste Operation angewendet worden ist.

Es gibt also asymmetrische Verfahren, die sich zur Verschlüsselung und zur elektronischen Unterschrift eignen. Daneben existieren auch asymmetrische Verfahren, die sich nur für elektronische Unterschriften eignen, oder mit denen nur eine vertrauliche Kommunikation oder Schlüsselabsprache möglich ist.

5.11. Das RSA-Verfahren

Das bekannteste, bewährteste und am besten untersuchte asymmetrische Verfahren, das heute den Rang eines internationalen Quasi-Standards einnimmt, wurde nach seinen Erfindern Ronald Rivest, Adi Shamir und Leonard Adleman benannt [RIVE78], das RSA-Verfahren. So wie der DES-Algorithmus **der** symmetrische Algorithmus ist, so ist der RSA-Algorithmus **der** asymmetrische Algorithmus.

Seine Sicherheit basiert auf der Schwierigkeit, eine große natürliche Zahl zu faktorisieren, d.h. alle Primzahlen zu finden, durch die sich eine Zahl ohne Rest teilen läßt.

Die mathematische Transformation E, um den Klartext M in den Schlüsseltext C zu überführen, lautet:

$$C = E(M) = M^e \bmod n$$

Die inverse Transformation D, um aus dem Schlüsseltext C den Klartext M wiederzuerhalten, lautet:

$$M = D(C) = C^d \bmod n$$

Die Nachricht M wird durch eine positive Integerzahl repräsentiert, die zwischen 0 und n-1 liegen muß. Wegen der Modulo-Rechnung liegt C ebenfalls zwischen 0 und n-1.

Nachrichten, die in ihrer numerischen Darstellung größer als n-1 sind, müssen in Blöcke aufgeteilt werden. Der RSA-Algorithmus ist also auch ein Blockalgorithmus.

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☑ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

Network Working Group
Request for Comments: 1889
Category: Standards Track

Audio-Video Transport Working Group
H. Schulzrinne
GMD Fokus
S. Casner
Precept Software, Inc.
R. Frederick
Xerox Palo Alto Research Center
V. Jacobson
Lawrence Berkeley National Laboratory
January 1996

RTP: A Transport Protocol for Real-Time Applications

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Abstract

This memorandum describes RTP, the real-time transport protocol. RTP provides end-to-end network transport functions suitable for applications transmitting real-time data, such as audio, video or simulation data, over multicast or unicast network services. RTP does not address resource reservation and does not guarantee quality-of-service for real-time services. The data transport is augmented by a control protocol (RTCP) to allow monitoring of the data delivery in a manner scalable to large multicast networks, and to provide minimal control and identification functionality. RTP and RTCP are designed to be independent of the underlying transport and network layers. The protocol supports the use of RTP-level translators and mixers.

Table of Contents

1. Introduction 3
2. RTP Use Scenarios 5
2.1 Simple Multicast Audio Conference 5
2.2 Audio and Video Conference 6
2.3 Mixers and Translators 6
3. Definitions 7
4. Byte Order, Alignment, and Time Format 9
5. RTP Data Transfer Protocol 10
5.1 RTP Fixed Header Fields 10
5.2 Multiplexing RTP Sessions 13

5.3 Profile-Specific Modifications to the RTP Header 14
5.3.1 RTP Header Extension 14
6. RTP Control Protocol -- RTCP 15
6.1 RTCP Packet Format 17
6.2 RTCP Transmission Interval 19
6.2.1 Maintaining the number of session members 21
6.2.2 Allocation of source description bandwidth 21
6.3 Sender and Receiver Reports 22
6.3.1 SR: Sender report RTCP packet 23
6.3.2 RR: Receiver report RTCP packet 28
6.3.3 Extending the sender and receiver reports 29
6.3.4 Analyzing sender and receiver reports 29
6.4 SDES: Source description RTCP packet 31
6.4.1 CNAME: Canonical end-point identifier SDES item 32
6.4.2 NAME: User name SDES item 34
6.4.3 EMAIL: Electronic mail address SDES item 34
6.4.4 PHONE: Phone number SDES item 34
6.4.5 LOC: Geographic user location SDES item 35
6.4.6 TOOL: Application or tool name SDES item 35
6.4.7 NOTE: Notice/status SDES item 35
6.4.8 PRIV: Private extensions SDES item 36
6.5 BYE: Goodbye RTCP packet 37
6.6 APP: Application-defined RTCP packet 38
7. RTP Translators and Mixers 39
7.1 General Description 39
7.2 RTCP Processing in Translators 41
7.3 RTCP Processing in Mixers 43
7.4 Cascaded Mixers 44
8. SSRC Identifier Allocation and Use 44
8.1 Probability of Collision 44
8.2 Collision Resolution and Loop Detection 45
9. Security 49
9.1 Confidentiality 49
9.2 Authentication and Message Integrity 50
10. RTP over Network and Transport Protocols 51
11. Summary of Protocol Constants 51
11.1 RTCP packet types 52
11.2 SDES types 52
12. RTP Profiles and Payload Format Specifications 53
A. Algorithms 56
A.1 RTP Data Header Validity Checks 59
A.2 RTCP Header Validity Checks 63
A.3 Determining the Number of RTP Packets Expected and Lost 63
A.4 Generating SDES RTCP Packets 64
A.5 Parsing RTCP SDES Packets 65
A.6 Generating a Random 32-bit Identifier 66
A.7 Computing the RTCP Transmission Interval 68

Meta 1002
p. 109

A.8 Estimating the Interarrival Jitter 71
B. Security Considerations 72
C. Addresses of Authors 72
D. Bibliography 73

1. Introduction

This memorandum specifies the real-time transport protocol (RTP), which provides end-to-end delivery services for data with real-time characteristics, such as interactive audio and video. Those services include payload type identification, sequence numbering, timestamping and delivery monitoring. Applications typically run RTP on top of UDP to make use of its multiplexing and checksum services; both protocols contribute parts of the transport protocol functionality. However, RTP may be used with other suitable underlying network or transport protocols (see Section 10). RTP supports data transfer to multiple destinations using multicast distribution if provided by the underlying network.

Note that RTP itself does not provide any mechanism to ensure timely delivery or provide other quality-of-service guarantees, but relies on lower-layer services to do so. It does not guarantee delivery or prevent out-of-order delivery, nor does it assume that the underlying network is reliable and delivers packets in sequence. The sequence numbers included in RTP allow the receiver to reconstruct the sender's packet sequence, but sequence numbers might also be used to determine the proper location of a packet, for example in video decoding, without necessarily decoding packets in sequence.

While RTP is primarily designed to satisfy the needs of multi-participant multimedia conferences, it is not limited to that particular application. Storage of continuous data, interactive distributed simulation, active badge, and control and measurement applications may also find RTP applicable.

This document defines RTP, consisting of two closely-linked parts:

- o the real-time transport protocol (RTP), to carry data that has real-time properties.

- o the RTP control protocol (RTCP), to monitor the quality of service and to convey information about the participants in an on-going session. The latter aspect of RTCP may be sufficient for "loosely controlled" sessions, i.e., where there is no explicit membership control and set-up, but it is not necessarily intended to support all of an application's control communication requirements. This functionality may be fully or partially subsumed by a separate session control protocol,

which is beyond the scope of this document.

RTP represents a new style of protocol following the principles of application level framing and integrated layer processing proposed by Clark and Tennenhouse [1]. That is, RTP is intended to be malleable to provide the information required by a particular application and will often be integrated into the application processing rather than being implemented as a separate layer. RTP is a protocol framework that is deliberately not complete. This document specifies those functions expected to be common across all the applications for which RTP would be appropriate. Unlike conventional protocols in which additional functions might be accommodated by making the protocol more general or by adding an option mechanism that would require parsing, RTP is intended to be tailored through modifications and/or additions to the headers as needed. Examples are given in Sections 5.3 and 6.3.3.

Therefore, in addition to this document, a complete specification of RTP for a particular application will require one or more companion documents (see Section 12):

- o a profile specification document, which defines a set of payload type codes and their mapping to payload formats (e.g., media encodings). A profile may also define extensions or modifications to RTP that are specific to a particular class of applications. Typically an application will operate under only one profile. A profile for audio and video data may be found in the companion RFC TBD.

- o payload format specification documents, which define how a particular payload, such as an audio or video encoding, is to be carried in RTP.

A discussion of real-time services and algorithms for their implementation as well as background discussion on some of the RTP design decisions can be found in [2].

Several RTP applications, both experimental and commercial, have already been implemented from draft specifications. These applications include audio and video tools along with diagnostic tools such as traffic monitors. Users of these tools number in the thousands. However, the current Internet cannot yet support the full potential demand for real-time services. High-bandwidth services using RTP, such as video, can potentially seriously degrade the quality of service of other network services. Thus, implementors should take appropriate precautions to limit accidental bandwidth usage. Application documentation should clearly outline the limitations and possible operational impact of high-bandwidth real-

Meta 1002
p. 111

time services on the Internet and other network services.

2. RTP Use Scenarios

The following sections describe some aspects of the use of RTP. The examples were chosen to illustrate the basic operation of applications using RTP, not to limit what RTP may be used for. In these examples, RTP is carried on top of IP and UDP, and follows the conventions established by the profile for audio and video specified in the companion Internet-Draft draft-ietf-avt-profile

2.1 Simple Multicast Audio Conference

A working group of the IETF meets to discuss the latest protocol draft, using the IP multicast services of the Internet for voice communications. Through some allocation mechanism the working group chair obtains a multicast group address and pair of ports. One port is used for audio data, and the other is used for control (RTCP) packets. This address and port information is distributed to the intended participants. If privacy is desired, the data and control packets may be encrypted as specified in Section 9.1, in which case an encryption key must also be generated and distributed. The exact details of these allocation and distribution mechanisms are beyond the scope of RTP.

The audio conferencing application used by each conference participant sends audio data in small chunks of, say, 20 ms duration. Each chunk of audio data is preceded by an RTP header; RTP header and data are in turn contained in a UDP packet. The RTP header indicates what type of audio encoding (such as PCM, ADPCM or LPC) is contained in each packet so that senders can change the encoding during a conference, for example, to accommodate a new participant that is connected through a low-bandwidth link or react to indications of network congestion.

The Internet, like other packet networks, occasionally loses and reorders packets and delays them by variable amounts of time. To cope with these impairments, the RTP header contains timing information and a sequence number that allow the receivers to reconstruct the timing produced by the source, so that in this example, chunks of audio are contiguously played out the speaker every 20 ms. This timing reconstruction is performed separately for each source of RTP packets in the conference. The sequence number can also be used by the receiver to estimate how many packets are being lost.

Since members of the working group join and leave during the conference, it is useful to know who is participating at any moment and how well they are receiving the audio data. For that purpose,

each instance of the audio application in the conference periodically multicasts a reception report plus the name of its user on the RTCP (control) port. The reception report indicates how well the current speaker is being received and may be used to control adaptive encodings. In addition to the user name, other identifying information may also be included subject to control bandwidth limits. A site sends the RTCP BYE packet (Section 6.5) when it leaves the conference.

2.2 Audio and Video Conference

If both audio and video media are used in a conference, they are transmitted as separate RTP sessions RTCP packets are transmitted for each medium using two different UDP port pairs and/or multicast addresses. There is no direct coupling at the RTP level between the audio and video sessions, except that a user participating in both sessions should use the same distinguished (canonical) name in the RTCP packets for both so that the sessions can be associated.

One motivation for this separation is to allow some participants in the conference to receive only one medium if they choose. Further explanation is given in Section 5.2. Despite the separation, synchronized playback of a source's audio and video can be achieved using timing information carried in the RTCP packets for both sessions.

2.3 Mixers and Translators

So far, we have assumed that all sites want to receive media data in the same format. However, this may not always be appropriate. Consider the case where participants in one area are connected through a low-speed link to the majority of the conference participants who enjoy high-speed network access. Instead of forcing everyone to use a lower-bandwidth, reduced-quality audio encoding, an RTP-level relay called a mixer may be placed near the low-bandwidth area. This mixer resynchronizes incoming audio packets to reconstruct the constant 20 ms spacing generated by the sender, mixes these reconstructed audio streams into a single stream, translates the audio encoding to a lower-bandwidth one and forwards the lower-bandwidth packet stream across the low-speed link. These packets might be unicast to a single recipient or multicast on a different address to multiple recipients. The RTP header includes a means for mixers to identify the sources that contributed to a mixed packet so that correct talker indication can be provided at the receivers.

Some of the intended participants in the audio conference may be connected with high bandwidth links but might not be directly reachable via IP multicast. For example, they might be behind an

Meta 1002
p. 113

application-level firewall that will not let any IP packets pass. For these sites, mixing may not be necessary, in which case another type of RTP-level relay called a translator may be used. Two translators are installed, one on either side of the firewall, with the outside one funneling all multicast packets received through a secure connection to the translator inside the firewall. The translator inside the firewall sends them again as multicast packets to a multicast group restricted to the site's internal network.

Mixers and translators may be designed for a variety of purposes. An example is a video mixer that scales the images of individual people in separate video streams and composites them into one video stream to simulate a group scene. Other examples of translation include the connection of a group of hosts speaking only IP/UDP to a group of hosts that understand only ST-II, or the packet-by-packet encoding translation of video streams from individual sources without resynchronization or mixing. Details of the operation of mixers and translators are given in Section 7.

3. Definitions

RTP payload: The data transported by RTP in a packet, for example audio samples or compressed video data. The payload format and interpretation are beyond the scope of this document.

RTP packet: A data packet consisting of the fixed RTP header, a possibly empty list of contributing sources (see below), and the payload data. Some underlying protocols may require an encapsulation of the RTP packet to be defined. Typically one packet of the underlying protocol contains a single RTP packet, but several RTP packets may be contained if permitted by the encapsulation method (see Section 10).

RTCP packet: A control packet consisting of a fixed header part similar to that of RTP data packets, followed by structured elements that vary depending upon the RTCP packet type. The formats are defined in Section 6. Typically, multiple RTCP packets are sent together as a compound RTCP packet in a single packet of the underlying protocol; this is enabled by the length field in the fixed header of each RTCP packet.

Port: The "abstraction that transport protocols use to distinguish among multiple destinations within a given host computer. TCP/IP protocols identify ports using small positive integers." [3] The transport selectors (TSEL) used by the OSI transport layer are equivalent to ports. RTP depends upon the lower-layer protocol to provide some mechanism such as ports to multiplex the RTP and RTCP packets of a session.

Meta 1002
p. 114

Transport address: The combination of a network address and port that identifies a transport-level endpoint, for example an IP address and a UDP port. Packets are transmitted from a source transport address to a destination transport address.

RTP session: The association among a set of participants communicating with RTP. For each participant, the session is defined by a particular pair of destination transport addresses (one network address plus a port pair for RTP and RTCP). The destination transport address pair may be common for all participants, as in the case of IP multicast, or may be different for each, as in the case of individual unicast network addresses plus a common port pair. In a multimedia session, each medium is carried in a separate RTP session with its own RTCP packets. The multiple RTP sessions are distinguished by different port number pairs and/or different multicast addresses.

Synchronization source (SSRC): The source of a stream of RTP packets, identified by a 32-bit numeric SSRC identifier carried in the RTP header so as not to be dependent upon the network address. All packets from a synchronization source form part of the same timing and sequence number space, so a receiver groups packets by synchronization source for playback. Examples of synchronization sources include the sender of a stream of packets derived from a signal source such as a microphone or a camera, or an RTP mixer (see below). A synchronization source may change its data format, e.g., audio encoding, over time. The SSRC identifier is a randomly chosen value meant to be globally unique within a particular RTP session (see Section 8). A participant need not use the same SSRC identifier for all the RTP sessions in a multimedia session; the binding of the SSRC identifiers is provided through RTCP (see Section 6.4.1). If a participant generates multiple streams in one RTP session, for example from separate video cameras, each must be identified as a different SSRC.

Contributing source (CSRC): A source of a stream of RTP packets that has contributed to the combined stream produced by an RTP mixer (see below). The mixer inserts a list of the SSRC identifiers of the sources that contributed to the generation of a particular packet into the RTP header of that packet. This list is called the CSRC list. An example application is audio conferencing where a mixer indicates all the talkers whose speech was combined to produce the outgoing packet, allowing the receiver to indicate the current talker, even though all the audio packets contain the same SSRC identifier (that of the mixer).

Meta 1002
p. 115

End system: An application that generates the content to be sent in RTP packets and/or consumes the content of received RTP packets. An end system can act as one or more synchronization sources in a particular RTP session, but typically only one.

Mixer: An intermediate system that receives RTP packets from one or more sources, possibly changes the data format, combines the packets in some manner and then forwards a new RTP packet. Since the timing among multiple input sources will not generally be synchronized, the mixer will make timing adjustments among the streams and generate its own timing for the combined stream. Thus, all data packets originating from a mixer will be identified as having the mixer as their synchronization source.

Translator: An intermediate system that forwards RTP packets with their synchronization source identifier intact. Examples of translators include devices that convert encodings without mixing, replicators from multicast to unicast, and application-level filters in firewalls.

Monitor: An application that receives RTCP packets sent by participants in an RTP session, in particular the reception reports, and estimates the current quality of service for distribution monitoring, fault diagnosis and long-term statistics. The monitor function is likely to be built into the application(s) participating in the session, but may also be a separate application that does not otherwise participate and does not send or receive the RTP data packets. These are called third party monitors.

Non-RTP means: Protocols and mechanisms that may be needed in addition to RTP to provide a usable service. In particular, for multimedia conferences, a conference control application may distribute multicast addresses and keys for encryption, negotiate the encryption algorithm to be used, and define dynamic mappings between RTP payload type values and the payload formats they represent for formats that do not have a predefined payload type value. For simple applications, electronic mail or a conference database may also be used. The specification of such protocols and mechanisms is outside the scope of this document.

4. Byte Order, Alignment, and Time Format

All integer fields are carried in network byte order, that is, most significant byte (octet) first. This byte order is commonly known as big-endian. The transmission order is described in detail in [4]. Unless otherwise noted, numeric constants are in decimal (base 10).

Meta 1002
p. 116

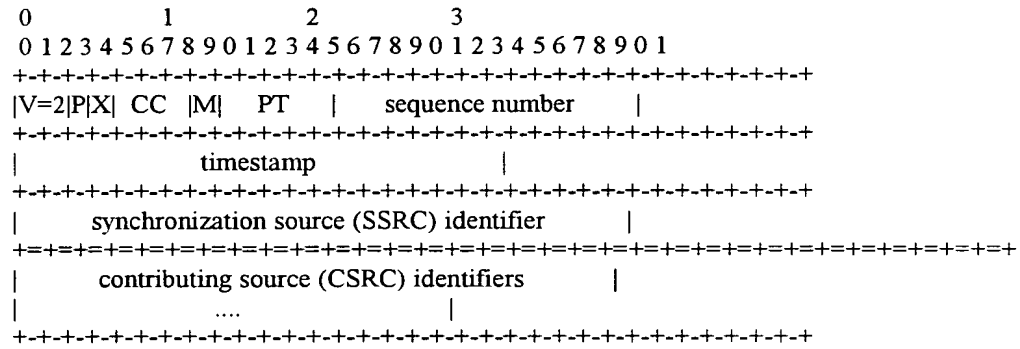
All header data is aligned to its natural length, i.e., 16-bit fields are aligned on even offsets, 32-bit fields are aligned at offsets divisible by four, etc. Octets designated as padding have the value zero.

Wallclock time (absolute time) is represented using the timestamp format of the Network Time Protocol (NTP), which is in seconds relative to 0h UTC on 1 January 1900 [5]. The full resolution NTP timestamp is a 64-bit unsigned fixed-point number with the integer part in the first 32 bits and the fractional part in the last 32 bits. In some fields where a more compact representation is appropriate, only the middle 32 bits are used; that is, the low 16 bits of the integer part and the high 16 bits of the fractional part. The high 16 bits of the integer part must be determined independently.

5. RTP Data Transfer Protocol

5.1 RTP Fixed Header Fields

The RTP header has the following format:



The first twelve octets are present in every RTP packet, while the list of CSRC identifiers is present only when inserted by a mixer. The fields have the following meaning:

version (V): 2 bits
This field identifies the version of RTP. The version defined by this specification is two (2). (The value 1 is used by the first draft version of RTP and the value 0 is used by the protocol initially implemented in the "vat" audio tool.)

padding (P): 1 bit
If the padding bit is set, the packet contains one or more additional padding octets at the end which are not part of the

Meta 1002
p. 117

payload. The last octet of the padding contains a count of how many padding octets should be ignored. Padding may be needed by some encryption algorithms with fixed block sizes or for carrying several RTP packets in a lower-layer protocol data unit.

extension (X): 1 bit
If the extension bit is set, the fixed header is followed by exactly one header extension, with a format defined in Section 5.3.1.

CSRC count (CC): 4 bits
The CSRC count contains the number of CSRC identifiers that follow the fixed header.

marker (M): 1 bit
The interpretation of the marker is defined by a profile. It is intended to allow significant events such as frame boundaries to be marked in the packet stream. A profile may define additional marker bits or specify that there is no marker bit by changing the number of bits in the payload type field (see Section 5.3).

payload type (PT): 7 bits
This field identifies the format of the RTP payload and determines its interpretation by the application. A profile specifies a default static mapping of payload type codes to payload formats. Additional payload type codes may be defined dynamically through non-RTP means (see Section 3). An initial set of default mappings for audio and video is specified in the companion profile Internet-Draft draft-ietf-avt-profile, and may be extended in future editions of the Assigned Numbers RFC [6]. An RTP sender emits a single RTP payload type at any given time; this field is not intended for multiplexing separate media streams (see Section 5.2).

sequence number: 16 bits
The sequence number increments by one for each RTP data packet sent, and may be used by the receiver to detect packet loss and to restore packet sequence. The initial value of the sequence number is random (unpredictable) to make known-plaintext attacks on encryption more difficult, even if the source itself does not encrypt, because the packets may flow through a translator that does. Techniques for choosing unpredictable numbers are discussed in [7].

timestamp: 32 bits
The timestamp reflects the sampling instant of the first octet in the RTP data packet. The sampling instant must be derived

Meta 1002
p. 118

from a clock that increments monotonically and linearly in time to allow synchronization and jitter calculations (see Section 6.3.1). The resolution of the clock must be sufficient for the desired synchronization accuracy and for measuring packet arrival jitter (one tick per video frame is typically not sufficient). The clock frequency is dependent on the format of data carried as payload and is specified statically in the profile or payload format specification that defines the format, or may be specified dynamically for payload formats defined through non-RTP means. If RTP packets are generated periodically, the nominal sampling instant as determined from the sampling clock is to be used, not a reading of the system clock. As an example, for fixed-rate audio the timestamp clock would likely increment by one for each sampling period. If an audio application reads blocks covering 160 sampling periods from the input device, the timestamp would be increased by 160 for each such block, regardless of whether the block is transmitted in a packet or dropped as silent.

The initial value of the timestamp is random, as for the sequence number. Several consecutive RTP packets may have equal timestamps if they are (logically) generated at once, e.g., belong to the same video frame. Consecutive RTP packets may contain timestamps that are not monotonic if the data is not transmitted in the order it was sampled, as in the case of MPEG interpolated video frames. (The sequence numbers of the packets as transmitted will still be monotonic.)

SSRC: 32 bits

The SSRC field identifies the synchronization source. This identifier is chosen randomly, with the intent that no two synchronization sources within the same RTP session will have the same SSRC identifier. An example algorithm for generating a random identifier is presented in Appendix A.6. Although the probability of multiple sources choosing the same identifier is low, all RTP implementations must be prepared to detect and resolve collisions. Section 8 describes the probability of collision along with a mechanism for resolving collisions and detecting RTP-level forwarding loops based on the uniqueness of the SSRC identifier. If a source changes its source transport address, it must also choose a new SSRC identifier to avoid being interpreted as a looped source.

CSRC list: 0 to 15 items, 32 bits each

The CSRC list identifies the contributing sources for the payload contained in this packet. The number of identifiers is given by the CC field. If there are more than 15 contributing sources, only 15 may be identified. CSRC identifiers are

Meta 1002
p. 119

inserted by mixers, using the SSRC identifiers of contributing sources. For example, for audio packets the SSRC identifiers of all sources that were mixed together to create a packet are listed, allowing correct talker indication at the receiver.

5.2 Multiplexing RTP Sessions

For efficient protocol processing, the number of multiplexing points should be minimized, as described in the integrated layer processing design principle [1]. In RTP, multiplexing is provided by the destination transport address (network address and port number) which define an RTP session. For example, in a teleconference composed of audio and video media encoded separately, each medium should be carried in a separate RTP session with its own destination transport address. It is not intended that the audio and video be carried in a single RTP session and demultiplexed based on the payload type or SSRC fields. Interleaving packets with different payload types but using the same SSRC would introduce several problems:

1. If one payload type were switched during a session, there would be no general means to identify which of the old values the new one replaced.

2. An SSRC is defined to identify a single timing and sequence number space. Interleaving multiple payload types would require different timing spaces if the media clock rates differ and would require different sequence number spaces to tell which payload type suffered packet loss.

3. The RTCP sender and receiver reports (see Section 6.3) can only describe one timing and sequence number space per SSRC and do not carry a payload type field.

4. An RTP mixer would not be able to combine interleaved streams of incompatible media into one stream.

5. Carrying multiple media in one RTP session precludes: the use of different network paths or network resource allocations if appropriate; reception of a subset of the media if desired, for example just audio if video would exceed the available bandwidth; and receiver implementations that use separate processes for the different media, whereas using separate RTP sessions permits either single- or multiple-process implementations.

Using a different SSRC for each medium but sending them in the same RTP session would avoid the first three problems but not the last two.

5.3 Profile-Specific Modifications to the RTP Header

The existing RTP data packet header is believed to be complete for the set of functions required in common across all the application classes that RTP might support. However, in keeping with the ALF design principle, the header may be tailored through modifications or additions defined in a profile specification while still allowing profile-independent monitoring and recording tools to function.

- The marker bit and payload type field carry profile-specific information, but they are allocated in the fixed header since many applications are expected to need them and might otherwise have to add another 32-bit word just to hold them. The octet containing these fields may be redefined by a profile to suit different requirements, for example with a more or fewer marker bits. If there are any marker bits, one should be located in the most significant bit of the octet since profile-independent monitors may be able to observe a correlation between packet loss patterns and the marker bit.

- Additional information that is required for a particular payload format, such as a video encoding, should be carried in the payload section of the packet. This might be in a header that is always present at the start of the payload section, or might be indicated by a reserved value in the data pattern.

- If a particular class of applications needs additional functionality independent of payload format, the profile under which those applications operate should define additional fixed fields to follow immediately after the SSRC field of the existing fixed header. Those applications will be able to quickly and directly access the additional fields while profile-independent monitors or recorders can still process the RTP packets by interpreting only the first twelve octets.

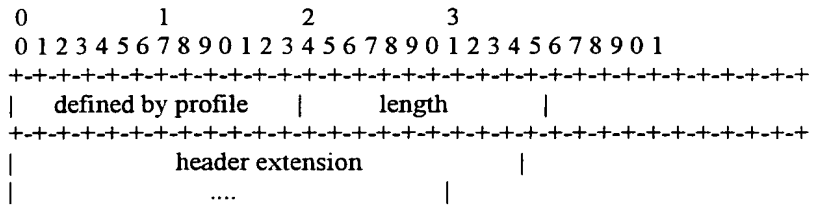
If it turns out that additional functionality is needed in common across all profiles, then a new version of RTP should be defined to make a permanent change to the fixed header.

5.3.1 RTP Header Extension

An extension mechanism is provided to allow individual implementations to experiment with new payload-format-independent functions that require additional information to be carried in the RTP data packet header. This mechanism is designed so that the header extension may be ignored by other interoperating implementations that have not been extended.

Meta 1002
p. 121

Note that this header extension is intended only for limited use. Most potential uses of this mechanism would be better done another way, using the methods described in the previous section. For example, a profile-specific extension to the fixed header is less expensive to process because it is not conditional nor in a variable location. Additional information required for a particular payload format should not use this header extension, but should be carried in the payload section of the packet.



If the X bit in the RTP header is one, a variable-length header extension is appended to the RTP header, following the CSRC list if present. The header extension contains a 16-bit length field that counts the number of 32-bit words in the extension, excluding the four-octet extension header (therefore zero is a valid length). Only a single extension may be appended to the RTP data header. To allow multiple interoperating implementations to each experiment independently with different header extensions, or to allow a particular implementation to experiment with more than one type of header extension, the first 16 bits of the header extension are left open for distinguishing identifiers or parameters. The format of these 16 bits is to be defined by the profile specification under which the implementations are operating. This RTP specification does not define any header extensions itself.

6. RTP Control Protocol -- RTCP

The RTP control protocol (RTCP) is based on the periodic transmission of control packets to all participants in the session, using the same distribution mechanism as the data packets. The underlying protocol must provide multiplexing of the data and control packets, for example using separate port numbers with UDP. RTCP performs four functions:

1. The primary function is to provide feedback on the quality of the data distribution. This is an integral part of the RTP's role as a transport protocol and is related to the flow and congestion control functions of other transport protocols. The feedback may be directly useful for control of adaptive encodings [8,9], but experiments with IP

Meta 1002
p. 122

multicasting have shown that it is also critical to get feedback from the receivers to diagnose faults in the distribution. Sending reception feedback reports to all participants allows one who is observing problems to evaluate whether those problems are local or global. With a distribution mechanism like IP multicast, it is also possible for an entity such as a network service provider who is not otherwise involved in the session to receive the feedback information and act as a third-party monitor to diagnose network problems. This feedback function is performed by the RTCP sender and receiver reports, described below in Section 6.3.

2. RTCP carries a persistent transport-level identifier for an RTP source called the canonical name or CNAME, Section 6.4.1. Since the SSRC identifier may change if a conflict is discovered or a program is restarted, receivers require the CNAME to keep track of each participant. Receivers also require the CNAME to associate multiple data streams from a given participant in a set of related RTP sessions, for example to synchronize audio and video.

3. The first two functions require that all participants send RTCP packets, therefore the rate must be controlled in order for RTP to scale up to a large number of participants. By having each participant send its control packets to all the others, each can independently observe the number of participants. This number is used to calculate the rate at which the packets are sent, as explained in Section 6.2.

4. A fourth, optional function is to convey minimal session control information, for example participant identification to be displayed in the user interface. This is most likely to be useful in "loosely controlled" sessions where participants enter and leave without membership control or parameter negotiation. RTCP serves as a convenient channel to reach all the participants, but it is not necessarily expected to support all the control communication requirements of an application. A higher-level session control protocol, which is beyond the scope of this document, may be needed.

Functions 1-3 are mandatory when RTP is used in the IP multicast environment, and are recommended for all environments. RTP application designers are advised to avoid mechanisms that can only work in unicast mode and will not scale to larger numbers.

Meta 1002
p. 123

6.1 RTCP Packet Format

This specification defines several RTCP packet types to carry a variety of control information:

SR: Sender report, for transmission and reception statistics from participants that are active senders

RR: Receiver report, for reception statistics from participants that are not active senders

SDES: Source description items, including CNAME

BYE: Indicates end of participation

APP: Application specific functions

Each RTCP packet begins with a fixed part similar to that of RTP data packets, followed by structured elements that may be of variable length according to the packet type but always end on a 32-bit boundary. The alignment requirement and a length field in the fixed part are included to make RTCP packets "stackable". Multiple RTCP packets may be concatenated without any intervening separators to form a compound RTCP packet that is sent in a single packet of the lower layer protocol, for example UDP. There is no explicit count of individual RTCP packets in the compound packet since the lower layer protocols are expected to provide an overall length to determine the end of the compound packet.

Each individual RTCP packet in the compound packet may be processed independently with no requirements upon the order or combination of packets. However, in order to perform the functions of the protocol, the following constraints are imposed:

- o Reception statistics (in SR or RR) should be sent as often as bandwidth constraints will allow to maximize the resolution of the statistics, therefore each periodically transmitted compound RTCP packet should include a report packet.

- o New receivers need to receive the CNAME for a source as soon as possible to identify the source and to begin associating media for purposes such as lip-sync, so each compound RTCP packet should also include the SDES CNAME.

- o The number of packet types that may appear first in the compound packet should be limited to increase the number of constant bits in the first word and the probability of successfully validating RTCP packets against misaddressed RTP

data packets or other unrelated packets.

Thus, all RTCP packets must be sent in a compound packet of at least two individual packets, with the following format recommended:

Encryption prefix: If and only if the compound packet is to be encrypted, it is prefixed by a random 32-bit quantity redrawn for every compound packet transmitted.

SR or RR: The first RTCP packet in the compound packet must always be a report packet to facilitate header validation as described in Appendix A.2. This is true even if no data has been sent nor received, in which case an empty RR is sent, and even if the only other RTCP packet in the compound packet is a BYE.

Additional RRs: If the number of sources for which reception statistics are being reported exceeds 31, the number that will fit into one SR or RR packet, then additional RR packets should follow the initial report packet.

SDES: An SDES packet containing a CNAME item must be included in each compound RTCP packet. Other source description items may optionally be included if required by a particular application, subject to bandwidth constraints (see Section 6.2.2).

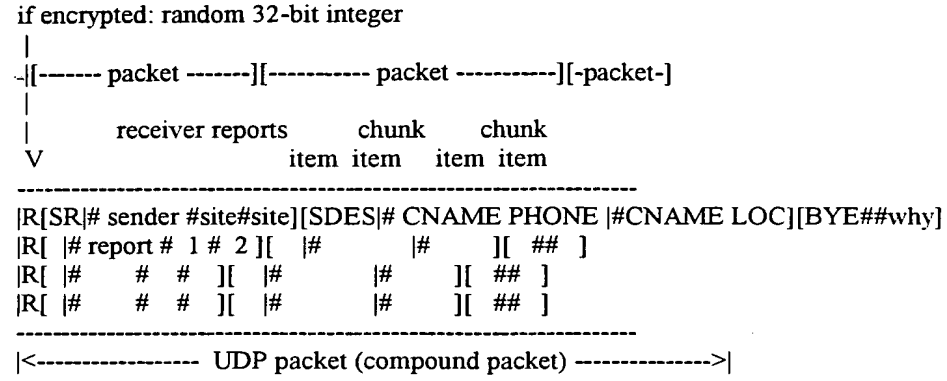
BYE or APP: Other RTCP packet types, including those yet to be defined, may follow in any order, except that BYE should be the last packet sent with a given SSRC/CSRC. Packet types may appear more than once.

It is advisable for translators and mixers to combine individual RTCP packets from the multiple sources they are forwarding into one compound packet whenever feasible in order to amortize the packet overhead (see Section 7). An example RTCP compound packet as might be produced by a mixer is shown in Fig. 1. If the overall length of a compound packet would exceed the maximum transmission unit (MTU) of the network path, it may be segmented into multiple shorter compound packets to be transmitted in separate packets of the underlying protocol. Note that each of the compound packets must begin with an SR or RR packet.

An implementation may ignore incoming RTCP packets with types unknown to it. Additional RTCP packet types may be registered with the Internet Assigned Numbers Authority (IANA).

Meta 1002
p. 125

6.2 RTCP Transmission Interval



#: SSRC/CSRC

Figure 1: Example of an RTCP compound packet

RTP is designed to allow an application to scale automatically over session sizes ranging from a few participants to thousands. For example, in an audio conference the data traffic is inherently self-limiting because only one or two people will speak at a time, so with multicast distribution the data rate on any given link remains relatively constant independent of the number of participants. However, the control traffic is not self-limiting. If the reception reports from each participant were sent at a constant rate, the control traffic would grow linearly with the number of participants. Therefore, the rate must be scaled down.

For each session, it is assumed that the data traffic is subject to an aggregate limit called the "session bandwidth" to be divided among the participants. This bandwidth might be reserved and the limit enforced by the network, or it might just be a reasonable share. The session bandwidth may be chosen based or some cost or a priori knowledge of the available network bandwidth for the session. It is somewhat independent of the media encoding, but the encoding choice may be limited by the session bandwidth. The session bandwidth parameter is expected to be supplied by a session management application when it invokes a media application, but media applications may also set a default based on the single-sender data bandwidth for the encoding selected for the session. The application may also enforce bandwidth limits based on multicast scope rules or other criteria.

Meta 1002
p. 126

Bandwidth calculations for control and data traffic include lower-layer transport and network protocols (e.g., UDP and IP) since that is what the resource reservation system would need to know. The application can also be expected to know which of these protocols are in use. Link level headers are not included in the calculation since the packet will be encapsulated with different link level headers as it travels.

The control traffic should be limited to a small and known fraction of the session bandwidth: small so that the primary function of the transport protocol to carry data is not impaired; known so that the control traffic can be included in the bandwidth specification given to a resource reservation protocol, and so that each participant can independently calculate its share. It is suggested that the fraction of the session bandwidth allocated to RTCP be fixed at 5%. While the value of this and other constants in the interval calculation is not critical, all participants in the session must use the same values so the same interval will be calculated. Therefore, these constants should be fixed for a particular profile.

The algorithm described in Appendix A.7 was designed to meet the goals outlined above. It calculates the interval between sending compound RTCP packets to divide the allowed control traffic bandwidth among the participants. This allows an application to provide fast response for small sessions where, for example, identification of all participants is important, yet automatically adapt to large sessions. The algorithm incorporates the following characteristics:

- o Senders are collectively allocated at least 1/4 of the control traffic bandwidth so that in sessions with a large number of receivers but a small number of senders, newly joining participants will more quickly receive the CNAME for the sending sites.

- o The calculated interval between RTCP packets is required to be greater than a minimum of 5 seconds to avoid having bursts of RTCP packets exceed the allowed bandwidth when the number of participants is small and the traffic isn't smoothed according to the law of large numbers.

- o The interval between RTCP packets is varied randomly over the range [0.5,1.5] times the calculated interval to avoid unintended synchronization of all participants [10]. The first RTCP packet sent after joining a session is also delayed by a random variation of half the minimum RTCP interval in case the application is started at multiple sites simultaneously, for example as initiated by a session announcement.

Meta 1002
p. 127

o A dynamic estimate of the average compound RTCP packet size is calculated, including all those received and sent, to automatically adapt to changes in the amount of control information carried.

This algorithm may be used for sessions in which all participants are allowed to send. In that case, the session bandwidth parameter is the product of the individual sender's bandwidth times the number of participants, and the RTCP bandwidth is 5% of that.

6.2.1 Maintaining the number of session members

Calculation of the RTCP packet interval depends upon an estimate of the number of sites participating in the session. New sites are added to the count when they are heard, and an entry for each is created in a table indexed by the SSRC or CSRC identifier (see Section 8.2) to keep track of them. New entries may not be considered valid until multiple packets carrying the new SSRC have been received (see Appendix A.1). Entries may be deleted from the table when an RTCP BYE packet with the corresponding SSRC identifier is received.

A participant may mark another site inactive, or delete it if not yet valid, if no RTP or RTCP packet has been received for a small number of RTCP report intervals (5 is suggested). This provides some robustness against packet loss. All sites must calculate roughly the same value for the RTCP report interval in order for this timeout to work properly.

Once a site has been validated, then if it is later marked inactive the state for that site should still be retained and the site should continue to be counted in the total number of sites sharing RTCP bandwidth for a period long enough to span typical network partitions. This is to avoid excessive traffic, when the partition heals, due to an RTCP report interval that is too small. A timeout of 30 minutes is suggested. Note that this is still larger than 5 times the largest value to which the RTCP report interval is expected to usefully scale, about 2 to 5 minutes.

6.2.2 Allocation of source description bandwidth

This specification defines several source description (SDES) items in addition to the mandatory CNAME item, such as NAME (personal name) and EMAIL (email address). It also provides a means to define new application-specific RTCP packet types. Applications should exercise caution in allocating control bandwidth to this additional information because it will slow down the rate at which reception reports and CNAME are sent, thus impairing the performance of the protocol. It is recommended that no more than 20% of the RTCP

Meta 1002
p. 128

bandwidth allocated to a single participant be used to carry the additional information. Furthermore, it is not intended that all SDES items should be included in every application. Those that are included should be assigned a fraction of the bandwidth according to their utility. Rather than estimate these fractions dynamically, it is recommended that the percentages be translated statically into report interval counts based on the typical length of an item.

For example, an application may be designed to send only CNAME, NAME and EMAIL and not any others. NAME might be given much higher priority than EMAIL because the NAME would be displayed continuously in the application's user interface, whereas EMAIL would be displayed only when requested. At every RTCP interval, an RR packet and an SDES packet with the CNAME item would be sent. For a small session operating at the minimum interval, that would be every 5 seconds on the average. Every third interval (15 seconds), one extra item would be included in the SDES packet. Seven out of eight times this would be the NAME item, and every eighth time (2 minutes) it would be the EMAIL item.

When multiple applications operate in concert using cross-application binding through a common CNAME for each participant, for example in a multimedia conference composed of an RTP session for each medium, the additional SDES information might be sent in only one RTP session. The other sessions would carry only the CNAME item.

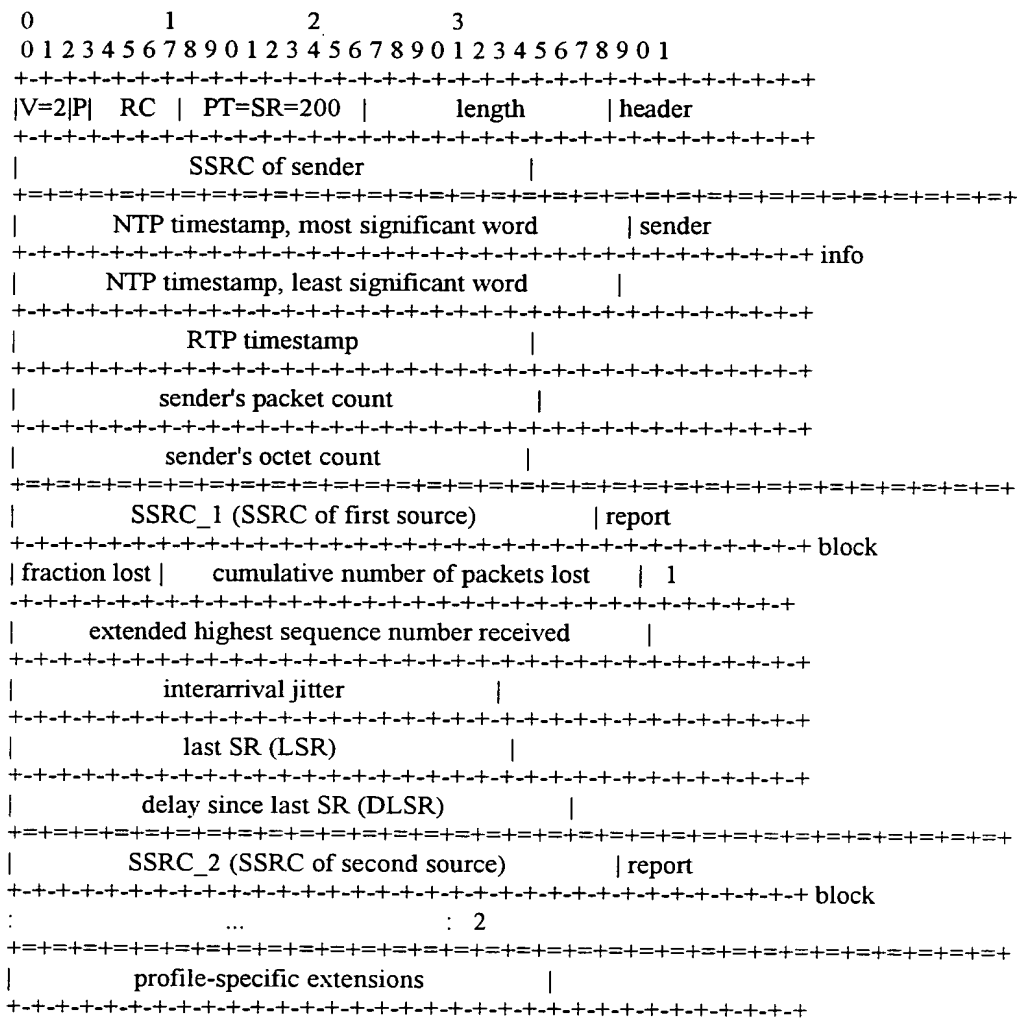
6.3 Sender and Receiver Reports

RTP receivers provide reception quality feedback using RTCP report packets which may take one of two forms depending upon whether or not the receiver is also a sender. The only difference between the sender report (SR) and receiver report (RR) forms, besides the packet type code, is that the sender report includes a 20-byte sender information section for use by active senders. The SR is issued if a site has sent any data packets during the interval since issuing the last report or the previous one, otherwise the RR is issued.

Both the SR and RR forms include zero or more reception report blocks, one for each of the synchronization sources from which this receiver has received RTP data packets since the last report. Reports are not issued for contributing sources listed in the CSRC list. Each reception report block provides statistics about the data received from the particular source indicated in that block. Since a maximum of 31 reception report blocks will fit in an SR or RR packet, additional RR packets may be stacked after the initial SR or RR packet as needed to contain the reception reports for all sources heard during the interval since the last report.

The next sections define the formats of the two reports, how they may be extended in a profile-specific manner if an application requires additional feedback information, and how the reports may be used. Details of reception reporting by translators and mixers is given in Section 7.

6.3.1 SR: Sender report RTCP packet



The sender report packet consists of three sections, possibly followed by a fourth profile-specific extension section if defined. The first section, the header, is 8 octets long. The fields have the following meaning:

Meta 1002
p. 130

version (V): 2 bits
Identifies the version of RTP, which is the same in RTCP packets as in RTP data packets. The version defined by this specification is two (2).

padding (P): 1 bit
If the padding bit is set, this RTCP packet contains some additional padding octets at the end which are not part of the control information. The last octet of the padding is a count of how many padding octets should be ignored. Padding may be needed by some encryption algorithms with fixed block sizes. In a compound RTCP packet, padding should only be required on the last individual packet because the compound packet is encrypted as a whole.

reception report count (RC): 5 bits
The number of reception report blocks contained in this packet. A value of zero is valid.

packet type (PT): 8 bits
Contains the constant 200 to identify this as an RTCP SR packet.

length: 16 bits
The length of this RTCP packet in 32-bit words minus one, including the header and any padding. (The offset of one makes zero a valid length and avoids a possible infinite loop in scanning a compound RTCP packet, while counting 32-bit words avoids a validity check for a multiple of 4.)

SSRC: 32 bits
The synchronization source identifier for the originator of this SR packet.

The second section, the sender information, is 20 octets long and is present in every sender report packet. It summarizes the data transmissions from this sender. The fields have the following meaning:

NTP timestamp: 64 bits
Indicates the wallclock time when this report was sent so that it may be used in combination with timestamps returned in reception reports from other receivers to measure round-trip propagation to those receivers. Receivers should expect that the measurement accuracy of the timestamp may be limited to far less than the resolution of the NTP timestamp. The measurement uncertainty of the timestamp is not indicated as it may not be known. A sender that can keep track of elapsed time but has no notion of wallclock time may use the elapsed time since joining

the session instead. This is assumed to be less than 68 years, so the high bit will be zero. It is permissible to use the sampling clock to estimate elapsed wallclock time. A sender that has no notion of wallclock or elapsed time may set the NTP timestamp to zero.

RTP timestamp: 32 bits
Corresponds to the same time as the NTP timestamp (above), but in the same units and with the same random offset as the RTP timestamps in data packets. This correspondence may be used for intra- and inter-media synchronization for sources whose NTP timestamps are synchronized, and may be used by media-independent receivers to estimate the nominal RTP clock frequency. Note that in most cases this timestamp will not be equal to the RTP timestamp in any adjacent data packet. Rather, it is calculated from the corresponding NTP timestamp using the relationship between the RTP timestamp counter and real time as maintained by periodically checking the wallclock time at a sampling instant.

sender's packet count: 32 bits
The total number of RTP data packets transmitted by the sender since starting transmission up until the time this SR packet was generated. The count is reset if the sender changes its SSRC identifier.

sender's octet count: 32 bits
The total number of payload octets (i.e., not including header or padding) transmitted in RTP data packets by the sender since starting transmission up until the time this SR packet was generated. The count is reset if the sender changes its SSRC identifier. This field can be used to estimate the average payload data rate.

The third section contains zero or more reception report blocks depending on the number of other sources heard by this sender since the last report. Each reception report block conveys statistics on the reception of RTP packets from a single synchronization source. Receivers do not carry over statistics when a source changes its SSRC identifier due to a collision. These statistics are:

SSRC_n (source identifier): 32 bits
The SSRC identifier of the source to which the information in this reception report block pertains.

fraction lost: 8 bits
The fraction of RTP data packets from source SSRC_n lost since the previous SR or RR packet was sent, expressed as a fixed

point number with the binary point at the left edge of the field. (That is equivalent to taking the integer part after multiplying the loss fraction by 256.) This fraction is defined to be the number of packets lost divided by the number of packets expected, as defined in the next paragraph. An implementation is shown in Appendix A.3. If the loss is negative due to duplicates, the fraction lost is set to zero. Note that a receiver cannot tell whether any packets were lost after the last one received, and that there will be no reception report block issued for a source if all packets from that source sent during the last reporting interval have been lost.

cumulative number of packets lost: 24 bits
The total number of RTP data packets from source SSRC_n that have been lost since the beginning of reception. This number is defined to be the number of packets expected less the number of packets actually received, where the number of packets received includes any which are late or duplicates. Thus packets that arrive late are not counted as lost, and the loss may be negative if there are duplicates. The number of packets expected is defined to be the extended last sequence number received, as defined next, less the initial sequence number received. This may be calculated as shown in Appendix A.3.

extended highest sequence number received: 32 bits
The low 16 bits contain the highest sequence number received in an RTP data packet from source SSRC_n, and the most significant 16 bits extend that sequence number with the corresponding count of sequence number cycles, which may be maintained according to the algorithm in Appendix A.1. Note that different receivers within the same session will generate different extensions to the sequence number if their start times differ significantly.

interarrival jitter: 32 bits
An estimate of the statistical variance of the RTP data packet interarrival time, measured in timestamp units and expressed as an unsigned integer. The interarrival jitter J is defined to be the mean deviation (smoothed absolute value) of the difference D in packet spacing at the receiver compared to the sender for a pair of packets. As shown in the equation below, this is equivalent to the difference in the "relative transit time" for the two packets; the relative transit time is the difference between a packet's RTP timestamp and the receiver's clock at the time of arrival, measured in the same units.

If Si is the RTP timestamp from packet i, and Ri is the time of arrival in RTP timestamp units for packet i, then for two packets i and j, D may be expressed as

D(i,j)=(Rj-Ri)-(Sj-Si)=(Rj-Sj)-(Ri-Si)

The interarrival jitter is calculated continuously as each data packet i is received from source SSRC_n, using this difference D for that packet and the previous packet i-1 in order of arrival (not necessarily in sequence), according to the formula

J=J+(|D(i-1,i)|-J)/16

Whenever a reception report is issued, the current value of J is sampled.

The jitter calculation is prescribed here to allow profile-independent monitors to make valid interpretations of reports coming from different implementations. This algorithm is the optimal first-order estimator and the gain parameter 1/16 gives a good noise reduction ratio while maintaining a reasonable rate of convergence [11]. A sample implementation is shown in Appendix A.8.

last SR timestamp (LSR): 32 bits
The middle 32 bits out of 64 in the NTP timestamp (as explained in Section 4) received as part of the most recent RTCP sender report (SR) packet from source SSRC_n. If no SR has been received yet, the field is set to zero.

delay since last SR (DLSR): 32 bits
The delay, expressed in units of 1/65536 seconds, between receiving the last SR packet from source SSRC_n and sending this reception report block. If no SR packet has been received yet from SSRC_n, the DLSR field is set to zero.

Let SSRC_r denote the receiver issuing this receiver report. Source SSRC_n can compute the round propagation delay to SSRC_r by recording the time A when this reception report block is received. It calculates the total round-trip time A-LSR using the last SR timestamp (LSR) field, and then subtracting this field to leave the round-trip propagation delay as (A- LSR - DLSR). This is illustrated in Fig. 2.

This may be used as an approximate measure of distance to cluster receivers, although some links have very asymmetric delays.

6.3.2 RR: Receiver report RTCP packet

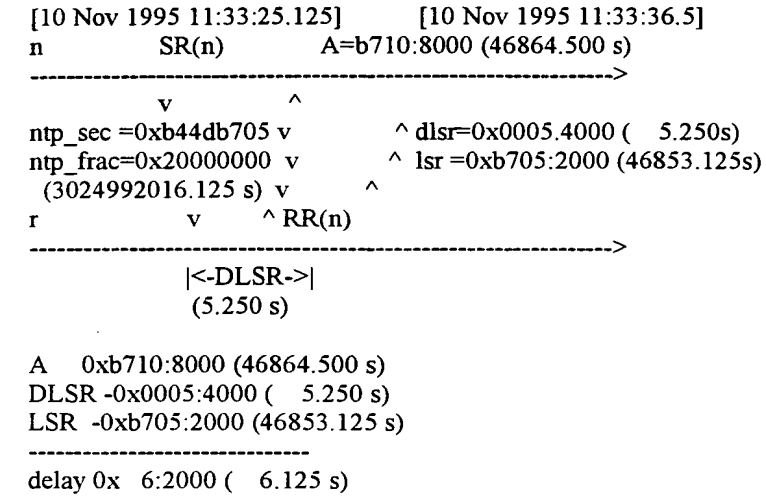
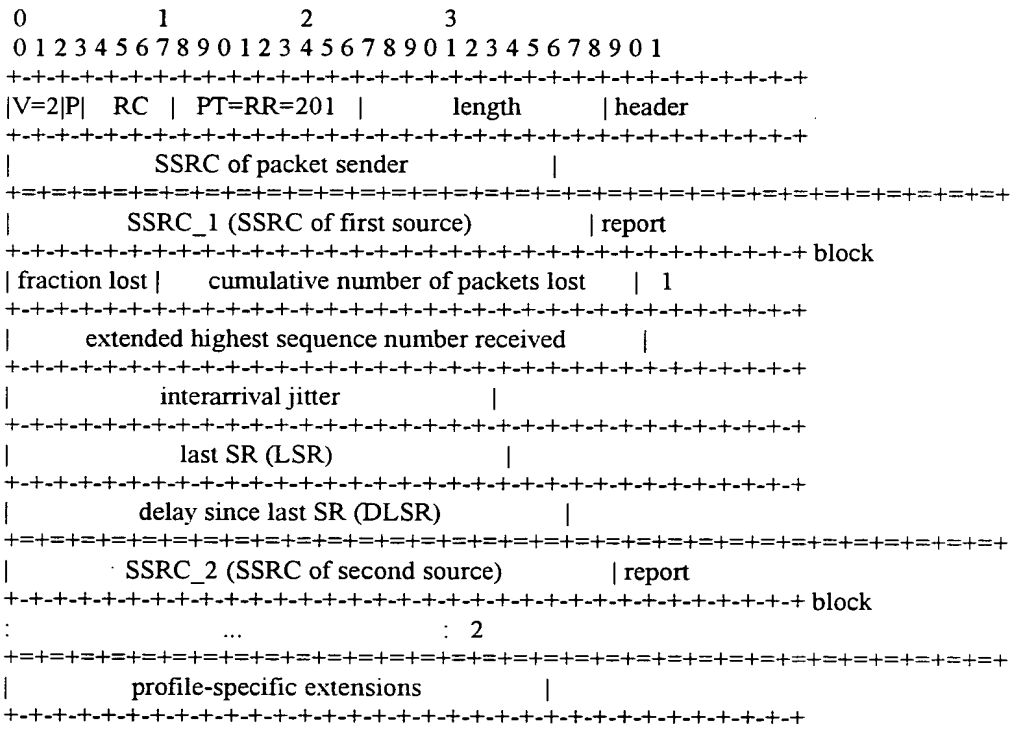


Figure 2: Example for round-trip time computation



Meta 1002
p. 135

The format of the receiver report (RR) packet is the same as that of the SR packet except that the packet type field contains the constant 201 and the five words of sender information are omitted (these are the NTP and RTP timestamps and sender's packet and octet counts). The remaining fields have the same meaning as for the SR packet.

An empty RR packet (RC = 0) is put at the head of a compound RTCP packet when there is no data transmission or reception to report.

6.3.3 Extending the sender and receiver reports

A profile should define profile- or application-specific extensions to the sender report and receiver if there is additional information that should be reported regularly about the sender or receivers. This method should be used in preference to defining another RTCP packet type because it requires less overhead:

- o fewer octets in the packet (no RTCP header or SSRC field);
- o simpler and faster parsing because applications running under that profile would be programmed to always expect the extension fields in the directly accessible location after the reception reports.

If additional sender information is required, it should be included first in the extension for sender reports, but would not be present in receiver reports. If information about receivers is to be included, that data may be structured as an array of blocks parallel to the existing array of reception report blocks; that is, the number of blocks would be indicated by the RC field.

6.3.4 Analyzing sender and receiver reports

It is expected that reception quality feedback will be useful not only for the sender but also for other receivers and third-party monitors. The sender may modify its transmissions based on the feedback; receivers can determine whether problems are local, regional or global; network managers may use profile-independent monitors that receive only the RTCP packets and not the corresponding RTP data packets to evaluate the performance of their networks for multicast distribution.

Cumulative counts are used in both the sender information and receiver report blocks so that differences may be calculated between any two reports to make measurements over both short and long time periods, and to provide resilience against the loss of a report. The difference between the last two reports received can be used to estimate the recent quality of the distribution. The NTP timestamp is

included so that rates may be calculated from these differences over the interval between two reports. Since that timestamp is independent of the clock rate for the data encoding, it is possible to implement encoding- and profile-independent quality monitors.

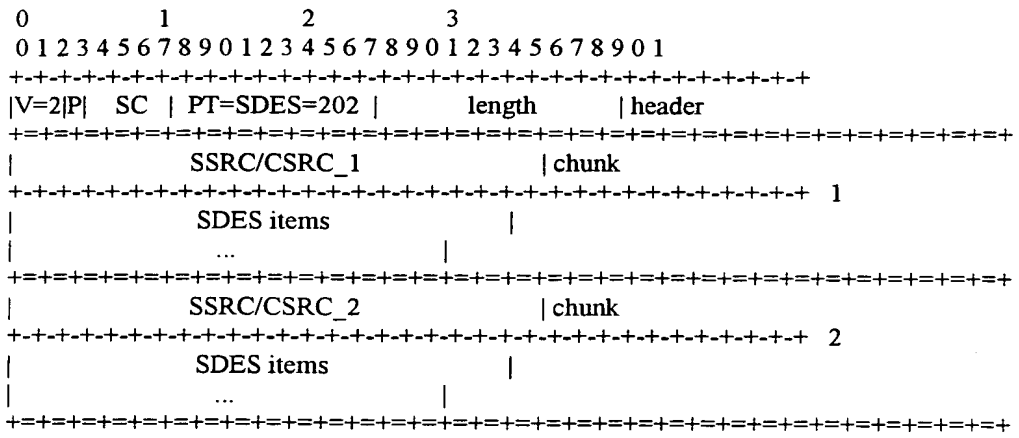
An example calculation is the packet loss rate over the interval between two reception reports. The difference in the cumulative number of packets lost gives the number lost during that interval. The difference in the extended last sequence numbers received gives the number of packets expected during the interval. The ratio of these two is the packet loss fraction over the interval. This ratio should equal the fraction lost field if the two reports are consecutive, but otherwise not. The loss rate per second can be obtained by dividing the loss fraction by the difference in NTP timestamps, expressed in seconds. The number of packets received is the number of packets expected minus the number lost. The number of packets expected may also be used to judge the statistical validity of any loss estimates. For example, 1 out of 5 packets lost has a lower significance than 200 out of 1000.

From the sender information, a third-party monitor can calculate the average payload data rate and the average packet rate over an interval without receiving the data. Taking the ratio of the two gives the average payload size. If it can be assumed that packet loss is independent of packet size, then the number of packets received by a particular receiver times the average payload size (or the corresponding packet size) gives the apparent throughput available to that receiver.

In addition to the cumulative counts which allow long-term packet loss measurements using differences between reports, the fraction lost field provides a short-term measurement from a single report. This becomes more important as the size of a session scales up enough that reception state information might not be kept for all receivers or the interval between reports becomes long enough that only one report might have been received from a particular receiver.

The interarrival jitter field provides a second short-term measure of network congestion. Packet loss tracks persistent congestion while the jitter measure tracks transient congestion. The jitter measure may indicate congestion before it leads to packet loss. Since the interarrival jitter field is only a snapshot of the jitter at the time of a report, it may be necessary to analyze a number of reports from one receiver over time or from multiple receivers, e.g., within a single network.

6.4 SDES: Source description RTCP packet



The SDES packet is a three-level structure composed of a header and zero or more chunks, each of of which is composed of items describing the source identified in that chunk. The items are described individually in subsequent sections.

version (V), padding (P), length:
As described for the SR packet (see Section 6.3.1).

packet type (PT): 8 bits
Contains the constant 202 to identify this as an RTCP SDES packet.

source count (SC): 5 bits
The number of SSRC/CSRC chunks contained in this SDES packet. A value of zero is valid but useless.

Each chunk consists of an SSRC/CSRC identifier followed by a list of zero or more items, which carry information about the SSRC/CSRC. Each chunk starts on a 32-bit boundary. Each item consists of an 8-bit type field, an 8-bit octet count describing the length of the text (thus, not including this two-octet header), and the text itself. Note that the text can be no longer than 255 octets, but this is consistent with the need to limit RTCP bandwidth consumption.

The text is encoded according to the UTF-2 encoding specified in Annex F of ISO standard 10646 [12,13]. This encoding is also known as UTF-8 or UTF-FSS. It is described in "File System Safe UCS Transformation Format (FSS_UTF)", X/Open Preliminary Specification, Document Number P316 and Unicode Technical Report #4. US-ASCII is a subset of this encoding and requires no additional encoding. The

Meta 1002
p. 138

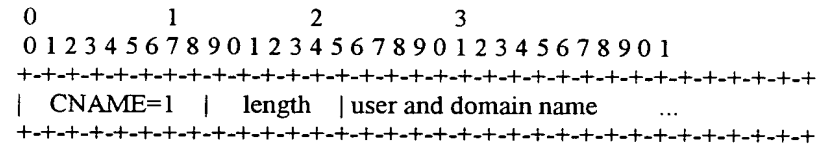
presence of multi-octet encodings is indicated by setting the most significant bit of a character to a value of one.

Items are contiguous, i.e., items are not individually padded to a 32-bit boundary. Text is not null terminated because some multi-octet encodings include null octets. The list of items in each chunk is terminated by one or more null octets, the first of which is interpreted as an item type of zero to denote the end of the list, and the remainder as needed to pad until the next 32-bit boundary. A chunk with zero items (four null octets) is valid but useless.

End systems send one SDES packet containing their own source identifier (the same as the SSRC in the fixed RTP header). A mixer sends one SDES packet containing a chunk for each contributing source from which it is receiving SDES information, or multiple complete SDES packets in the format above if there are more than 31 such sources (see Section 7).

The SDES items currently defined are described in the next sections. Only the CNAME item is mandatory. Some items shown here may be useful only for particular profiles, but the item types are all assigned from one common space to promote shared use and to simplify profile-independent applications. Additional items may be defined in a profile by registering the type numbers with IANA.

6.4.1 CNAME: Canonical end-point identifier SDES item



The CNAME identifier has the following properties:

- o Because the randomly allocated SSRC identifier may change if a conflict is discovered or if a program is restarted, the CNAME item is required to provide the binding from the SSRC identifier to an identifier for the source that remains constant.

- o Like the SSRC identifier, the CNAME identifier should also be unique among all participants within one RTP session.

- o To provide a binding across multiple media tools used by one participant in a set of related RTP sessions, the CNAME should be fixed for that participant.

o To facilitate third-party monitoring, the CNAME should be suitable for either a program or a person to locate the source.

Therefore, the CNAME should be derived algorithmically and not entered manually, when possible. To meet these requirements, the following format should be used unless a profile specifies an alternate syntax or semantics. The CNAME item should have the format "user@host", or "host" if a user name is not available as on single-user systems. For both formats, "host" is either the fully qualified domain name of the host from which the real-time data originates, formatted according to the rules specified in RFC 1034 [14], RFC 1035 [15] and Section 2.1 of RFC 1123 [16]; or the standard ASCII representation of the host's numeric address on the interface used for the RTP communication. For example, the standard ASCII representation of an IP Version 4 address is "dotted decimal", also known as dotted quad. Other address types are expected to have ASCII representations that are mutually unique. The fully qualified domain name is more convenient for a human observer and may avoid the need to send a NAME item in addition, but it may be difficult or impossible to obtain reliably in some operating environments. Applications that may be run in such environments should use the ASCII representation of the address instead.

Examples are "doe@sleepy.megacorp.com" or "doe@192.0.2.89" for a multi-user system. On a system with no user name, examples would be "sleepy.megacorp.com" or "192.0.2.89".

The user name should be in a form that a program such as "finger" or "talk" could use, i.e., it typically is the login name rather than the personal name. The host name is not necessarily identical to the one in the participant's electronic mail address.

This syntax will not provide unique identifiers for each source if an application permits a user to generate multiple sources from one host. Such an application would have to rely on the SSRC to further identify the source, or the profile for that application would have to specify additional syntax for the CNAME identifier.

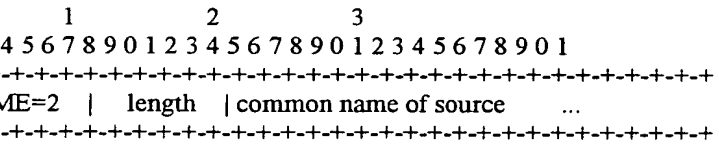
If each application creates its CNAME independently, the resulting CNAMEs may not be identical as would be required to provide a binding across multiple media tools belonging to one participant in a set of related RTP sessions. If cross-media binding is required, it may be necessary for the CNAME of each tool to be externally configured with the same value by a coordination tool.

Application writers should be aware that private network address assignments such as the Net-10 assignment proposed in RFC 1597 [17] may create network addresses that are not globally unique. This would

Meta 1002
p. 140

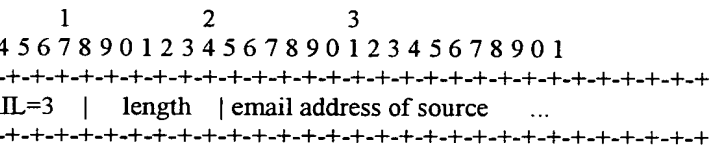
lead to non-unique CNAMEs if hosts with private addresses and no direct IP connectivity to the public Internet have their RTP packets forwarded to the public Internet through an RTP-level translator. (See also RFC 1627 [18].) To handle this case, applications may provide a means to configure a unique CNAME, but the burden is on the translator to translate CNAMEs from private addresses to public addresses if necessary to keep private addresses from being exposed.

6.4.2 NAME: User name SDES item



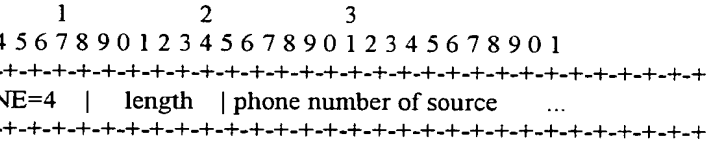
This is the real name used to describe the source, e.g., "John Doe, Bit Recycler, Megacorp". It may be in any form desired by the user. For applications such as conferencing, this form of name may be the most desirable for display in participant lists, and therefore might be sent most frequently of those items other than CNAME. Profiles may establish such priorities. The NAME value is expected to remain constant at least for the duration of a session. It should not be relied upon to be unique among all participants in the session.

6.4.3 EMAIL: Electronic mail address SDES item



The email address is formatted according to RFC 822 [19], for example, "John.Doe@megacorp.com". The EMAIL value is expected to remain constant for the duration of a session.

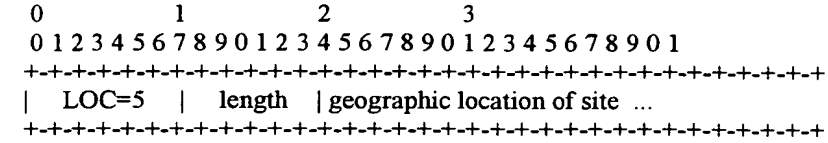
6.4.4 PHONE: Phone number SDES item



The phone number should be formatted with the plus sign replacing the international access code. For example, "+1 908 555 1212" for a number in the United States.

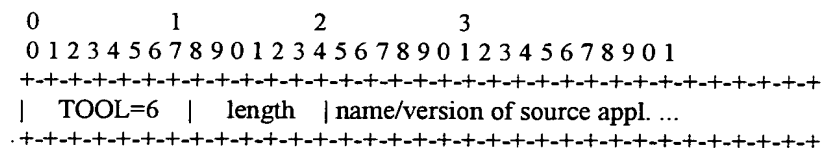
Meta 1002
p. 141

6.4.5 LOC: Geographic user location SDES item



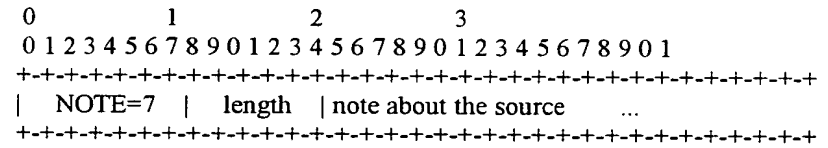
Depending on the application, different degrees of detail are appropriate for this item. For conference applications, a string like "Murray Hill, New Jersey" may be sufficient, while, for an active badge system, strings like "Room 2A244, AT&T BL MH" might be appropriate. The degree of detail is left to the implementation and/or user, but format and content may be prescribed by a profile. The LOC value is expected to remain constant for the duration of a session, except for mobile hosts.

6.4.6 TOOL: Application or tool name SDES item



A string giving the name and possibly version of the application generating the stream, e.g., "videotool 1.2". This information may be useful for debugging purposes and is similar to the Mailer or Mail-System-Version SMTP headers. The TOOL value is expected to remain constant for the duration of the session.

6.4.7 NOTE: Notice/status SDES item

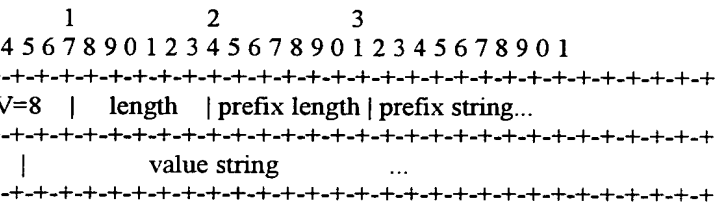


The following semantics are suggested for this item, but these or other semantics may be explicitly defined by a profile. The NOTE item is intended for transient messages describing the current state of the source, e.g., "on the phone, can't talk". Or, during a seminar, this item might be used to convey the title of the talk. It should be used only to carry exceptional information and should not be included routinely by all participants because this would slow down the rate at which reception reports and CNAME are sent, thus impairing the performance of the protocol. In particular, it should not be included

as an item in a user's configuration file nor automatically generated as in a quote-of-the-day.

Since the NOTE item may be important to display while it is active, the rate at which other non-CNAME items such as NAME are transmitted might be reduced so that the NOTE item can take that part of the RTCP bandwidth. When the transient message becomes inactive, the NOTE item should continue to be transmitted a few times at the same repetition rate but with a string of length zero to signal the receivers. However, receivers should also consider the NOTE item inactive if it is not received for a small multiple of the repetition rate, or perhaps 20-30 RTCP intervals.

6.4.8 PRIV: Private extensions SDES item



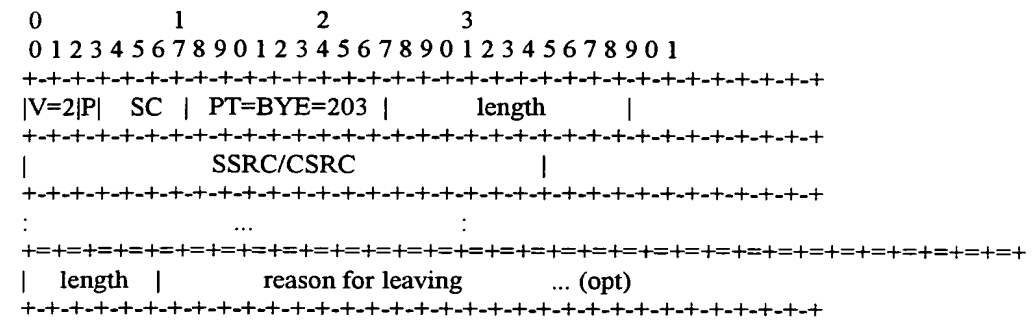
This item is used to define experimental or application-specific SDES extensions. The item contains a prefix consisting of a length-string pair, followed by the value string filling the remainder of the item and carrying the desired information. The prefix length field is 8 bits long. The prefix string is a name chosen by the person defining the PRIV item to be unique with respect to other PRIV items this application might receive. The application creator might choose to use the application name plus an additional subtype identification if needed. Alternatively, it is recommended that others choose a name based on the entity they represent, then coordinate the use of the name within that entity.

Note that the prefix consumes some space within the item's total length of 255 octets, so the prefix should be kept as short as possible. This facility and the constrained RTCP bandwidth should not be overloaded; it is not intended to satisfy all the control communication requirements of all applications.

SDES PRIV prefixes will not be registered by IANA. If some form of the PRIV item proves to be of general utility, it should instead be assigned a regular SDES item type registered with IANA so that no prefix is required. This simplifies use and increases transmission efficiency.

Meta 1002
p. 143

6.5 BYE: Goodbye RTCP packet



The BYE packet indicates that one or more sources are no longer active.

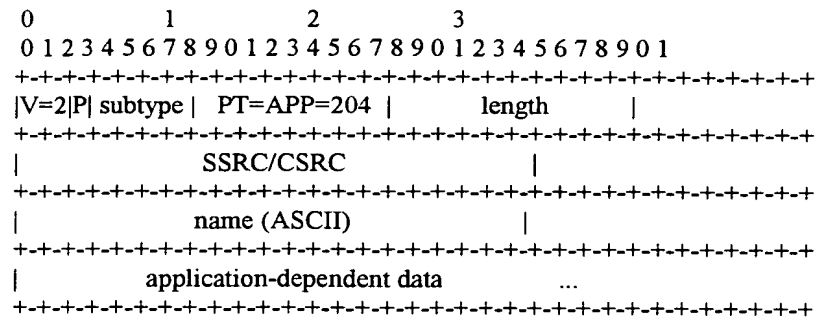
version (V), padding (P), length:
As described for the SR packet (see Section 6.3.1).

packet type (PT): 8 bits
Contains the constant 203 to identify this as an RTCP BYE packet.

source count (SC): 5 bits
The number of SSRC/CSRC identifiers included in this BYE packet. A count value of zero is valid, but useless.

If a BYE packet is received by a mixer, the mixer forwards the BYE packet with the SSRC/CSRC identifier(s) unchanged. If a mixer shuts down, it should send a BYE packet listing all contributing sources it handles, as well as its own SSRC identifier. Optionally, the BYE packet may include an 8-bit octet count followed by that many octets of text indicating the reason for leaving, e.g., "camera malfunction" or "RTP loop detected". The string has the same encoding as that described for SDES. If the string fills the packet to the next 32-bit boundary, the string is not null terminated. If not, the BYE packet is padded with null octets.

6.6 APP: Application-defined RTCP packet



The APP packet is intended for experimental use as new applications and new features are developed, without requiring packet type value registration. APP packets with unrecognized names should be ignored. After testing and if wider use is justified, it is recommended that each APP packet be redefined without the subtype and name fields and registered with the Internet Assigned Numbers Authority using an RTCP packet type.

version (V), padding (P), length:
As described for the SR packet (see Section 6.3.1).

subtype: 5 bits
May be used as a subtype to allow a set of APP packets to be defined under one unique name, or for any application-dependent data.

packet type (PT): 8 bits
Contains the constant 204 to identify this as an RTCP APP packet.

name: 4 octets
A name chosen by the person defining the set of APP packets to be unique with respect to other APP packets this application might receive. The application creator might choose to use the application name, and then coordinate the allocation of subtype values to others who want to define new packet types for the application. Alternatively, it is recommended that others choose a name based on the entity they represent, then coordinate the use of the name within that entity. The name is interpreted as a sequence of four ASCII characters, with uppercase and lowercase characters treated as distinct.

Meta 1002
p. 145

application-dependent data: variable length
Application-dependent data may or may not appear in an APP packet. It is interpreted by the application and not RTP itself. It must be a multiple of 32 bits long.

7. RTP Translators and Mixers

In addition to end systems, RTP supports the notion of "translators" and "mixers", which could be considered as "intermediate systems" at the RTP level. Although this support adds some complexity to the protocol, the need for these functions has been clearly established by experiments with multicast audio and video applications in the Internet. Example uses of translators and mixers given in Section 2.3 stem from the presence of firewalls and low bandwidth connections, both of which are likely to remain.

7.1 General Description

An RTP translator/mixer connects two or more transport-level "clouds". Typically, each cloud is defined by a common network and transport protocol (e.g., IP/UDP), multicast address or pair of unicast addresses, and transport level destination port. (Network-level protocol translators, such as IP version 4 to IP version 6, may be present within a cloud invisibly to RTP.) One system may serve as a translator or mixer for a number of RTP sessions, but each is considered a logically separate entity.

In order to avoid creating a loop when a translator or mixer is installed, the following rules must be observed:

- o Each of the clouds connected by translators and mixers participating in one RTP session either must be distinct from all the others in at least one of these parameters (protocol, address, port), or must be isolated at the network level from the others.

- o A derivative of the first rule is that there must not be multiple translators or mixers connected in parallel unless by some arrangement they partition the set of sources to be forwarded.

Similarly, all RTP end systems that can communicate through one or more RTP translators or mixers share the same SSRC space, that is, the SSRC identifiers must be unique among all these end systems. Section 8.2 describes the collision resolution algorithm by which SSRC identifiers are kept unique and loops are detected.

There may be many varieties of translators and mixers designed for different purposes and applications. Some examples are to add or remove encryption, change the encoding of the data or the underlying protocols, or replicate between a multicast address and one or more unicast addresses. The distinction between translators and mixers is that a translator passes through the data streams from different sources separately, whereas a mixer combines them to form one new stream:

Translator: Forwards RTP packets with their SSRC identifier intact; this makes it possible for receivers to identify individual sources even though packets from all the sources pass through the same translator and carry the translator's network source address. Some kinds of translators will pass through the data untouched, but others may change the encoding of the data and thus the RTP data payload type and timestamp. If multiple data packets are re-encoded into one, or vice versa, a translator must assign new sequence numbers to the outgoing packets. Losses in the incoming packet stream may induce corresponding gaps in the outgoing sequence numbers. Receivers cannot detect the presence of a translator unless they know by some other means what payload type or transport address was used by the original source.

Mixer: Receives streams of RTP data packets from one or more sources, possibly changes the data format, combines the streams in some manner and then forwards the combined stream. Since the timing among multiple input sources will not generally be synchronized, the mixer will make timing adjustments among the streams and generate its own timing for the combined stream, so it is the synchronization source. Thus, all data packets forwarded by a mixer will be marked with the mixer's own SSRC identifier. In order to preserve the identity of the original sources contributing to the mixed packet, the mixer should insert their SSRC identifiers into the CSRC identifier list following the fixed RTP header of the packet. A mixer that is also itself a contributing source for some packet should explicitly include its own SSRC identifier in the CSRC list for that packet.

For some applications, it may be acceptable for a mixer not to identify sources in the CSRC list. However, this introduces the danger that loops involving those sources could not be detected.

The advantage of a mixer over a translator for applications like audio is that the output bandwidth is limited to that of one source even when multiple sources are active on the input side. This may be important for low-bandwidth links. The disadvantage is that receivers on the output side don't have any control over which sources are

Meta 1002
p. 147

passed through or muted, unless some mechanism is implemented for remote control of the mixer. The regeneration of synchronization information by mixers also means that receivers can't do inter-media synchronization of the original streams. A multi-media mixer could do it.

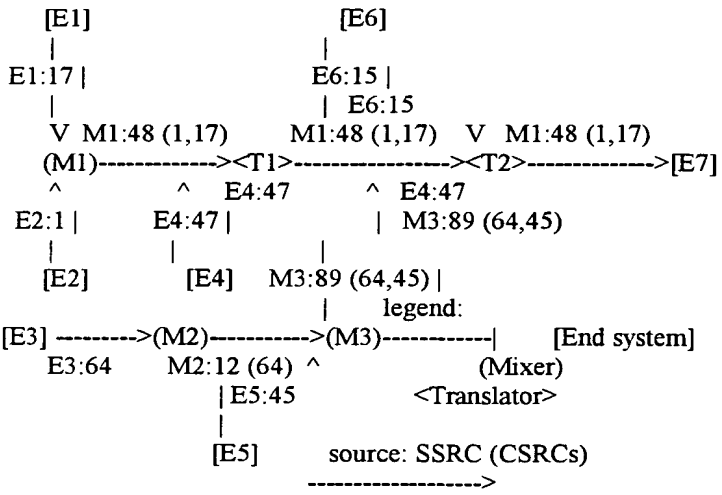


Figure 3: Sample RTP network with end systems, mixers and translators

A collection of mixers and translators is shown in Figure 3 to illustrate their effect on SSRC and CSRC identifiers. In the figure, end systems are shown as rectangles (named E), translators as triangles (named T) and mixers as ovals (named M). The notation "M1: 48(1,17)" designates a packet originating a mixer M1, identified with M1's (random) SSRC value of 48 and two CSRC identifiers, 1 and 17, copied from the SSRC identifiers of packets from E1 and E2.

7.2 RTCP Processing in Translators

In addition to forwarding data packets, perhaps modified, translators and mixers must also process RTCP packets. In many cases, they will take apart the compound RTCP packets received from end systems to aggregate SDES information and to modify the SR or RR packets. Retransmission of this information may be triggered by the packet arrival or by the RTCP interval timer of the translator or mixer itself.

A translator that does not modify the data packets, for example one that just replicates between a multicast address and a unicast address, may simply forward RTCP packets unmodified as well. A

Meta 1002
p. 148

translator that transforms the payload in some way must make corresponding transformations in the SR and RR information so that it still reflects the characteristics of the data and the reception quality. These translators must not simply forward RTCP packets. In general, a translator should not aggregate SR and RR packets from different sources into one packet since that would reduce the accuracy of the propagation delay measurements based on the LSR and DLSR fields.

SR sender information: A translator does not generate its own sender information, but forwards the SR packets received from one cloud to the others. The SSRC is left intact but the sender information must be modified if required by the translation. If a translator changes the data encoding, it must change the "sender's byte count" field. If it also combines several data packets into one output packet, it must change the "sender's packet count" field. If it changes the timestamp frequency, it must change the "RTP timestamp" field in the SR packet.

SR/RR reception report blocks: A translator forwards reception reports received from one cloud to the others. Note that these flow in the direction opposite to the data. The SSRC is left intact. If a translator combines several data packets into one output packet, and therefore changes the sequence numbers, it must make the inverse manipulation for the packet loss fields and the "extended last sequence number" field. This may be complex. In the extreme case, there may be no meaningful way to translate the reception reports, so the translator may pass on no reception report at all or a synthetic report based on its own reception. The general rule is to do what makes sense for a particular translation.

A translator does not require an SSRC identifier of its own, but may choose to allocate one for the purpose of sending reports about what it has received. These would be sent to all the connected clouds, each corresponding to the translation of the data stream as sent to that cloud, since reception reports are normally multicast to all participants.

SDES: Translators typically forward without change the SDES information they receive from one cloud to the others, but may, for example, decide to filter non-CNAME SDES information if bandwidth is limited. The CNAMEs must be forwarded to allow SSRC identifier collision detection to work. A translator that generates its own RR packets must send SDES CNAME information about itself to the same clouds that it sends those RR packets.

BYE: Translators forward BYE packets unchanged. Translators with their own SSRC should generate BYE packets with that SSRC identifier if they are about to cease forwarding packets.

APP: Translators forward APP packets unchanged.

7.3 RTCP Processing in Mixers

Since a mixer generates a new data stream of its own, it does not pass through SR or RR packets at all and instead generates new information for both sides.

SR sender information: A mixer does not pass through sender information from the sources it mixes because the characteristics of the source streams are lost in the mix. As a synchronization source, the mixer generates its own SR packets with sender information about the mixed data stream and sends them in the same direction as the mixed stream.

SR/RR reception report blocks: A mixer generates its own reception reports for sources in each cloud and sends them out only to the same cloud. It does not send these reception reports to the other clouds and does not forward reception reports from one cloud to the others because the sources would not be SSRCs there (only CSRCs).

SDES: Mixers typically forward without change the SDES information they receive from one cloud to the others, but may, for example, decide to filter non-CNAME SDES information if bandwidth is limited. The CNAMEs must be forwarded to allow SSRC identifier collision detection to work. (An identifier in a CSRC list generated by a mixer might collide with an SSRC identifier generated by an end system.) A mixer must send SDES CNAME information about itself to the same clouds that it sends SR or RR packets.

Since mixers do not forward SR or RR packets, they will typically be extracting SDES packets from a compound RTCP packet. To minimize overhead, chunks from the SDES packets may be aggregated into a single SDES packet which is then stacked on an SR or RR packet originating from the mixer. The RTCP packet rate may be different on each side of the mixer.

A mixer that does not insert CSRC identifiers may also refrain from forwarding SDES CNAMEs. In this case, the SSRC identifier spaces in the two clouds are independent. As mentioned earlier, this mode of operation creates a danger that loops can't be detected.

BYE: Mixers need to forward BYE packets. They should generate BYE packets with their own SSRC identifiers if they are about to cease forwarding packets.

APP: The treatment of APP packets by mixers is application-specific.

7.4 Cascaded Mixers

An RTP session may involve a collection of mixers and translators as shown in Figure 3. If two mixers are cascaded, such as M2 and M3 in the figure, packets received by a mixer may already have been mixed and may include a CSRC list with multiple identifiers. The second mixer should build the CSRC list for the outgoing packet using the CSRC identifiers from already-mixed input packets and the SSRC identifiers from unmixed input packets. This is shown in the output arc from mixer M3 labeled M3:89(64,45) in the figure. As in the case of mixers that are not cascaded, if the resulting CSRC list has more than 15 identifiers, the remainder cannot be included.

8. SSRC Identifier Allocation and Use

The SSRC identifier carried in the RTP header and in various fields of RTCP packets is a random 32-bit number that is required to be globally unique within an RTP session. It is crucial that the number be chosen with care in order that participants on the same network or starting at the same time are not likely to choose the same number.

It is not sufficient to use the local network address (such as an IPv4 address) for the identifier because the address may not be unique. Since RTP translators and mixers enable interoperation among multiple networks with different address spaces, the allocation patterns for addresses within two spaces might result in a much higher rate of collision than would occur with random allocation.

Multiple sources running on one host would also conflict.

It is also not sufficient to obtain an SSRC identifier simply by calling random() without carefully initializing the state. An example of how to generate a random identifier is presented in Appendix A.6.

8.1 Probability of Collision

Since the identifiers are chosen randomly, it is possible that two or more sources will choose the same number. Collision occurs with the highest probability when all sources are started simultaneously, for example when triggered automatically by some session management event. If N is the number of sources and L the length of the identifier (here, 32 bits), the probability that two sources

independently pick the same value can be approximated for large N [20] as 1 - exp(-N**2 / 2**(L+1)). For N=1000, the probability is roughly 10**-4.

The typical collision probability is much lower than the worst-case above. When one new source joins an RTP session in which all the other sources already have unique identifiers, the probability of collision is just the fraction of numbers used out of the space. Again, if N is the number of sources and L the length of the identifier, the probability of collision is N / 2**L. For N=1000, the probability is roughly 2*10**-7.

The probability of collision is further reduced by the opportunity for a new source to receive packets from other participants before sending its first packet (either data or control). If the new source keeps track of the other participants (by SSRC identifier), then before transmitting its first packet the new source can verify that its identifier does not conflict with any that have been received, or else choose again.

8.2 Collision Resolution and Loop Detection

Although the probability of SSRC identifier collision is low, all RTP implementations must be prepared to detect collisions and take the appropriate actions to resolve them. If a source discovers at any time that another source is using the same SSRC identifier as its own, it must send an RTCP BYE packet for the old identifier and choose another random one. If a receiver discovers that two other sources are colliding, it may keep the packets from one and discard the packets from the other when this can be detected by different source transport addresses or CNAMEs. The two sources are expected to resolve the collision so that the situation doesn't last.

Because the random identifiers are kept globally unique for each RTP session, they can also be used to detect loops that may be introduced by mixers or translators. A loop causes duplication of data and control information, either unmodified or possibly mixed, as in the following examples:

- o A translator may incorrectly forward a packet to the same multicast group from which it has received the packet, either directly or through a chain of translators. In that case, the same packet appears several times, originating from different network sources.

- o Two translators incorrectly set up in parallel, i.e., with the same multicast groups on both sides, would both forward packets from one multicast group to the other. Unidirectional

Meta 1002
p. 152

translators would produce two copies; bidirectional translators would form a loop.

o A mixer can close a loop by sending to the same transport destination upon which it receives packets, either directly or through another mixer or translator. In this case a source might show up both as an SSRC on a data packet and a CSRC in a mixed data packet.

A source may discover that its own packets are being looped, or that packets from another source are being looped (a third-party loop).

Both loops and collisions in the random selection of a source identifier result in packets arriving with the same SSRC identifier but a different source transport address, which may be that of the end system originating the packet or an intermediate system. Consequently, if a source changes its source transport address, it must also choose a new SSRC identifier to avoid being interpreted as a looped source. Loops or collisions occurring on the far side of a translator or mixer cannot be detected using the source transport address if all copies of the packets go through the translator or mixer, however collisions may still be detected when chunks from two RTCP SDES packets contain the same SSRC identifier but different CNAMEs.

To detect and resolve these conflicts, an RTP implementation must include an algorithm similar to the one described below. It ignores packets from a new source or loop that collide with an established source. It resolves collisions with the participant's own SSRC identifier by sending an RTCP BYE for the old identifier and choosing a new one. However, when the collision was induced by a loop of the participant's own packets, the algorithm will choose a new identifier only once and thereafter ignore packets from the looping source transport address. This is required to avoid a flood of BYE packets.

This algorithm depends upon the source transport address being the same for both RTP and RTCP packets from a source. The algorithm would require modifications to support applications that don't meet this constraint.

This algorithm requires keeping a table indexed by source identifiers and containing the source transport address from which the identifier was (first) received, along with other state for that source. Each SSRC or CSRC identifier received in a data or control packet is looked up in this table in order to process that data or control information. For control packets, each element with its own SSRC, for example an SDES chunk, requires a separate lookup. (The SSRC in a reception report block is an exception.) If the SSRC or CSRC is not

found, a new entry is created. These table entries are removed when an RTCP BYE packet is received with the corresponding SSRC, or after no packets have arrived for a relatively long time (see Section 6.2.1).

In order to track loops of the participant's own data packets, it is also necessary to keep a separate list of source transport addresses (not identifiers) that have been found to be conflicting. Note that this should be a short list, usually empty. Each element in this list stores the source address plus the time when the most recent conflicting packet was received. An element may be removed from the list when no conflicting packet has arrived from that source for a time on the order of 10 RTCP report intervals (see Section 6.2).

For the algorithm as shown, it is assumed that the participant's own source identifier and state are included in the source identifier table. The algorithm could be restructured to first make a separate comparison against the participant's own source identifier.

```
IF the SSRC or CSRC identifier is not found in the source
  identifier table:
THEN create a new entry storing the source transport address
    and the SSRC or CSRC along with other state.
    CONTINUE with normal processing.

(identifier is found in the table)

IF the source transport address from the packet matches
  the one saved in the table entry for this identifier:
THEN CONTINUE with normal processing.

(an identifier collision or a loop is indicated)

IF the source identifier is not the participant's own:
THEN IF the source identifier is from an RTCP SDES chunk
      containing a CNAME item that differs from the CNAME
      in the table entry:
    THEN (optionally) count a third-party collision.
    ELSE (optionally) count a third-party loop.
    ABORT processing of data packet or control element.

(a collision or loop of the participant's own data)

IF the source transport address is found in the list of
  conflicting addresses:
THEN IF the source identifier is not from an RTCP SDES chunk
      containing a CNAME item OR if that CNAME is the
      participant's own:
```

```
        THEN (optionally) count occurrence of own traffic looped.
             mark current time in conflicting address list entry.
             ABORT processing of data packet or control element.
    log occurrence of a collision.
    create a new entry in the conflicting address list and
    mark current time.
    send an RTCP BYE packet with the old SSRC identifier.
    choose a new identifier.
    create a new entry in the source identifier table with the
      old SSRC plus the source transport address from the packet
      being processed.
    CONTINUE with normal processing.
```

In this algorithm, packets from a newly conflicting source address will be ignored and packets from the original source will be kept. (If the original source was through a mixer and later the same source is received directly, the receiver may be well advised to switch unless other sources in the mix would be lost.) If no packets arrive from the original source for an extended period, the table entry will be timed out and the new source will be able to take over. This might occur if the original source detects the collision and moves to a new source identifier, but in the usual case an RTCP BYE packet will be received from the original source to delete the state without having to wait for a timeout.

When a new SSRC identifier is chosen due to a collision, the candidate identifier should first be looked up in the source identifier table to see if it was already in use by some other source. If so, another candidate should be generated and the process repeated.

A loop of data packets to a multicast destination can cause severe network flooding. All mixers and translators are required to implement a loop detection algorithm like the one here so that they can break loops. This should limit the excess traffic to no more than one duplicate copy of the original traffic, which may allow the session to continue so that the cause of the loop can be found and fixed. However, in extreme cases where a mixer or translator does not properly break the loop and high traffic levels result, it may be necessary for end systems to cease transmitting data or control packets entirely. This decision may depend upon the application. An error condition should be indicated as appropriate. Transmission might be attempted again periodically after a long, random time (on the order of minutes).

Meta 1002
p. 155

9. Security

Lower layer protocols may eventually provide all the security services that may be desired for applications of RTP, including authentication, integrity, and confidentiality. These services have recently been specified for IP. Since the need for a confidentiality service is well established in the initial audio and video applications that are expected to use RTP, a confidentiality service is defined in the next section for use with RTP and RTCP until lower layer services are available. The overhead on the protocol for this service is low, so the penalty will be minimal if this service is obsoleted by lower layer services in the future.

Alternatively, other services, other implementations of services and other algorithms may be defined for RTP in the future if warranted. The selection presented here is meant to simplify implementation of interoperable, secure applications and provide guidance to implementors. No claim is made that the methods presented here are appropriate for a particular security need. A profile may specify which services and algorithms should be offered by applications, and may provide guidance as to their appropriate use.

Key distribution and certificates are outside the scope of this document.

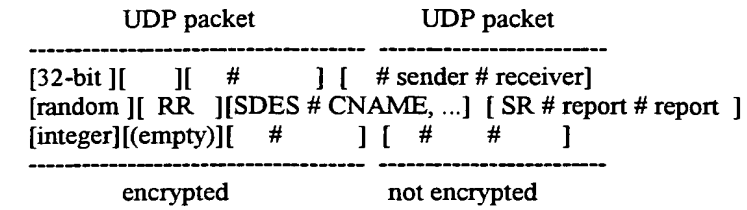
9.1 Confidentiality

Confidentiality means that only the intended receiver(s) can decode the received packets; for others, the packet contains no useful information. Confidentiality of the content is achieved by encryption.

When encryption of RTP or RTCP is desired, all the octets that will be encapsulated for transmission in a single lower-layer packet are encrypted as a unit. For RTCP, a 32-bit random number is prepended to the unit before encryption to deter known plaintext attacks. For RTP, no prefix is required because the sequence number and timestamp fields are initialized with random offsets.

For RTCP, it is allowed to split a compound RTCP packet into two lower-layer packets, one to be encrypted and one to be sent in the clear. For example, SDES information might be encrypted while reception reports were sent in the clear to accommodate third-party monitors that are not privy to the encryption key. In this example, depicted in Fig. 4, the SDES information must be appended to an RR packet with no reports (and the encrypted) to satisfy the requirement that all compound RTCP packets begin with an SR or RR packet.

Meta 1002
p. 156



#: SSRC

Figure 4: Encrypted and non-encrypted RTCP packets

The presence of encryption and the use of the correct key are confirmed by the receiver through header or payload validity checks. Examples of such validity checks for RTP and RTCP headers are given in Appendices A.1 and A.2.

The default encryption algorithm is the Data Encryption Standard (DES) algorithm in cipher block chaining (CBC) mode, as described in Section 1.1 of RFC 1423 [21], except that padding to a multiple of 8 octets is indicated as described for the P bit in Section 5.1. The initialization vector is zero because random values are supplied in the RTP header or by the random prefix for compound RTCP packets. For details on the use of CBC initialization vectors, see [22]. Implementations that support encryption should always support the DES algorithm in CBC mode as the default to maximize interoperability. This method is chosen because it has been demonstrated to be easy and practical to use in experimental audio and video tools in operation on the Internet. Other encryption algorithms may be specified dynamically for a session by non-RTP means.

As an alternative to encryption at the RTP level as described above, profiles may define additional payload types for encrypted encodings. Those encodings must specify how padding and other aspects of the encryption should be handled. This method allows encrypting only the data while leaving the headers in the clear for applications where that is desired. It may be particularly useful for hardware devices that will handle both decryption and decoding.

9.2 Authentication and Message Integrity

Authentication and message integrity are not defined in the current specification of RTP since these services would not be directly feasible without a key management infrastructure. It is expected that authentication and integrity services will be provided by lower layer protocols in the future.

10. RTP over Network and Transport Protocols

This section describes issues specific to carrying RTP packets within particular network and transport protocols. The following rules apply unless superseded by protocol-specific definitions outside this specification.

RTP relies on the underlying protocol(s) to provide demultiplexing of RTP data and RTCP control streams. For UDP and similar protocols, RTP uses an even port number and the corresponding RTCP stream uses the next higher (odd) port number. If an application is supplied with an odd number for use as the RTP port, it should replace this number with the next lower (even) number.

RTP data packets contain no length field or other delineation, therefore RTP relies on the underlying protocol(s) to provide a length indication. The maximum length of RTP packets is limited only by the underlying protocols.

If RTP packets are to be carried in an underlying protocol that provides the abstraction of a continuous octet stream rather than messages (packets), an encapsulation of the RTP packets must be defined to provide a framing mechanism. Framing is also needed if the underlying protocol may contain padding so that the extent of the RTP payload cannot be determined. The framing mechanism is not defined here.

A profile may specify a framing method to be used even when RTP is carried in protocols that do provide framing in order to allow carrying several RTP packets in one lower-layer protocol data unit, such as a UDP packet. Carrying several RTP packets in one network or transport packet reduces header overhead and may simplify synchronization between different streams.

11. Summary of Protocol Constants

This section contains a summary listing of the constants defined in this specification.

The RTP payload type (PT) constants are defined in profiles rather than this document. However, the octet of the RTP header which contains the marker bit(s) and payload type must avoid the reserved values 200 and 201 (decimal) to distinguish RTP packets from the RTCP SR and RR packet types for the header validation procedure described in Appendix A.1. For the standard definition of one marker bit and a 7-bit payload type field as shown in this specification, this restriction means that payload types 72 and 73 are reserved.

Meta 1002
p. 158

11.1 RTCP packet types

abbrev.	name	value
SR	sender report	200
RR	receiver report	201
SDES	source description	202
BYE	goodbye	203
APP	application-defined	204

These type values were chosen in the range 200-204 for improved header validity checking of RTCP packets compared to RTP packets or other unrelated packets. When the RTCP packet type field is compared to the corresponding octet of the RTP header, this range corresponds to the marker bit being 1 (which it usually is not in data packets) and to the high bit of the standard payload type field being 1 (since the static payload types are typically defined in the low half). This range was also chosen to be some distance numerically from 0 and 255 since all-zeros and all-ones are common data patterns.

Since all compound RTCP packets must begin with SR or RR, these codes were chosen as an even/odd pair to allow the RTCP validity check to test the maximum number of bits with mask and value.

Other constants are assigned by IANA. Experimenters are encouraged to register the numbers they need for experiments, and then unregister those which prove to be unneeded.

11.2 SDES types

abbrev.	name	value
END	end of SDES list	0
CNAME	canonical name	1
NAME	user name	2
EMAIL	user's electronic mail address	3
PHONE	user's phone number	4
LOC	geographic user location	5
TOOL	name of application or tool	6
NOTE	notice about the source	7
PRIV	private extensions	8

Other constants are assigned by IANA. Experimenters are encouraged to register the numbers they need for experiments, and then unregister those which prove to be unneeded.

12. RTP Profiles and Payload Format Specifications

A complete specification of RTP for a particular application will require one or more companion documents of two types described here: profiles, and payload format specifications.

RTP may be used for a variety of applications with somewhat differing requirements. The flexibility to adapt to those requirements is provided by allowing multiple choices in the main protocol specification, then selecting the appropriate choices or defining extensions for a particular environment and class of applications in a separate profile document. Typically an application will operate under only one profile so there is no explicit indication of which profile is in use. A profile for audio and video applications may be found in the companion Internet-Draft draft-ietf-avt-profile for

The second type of companion document is a payload format specification, which defines how a particular kind of payload data, such as H.261 encoded video, should be carried in RTP. These documents are typically titled "RTP Payload Format for XYZ Audio/Video Encoding". Payload formats may be useful under multiple profiles and may therefore be defined independently of any particular profile. The profile documents are then responsible for assigning a default mapping of that format to a payload type value if needed.

Within this specification, the following items have been identified for possible definition within a profile, but this list is not meant to be exhaustive:

RTP data header: The octet in the RTP data header that contains the marker bit and payload type field may be redefined by a profile to suit different requirements, for example with more or fewer marker bits (Section 5.3).

Payload types: Assuming that a payload type field is included, the profile will usually define a set of payload formats (e.g., media encodings) and a default static mapping of those formats to payload type values. Some of the payload formats may be defined by reference to separate payload format specifications. For each payload type defined, the profile must specify the RTP timestamp clock rate to be used (Section 5.1).

RTP data header additions: Additional fields may be appended to the fixed RTP data header if some additional functionality is required across the profile's class of applications independent of payload type (Section 5.3).

Meta 1002
p. 160

RTP data header extensions: The contents of the first 16 bits of the RTP data header extension structure must be defined if use of that mechanism is to be allowed under the profile for implementation-specific extensions (Section 5.3.1).

RTCP packet types: New application-class-specific RTCP packet types may be defined and registered with IANA.

RTCP report interval: A profile should specify that the values suggested in Section 6.2 for the constants employed in the calculation of the RTCP report interval will be used. Those are the RTCP fraction of session bandwidth, the minimum report interval, and the bandwidth split between senders and receivers. A profile may specify alternate values if they have been demonstrated to work in a scalable manner.

SR/RR extension: An extension section may be defined for the RTCP SR and RR packets if there is additional information that should be reported regularly about the sender or receivers (Section 6.3.3).

SDES use: The profile may specify the relative priorities for RTCP SDES items to be transmitted or excluded entirely (Section 6.2.2); an alternate syntax or semantics for the CNAME item (Section 6.4.1); the format of the LOC item (Section 6.4.5); the semantics and use of the NOTE item (Section 6.4.7); or new SDES item types to be registered with IANA.

Security: A profile may specify which security services and algorithms should be offered by applications, and may provide guidance as to their appropriate use (Section 9).

String-to-key mapping: A profile may specify how a user-provided password or pass phrase is mapped into an encryption key.

Underlying protocol: Use of a particular underlying network or transport layer protocol to carry RTP packets may be required.

Transport mapping: A mapping of RTP and RTCP to transport-level addresses, e.g., UDP ports, other than the standard mapping defined in Section 10 may be specified.

Encapsulation: An encapsulation of RTP packets may be defined to allow multiple RTP data packets to be carried in one lower-layer packet or to provide framing over underlying protocols that do not already do so (Section 10).

Meta 1002
p. 161

It is not expected that a new profile will be required for every application. Within one application class, it would be better to extend an existing profile rather than make a new one in order to facilitate interoperation among the applications since each will typically run under only one profile. Simple extensions such as the definition of additional payload type values or RTCP packet types may be accomplished by registering them through the Internet Assigned Numbers Authority and publishing their descriptions in an addendum to the profile or in a payload format specification.

Meta 1002
p. 162

A. Algorithms

We provide examples of C code for aspects of RTP sender and receiver algorithms. There may be other implementation methods that are faster in particular operating environments or have other advantages. These implementation notes are for informational purposes only and are meant to clarify the RTP specification.

The following definitions are used for all examples; for clarity and brevity, the structure definitions are only valid for 32-bit big-endian (most significant octet first) architectures. Bit fields are assumed to be packed tightly in big-endian bit order, with no additional padding. Modifications would be required to construct a portable implementation.

```
/*
 * rtp.h  --  RTP header file (RFC XXXX)
 */
#include <sys/types.h>

/*
 * The type definitions below are valid for 32-bit architectures and
 * may have to be adjusted for 16- or 64-bit architectures.
 */
typedef unsigned char  u_int8;
typedef unsigned short u_int16;
typedef unsigned int   u_int32;
typedef        short int16;

/*
 * Current protocol version.
 */
#define RTP_VERSION    2

#define RTP_SEQ_MOD (1<<16)
#define RTP_MAX_SDES 255      /* maximum text length for SDES */

typedef enum {
   RTCP_SR   = 200,
   RTCP_RR   = 201,
   RTCP_SDES = 202,
   RTCP_BYE  = 203,
   RTCP_APP  = 204
} rtcp_type_t;

typedef enum {
   RTCP_SDES_END   = 0,
   RTCP_SDES_CNAME = 1,
```


Meta 1002
p. 163

```
    RTCP_SDES_NAME  = 2,
    RTCP_SDES_EMAIL = 3,
    RTCP_SDES_PHONE = 4,
    RTCP_SDES_LOC   = 5,
    RTCP_SDES_TOOL  = 6,
    RTCP_SDES_NOTE  = 7,
    RTCP_SDES_PRIV  = 8
  } rtcp_sdes_type_t;

  /*
   * RTP data header
   */
  typedef struct {
    unsigned int version:2;  /* protocol version */
    unsigned int p:1;        /* padding flag */
    unsigned int x:1;        /* header extension flag */
    unsigned int cc:4;       /* CSRC count */
    unsigned int m:1;        /* marker bit */
    unsigned int pt:7;       /* payload type */
    u_int16 seq;             /* sequence number */
    u_int32 ts;              /* timestamp */
    u_int32 ssrc;            /* synchronization source */
    u_int32 csrc[1];         /* optional CSRC list */
  } rtp_hdr_t;

  /*
   * RTCP common header word
   */
  typedef struct {
    unsigned int version:2;  /* protocol version */
    unsigned int p:1;        /* padding flag */
    unsigned int count:5;    /* varies by packet type */
    unsigned int pt:8;       /* RTCP packet type */
    u_int16 length;          /* pkt len in words, w/o this word */
  } rtcp_common_t;

  /*
   * Big-endian mask for version, padding bit and packet type pair
   */
  #define RTCP_VALID_MASK (0xc000 | 0x2000 | 0xfe)
  #define RTCP_VALID_VALUE ((RTP_VERSION << 14) | RTCP_SR)

  /*
   * Reception report block
   */
  typedef struct {
    u_int32 ssrc;            /* data source being reported */
    unsigned int fraction:8; /* fraction lost since last SR/RR */
```

```
    int lost:24;          /* cumul. no. pkts lost (signed!) */
    u_int32 last_seq;     /* extended last seq. no. received */
    u_int32 jitter;       /* interarrival jitter */
    u_int32 lsr;          /* last SR packet from this source */
    u_int32 dlsr;         /* delay since last SR packet */
  } rtcp_rr_t;

  /*
   * SDES item
   */
  typedef struct {
    u_int8 type;          /* type of item (rtcp_sdes_type_t) */
    u_int8 length;        /* length of item (in octets) */
    char data[1];         /* text, not null-terminated */
  } rtcp_sdes_item_t;

  /*
   * One RTCP packet
   */
  typedef struct {
    rtcp_common_t common;    /* common header */
    union {
      /* sender report (SR) */
      struct {
        u_int32 ssrc;     /* sender generating this report */
        u_int32 ntp_sec;  /* NTP timestamp */
        u_int32 ntp_frac;
        u_int32 rtp_ts;   /* RTP timestamp */
        u_int32 psent;    /* packets sent */
        u_int32 osent;    /* octets sent */
        rtcp_rr_t rr[1];  /* variable-length list */
      } sr;

      /* reception report (RR) */
      struct {
        u_int32 ssrc;     /* receiver generating this report */
        rtcp_rr_t rr[1];  /* variable-length list */
      } rr;

      /* source description (SDES) */
      struct rtcp_sdes {
        u_int32 src;      /* first SSRC/CSRC */
        rtcp_sdes_item_t item[1]; /* list of SDES items */
      } sdes;

      /* BYE */
      struct {
        u_int32 src[1];   /* list of sources */
```

```
        /* can't express trailing text for reason */
      } bye;
    } r;
  } rtcp_t;

  typedef struct rtcp_sdes rtcp_sdes_t;

  /*
   * Per-source state information
   */
  typedef struct {
    u_int16 max_seq;      /* highest seq. number seen */
    u_int32 cycles;       /* shifted count of seq. number cycles */
    u_int32 base_seq;     /* base seq number */
    u_int32 bad_seq;      /* last 'bad' seq number + 1 */
    u_int32 probation;    /* sequ. packets till source is valid */
    u_int32 received;     /* packets received */
    u_int32 expected_prior; /* packet expected at last interval */
    u_int32 received_prior; /* packet received at last interval */
    u_int32 transit;      /* relative trans time for prev pkt */
    u_int32 jitter;       /* estimated jitter */
    /* ... */
  } source;
```

A.1 RTP Data Header Validity Checks

An RTP receiver should check the validity of the RTP header on incoming packets since they might be encrypted or might be from a different application that happens to be misaddressed. Similarly, if encryption is enabled, the header validity check is needed to verify that incoming packets have been correctly decrypted, although a failure of the header validity check (e.g., unknown payload type) may not necessarily indicate decryption failure.

Only weak validity checks are possible on an RTP data packet from a source that has not been heard before:

- o RTP version field must equal 2.
- o The payload type must be known, in particular it must not be equal to SR or RR.
- o If the P bit is set, then the last octet of the packet must contain a valid octet count, in particular, less than the total packet length minus the header size.
- o The X bit must be zero if the profile does not specify that the header extension mechanism may be used. Otherwise, the

Meta 1002
p. 166

extension length field must be less than the total packet size minus the fixed header length and padding.

o The length of the packet must be consistent with CC and payload type (if payloads have a known length).

The last three checks are somewhat complex and not always possible, leaving only the first two which total just a few bits. If the SSRC identifier in the packet is one that has been received before, then the packet is probably valid and checking if the sequence number is in the expected range provides further validation. If the SSRC identifier has not been seen before, then data packets carrying that identifier may be considered invalid until a small number of them arrive with consecutive sequence numbers.

The routine update_seq shown below ensures that a source is declared valid only after MIN_SEQUENTIAL packets have been received in sequence. It also validates the sequence number seq of a newly received packet and updates the sequence state for the packet's source in the structure to which s points.

When a new source is heard for the first time, that is, its SSRC identifier is not in the table (see Section 8.2), and the per-source state is allocated for it, s->probation should be set to the number of sequential packets required before declaring a source valid (parameter MIN_SEQUENTIAL) and s->max_seq initialized to seq-1 s->probation marks the source as not yet valid so the state may be discarded after a short timeout rather than a long one, as discussed in Section 6.2.1.

After a source is considered valid, the sequence number is considered valid if it is no more than MAX_DROPOUT ahead of s->max_seq nor more than MAX_MISORDER behind. If the new sequence number is ahead of max_seq modulo the RTP sequence number range (16 bits), but is smaller than max_seq , it has wrapped around and the (shifted) count of sequence number cycles is incremented. A value of one is returned to indicate a valid sequence number.

Otherwise, the value zero is returned to indicate that the validation failed, and the bad sequence number is stored. If the next packet received carries the next higher sequence number, it is considered the valid start of a new packet sequence presumably caused by an extended dropout or a source restart. Since multiple complete sequence number cycles may have been missed, the packet loss statistics are reset.

Typical values for the parameters are shown, based on a maximum misordering time of 2 seconds at 50 packets/second and a maximum

Meta 1002
p. 167

dropout of 1 minute. The dropout parameter MAX_DROPOUT should be a small fraction of the 16-bit sequence number space to give a reasonable probability that new sequence numbers after a restart will not fall in the acceptable range for sequence numbers from before the restart.

```
void init_seq(source *s, u_int16 seq)
{
   s->base_seq = seq - 1;
   s->max_seq = seq;
   s->bad_seq = RTP_SEQ_MOD + 1;
   s->cycles = 0;
   s->received = 0;
   s->received_prior = 0;
   s->expected_prior = 0;
   /* other initialization */
}

int update_seq(source *s, u_int16 seq)
{
   u_int16 udelta = seq - s->max_seq;
   const int MAX_DROPOUT = 3000;
   const int MAX_MISORDER = 100;
   const int MIN_SEQUENTIAL = 2;

   /*
    * Source is not valid until MIN_SEQUENTIAL packets with
    * sequential sequence numbers have been received.
    */
   if (s->probation) {
      /* packet is in sequence */
      if (seq == s->max_seq + 1) {
         s->probation--;
         s->max_seq = seq;
         if (s->probation == 0) {
            init_seq(s, seq);
            s->received++;
            return 1;
         }
      } else {
         s->probation = MIN_SEQUENTIAL - 1;
         s->max_seq = seq;
      }
      return 0;
   } else if (udelta < MAX_DROPOUT) {
      /* in order, with permissible gap */
      if (seq < s->max_seq) {
         /*
```

```
            * Sequence number wrapped - count another 64K cycle.
            */
           s->cycles += RTP_SEQ_MOD;
         }
         s->max_seq = seq;
       } else if (udelta <= RTP_SEQ_MOD - MAX_MISORDER) {
         /* the sequence number made a very large jump */
         if (seq == s->bad_seq) {
           /*
            * Two sequential packets -- assume that the other side
            * restarted without telling us so just re-sync
            * (i.e., pretend this was the first packet).
            */
           init_seq(s, seq);
         }
         else {
           s->bad_seq = (seq + 1) & (RTP_SEQ_MOD-1);
           return 0;
         }
       } else {
         /* duplicate or reordered packet */
       }
       s->received++;
       return 1;
     }
```

The validity check can be made stronger requiring more than two packets in sequence. The disadvantages are that a larger number of initial packets will be discarded and that high packet loss rates could prevent validation. However, because the RTCP header validation is relatively strong, if an RTCP packet is received from a source before the data packets, the count could be adjusted so that only two packets are required in sequence. If initial data loss for a few seconds can be tolerated, an application could choose to discard all data packets from a source until a valid RTCP packet has been received from that source.

Depending on the application and encoding, algorithms may exploit additional knowledge about the payload format for further validation. For payload types where the timestamp increment is the same for all packets, the timestamp values can be predicted from the previous packet received from the same source using the sequence number difference (assuming no change in payload type).

A strong "fast-path" check is possible since with high probability the first four octets in the header of a newly received RTP data packet will be just the same as that of the previous packet from the same SSRC except that the sequence number will have increased by one.

Similarly, a single-entry cache may be used for faster SSRC lookups in applications where data is typically received from one source at a time.

A.2 RTCP Header Validity Checks

The following checks can be applied to RTCP packets.

- o RTP version field must equal 2.
- o The payload type field of the first RTCP packet in a compound packet must be equal to SR or RR.
- o The padding bit (P) should be zero for the first packet of a compound RTCP packet because only the last should possibly need padding.
- o The length fields of the individual RTCP packets must total to the overall length of the compound RTCP packet as received. This is a fairly strong check.

The code fragment below performs all of these checks. The packet type is not checked for subsequent packets since unknown packet types may be present and should be ignored.

```
u_int32 len;      /* length of compound RTCP packet in words */
rtcp_t *r;        /* RTCP header */
rtcp_t *end;      /* end of compound RTCP packet */

if ((*(u_int16 *)r & RTCP_VALID_MASK) != RTCP_VALID_VALUE) {
   /* something wrong with packet format */
}
end = (rtcp_t *)((u_int32 *)r + len);

do r = (rtcp_t *)((u_int32 *)r + r->common.length + 1);
while (r < end && r->common.version == 2);

if (r != end) {
   /* something wrong with packet format */
}
```

A.3 Determining the Number of RTP Packets Expected and Lost

In order to compute packet loss rates, the number of packets expected and actually received from each source needs to be known, using per-source state information defined in struct source referenced via pointer s in the code below. The number of packets received is simply the count of packets as they arrive, including any late or duplicate

packets. The number of packets expected can be computed by the receiver as the difference between the highest sequence number received (s->max_seq) and the first sequence number received (s->base_seq). Since the sequence number is only 16 bits and will wrap around, it is necessary to extend the highest sequence number with the (shifted) count of sequence number wraparounds (s->cycles). Both the received packet count and the count of cycles are maintained the RTP header validity check routine in Appendix A.1.

```
extended_max = s->cycles + s->max_seq;
expected = extended_max - s->base_seq + 1;
```

The number of packets lost is defined to be the number of packets expected less the number of packets actually received:

```
lost = expected - s->received;
```

Since this number is carried in 24 bits, it should be clamped at 0xffffff rather than wrap around to zero.

The fraction of packets lost during the last reporting interval (since the previous SR or RR packet was sent) is calculated from differences in the expected and received packet counts across the interval, where expected_prior and received_prior are the values saved when the previous reception report was generated:

```
expected_interval = expected - s->expected_prior;
s->expected_prior = expected;
received_interval = s->received - s->received_prior;
s->received_prior = s->received;
lost_interval = expected_interval - received_interval;
if (expected_interval == 0 || lost_interval <= 0) fraction = 0;
else fraction = (lost_interval << 8) / expected_interval;
```

The resulting fraction is an 8-bit fixed point number with the binary point at the left edge.

A.4 Generating SDES RTCP Packets

This function builds one SDES chunk into buffer b composed of argc items supplied in arrays type , value and length b

```
char *rtp_write_sdes(char *b, u_int32 src, int argc,
                     rtcp_sdes_type_t type[], char *value[],
                     int length[])
{
  rtcp_sdes_t *s = (rtcp_sdes_t *)b;
  rtcp_sdes_item_t *rsp;
```


Meta 1002
p. 171

```
    int i;
    int len;
    int pad;

    /* SSRC header */
    s->src = src;
    rsp = &s->item[0];

    /* SDES items */
    for (i = 0; i < argc; i++) {
      rsp->type = type[i];
      len = length[i];
      if (len > RTP_MAX_SDES) {
        /* invalid length, may want to take other action */
        len = RTP_MAX_SDES;
      }
      rsp->length = len;
      memcpy(rsp->data, value[i], len);
      rsp = (rtcp_sdes_item_t *)&rsp->data[len];
    }

    /* terminate with end marker and pad to next 4-octet boundary */
    len = ((char *) rsp) - b;
    pad = 4 - (len & 0x3);
    b = (char *) rsp;
    while (pad--) *b++ = RTCP_SDES_END;

    return b;
}
```

A.5 Parsing RTCP SDES Packets

This function parses an SDES packet, calling functions find_member() to find a pointer to the information for a session member given the SSRC identifier and member_sdes() to store the new SDES information for that member. This function expects a pointer to the header of the RTCP packet.

```
void rtp_read_sdes(rtcp_t *r)
{
  int count = r->common.count;
  rtcp_sdes_t *sd = &r->r.sdes;
  rtcp_sdes_item_t *rsp, *rspn;
  rtcp_sdes_item_t *end = (rtcp_sdes_item_t *)
                ((u_int32 *)r + r->common.length + 1);
  source *s;

  while (--count >= 0) {
```

```
        rsp = &sd->item[0];
        if (rsp >= end) break;
        s = find_member(sd->src);

        for (; rsp->type; rsp = rspn ) {
          rspn = (rtcp_sdes_item_t *)((char*)rsp+rsp->length+2);
          if (rspn >= end) {
            rsp = rspn;
            break;
          }
          member_sdes(s, rsp->type, rsp->data, rsp->length);
        }
        sd = (rtcp_sdes_t *)
          ((u_int32 *)sd + (((char *)rsp - (char *)sd) >> 2)+1);
      }
      if (count >= 0) {
        /* invalid packet format */
      }
    }
```

A.6 Generating a Random 32-bit Identifier

The following subroutine generates a random 32-bit identifier using the MD5 routines published in RFC 1321 [23]. The system routines may not be present on all operating systems, but they should serve as hints as to what kinds of information may be used. Other system calls that may be appropriate include

o getdomainname() ,

o getwd() , or

o getrusage()

"Live" video or audio samples are also a good source of random numbers, but care must be taken to avoid using a turned-off microphone or blinded camera as a source [7].

Use of this or similar routine is suggested to generate the initial seed for the random number generator producing the RTCP period (as shown in Appendix A.7), to generate the initial values for the sequence number and timestamp, and to generate SSRC values. Since this routine is likely to be CPU-intensive, its direct use to generate RTCP periods is inappropriate because predictability is not an issue. Note that this routine produces the same result on repeated calls until the value of the system clock changes unless different values are supplied for the type argument.

Meta 1002
p. 173

```
/*
 * Generate a random 32-bit quantity.
 */
#include <sys/types.h>  /* u_long */
#include <sys/time.h>   /* gettimeofday() */
#include <unistd.h>     /* get..() */
#include <stdio.h>      /* printf() */
#include <time.h>       /* clock() */
#include <sys/utsname.h> /* uname() */
#include "global.h"     /* from RFC 1321 */
#include "md5.h"        /* from RFC 1321 */

#define MD_CTX MD5_CTX
#define MDInit MD5Init
#define MDUpdate MD5Update
#define MDFinal MD5Final

static u_long md_32(char *string, int length)
{
   MD_CTX context;
   union {
      char   c[16];
      u_long x[4];
   } digest;
   u_long r;
   int i;

   MDInit (&context);
   MDUpdate (&context, string, length);
   MDFinal ((unsigned char *)&digest, &context);
   r = 0;
   for (i = 0; i < 3; i++) {
      r ^= digest.x[i];
   }
   return r;
}                       /* md_32 */


/*
 * Return random unsigned 32-bit quantity. Use 'type' argument if you
 * need to generate several different values in close succession.
 */
u_int32 random32(int type)
{
   struct {
      int     type;
      struct  timeval tv;
      clock_t cpu;
```

```
        pid_t  pid;
        u_long  hid;
        uid_t  uid;
        gid_t  gid;
        struct  utsname name;
    } s;

    gettimeofday(&s.tv, 0);
    uname(&s.name);
    s.type = type;
    s.cpu  = clock();
    s.pid  = getpid();
    s.hid  = gethostid();
    s.uid  = getuid();
    s.gid  = getgid();

    return md_32((char *)&s, sizeof(s));
}                       /* random32 */
```

A.7 Computing the RTCP Transmission Interval

The following function returns the time between transmissions of RTCP packets, measured in seconds. It should be called after sending one compound RTCP packet to calculate the delay until the next should be sent. This function should also be called to calculate the delay before sending the first RTCP packet upon startup rather than send the packet immediately. This avoids any burst of RTCP packets if an application is started at many sites simultaneously, for example as a result of a session announcement.

The parameters have the following meaning:

rtcp_bw: The target RTCP bandwidth, i.e., the total bandwidth that will be used for RTCP packets by all members of this session, in octets per second. This should be 5% of the "session bandwidth" parameter supplied to the application at startup.

senders: Number of active senders since sending last report, known from construction of receiver reports for this RTCP packet. Includes ourselves, if we also sent during this interval.

members: The estimated number of session members, including ourselves. Incremented as we discover new session members from the receipt of RTP or RTCP packets, and decremented as session members leave (via RTCP BYE) or their state is timed out (30 minutes is recommended). On the first call, this parameter should have the value 1.

Meta 1002
p. 175

we_sent: Flag that is true if we have sent data during the last two RTCP intervals. If the flag is true, the compound RTCP packet just sent contained an SR packet.

packet_size: The size of the compound RTCP packet just sent, in octets, including the network encapsulation (e.g., 28 octets for UDP over IP).

avg_rtcp_size: Pointer to estimator for compound RTCP packet size; initialized and updated by this function for the packet just sent, and also updated by an identical line of code in the RTCP receive routine for every RTCP packet received from other participants in the session.

initial: Flag that is true for the first call upon startup to calculate the time until the first report should be sent.

```
#include <math.h>

double rtcp_interval(int members,
                     int senders,
                     double rtcp_bw,
                     int we_sent,
                     int packet_size,
                     int *avg_rtcp_size,
                     int initial)
{
  /*
   * Minimum time between RTCP packets from this site (in seconds).
   * This time prevents the reports from `clumping' when sessions
   * are small and the law of large numbers isn't helping to smooth
   * out the traffic.  It also keeps the report interval from
   * becoming ridiculously small during transient outages like a
   * network partition.
   */
  double const RTCP_MIN_TIME = 5.;
  /*
   * Fraction of the RTCP bandwidth to be shared among active
   * senders.  (This fraction was chosen so that in a typical
   * session with one or two active senders, the computed report
   * time would be roughly equal to the minimum report time so that
   * we don't unnecessarily slow down receiver reports.) The
   * receiver fraction must be 1 - the sender fraction.
   */
  double const RTCP_SENDER_BW_FRACTION = 0.25;
  double const RTCP_RCVR_BW_FRACTION = (1-RTCP_SENDER_BW_FRACTION);
  /*
   * Gain (smoothing constant) for the low-pass filter that
```

```
 * estimates the average RTCP packet size (see Cadzow reference).
 */
double const RTCP_SIZE_GAIN = (1./16.);

double t;                   /* interval */
double rtcp_min_time = RTCP_MIN_TIME;
int n;                      /* no. of members for computation */

/*
 * Very first call at application start-up uses half the min
 * delay for quicker notification while still allowing some time
 * before reporting for randomization and to learn about other
 * sources so the report interval will converge to the correct
 * interval more quickly.  The average RTCP size is initialized
 * to 128 octets which is conservative (it assumes everyone else
 * is generating SRs instead of RRs: 20 IP + 8 UDP + 52 SR + 48
 * SDES CNAME).
 */
if (initial) {
    rtcp_min_time /= 2;
    *avg_rtcp_size = 128;
}

/*
 * If there were active senders, give them at least a minimum
 * share of the RTCP bandwidth.  Otherwise all participants share
 * the RTCP bandwidth equally.
 */
n = members;
if (senders > 0 && senders < members * RTCP_SENDER_BW_FRACTION) {
    if (we_sent) {
        rtcp_bw *= RTCP_SENDER_BW_FRACTION;
        n = senders;
    } else {
        rtcp_bw *= RTCP_RCVR_BW_FRACTION;
        n -= senders;
    }
}

/*
 * Update the average size estimate by the size of the report
 * packet we just sent.
 */
*avg_rtcp_size += (packet_size - *avg_rtcp_size)*RTCP_SIZE_GAIN;

/*
 * The effective number of sites times the average packet size is
 * the total number of octets sent when each site sends a report.
```

```
     * Dividing this by the effective bandwidth gives the time
     * interval over which those packets must be sent in order to
     * meet the bandwidth target, with a minimum enforced.  In that
     * time interval we send one report so this time is also our
     * average time between reports.
     */
    t = (*avg_rtcp_size) * n / rtcp_bw;
    if (t < rtcp_min_time) t = rtcp_min_time;

    /*
     * To avoid traffic bursts from unintended synchronization with
     * other sites, we then pick our actual next report interval as a
     * random number uniformly distributed between 0.5*t and 1.5*t.
     */
    return t * (drand48() + 0.5);
}
```

A.8 Estimating the Interarrival Jitter

The code fragments below implement the algorithm given in Section 6.3.1 for calculating an estimate of the statistical variance of the RTP data interarrival time to be inserted in the interarrival jitter field of reception reports. The inputs are r->ts , the timestamp from the incoming packet, and arrival , the current time in the same units. Here s points to state for the source; s->transit holds the relative transit time for the previous packet, and s->jitter holds the estimated jitter. The jitter field of the reception report is measured in timestamp units and expressed as an unsigned integer, but the jitter estimate is kept in a floating point. As each data packet arrives, the jitter estimate is updated:

```
int transit = arrival - r->ts;
int d = transit - s->transit;
s->transit = transit;
if (d < 0) d = -d;
s->jitter += (1./16.) * ((double)d - s->jitter);
```

When a reception report block (to which rr points) is generated for this member, the current jitter estimate is returned:

```
rr->jitter = (u_int32) s->jitter;
```

Alternatively, the jitter estimate can be kept as an integer, but scaled to reduce round-off error. The calculation is the same except for the last line:

```
s->jitter += d - ((s->jitter + 8) >> 4);
```


Meta 1002
p. 178

In this case, the estimate is sampled for the reception report as:

```
rr->jitter = s->jitter >> 4;
```

B. Security Considerations

RTP suffers from the same security liabilities as the underlying protocols. For example, an impostor can fake source or destination network addresses, or change the header or payload. Within RTCP, the CNAME and NAME information may be used to impersonate another participant. In addition, RTP may be sent via IP multicast, which provides no direct means for a sender to know all the receivers of the data sent and therefore no measure of privacy. Rightly or not, users may be more sensitive to privacy concerns with audio and video communication than they have been with more traditional forms of network communication [24]. Therefore, the use of security mechanisms with RTP is important. These mechanisms are discussed in Section 9.

RTP-level translators or mixers may be used to allow RTP traffic to reach hosts behind firewalls. Appropriate firewall security principles and practices, which are beyond the scope of this document, should be followed in the design and installation of these devices and in the admission of RTP applications for use behind the firewall.

C. Authors' Addresses

Henning Schulzrinne
GMD Fokus
Hardenbergplatz 2
D-10623 Berlin
Germany

EMail: schulzrinne@fokus.gmd.de

Stephen L. Casner
Precept Software, Inc.
21580 Stevens Creek Boulevard, Suite 207
Cupertino, CA 95014
United States

EMail: casner@precept.com

Meta 1002
p. 179

Ron Frederick
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304
United States

EMail: frederic@parc.xerox.com

Van Jacobson
MS 46a-1121
Lawrence Berkeley National Laboratory
Berkeley, CA 94720
United States

EMail: van@ee.lbl.gov

Acknowledgments

This memorandum is based on discussions within the IETF Audio/Video Transport working group chaired by Stephen Casner. The current protocol has its origins in the Network Voice Protocol and the Packet Video Protocol (Danny Cohen and Randy Cole) and the protocol implemented by the vat application (Van Jacobson and Steve McCanne). Christian Huitema provided ideas for the random identifier generator.

D. Bibliography

[1] D. D. Clark and D. L. Tennenhouse, "Architectural considerations for a new generation of protocols," in SIGCOMM Symposium on Communications Architectures and Protocols , (Philadelphia, Pennsylvania), pp. 200--208, IEEE, Sept. 1990. Computer Communications Review, Vol. 20(4), Sept. 1990.

[2] H. Schulzrinne, "Issues in designing a transport protocol for audio and video conferences and other multiparticipant real-time applications", Work in Progress.

[3] D. E. Comer, Internetworking with TCP/IP , vol. 1. Englewood Cliffs, New Jersey: Prentice Hall, 1991.

[4] Postel, J., "Internet Protocol", STD 5, RFC 791, USC/Information Sciences Institute, September 1981.

[5] Mills, D., "Network Time Protocol Version 3", RFC 1305, UDEL, March 1992.

Meta 1002
p. 180

[6] Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC 1700, USC/Information Sciences Institute, October 1994.

[7] Eastlake, D., Crocker, S., and J. Schiller, "Randomness Recommendations for Security", RFC 1750, DEC, Cybercash, MIT, December 1994.

[8] J.-C. Bolot, T. Turletti, and I. Wakeman, "Scalable feedback control for multicast video distribution in the internet," in SIGCOMM Symposium on Communications Architectures and Protocols , (London, England), pp. 58--67, ACM, Aug. 1994.

[9] I. Busse, B. Deffner, and H. Schulzrinne, "Dynamic QoS control of multimedia applications based on RTP," Computer Communications , Jan. 1996.

[10] S. Floyd and V. Jacobson, "The synchronization of periodic routing messages," in SIGCOMM Symposium on Communications Architectures and Protocols (D. P. Sidhu, ed.), (San Francisco, California), pp. 33--44, ACM, Sept. 1993. also in [25].

[11] J. A. Cadzow, Foundations of digital signal processing and data analysis New York, New York: Macmillan, 1987.

[12] International Standards Organization, "ISO/IEC DIS 10646-1:1993 information technology -- universal multiple-octet coded character set (UCS) -- part I: Architecture and basic multilingual plane," 1993.

[13] The Unicode Consortium, The Unicode Standard New York, New York: Addison-Wesley, 1991.

[14] Mockapetris, P., "Domain Names - Concepts and Facilities", STD 13, RFC 1034, USC/Information Sciences Institute, November 1987.

[15] Mockapetris, P., "Domain Names - Implementation and Specification", STD 13, RFC 1035, USC/Information Sciences Institute, November 1987.

[16] Braden, R., "Requirements for Internet Hosts - Application and Support", STD 3, RFC 1123, Internet Engineering Task Force, October 1989.

[17] Rekhter, Y., Moskowitz, R., Karrenberg, D., and G. de Groot, "Address Allocation for Private Internets", RFC 1597, T.J. Watson Research Center, IBM Corp., Chrysler Corp., RIPE NCC, March 1994.

Meta 1002
p. 181

[18] Lear, E., Fair, E., Crocker, D., and T. Kessler, "Network 10 Considered Harmful (Some Practices Shouldn't be Codified)", RFC 1627, Silicon Graphics, Inc., Apple Computer, Inc., Silicon Graphics, Inc., July 1994.

[19] Crocker, D., "Standard for the Format of ARPA Internet Text Messages", STD 11, RFC 822, UDEL, August 1982.

[20] W. Feller, An Introduction to Probability Theory and its Applications, Volume 1 , vol. 1. New York, New York: John Wiley and Sons, third ed., 1968.

[21] Balenson, D., "Privacy Enhancement for Internet Electronic Mail: Part III: Algorithms, Modes, and Identifiers", RFC 1423, TIS, IAB IRTF PSRG, IETF PEM WG, February 1993.

[22] V. L. Voydock and S. T. Kent, "Security mechanisms in high-level network protocols," ACM Computing Surveys , vol. 15, pp. 135--171, June 1983.

[23] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, MIT Laboratory for Computer Science and RSA Data Security, Inc., April 1992.

[24] S. Stubblebine, "Security services for multimedia conferencing," in 16th National Computer Security Conference , (Baltimore, Maryland), pp. 391--395, Sept. 1993.

[25] S. Floyd and V. Jacobson, "The synchronization of periodic routing messages," IEEE/ACM Transactions on Networking , vol. 2, pp. 122-136, April 1994.

Meta 1002
p. 182

Network Working Group
Request for Comments: 2409
Category: Standards Track

D. Harkins
D. Carrel
cisco Systems

November 1998

The Internet Key Exchange (IKE)

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1998). All Rights Reserved.

Table Of Contents

1 Abstract 2
2 Discussion 2
3 Terms and Definitions 3
3.1 Requirements Terminology 3
3.2 Notation 3
3.3 Perfect Forward Secrecty 5
3.4 Security Association 5
4 Introduction 5
5 Exchanges 8
5.1 Authentication with Digital Signatures 10
5.2 Authentication with Public Key Encryption 12
5.3 A Revised method of Authentication with Public Key Encryption 13
5.4 Authentication with a Pre-Shared Key 16
5.5 Quick Mode 16
5.6 New Group Mode 20
5.7 ISAKMP Informational Exchanges 20
6 Oakley Groups 21
6.1 First Oakley Group 21
6.2 Second Oakley Group 22
6.3 Third Oakley Group 22
6.4 Fourth Oakley Group 23
7 Payload Explosion of Complete Exchange 23
7.1 Phase 1 with Main Mode 23
7.2 Phase 2 with Quick Mode 25
8 Perfect Forward Secrecy Example 27
9 Implementation Hints 27

10 Security Considerations 28
11 IANA Considerations 30
12 Acknowledgments 31
13 References 31
Appendix A 33
Appendix B 37
Authors' Addresses 40
Authors' Note 40
Full Copyright Statement 41

1. Abstract

ISAKMP ([MSST98]) provides a framework for authentication and key exchange but does not define them. ISAKMP is designed to be key exchange independant; that is, it is designed to support many different key exchanges.

Oakley ([Orm96]) describes a series of key exchanges-- called "modes"-- and details the services provided by each (e.g. perfect forward secrecy for keys, identity protection, and authentication).

SKEME ([SKEME]) describes a versatile key exchange technique which provides anonymity, repudiability, and quick key refreshment.

This document describes a protocol using part of Oakley and part of SKEME in conjunction with ISAKMP to obtain authenticated keying material for use with ISAKMP, and for other security associations such as AH and ESP for the IETF IPsec DOI.

2. Discussion

This memo describes a hybrid protocol. The purpose is to negotiate, and provide authenticated keying material for, security associations in a protected manner.

Processes which implement this memo can be used for negotiating virtual private networks (VPNs) and also for providing a remote user from a remote site (whose IP address need not be known beforehand) access to a secure host or network.

Client negotiation is supported. Client mode is where the negotiating parties are not the endpoints for which security association negotiation is taking place. When used in client mode, the identities of the end parties remain hidden.

This does not implement the entire Oakley protocol, but only a subset necessary to satisfy its goals. It does not claim conformance or compliance with the entire Oakley protocol nor is it dependant in any way on the Oakley protocol.

Likewise, this does not implement the entire SKEME protocol, but only the method of public key encryption for authentication and its concept of fast re-keying using an exchange of nonces. This protocol is not dependant in any way on the SKEME protocol.

3. Terms and Definitions

3.1 Requirements Terminology

Keywords "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT" and "MAY" that appear in this document are to be interpreted as described in [Bra97].

3.2 Notation

The following notation is used throughout this memo.

HDR is an ISAKMP header whose exchange type is the mode. When writen as HDR* it indicates payload encryption.

SA is an SA negotiation payload with one or more proposals. An initiator MAY provide multiple proposals for negotiation; a responder MUST reply with only one.

<P>_b indicates the body of payload <P>-- the ISAKMP generic vpayload is not included.

SAi_b is the entire body of the SA payload (minus the ISAKMP generic header)-- i.e. the DOI, situation, all proposals and all transforms offered by the Initiator.

CKY-I and CKY-R are the Initiator's cookie and the Responder's cookie, respectively, from the ISAKMP header.

g^xi and g^xr are the Diffie-Hellman ([DH]) public values of the initiator and responder respectively.

g^xy is the Diffie-Hellman shared secret.

KE is the key exchange payload which contains the public information exchanged in a Diffie-Hellman exchange. There is no particular encoding (e.g. a TLV) used for the data of a KE payload.

Nx is the nonce payload; x can be: i or r for the ISAKMP initiator and responder respectively.

IDx is the identification payload for "x". x can be: "ii" or "ir" for the ISAKMP initiator and responder respectively during phase one negotiation; or "ui" or "ur" for the user initiator and responder respectively during phase two. The ID payload format for the Internet DOI is defined in [Pip97].

SIG is the signature payload. The data to sign is exchange-specific.

CERT is the certificate payload.

HASH (and any derivitive such as HASH(2) or HASH_I) is the hash payload. The contents of the hash are specific to the authentication method.

prf(key, msg) is the keyed pseudo-random function-- often a keyed hash function-- used to generate a deterministic output that appears pseudo-random. prf's are used both for key derivations and for authentication (i.e. as a keyed MAC). (See [KBC96]).

SKEYID is a string derived from secret material known only to the active players in the exchange.

SKEYID_e is the keying material used by the ISAKMP SA to protect the confidentiality of its messages.

SKEYID_a is the keying material used by the ISAKMP SA to authenticate its messages.

SKEYID_d is the keying material used to derive keys for non-ISAKMP security associations.

<x>y indicates that "x" is encrypted with the key "y".

--> signifies "initiator to responder" communication (requests).

<-- signifies "responder to initiator" communication (replies).

| signifies concatenation of information-- e.g. X | Y is the concatentation of X with Y.

[x] indicates that x is optional.

Message encryption (when noted by a '*' after the ISAKMP header) MUST begin immediately after the ISAKMP header. When communication is protected, all payloads following the ISAKMP header MUST be encrypted. Encryption keys are generated from SKEYID_e in a manner that is defined for each algorithm.

3.3 Perfect Forward Secrecy

When used in the memo Perfect Forward Secrecy (PFS) refers to the notion that compromise of a single key will permit access to only data protected by a single key. For PFS to exist the key used to protect transmission of data MUST NOT be used to derive any additional keys, and if the key used to protect transmission of data was derived from some other keying material, that material MUST NOT be used to derive any more keys.

Perfect Forward Secrecy for both keys and identities is provided in this protocol. (Sections 5.5 and 8).

3.4 Security Association

A security association (SA) is a set of policy and key(s) used to protect information. The ISAKMP SA is the shared policy and key(s) used by the negotiating peers in this protocol to protect their communication.

4. Introduction

Oakley and SKEME each define a method to establish an authenticated key exchange. This includes payloads construction, the information payloads carry, the order in which they are processed and how they are used.

While Oakley defines "modes", ISAKMP defines "phases". The relationship between the two is very straightforward and IKE presents different exchanges as modes which operate in one of two phases.

Phase 1 is where the two ISAKMP peers establish a secure, authenticated channel with which to communicate. This is called the ISAKMP Security Association (SA). "Main Mode" and "Aggressive Mode" each accomplish a phase 1 exchange. "Main Mode" and "Aggressive Mode" MUST ONLY be used in phase 1.

Phase 2 is where Security Associations are negotiated on behalf of services such as IPsec or any other service which needs key material and/or parameter negotiation. "Quick Mode" accomplishes a phase 2 exchange. "Quick Mode" MUST ONLY be used in phase 2.

"New Group Mode" is not really a phase 1 or phase 2. It follows phase 1, but serves to establish a new group which can be used in future negotiations. "New Group Mode" MUST ONLY be used after phase 1.

The ISAKMP SA is bi-directional. That is, once established, either party may initiate Quick Mode, Informational, and New Group Mode Exchanges. Per the base ISAKMP document, the ISAKMP SA is identified by the Initiator's cookie followed by the Responder's cookie-- the role of each party in the phase 1 exchange dictates which cookie is the Initiator's. The cookie order established by the phase 1 exchange continues to identify the ISAKMP SA regardless of the direction the Quick Mode, Informational, or New Group exchange. In other words, the cookies MUST NOT swap places when the direction of the ISAKMP SA changes.

With the use of ISAKMP phases, an implementation can accomplish very fast keying when necessary. A single phase 1 negotiation may be used for more than one phase 2 negotiation. Additionally a single phase 2 negotiation can request multiple Security Associations. With these optimizations, an implementation can see less than one round trip per SA as well as less than one DH exponentiation per SA. "Main Mode" for phase 1 provides identity protection. When identity protection is not needed, "Aggressive Mode" can be used to reduce round trips even further. Developer hints for doing these optimizations are included below. It should also be noted that using public key encryption to authenticate an Aggressive Mode exchange will still provide identity protection.

This protocol does not define its own DOI per se. The ISAKMP SA, established in phase 1, MAY use the DOI and situation from a non-ISAKMP service (such as the IETF IPSec DOI [Pip97]). In this case an implementation MAY choose to restrict use of the ISAKMP SA for establishment of SAs for services of the same DOI. Alternately, an ISAKMP SA MAY be established with the value zero in both the DOI and situation (see [MSST98] for a description of these fields) and in this case implementations will be free to establish security services for any defined DOI using this ISAKMP SA. If a DOI of zero is used for establishment of a phase 1 SA, the syntax of the identity payloads used in phase 1 is that defined in [MSST98] and not from any DOI-- e.g. [Pip97]-- which may further expand the syntax and semantics of identities.

The following attributes are used by IKE and are negotiated as part of the ISAKMP Security Association. (These attributes pertain only to the ISAKMP Security Association and not to any Security Associations that ISAKMP may be negotiating on behalf of other services.)

- encryption algorithm

- hash algorithm

- authentication method

- information about a group over which to do Diffie-Hellman.

All of these attributes are mandatory and MUST be negotiated. In addition, it is possible to optionally negotiate a psuedo-random function ("prf"). (There are currently no negotiable pseudo-random functions defined in this document. Private use attribute values can be used for prf negotiation between consenting parties). If a "prf" is not negotiation, the HMAC (see [KBC96]) version of the negotiated hash algorithm is used as a pseudo-random function. Other non-mandatory attributes are described in Appendix A. The selected hash algorithm MUST support both native and HMAC modes.

The Diffie-Hellman group MUST be either specified using a defined group description (section 6) or by defining all attributes of a group (section 5.6). Group attributes (such as group type or prime--see Appendix A) MUST NOT be offered in conjunction with a previously defined group (either a reserved group description or a private use description that is established after conclusion of a New Group Mode exchange).

IKE implementations MUST support the following attribute values:

- DES [DES] in CBC mode with a weak, and semi-weak, key check (weak and semi-weak keys are referenced in [Sch96] and listed in Appendix A). The key is derived according to Appendix B.

- MD5 [MD5] and SHA [SHA}.

- Authentication via pre-shared keys.

- MODP over default group number one (see below).

In addition, IKE implementations SHOULD support: 3DES for encryption; Tiger ([TIGER]) for hash; the Digital Signature Standard, RSA [RSA] signatures and authentication with RSA public key encryption; and MODP group number 2. IKE implementations MAY support any additional encryption algorithms defined in Appendix A and MAY support ECP and EC2N groups.

The IKE modes described here MUST be implemented whenever the IETF IPsec DOI [Pip97] is implemented. Other DOIs MAY use the modes described here.

5. Exchanges

There are two basic methods used to establish an authenticated key exchange: Main Mode and Aggressive Mode. Each generates authenticated keying material from an ephemeral Diffie-Hellman exchange. Main Mode MUST be implemented; Aggressive Mode SHOULD be implemented. In addition, Quick Mode MUST be implemented as a mechanism to generate fresh keying material and negotiate non-ISAKMP security services. In addition, New Group Mode SHOULD be implemented as a mechanism to define private groups for Diffie-Hellman exchanges. Implementations MUST NOT switch exchange types in the middle of an exchange.

Exchanges conform to standard ISAKMP payload syntax, attribute encoding, timeouts and retransmits of messages, and informational messages-- e.g a notify response is sent when, for example, a proposal is unacceptable, or a signature verification or decryption was unsuccessful, etc.

The SA payload MUST precede all other payloads in a phase 1 exchange. Except where otherwise noted, there are no requirements for ISAKMP payloads in any message to be in any particular order.

The Diffie-Hellman public value passed in a KE payload, in either a phase 1 or phase 2 exchange, MUST be the length of the negotiated Diffie-Hellman group enforced, if necessary, by pre-pending the value with zeros.

The length of nonce payload MUST be between 8 and 256 bytes inclusive.

Main Mode is an instantiation of the ISAKMP Identity Protect Exchange: The first two messages negotiate policy; the next two exchange Diffie-Hellman public values and ancillary data (e.g. nonces) necessary for the exchange; and the last two messages authenticate the Diffie-Hellman Exchange. The authentication method negotiated as part of the initial ISAKMP exchange influences the composition of the payloads but not their purpose. The XCHG for Main Mode is ISAKMP Identity Protect.

Similarly, Aggressive Mode is an instantiation of the ISAKMP Aggressive Exchange. The first two messages negotiate policy, exchange Diffie-Hellman public values and ancillary data necessary for the exchange, and identities. In addition the second message authenticates the responder. The third message authenticates the initiator and provides a proof of participation in the exchange. The XCHG for Aggressive Mode is ISAKMP Aggressive. The final message MAY NOT be sent under protection of the ISAKMP SA allowing each party to

postpone exponentiation, if desired, until negotiation of this exchange is complete. The graphic depictions of Aggressive Mode show the final payload in the clear; it need not be.

Exchanges in IKE are not open ended and have a fixed number of messages. Receipt of a Certificate Request payload MUST NOT extend the number of messages transmitted or expected.

Security Association negotiation is limited with Aggressive Mode. Due to message construction requirements the group in which the Diffie-Hellman exchange is performed cannot be negotiated. In addition, different authentication methods may further constrain attribute negotiation. For example, authentication with public key encryption cannot be negotiated and when using the revised method of public key encryption for authentication the cipher and hash cannot be negotiated. For situations where the rich attribute negotiation capabilities of IKE are required Main Mode may be required.

Quick Mode and New Group Mode have no analog in ISAKMP. The XCHG values for Quick Mode and New Group Mode are defined in Appendix A.

Main Mode, Aggressive Mode, and Quick Mode do security association negotiation. Security Association offers take the form of Tranform Payload(s) encapsulated in Proposal Payload(s) encapsulated in Security Association (SA) payload(s). If multiple offers are being made for phase 1 exchanges (Main Mode and Aggressive Mode) they MUST take the form of multiple Transform Payloads for a single Proposal Payload in a single SA payload. To put it another way, for phase 1 exchanges there MUST NOT be multiple Proposal Payloads for a single SA payload and there MUST NOT be multiple SA payloads. This document does not proscribe such behavior on offers in phase 2 exchanges.

There is no limit on the number of offers the initiator may send to the responder but conformant implementations MAY choose to limit the number of offers it will inspect for performance reasons.

During security association negotiation, initiators present offers for potential security associations to responders. Responders MUST NOT modify attributes of any offer, attribute encoding excepted (see Appendix A). If the initiator of an exchange notices that attribute values have changed or attributes have been added or deleted from an offer made, that response MUST be rejected.

Four different authentication methods are allowed with either Main Mode or Aggressive Mode-- digital signature, two forms of authentication with public key encryption, or pre-shared key. The value SKEYID is computed seperately for each authentication method.

For signatures: SKEYID = prf(Ni_b | Nr_b, g^xy)
For public key encryption: SKEYID = prf(hash(Ni_b | Nr_b), CKY-I | CKY-R)
For pre-shared keys: SKEYID = prf(pre-shared-key, Ni_b | Nr_b)

The result of either Main Mode or Aggressive Mode is three groups of authenticated keying material:

SKEYID_d = prf(SKEYID, g^xy | CKY-I | CKY-R | 0)
SKEYID_a = prf(SKEYID, SKEYID_d | g^xy | CKY-I | CKY-R | 1)
SKEYID_e = prf(SKEYID, SKEYID_a | g^xy | CKY-I | CKY-R | 2)

and agreed upon policy to protect further communications. The values of 0, 1, and 2 above are represented by a single octet. The key used for encryption is derived from SKEYID_e in an algorithm-specific manner (see appendix B).

To authenticate either exchange the initiator of the protocol generates HASH_I and the responder generates HASH_R where:

HASH_I = prf(SKEYID, g^xi | g^xr | CKY-I | CKY-R | SAi_b | IDii_b)
HASH_R = prf(SKEYID, g^xr | g^xi | CKY-R | CKY-I | SAi_b | IDir_b)

For authentication with digital signatures, HASH_I and HASH_R are signed and verified; for authentication with either public key encryption or pre-shared keys, HASH_I and HASH_R directly authenticate the exchange. The entire ID payload (including ID type, port, and protocol but excluding the generic header) is hashed into both HASH_I and HASH_R.

As mentioned above, the negotiated authentication method influences the content and use of messages for Phase 1 Modes, but not their intent. When using public keys for authentication, the Phase 1 exchange can be accomplished either by using signatures or by using public key encryption (if the algorithm supports it). Following are Phase 1 exchanges with different authentication options.

5.1 IKE Phase 1 Authenticated With Signatures

Using signatures, the ancillary information exchanged during the second roundtrip are nonces; the exchange is authenticated by signing a mutually obtainable hash. Main Mode with signature authentication is described as follows:

```
Initiator                Responder
-----------              -----------
HDR, SA                  -->
                    <--  HDR, SA
HDR, KE, Ni              -->
                    <--  HDR, KE, Nr
HDR*, IDii, [ CERT, ] SIG_I -->
                    <--  HDR*, IDir, [ CERT, ] SIG_R
```

Aggressive mode with signatures in conjunction with ISAKMP is described as follows:

```
Initiator                Responder
-----------              -----------
HDR, SA, KE, Ni, IDii    -->
                    <--  HDR, SA, KE, Nr, IDir,
                           [ CERT, ] SIG_R
HDR, [ CERT, ] SIG_I     -->
```

In both modes, the signed data, SIG_I or SIG_R, is the result of the negotiated digital signature algorithm applied to HASH_I or HASH_R respectively.

In general the signature will be over HASH_I and HASH_R as above using the negotiated prf, or the HMAC version of the negotiated hash function (if no prf is negotiated). However, this can be overridden for construction of the signature if the signature algorithm is tied to a particular hash algorithm (e.g. DSS is only defined with SHA's 160 bit output). In this case, the signature will be over HASH_I and HASH_R as above, except using the HMAC version of the hash algorithm associated with the signature method. The negotiated prf and hash function would continue to be used for all other prescribed pseudo-random functions.

Since the hash algorithm used is already known there is no need to encode its OID into the signature. In addition, there is no binding between the OIDs used for RSA signatures in PKCS #1 and those used in this document. Therefore, RSA signatures MUST be encoded as a private key encryption in PKCS #1 format and not as a signature in PKCS #1 format (which includes the OID of the hash algorithm). DSS signatures MUST be encoded as r followed by s.

One or more certificate payloads MAY be optionally passed.

5.2 Phase 1 Authenticated With Public Key Encryption

Using public key encryption to authenticate the exchange, the ancillary information exchanged is encrypted nonces. Each party's ability to reconstruct a hash (proving that the other party decrypted the nonce) authenticates the exchange.

In order to perform the public key encryption, the initiator must already have the responder's public key. In the case where the responder has multiple public keys, a hash of the certificate the initiator is using to encrypt the ancillary information is passed as part of the third message. In this way the responder can determine which corresponding private key to use to decrypt the encrypted payloads and identity protection is retained.

In addition to the nonce, the identities of the parties (IDii and IDir) are also encrypted with the other party's public key. If the authentication method is public key encryption, the nonce and identity payloads MUST be encrypted with the public key of the other party. Only the body of the payloads are encrypted, the payload headers are left in the clear.

When using encryption for authentication, Main Mode is defined as follows.

```
Initiator                Responder
-----------              -----------
HDR, SA                  -->
                  <--    HDR, SA
HDR, KE, [ HASH(1), ]
 <IDii_b>PubKey_r,
  <Ni_b>PubKey_r         -->
                           HDR, KE, <IDir_b>PubKey_i,
                  <--        <Nr_b>PubKey_i
HDR*, HASH_I             -->
                  <--    HDR*, HASH_R
```

Aggressive Mode authenticated with encryption is described as follows:

```
Initiator                Responder
-----------              -----------
HDR, SA, [ HASH(1),] KE,
 <IDii_b>Pubkey_r,
  <Ni_b>Pubkey_r         -->
                           HDR, SA, KE, <IDir_b>PubKey_i,
                  <--        <Nr_b>PubKey_i, HASH_R
HDR, HASH_I              -->
```

Where HASH(1) is a hash (using the negotiated hash function) of the certificate which the initiator is using to encrypt the nonce and identity.

RSA encryption MUST be encoded in PKCS #1 format. While only the body of the ID and nonce payloads is encrypted, the encrypted data must be preceded by a valid ISAKMP generic header. The payload length is the length of the entire encrypted payload plus header. The PKCS #1 encoding allows for determination of the actual length of the cleartext payload upon decryption.

Using encryption for authentication provides for a plausably deniable exchange. There is no proof (as with a digital signature) that the conversation ever took place since each party can completely reconstruct both sides of the exchange. In addition, security is added to secret generation since an attacker would have to successfully break not only the Diffie-Hellman exchange but also both RSA encryptions. This exchange was motivated by [SKEME].

Note that, unlike other authentication methods, authentication with public key encryption allows for identity protection with Aggressive Mode.

5.3 Phase 1 Authenticated With a Revised Mode of Public Key Encryption

Authentication with Public Key Encryption has significant advantages over authentication with signatures (see section 5.2 above). Unfortunately, this is at the cost of 4 public key operations-- two public key encryptions and two private key decryptions. This authentication mode retains the advantages of authentication using public key encryption but does so with half the public key operations.

In this mode, the nonce is still encrypted using the public key of the peer, however the peer's identity (and the certificate if it is sent) is encrypted using the negotiated symmetric encryption algorithm (from the SA payload) with a key derived from the nonce. This solution adds minimal complexity and state yet saves two costly public key operations on each side. In addition, the Key Exchange payload is also encrypted using the same derived key. This provides additional protection against cryptanalysis of the Diffie-Hellman exchange.

As with the public key encryption method of authentication (section 5.2), a HASH payload may be sent to identify a certificate if the responder has multiple certificates which contain useable public keys (e.g. if the certificate is not for signatures only, either due to certificate restrictions or algorithmic restrictions). If the HASH

payload is sent it MUST be the first payload of the second message exchange and MUST be followed by the encrypted nonce. If the HASH payload is not sent, the first payload of the second message exchange MUST be the encrypted nonce. In addition, the initiator my optionally send a certificate payload to provide the responder with a public key with which to respond.

When using the revised encryption mode for authentication, Main Mode is defined as follows.

```
Initiator                Responder
-----------              -----------
HDR, SA                  -->
                   <--   HDR, SA
HDR, [ HASH(1), ]
 <Ni_b>Pubkey_r,
 <KE_b>Ke_i,
 <IDii_b>Ke_i,
 [<<Cert-I_b>Ke_i]       -->
                         HDR, <Nr_b>PubKey_i,
                           <KE_b>Ke_r,
                   <--     <IDir_b>Ke_r,
HDR*, HASH_I              -->
                   <--   HDR*, HASH_R
```

Aggressive Mode authenticated with the revised encryption method is described as follows:

```
Initiator                Responder
-----------              -----------
HDR, SA, [ HASH(1),]
 <Ni_b>Pubkey_r,
 <KE_b>Ke_i, <IDii_b>Ke_i
 [, <Cert-I_b>Ke_i ]     -->
                         HDR, SA, <Nr_b>PubKey_i,
                           <KE_b>Ke_r, <IDir_b>Ke_r,
                   <--     HASH_R
HDR, HASH_I               -->
```

where HASH(1) is identical to section 5.2. Ke_i and Ke_r are keys to the symmetric encryption algorithm negotiated in the SA payload exchange. Only the body of the payloads are encrypted (in both public key and symmetric operations), the generic payload headers are left in the clear. The payload length includes that added to perform encryption.

The symmetric cipher keys are derived from the decrypted nonces as follows. First the values Ne_i and Ne_r are computed:

Meta 1002
p. 196

Ne_i = prf(Ni_b, CKY-I)
Ne_r = prf(Nr_b, CKY-R)

The keys Ke_i and Ke_r are then taken from Ne_i and Ne_r respectively in the manner described in Appendix B used to derive symmetric keys for use with the negotiated encryption algorithm. If the length of the output of the negotiated prf is greater than or equal to the key length requirements of the cipher, Ke_i and Ke_r are derived from the most significant bits of Ne_i and Ne_r respectively. If the desired length of Ke_i and Ke_r exceed the length of the output of the prf the necessary number of bits is obtained by repeatedly feeding the results of the prf back into itself and concatenating the result until the necessary number has been achieved. For example, if the negotiated encryption algorithm requires 320 bits of key and the output of the prf is only 128 bits, Ke_i is the most significant 320 bits of K, where

K = K1 | K2 | K3 and
K1 = prf(Ne_i, 0)
K2 = prf(Ne_i, K1)
K3 = prf(Ne_i, K2)

For brevity, only derivation of Ke_i is shown; Ke_r is identical. The length of the value 0 in the computation of K1 is a single octet. Note that Ne_i, Ne_r, Ke_i, and Ke_r are all ephemeral and MUST be discarded after use.

Save the requirements on the location of the optional HASH payload and the mandatory nonce payload there are no further payload requirements. All payloads-- in whatever order-- following the encrypted nonce MUST be encrypted with Ke_i or Ke_r depending on the direction.

If CBC mode is used for the symmetric encryption then the initialization vectors (IVs) are set as follows. The IV for encrypting the first payload following the nonce is set to 0 (zero). The IV for subsequent payloads encrypted with the ephemeral symmetric cipher key, Ke_i, is the last ciphertext block of the previous payload. Encrypted payloads are padded up to the nearest block size. All padding bytes, except for the last one, contain 0x00. The last byte of the padding contains the number of the padding bytes used, excluding the last one. Note that this means there will always be padding.

5.4 Phase 1 Authenticated With a Pre-Shared Key

A key derived by some out-of-band mechanism may also be used to authenticate the exchange. The actual establishment of this key is out of the scope of this document.

When doing a pre-shared key authentication, Main Mode is defined as follows:

```
Initiator                Responder
---------                ---------
HDR, SA          -->
                 <--  HDR, SA
HDR, KE, Ni      -->
                 <--  HDR, KE, Nr
HDR*, IDii, HASH_I -->
                 <--  HDR*, IDir, HASH_R
```

Aggressive mode with a pre-shared key is described as follows:

```
Initiator                Responder
---------                ---------
HDR, SA, KE, Ni, IDii -->
                 <--  HDR, SA, KE, Nr, IDir, HASH_R
HDR, HASH_I      -->
```

When using pre-shared key authentication with Main Mode the key can only be identified by the IP address of the peers since HASH_I must be computed before the initiator has processed IDir. Aggressive Mode allows for a wider range of identifiers of the pre-shared secret to be used. In addition, Aggressive Mode allows two parties to maintain multiple, different pre-shared keys and identify the correct one for a particular exchange.

5.5 Phase 2 - Quick Mode

Quick Mode is not a complete exchange itself (in that it is bound to a phase 1 exchange), but is used as part of the SA negotiation process (phase 2) to derive keying material and negotiate shared policy for non-ISAKMP SAs. The information exchanged along with Quick Mode MUST be protected by the ISAKMP SA-- i.e. all payloads except the ISAKMP header are encrypted. In Quick Mode, a HASH payload MUST immediately follow the ISAKMP header and a SA payload MUST immediately follow the HASH. This HASH authenticates the message and also provides liveliness proofs.

The message ID in the ISAKMP header identifies a Quick Mode in progress for a particular ISAKMP SA which itself is identified by the cookies in the ISAKMP header. Since each instance of a Quick Mode uses a unique initialization vector (see Appendix B) it is possible to have multiple simultaneous Quick Modes, based off a single ISAKMP SA, in progress at any one time.

Quick Mode is essentially a SA negotiation and an exchange of nonces that provides replay protection. The nonces are used to generate fresh key material and prevent replay attacks from generating bogus security associations. An optional Key Exchange payload can be exchanged to allow for an additional Diffie-Hellman exchange and exponentiation per Quick Mode. While use of the key exchange payload with Quick Mode is optional it MUST be supported.

Base Quick Mode (without the KE payload) refreshes the keying material derived from the exponentiation in phase 1. This does not provide PFS. Using the optional KE payload, an additional exponentiation is performed and PFS is provided for the keying material.

The identities of the SAs negotiated in Quick Mode are implicitly assumed to be the IP addresses of the ISAKMP peers, without any implied constraints on the protocol or port numbers allowed, unless client identifiers are specified in Quick Mode. If ISAKMP is acting as a client negotiator on behalf of another party, the identities of the parties MUST be passed as IDci and then IDcr. Local policy will dictate whether the proposals are acceptable for the identities specified. If the client identities are not acceptable to the Quick Mode responder (due to policy or other reasons), a Notify payload with Notify Message Type INVALID-ID-INFORMATION (18) SHOULD be sent.

The client identities are used to identify and direct traffic to the appropriate tunnel in cases where multiple tunnels exist between two peers and also to allow for unique and shared SAs with different granularities.

All offers made during a Quick Mode are logically related and must be consistant. For example, if a KE payload is sent, the attribute describing the Diffie-Hellman group (see section 6.1 and [Pip97]) MUST be included in every transform of every proposal of every SA being negotiated. Similarly, if client identities are used, they MUST apply to every SA in the negotiation.

Quick Mode is defined as follows:

```
Initiator                Responder
-----------              -----------
HDR*, HASH(1), SA, Ni
  [, KE ] [, IDci, IDcr ] -->
                    <--  HDR*, HASH(2), SA, Nr
                           [, KE ] [, IDci, IDcr ]
HDR*, HASH(3)              -->
```

Where:
HASH(1) is the prf over the message id (M-ID) from the ISAKMP header concatenated with the entire message that follows the hash including all payload headers, but excluding any padding added for encryption. HASH(2) is identical to HASH(1) except the initiator's nonce-- Ni, minus the payload header-- is added after M-ID but before the complete message. The addition of the nonce to HASH(2) is for a liveliness proof. HASH(3)-- for liveliness-- is the prf over the value zero represented as a single octet, followed by a concatenation of the message id and the two nonces-- the initiator's followed by the responder's-- minus the payload header. In other words, the hashes for the above exchange are:

HASH(1) = prf(SKEYID_a, M-ID | SA | Ni [| KE] [| IDci | IDcr)
HASH(2) = prf(SKEYID_a, M-ID | Ni_b | SA | Nr [| KE] [| IDci | IDcr)
HASH(3) = prf(SKEYID_a, 0 | M-ID | Ni_b | Nr_b)

With the exception of the HASH, SA, and the optional ID payloads, there are no payload ordering restrictions on Quick Mode. HASH(1) and HASH(2) may differ from the illustration above if the order of payloads in the message differs from the illustrative example or if any optional payloads, for example a notify payload, have been chained to the message.

If PFS is not needed, and KE payloads are not exchanged, the new keying material is defined as

KEYMAT = prf(SKEYID_d, protocol | SPI | Ni_b | Nr_b).

If PFS is desired and KE payloads were exchanged, the new keying material is defined as

KEYMAT = prf(SKEYID_d, g(qm)^xy | protocol | SPI | Ni_b | Nr_b)

where g(qm)^xy is the shared secret from the ephemeral Diffie-Hellman exchange of this Quick Mode.

In either case, "protocol" and "SPI" are from the ISAKMP Proposal Payload that contained the negotiated Transform.

A single SA negotiation results in two security assocations-- one inbound and one outbound. Different SPIs for each SA (one chosen by the initiator, the other by the responder) guarantee a different key for each direction. The SPI chosen by the destination of the SA is used to derive KEYMAT for that SA.

For situations where the amount of keying material desired is greater than that supplied by the prf, KEYMAT is expanded by feeding the results of the prf back into itself and concatenating results until the required keying material has been reached. In other words,

```
KEYMAT = K1 | K2 | K3 | ...
where
 K1 = prf(SKEYID_d, [ g(qm)^xy | ] protocol | SPI | Ni_b | Nr_b)
 K2 = prf(SKEYID_d, K1 | [ g(qm)^xy | ] protocol | SPI | Ni_b |
 Nr_b)
 K3 = prf(SKEYID_d, K2 | [ g(qm)^xy | ] protocol | SPI | Ni_b |
 Nr_b)
 etc.
```

This keying material (whether with PFS or without, and whether derived directly or through concatenation) MUST be used with the negotiated SA. It is up to the service to define how keys are derived from the keying material.

In the case of an ephemeral Diffie-Hellman exchange in Quick Mode, the exponential (g(qm)^xy) is irretreivably removed from the current state and SKEYID_e and SKEYID_a (derived from phase 1 negotiation) continue to protect and authenticate the ISAKMP SA and SKEYID_d continues to be used to derive keys.

Using Quick Mode, multiple SA's and keys can be negotiated with one exchange as follows:

```
 Initiator                Responder
-----------               -----------
HDR*, HASH(1), SA0, SA1, Ni,
  [, KE ] [, IDci, IDcr ] -->
                    <--   HDR*, HASH(2), SA0, SA1, Nr,
                          [, KE ] [, IDci, IDcr ]
HDR*, HASH(3)             -->
```

The keying material is derived identically as in the case of a single SA. In this case (negotiation of two SA payloads) the result would be four security associations-- two each way for both SAs.

5.6 New Group Mode

New Group Mode MUST NOT be used prior to establishment of an ISAKMP SA. The description of a new group MUST only follow phase 1 negotiation. (It is not a phase 2 exchange, though).

```
   Initiator                Responder
   ----------               ----------
   HDR*, HASH(1), SA      -->
                  <--   HDR*, HASH(2), SA
```

where HASH(1) is the prf output, using SKEYID_a as the key, and the message-ID from the ISAKMP header concatenated with the entire SA proposal, body and header, as the data; HASH(2) is the prf output, using SKEYID_a as the key, and the message-ID from the ISAKMP header concatenated with the reply as the data. In other words the hashes for the above exchange are:

```
HASH(1) = prf(SKEYID_a, M-ID | SA)
HASH(2) = prf(SKEYID_a, M-ID | SA)
```

The proposal will specify the characteristics of the group (see appendix A, "Attribute Assigned Numbers"). Group descriptions for private Groups MUST be greater than or equal to 2^15. If the group is not acceptable, the responder MUST reply with a Notify payload with the message type set to ATTRIBUTES-NOT-SUPPORTED (13).

ISAKMP implementations MAY require private groups to expire with the SA under which they were established.

Groups may be directly negotiated in the SA proposal with Main Mode. To do this the component parts-- for a MODP group, the type, prime and generator; for a EC2N group the type, the Irreducible Polynomial, Group Generator One, Group Generator Two, Group Curve A, Group Curve B and Group Order-- are passed as SA attributes (see Appendix A). Alternately, the nature of the group can be hidden using New Group Mode and only the group identifier is passed in the clear during phase 1 negotiation.

5.7 ISAKMP Informational Exchanges

This protocol protects ISAKMP Informational Exchanges when possible. Once the ISAKMP security association has been established (and SKEYID_e and SKEYID_a have been generated) ISAKMP Information Exchanges, when used with this protocol, are as follows:

Initiator		Responder
HDR*, HASH(1), N/D	-->	

where N/D is either an ISAKMP Notify Payload or an ISAKMP Delete Payload and HASH(1) is the prf output, using SKEYID_a as the key, and a M-ID unique to this exchange concatenated with the entire informational payload (either a Notify or Delete) as the data. In other words, the hash for the above exchange is:

HASH(1) = prf(SKEYID_a, M-ID | N/D)

As noted the message ID in the ISAKMP header-- and used in the prf computation-- is unique to this exchange and MUST NOT be the same as the message ID of another phase 2 exchange which generated this informational exchange. The derivation of the initialization vector, used with SKEYID_e to encrypt this message, is described in Appendix B.

If the ISAKMP security association has not yet been established at the time of the Informational Exchange, the exchange is done in the clear without an accompanying HASH payload.

6 Oakley Groups

With IKE, the group in which to do the Diffie-Hellman exchange is negotiated. Four groups-- values 1 through 4-- are defined below. These groups originated with the Oakley protocol and are therefore called "Oakley Groups". The attribute class for "Group" is defined in Appendix A. All values 2^15 and higher are used for private group identifiers. For a discussion on the strength of the default Oakley groups please see the Security Considerations section below.

These groups were all generated by Richard Schroeppel at the University of Arizona. Properties of these groups are described in [Orm96].

6.1 First Oakley Default Group

Oakley implementations MUST support a MODP group with the following prime and generator. This group is assigned id 1 (one).

The prime is: 2^768 - 2 ^704 - 1 + 2^64 * { [2^638 pi] + 149686 }
Its hexadecimal value is

FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1
29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD
EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245
E485B576 625E7EC6 F44C42E9 A63A3620 FFFFFFFF FFFFFFFF

The generator is: 2.

6.2 Second Oakley Group

IKE implementations SHOULD support a MODP group with the following prime and generator. This group is assigned id 2 (two).

The prime is 2^1024 - 2^960 - 1 + 2^64 * { [2^894 pi] + 129093 }.
Its hexadecimal value is

FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1
29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD
EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245
E485B576 625E7EC6 F44C42E9 A637ED6B 0BFF5CB6 F406B7ED
EE386BFB 5A899FA5 AE9F2411 7C4B1FE6 49286651 ECE65381
FFFFFFFF FFFFFFFF

The generator is 2 (decimal)

6.3 Third Oakley Group

IKE implementations SHOULD support a EC2N group with the following characteristics. This group is assigned id 3 (three). The curve is based on the Galois Field GF[2^155]. The field size is 155. The irreducible polynomial for the field is:
u^155 + u^62 + 1.
The equation for the elliptic curve is:
y^2 + xy = x^3 + ax^2 + b.

Field Size: 155
Group Prime/Irreducible Polynomial:
0x080000000000000000000004000000000000001
Group Generator One: 0x7b
Group Curve A: 0x0
Group Curve B: 0x07338f

Group Order: 0X080000000000000000057db5698537193aef944

The data in the KE payload when using this group is the value x from the solution (x,y), the point on the curve chosen by taking the randomly chosen secret Ka and computing Ka*P, where * is the repetition of the group addition and double operations, P is the curve point with x coordinate equal to generator 1 and the y

coordinate determined from the defining equation. The equation of curve is implicitly known by the Group Type and the A and B coefficients. There are two possible values for the y coordinate; either one can be used successfully (the two parties need not agree on the selection).

6.4 Fourth Oakley Group

IKE implementations SHOULD support a EC2N group with the following characteristics. This group is assigned id 4 (four). The curve is based on the Galois Field GF[2^185]. The field size is 185. The irreducible polynomial for the field is:

u^185 + u^69 + 1. The

equation for the elliptic curve is:

y^2 + xy = x^3 + ax^2 + b.

Field Size: 185
Group Prime/Irreducible Polynomial:
0x020000000000000000000000000020000000000000000001
Group Generator One: 0x18
Group Curve A: 0x0
Group Curve B: 0x1ee9

Group Order: 0X01ffffffffffffffffffffdbf2f889b73e484175f94ebc

The data in the KE payload when using this group will be identical to that as when using Oakley Group 3 (three).

Other groups can be defined using New Group Mode. These default groups were generated by Richard Schroeppel at the University of Arizona. Properties of these primes are described in [Orm96].

7. Payload Explosion for a Complete IKE Exchange

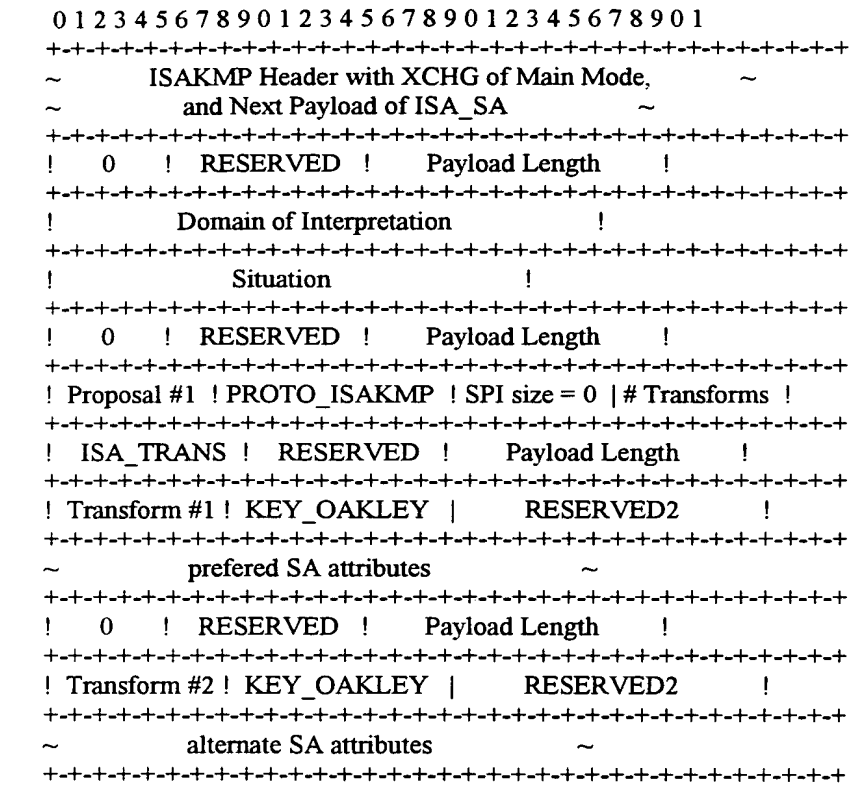
This section illustrates how the IKE protocol is used to:

- establish a secure and authenticated channel between ISAKMP processes (phase 1); and

- generate key material for, and negotiate, an IPsec SA (phase 2).

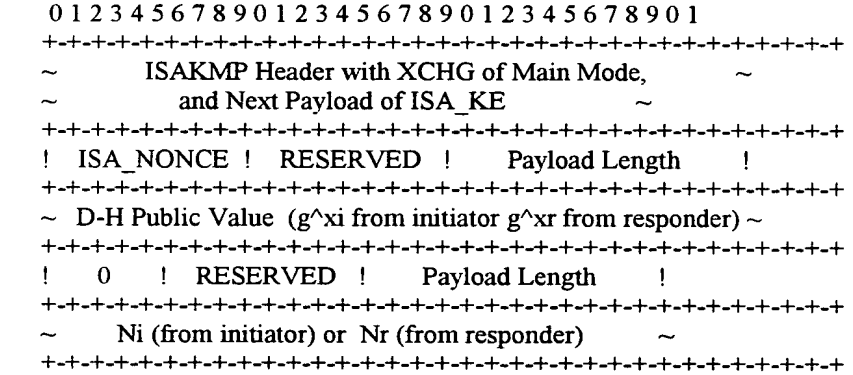
7.1 Phase 1 using Main Mode

The following diagram illustrates the payloads exchanged between the two parties in the first round trip exchange. The initiator MAY propose several proposals; the responder MUST reply with one.

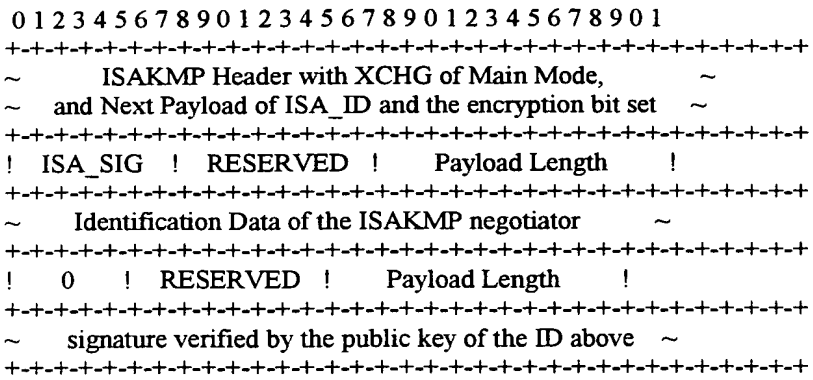


The responder replies in kind but selects, and returns, one transform proposal (the ISAKMP SA attributes).

The second exchange consists of the following payloads:



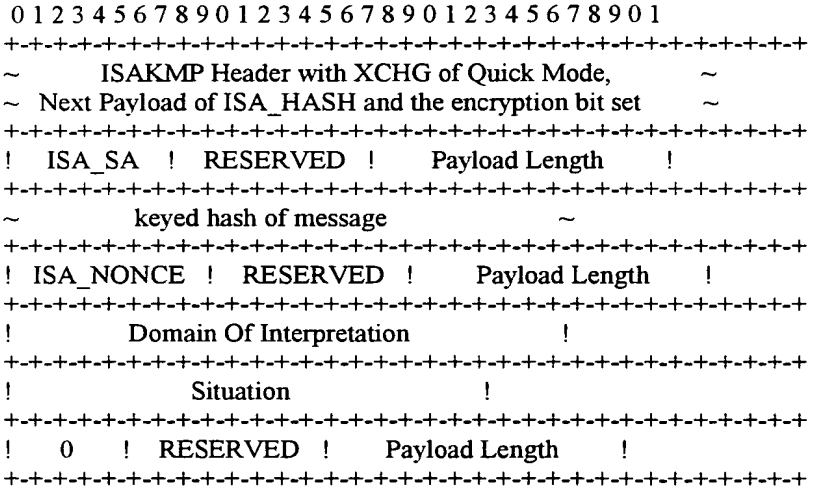
The shared keys, SKEYID_e and SKEYID_a, are now used to protect and authenticate all further communication. Note that both SKEYID_e and SKEYID_a are unauthenticated.



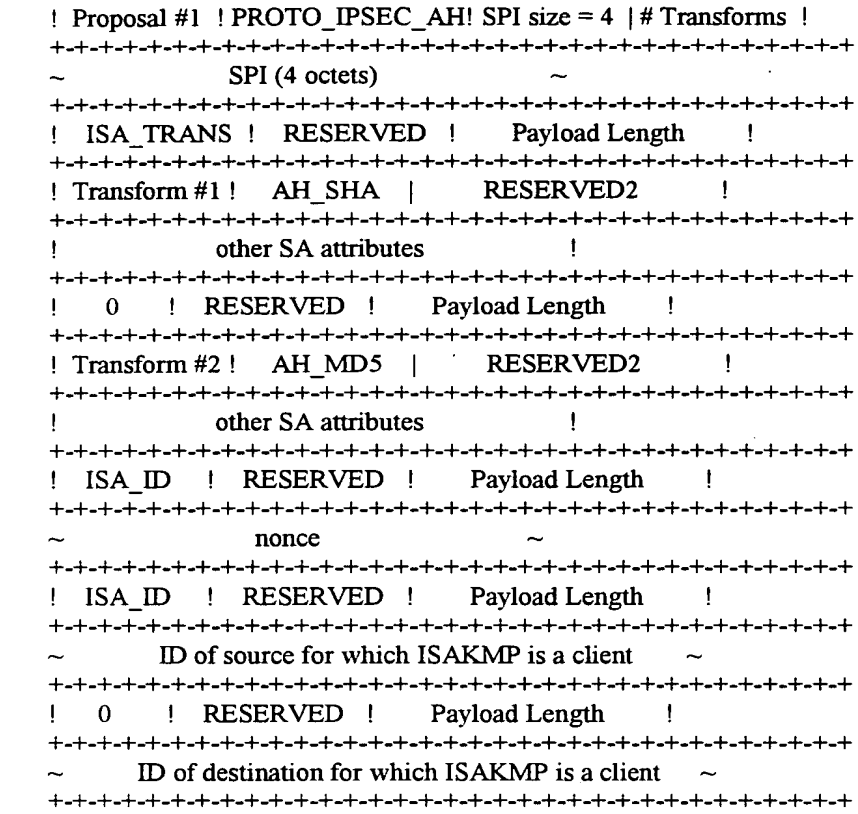
The key exchange is authenticated over a signed hash as described in section 5.1. Once the signature has been verified using the authentication algorithm negotiated as part of the ISAKMP SA, the shared keys, SKEYID_e and SKEYID_a can be marked as authenticated. (For brevity, certificate payloads were not exchanged).

7.2 Phase 2 using Quick Mode

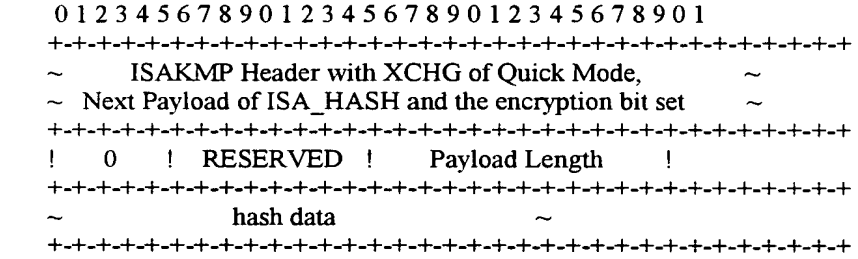
The following payloads are exchanged in the first round of Quick Mode with ISAKMP SA negotiation. In this hypothetical exchange, the ISAKMP negotiators are proxies for other parties which have requested authentication.



Meta 1002
p. 207



where the contents of the hash are described in 5.5 above. The responder replies with a similar message which only contains one transform-- the selected AH transform. Upon receipt, the initiator can provide the key engine with the negotiated security association and the keying material. As a check against replay attacks, the responder waits until receipt of the next message.



where the contents of the hash are described in 5.5 above.

Meta 1002
p. 208

8. Perfect Forward Secrecy Example

This protocol can provide PFS of both keys and identities. The identies of both the ISAKMP negotiating peer and, if applicable, the identities for whom the peers are negotiating can be protected with PFS.

To provide Perfect Forward Secrecy of both keys and all identities, two parties would perform the following:

- o A Main Mode Exchange to protect the identities of the ISAKMP peers.
  This establishes an ISAKMP SA.
- o A Quick Mode Exchange to negotiate other security protocol protection.
  This establishes a SA on each end for this protocol.
- o Delete the ISAKMP SA and its associated state.

Since the key for use in the non-ISAKMP SA was derived from the single ephemeral Diffie-Hellman exchange PFS is preserved.

To provide Perfect Forward Secrecy of merely the keys of a non-ISAKMP security association, it in not necessary to do a phase 1 exchange if an ISAKMP SA exists between the two peers. A single Quick Mode in which the optional KE payload is passed, and an additional Diffie-Hellman exchange is performed, is all that is required. At this point the state derived from this Quick Mode must be deleted from the ISAKMP SA as described in section 5.5.

9. Implementation Hints

Using a single ISAKMP Phase 1 negotiation makes subsequent Phase 2 negotiations extremely quick. As long as the Phase 1 state remains cached, and PFS is not needed, Phase 2 can proceed without any exponentiation. How many Phase 2 negotiations can be performed for a single Phase 1 is a local policy issue. The decision will depend on the strength of the algorithms being used and level of trust in the peer system.

An implementation may wish to negotiate a range of SAs when performing Quick Mode. By doing this they can speed up the "re-keying". Quick Mode defines how KEYMAT is defined for a range of SAs. When one peer feels it is time to change SAs they simply use the next one within the stated range. A range of SAs can be established by negotiating multiple SAs (identical attributes, different SPIs) with one Quick Mode.

An optimization that is often useful is to establish Security Associations with peers before they are needed so that when they become needed they are already in place. This ensures there would be no delays due to key management before initial data transmission. This optimization is easily implemented by setting up more than one Security Association with a peer for each requested Security Association and caching those not immediately used.

Also, if an ISAKMP implementation is alerted that a SA will soon be needed (e.g. to replace an existing SA that will expire in the near future), then it can establish the new SA before that new SA is needed.

The base ISAKMP specification describes conditions in which one party of the protocol may inform the other party of some activity-- either deletion of a security association or in response to some error in the protocol such as a signature verification failed or a payload failed to decrypt. It is strongly suggested that these Informational exchanges not be responded to under any circumstances. Such a condition may result in a "notify war" in which failure to understand a message results in a notify to the peer who cannot understand it and sends his own notify back which is also not understood.

10. Security Considerations

This entire memo discusses a hybrid protocol, combining parts of Oakley and parts of SKEME with ISAKMP, to negotiate, and derive keying material for, security associations in a secure and authenticated manner.

Confidentiality is assured by the use of a negotiated encryption algorithm. Authentication is assured by the use of a negotiated method: a digital signature algorithm; a public key algorithm which supports encryption; or, a pre-shared key. The confidentiality and authentication of this exchange is only as good as the attributes negotiated as part of the ISAKMP security association.

Repeated re-keying using Quick Mode can consume the entropy of the Diffie-Hellman shared secret. Implementors should take note of this fact and set a limit on Quick Mode Exchanges between exponentiations. This memo does not prescribe such a limit.

Perfect Forward Secrecy (PFS) of both keying material and identities is possible with this protocol. By specifying a Diffie-Hellman group, and passing public values in KE payloads, ISAKMP peers can establish PFS of keys-- the identities would be protected by SKEYID_e from the ISAKMP SA and would therefore not be protected by PFS. If PFS of both keying material and identities is desired, an ISAKMP peer MUST

establish only one non-ISAKMP security association (e.g. IPsec Security Association) per ISAKMP SA. PFS for keys and identities is accomplished by deleting the ISAKMP SA (and optionally issuing a DELETE message) upon establishment of the single non-ISAKMP SA. In this way a phase one negotiation is uniquely tied to a single phase two negotiation, and the ISAKMP SA established during phase one negotiation is never used again.

The strength of a key derived from a Diffie-Hellman exchange using any of the groups defined here depends on the inherent strength of the group, the size of the exponent used, and the entropy provided by the random number generator used. Due to these inputs it is difficult to determine the strength of a key for any of the defined groups. The default Diffie-Hellman group (number one) when used with a strong random number generator and an exponent no less than 160 bits is sufficient to use for DES. Groups two through four provide greater security. Implementations should make note of these conservative estimates when establishing policy and negotiating security parameters.

Note that these limitations are on the Diffie-Hellman groups themselves. There is nothing in IKE which prohibits using stronger groups nor is there anything which will dilute the strength obtained from stronger groups. In fact, the extensible framework of IKE encourages the definition of more groups; use of elliptical curve groups will greatly increase strength using much smaller numbers.

For situations where defined groups provide insufficient strength New Group Mode can be used to exchange a Diffie-Hellman group which provides the necessary strength. In is incumbent upon implementations to check the primality in groups being offered and independently arrive at strength estimates.

It is assumed that the Diffie-Hellman exponents in this exchange are erased from memory after use. In particular, these exponents must not be derived from long-lived secrets like the seed to a pseudo-random generator.

IKE exchanges maintain running initialization vectors (IV) where the last ciphertext block of the last message is the IV for the next message. To prevent retransmissions (or forged messages with valid cookies) from causing exchanges to get out of sync IKE implementations SHOULD NOT update their running IV until the decrypted message has passed a basic sanity check and has been determined to actually advance the IKE state machine-- i.e. it is not a retransmission.

Meta 1002
p. 211

While the last roundtrip of Main Mode (and optionally the last message of Aggressive Mode) is encrypted it is not, strictly speaking, authenticated. An active substitution attack on the ciphertext could result in payload corruption. If such an attack corrupts mandatory payloads it would be detected by an authentication failure, but if it corrupts any optional payloads (e.g. notify payloads chained onto the last message of a Main Mode exchange) it might not be detectable.

11. IANA Considerations

This document contains many "magic numbers" to be maintained by the IANA. This section explains the criteria to be used by the IANA to assign additional numbers in each of these lists.

11.1 Attribute Classes

Attributes negotiated in this protocol are identified by their class. Requests for assignment of new classes must be accompanied by a standards-track RFC which describes the use of this attribute.

11.2 Encryption Algorithm Class

Values of the Encryption Algorithm Class define an encryption algorithm to use when called for in this document. Requests for assignment of new encryption algorithm values must be accompanied by a reference to a standards-track or Informational RFC or a reference to published cryptographic literature which describes this algorithm.

11.3 Hash Algorithm

Values of the Hash Algorithm Class define a hash algorithm to use when called for in this document. Requests for assignment of new hash algorithm values must be accompanied by a reference to a standards-track or Informational RFC or a reference to published cryptographic literature which describes this algorithm. Due to the key derivation and key expansion uses of HMAC forms of hash algorithms in IKE, requests for assignment of new hash algorithm values must take into account the cryptographic properties-- e.g it's resistance to collision-- of the hash algorithm itself.

11.4 Group Description and Group Type

Values of the Group Description Class identify a group to use in a Diffie-Hellman exchange. Values of the Group Type Class define the type of group. Requests for assignment of new groups must be accompanied by a reference to a standards-track or Informational RFC which describes this group. Requests for assignment of new group

types must be accompanied by a reference to a standards-track or Informational RFC or by a reference to published cryptographic or mathmatical literature which describes the new type.

11.5 Life Type

Values of the Life Type Class define a type of lifetime to which the ISAKMP Security Association applies. Requests for assignment of new life types must be accompanied by a detailed description of the units of this type and its expiry.

12. Acknowledgements

This document is the result of close consultation with Hugo Krawczyk, Douglas Maughan, Hilarie Orman, Mark Schertler, Mark Schneider, and Jeff Turner. It relies on protocols which were written by them. Without their interest and dedication, this would not have been written.

Special thanks Rob Adams, Cheryl Madson, Derrell Piper, Harry Varnis, and Elfed Weaver for technical input, encouragement, and various sanity checks along the way.

We would also like to thank the many members of the IPSec working group that contributed to the development of this protocol over the past year.

13. References

[CAST] Adams, C., "The CAST-128 Encryption Algorithm", RFC 2144, May 1997.

[BLOW] Schneier, B., "The Blowfish Encryption Algorithm", Dr. Dobb's Journal, v. 19, n. 4, April 1994.

[Bra97] Bradner, S., "Key Words for use in RFCs to indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[DES] ANSI X3.106, "American National Standard for Information Systems-Data Link Encryption", American National Standards Institute, 1983.

[DH] Diffie, W., and Hellman M., "New Directions in Cryptography", IEEE Transactions on Information Theory, V. IT-22, n. 6, June 1977.

[DSS] NIST, "Digital Signature Standard", FIPS 186, National Institute of Standards and Technology, U.S. Department of Commerce, May, 1994.

[IDEA] Lai, X., "On the Design and Security of Block Ciphers," ETH Series in Information Processing, v. 1, Konstanz: Hartung-Gorre Verlag, 1992

[KBC96] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[SKEME] Krawczyk, H., "SKEME: A Versatile Secure Key Exchange Mechanism for Internet", from IEEE Proceedings of the 1996 Symposium on Network and Distributed Systems Security.

[MD5] Rivest, R., "The MD5 Message Digest Algorithm", RFC 1321, April 1992.

[MSST98] Maughhan, D., Schertler, M., Schneider, M., and J. Turner, "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, November 1998.

[Orm96] Orman, H., "The Oakley Key Determination Protocol", RFC 2412, November 1998.

[PKCS1] RSA Laboratories, "PKCS #1: RSA Encryption Standard", November 1993.

[Pip98] Piper, D., "The Internet IP Security Domain Of Interpretation for ISAKMP", RFC 2407, November 1998.

[RC5] Rivest, R., "The RC5 Encryption Algorithm", Dr. Dobb's Journal, v. 20, n. 1, January 1995.

[RSA] Rivest, R., Shamir, A., and Adleman, L., "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM, v. 21, n. 2, February 1978.

[Sch96] Schneier, B., "Applied Cryptography, Protocols, Algorithms, and Source Code in C", 2nd edition.

[SHA] NIST, "Secure Hash Standard", FIPS 180-1, National Institue of Standards and Technology, U.S. Department of Commerce, May 1994.

[TIGER] Anderson, R., and Biham, E., "Fast Software Encryption", Springer LNCS v. 1039, 1996.

Meta 1002
p. 214

Appendix A

This is a list of DES Weak and Semi-Weak keys. The keys come from [Sch96]. All keys are listed in hexidecimal.

DES Weak Keys
0101 0101 0101 0101
1F1F 1F1F E0E0 E0E0
E0E0 E0E0 1F1F 1F1F
FEFE FEFE FEFE FEFE

DES Semi-Weak Keys
01FE 01FE 01FE 01FE
1FE0 1FE0 0EF1 0EF1
01E0 01E0 01F1 01F1
1FFE 1FFE 0EFE 0EFE
011F 011F 010E 010E
E0FE E0FE F1FE F1FE

FE01 FE01 FE01 FE01
E01F E01F F10E F10E
E001 E001 F101 F101
FE1F FE1F FE0E FE0E
1F01 1F01 0E01 0E01
FEE0 FEE0 FEF1 FEF1

Attribute Assigned Numbers

Attributes negotiated during phase one use the following definitions. Phase two attributes are defined in the applicable DOI specification (for example, IPsec attributes are defined in the IPsec DOI), with the exception of a group description when Quick Mode includes an ephemeral Diffie-Hellman exchange. Attribute types can be either Basic (B) or Variable-length (V). Encoding of these attributes is defined in the base ISAKMP specification as Type/Value (Basic) and Type/Length/Value (Variable).

Attributes described as basic MUST NOT be encoded as variable. Variable length attributes MAY be encoded as basic attributes if their value can fit into two octets. If this is the case, an attribute offered as variable (or basic) by the initiator of this protocol MAY be returned to the initiator as a basic (or variable).

Attribute Classes

class	value	type
Encryption Algorithm	1	B
Hash Algorithm	2	B
Authentication Method	3	B
Group Description	4	B
Group Type	5	B
Group Prime/Irreducible Polynomial	6	V
Group Generator One	7	V
Group Generator Two	8	V
Group Curve A	9	V
Group Curve B	10	V
Life Type	11	B
Life Duration	12	V
PRF	13	B
Key Length	14	B
Field Size	15	B
Group Order	16	V

values 17-16383 are reserved to IANA. Values 16384-32767 are for private use among mutually consenting parties.

Class Values

- Encryption Algorithm		Defined In
DES-CBC	1	RFC 2405
IDEA-CBC	2	
Blowfish-CBC	3	
RC5-R16-B64-CBC	4	
3DES-CBC	5	
CAST-CBC	6	

values 7-65000 are reserved to IANA. Values 65001-65535 are for private use among mutually consenting parties.

- Hash Algorithm		Defined In
MD5	1	RFC 1321
SHA	2	FIPS 180-1
Tiger	3	See Reference [TIGER]

values 4-65000 are reserved to IANA. Values 65001-65535 are for private use among mutually consenting parties.

- Authentication Method

pre-shared key	1
DSS signatures	2
RSA signatures	3
Encryption with RSA	4
Revised encryption with RSA	5

values 6-65000 are reserved to IANA. Values 65001-65535 are for private use among mutually consenting parties.

- Group Description

default 768-bit MODP group (section 6.1)	1
alternate 1024-bit MODP group (section 6.2)	2
EC2N group on GP[2^155] (section 6.3)	3
EC2N group on GP[2^185] (section 6.4)	4

values 5-32767 are reserved to IANA. Values 32768-65535 are for private use among mutually consenting parties.

- Group Type

MODP (modular exponentiation group)	1
ECP (elliptic curve group over GF[P])	2
EC2N (elliptic curve group over GF[2^N])	3

values 4-65000 are reserved to IANA. Values 65001-65535 are for private use among mutually consenting parties.

- Life Type

seconds	1
kilobytes	2

values 3-65000 are reserved to IANA. Values 65001-65535 are for private use among mutually consenting parties. For a given "Life Type" the value of the "Life Duration" attribute defines the actual length of the SA life-- either a number of seconds, or a number of kbytes protected.

- PRF

There are currently no pseudo-random functions defined.

values 1-65000 are reserved to IANA. Values 65001-65535 are for private use among mutually consenting parties.

- Key Length

When using an Encryption Algorithm that has a variable length key, this attribute specifies the key length in bits. (MUST use network byte order). This attribute MUST NOT be used when the specified Encryption Algorithm uses a fixed length key.

- Field Size

The field size, in bits, of a Diffie-Hellman group.

- Group Order

The group order of an elliptical curve group. Note the length of this attribute depends on the field size.

Additional Exchanges Defined-- XCHG values

Quick Mode	32
New Group Mode	33

Appendix B

This appendix describes encryption details to be used ONLY when encrypting ISAKMP messages. When a service (such as an IPSEC transform) utilizes ISAKMP to generate keying material, all encryption algorithm specific details (such as key and IV generation, padding, etc...) MUST be defined by that service. ISAKMP does not purport to ever produce keys that are suitable for any encryption algorithm. ISAKMP produces the requested amount of keying material from which the service MUST generate a suitable key. Details, such as weak key checks, are the responsibility of the service.

Use of negotiated PRFs may require the PRF output to be expanded due to the PRF feedback mechanism employed by this document. For example, if the (ficticious) DOORAK-MAC requires 24 bytes of key but produces only 8 bytes of output, the output must be expanded three times before being used as the key for another instance of itself. The output of a PRF is expanded by feeding back the results of the PRF into itself to generate successive blocks. These blocks are concatenated until the requisite number of bytes has been acheived. For example, for pre-shared key authentication with DOORAK-MAC as the negotiated PRF:

```
 BLOCK1-8 = prf(pre-shared-key, Ni_b | Nr_b)
 BLOCK9-16 = prf(pre-shared-key, BLOCK1-8 | Ni_b | Nr_b)
 BLOCK17-24 = prf(pre-shared-key, BLOCK9-16 | Ni_b | Nr_b)
and
 SKEYID = BLOCK1-8 | BLOCK9-16 | BLOCK17-24
```

so therefore to derive SKEYID_d:

```
 BLOCK1-8 = prf(SKEYID, g^xy | CKY-I | CKY-R | 0)
 BLOCK9-16 = prf(SKEYID, BLOCK1-8 | g^xy | CKY-I | CKY-R | 0)
 BLOCK17-24 = prf(SKEYID, BLOCK9-16 | g^xy | CKY-I | CKY-R | 0)
and
 SKEYID_d = BLOCK1-8 | BLOCK9-16 | BLOCK17-24
```

Subsequent PRF derivations are done similarly.

Encryption keys used to protect the ISAKMP SA are derived from SKEYID_e in an algorithm-specific manner. When SKEYID_e is not long enough to supply all the necessary keying material an algorithm requires, the key is derived from feeding the results of a pseudo-random function into itself, concatenating the results, and taking the highest necessary bits.

For example, if (ficticious) algorithm AKULA requires 320-bits of key (and has no weak key check) and the prf used to generate SKEYID_e only generates 120 bits of material, the key for AKULA, would be the first 320-bits of Ka, where:

Ka = K1 | K2 | K3
and
K1 = prf(SKEYID_e, 0)
K2 = prf(SKEYID_e, K1)
K3 = prf(SKEYID_e, K2)

where prf is the negotiated prf or the HMAC version of the negotiated hash function (if no prf was negotiated) and 0 is represented by a single octet. Each result of the prf provides 120 bits of material for a total of 360 bits. AKULA would use the first 320 bits of that 360 bit string.

In phase 1, material for the initialization vector (IV material) for CBC mode encryption algorithms is derived from a hash of a concatenation of the initiator's public Diffie-Hellman value and the responder's public Diffie-Hellman value using the negotiated hash algorithm. This is used for the first message only. Each message should be padded up to the nearest block size using bytes containing 0x00. The message length in the header MUST include the length of the pad since this reflects the size of the ciphertext. Subsequent messages MUST use the last CBC encryption block from the previous message as their initialization vector.

In phase 2, material for the initialization vector for CBC mode encryption of the first message of a Quick Mode exchange is derived from a hash of a concatenation of the last phase 1 CBC output block and the phase 2 message id using the negotiated hash algorithm. The IV for subsequent messages within a Quick Mode exchange is the CBC output block from the previous message. Padding and IVs for subsequent messages are done as in phase 1.

After the ISAKMP SA has been authenticated all Informational Exchanges are encrypted using SKEYID_e. The initiaization vector for these exchanges is derived in exactly the same fashion as that for a Quick Mode-- i.e. it is derived from a hash of a concatenation of the last phase 1 CBC output block and the message id from the ISAKMP header of the Informational Exchange (not the message id from the message that may have prompted the Informational Exchange).

Note that the final phase 1 CBC output block, the result of encryption/decryption of the last phase 1 message, must be retained in the ISAKMP SA state to allow for generation of unique IVs for each Quick Mode. Each post- phase 1 exchange (Quick Modes and

Informational Exchanges) generates IVs independantly to prevent IVs from getting out of sync when two different exchanges are started simultaneously.

In all cases, there is a single bidirectional cipher/IV context. Having each Quick Mode and Informational Exchange maintain a unique context prevents IVs from getting out of sync.

The key for DES-CBC is derived from the first eight (8) non-weak and non-semi-weak (see Appendix A) bytes of SKEYID_e. The IV is the first 8 bytes of the IV material derived above.

The key for IDEA-CBC is derived from the first sixteen (16) bytes of SKEYID_e. The IV is the first eight (8) bytes of the IV material derived above.

The key for Blowfish-CBC is either the negotiated key size, or the first fifty-six (56) bytes of a key (if no key size is negotiated) derived in the aforementioned pseudo-random function feedback method. The IV is the first eight (8) bytes of the IV material derived above.

The key for RC5-R16-B64-CBC is the negotiated key size, or the first sixteen (16) bytes of a key (if no key size is negotiated) derived from the aforementioned pseudo-random function feedback method if necessary. The IV is the first eight (8) bytes of the IV material derived above. The number of rounds MUST be 16 and the block size MUST be 64.

The key for 3DES-CBC is the first twenty-four (24) bytes of a key derived in the aforementioned pseudo-random function feedback method. 3DES-CBC is an encrypt-decrypt-encrypt operation using the first, middle, and last eight (8) bytes of the entire 3DES-CBC key. The IV is the first eight (8) bytes of the IV material derived above.

The key for CAST-CBC is either the negotiated key size, or the first sixteen (16) bytes of a key derived in the aforementioned pseudo-random function feedback method. The IV is the first eight (8) bytes of the IV material derived above.

Support for algorithms other than DES-CBC is purely optional. Some optional algorithms may be subject to intellectual property claims.

Authors' Addresses

Dan Harkins
cisco Systems
170 W. Tasman Dr.
San Jose, California, 95134-1706
United States of America

Phone: +1 408 526 4000
EMail: dharkins@cisco.com

Dave Carrel
76 Lippard Ave.
San Francisco, CA 94131-2947
United States of America

Phone: +1 415 337 8469
EMail: carrel@ipsec.org

Authors' Note

The authors encourage independent implementation, and interoperability testing, of this hybrid protocol.

Meta 1002
p. 222

Full Copyright Statement

Copyright (C) The Internet Society (1998). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Network Working Group
Request for Comments: 2402
Obsoletes: 1826
Category: Standards Track

S. Kent
BBN Corp
R. Atkinson
@Home Network
November 1998

IP Authentication Header

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1998). All Rights Reserved.

Table of Contents

1. Introduction 2
2. Authentication Header Format 3
2.1 Next Header 4
2.2 Payload Length 4
2.3 Reserved 4
2.4 Security Parameters Index (SPI) 4
2.5 Sequence Number 5
2.6 Authentication Data 5
3. Authentication Header Processing 5
3.1 Authentication Header Location 5
3.2 Authentication Algorithms 7
3.3 Outbound Packet Processing 8
3.3.1 Security Association Lookup 8
3.3.2 Sequence Number Generation 8
3.3.3 Integrity Check Value Calculation 9
3.3.3.1 Handling Mutable Fields 9
3.3.3.1.1 ICV Computation for IPv4 10
3.3.3.1.1.1 Base Header Fields 10
3.3.3.1.1.2 Options 11
3.3.3.1.2 ICV Computation for IPv6 11
3.3.3.1.2.1 Base Header Fields 11
3.3.3.1.2.2 Extension Headers Containing Options 11
3.3.3.1.2.3 Extension Headers Not Containing Options 11
3.3.3.2 Padding 12
3.3.3.2.1 Authentication Data Padding 12

Meta 1002
p. 224

3.3.3.2.2 Implicit Packet Padding 12
3.3.4 Fragmentation 12
3.4 Inbound Packet Processing 13
3.4.1 Reassembly 13
3.4.2 Security Association Lookup 13
3.4.3 Sequence Number Verification 13
3.4.4 Integrity Check Value Verification 15
4. Auditing 15
5. Conformance Requirements 16
6. Security Considerations 16
7. Differences from RFC 1826 16
Acknowledgements 17
Appendix A -- Mutability of IP Options/Extension Headers 18
A1. IPv4 Options 18
A2. IPv6 Extension Headers 19
References 20
Disclaimer 21
Author Information 22
Full Copyright Statement 22

1. Introduction

The IP Authentication Header (AH) is used to provide connectionless integrity and data origin authentication for IP datagrams (hereafter referred to as just "authentication"), and to provide protection against replays. This latter, optional service may be selected, by the receiver, when a Security Association is established. (Although the default calls for the sender to increment the Sequence Number used for anti-replay, the service is effective only if the receiver checks the Sequence Number.) AH provides authentication for as much of the IP header as possible, as well as for upper level protocol data. However, some IP header fields may change in transit and the value of these fields, when the packet arrives at the receiver, may not be predictable by the sender. The values of such fields cannot be protected by AH. Thus the protection provided to the IP header by AH is somewhat piecemeal.

AH may be applied alone, in combination with the IP Encapsulating Security Payload (ESP) [KA97b], or in a nested fashion through the use of tunnel mode (see "Security Architecture for the Internet Protocol" [KA97a], hereafter referred to as the Security Architecture document). Security services can be provided between a pair of communicating hosts, between a pair of communicating security gateways, or between a security gateway and a host. ESP may be used to provide the same security services, and it also provides a confidentiality (encryption) service. The primary difference between the authentication provided by ESP and AH is the extent of the coverage. Specifically, ESP does not protect any IP header fields

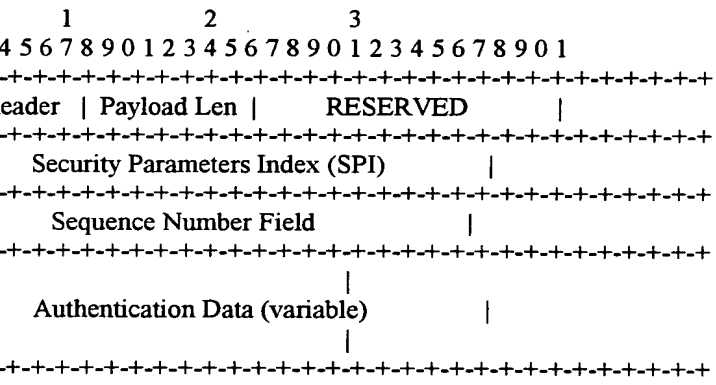
unless those fields are encapsulated by ESP (tunnel mode). For more details on how to use AH and ESP in various network environments, see the Security Architecture document [KA97a].

It is assumed that the reader is familiar with the terms and concepts described in the Security Architecture document. In particular, the reader should be familiar with the definitions of security services offered by AH and ESP, the concept of Security Associations, the ways in which AH can be used in conjunction with ESP, and the different key management options available for AH and ESP. (With regard to the last topic, the current key management options required for both AH and ESP are manual keying and automated keying via IKE [HC98].)

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in RFC 2119 [Bra97].

2. Authentication Header Format

The protocol header (IPv4, IPv6, or Extension) immediately preceding the AH header will contain the value 51 in its Protocol (IPv4) or Next Header (IPv6, Extension) field [STD-2].



The following subsections define the fields that comprise the AH format. All the fields described here are mandatory, i.e., they are always present in the AH format and are included in the Integrity Check Value (ICV) computation (see Sections 2.6 and 3.3.3).

2.1 Next Header

The Next Header is an 8-bit field that identifies the type of the next payload after the Authentication Header. The value of this field is chosen from the set of IP Protocol Numbers defined in the most recent "Assigned Numbers" [STD-2] RFC from the Internet Assigned Numbers Authority (IANA).

2.2 Payload Length

This 8-bit field specifies the length of AH in 32-bit words (4-byte units), minus "2". (All IPv6 extension headers, as per RFC 1883, encode the "Hdr Ext Len" field by first subtracting 1 (64-bit word) from the header length (measured in 64-bit words). AH is an IPv6 extension header. However, since its length is measured in 32-bit words, the "Payload Length" is calculated by subtracting 2 (32 bit words).) In the "standard" case of a 96-bit authentication value plus the 3 32-bit word fixed portion, this length field will be "4". A "null" authentication algorithm may be used only for debugging purposes. Its use would result in a "1" value for this field for IPv4 or a "2" for IPv6, as there would be no corresponding Authentication Data field (see Section 3.3.3.2.1 on "Authentication Data Padding").

2.3 Reserved

This 16-bit field is reserved for future use. It MUST be set to "zero." (Note that the value is included in the Authentication Data calculation, but is otherwise ignored by the recipient.)

2.4 Security Parameters Index (SPI)

The SPI is an arbitrary 32-bit value that, in combination with the destination IP address and security protocol (AH), uniquely identifies the Security Association for this datagram. The set of SPI values in the range 1 through 255 are reserved by the Internet Assigned Numbers Authority (IANA) for future use; a reserved SPI value will not normally be assigned by IANA unless the use of the assigned SPI value is specified in an RFC. It is ordinarily selected by the destination system upon establishment of an SA (see the Security Architecture document for more details).

The SPI value of zero (0) is reserved for local, implementation-specific use and MUST NOT be sent on the wire. For example, a key management implementation MAY use the zero SPI value to mean "No Security Association Exists" during the period when the IPsec implementation has requested that its key management entity establish a new SA, but the SA has not yet been established.

Meta 1002
p. 227

2.5 Sequence Number

This unsigned 32-bit field contains a monotonically increasing counter value (sequence number). It is mandatory and is always present even if the receiver does not elect to enable the anti-replay service for a specific SA. Processing of the Sequence Number field is at the discretion of the receiver, i.e., the sender MUST always transmit this field, but the receiver need not act upon it (see the discussion of Sequence Number Verification in the "Inbound Packet Processing" section below).

The sender's counter and the receiver's counter are initialized to 0 when an SA is established. (The first packet sent using a given SA will have a Sequence Number of 1; see Section 3.3.2 for more details on how the Sequence Number is generated.) If anti-replay is enabled (the default), the transmitted Sequence Number must never be allowed to cycle. Thus, the sender's counter and the receiver's counter MUST be reset (by establishing a new SA and thus a new key) prior to the transmission of the 2^32nd packet on an SA.

2.6 Authentication Data

This is a variable-length field that contains the Integrity Check Value (ICV) for this packet. The field must be an integral multiple of 32 bits in length. The details of the ICV computation are described in Section 3.3.2 below. This field may include explicit padding. This padding is included to ensure that the length of the AH header is an integral multiple of 32 bits (IPv4) or 64 bits (IPv6). All implementations MUST support such padding. Details of how to compute the required padding length are provided below. The authentication algorithm specification MUST specify the length of the ICV and the comparison rules and processing steps for validation.

3. Authentication Header Processing

3.1 Authentication Header Location

Like ESP, AH may be employed in two ways: transport mode or tunnel mode. The former mode is applicable only to host implementations and provides protection for upper layer protocols, in addition to selected IP header fields. (In this mode, note that for "bump-in-the-stack" or "bump-in-the-wire" implementations, as defined in the Security Architecture document, inbound and outbound IP fragments may require an IPsec implementation to perform extra IP reassembly/fragmentation in order to both conform to this specification and provide transparent IPsec support. Special care is required to perform such operations within these implementations when multiple interfaces are in use.)

In transport mode, AH is inserted after the IP header and before an upper layer protocol, e.g., TCP, UDP, ICMP, etc. or before any other IPsec headers that have already been inserted. In the context of IPv4, this calls for placing AH after the IP header (and any options that it contains), but before the upper layer protocol. (Note that the term "transport" mode should not be misconstrued as restricting its use to TCP and UDP. For example, an ICMP message MAY be sent using either "transport" mode or "tunnel" mode.) The following diagram illustrates AH transport mode positioning for a typical IPv4 packet, on a "before and after" basis.

```
          BEFORE APPLYING AH
        ------------------------
IPv4    |orig IP hdr |    |     |
        |(any options)| TCP | Data |
        ------------------------

          AFTER APPLYING AH
        ---------------------------------
IPv4    |orig IP hdr |    |    |     |
        |(any options)| AH | TCP | Data |
        ---------------------------------
        |<------- authenticated ------->|
           except for mutable fields
```

In the IPv6 context, AH is viewed as an end-to-end payload, and thus should appear after hop-by-hop, routing, and fragmentation extension headers. The destination options extension header(s) could appear either before or after the AH header depending on the semantics desired. The following diagram illustrates AH transport mode positioning for a typical IPv6 packet.

```
              BEFORE APPLYING AH
            ------------------------------------
IPv6    |           | ext hdrs |    |     |
        | orig IP hdr |if present| TCP | Data |
        ------------------------------------

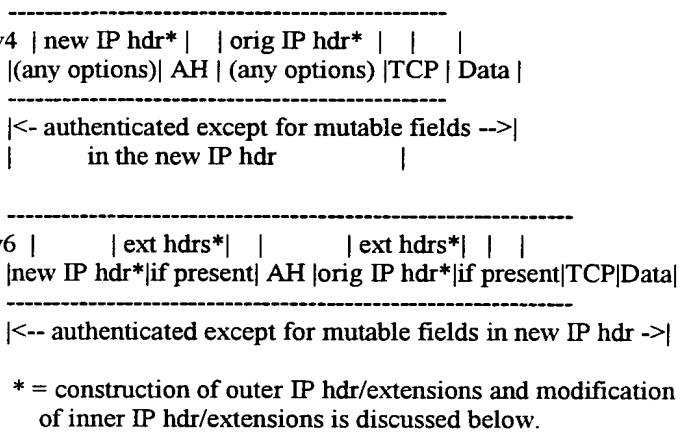
              AFTER APPLYING AH
            ------------------------------------------------------
IPv6    |           |hop-by-hop, dest*, |    | dest |    |     |
        |orig IP hdr  |routing, fragment. | AH | opt* | TCP | Data |
        ------------------------------------------------------
        |<---- authenticated except for mutable fields ---------->|

           * = if present, could be before AH, after AH, or both
```


Meta 1002
p. 229

ESP and AH headers can be combined in a variety of modes. The IPsec Architecture document describes the combinations of security associations that must be supported.

Tunnel mode AH may be employed in either hosts or security gateways (or in so-called "bump-in-the-stack" or "bump-in-the-wire" implementations, as defined in the Security Architecture document). When AH is implemented in a security gateway (to protect transit traffic), tunnel mode must be used. In tunnel mode, the "inner" IP header carries the ultimate source and destination addresses, while an "outer" IP header may contain distinct IP addresses, e.g., addresses of security gateways. In tunnel mode, AH protects the entire inner IP packet, including the entire inner IP header. The position of AH in tunnel mode, relative to the outer IP header, is the same as for AH in transport mode. The following diagram illustrates AH tunnel mode positioning for typical IPv4 and IPv6 packets.



* = construction of outer IP hdr/extensions and modification of inner IP hdr/extensions is discussed below.

3.2 Authentication Algorithms

The authentication algorithm employed for the ICV computation is specified by the SA. For point-to-point communication, suitable authentication algorithms include keyed Message Authentication Codes (MACs) based on symmetric encryption algorithms (e.g., DES) or on one-way hash functions (e.g., MD5 or SHA-1). For multicast communication, one-way hash algorithms combined with asymmetric signature algorithms are appropriate, though performance and space considerations currently preclude use of such algorithms. The mandatory-to-implement authentication algorithms are described in Section 5 "Conformance Requirements". Other algorithms MAY be supported.

Meta 1002
p. 230

3.3 Outbound Packet Processing

In transport mode, the sender inserts the AH header after the IP header and before an upper layer protocol header, as described above. In tunnel mode, the outer and inner IP header/extensions can be inter-related in a variety of ways. The construction of the outer IP header/extensions during the encapsulation process is described in the Security Architecture document.

If there is more than one IPsec header/extension required, the order of the application of the security headers MUST be defined by security policy. For simplicity of processing, each IPsec header SHOULD ignore the existence (i.e., not zero the contents or try to predict the contents) of IPsec headers to be applied later. (While a native IP or bump-in-the-stack implementation could predict the contents of later IPsec headers that it applies itself, it won't be possible for it to predict any IPsec headers added by a bump-in-the-wire implementation between the host and the network.)

3.3.1 Security Association Lookup

AH is applied to an outbound packet only after an IPsec implementation determines that the packet is associated with an SA that calls for AH processing. The process of determining what, if any, IPsec processing is applied to outbound traffic is described in the Security Architecture document.

3.3.2 Sequence Number Generation

The sender's counter is initialized to 0 when an SA is established. The sender increments the Sequence Number for this SA and inserts the new value into the Sequence Number Field. Thus the first packet sent using a given SA will have a Sequence Number of 1.

If anti-replay is enabled (the default), the sender checks to ensure that the counter has not cycled before inserting the new value in the Sequence Number field. In other words, the sender MUST NOT send a packet on an SA if doing so would cause the Sequence Number to cycle. An attempt to transmit a packet that would result in Sequence Number overflow is an auditable event. (Note that this approach to Sequence Number management does not require use of modular arithmetic.)

The sender assumes anti-replay is enabled as a default, unless otherwise notified by the receiver (see 3.4.3). Thus, if the counter has cycled, the sender will set up a new SA and key (unless the SA was configured with manual key management).

Meta 1002
p. 231

If anti-replay is disabled, the sender does not need to monitor or reset the counter, e.g., in the case of manual key management (see Section 5.) However, the sender still increments the counter and when it reaches the maximum value, the counter rolls over back to zero.

3.3.3 Integrity Check Value Calculation

The AH ICV is computed over:

- o IP header fields that are either immutable in transit or that are predictable in value upon arrival at the endpoint for the AH SA
- o the AH header (Next Header, Payload Len, Reserved, SPI, Sequence Number, and the Authentication Data (which is set to zero for this computation), and explicit padding bytes (if any))
- o the upper level protocol data, which is assumed to be immutable in transit

3.3.3.1 Handling Mutable Fields

If a field may be modified during transit, the value of the field is set to zero for purposes of the ICV computation. If a field is mutable, but its value at the (IPsec) receiver is predictable, then that value is inserted into the field for purposes of the ICV calculation. The Authentication Data field is also set to zero in preparation for this computation. Note that by replacing each field's value with zero, rather than omitting the field, alignment is preserved for the ICV calculation. Also, the zero-fill approach ensures that the length of the fields that are so handled cannot be changed during transit, even though their contents are not explicitly covered by the ICV.

As a new extension header or IPv4 option is created, it will be defined in its own RFC and SHOULD include (in the Security Considerations section) directions for how it should be handled when calculating the AH ICV. If the IP (v4 or v6) implementation encounters an extension header that it does not recognize, it will discard the packet and send an ICMP message. IPsec will never see the packet. If the IPsec implementation encounters an IPv4 option that it does not recognize, it should zero the whole option, using the second byte of the option as the length. IPv6 options (in Destination extension headers or Hop by Hop extension header) contain a flag indicating mutability, which determines appropriate processing for such options.

Meta 1002
p. 232

3.3.3.1.1 ICV Computation for IPv4

3.3.3.1.1.1 Base Header Fields

The IPv4 base header fields are classified as follows:

Immutable
- Version
- Internet Header Length
- Total Length
- Identification
- Protocol (This should be the value for AH.)
- Source Address
- Destination Address (without loose or strict source routing)

Mutable but predictable
- Destination Address (with loose or strict source routing)

Mutable (zeroed prior to ICV calculation)
- Type of Service (TOS)
- Flags
- Fragment Offset
- Time to Live (TTL)
- Header Checksum

TOS -- This field is excluded because some routers are known to change the value of this field, even though the IP specification does not consider TOS to be a mutable header field.

Flags -- This field is excluded since an intermediate router might set the DF bit, even if the source did not select it.

Fragment Offset -- Since AH is applied only to non-fragmented IP packets, the Offset Field must always be zero, and thus it is excluded (even though it is predictable).

TTL -- This is changed en-route as a normal course of processing by routers, and thus its value at the receiver is not predictable by the sender.

Header Checksum -- This will change if any of these other fields changes, and thus its value upon reception cannot be predicted by the sender.

Meta 1002
p. 233

3.3.3.1.1.2 Options

For IPv4 (unlike IPv6), there is no mechanism for tagging options as mutable in transit. Hence the IPv4 options are explicitly listed in Appendix A and classified as immutable, mutable but predictable, or mutable. For IPv4, the entire option is viewed as a unit; so even though the type and length fields within most options are immutable in transit, if an option is classified as mutable, the entire option is zeroed for ICV computation purposes.

3.3.3.1.2 ICV Computation for IPv6

3.3.3.1.2.1 Base Header Fields

The IPv6 base header fields are classified as follows:

Immutable
- Version
- Payload Length
- Next Header (This should be the value for AH.)
- Source Address
- Destination Address (without Routing Extension Header)

Mutable but predictable
- Destination Address (with Routing Extension Header)

Mutable (zeroed prior to ICV calculation)
- Class
- Flow Label
- Hop Limit

3.3.3.1.2.2 Extension Headers Containing Options

IPv6 options in the Hop-by-Hop and Destination Extension Headers contain a bit that indicates whether the option might change (unpredictably) during transit. For any option for which contents may change en-route, the entire "Option Data" field must be treated as zero-valued octets when computing or verifying the ICV. The Option Type and Opt Data Len are included in the ICV calculation. All options for which the bit indicates immutability are included in the ICV calculation. See the IPv6 specification [DH95] for more information.

3.3.3.1.2.3 Extension Headers Not Containing Options

The IPv6 extension headers that do not contain options are explicitly listed in Appendix A and classified as immutable, mutable but predictable, or mutable.

Meta 1002
p. 234

3.3.3.2 Padding

3.3.3.2.1 Authentication Data Padding

As mentioned in section 2.6, the Authentication Data field explicitly includes padding to ensure that the AH header is a multiple of 32 bits (IPv4) or 64 bits (IPv6). If padding is required, its length is determined by two factors:

- the length of the ICV
- the IP protocol version (v4 or v6)

For example, if the output of the selected algorithm is 96-bits, no padding is required for either IPv4 or for IPv6. However, if a different length ICV is generated, due to use of a different algorithm, then padding may be required depending on the length and IP protocol version. The content of the padding field is arbitrarily selected by the sender. (The padding is arbitrary, but need not be random to achieve security.) These padding bytes are included in the Authentication Data calculation, counted as part of the Payload Length, and transmitted at the end of the Authentication Data field to enable the receiver to perform the ICV calculation.

3.3.3.2.2 Implicit Packet Padding

For some authentication algorithms, the byte string over which the ICV computation is performed must be a multiple of a blocksize specified by the algorithm. If the IP packet length (including AH) does not match the blocksize requirements for the algorithm, implicit padding MUST be appended to the end of the packet, prior to ICV computation. The padding octets MUST have a value of zero. The blocksize (and hence the length of the padding) is specified by the algorithm specification. This padding is not transmitted with the packet. Note that MD5 and SHA-1 are viewed as having a 1-byte blocksize because of their internal padding conventions.

3.3.4 Fragmentation

If required, IP fragmentation occurs after AH processing within an IPsec implementation. Thus, transport mode AH is applied only to whole IP datagrams (not to IP fragments). An IP packet to which AH has been applied may itself be fragmented by routers en route, and such fragments must be reassembled prior to AH processing at a receiver. In tunnel mode, AH is applied to an IP packet, the payload of which may be a fragmented IP packet. For example, a security gateway or a "bump-in-the-stack" or "bump-in-the-wire" IPsec implementation (see the Security Architecture document for details) may apply tunnel mode AH to such fragments.

3.4 Inbound Packet Processing

If there is more than one IPsec header/extension present, the processing for each one ignores (does not zero, does not use) any IPsec headers applied subsequent to the header being processed.

3.4.1 Reassembly

If required, reassembly is performed prior to AH processing. If a packet offered to AH for processing appears to be an IP fragment, i.e., the OFFSET field is non-zero or the MORE FRAGMENTS flag is set, the receiver MUST discard the packet; this is an auditable event. The audit log entry for this event SHOULD include the SPI value, date/time, Source Address, Destination Address, and (in IPv6) the Flow ID.

NOTE: For packet reassembly, the current IPv4 spec does NOT require either the zero'ing of the OFFSET field or the clearing of the MORE FRAGMENTS flag. In order for a reassembled packet to be processed by IPsec (as opposed to discarded as an apparent fragment), the IP code must do these two things after it reassembles a packet.

3.4.2 Security Association Lookup

Upon receipt of a packet containing an IP Authentication Header, the receiver determines the appropriate (unidirectional) SA, based on the destination IP address, security protocol (AH), and the SPI. (This process is described in more detail in the Security Architecture document.) The SA indicates whether the Sequence Number field will be checked, specifies the algorithm(s) employed for ICV computation, and indicates the key(s) required to validate the ICV.

If no valid Security Association exists for this session (e.g., the receiver has no key), the receiver MUST discard the packet; this is an auditable event. The audit log entry for this event SHOULD include the SPI value, date/time, Source Address, Destination Address, and (in IPv6) the Flow ID.

3.4.3 Sequence Number Verification

All AH implementations MUST support the anti-replay service, though its use may be enabled or disabled by the receiver on a per-SA basis. (Note that there are no provisions for managing transmitted Sequence Number values among multiple senders directing traffic to a single SA (irrespective of whether the destination address is unicast, broadcast, or multicast). Thus the anti-replay service SHOULD NOT be used in a multi-sender environment that employs a single SA.)

If the receiver does not enable anti-replay for an SA, no inbound checks are performed on the Sequence Number. However, from the perspective of the sender, the default is to assume that anti-replay is enabled at the receiver. To avoid having the sender do unnecessary sequence number monitoring and SA setup (see section 3.3.2), if an SA establishment protocol such as IKE is employed, the receiver SHOULD notify the sender, during SA establishment, if the receiver will not provide anti-replay protection.

If the receiver has enabled the anti-replay service for this SA, the receiver packet counter for the SA MUST be initialized to zero when the SA is established. For each received packet, the receiver MUST verify that the packet contains a Sequence Number that does not duplicate the Sequence Number of any other packets received during the life of this SA. This SHOULD be the first AH check applied to a packet after it has been matched to an SA, to speed rejection of duplicate packets.

Duplicates are rejected through the use of a sliding receive window. (How the window is implemented is a local matter, but the following text describes the functionality that the implementation must exhibit.) A MINIMUM window size of 32 MUST be supported; but a window size of 64 is preferred and SHOULD be employed as the default. Another window size (larger than the MINIMUM) MAY be chosen by the receiver. (The receiver does NOT notify the sender of the window size.)

The "right" edge of the window represents the highest, validated Sequence Number value received on this SA. Packets that contain Sequence Numbers lower than the "left" edge of the window are rejected. Packets falling within the window are checked against a list of received packets within the window. An efficient means for performing this check, based on the use of a bit mask, is described in the Security Architecture document.

If the received packet falls within the window and is new, or if the packet is to the right of the window, then the receiver proceeds to ICV verification. If the ICV validation fails, the receiver MUST discard the received IP datagram as invalid; this is an auditable event. The audit log entry for this event SHOULD include the SPI value, date/time, Source Address, Destination Address, the Sequence Number, and (in IPv6) the Flow ID. The receive window is updated only if the ICV verification succeeds.

DISCUSSION:

Note that if the packet is either inside the window and new, or is outside the window on the "right" side, the receiver MUST authenticate the packet before updating the Sequence Number window data.

3.4.4 Integrity Check Value Verification

The receiver computes the ICV over the appropriate fields of the packet, using the specified authentication algorithm, and verifies that it is the same as the ICV included in the Authentication Data field of the packet. Details of the computation are provided below.

If the computed and received ICV's match, then the datagram is valid, and it is accepted. If the test fails, then the receiver MUST discard the received IP datagram as invalid; this is an auditable event. The audit log entry SHOULD include the SPI value, date/time received, Source Address, Destination Address, and (in IPv6) the Flow ID.

DISCUSSION:

Begin by saving the ICV value and replacing it (but not any Authentication Data padding) with zero. Zero all other fields that may have been modified during transit. (See section 3.3.3.1 for a discussion of which fields are zeroed before performing the ICV calculation.) Check the overall length of the packet, and if it requires implicit padding based on the requirements of the authentication algorithm, append zero-filled bytes to the end of the packet as required. Perform the ICV computation and compare the result with the saved value, using the comparison rules defined by the algorithm specification. (For example, if a digital signature and one-way hash are used for the ICV computation, the matching process is more complex.)

4. Auditing

Not all systems that implement AH will implement auditing. However, if AH is incorporated into a system that supports auditing, then the AH implementation MUST also support auditing and MUST allow a system administrator to enable or disable auditing for AH. For the most part, the granularity of auditing is a local matter. However, several auditable events are identified in this specification and for each of these events a minimum set of information that SHOULD be included in an audit log is defined. Additional information also MAY be included in the audit log for each of these events, and additional events, not explicitly called out in this specification, also MAY

result in audit log entries. There is no requirement for the receiver to transmit any message to the purported sender in response to the detection of an auditable event, because of the potential to induce denial of service via such action.

5. Conformance Requirements

Implementations that claim conformance or compliance with this specification MUST fully implement the AH syntax and processing described here and MUST comply with all requirements of the Security Architecture document. If the key used to compute an ICV is manually distributed, correct provision of the anti-replay service would require correct maintenance of the counter state at the sender, until the key is replaced, and there likely would be no automated recovery provision if counter overflow were imminent. Thus a compliant implementation SHOULD NOT provide this service in conjunction with SAs that are manually keyed. A compliant AH implementation MUST support the following mandatory-to-implement algorithms:

- HMAC with MD5 [MG97a]
- HMAC with SHA-1 [MG97b]

6. Security Considerations

Security is central to the design of this protocol, and these security considerations permeate the specification. Additional security-relevant aspects of using the IPsec protocol are discussed in the Security Architecture document.

7. Differences from RFC 1826

This specification of AH differs from RFC 1826 [ATK95] in several important respects, but the fundamental features of AH remain intact. One goal of the revision of RFC 1826 was to provide a complete framework for AH, with ancillary RFCs required only for algorithm specification. For example, the anti-replay service is now an integral, mandatory part of AH, not a feature of a transform defined in another RFC. Carriage of a sequence number to support this service is now required at all times. The default algorithms required for interoperability have been changed to HMAC with MD5 or SHA-1 (vs. keyed MD5), for security reasons. The list of IPv4 header fields excluded from the ICV computation has been expanded to include the OFFSET and FLAGS fields.

Another motivation for revision was to provide additional detail and clarification of subtle points. This specification provides rationale for exclusion of selected IPv4 header fields from AH coverage and provides examples on positioning of AH in both the IPv4

and v6 contexts. Auditing requirements have been clarified in this version of the specification. Tunnel mode AH was mentioned only in passing in RFC 1826, but now is a mandatory feature of AH. Discussion of interactions with key management and with security labels have been moved to the Security Architecture document.

Acknowledgements

For over 3 years, this document has evolved through multiple versions and iterations. During this time, many people have contributed significant ideas and energy to the process and the documents themselves. The authors would like to thank Karen Seo for providing extensive help in the review, editing, background research, and coordination for this version of the specification. The authors would also like to thank the members of the IPsec and IPng working groups, with special mention of the efforts of (in alphabetic order): Steve Bellovin, Steve Deering, Francis Dupont, Phil Karn, Frank Kastenholz, Perry Metzger, David Mihelcic, Hilarie Orman, Norman Shulman, William Simpson, and Nina Yuan.

Appendix A -- Mutability of IP Options/Extension Headers

A1. IPv4 Options

This table shows how the IPv4 options are classified with regard to "mutability". Where two references are provided, the second one supercedes the first. This table is based in part on information provided in RFC1700, "ASSIGNED NUMBERS", (October 1994).

Copy	Opt. Class	#	Name	Reference
IMMUTABLE -- included in ICV calculation				
0	0	0	End of Options List	[RFC791]
0	0	1	No Operation	[RFC791]
1	0	2	Security	[RFC1108(historic but in use)]
1	0	5	Extended Security	[RFC1108(historic but in use)]
1	0	6	Commercial Security	[expired I-D, now US MIL STD]
1	0	20	Router Alert	[RFC2113]
1	0	21	Sender Directed Multi-Destination Delivery	[RFC1770]
MUTABLE -- zeroed				
1	0	3	Loose Source Route	[RFC791]
0	2	4	Time Stamp	[RFC791]
0	0	7	Record Route	[RFC791]
1	0	9	Strict Source Route	[RFC791]
0	2	18	Traceroute	[RFC1393]
EXPERIMENTAL, SUPERCEDED -- zeroed				
1	0	8	Stream ID	[RFC791, RFC1122 (Host Req)]
0	0	11	MTU Probe	[RFC1063, RFC1191 (PMTU)]
0	0	12	MTU Reply	[RFC1063, RFC1191 (PMTU)]
1	0	17	Extended Internet Proto	[RFC1385, RFC1883 (IPv6)]
0	0	10	Experimental Measurement	[ZSu]
1	2	13	Experimental Flow Control	[Finn]
1	0	14	Experimental Access Ctl	[Estrin]
0	0	15	???	[VerSteeg]
1	0	16	IMI Traffic Descriptor	[Lee]
1	0	19	Address Extension	[Ullmann IPv7]

NOTE: Use of the Router Alert option is potentially incompatible with use of IPsec. Although the option is immutable, its use implies that each router along a packet's path will "process" the packet and consequently might change the packet. This would happen on a hop by hop basis as the packet goes from router to router. Prior to being processed by the application to which the option contents are directed, e.g., RSVP/IGMP, the packet should encounter AH processing.

However, AH processing would require that each router along the path is a member of a multicast-SA defined by the SPI. This might pose problems for packets that are not strictly source routed, and it requires multicast support techniques not currently available.

NOTE: Addition or removal of any security labels (BSO, ESO, CIPSO) by systems along a packet's path conflicts with the classification of these IP Options as immutable and is incompatible with the use of IPsec.

NOTE: End of Options List options SHOULD be repeated as necessary to ensure that the IP header ends on a 4 byte boundary in order to ensure that there are no unspecified bytes which could be used for a covert channel.

A2. IPv6 Extension Headers

This table shows how the IPv6 Extension Headers are classified with regard to "mutability".

Option/Extension Name	Reference
MUTABLE BUT PREDICTABLE -- included in ICV calculation	
Routing (Type 0)	[RFC1883]
BIT INDICATES IF OPTION IS MUTABLE (CHANGES UNPREDICTABLY DURING TRANSIT)	
Hop by Hop options	[RFC1883]
Destination options	[RFC1883]
NOT APPLICABLE	
Fragmentation	[RFC1883]

Options -- IPv6 options in the Hop-by-Hop and Destination Extension Headers contain a bit that indicates whether the option might change (unpredictably) during transit. For any option for which contents may change en-route, the entire "Option Data" field must be treated as zero-valued octets when computing or verifying the ICV. The Option Type and Opt Data Len are included in the ICV calculation. All options for which the bit indicates immutability are included in the ICV calculation. See the IPv6 specification [DH95] for more information.

Routing (Type 0) -- The IPv6 Routing Header "Type 0" will rearrange the address fields within the packet during transit from source to destination. However, the contents of the packet as it will appear at the receiver are known to the sender and to all intermediate hops. Hence, the

IPv6 Routing Header "Type 0" is included in the Authentication Data calculation as mutable but predictable. The sender must order the field so that it appears as it will at the receiver, prior to performing the ICV computation.

Fragmentation -- Fragmentation occurs after outbound IPsec processing (section 3.3) and reassembly occurs before inbound IPsec processing (section 3.4). So the Fragmentation Extension Header, if it exists, is not seen by IPsec.

Note that on the receive side, the IP implementation could leave a Fragmentation Extension Header in place when it does re-assembly. If this happens, then when AH receives the packet, before doing ICV processing, AH MUST "remove" (or skip over) this header and change the previous header's "Next Header" field to be the "Next Header" field in the Fragmentation Extension Header.

Note that on the send side, the IP implementation could give the IPsec code a packet with a Fragmentation Extension Header with Offset of 0 (first fragment) and a More Fragments Flag of 0 (last fragment). If this happens, then before doing ICV processing, AH MUST first "remove" (or skip over) this header and change the previous header's "Next Header" field to be the "Next Header" field in the Fragmentation Extension Header.

References

[ATK95] Atkinson, R., "The IP Authentication Header", RFC 1826, August 1995.

[Bra97] Bradner, S., "Key words for use in RFCs to Indicate Requirement Level", BCP 14, RFC 2119, March 1997.

[DH95] Deering, S., and B. Hinden, "Internet Protocol version 6 (IPv6) Specification", RFC 1883, December 1995.

[HC98] Harkins, D., and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

[KA97a] Kent, S., and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[KA97b] Kent, S., and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.

Meta 1002
p. 243

[MG97a] Madson, C., and R. Glenn, "The Use of HMAC-MD5-96 within ESP and AH", RFC 2403, November 1998.

[MG97b] Madson, C., and R. Glenn, "The Use of HMAC-SHA-1-96 within ESP and AH", RFC 2404, November 1998.

[STD-2] Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC 1700, October 1994. See also: http://www.iana.org/numbers.html

Disclaimer

The views and specification here are those of the authors and are not necessarily those of their employers. The authors and their employers specifically disclaim responsibility for any problems arising from correct or incorrect implementation or use of this specification.

Author Information

Stephen Kent
BBN Corporation
70 Fawcett Street
Cambridge, MA 02140
USA

Phone: +1 (617) 873-3988
EMail: kent@bbn.com

Randall Atkinson
@Home Network
425 Broadway,
Redwood City, CA 94063
USA

Phone: +1 (415) 569-5000
EMail: rja@corp.home.net

Meta 1002
p. 244

Copyright (C) The Internet Society (1998). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Network Working Group
Request for Comments: 2406
Obsoletes: 1827
Category: Standards Track

S. Kent
BBN Corp
R. Atkinson
@Home Network

November 1998

IP Encapsulating Security Payload (ESP)

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1998). All Rights Reserved.

Table of Contents

1. Introduction 2
2. Encapsulating Security Payload Packet Format 3
2.1 Security Parameters Index 4
2.2 Sequence Number 4
2.3 Payload Data 5
2.4 Padding (for Encryption 5
2.5 Pad Length 7
2.6 Next Header 7
2.7 Authentication Data 7
3. Encapsulating Security Protocol Processing 7
3.1 ESP Header Location 7
3.2 Algorithms 10
3.2.1 Encryption Algorithms 10
3.2.2 Authentication Algorithms 10
3.3 Outbound Packet Processing 10
3.3.1 Security Association Lookup 11
3.3.2 Packet Encryption 11
3.3.3 Sequence Number Generation 12
3.3.4 Integrity Check Value Calculation 12
3.3.5 Fragmentation 13
3.4 Inbound Packet Processing 13
3.4.1 Reassembly 13
3.4.2 Security Association Lookup 13
3.4.3 Sequence Number Verification 14
3.4.4 Integrity Check Value Verification 15

Meta 1002
p. 246

3.4.5 Packet Decryption....16
4. Auditing....17
5. Conformance Requirements....18
6. Security Considerations....18
7. Differences from RFC 1827....18
Acknowledgements....19
References....19
Disclaimer....20
Author Information....21
Full Copyright Statement....22

1. Introduction

The Encapsulating Security Payload (ESP) header is designed to provide a mix of security services in IPv4 and IPv6. ESP may be applied alone, in combination with the IP Authentication Header (AH) [KA97b], or in a nested fashion, e.g., through the use of tunnel mode (see "Security Architecture for the Internet Protocol" [KA97a], hereafter referred to as the Security Architecture document). Security services can be provided between a pair of communicating hosts, between a pair of communicating security gateways, or between a security gateway and a host. For more details on how to use ESP and AH in various network environments, see the Security Architecture document [KA97a].

The ESP header is inserted after the IP header and before the upper layer protocol header (transport mode) or before an encapsulated IP header (tunnel mode). These modes are described in more detail below.

ESP is used to provide confidentiality, data origin authentication, connectionless integrity, an anti-replay service (a form of partial sequence integrity), and limited traffic flow confidentiality. The set of services provided depends on options selected at the time of Security Association establishment and on the placement of the implementation. Confidentiality may be selected independent of all other services. However, use of confidentiality without integrity/authentication (either in ESP or separately in AH) may subject traffic to certain forms of active attacks that could undermine the confidentiality service (see [Bel96]). Data origin authentication and connectionless integrity are joint services (hereafter referred to jointly as "authentication) and are offered as an option in conjunction with (optional) confidentiality. The anti-replay service may be selected only if data origin authentication is selected, and its election is solely at the discretion of the receiver. (Although the default calls for the sender to increment the Sequence Number used for anti-replay, the service is effective only if the receiver checks the Sequence Number.) Traffic flow

Meta 1002
p. 247

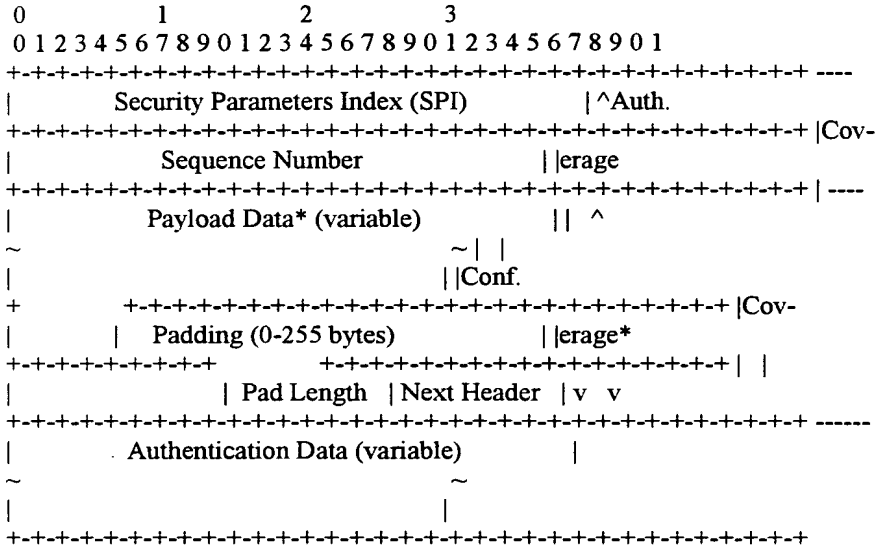
confidentiality requires selection of tunnel mode, and is most effective if implemented at a security gateway, where traffic aggregation may be able to mask true source-destination patterns. Note that although both confidentiality and authentication are optional, at least one of them MUST be selected.

It is assumed that the reader is familiar with the terms and concepts described in the Security Architecture document. In particular, the reader should be familiar with the definitions of security services offered by ESP and AH, the concept of Security Associations, the ways in which ESP can be used in conjunction with the Authentication Header (AH), and the different key management options available for ESP and AH. (With regard to the last topic, the current key management options required for both AH and ESP are manual keying and automated keying via IKE [HC98].)

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in RFC 2119 [Bra97].

2. Encapsulating Security Payload Packet Format

The protocol header (IPv4, IPv6, or Extension) immediately preceding the ESP header will contain the value 50 in its Protocol (IPv4) or Next Header (IPv6, Extension) field [STD-2].



* If included in the Payload field, cryptographic synchronization data, e.g., an Initialization Vector (IV, see

Meta 1002
p. 248

Section 2.3), usually is not encrypted per se, although it often is referred to as being part of the ciphertext.

The following subsections define the fields in the header format. "Optional" means that the field is omitted if the option is not selected, i.e., it is present in neither the packet as transmitted nor as formatted for computation of an Integrity Check Value (ICV, see Section 2.7). Whether or not an option is selected is defined as part of Security Association (SA) establishment. Thus the format of ESP packets for a given SA is fixed, for the duration of the SA. In contrast, "mandatory" fields are always present in the ESP packet format, for all SAs.

2.1 Security Parameters Index

The SPI is an arbitrary 32-bit value that, in combination with the destination IP address and security protocol (ESP), uniquely identifies the Security Association for this datagram. The set of SPI values in the range 1 through 255 are reserved by the Internet Assigned Numbers Authority (IANA) for future use; a reserved SPI value will not normally be assigned by IANA unless the use of the assigned SPI value is specified in an RFC. It is ordinarily selected by the destination system upon establishment of an SA (see the Security Architecture document for more details). The SPI field is mandatory.

The SPI value of zero (0) is reserved for local, implementation-specific use and MUST NOT be sent on the wire. For example, a key management implementation MAY use the zero SPI value to mean "No Security Association Exists" during the period when the IPsec implementation has requested that its key management entity establish a new SA, but the SA has not yet been established.

2.2 Sequence Number

This unsigned 32-bit field contains a monotonically increasing counter value (sequence number). It is mandatory and is always present even if the receiver does not elect to enable the anti-replay service for a specific SA. Processing of the Sequence Number field is at the discretion of the receiver, i.e., the sender MUST always transmit this field, but the receiver need not act upon it (see the discussion of Sequence Number Verification in the "Inbound Packet Processing" section below).

The sender's counter and the receiver's counter are initialized to 0 when an SA is established. (The first packet sent using a given SA will have a Sequence Number of 1; see Section 3.3.3 for more details on how the Sequence Number is generated.) If anti-replay is enabled

(the default), the transmitted Sequence Number must never be allowed to cycle. Thus, the sender's counter and the receiver's counter MUST be reset (by establishing a new SA and thus a new key) prior to the transmission of the 2^32nd packet on an SA.

2.3 Payload Data

Payload Data is a variable-length field containing data described by the Next Header field. The Payload Data field is mandatory and is an integral number of bytes in length. If the algorithm used to encrypt the payload requires cryptographic synchronization data, e.g., an Initialization Vector (IV), then this data MAY be carried explicitly in the Payload field. Any encryption algorithm that requires such explicit, per-packet synchronization data MUST indicate the length, any structure for such data, and the location of this data as part of an RFC specifying how the algorithm is used with ESP. If such synchronization data is implicit, the algorithm for deriving the data MUST be part of the RFC.

Note that with regard to ensuring the alignment of the (real) ciphertext in the presence of an IV:

- o For some IV-based modes of operation, the receiver treats the IV as the start of the ciphertext, feeding it into the algorithm directly. In these modes, alignment of the start of the (real) ciphertext is not an issue at the receiver.
- o In some cases, the receiver reads the IV in separately from the ciphertext. In these cases, the algorithm specification MUST address how alignment of the (real) ciphertext is to be achieved.

2.4 Padding (for Encryption)

Several factors require or motivate use of the Padding field.

- o If an encryption algorithm is employed that requires the plaintext to be a multiple of some number of bytes, e.g., the block size of a block cipher, the Padding field is used to fill the plaintext (consisting of the Payload Data, Pad Length and Next Header fields, as well as the Padding) to the size required by the algorithm.

- o Padding also may be required, irrespective of encryption algorithm requirements, to ensure that the resulting ciphertext terminates on a 4-byte boundary. Specifically,

the Pad Length and Next Header fields must be right aligned within a 4-byte word, as illustrated in the ESP packet format figure above, to ensure that the Authentication Data field (if present) is aligned on a 4-byte boundary.

o Padding beyond that required for the algorithm or alignment reasons cited above, may be used to conceal the actual length of the payload, in support of (partial) traffic flow confidentiality. However, inclusion of such additional padding has adverse bandwidth implications and thus its use should be undertaken with care.

The sender MAY add 0-255 bytes of padding. Inclusion of the Padding field in an ESP packet is optional, but all implementations MUST support generation and consumption of padding.

a. For the purpose of ensuring that the bits to be encrypted are a multiple of the algorithm's blocksize (first bullet above), the padding computation applies to the Payload Data exclusive of the IV, the Pad Length, and Next Header fields.

b. For the purposes of ensuring that the Authentication Data is aligned on a 4-byte boundary (second bullet above), the padding computation applies to the Payload Data inclusive of the IV, the Pad Length, and Next Header fields.

If Padding bytes are needed but the encryption algorithm does not specify the padding contents, then the following default processing MUST be used. The Padding bytes are initialized with a series of (unsigned, 1-byte) integer values. The first padding byte appended to the plaintext is numbered 1, with subsequent padding bytes making up a monotonically increasing sequence: 1, 2, 3, ... When this padding scheme is employed, the receiver SHOULD inspect the Padding field. (This scheme was selected because of its relative simplicity, ease of implementation in hardware, and because it offers limited protection against certain forms of "cut and paste" attacks in the absence of other integrity measures, if the receiver checks the padding values upon decryption.)

Any encryption algorithm that requires Padding other than the default described above, MUST define the Padding contents (e.g., zeros or random data) and any required receiver processing of these Padding bytes in an RFC specifying how the algorithm is used with ESP. In such circumstances, the content of the Padding field will be determined by the encryption algorithm and mode selected and defined in the corresponding algorithm RFC. The relevant algorithm RFC MAY specify that a receiver MUST inspect the Padding field or that a

receiver MUST inform senders of how the receiver will handle the Padding field.

2.5 Pad Length

The Pad Length field indicates the number of pad bytes immediately preceding it. The range of valid values is 0-255, where a value of zero indicates that no Padding bytes are present. The Pad Length field is mandatory.

2.6 Next Header

The Next Header is an 8-bit field that identifies the type of data contained in the Payload Data field, e.g., an extension header in IPv6 or an upper layer protocol identifier. The value of this field is chosen from the set of IP Protocol Numbers defined in the most recent "Assigned Numbers" [STD-2] RFC from the Internet Assigned Numbers Authority (IANA). The Next Header field is mandatory.

2.7 Authentication Data

The Authentication Data is a variable-length field containing an Integrity Check Value (ICV) computed over the ESP packet minus the Authentication Data. The length of the field is specified by the authentication function selected. The Authentication Data field is optional, and is included only if the authentication service has been selected for the SA in question. The authentication algorithm specification MUST specify the length of the ICV and the comparison rules and processing steps for validation.

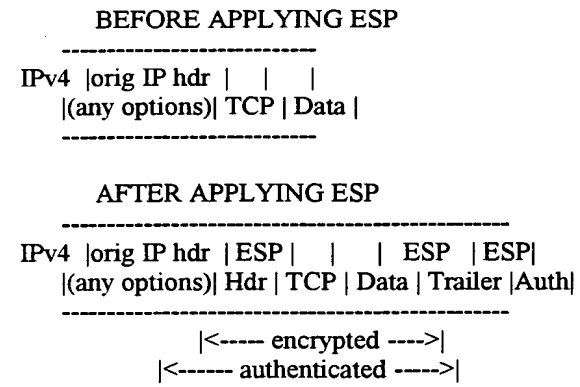
3. Encapsulating Security Protocol Processing

3.1 ESP Header Location

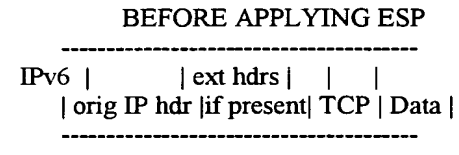
Like AH, ESP may be employed in two ways: transport mode or tunnel mode. The former mode is applicable only to host implementations and provides protection for upper layer protocols, but not the IP header. (In this mode, note that for "bump-in-the-stack" or "bump-in-the-wire" implementations, as defined in the Security Architecture document, inbound and outbound IP fragments may require an IPsec implementation to perform extra IP reassembly/fragmentation in order to both conform to this specification and provide transparent IPsec support. Special care is required to perform such operations within these implementations when multiple interfaces are in use.)

In transport mode, ESP is inserted after the IP header and before an upper layer protocol, e.g., TCP, UDP, ICMP, etc. or before any other IPsec headers that have already been inserted. In the context of

IPv4, this translates to placing ESP after the IP header (and any options that it contains), but before the upper layer protocol. (Note that the term "transport" mode should not be misconstrued as restricting its use to TCP and UDP. For example, an ICMP message MAY be sent using either "transport" mode or "tunnel" mode.) The following diagram illustrates ESP transport mode positioning for a typical IPv4 packet, on a "before and after" basis. (The "ESP trailer" encompasses any Padding, plus the Pad Length, and Next Header fields.)

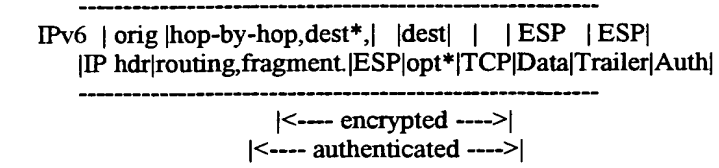


In the IPv6 context, ESP is viewed as an end-to-end payload, and thus should appear after hop-by-hop, routing, and fragmentation extension headers. The destination options extension header(s) could appear either before or after the ESP header depending on the semantics desired. However, since ESP protects only fields after the ESP header, it generally may be desirable to place the destination options header(s) after the ESP header. The following diagram illustrates ESP transport mode positioning for a typical IPv6 packet.



Meta 1002
p. 253

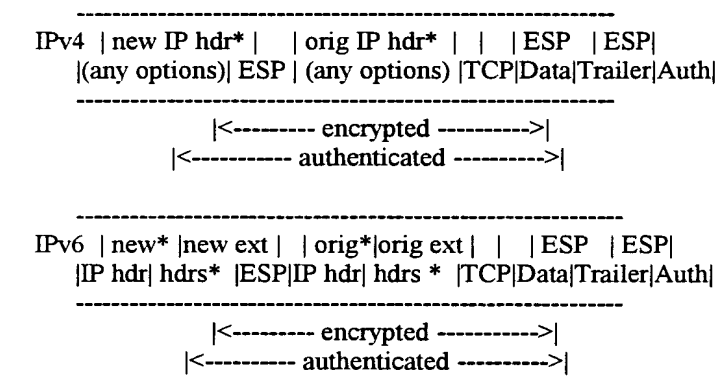
AFTER APPLYING ESP



* = if present, could be before ESP, after ESP, or both

ESP and AH headers can be combined in a variety of modes. The IPsec Architecture document describes the combinations of security associations that must be supported.

Tunnel mode ESP may be employed in either hosts or security gateways. When ESP is implemented in a security gateway (to protect subscriber transit traffic), tunnel mode must be used. In tunnel mode, the "inner" IP header carries the ultimate source and destination addresses, while an "outer" IP header may contain distinct IP addresses, e.g., addresses of security gateways. In tunnel mode, ESP protects the entire inner IP packet, including the entire inner IP header. The position of ESP in tunnel mode, relative to the outer IP header, is the same as for ESP in transport mode. The following diagram illustrates ESP tunnel mode positioning for typical IPv4 and IPv6 packets.



* = if present, construction of outer IP hdr/extensions and modification of inner IP hdr/extensions is discussed below.

3.2 Algorithms

The mandatory-to-implement algorithms are described in Section 5, "Conformance Requirements". Other algorithms MAY be supported. Note that although both confidentiality and authentication are optional, at least one of these services MUST be selected hence both algorithms MUST NOT be simultaneously NULL.

3.2.1 Encryption Algorithms

The encryption algorithm employed is specified by the SA. ESP is designed for use with symmetric encryption algorithms. Because IP packets may arrive out of order, each packet must carry any data required to allow the receiver to establish cryptographic synchronization for decryption. This data may be carried explicitly in the payload field, e.g., as an IV (as described above), or the data may be derived from the packet header. Since ESP makes provision for padding of the plaintext, encryption algorithms employed with ESP may exhibit either block or stream mode characteristics. Note that since encryption (confidentiality) is optional, this algorithm may be "NULL".

3.2.2 Authentication Algorithms

The authentication algorithm employed for the ICV computation is specified by the SA. For point-to-point communication, suitable authentication algorithms include keyed Message Authentication Codes (MACs) based on symmetric encryption algorithms (e.g., DES) or on one-way hash functions (e.g., MD5 or SHA-1). For multicast communication, one-way hash algorithms combined with asymmetric signature algorithms are appropriate, though performance and space considerations currently preclude use of such algorithms. Note that since authentication is optional, this algorithm may be "NULL".

3.3 Outbound Packet Processing

In transport mode, the sender encapsulates the upper layer protocol information in the ESP header/trailer, and retains the specified IP header (and any IP extension headers in the IPv6 context). In tunnel mode, the outer and inner IP header/extensions can be inter-related in a variety of ways. The construction of the outer IP header/extensions during the encapsulation process is described in the Security Architecture document. If there is more than one IPsec header/extension required by security policy, the order of the application of the security headers MUST be defined by security policy.

3.3.1 Security Association Lookup

ESP is applied to an outbound packet only after an IPsec implementation determines that the packet is associated with an SA that calls for ESP processing. The process of determining what, if any, IPsec processing is applied to outbound traffic is described in the Security Architecture document.

3.3.2 Packet Encryption

In this section, we speak in terms of encryption always being applied because of the formatting implications. This is done with the understanding that "no confidentiality" is offered by using the NULL encryption algorithm. Accordingly, the sender:

1. encapsulates (into the ESP Payload field):
   - for transport mode -- just the original upper layer protocol information.
   - for tunnel mode -- the entire original IP datagram.
2. adds any necessary padding.
3. encrypts the result (Payload Data, Padding, Pad Length, and Next Header) using the key, encryption algorithm, algorithm mode indicated by the SA and cryptographic synchronization data (if any).
   - If explicit cryptographic synchronization data, e.g., an IV, is indicated, it is input to the encryption algorithm per the algorithm specification and placed in the Payload field.
   - If implicit cryptographic synchronication data, e.g., an IV, is indicated, it is constructed and input to the encryption algorithm as per the algorithm specification.

The exact steps for constructing the outer IP header depend on the mode (transport or tunnel) and are described in the Security Architecture document.

If authentication is selected, encryption is performed first, before the authentication, and the encryption does not encompass the Authentication Data field. This order of processing facilitates rapid detection and rejection of replayed or bogus packets by the receiver, prior to decrypting the packet, hence potentially reducing the impact of denial of service attacks. It also allows for the possibility of parallel processing of packets at the receiver, i.e., decryption can take place in parallel with authentication. Note that since the Authentication Data is not protected by encryption, a keyed authentication algorithm must be employed to compute the ICV.

Meta 1002
p. 256

3.3.3 Sequence Number Generation

The sender's counter is initialized to 0 when an SA is established. The sender increments the Sequence Number for this SA and inserts the new value into the Sequence Number field. Thus the first packet sent using a given SA will have a Sequence Number of 1.

If anti-replay is enabled (the default), the sender checks to ensure that the counter has not cycled before inserting the new value in the Sequence Number field. In other words, the sender MUST NOT send a packet on an SA if doing so would cause the Sequence Number to cycle. An attempt to transmit a packet that would result in Sequence Number overflow is an auditable event. (Note that this approach to Sequence Number management does not require use of modular arithmetic.)

The sender assumes anti-replay is enabled as a default, unless otherwise notified by the receiver (see 3.4.3). Thus, if the counter has cycled, the sender will set up a new SA and key (unless the SA was configured with manual key management).

If anti-replay is disabled, the sender does not need to monitor or reset the counter, e.g., in the case of manual key management (see Section 5). However, the sender still increments the counter and when it reaches the maximum value, the counter rolls over back to zero.

3.3.4 Integrity Check Value Calculation

If authentication is selected for the SA, the sender computes the ICV over the ESP packet minus the Authentication Data. Thus the SPI, Sequence Number, Payload Data, Padding (if present), Pad Length, and Next Header are all encompassed by the ICV computation. Note that the last 4 fields will be in ciphertext form, since encryption is performed prior to authentication.

For some authentication algorithms, the byte string over which the ICV computation is performed must be a multiple of a blocksize specified by the algorithm. If the length of this byte string does not match the blocksize requirements for the algorithm, implicit padding MUST be appended to the end of the ESP packet, (after the Next Header field) prior to ICV computation. The padding octets MUST have a value of zero. The blocksize (and hence the length of the padding) is specified by the algorithm specification. This padding is not transmitted with the packet. Note that MD5 and SHA-1 are viewed as having a 1-byte blocksize because of their internal padding conventions.

3.3.5 Fragmentation

If necessary, fragmentation is performed after ESP processing within an IPsec implementation. Thus, transport mode ESP is applied only to whole IP datagrams (not to IP fragments). An IP packet to which ESP has been applied may itself be fragmented by routers en route, and such fragments must be reassembled prior to ESP processing at a receiver. In tunnel mode, ESP is applied to an IP packet, the payload of which may be a fragmented IP packet. For example, a security gateway or a "bump-in-the-stack" or "bump-in-the-wire" IPsec implementation (as defined in the Security Architecture document) may apply tunnel mode ESP to such fragments.

NOTE: For transport mode -- As mentioned at the beginning of Section 3.1, bump-in-the-stack and bump-in-the-wire implementations may have to first reassemble a packet fragmented by the local IP layer, then apply IPsec, and then fragment the resulting packet.

NOTE: For IPv6 -- For bump-in-the-stack and bump-in-the-wire implementations, it will be necessary to walk through all the extension headers to determine if there is a fragmentation header and hence that the packet needs reassembling prior to IPsec processing.

3.4 Inbound Packet Processing

3.4.1 Reassembly

If required, reassembly is performed prior to ESP processing. If a packet offered to ESP for processing appears to be an IP fragment, i.e., the OFFSET field is non-zero or the MORE FRAGMENTS flag is set, the receiver MUST discard the packet; this is an auditable event. The audit log entry for this event SHOULD include the SPI value, date/time received, Source Address, Destination Address, Sequence Number, and (in IPv6) the Flow ID.

NOTE: For packet reassembly, the current IPv4 spec does NOT require either the zero'ing of the OFFSET field or the clearing of the MORE FRAGMENTS flag. In order for a reassembled packet to be processed by IPsec (as opposed to discarded as an apparent fragment), the IP code must do these two things after it reassembles a packet.

3.4.2 Security Association Lookup

Upon receipt of a (reassembled) packet containing an ESP Header, the receiver determines the appropriate (unidirectional) SA, based on the destination IP address, security protocol (ESP), and the SPI. (This process is described in more detail in the Security Architecture document.) The SA indicates whether the Sequence Number field will

be checked, whether the Authentication Data field should be present, and it will specify the algorithms and keys to be employed for decryption and ICV computations (if applicable).

If no valid Security Association exists for this session (for example, the receiver has no key), the receiver MUST discard the packet; this is an auditable event. The audit log entry for this event SHOULD include the SPI value, date/time received, Source Address, Destination Address, Sequence Number, and (in IPv6) the cleartext Flow ID.

3.4.3 Sequence Number Verification

All ESP implementations MUST support the anti-replay service, though its use may be enabled or disabled by the receiver on a per-SA basis. This service MUST NOT be enabled unless the authentication service also is enabled for the SA, since otherwise the Sequence Number field has not been integrity protected. (Note that there are no provisions for managing transmitted Sequence Number values among multiple senders directing traffic to a single SA (irrespective of whether the destination address is unicast, broadcast, or multicast). Thus the anti-replay service SHOULD NOT be used in a multi-sender environment that employs a single SA.)

If the receiver does not enable anti-replay for an SA, no inbound checks are performed on the Sequence Number. However, from the perspective of the sender, the default is to assume that anti-replay is enabled at the receiver. To avoid having the sender do unnecessary sequence number monitoring and SA setup (see section 3.3.3), if an SA establishment protocol such as IKE is employed, the receiver SHOULD notify the sender, during SA establishment, if the receiver will not provide anti-replay protection.

If the receiver has enabled the anti-replay service for this SA, the receive packet counter for the SA MUST be initialized to zero when the SA is established. For each received packet, the receiver MUST verify that the packet contains a Sequence Number that does not duplicate the Sequence Number of any other packets received during the life of this SA. This SHOULD be the first ESP check applied to a packet after it has been matched to an SA, to speed rejection of duplicate packets.

Duplicates are rejected through the use of a sliding receive window. (How the window is implemented is a local matter, but the following text describes the functionality that the implementation must exhibit.) A MINIMUM window size of 32 MUST be supported; but a window size of 64 is preferred and SHOULD be employed as the default.

Another window size (larger than the MINIMUM) MAY be chosen by the receiver. (The receiver does NOT notify the sender of the window size.)

The "right" edge of the window represents the highest, validated Sequence Number value received on this SA. Packets that contain Sequence Numbers lower than the "left" edge of the window are rejected. Packets falling within the window are checked against a list of received packets within the window. An efficient means for performing this check, based on the use of a bit mask, is described in the Security Architecture document.

If the received packet falls within the window and is new, or if the packet is to the right of the window, then the receiver proceeds to ICV verification. If the ICV validation fails, the receiver MUST discard the received IP datagram as invalid; this is an auditable event. The audit log entry for this event SHOULD include the SPI value, date/time received, Source Address, Destination Address, the Sequence Number, and (in IPv6) the Flow ID. The receive window is updated only if the ICV verification succeeds.

DISCUSSION:

Note that if the packet is either inside the window and new, or is outside the window on the "right" side, the receiver MUST authenticate the packet before updating the Sequence Number window data.

3.4.4 Integrity Check Value Verification

If authentication has been selected, the receiver computes the ICV over the ESP packet minus the Authentication Data using the specified authentication algorithm and verifies that it is the same as the ICV included in the Authentication Data field of the packet. Details of the computation are provided below.

If the computed and received ICV's match, then the datagram is valid, and it is accepted. If the test fails, then the receiver MUST discard the received IP datagram as invalid; this is an auditable event. The log data SHOULD include the SPI value, date/time received, Source Address, Destination Address, the Sequence Number, and (in IPv6) the cleartext Flow ID.

DISCUSSION:

Begin by removing and saving the ICV value (Authentication Data field). Next check the overall length of the ESP packet minus the Authentication Data. If implicit padding is required, based on

the blocksize of the authentication algorithm, append zero-filled bytes to the end of the ESP packet directly after the Next Header field. Perform the ICV computation and compare the result with the saved value, using the comparison rules defined by the algorithm specification. (For example, if a digital signature and one-way hash are used for the ICV computation, the matching process is more complex.)

3.4.5 Packet Decryption

As in section 3.3.2, "Packet Encryption", we speak here in terms of encryption always being applied because of the formatting implications. This is done with the understanding that "no confidentiality" is offered by using the NULL encryption algorithm. Accordingly, the receiver:

1. decrypts the ESP Payload Data, Padding, Pad Length, and Next Header using the key, encryption algorithm, algorithm mode, and cryptographic synchronization data (if any), indicated by the SA.
   - If explicit cryptographic synchronization data, e.g., an IV, is indicated, it is taken from the Payload field and input to the decryption algorithm as per the algorithm specification.
   - If implicit cryptographic synchronization data, e.g., an IV, is indicated, a local version of the IV is constructed and input to the decryption algorithm as per the algorithm specification.
2. processes any padding as specified in the encryption algorithm specification. If the default padding scheme (see Section 2.4) has been employed, the receiver SHOULD inspect the Padding field before removing the padding prior to passing the decrypted data to the next layer.
3. reconstructs the original IP datagram from:
   - for transport mode -- original IP header plus the original upper layer protocol information in the ESP Payload field
   - for tunnel mode -- tunnel IP header + the entire IP datagram in the ESP Payload field.

The exact steps for reconstructing the original datagram depend on the mode (transport or tunnel) and are described in the Security Architecture document. At a minimum, in an IPv6 context, the receiver SHOULD ensure that the decrypted data is 8-byte aligned, to facilitate processing by the protocol identified in the Next Header field.

If authentication has been selected, verification and decryption MAY be performed serially or in parallel. If performed serially, then ICV verification SHOULD be performed first. If performed in parallel, verification MUST be completed before the decrypted packet is passed on for further processing. This order of processing facilitates rapid detection and rejection of replayed or bogus packets by the receiver, prior to decrypting the packet, hence potentially reducing the impact of denial of service attacks. Note:

If the receiver performs decryption in parallel with authentication, care must be taken to avoid possible race conditions with regard to packet access and reconstruction of the decrypted packet.

Note that there are several ways in which the decryption can "fail":

a. The selected SA may not be correct -- The SA may be mis-selected due to tampering with the SPI, destination address, or IPsec protocol type fields. Such errors, if they map the packet to another extant SA, will be indistinguishable from a corrupted packet, (case c). Tampering with the SPI can be detected by use of authentication. However, an SA mismatch might still occur due to tampering with the IP Destination Address or the IPsec protocol type field.

b. The pad length or pad values could be erroneous -- Bad pad lengths or pad values can be detected irrespective of the use of authentication.

c. The encrypted ESP packet could be corrupted -- This can be detected if authentication is selected for the SA.,

In case (a) or (c), the erroneous result of the decryption operation (an invalid IP datagram or transport-layer frame) will not necessarily be detected by IPsec, and is the responsibility of later protocol processing.

4. Auditing

Not all systems that implement ESP will implement auditing. However, if ESP is incorporated into a system that supports auditing, then the ESP implementation MUST also support auditing and MUST allow a system administrator to enable or disable auditing for ESP. For the most part, the granularity of auditing is a local matter. However, several auditable events are identified in this specification and for each of these events a minimum set of information that SHOULD be included in an audit log is defined. Additional information also MAY be included in the audit log for each of these events, and additional

events, not explicitly called out in this specification, also MAY result in audit log entries. There is no requirement for the receiver to transmit any message to the purported sender in response to the detection of an auditable event, because of the potential to induce denial of service via such action.

5. Conformance Requirements

Implementations that claim conformance or compliance with this specification MUST implement the ESP syntax and processing described here and MUST comply with all requirements of the Security Architecture document. If the key used to compute an ICV is manually distributed, correct provision of the anti-replay service would require correct maintenance of the counter state at the sender, until the key is replaced, and there likely would be no automated recovery provision if counter overflow were imminent. Thus a compliant implementation SHOULD NOT provide this service in conjunction with SAs that are manually keyed. A compliant ESP implementation MUST support the following mandatory-to-implement algorithms:

- DES in CBC mode [MD97]
- HMAC with MD5 [MG97a]
- HMAC with SHA-1 [MG97b]
- NULL Authentication algorithm
- NULL Encryption algorithm

Since ESP encryption and authentication are optional, support for the 2 "NULL" algorithms is required to maintain consistency with the way these services are negotiated. NOTE that while authentication and encryption can each be "NULL", they MUST NOT both be "NULL".

6. Security Considerations

Security is central to the design of this protocol, and thus security considerations permeate the specification. Additional security-relevant aspects of using the IPsec protocol are discussed in the Security Architecture document.

7. Differences from RFC 1827

This document differs from RFC 1827 [ATK95] in several significant ways. The major difference is that, this document attempts to specify a complete framework and context for ESP, whereas RFC 1827 provided a "shell" that was completed through the definition of transforms. The combinatorial growth of transforms motivated the reformulation of the ESP specification as a more complete document, with options for security services that may be offered in the context of ESP. Thus, fields previously defined in transform documents are

now part of this base ESP specification. For example, the fields necessary to support authentication (and anti-replay) are now defined here, even though the provision of this service is an option. The fields used to support padding for encryption, and for next protocol identification, are now defined here as well. Packet processing consistent with the definition of these fields also is included in the document.

Acknowledgements

Many of the concepts embodied in this specification were derived from or influenced by the US Government's SP3 security protocol, ISO/IEC's NLSP, or from the proposed swIPe security protocol. [SDNS89, ISO92, IB93].

For over 3 years, this document has evolved through multiple versions and iterations. During this time, many people have contributed significant ideas and energy to the process and the documents themselves. The authors would like to thank Karen Seo for providing extensive help in the review, editing, background research, and coordination for this version of the specification. The authors would also like to thank the members of the IPsec and IPng working groups, with special mention of the efforts of (in alphabetic order): Steve Bellovin, Steve Deering, Phil Karn, Perry Metzger, David Mihelcic, Hilarie Orman, Norman Shulman, William Simpson and Nina Yuan.

References

[ATK95] Atkinson, R., "IP Encapsulating Security Payload (ESP)", RFC 1827, August 1995.

[Bel96] Steven M. Bellovin, "Problem Areas for the IP Security Protocols", Proceedings of the Sixth Usenix Unix Security Symposium, July, 1996.

[Bra97] Bradner, S., "Key words for use in RFCs to Indicate Requirement Level", BCP 14, RFC 2119, March 1997.

[HC98] Harkins, D., and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

[IB93] John Ioannidis & Matt Blaze, "Architecture and Implementation of Network-layer Security Under Unix", Proceedings of the USENIX Security Symposium, Santa Clara, CA, October 1993.

[ISO92] ISO/IEC JTC1/SC6, Network Layer Security Protocol, ISO-IEC DIS 11577, International Standards Organisation, Geneva, Switzerland, 29 November 1992.

[KA97a] Kent, S., and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[KA97b] Kent, S., and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.

[MD97] Madson, C., and N. Doraswamy, "The ESP DES-CBC Cipher Algorithm With Explicit IV", RFC 2405, November 1998.

[MG97a] Madson, C., and R. Glenn, "The Use of HMAC-MD5-96 within ESP and AH", RFC 2403, November 1998.

[MG97b] Madson, C., and R. Glenn, "The Use of HMAC-SHA-1-96 within ESP and AH", RFC 2404, November 1998.

[STD-2] Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC 1700, October 1994. See also: http://www.iana.org/numbers.html

[SDNS89] SDNS Secure Data Network System, Security Protocol 3, SP3, Document SDN.301, Revision 1.5, 15 May 1989, as published in NIST Publication NIST-IR-90-4250, February 1990.

Disclaimer

The views and specification here are those of the authors and are not necessarily those of their employers. The authors and their employers specifically disclaim responsibility for any problems arising from correct or incorrect implementation or use of this specification.

Meta 1002
p. 265

Author Information

Stephen Kent
BBN Corporation
70 Fawcett Street
Cambridge, MA 02140
USA

Phone: +1 (617) 873-3988
EMail: kent@bbn.com

Randall Atkinson
@Home Network
425 Broadway,
Redwood City, CA 94063
USA

Phone: +1 (415) 569-5000
EMail: rja@corp.home.net

Meta 1002
p. 266

Full Copyright Statement

Copyright (C) The Internet Society (1998). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents, Box PCT
United States Patent and Trademark Office
Washington, D.C. 20231
www.uspto.gov

U.S. APPLICATION NUMBER NO.	FIRST NAMED APPLICANT	ATTY. DOCKET NO.
10/181,564	Martin Euchner	1454.1302

INTERNATIONAL APPLICATION NO.	
PCT/DE01/00021	
I.A. FILING DATE	PRIORITY DATE
01/05/2001	01/18/2000

21171
STAAS & HALSEY LLP
700 11TH STREET, NW
SUITE 500
WASHINGTON, DC 20001

CONFIRMATION NO. 4619

371 ACCEPTANCE LETTER

OC000000008792743

Date Mailed: 09/18/2002

NOTICE OF ACCEPTANCE OF APPLICATION UNDER 35 U.S.C 371 AND 37 CFR 1.494 OR 1.495

The applicant is hereby advised that the United States Patent and Trademark Office in its capacity as an Elected Office (37 CFR 1.495) , has determined that the above identified international application has met the requirements of 35 U.S.C. 371, and is ACCEPTED for national patentability examination in the United States Patent and Trademark Office.

The United States Application Number assigned to the application is shown above and the relevant dates are:

07/18/2002	07/18/2002
DATE OF RECEIPT OF 35 U.S.C. 371(c)(1), (c)(2) and (c)(4) REQUIREMENTS	DATE OF RECEIPT OF ALL 35 U.S.C. REQUIREMENTS

A Filing Receipt (PTO-103X) will be issued for the present application in due course. **THE DATE APPEARING ON THE FILING RECEIPT AS THE " FILING DATE" IS THE DATE ON WHICH THE LAST OF THE 35 U.S.C. 371 REQUIREMENTS HAS BEEN RECEIVED IN THE OFFICE. THIS DATE IS SHOWN ABOVE.** *The filing date of the above identified application is the international filing date of the international application (Article 11(3) and 35 U.S.C. 363).* Once the Filing Receipt has been received, send all correspondence to the Group Art Unit designated thereon.

The following items have been received:

- U.S. Basic National Fee
- Assignee Statement
- Copy of IPE Report
- Copy of references cited in ISR
- Copy of the International Application
- Copy of the International Search Report
- Information Disclosure Statements
- Oath or Declaration
- Preliminary Amendments

- Request for Immediate Examination

Applicant is reminded that any communications to the United States Patent and Trademark Office must be mailed to the address given in the heading and include the U.S. application no. shown above (37 CFR 1.5)

BARBARA A CAMPBELL
Telephone: (703) 305-3631

PART 3 - OFFICE COPY

FORM PCT/DO/EO/903 (371 Acceptance Notice)

PATENT COOPERATION TREATY

From the INTERNATIONAL BUREAU

PCT

NOTIFICATION OF ELECTION

(PCT Rule 61.2)

To:

Commissioner
US Department of Commerce
United States Patent and Trademark Office, PCT
2011 South Clark Place Room
CP2/5C24
Arlington, VA 22202
ETATS-UNIS D'AMERIQUE

in its capacity as elected Office

Date of mailing (day/month/year) 15 October 2001 (15.10.01)	
International application No. PCT/DE01/00021	**Applicant's or agent's file reference** 2000P01063WO
International filing date (day/month/year) 05 January 2001 (05.01.01)	**Priority date** (day/month/year) 18 January 2000 (18.01.00)
Applicant EUCHNER, Martin	

1. The designated Office is hereby notified of its election made:

[X] in the demand filed with the International Preliminary Examining Authority on:

07 August 2001 (07.08.01)

[ ] in a notice effecting later election filed with the International Bureau on:

2. The election [X] was

[ ] was not

made before the expiration of 19 months from the priority date or, where Rule 32 applies, within the time limit under Rule 32.2(b).

The International Bureau of WIPO **34, chemin des Colombettes** **1211 Geneva 20, Switzerland**	Authorized officer Kari HUYNH-KHUONG
Facsimile No.: (41-22) 740.14.35	Telephone No.: (41-22) 338.83.38

Meta 1002
p. 270

Form PCT/IB/331 (July 1992)

DE0100021

THIS PAGE BLANK (USPTO)

NE 10/181564

Translation

4

PATENT COOPERATION TREATY

PCT

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

(PCT Article 36 and Rule 70)

Applicant's or agent's file reference 2000P01063WO	**FOR FURTHER ACTION**	See Notification of Transmittal of International Preliminary Examination Report (Form PCT/IPEA/416)
International application No. PCT/DE01/00021	International filing date (*day/month/year*) 05 January 2001 (05.01.01)	Priority date (*day/month/year*) 18 January 2000 (18.01.00)
International Patent Classification (IPC) or national classification and IPC H04L 29/00		
Applicant SIEMENS AKTIENGESELLSCHAFT		

1. This international preliminary examination report has been prepared by this International Preliminary Examining Authority and is transmitted to the applicant according to Article 36.

2. This REPORT consists of a total of 2 sheets, including this cover sheet.

☐ This report is also accompanied by ANNEXES, i.e., sheets of the description, claims and/or drawings which have been amended and are the basis for this report and/or sheets containing rectifications made before this Authority (see Rule 70.16 and Section 607 of the Administrative Instructions under the PCT).

These annexes consist of a total of _____ sheets.

3. This report contains indications relating to the following items:

- I ☒ Basis of the report
- II ☐ Priority
- III ☒ Non-establishment of opinion with regard to novelty, inventive step and industrial applicability
- IV ☐ Lack of unity of invention
- V ☒ Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement
- VI ☐ Certain documents cited
- VII ☐ Certain defects in the international application
- VIII ☐ Certain observations on the international application

Date of submission of the demand 07 August 2001 (07.08.01)	Date of completion of this report 09 March 2002 (09.03.2002)
Name and mailing address of the IPEA/EP Facsimile No.	Authorized officer Telephone No.

Form PCT/IPEA/409 (cover sheet) (July 1998)

THIS PAGE BLANK (USPTO)

I. Basis of report

1. This report has been drawn up on the basis of the international application as originally filed.

III. Non-establishment of opinion with regard to novelty, inventive step and industrial applicability

2. The questions whether the claimed invention appears to be novel, to involve an inventive step, or to be industrially applicable have not been examined (PCT Article 34 (4) (a) (i) and (ii); see also the international search report) in respect of:

2.1 Applications with unnecessary independent claims (generally no more than one independent claim per category is required; PCT Article 6)

2.2 Subject for which a search has not been carried out (PCT Article 17(2)(a), PCT rule 66.1(e)), e.g.

a) claimed subjects pursuant to PCT rule 39.1
b) applications of which the description, claims or drawings meet the prescribed requirements so inadequately that no meaningful opinion could be formed

2.3 Claimed subject matter pursuant to PCT Rule 67.1

V. Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability

3. Insofar as the international preliminary examination has been carried out (see box III), the following should be noted:

In the light of the documents listed in the international search report, it is established that the invention as characterized in at least one independent claim apparently does not meet the requirements of PCT Article 33(1), that is, it cannot be considered novel and/or to invilve an inventive step.

THIS PAGE BLANK (USPTO)

PCT-ANTRAG 2000P01063WO

Original (für **EINREICHUNG**) - gedruckt am 13.11.2000 09:21:25 AM

0 **0-1**	**Vom Anmeldeamt auszufüllen** Internationales Aktenzeichen.	
0-2	Internationales Anmeldedatum	
0-3	Name des Anmeldeamts und "PCT International Application"	
0-4 0-4-1	**Formular - PCT/RO/101 PCT-Antrag** erstellt durch Benutzung von	PCT-EASY Version 2.91 (aktualisiert 01.07.2000)
0-5	**Antragsersuchen** Der Unterzeichnete beantragt, daß die vorliegende internationale Anmeldung nach dem Vertrag über die internationale Zusammenarbeit auf dem Gebiet des Patentwesens behandelt wird	
0-6	**(Vom Anmelder gewähltes) Anmeldeamt**	Deutsches Patent- und Markenamt (RO/DE)
0-7	**Aktenzeichen des Anmelders oder Anwalts**	2000P01063WO
I	**Bezeichnung der Erfindung**	VERFAHREN, SYSTEM ZUR ÜBERMITTLUNG VON DATEN VON EINEM SENDER ZU EINEM EMPFÄNGER UND SENDER BZW. EMPFÄNGER HIERZU
II	**Anmelder**	
II-1	Diese Person ist	nur Anmelder
II-2	Anmelder für	Alle Bestimmungsstaaten mit Ausnahme von US
II-4	Name	SIEMENS AKTIENGESELLSCHAFT
II-5	Anschrift:	Wittelsbacherplatz 2 D-80333 München Deutschland
II-6	Staatsangehörigkeit (Staat)	DE
II-7	Sitz/Wohnsitz (Staat)	DE
II-8	Telefonnr.	089 636 828 19
II-9	Telefaxnr.	089 636 818 57
III-1	**Anmelder und/oder Erfinder**	
III-1-1	Diese Person ist	Anmelder und Erfinder
III-1-2	Anmelder für	Nur US
III-1-4	Name (FAMILIENNAME, Vorname)	EUCHNER, Martin
III-1-5	Anschrift:	Lorenzstraße 2 D-81737 München Deutschland
III-1-6	Staatsangehörigkeit (Staat)	DE
III-1-7	Sitz/Wohnsitz (Staat)	DE

Flu 5.1.01

THIS PAGE BLANK (USPTO)

PCT-ANTRAG 2000P01063WO

Original (für **EINREICHUNG**) - gedruckt am 13.11.2000 09:21:25 AM

IV-1	**Anwalt oder gemeinsamer Vertreter; oder besondere Zustellanschrift** Die unten bezeichnete Person ist/wird hiermit bestellt, um den (die) Anmelder vor den internationalen Behörden zu vertreten, und zwar als:	gemeinsamer Vertreter
IV-1-1	Name	SIEMENS AKTIENGESELLSCHAFT
IV-1-2	Anschrift:	Postfach 22 16 34 D-80506 München Deutschland
IV-1-3	Telefonnr.	089 636 828 19
IV-1-4	Telefaxnr.	089 636 818 57
V	**Bestimmung von Staaten**	
V-1	Regionales Patent (andere Schutzrechtsarten oder Verfahren sind ggf. in Klammern nach der (den) betreffenden Bestimmung(en) angegeben)	EP: AT BE CH&LI CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE und jeder weitere Staat, der Mitgliedsstaat des Europäischen Patentübereinkommens und Vertragsstaat des PCT ist
V-2	Nationales Patent (andere Schutzrechtsarten oder Verfahren sind ggf. in Klammern nach der (den) betreffenden Bestimmung(en) angegeben)	CA JP US
V-5	**Erklärung bzgl. vorsorglicher Bestimmungen** Zusätzlich zu den unter Punkten V-1, V-2 and V-3 vorgenommenen Bestimmungen nimmt der Anmelder nach Regel 4.9 Absatz b auch alle anderen nach dem PCT zulässigen Bestimmungen vor mit Ausnahme der nachstehend unter Punkt V-6 angegebenen Staaten. Der Anmelder erklärt, daß diese zusätzlichen Bestimmungen unter dem Vorbehalt einer Bestätigung stehen und jede zusätzliche Bestimmung, die vor Ablauf von 15 Monaten ab dem Prioritätsdatum nicht bestätigt wurde, nach Ablauf dieser Frist als vom Anmelder zurückgenommen gilt.	
V-6	**Staaten, die von der Erklärung über vorsorgliche Bestimmungen ausgenommen werden**	KEINE
VI-1	**Priorität einer früheren nationalen Anmeldung beansprucht**	
VI-1-1	Anmeldedatum	18 Januar 2000 (18.01.2000)
VI-1-2	Aktenzeichen	100 01 855.6
VI-1-3	Staat	DE
VI-2	**Ersuchen um Erstellung eines Prioritätsbeleges** Das Anmeldeamt wird ersucht, eine beglaubigte Abschrift der in der (den) nachstehend genannten Zeile(n) bezeichneten früheren Anmeldung(en) zu erstellen und dem internationalen Büro zu übermitteln:	VI-1
VII-1	**Gewählte Internationale Recherchenbehörde**	Europäisches Patentamt (EPA) (ISA/EP)

THIS PAGE BLANK (USPTO)

PCT-ANTRAG

2000P01063WO

Original (für **EINREICHUNG**) - gedruckt am 13.11.2000 09:21:25 AM

VIII	Kontrolliste	Anzahl der Blätter	Elektronische Datei(en) beigefügt
VIII-1	Antrag	4	-
VIII-2	Beschreibung	15	-
VIII-3	Ansprüche	3	-
VIII-4	Zusammenfassung	1	1999p01093.txt
VIII-5	Zeichnung(en)	3	-
VIII-7	INSGESAMT	26	
	Beigefügte Unterlagen	Unterlage(n) in Papierform beigefügt	Elektronische Datei(en) beigefügt
VIII-8	Blatt für die Gebührenberechnung	✓	-
VIII-16	PCT-EASY-Diskette	-	Diskette
VIII-17	Sonstige (einzeln aufgeführt):	Kopie der Ursprungsfassung	-
VIII-18	**Nr. der Abb. der Zeichn., die mit der Zusammenf. veröffentlicht werden soll**		
VIII-19	**Sprache der int. Anmeldung**	Deutsch	
IX-1	**Unterschrift des Anmelders oder Anwalts**		
IX-1-1	Name	SIEMENS AKTIENGESELLSCHAFT	
IX-1-2	Name der unterzeichnenden Person	Hashuber i.V. Hashuber	
IX-1-3	Eigenschaft	Nr. 144/74 Ang-AV	
IX-2	**Unterschrift des Anmelders oder Anwalts**		
IX-2-1	Name (FAMILIENNAME, Vorname)	EUCHNER, Martin	

VOM ANMELDEAMT AUSZUFÜLLEN

10-1	**Datum des tatsächlichen Eingangs dieser internationalen Anmeldung**	
10-2	**Zeichnung(en):**	
10-2-1	Eingegangen	
10-2-2	Nicht eingegangen	
10-3	**Geändertes Eingangsdatum aufgrund nachträglich, jedoch fristgerecht eingeg. Unterlage(n) oder Zeichnung(en) zur Vervollständigung dieser int. Anmeldung**	
10-4	**Datum des fristgerechten Eingangs der Berichtigung nach PCT Artikel 11(2)**	
10-5	**Internationale Recherchenbehörde**	ISA/EP
10-6	**Übermittlung des Recherchenexemplars bis zur Zahlung der Recherchengebühr aufgeschoben**	

THIS PAGE BLANK (USPTO)

PCT-ANTRAG 2000P01063WO

Original (für **EINREICHUNG**) - gedruckt am 13.11.2000 09:21:25 AM

VIII	Kontrolliste	Anzahl der Blätter	Elektronische Datei(en) beigefügt
VIII-1	Antrag	4	-
VIII-2	Beschreibung	15	-
VIII-3	Ansprüche	3	-
VIII-4	Zusammenfassung	1	1999p01093.txt
VIII-5	Zeichnung(en)	3	-
VIII-7	INSGESAMT	26	
	Beigefügte Unterlagen	Unterlage(n) in Papierform beigefügt	Elektronische Datei(en) beigefügt
VIII-8	Blatt für die Gebührenberechnung	✓	-
VIII-16	PCT-EASY-Diskette	-	Diskette
VIII-17	Sonstige (einzeln aufgeführt):	Kopie der Ursprungsfassung	-
VIII-18	**Nr. der Abb. der Zeichn., die mit der Zusammenf. veröffentlicht werden soll**		
VIII-19	**Sprache der int. Anmeldung**	Deutsch	
IX-1	**Unterschrift des Anmelders oder Anwalts**		
IX-1-1	Name	SIEMENS AKTIENGESELLSCHAFT	
IX-1-2	Name der unterzeichnenden Person	Hashuber i.V. Hashuber	
IX-1-3	Eigenschaft	Nr. 144/74 Ang-AV	
IX-2	**Unterschrift des Anmelders oder Anwalts**	Martin Euchner	
IX-2-1	Name (FAMILIENNAME, Vorname)	EUCHNER, Martin	

VOM ANMELDEAMT AUSZUFÜLLEN

10-1	**Datum des tatsächlichen Eingangs dieser internationalen Anmeldung**	
10-2	**Zeichnung(en):**	
10-2-1	Eingegangen	
10-2-2	Nicht eingegangen	
10-3	**Geändertes Eingangsdatum aufgrund nachträglich, jedoch fristgerecht eingeg. Unterlage(n) oder Zeichnung(en) zur Vervollständigung dieser int. Anmeldung**	
10-4	**Datum des fristgerechten Eingangs der Berichtigung nach PCT Artikel 11(2)**	
10-5	**Internationale Recherchenbehörde**	ISA/EP
10-6	**Übermittlung des Recherchenexemplars bis zur Zahlung der Recherchengebühr aufgeschoben**	

THIS PAGE BLANK (USPTO)

PCT-ANTRAG

2000P01063WO

Original (für **EINREICHUNG**) - gedruckt am 13.11.2000 09:21:25 AM

VOM INTERNATIONALEN BÜRO AUSZUFÜLLEN

11-1	**Datum des Eingangs des Aktenexemplars beim Internationalen Büro**	

THIS PAGE BLANK (USPTO)

PCT (ANHANG - BLATT FÜR DIE GEBÜHRENBERECHNUNG)

2000P01063WO

Original (für **EINREICHUNG**) - gedruckt am 13.11.2000 09:21:25 AM

(Dieses Blatt zählt nicht als Blatt der internationalen Anmeldung und ist nicht Teil derselben)

0 0-1	**Vom Anmeldeamt auszufüllen** Internationales Aktenzeichen.				
0-2	Eingangsstempel des Anmeldeamts				
0-4 0-4-1	**Formular - PCT/RO/101 (Anlage) PCT Blatt für die Gebührenberechnung** erstellt durch Benutzung von	PCT-EASY Version 2.91 (aktualisiert 01.07.2000)			
0-9	**Aktenzeichen des Anmelders oder Anwalts**	2000P01063WO			
2	**Anmelder**	SIEMENS AKTIENGESELLSCHAFT, et al.			
12	**Berechnung der vorgeschriebenen Gebühren**		Höhe der Gebühr/Multiplikator	Gesamtbeträge (DEM)	
12-1	Übermittlungsgebühr **T**		⇨	175	
12-2	Recherchengebühr **S**		⇨	1.848.26	
12-3	Internationale Gebühr Grundgebühr (erste 30 Blätter) **b1**		799.93		
12-4	Anzahl der Blätter über 30		0		
12-5	Zusatzblattgebühr **(X)**		17.6		
12-6	Gesamtbetrag der weiteren Gebühren **b2**		0		
12-7	b1 + b2 = **B**		799.93		
12-8	Bestimmungsgebühren Anzahl der in der internationalen Anmeldung vorgenommenen Bestimmungen		4		
12-9	Anzahl der zu zahlenden Bestimmungsgebühren (höchstens 8)		4		
12-10	Bestimmungsgebühr **(X)**		172.11		
12-11	Gesamtbetrag der Bestimmungsgebühren **D**		688.44		
12-12	PCT-EASY-Gebührenermäßigung **R**		-246.43		
12-13	Gesamtbetrag der internationalen Gebühr (B+D-R) **I**		⇨	1.241.94	
12-14	Gebühr für Prioritätsbeleg Anzahl der beantragten Prioritätsbelege		1		
12-15	Gebühr per Prioritätsbeleg **(X)**		35		
12-16	Gesamtbetrag Gebühr für Prioritätsbeleg(e) **P**		⇨	35	
12-17	**Gesamtbetrag der zu zahlenden Gebühren (T+S+I+P)**		⇨	3.300.2	
12-19	**Zahlungsart**	Sonstige: Abbuchung durch gesonderte Zahlungsliste			
12-20	**Anweisungen betreffend laufendes Konto** Das Anmeldeamt:	Deutsches Patent- und Markenamt (RO/DE)			

THIS PAGE BLANK (USPTO)

PCT (ANHANG - BLATT FÜR DIE GEBÜHRENBERECHNUNG)

2000P01063WO

Original (für **EINREICHUNG**) - gedruckt am 13.11.2000 09:21:25 AM

12-20-3	wird beauftragt, die Gebühr für die Ausstellung des Prioritätsbelegs (der Prioritätsbelege) und seine Übermittlung an das Internationale Büro der WIPO von meinem laufenden Konto abzubuchen	✓
12-21	Nummer des laufenden Kontos	409 022 601
12-22	Datum	13 November 2000 (13.11.2000)
12-23	Name und Unterschrift	SIEMENS AKTIENGESELLSCHAFT i.V. Haslbeck

PRÜFPROTOKOLL UND BEMERKUNGEN

13-2-1	**Prüfergebnisse Antrag**	Grün? Die Bezeichnung der Erfindung muß kurz und genau gefaßt sein. Bitte überprüfen.
13-2-2	**Prüfergebnisse Staaten**	Grün? Es können mehr Bestimmungen vorgenommen werden. Die folgenden Staaten sind nicht bestimmt worden: AP:(GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW); EA:(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM); OA:(BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG); AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CH, LI, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW. Bitte überprüfen.
13-2-6	**Prüfergebnisse Inhalt**	Grün? Nr. der Abb. der Zeichnungen, die mit der Zusammenf. veröffentlicht werden soll, nicht angegeben. Bitte überpüfen.
13-2-7	**Prüfergebnisse Gebühren**	Grün? Bitte bestätigen, daß das Gebührenverzeichnis in der zur Zeit geltenden Fassung benutzt wurde

OIPE JC98 JUL 18 2002 PATENT & TRADEMARK OFFICE

THIS PAGE BLANK (USPTO)

VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS

Absender: INTERNATIONALE RECHERCHENBEHÖRDE

PCT

MITTEILUNG ÜBER DIE ÜBERMITTLUNG DES INTERNATIONALEN RECHERCHENBERICHTS ODER DER ERKLÄRUNG

(Regel 44.1 PCT)

An
SIEMENS AG
Postfach 22 16 34
D-80506 München
GERMANY

CT AM Mch P/Ri
Eing. 18. Okt. 2001
GR
Frist

PRUF gesteht

Absendedatum *(Tag/Monat/Jahr)* 18/10/2001	
Aktenzeichen des Anmelders oder Anwalts 2000P01063WO	**WEITERES VORGEHEN** siehe Punkte 1 und 4 unten
Internationales Aktenzeichen PCT/DE 01/00021	Internationales Anmeldedatum *(Tag/Monat/Jahr)* 05/01/2001
Anmelder SIEMENS AG	

1. [X] Dem Anmelder wird mitgeteilt, daß der internationale Recherchenbericht erstellt wurde und ihm hiermit übermittelt wird.

Einreichung von Änderungen und einer Erklärung nach Artikel 19:
Der Anmelder kann auf eigenen Wunsch die Ansprüche der internationalen Anmeldung ändern (siehe Regel 46):

Bis wann sind Änderungen einzureichen?

Die Frist zur Einreichung solcher Änderungen beträgt üblicherweise zwei Monate ab der Übermittlung des internationalen Recherchenberichts; weitere Einzelheiten sind den Anmerkungen auf dem Beiblatt zu entnehmen.

Wo sind Änderungen einzureichen?

Unmittelbar beim Internationalen Büro der WIPO, 34, CHEMIN des Colombettes, CH-1211 Genf 20, Telefaxnr.: (41-22) 740.14.35

Nähere Hinweise sind den Anmerkungen auf dem Beiblatt zu entnehmen.

2. [ ] Dem Anmelder wird mitgeteilt, daß kein internationaler Recherchenbericht erstellt wird und daß ihm hiermit die Erklärung nach Artikel 17(2)a) übermittelt wird.

3. [ ] **Hinsichtlich des Widerspruchs** gegen die Entrichtung einer zusätzlichen Gebühr (zusätzlicher Gebühren) nach Regel 40.2 wird dem Anmelder mitgeteilt, daß

[ ] der Widerspruch und die Entscheidung hierüber zusammen mit seinem Antrag auf Übermittlung des Wortlauts sowohl des Widerspruchs als auch der Entscheidung hierüber an die Bestimmungsämter dem Internationalen Büro übermittelt worden sind.

[ ] noch keine Entscheidung über den Widerspruch vorliegt; der Anmelder wird benachrichtigt, sobald eine Entscheidung getroffen wurde.

4. **Weiteres Vorgehen:** Der Anmelder wird auf folgendes aufmerksam gemacht:

Kurz nach Ablauf von **18 Monaten** seit dem Prioritätsdatum wird die internationale Anmeldung vom Internationalen Büro veröffentlicht. Will der Anmelder die Veröffentlichung verhindern oder auf einen späteren Zeitpunkt verschieben, so muß gemäß Regel 90bis.1 bzw. 90bis.3 vor Abschluß der technischen Vorbereitungen für die internationale Veröffentlichung eine Erklärung über die Zurücknahme der internationalen Anmeldung oder des Prioritätsanspruchs beim Internationalen Büro eingehen.

Innerhalb von **19 Monaten** seit dem Prioritätsdatum ist ein Antrag auf internationale vorläufige Prüfung einzureichen, wenn der Anmelder den Eintritt in die nationale Phase bis zu 30 Monaten seit dem Prioritätsdatum (in manchen Ämtern sogar noch länger) verschieben möchte.

Innerhalb von **20 Monaten** seit dem Prioritätsdatum muß der Anmelder die für den Eintritt in die nationale Phase vorgeschriebenen Handlungen vor allen Bestimmungsämtern vornehmen, die nicht innerhalb von 19 Monaten seit dem Prioritätsdatum in der Anmeldung oder einer nachträglichen Auswahlerklärung ausgewählt wurden oder nicht ausgewählt werden konnten, da für sie Kapitel II des Vertrages nicht verbindlich ist.

Name und Postanschrift der Internationalen Recherchenbehörde Europäisches Patentamt, P.B. 5818 Patentlaan 2 NL-2280 HV Rijswijk Tel. (+31-70) 340-2040 Fax: (+31-70) 340-3016	Bevollmächtigter Bediensteter Theresia Van Deursen

Formblatt PCT/ISA/220 (Juli 1998)

(Siehe Anmerkungen auf Beiblatt)

THIS PAGE BLANK (USPTO)

ANMERKUNGEN ZU FORMBLATT PCT/ISA/220

Diese Anmerkungen sollen grundlegende Hinweise zur Einreichung von Änderungen gemäß Artikel 19 geben. Diesen Anmerkungen liegen die Erfordernisse des Vertrags über die internationale Zusammenarbeit auf dem Gebiet des Patentwesens (PCT), der Ausführungsordnung und der Verwaltungsrichtlinien zu diesem Vertrag zugrunde. Bei Abweichungen zwischen diesen Anmerkungen und obengenannten Texten sind letztere maßgebend. Nähere Einzelheiten sind dem PCT-Leitfaden für Anmelder, einer Veröffentlichung der WIPO, zu entnehmen.
Die in diesen Anmerkungen verwendeten Begriffe "Artikel", "Regel" und "Abschnitt" beziehen sich jeweils auf die Bestimmungen des PCT-Vertrags, der PCT-Ausführungsordnung bzw. der PCT-Verwaltungsrichtlinien.

HINWEISE ZU ÄNDERUNGEN GEMÄSS ARTIKEL 19

Nach Erhalt des internationalen Recherchenberichts hat der Anmelder die Möglichkeit, einmal die Ansprüche der internationalen Anmeldung zu ändern. Es ist jedoch zu betonen, daß, da alle Teile der internationalen Anmeldung (Ansprüche, Beschreibung und Zeichnungen) während des internationalen vorläufigen Prüfungsverfahrens geändert werden können, normalerweise keine Notwendigkeit besteht, Änderungen der Ansprüche nach Artikel 19 einzureichen, außer wenn der Anmelder z.B. zum Zwecke eines vorläufigen Schutzes die Veröffentlichung dieser Anspruche wünscht oder ein anderer Grund für eine Änderung der Ansprüche vor ihrer internationalen Veröffentlichung vorliegt. Weiterhin ist zu beachten, daß ein vorläufiger Schutz nur in einigen Staaten erhältlich ist.

Welche Teile der internationalen Anmeldung können geändert werden?

Im Rahmen von Artikel 19 können nur die Ansprüche geändert werden.

In der internationalen Phase können die Ansprüche auch nach Artikel 34 vor der mit der internationalen vorläufigen Prüfung beauftragten Behörde geändert (oder nochmals geändert) werden. Die Beschreibung und die Zeichnungen können nur nach Artikel 34 vor der mit der internationalen vorläufigen Prüfung beauftragten Behörde geändert werden.

Beim Eintritt in die nationale Phase können alle Teile der internationalen Anmeldung nach Artikel 28 oder gegebenenfalls Artikel 41 geändert werden.

Bis wann sind Änderungen einzureichen?

Innerhalb von zwei Monaten ab der Übermittlung des internationalen Recherchenberichts oder innerhalb von sechzehn Monaten ab dem Prioritätsdatum, je nachdem, welche Frist später abläuft. Die Änderungen gelten jedoch als rechtzeitig eingereicht, wenn sie dem Internationalen Büro nach Ablauf der maßgebenden Frist, aber noch vor Abschluß der technischen Vorbereitungen für die internationale Veröffentlichung (Regel 46.1) zugehen.

Wo sind die Änderungen nicht einzureichen?

Die Änderungen können nur beim Internationalen Büro, nicht aber beim Anmeldeamt oder der Internationalen Recherchenbehörde eingereicht werden (Regel 46.2).

Falls ein Antrag auf internationale vorläufige Prüfung eingereicht wurde/wird, siehe unten.

In welcher Form können Änderungen erfolgen?

Eine Änderung kann erfolgen durch Streichung eines oder mehrerer ganzer Ansprüche, durch Hinzufügung eines oder mehrerer neuer Ansprüche oder durch Änderung des Wortlauts eines oder mehrerer Ansprüche in der eingereichten Fassung.

Für jedes Anspruchsblatt, das sich aufgrund einer oder mehrerer Änderungen von dem ursprünglich eingereichten Blatt unterscheidet, ist ein Ersatzblatt einzureichen.

Alle Ansprüche, die auf einem Ersatzblatt erscheinen, sind mit arabischen Ziffern zu numerieren. Wird ein Anspruch gestrichen, so brauchen, die anderen Ansprüche nicht neu numeriert zu werden. Im Fall einer Neunumerierung sind die Ansprüche fortlaufend zu numerieren (Verwaltungsrichtlinien, Abschnitt 205 b)).

Die Änderungen sind in der Sprache abzufassen, in der dieinternationale Anmeldung veröffentlicht wird.

Welche Unterlagen sind den Änderungen beizufügen?

Begleitschreiben (Abschnitt 205 b)):

Die Änderungen sind mit einem Begleitschreiben einzureichen.

Das Begleitschreiben wird nicht zusammen mit der internationalen Anmeldung und den geänderten Ansprüchen veröffentlicht. Es ist nicht zu verwechseln mit der "Erklärung nach Artikel 19(1)" (siehe unten, "Erklärung nach Artikel 19 (1)").

Das Begleitschreiben ist nach Wahl des Anmelders in englischer oder französischer Sprache abzufassen. Bei englischsprachigen internationalen Anmeldungen ist das Begleitschreiben aber ebenfalls in englischer, bei französischsprachigen internationalen Anmeldungen in französischer Sprache abzufassen.

BNSDOCID: <XSISA220NODEP4_1_>

THIS PAGE BLANK (USPTO)

ANMERKUNGEN ZU FORMBLATT PCT/ISA/220 (Fortsetzung)

Im Begleitschreiben sind die Unterschiede zwischen den Ansprüchen in der eingereichten Fassung und den geänderten Ansprüchen anzugeben. So ist insbesondere zu jedem Anspruch in der internationalen Anmeldung anzugeben (gleichlautende Angaben zu verschiedenen Ansprüchen können zusammengefaßt werden), ob

i) der Anspruch unverändert ist;

ii) der Anspruch gestrichen worden ist;

iii) der Anspruch neu ist;

iv) der Anspruch einen oder mehrere Ansprüche in der eingereichten Fassung ersetzt;

v) der Anspruch auf die Teilung eines Anspruchs in der eingereichten Fassung zurückzuführen ist.

Im folgenden sind Beispiele angegeben, wie Änderungen im Begleitschreiben zu erläutern sind:

1. [Wenn anstelle von ursprünglich 48 Ansprüchen nach der Änderung einiger Ansprüche 51 Ansprüche existieren]:
"Die Ansprüche 1 bis 29, 31, 32, 34, 35, 37 bis 48 werden durch geänderte Ansprüche gleicher Numerierung ersetzt; Ansprüche 30, 33 und 36 unverändert; neue Ansprüche 49 bis 51 hinzugefügt."

2. [Wenn anstelle von ursprünglich 15 Ansprüchen nach der Änderung aller Ansprüche 11 Ansprüche existieren]:
"Geänderte Ansprüche 1 bis 11 treten an die Stelle der Ansprüche 1 bis 15."

3. [Wenn ursprünglich 14 Ansprüche existierten und die Änderungen darin bestehen, daß einige Ansprüche gestrichen werden und neue Ansprüche hinzugefügt werden]:
Ansprüche 1 bis 6 und 14 unverändert; Ansprüche 7 bis 13 gestrichen; neue Ansprüche 15, 16 und 17 hinzugefügt."Oder" Ansprüche 7 bis 13 gestrichen; neue Ansprüche 15, 16 und 17 hinzugefügt; alle übrigen Ansprüche unverändert."

4. [Wenn verschiedene Arten von Änderungen durchgeführt werden]:
"Ansprüche 1-10 unverändert; Ansprüche 11 bis 13, 18 und 19 gestrichen; Ansprüche 14, 15 und 16 durch geänderten Anspruch 14 ersetzt; Anspruch 17 in geänderte Ansprüche 15, 16 und 17 unterteilt; neue Ansprüche 20 und 21 hinzugefügt."

"Erklärung nach Artikel 19(1)" (Regel 46.4)

Den Änderungen kann eine Erklärung beigefügt werden, mit der die Änderungen erläutert und ihre Auswirkungen auf die Beschreibung und die Zeichnungen dargelegt werden (die nicht nach Artikel 19 (1) geändert werden können).

Die Erklärung wird zusammen mit der internationalen Anmeldung und den geänderten Ansprüchen veröffentlicht.

Sie ist in der Sprache abzufassen, in der die internationale Anmeldung veröffentlicht wird.

Sie muß kurz gehalten sein und darf, wenn in englischer Sprache abgefaßt oder ins Englische übersetzt, nicht mehr als 500 Wörter umfassen

Die Erklärung ist nicht zu verwechseln mit dem Begleitschreiben, das auf die Unterschiede zwischen den Ansprüchen in der eingereichten Fassung und den geänderten Ansprüchen hinweist, und ersetzt letzteres nicht. Sie ist auf einem gesonderten Blatt einzureichen und in der Überschrift als solche zu kennzeichnen, vorzugsweise mit den Worten "Erklärung nach Artikel 19 (1)".

Die Erklärung darf keine herabsetzenden Äußerungen über den internationalen Recherchenbericht oder die Bedeutung von in dem Bericht angeführten Veröffentlichungen enthalten. Sie darf auf im internationalen Recherchenbericht angeführte Veröffentlichungen, die sich auf einen bestimmten Anspruch beziehen, nur im Zusammenhang mit einer Änderung dieses Anspruchs Bezug nehmen.

Auswirkungen eines bereits gestellten Antrags auf internationale vorläufige Prüfung

Ist zum Zeitpunkt der Einreichung von Änderungen nach Artikel 19 bereits ein Antrag auf internationale vorläufige Prüfung gestellt worden, so sollte der Anmelder in seinem Interesse gleichzeitig mit der Einreichung der Änderungen beim Internationalen Büro auch eine Kopie der Änderungen bei der mit der internationalen vorläufigen Prüfung beauftragen Behörde einreichen (siehe Regel 62.2 a), erster Satz).

Auswirkungen von Änderungen hinsichtlich der Übersetzung der internationalen Anmeldung beim Eintritt in die nationale Phase

Der Anmelder wird darauf hingewiesen, daß bei Eintritt in die nationale Phase möglicherweise anstatt oder zusätzlich zu der Übersetzung der Ansprüche in der eingereichten Fassung eine Übersetzung der nach Artikel 19 geänderten Ansprüche an die bestimmten/ausgewählten Ämter zu übermitteln ist.

Nähere Einzelheiten über die Erfordernisse jedes bestimmten/ausgewählten Amts sind Band II des PCT-Leitfadens für Anmelder zu entnehmen.

BNSDOCID: <XSISA220NODEP4_I_>

THIS PAGE BLANK (USPTO)

VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS

PCT

INTERNATIONALER RECHERCHENBERICHT

(Artikel 18 sowie Regeln 43 und 44 PCT)

Aktenzeichen des Anmelders oder Anwalts	**WEITERES VORGEHEN**	siehe Mitteilung über die Übermittlung des internationalen Recherchenberichts (Formblatt PCT/ISA/220) sowie, soweit zutreffend, nachstehender Punkt 5
2000P01063WO		
Internationales Aktenzeichen	Internationales Anmeldedatum *(Tag/Monat/Jahr)*	(Frühestes) Prioritätsdatum *(Tag/Monat/Jahr)*
PCT/DE 01/ 00021	05/01/2001	18/01/2000
Anmelder		
SIEMENS AG		

Dieser internationale Recherchenbericht wurde von der Internationalen Recherchenbehörde erstellt und wird dem Anmelder gemäß Artikel 18 übermittelt. Eine Kopie wird dem Internationalen Büro übermittelt.

Dieser internationale Recherchenbericht umfaßt insgesamt 3 Blätter.

[X] Darüber hinaus liegt ihm jeweils eine Kopie der in diesem Bericht genannten Unterlagen zum Stand der Technik bei.

1. **Grundlage des Berichts**

   a. Hinsichtlich der **Sprache** ist die internationale Recherche auf der Grundlage der internationalen Anmeldung in der Sprache durchgeführt worden, in der sie eingereicht wurde, sofern unter diesem Punkt nichts anderes angegeben ist.

   [ ] Die internationale Recherche ist auf der Grundlage einer bei der Behörde eingereichten Übersetzung der internationalen Anmeldung (Regel 23.1 b)) durchgeführt worden.

   b. Hinsichtlich der in der internationalen Anmeldung offenbarten **Nucleotid– und/oder Aminosäuresequenz** ist die internationale Recherche auf der Grundlage des Sequenzprotokolls durchgeführt worden, das

   [ ] in der internationalen Anmeldung in Schriftlicher Form enthalten ist.

   [ ] zusammen mit der internationalen Anmeldung in computerlesbarer Form eingereicht worden ist.

   [ ] bei der Behörde nachträglich in schriftlicher Form eingereicht worden ist.

   [ ] bei der Behörde nachträglich in computerlesbarer Form eingereicht worden ist.

   [ ] Die Erklärung, daß das nachträglich eingereichte schriftliche Sequenzprotokoll nicht über den Offenbarungsgehalt der internationalen Anmeldung im Anmeldezeitpunkt hinausgeht, wurde vorgelegt.

   [ ] Die Erklärung, daß die in computerlesbarer Form erfaßten Informationen dem schriftlichen Sequenzprotokoll entsprechen, wurde vorgelegt.

2. [ ] **Bestimmte Ansprüche haben sich als nicht recherchierbar erwiesen** (siehe Feld I).

3. [ ] **Mangelnde Einheitlichkeit der Erfindung** (siehe Feld II).

4. Hinsichtlich der **Bezeichnung der Erfindung**

   [X] wird der vom Anmelder eingereichte Wortlaut genehmigt.

   [ ] wurde der Wortlaut von der Behörde wie folgt festgesetzt:

5. Hinsichtlich der **Zusammenfassung**

   [X] wird der vom Anmelder eingereichte Wortlaut genehmigt.

   [ ] wurde der Wortlaut nach Regel 38.2b) in der in Feld III angegebenen Fassung von der Behörde festgesetzt. Der Anmelder kann der Behörde innerhalb eines Monats nach dem Datum der Absendung dieses internationalen Recherchenberichts eine Stellungnahme vorlegen.

6. Folgende Abbildung der **Zeichnungen** ist mit der Zusammenfassung zu veröffentlichen: Abb. Nr. 2

   [ ] wie vom Anmelder vorgeschlagen

   [ ] keine der Abb.

   [X] weil der Anmelder selbst keine Abbildung vorgeschlagen hat.

   [ ] weil diese Abbildung die Erfindung besser kennzeichnet.

THIS PAGE BLANK (USPTO)

INTERNATIONALER RECHERCHENBERICHT

Internationales Aktenzeichen
PCT/DE 01/00021

A. KLASSIFIZIERUNG DES ANMELDUNGSGEGENSTANDES
IPK 7 H04L29/06

Nach der Internationalen Patentklassifikation (IPK) oder nach der nationalen Klassifikation und der IPK

B. RECHERCHIERTE GEBIETE

Recherchierter Mindestprüfstoff (Klassifikationssystem und Klassifikationssymbole)
IPK 7 H04L

Recherchierte aber nicht zum Mindestprüfstoff gehörende Veröffentlichungen, soweit diese unter die recherchierten Gebiete fallen

Während der internationalen Recherche konsultierte elektronische Datenbank (Name der Datenbank und evtl. verwendete Suchbegriffe)
EPO-Internal, WPI Data, PAJ, INSPEC, IBM-TDB, COMPENDEX

C. ALS WESENTLICH ANGESEHENE UNTERLAGEN

Kategorie°	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
X	WO 99 11019 A (V ONE CORP) 4. März 1999 (1999-03-04) Zusammenfassung Seite 1, Zeile 4 - Zeile 10 Seite 11, Zeile 3 - Zeile 15 Seite 15, Zeile 16 -Seite 16, Zeile 22 --- -/--	1-16

[X] Weitere Veröffentlichungen sind der Fortsetzung von Feld C zu entnehmen

[X] Siehe Anhang Patentfamilie

° Besondere Kategorien von angegebenen Veröffentlichungen :
"A" Veröffentlichung, die den allgemeinen Stand der Technik definiert, aber nicht als besonders bedeutsam anzusehen ist
"E" älteres Dokument, das jedoch erst am oder nach dem internationalen Anmeldedatum veröffentlicht worden ist
"L" Veröffentlichung, die geeignet ist, einen Prioritätsanspruch zweifelhaft erscheinen zu lassen, oder durch die das Veröffentlichungsdatum einer anderen im Recherchenbericht genannten Veröffentlichung belegt werden soll oder die aus einem anderen besonderen Grund angegeben ist (wie ausgeführt)
"O" Veröffentlichung, die sich auf eine mündliche Offenbarung, eine Benutzung, eine Ausstellung oder andere Maßnahmen bezieht
"P" Veröffentlichung, die vor dem internationalen Anmeldedatum, aber nach dem beanspruchten Prioritätsdatum veröffentlicht worden ist
"T" Spätere Veröffentlichung, die nach dem internationalen Anmeldedatum oder dem Prioritätsdatum veröffentlicht worden ist und mit der Anmeldung nicht kollidiert, sondern nur zum Verständnis des der Erfindung zugrundeliegenden Prinzips oder der ihr zugrundeliegenden Theorie angegeben ist
"X" Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann allein aufgrund dieser Veröffentlichung nicht als neu oder auf erfinderischer Tätigkeit beruhend betrachtet werden
"Y" Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann nicht als auf erfinderischer Tätigkeit beruhend betrachtet werden, wenn die Veröffentlichung mit einer oder mehreren anderen Veröffentlichungen dieser Kategorie in Verbindung gebracht wird und diese Verbindung für einen Fachmann naheliegend ist
"&" Veröffentlichung, die Mitglied derselben Patentfamilie ist

Datum des Abschlusses der internationalen Recherche	Absendedatum des internationalen Recherchenberichts
10. Oktober 2001	18/10/2001

Name und Postanschrift der Internationalen Recherchenbehörde	Bevollmächtigter Bediensteter
Europäisches Patentamt, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Adkhis, F

1

Meta 1002
p. 298

Formblatt PCT/ISA/210 (Blatt 2) (Juli 1992)

Internationales Aktenzeichen
PCT/DE 01/00021

C.(Fortsetzung) ALS WESENTLICH ANGESEHENE UNTERLAGEN

Kategorie°	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
X	HUNT R: "Internet/Intranet firewall security-policy, architecture and transaction services" COMPUTER COMMUNICATIONS, BUTTERWORTHS & CO. PUBLISHERS LTD, GB, Bd. 21, Nr. 13, 1. September 1998 (1998-09-01), Seiten 1107-1123, XP004146571 ISSN: 0140-3664 Zusammenfassung Seite 1113, rechte Spalte, Zeile 1 -Seite 1114, linke Spalte, Zeile 23 Seite 1119, linke Spalte, Zeile 28 -Seite 1120, linke Spalte, Zeile 39	1-16

THIS PAGE BLANK (USPTO)

INTERNATIONALER RECHERCHENBERICHT

Angaben zu Veröffentlichungen, die zur selben Patentfamilie gehören

Internationales Aktenzeichen
PCT/DE 01/00021

Im Recherchenbericht angeführtes Patentdokument		Datum der Veröffentlichung	Mitglied(er) der Patentfamilie		Datum der Veröffentlichung
WO 9911019	A	04-03-1999	US	6061796 A	09-05-2000
			AU	8915298 A	16-03-1999
			WO	9911019 A1	04-03-1999
			US	6158011 A	05-12-2000

Formblatt PCT/ISA/210 (Anhang Patentfamilie)(Juli 1992)

THIS PAGE BLANK (USPTO)

NK

VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS

PCT

INTERNATIONALER RECHERCHENBERICHT

(Artikel 18 sowie Regeln 43 und 44 PCT)

Aktenzeichen des Anmelders oder Anwalts	**WEITERES VORGEHEN**	siehe Mitteilung über die Übermittlung des internationalen Recherchenberichts (Formblatt PCT/ISA/220) sowie, soweit zutreffend, nachstehender Punkt 5
2000P01063WO		
Internationales Aktenzeichen	Internationales Anmeldedatum *(Tag/Monat/Jahr)*	(Frühestes) Prioritätsdatum *(Tag/Monat/Jahr)*
PCT/DE 01/ 00021	05/01/2001	18/01/2000
Anmelder		
SIEMENS AG		

Dieser internationale Recherchenbericht wurde von der Internationalen Recherchenbehörde erstellt und wird dem Anmelder gemäß Artikel 18 übermittelt. Eine Kopie wird dem Internationalen Büro übermittelt.

Dieser internationale Recherchenbericht umfaßt insgesamt 3 Blätter.

[X] Darüber hinaus liegt ihm jeweils eine Kopie der in diesem Bericht genannten Unterlagen zum Stand der Technik bei.

1. **Grundlage des Berichts**

   a. Hinsichtlich der **Sprache** ist die internationale Recherche auf der Grundlage der internationalen Anmeldung in der Sprache durchgeführt worden, in der sie eingereicht wurde, sofern unter diesem Punkt nichts anderes angegeben ist.

   [ ] Die internationale Recherche ist auf der Grundlage einer bei der Behörde eingereichten Übersetzung der internationalen Anmeldung (Regel 23.1 b)) durchgeführt worden.

   b. Hinsichtlich der in der internationalen Anmeldung offenbarten **Nucleotid- und/oder Aminosäuresequenz** ist die internationale Recherche auf der Grundlage des Sequenzprotokolls durchgeführt worden, das

   [ ] in der internationalen Anmeldung in Schriftlicher Form enthalten ist.

   [ ] zusammen mit der internationalen Anmeldung in computerlesbarer Form eingereicht worden ist.

   [ ] bei der Behörde nachträglich in schriftlicher Form eingereicht worden ist.

   [ ] bei der Behörde nachträglich in computerlesbarer Form eingereicht worden ist.

   [ ] Die Erklärung, daß das nachträglich eingereichte schriftliche Sequenzprotokoll nicht über den Offenbarungsgehalt der internationalen Anmeldung im Anmeldezeitpunkt hinausgeht, wurde vorgelegt.

   [ ] Die Erklärung, daß die in computerlesbarer Form erfaßten Informationen dem schriftlichen Sequenzprotokoll entsprechen, wurde vorgelegt.

2. [ ] **Bestimmte Ansprüche haben sich als nicht recherchierbar erwiesen** (siehe Feld I).

3. [ ] **Mangelnde Einheitlichkeit der Erfindung** (siehe Feld II).

4. Hinsichtlich der **Bezeichnung der Erfindung**

   [X] wird der vom Anmelder eingereichte Wortlaut genehmigt.

   [ ] wurde der Wortlaut von der Behörde wie folgt festgesetzt:

5. Hinsichtlich der **Zusammenfassung**

   [X] wird der vom Anmelder eingereichte Wortlaut genehmigt.

   [ ] wurde der Wortlaut nach Regel 38.2b) in der in Feld III angegebenen Fassung von der Behörde festgesetzt. Der Anmelder kann der Behörde innerhalb eines Monats nach dem Datum der Absendung dieses internationalen Recherchenberichts eine Stellungnahme vorlegen.

6. Folgende Abbildung der **Zeichnungen** ist mit der Zusammenfassung zu veröffentlichen: Abb. Nr. 2

   [ ] wie vom Anmelder vorgeschlagen [ ] keine der Abb.

   [X] weil der Anmelder selbst keine Abbildung vorgeschlagen hat.

   [ ] weil diese Abbildung die Erfindung besser kennzeichnet.

Meta 1002
p. 304

THIS PAGE BLANK (USPTO)

INTERNATIONALER RECHERCHENBERICHT

Internationales Aktenzeichen
[redacted]/DE 01/00021

A. KLASSIFIZIERUNG DES ANMELDUNGSGEGENSTANDES
IPK 7 H04L29/06

Nach der Internationalen Patentklassifikation (IPK) oder nach der nationalen Klassifikation und der IPK

B. RECHERCHIERTE GEBIETE

Recherchierter Mindestprüfstoff (Klassifikationssystem und Klassifikationssymbole)
IPK 7 H04L

Recherchierte aber nicht zum Mindestprüfstoff gehörende Veröffentlichungen, soweit diese unter die recherchierten Gebiete fallen

Während der internationalen Recherche konsultierte elektronische Datenbank (Name der Datenbank und evtl. verwendete Suchbegriffe)
EPO-Internal, WPI Data, PAJ, INSPEC, IBM-TDB, COMPENDEX

C. ALS WESENTLICH ANGESEHENE UNTERLAGEN

Kategorie°	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
X	WO 99 11019 A (V ONE CORP) 4. März 1999 (1999-03-04) Zusammenfassung Seite 1, Zeile 4 - Zeile 10 Seite 11, Zeile 3 - Zeile 15 Seite 15, Zeile 16 -Seite 16, Zeile 22 --- -/--	1-16

[X] Weitere Veröffentlichungen sind der Fortsetzung von Feld C zu entnehmen

[X] Siehe Anhang Patentfamilie

° Besondere Kategorien von angegebenen Veröffentlichungen :
"A" Veröffentlichung, die den allgemeinen Stand der Technik definiert, aber nicht als besonders bedeutsam anzusehen ist
"E" älteres Dokument, das jedoch erst am oder nach dem internationalen Anmeldedatum veröffentlicht worden ist
"L" Veröffentlichung, die geeignet ist, einen Prioritätsanspruch zweifelhaft erscheinen zu lassen, oder durch die das Veröffentlichungsdatum einer anderen im Recherchenbericht genannten Veröffentlichung belegt werden soll oder die aus einem anderen besonderen Grund angegeben ist (wie ausgeführt)
"O" Veröffentlichung, die sich auf eine mündliche Offenbarung, eine Benutzung, eine Ausstellung oder andere Maßnahmen bezieht
"P" Veröffentlichung, die vor dem internationalen Anmeldedatum, aber nach dem beanspruchten Prioritätsdatum veröffentlicht worden ist
"T" Spätere Veröffentlichung, die nach dem internationalen Anmeldedatum oder dem Prioritätsdatum veröffentlicht worden ist und mit der Anmeldung nicht kollidiert, sondern nur zum Verständnis des der Erfindung zugrundeliegenden Prinzips oder der ihr zugrundeliegenden Theorie angegeben ist
"X" Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann allein aufgrund dieser Veröffentlichung nicht als neu oder auf erfinderischer Tätigkeit beruhend betrachtet werden
"Y" Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann nicht als auf erfinderischer Tätigkeit beruhend betrachtet werden, wenn die Veröffentlichung mit einer oder mehreren anderen Veröffentlichungen dieser Kategorie in Verbindung gebracht wird und diese Verbindung für einen Fachmann naheliegend ist
"&" Veröffentlichung, die Mitglied derselben Patentfamilie ist

Datum des Abschlusses der internationalen Recherche	Absendedatum des internationalen Recherchenberichts
10. Oktober 2001	18/10/2001

Name und Postanschrift der Internationalen Recherchenbehörde	Bevollmächtigter Bediensteter
Europäisches Patentamt, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Adkhis, F

1

Meta 1002
p. 306

Formblatt PCT/ISA/210 (Blatt 2) (Juli 1992)

THIS PAGE BLANK (USPTO)

INTERNATIONALER RECHERCHENBERICHT

Internationales Aktenzeichen
[redacted]/DE 01/00021

C.(Fortsetzung) ALS WESENTLICH AN[redacted]EHENE UNTERLAGEN

Kategorie°	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
X	HUNT R: "Internet/Intranet firewall security-policy, architecture and transaction services" COMPUTER COMMUNICATIONS, BUTTERWORTHS & CO. PUBLISHERS LTD, GB, Bd. 21, Nr. 13, 1. September 1998 (1998-09-01), Seiten 1107-1123, XP004146571 ISSN: 0140-3664 Zusammenfassung Seite 1113, rechte Spalte, Zeile 1 -Seite 1114, linke Spalte, Zeile 23 Seite 1119, linke Spalte, Zeile 28 -Seite 1120, linke Spalte, Zeile 39 -----	1-16

1

Meta 1002
p. 308

Formblatt PCT/ISA/210 (Fortsetzung von Blatt 2) (Juli 1992)

THIS PAGE BLANK (USPTO)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
/DE 01/00021

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9911019 A	04-03-1999	US 6061796 A	09-05-2000
		AU 8915298 A	16-03-1999
		WO 9911019 A1	04-03-1999
		US 6158011 A	05-12-2000

THIS PAGE BLANK (USPTO)

THIS PAGE BLANK (USPTO)

VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS

Absender: DIE MIT DER INTERNATIONALEN VORLÄUFIGEN PRÜFUNG BEAUFTRAGTE BEHÖRDE

PCT

An

SIEMENS AG
Postfach 22 16 34
D-80506 München
ALLEMAGNE

CT IPS AM Mch PVR
Eing. 14. März 2002
GR Frist 18.05.02

MITTEILUNG ÜBER DIE ÜBERSENDUNG DES INTERNATIONALEN VORLÄUFIGEN PRÜFUNGSBERICHTS

(Regel 71.1 PCT)

Absendedatum *(Tag/Monat/Jahr)*	13/03/2002
Aktenzeichen des Anmelders oder Anwalts: 2000P01063WO	**WICHTIGE MITTEILUNG**

Internationales Aktenzeichen	Internationales Anmeldedatum *(Tag/Monat/Jahr)*	Prioritätsdatum *(Tag/Monat/Jahr)*
PCT/DE 01/00021	05/01/2001	18/01/2000

Anmelder: SIEMENS AG

1. Dem Anmelder wird mitgeteilt, daß ihm die mit der internationalen vorläufigen Prüfung beauftragte Behörde hiermit den zu der internationalen Anmeldung erstellten internationalen vorläufigen Prüfungsbericht, gegebenfalls mit den dazugehörigen Anlagen, übermittelt.

2. Eine Kopie des Berichts wird - gegebenfalls mit den dazugehörigen Anlagen - dem Internationalen Büro zur Weiterleitung an alle ausgewählten Ämter übermittelt.

3. Auf Wunsch eines ausgewählten Amts wird das Internationale Büro eine Übersetzung des Berichts (jedoch nicht der Anlagen) ins Englische anfertigen und diesem Amt übermitteln.

4. **ERINNERUNG**

Zum Eintritt in die nationale Phase hat der Anmelder vor jedem ausgewählten Amt innerhalb von 30 Monaten ab dem Prioritätsdatum (oder in manchen Ämtern noch später) bestimmte Handlungen (Einreichung von Übersetzungen und Entrichtung nationaler Gebühren) vorzunehmen (Artikel 39 (1)) (siehe auch die durch das Internationale Büro mit Formblatt PCT/IB/301 übermittelte Information).

Ist einem ausgewählten Amt eine Übersetzung der internationalen Anmeldung zu übermitteln, so muß diese Übersetzung auch Übersetzungen aller Anlagen zum internationalen vorläufigen Prüfungsbericht enthalten. Es ist Aufgabe des Anmelders, solche Übersetzungen anzufertigen und den betroffenen ausgewählten Ämtern direkt zuzuleiten.

Weitere Einzelheiten zu dem maßgebenden Fristen und Erfordernissen der ausgewählten Ämter sind Band II des PCT-Leitfadens für Anmelder zu entnehmen.

Name und Postanschrift der mit der internationalen vorläufigen Prüfung beauftragten Behörde
Europäisches Patentamt
D-80298 München
Tel. (+49-89) 2399-0, Tx: 523656 epmu d
Fax: (+49-89) 2399-4465

Bevollmächtigter Bediensteter
BENKAABA A
Tel. (+49-89) 2399 2828

THIS PAGE BLANK (USPTO)

VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS

PCT

INTERNATIONALER VORLÄUFIGER PRÜFUNGSBERICHT

(Artikel 36 und Regel 70 PCT)

Aktenzeichen des Anmelders oder Anwalts	WEITERES VORGEHEN	siehe Mitteilung über die Übersendung des internationalen vorläufigen Prüfungsberichts (Formblatt PCT/IPEA/416)
2000P01063WO		
Internationales Aktenzeichen	Internationales Anmeldedatum *(Tag/Monat/Jahr)*	Prioritätsdatum *(Tag/Monat/Jahr)*
PCT/DE 01/ 00021	05/01/2001	18/01/2000
Internationale Patentklassifikation (IPK) oder nationale Klassifikation und IPK	H04L29/00	
Anmelder	SIEMENS AG	

1. Der internationale vorläufige Prüfungsbericht wurde von der mit der internationalen vorläufigen Prüfung beauftragten Behörde erstellt und wird dem Anmelder gemäß Artikel 36 übermittelt.

2. Dieser BERICHT umfaßt insgesamt 2 Blätter einschließlich dieses Deckblatts.

☐ Außerdem liegen dem Bericht ANLAGEN bei; dabei handelt es sich um Blätter mit Beschreibungen, Ansprüchen und/oder Zeichnungen, die geändert wurden und diesem Bericht zugrunde liegen, und/oder Blätter mit vor dieser Behörde vorgenommenen Berichtigungen (siehe Regel 70.16 und Abschnitt 607 der Verwaltungsvorschriften zum PCT)

Diese Anlagen umfassen insgesamt _____ Blätter.

3. Dieser Bericht enthält Angaben und die entsprechenden Seiten zu folgenden Punkten:

- I ☒ Grundlage des Berichts
- II ☐ Priorität
- III ☒ Keine Erstellung eines Gutachtens über Neuheit, erfinderische Tätigkeit und gewerbliche Anwendbarkeit
- IV ☐ Mangelnde Einheitlichkeit der Erfindung
- V ☒ Begründete Feststellung nach Artikel 35(2) hinsichtlich der Neuheit, der erfinderischen Tätigkeit und der gewerblichen Anwendbarkeit; Unterlagen und Erklärungen zur Stützung dieser Feststellung
- VI ☐ Bestimmte angeführte Unterlagen
- VII ☐ Bestimmte Mängel der internationalen Anmeldung
- VIII ☐ Bestimmte Bemerkungen zur internationalen Anmeldung

Datum der Einreichung des Antrags	Datum der Fertigstellung dieses Berichts
06/08/2001	09/03/2002
Name und Postanschrift der mit der internationalen vorläufigen Prüfung beauftragten Behörde Europäisches Patentamt D-80298 München Tel. (+49-89) 2399-0, Tx: 523656 epmu d Fax: (+49-89) 2399-4465	Bevollmächtigter Bediensteter HEY S A Tel. (+49-89) 2399 2828

Formblatt PCT/IPEA/409 (Deckblatt)(Juli 1998)

EUROPÄISCHES PATENTAMT · EUROPEAN PATENT OFFICE · OFFICE EUROPÉEN DES BREVETS

THIS PAGE BLANK (USPTO)

I. Grundlage des Berichts

1. Grundlage dieses internationalen vorläufigen Prüfungsberichts ist die Anmeldung in der ursprünglich eingereichten Fassung.

III. Keine Erstellung eines Gutachtens über Neuheit, erfinderische Tätigkeit und gewerbliche Anwendbarkeit

2. Die Frage, ob die beanspruchte Erfindung als neu , auf erfinderischer Tätigkeit beruhend und gewerblich anwendbar anzusehen ist, war und wird nicht Gegenstand einer internationalen vorläufigen Prüfung (Art. 34 (4) a) i), ii) PCT; s.a. internationaler Recherchenbericht) für:

2.1 Anmeldungen mit unnötigen unabhängigen Ansprüchen (im allgemeinen ist nicht mehr als ein unabhängiger Anspruch pro Kategorie erforderlich; Artikel 6 PCT)

2.2 nicht recherchierte Gegenstände (Artikel 17 (2) a), Regel 66.1 e) PCT), z.B.
- a) beanspruchte Gegenstände nach Regel 39.1 PCT
- b) Anmeldungen, deren Beschreibung, Ansprüche oder Zeichnungen den vorgeschriebenen Anforderungen so wenig entsprechen, daß eine sinnvolle Recherche nicht durchgeführt werden kann.

2.3 beanspruchte Gegenstände nach Regel 67.1 PCT

V. Begründete Feststellung nach Artikel 35(2) hinsichtlich der Neuheit, der erfinderischen Tätigkeit und der gewerblichen Anwendbarkeit

3. Soweit die internationale vorläufige Prüfung durchgeführt wurde (s. Punkt III), ist folgendes anzumerken:

In Anbetracht der im internationalen Recherchenbericht angeführten Unterlagen wird festgestellt, daß die Erfindung, wie sie in mindestens einem unabhängigen Anspruch gekennzeichnet ist, die in Artikel 33 (1) PCT aufgeführten Kriterien allem Anschein nach nicht erfüllt, d.h. nicht als neu und/oder auf erfinderischer Tätigkeit beruhend anzusehen ist.

THIS PAGE BLANK (USPTO)

VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS

PCT

REC'D 15 MAR 2002
PCT

INTERNATIONALER VORLÄUFIGER PRÜFUNGSBERICHT

(Artikel 36 und Regel 70 PCT)

Aktenzeichen des Anmelders oder Anwalts	**WEITERES VORGEHEN**	siehe Mitteilung über die Übersendung des internationalen vorläufigen Prüfungsberichts (Formblatt PCT/IPEA/416)
2000P01063WO		
Internationales Aktenzeichen	Internationales Anmeldedatum *(Tag/Monat/Jahr)*	Prioritätsdatum *(Tag/Monat/Jahr)*
PCT/DE 01/ 00021	05/01/2001	18/01/2000
Internationale Patentklassifikation (IPK) oder nationale Klassifikation und IPK		
H04L29/00		
Anmelder		
SIEMENS AG		

1. Der internationale vorläufige Prüfungsbericht wurde von der mit der internationalen vorläufigen Prüfung beauftragten Behörde erstellt und wird dem Anmelder gemäß Artikel 36 übermittelt.

2. Dieser **BERICHT** umfaßt insgesamt 2 Blätter einschließlich dieses Deckblatts.

☐ Außerdem liegen dem Bericht **ANLAGEN** bei; dabei handelt es sich um Blätter mit Beschreibungen, Ansprüchen und/oder Zeichnungen, die geändert wurden und diesem Bericht zugrunde liegen, und/oder Blätter mit vor dieser Behörde vorgenommenen Berichtigungen (siehe Regel 70.16 und Abschnitt 607 der Verwaltungsvorschriften zum PCT)

Diese Anlagen umfassen insgesamt _____ Blätter.

3. Dieser Bericht enthält Angaben und die entsprechenden Seiten zu folgenden Punkten:

- I ☒ Grundlage des Berichts
- II ☐ Priorität
- III ☒ Keine Erstellung eines Gutachtens über Neuheit, erfinderische Tätigkeit und gewerbliche Anwendbarkeit
- IV ☐ Mangelnde Einheitlichkeit der Erfindung
- V ☒ Begründete Feststellung nach Artikel 35(2) hinsichtlich der Neuheit, der erfinderischen Tätigkeit und der gewerblichen Anwendbarkeit; Unterlagen und Erklärungen zur Stützung dieser Feststellung
- VI ☐ Bestimmte angeführte Unterlagen
- VII ☐ Bestimmte Mängel der internationalen Anmeldung
- VIII ☐ Bestimmte Bemerkungen zur internationalen Anmeldung

Datum der Einreichung des Antrags	Datum der Fertigstellung dieses Berichts
06/08/2001	09/03/2002
Name und Postanschrift der mit der internationalen vorläufigen Prüfung beauftragten Behörde	Bevollmächtigter Bediensteter
Europäisches Patentamt D-80298 München Tel. (+49-89) 2399-0, Tx: 523656 epmu d Fax: (+49-89) 2399-4465	HEY S A Tel. (+49-89) 2399 2828

EUROPÄISCHES PATENTAMT · EUROPEAN PATENT OFFICE · OFFICE EUROPÉEN DES BREVETS

THIS PAGE BLANK (USPTO)

I. Grundlage des Berichts

1. Grundlage dieses internationalen vorläufigen Prüfungsberichts ist die Anmeldung in der ursprünglich eingereichten Fassung.

III. Keine Erstellung eines Gutachtens über Neuheit, erfinderische Tätigkeit und gewerbliche Anwendbarkeit

2. Die Frage, ob die beanspruchte Erfindung als neu , auf erfinderischer Tätigkeit beruhend und gewerblich anwendbar anzusehen ist, war und wird nicht Gegenstand einer internationalen vorläufigen Prüfung (Art. 34 (4) a) i), ii) PCT; s.a. internationaler Recherchenbericht) für:

2.1 Anmeldungen mit unnötigen unabhängigen Ansprüchen (im allgemeinen ist nicht mehr als ein unabhängiger Anspruch pro Kategorie erforderlich; Artikel 6 PCT)

2.2 nicht recherchierte Gegenstände (Artikel 17 (2) a), Regel 66.1 e) PCT), z.B.
- a) beanspruchte Gegenstände nach Regel 39.1 PCT
- b) Anmeldungen, deren Beschreibung, Ansprüche oder Zeichnungen den vorgeschriebenen Anforderungen so wenig entsprechen, daß eine sinnvolle Recherche nicht durchgeführt werden kann.

2.3 beanspruchte Gegenstände nach Regel 67.1 PCT

V. Begründete Feststellung nach Artikel 35(2) hinsichtlich der Neuheit, der erfinderischen Tätigkeit und der gewerblichen Anwendbarkeit

3. Soweit die internationale vorläufige Prüfung durchgeführt wurde (s. Punkt III), ist folgendes anzumerken:

In Anbetracht der im internationalen Recherchenbericht angeführten Unterlagen wird festgestellt, daß die Erfindung, wie sie in mindestens einem unabhängigen Anspruch gekennzeichnet ist, die in Artikel 33 (1) PCT aufgeführten Kriterien allem Anschein nach nicht erfüllt, d.h. nicht als neu und/oder auf erfinderischer Tätigkeit beruhend anzusehen ist.

THIS PAGE BLANK (USPTO)

(12) NACH DEM VERTRAG [redacted]R DIE INTERNATIONALE ZUSAMMENAR[redacted] AUF DEM GEBIET DES PATENTWESENS (PCT) VERÖFFENTLICHTE INTERNATIONALE ANMELDUNG

(19) Weltorganisation für geistiges Eigentum
Internationales Büro

(43) Internationales Veröffentlichungsdatum
26. Juli 2001 (26.07.2001)

PCT

(10) Internationale Veröffentlichungsnummer
WO 01/54371 A2

(51) Internationale Patentklassifikation[7]: H04L 29/00

(21) Internationales Aktenzeichen: PCT/DE01/00021

(22) Internationales Anmeldedatum:
5. Januar 2001 (05.01.2001)

(25) Einreichungssprache: Deutsch

(26) Veröffentlichungssprache: Deutsch

(30) Angaben zur Priorität:
100 01 855.6 18. Januar 2000 (18.01.2000) DE

(71) Anmelder *(für alle Bestimmungsstaaten mit Ausnahme von US)*: **SIEMENS AKTIENGESELLSCHAFT** [DE/DE]; Wittelsbacherplatz 2, 80333 München (DE).

(72) Erfinder; und
(75) Erfinder/Anmelder *(nur für US)*: **EUCHNER, Martin** [DE/DE]; Lorenzstrasse 2, 81737 München (DE).

(74) Gemeinsamer Vertreter: **SIEMENS AKTIENGESELLSCHAFT**; Postfach 22 16 34, 80506 München (DE).

(81) Bestimmungsstaaten *(national)*: CA, JP, US.

(84) Bestimmungsstaaten *(regional)*: europäisches Patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Veröffentlicht:
— *ohne internationalen Recherchenbericht und erneut zu veröffentlichen nach Erhalt des Berichts*

Zur Erklärung der Zweibuchstaben-Codes, und der anderen Abkürzungen wird auf die Erklärungen ("Guidance Notes on Codes and Abbreviations") am Anfang jeder regulären Ausgabe der PCT-Gazette verwiesen.

(54) Title: METHOD, SYSTEM FOR TRANSMITTING DATA FROM A TRANSMITTER TO A RECEIVER AND TRANSMITTER OR RECEIVER

(54) Bezeichnung: VERFAHREN, SYSTEM ZUR ÜBERMITTLUNG VON DATEN VON EINEM SENDER ZU EINEM EMPFÄNGER UND SENDER BZW. EMPFÄNGER HIERZU

(57) Abstract: A method and system for transmitting data from a transmitter to a receiver, wherein data from a transmitter is extended to include authentication data on the application level by means an application protocol. Said authentication data is used by the receiver to determine whether the transmitter is known by the receiver. If the transmitter is known by the receiver, the data is accepted. If not, the data is rejected.

(57) Zusammenfassung: Es wird ein Verfahren zur Übermittlung von Daten von einem Sender zu einem Empfänger angegeben, bei dem von dem Sender die Daten mittels eines Anwendungsprotokolls auf der Anwendungsschicht um Authentisierungsdaten erweitert werden. Anhand der Authentisierungsdaten wird von dem Empfänger ermittelt, ob er den Sender kennt. Falls der Empfänger den Sender kennt, werden die Daten entgegengenommen, ansonsten werden die Daten verworfen.

WO 01/54371 A2

THIS PAGE BLANK (USPTO)

Beschreibung

Verfahren, System zur Übermittlung von Daten von einem Sender zu einem Empfänger und Sender bzw. Empfänger hierzu

Die Erfindung betrifft ein Verfahren und ein System und zur Übermittlung von Daten von einem Sender zu einem Empfänger und den Sender bzw. den Empfänger hierzu.

Die Übermittlung von Daten ist z.B. bekannt nach dem OSI-Referenzmodell [1]. Das OSI-Referenzmodell (kurz: OSI-Modell) der International Standards Organisation (ISO) umfaßt sieben Schichten, deren jede eine (bezogen auf die Abstraktion) unterschiedliche Funktionalität aufweist. Im OSI-Modell entspricht Schicht 1 einer Bitübertragungsschicht, dort erfolgt von Sender zu Empfänger eine Übermittlung von Daten bzw. Nachrichten anhand eines Bitübertragungsprotokolls. In Schicht 2 ist eine Sicherungsschicht vorgesehen, zur Übertragung von Schicht 2 des Senders zur Schicht 2 des Empfängers wird ein Sicherungsprotokoll verwendet. Analog werden in Schicht 3, einer Vermittlungsschicht, zwischen Sender und Empfänger ein Vermittlungsprotokoll und in Schicht 4, einer Transportschicht, ein Transportprotokoll eingesetzt. Im OSI-Modell entsprechen die Schicht 5 einer Sitzungsschicht mit einem Sitzungsprotokoll, Schicht 6 einer Darstellungsschicht mit einem Darstellungsprotokoll und Schicht 7 einer Anwendungsschicht mit einem Anwendungsprotokoll. In der Praxis ist es im Rahmen vieler Anwendungen nicht immer möglich, strenge Grenzen gerade zwischen den oberen Protokollschichten, insbesondere den Schichten 5 bis 7, exakt zu benennen. Beispielsweise im Rahmen der Internet-Telefonie, d.h. der Wahrnehmung des Telefondienstes über das Medium Internet, verschmelzen die drei Schichten oberhalb der Transportschicht 4 zu einer "Anwendungsschicht", für das analog obiger Ausführung ein "Anwendungsprotokoll" eingesetzt wird.

Eine Besonderheit des OSI-Modells und damit aller an das OSI-Modell angelehnter Kommunikationsmodelle ist es, daß durch die Unterteilung in Schichten in jeder Schicht die unteren Schichten völlig transparent zu der aktuell betrachteten Schicht Funktionen wahrnehmen und an diese obere Schicht einen Dienst erbringen, der durch die Gesamtheit der Funktionalitäten der unteren Schichten bestimmt ist. "Transparent" bedeutet in diesem Zusammenhang, daß sich die aktuell betrachtete Schicht nicht um die Funktionalitäten der unteren Schichten kümmern muß. Im hier besprochenen Fall kann also das Anwendungsprotokoll zwischen einem Sender der Anwendungsschicht und einem Empfänger der Anwendungsschicht eingesetzt werden. Darunter mögen zahlreiche Dienste, beispielsweise zur Vermittlung, zur Sicherung oder zur tatsächlichen Übertragung über einen physikalischen Kanal, liegen, um die man sich aus Sicht des Anwendungsprotokolls, insbesondere als Anwender dieses Protokolls, nicht mehr kümmern muß.

Analog zu dieser Betrachtung gibt es zwischen den einzelnen Schichten sogenannte Protokolldateneinheiten (engl.: Protocol Data Unit, PDU), die für jedes Protokoll der jeweiligen Schicht eigens ausgeführt sein können. So kann für jede Schicht in dem jeweiligen Protokoll ein eigener Header vorgesehen werden, der administrative Informationen für das Protokoll der jeweiligen Schicht enthält, wobei gemäß OSI-Modell dieser Header auch nur für die Schicht des jeweiligen Protokolls sichtbar und verwendbar ist. Details zu der Konzeption des OSI-Modells finden sich in zahlreichen Literaturstellen, unter anderem der oben genannten.

Wird nachfolgend von Anwendungsschicht gesprochen, so sind damit die Schichten oberhalb der Transportschicht (Schichten größer 4) bezeichnet. Das Anwendungsprotokoll bezeichnet ferner das Protokoll für die Kommunikation zwischen einer oberhalb der Transportschicht liegenden Anwendungsschicht

eines Senders und einer oberhalb der Transportschicht angesiedelten Anwendungsschicht eines Empfängers.

Allgemein bekannt ist ein Message Authentication Code (MAC), der eine kryptographische Prüfsumme bezeichnet, anhand derer eine Veränderung in einer Nachricht bzw. in Daten erkannt werden soll (siehe [2], S.61ff.).

Eine Einweg-Hash-Funktion ist beispielsweise aus [2], S.68ff., bekannt. Mit solch einer Einweg-Hash-Funktion ist es nicht möglich, zu einem gegebenen Funktionswert den passenden Eingabewert zu berechnen. Ein anderes Merkmal ist die Kollisionsfreiheit, d.h. es darf nicht möglich sein, zu zwei verschiedenen Eingabewerten den gleichen Ausgabewert mittels der Einweg-Hash-Funktion zu erhalten.

Ein Unterschied zwischen dem Hash-Code und dem MAC besteht darin, daß der MAC zu seiner Berechnung einen geheimen Schlüssel erfordert, wohingegen die Hash-Funktion schlüsselunabhängig und öffentlich bekannt sein kann.

Weiterhin ist ein asymmetrisches kryptographisches Verfahren (auch: Public-Key-Verfahren) bekannt, z.B. aus [2], S.73ff. Jeder Teilnehmer an einem asymmetrischen kryptographischen Verfahren erhält zwei Schlüssel, einen öffentlichen und einen geheimen Schlüssel. Grundsätzlich ist der geheime Schlüssel aus dem öffentlichen Schlüssel ableitbar, wobei diese Aufgabe so komplex wie möglich sein soll. Mit dem asymmetrischen Verfahren kann weiterhin eine elektronische Unterschrift (Authentifikation) vorgenommen und/oder der Inhalt einer Nachricht (mit dem öffentlichen Schlüssel des Empfängers) so verschlüsselt werden, daß nur der Empfänger mit seinem geheimen Schlüssel die Nachricht wieder entschlüsseln kann.

Schließlich gibt es auch ein symmetrisches Verfahren, das einen (geheimen) Schlüssel erfordert, der sowohl zur Ver- als auch Entschlüsselung eingesetzt wird. Ein Beispiel für ein

symmetrisches Verschlüsselungsverfahren ist der DES-Algorithmus.

Ein Kommunikationssystem unterliegt einer Vielzahl von möglichen Angriffen, die u.a. auf den Inhalt der im Kommunikationssystem ausgetauschten Nachrichten oder auf die Verfügbarkeit des Kommunikationssystems abzielen können. Nimmt man das Beispiel der Internet-Telefonie, so ist es zum einen von Bedeutung, daß der Inhalt des Telefonats nicht von einem unberechtigten Dritten mitgehört werden kann, zum anderen ist es auch erforderlich, daß der Dritte nicht zahllose Anrufe initiieren kann und somit den Empfänger stört und das Kommunikationssystem unnötig belastet. Solche Angriffe werden auch als Denial-of-Service-Angriffe bezeichnet. Mögliche Beispiele für solche Angriffe sind Massendaten bzw. Massennachrichten, die automatisch erzeugt und zu einem oder mehreren Empfängern geschickt und dort Verfügbarkeit und Performanz in erheblichem Umfang belasten. Im Beispiel Internet-Telefonie ist es zumindest unerwünscht, daß unablässig ein Internet-Telefon klingelt und somit neben der unnötigen Netzlast auch noch die Aufmerksamkeit des Empfängers abgelenkt wird. Ebenso wird es als störend empfunden, wenn ein Angreifer durch das Senden unerlaubter Sprachdaten den Kommunikationsfluß zwischen Sender und Empfänger stört.

Zur Abwehr von Angriffen auf ein privates Kommunikationsnetz werden vielfach sogenannte "Firewalls" eingesetzt, die insbesondere eine Abgrenzung des privaten Kommunikationsnetzes von öffentlichen Netzen sicherstellen. Allerdings ist es für einen Angreifer leicht möglich, seine unsinnigen (Massen)Daten auch an Adressaten innerhalb des privaten Netzes, also hinter die Firewall, zu senden. Dort werden diese Daten decodiert und gegebenenfalls wiedergegeben. Dadurch erreicht der Angreifer, daß bspw. bei der Wiedergabe solcher unsinniger Daten nur lästige Tonstörungen oder Rauschen wahrgenommen und damit das private

Netz in seiner Bandbreite und Verfügbarkeit eingeschränkt wird. Im Extremfall ist auch ein Absturz eines Teils oder des gesamten privaten Kommunikationsnetzes möglich.

Ferner ist ein RTP-Protokoll für die Übertragung von Mediendaten (sog. "Payload"), d.h. von Bild- oder Tondaten, , bekannt [3].

Ein Ansatz zur Abhaltung von unerwünschten Daten liefert das IPSEC-Protokoll [5] bzw. [6]. Hier können Datenpakete des Internetprotokolls (IP-Datenpakete) eingekapselt und bezüglich Vertraulichkeit und/oder Integrität (implizite Absenderauthentifikation) geschützt werden. Ferner bietet das IPSEC-Protokoll ein Keymanagement-Verfahren mit einem sogenannten Cookie-Mechanismus (siehe [4]), anhand dessen die oben diskutierten Massendatenangriffe (Denial-of-Service-Angriffe) während der Keymanagement-Phase abgewehrt werden können. Bei dem Cookie-Mechanismus werden schnelle Einweg-Hash-Funktionen (z.B. SHA-1), Zufallszahlen und IP-Adressen miteinander verknüpft. Allerdings ist das erste "Cookie", das vom Sender zum Empfänger übertragen wird, nicht abgesichert, wodurch eine Sicherheitslücke entsteht. Ferner ist das IPSEC-Cookie-Verfahren nicht geeignet, um auf der Anwendungsschicht (für das Anwendungsprotokoll) auch unter Echtzeitbedingungen einen Schutz gegen derartige unerwünschte Massendaten zu gewährleisten. Nachteilig für IPSEC ist wegen der OSI Schichtengliederung allerdings unter anderem, daß eine Verknüpfung von IPSEC mit den Sicherheitsfunktionen einer Anwendungsschicht sich so nicht bewerkstelligen läßt.

Die **Aufgabe** der Erfindung besteht darin, Daten von einem Sender zu einem Empfänger zu übertragen, wobei bei dem Empfänger sichergestellt wird, daß es sich nicht um unerwünschte Daten, die beispielsweise von einem Denial-of-Service-Angriff herrühren, handelt.

Diese Aufgabe wird gemäß den Merkmalen der unabhängigen Patentansprüche gelöst.

Zunächst wird zur Lösung der Aufgabe ein Verfahren zur Übermittlung von Daten von einem Sender zu einem Empfänger angegeben, bei dem von dem Sender die Daten mittels eines Anwendungsprotokolls auf der Anwendungsschicht um Authentisierungsdaten erweitert werden. Anhand der Authentisierungsdaten wird von dem Empfänger ermittelt, ob er den Sender kennt. Falls der Empfänger den Sender kennt, werden die Daten entgegengenommen, ansonsten werden die Daten verworfen.

Wie oben ausgeführt, haben je nach Protokollstruktur (vgl. OSI-Modell) Daten in jeder Protokollschicht bezogen auf diese Protokollschicht ihre eigene Transparenz, d.h. die Dienste in darunterliegenden Protokollschichten werden (unsichtbar für die aktuelle Protokollschicht) wahrgenommen. Vorteilhaft sind insbesondere Datenpakete der Anwendungsschicht von den administrativen Daten darunterliegender Schichten befreit, d.h., die Datenpakete in der Anwendungsschicht weisen bis auf die administrative Information für die Anwendungsschicht lediglich die eigentlich zu kommunizierenden Daten auf. Demgemäß ist die Authentisierung in der Anwendungsschicht besonders vorteilhaft, weil die Datenpakete selbst, im Gegensatz zu darunterliegenden Schichten, eine deutlich reduzierte Größe haben. Diese geringe Größe wirkt sich vorteilhaft aus auf Echtzeit und Verfügbarkeit des gesamten Kommunikationssystems. Wenn der Empfänger den Sender kennt, d.h. insbesondere, wenn sich der Sender gegenüber dem Empfänger erfolgreich authentifiziert, wird die Nachricht des Senders entgegengenommen, insbesondere werden die Daten abgespeichert. Ansonsten, d.h. wenn der Empfänger den Sender nicht kennt, werden die Daten verworfen, d.h. eine Zwischenspeicherung entfällt. Dies ist besonders vorteilhaft, wenn eine solche Entscheidung automatisch getroffen wird.

Insbesondere können auf der Anwendungsschicht vorteilhaft vorhandene Sicherheitsfunktionen und Schlüssel den einzelnen Anwendungsbenutzern zugeordnet werden. Dies läßt sich vorteilhaft in einer oben skizzierten Filterfunktion mit den Anwendungsdaten verknüpfen. Anwendungsnahe Spezialfunktionen, wie Kompressionsverfahren für Bild und/oder Ton, können vorteilhaft mit Sicherheitsfunktionen in der Anwendungsschicht kombiniert werden, wodurch sich Performance der Endsysteme steigern läßt und der Implementierungsaufwand reduziert wird; dies trifft auch auf oben genannten Filterungsmechanismus zu.

Durch den beschriebenen Authentifikationsmechanismus in der Anwendungsschicht ist es möglich, Sender und Empfänger, die einander kennen, abzugrenzen gegenüber Sendern, die vom Empfänger nicht gekannt werden und von denen der Empfänger auch keine Nachrichten entgegennimmt. Somit läßt sich ein Angriff der eingangs beschriebenen Art ("Denial-of-Service"-Angriff) wirksam verhindern, d.h. unerwünschte Massendaten werden bereits mit dem Eintreffen beim Sender verworfen.

Hierbei sei besonders angemerkt, daß es sich bei dem Empfänger keineswegs um einen Endempfänger oder Adressaten handeln muß. Vielmehr ist es möglich, daß der Empfänger selbst eine vermittelnde Instanz, z.B. ein Vermittlungsknoten oder eine Firewall mit Vermittlungsfunktionalität, ist und somit im Hinblick auf die Authentifikation des Senders für den Endempfänger handelt. So kann, bezogen auf das Beispiel "Firewall" bereits vor Eintritt in das private Kommunikationsnetzwerk eine Erzeugung von überflüssiger Netzlast durch Verwerfen der unerwünschten Massendaten unterbunden werden.
In diesem Szenario (Authentifikation in vermittelnder Instanz) ist es insbesondere von Vorteil, wenn die Authentisierungsdaten anhand eines asymmetrischen Verfahrens erzeugt werden, da somit auch von der vermittelnden Instanz, im Beispiel der Firewall, ermittelt werden kann, von wem die

Daten kommen, ohne daß dazu das Geheimnis für die Entschlüsselung (wie im symmetrischen Verschlüsselungsfall) bereits in der Firewall bekannt sein muß. Die Firewall verwirft unerwünschte Massendaten, wenn ihr der Sender nicht bekannt ist; im anderen Fall, also wenn der vermittelnden Instanz der Sender bekannt ist und die Daten keinesfalls unerwünscht sind, werden diese nach Verifikation des Senders an den Empfänger weitergeleitet. Der Empfänger kann immer noch anhand seines privaten Schlüssels unabhängig von der Überprüfung der Herkunft der Daten, was bereits in der Firewall geschehen ist, die Daten entschlüsseln und lokal weiterverarbeiten (darstellen, abspeichern, etc.). Es sei nochmals betont, daß die Entschlüsselung (im Hinblick auf die Vertraulichkeit) der Daten in dem beschriebenen asymmetrischen Verfahren vom Adressaten mit Hilfe seines privaten Schlüssels durchgeführt wird, wohingegen die Herkunft der Nachricht (die für diesen Adressaten bestimmt ist) auch in der vermittelnden Instanz mit Hilfe des öffentlichen Schlüssels des Absenders - unabhängig vom Inhalt der Nachricht - überprüft werden kann. So kann eine vorgeschaltete Filterfunktionalität von der vermittelnden Instanz erfolgreich wahrgenommen werden. Somit ergibt sich eine klare funktionale Trennung der Filterfunktion gegenüber der Anwendungsfunktion; die Komplexität der Endgeräte kann dadurch vereinfacht und der Netzverkehr im privaten Kommunikationetzwerk reduziert werden.

Eine Weiterbildung besteht darin, daß die Übermittlung der Daten paketorientiert durchgeführt wird.

Eine andere Weiterbildung ist es, daß die Authentisierungsdaten bestimmt werden, indem mindestens ein Teil einer auf der Anwendungsschicht verfügbaren Protokolldateneinheit des Anwendungsprotokolls von dem Sender mit einem Geheimnis verschlüsselt wird. Dabei kann das Geheimnis zwischen Sender und Empfänger ein Schlüssel zur symmetrischen Verschlüsseln oder ein Schlüsselpaar zur

asymmetrischen Verschlüsselung (Public-Key-Verfahren, siehe z.B. Beschreibung mit vermittelnder Instanz) sein. Durch das Entschlüsseln des Geheimnisses wird insbesondere auf der Seite des Empfängers bestimmt, ob der Empfänger den Sender kennt oder nicht. Dies ist vorteilhaft, wenn ohnehin auf Anwendungsebene eine Datenverschlüsselung vorhanden ist, die auch zu diesem Zweck herangezogen werden kann.

Eine Ausgestaltung besteht darin, daß die Authentisierungsdaten bestimmt werden anhand eines Teils der auf der Anwendungsschicht verfügbaren Protokolldateneinheit, insbesondere einer Sequenznummer und/oder eines Zeitstempels umfaßt.

Insbesondere ist es eine Weiterbildung, daß zur Sicherung eine Einweg-Hash-Funktion eingesetzt wird. Auch kann die Sicherung mittels eines Message Authentication Codes (MAC) erfolgen, wobei zusätzlich ein Schlüssel erforderlich ist, der nur dem Sender und dem Empfänger bekannt ist.

Eine andere Ausgestaltung besteht darin, daß vor der Übermittlung der Daten vom Sender zum Empfänger eine Authentifikation zwischen Sender und Empfänger durchgeführt wird.

Eine andere Ausgestaltung besteht darin, daß das beschriebene Verfahren in paketvermittelten Telefondiensten eingesetzt wird, insbesondere im Rahmen der Internet-Telefonie. Alternativ kann das Verfahren in Vermittlungsknoten bzw. Vermittlungsanlagen eingesetzt werden.

Eine andere Lösung der Aufgabe besteht darin, ein System zur Übermittlung von Daten anzugeben, bei dem ein Sender und ein Empfänger vorgesehen sind, wobei der Sender die Daten mittels eines Anwendungsprotokolls auf der Anwendungsschicht um Authentisierungsdaten erweitert. Der Empfänger überprüft anhand der Authentisierungsdaten, ob ihm der Sender bekannt

ist. Falls dem Empfänger der Sender bekannt ist, werden die Daten des Senders verwendet, ansonsten verwirft der Empfänger die Daten des ihm unbekannten Senders.

Auch wird zur Lösung der Aufgabe ein Sender zum Versenden von Daten zu einem Empfänger angegeben, der die Daten mittels eines Anwendungsprotokolls auf der Anwendungsschicht um Authentisierungsdaten erweitert und an den Empfänger versendet.

Schließlich wird zur Lösung der Aufgabe ein Empfänger zum Empfangen von Daten eines Senders angegeben, der insbesondere nach den obigen Ausführungen ausgestaltet ist, wobei der Empfänger anhand von Authentisierungsdaten innerhalb des Anwendungsprotokolls auf der Anwendungsschicht bestimmt, ob ihm der Sender bekannt ist. Falls er den Sender kennt, verwendet er die Daten, ansonsten verwirft der Empfänger die Daten.

Das System bzw. der Sender und der Empfänger sind insbesondere geeignet zur Durchführung des erfindungsgemäßen Verfahrens oder einer seiner vorstehend erläuterten Weiterbildungen.

Ausführungsbeispiele werden nachfolgend anhand der Zeichnungen dargestellt und erläutert.

Es zeigen

Fig.1 eine Protokollarchitektur (Protokoll-Stack) eines Kommunikationssystems im Schichtenmodell;

Fig.2 eine Anwendungsschicht mit möglichen Protokollen;

Fig.3 ein Flußdiagramm zwischen Sender und Empfänger zur Übertragung von Daten vom Sender zum Empfänger;

Fig.4 eine Protokolldateneinheit (PDU = Protocol Data Unit) des Anwendungsprotokolls (Schicht >5) mit Authentisierungsdaten.

In **Fig.1** ist eine Protokollarchitektur (Protokoll-Stack) eines Kommunikationssystems im Schichtenmodell, in Anlehnung an das eingangs erwähnte OSI-Modell, dargestellt. Dazu sind ein Sender 104 und ein Empfänger 105 gezeigt, die über eine physikalische Verbindung 103 Nachrichten austauschen. Sowohl Sender 104 als auch Empfänger 105, weisen die gleiche Protokollarchitektur auf: Schicht 1 110 bzw. 111 stellt eine Übertragungsschicht, Schicht 2 109, 112 eine Sicherungsschicht, Schicht 3 108, 113 eine Vermittlungsschicht, Schicht 4 107, 114 eine Transportschicht und Schicht 5 101, 106 eine Anwendungsschicht dar. Zwischen den einzelnen Schichten existiert von Sender 104 zu Empfänger 105 jeweils ein eigenes Protokoll, bei der Anwendungsschicht 101 bzw. 106 ist dies beispielhaft das Anwendungsprotokoll 102. Von Sender 104 zu Empfänger 105 werden aus Sicht der Schicht 5 101, 106 Protokolldateneinheiten übermittelt, die das in dem Anwendungsprotokoll 102 vorgeschriebene Format haben. Die unterhalb der Anwendungsschicht liegenden Schichten stellen Funktionalitäten dar, die transparent für die Anwendungsschicht erbracht werden. Nach dem OSI-Modell sind die Funktionalitäten zweckmäßig weitgehend voneinander unabhängig gewählt.

Fig.2 zeigt eine Anwendungsschicht mit möglichen Protokollen, bezogen auf eine Anwendung betreffend die Übertragung von Mediendaten (Audio- und Videodaten). So stellen 101 bzw. 106 die bereits aus Fig.1 bekannte Anwendungsschicht (Schicht 5) für Sender 104 und Empfänger 105 dar. Diese Anwendungsschicht ist wiederum unterteilt in mehrere Unterschichten. So gibt es unmittelbar oberhalb der Schicht 4 das Real Time Transport Protokoll (RTP) 201. Dieses Protokoll ermöglicht eine Übertragung von Sprach- und/oder Bilddaten in Echtzeit von

einem zu gegebenenfalls mehreren Punkten (Adressaten). In einer Schicht 206 ist eine Sicherheitsanwendung vorgesehen, die Sicherheitsfunktionen der Anwendungsschicht zusammenfaßt; zugehörige Sicherheitsprotokolle sorgen für Dienste wie

- Benutzerauthentifikation,
- Zugangskontrolle,
- Vertraulichkeit,
- Integrität,
- Verbindlichkeit der Anwendungsdaten und
- Abrechnung.

Je nach Anwendungsfall kommen Audiodaten 203 und/oder Videodaten 202 zum Einsatz. Für beide Alternativen gibt es unterschiedliche Komprimierungsstandards, von denen beispielhaft einige in Fig.2 angeführt sind. Für die Audiodaten sind dies die Standards G.711, G.722, G.723.1 und G.729. Für die Videodaten sind das die beiden Bildkompressionsstandards H.261 und H.263. In der nächsten Abstraktionsstufe (analog zum OSI-Modell) folgen nach der Komprimierung bzw. Dekomprimierung die unterschiedlichen Möglichkeiten einer Anwendung von Audio- bzw. Videodaten (siehe Block 204). Noch eine Abstraktionsebene darüber befindet sich beispielsweise ein User-Interface 205, das für einen Benutzer völlig transparent die darunterliegenden und eben beschriebenen Dienste zur Verfügung stellt. Beispielsweise kann somit ein Benutzer, unter Berücksichtigung sowohl der Audio- als auch der Video-Daten eine Bild-Telefonie, z.B. über das Internet, wahrnehmen, wobei er sich im einzelnen um keinerlei Details der Dienste, die die darunterliegenden Schichten erbringen, Gedanken machen muß. Er nutzt somit den für ihn sichtbaren Dienst der Bild-Telefonie über das Internet transparent. Dabei können unterhalb seines Dienstes verschiedene alternative Implementierungsformen liegen.

Fig.3 zeigt ein Flußdiagramm zwischen Sender und Empfänger zur Übertragung von Daten von dem Sender 308 zu dem Empfänger

307. In einem Schritt 301 werden Authentisierungsdaten auf der Anwendungsschicht erstellt und der für das Anwendungsprotokoll relevanten Protokolldateneinheit (PDU) angehängt. Gemäß den Vereinbarungen des Anwendungsprotokolls werden die Daten einschließlich der Authentisierungsdaten an den Empfänger 307 übermittelt (siehe Verbindung 302). Dies geschieht unter Wahrnehmung der unter dem Anwendungsprotokoll verfügbaren Funktionalitäten bzw. Dienste. In einem Block 303 werden bei dem Empfänger 307 die Authentisierungsdaten geprüft und abgefragt (in einem Block 304), ob der Sender 308 dem Empfänger 307 bekannt ist. Ist dies nicht der Fall, so wird zu einem Block, 305 verzweigt und die Daten werden verworfen. In dem Fall, in dem der Sender 308 dem Empfänger 307 bekannt ist, wird zu einem Block 306 verzweigt und die Daten werden weiter verarbeitet. Die Authentifikation zwischen Sender und Empfänger war erfolgreich, bei den übermittelten Daten handelt es sich nicht um unerwünschte Massendaten.

Fig.4 zeigt eine Protokolldateneinheit (PDU) des Anwendungsprotokolls mit Authentisierungsdaten. Dargestellt ist in Fig.4 eine Protokolldateneinheit (PDU) des Real Time Transport Protokolls (RTP, siehe 201 in Fig.2) 401. Ein solches RTP-Paket 401 umfaßt einen RTP-Header 406, verschlüsselte Mediendaten 404 und Authentisierungsdaten 405. Der RTP-Header 406 umfaßt unter anderem eine Sequenznummer 402 und einen Zeitstempel 403. Sowohl dem Sender als auch dem Empfänger ist ein gemeinsames Geheimnis K bekannt, hier angedeutet als ein Schlüssel, anhand dessen ein Message Authentication Code (MAC), abhängig von der Sequenznummer 402 und dem Zeitstempel 403 erzeugt wird (siehe Feld 405). Vorteilhaft ist im Hinblick auf die Verschlüsselung beispielsweise mittels DES-Algorithmus eine Feldlänge von 64 Bit.

Wie bereits beschrieben, wird ein solches vom Sender abgesandtes Paket auf Seiten des Empfängers derart

authentifiziert, daß (bezogen auf die Anwendungsschicht) abhängig von dem beim Empfänger bekannten Schlüssel (K) ein Message Authentication Code von Feld 402 (Sequenznummer) und Feld 403 (Zeitstempel) erzeugt wird. Ist dieser Message Authentication Code gleich dem Feld 405, so handelt es sich bei dem angekommenen RTP-Paket 401 um ein von einem bekannten Sender stammendes Datenpaket, die Daten werden verarbeitet, beispielsweise dargestellt oder abgespeichert. Im anderen Fall handelt es sich um nicht authentifizierte Daten, der Sender ist dem Empfänger unbekannt und das gesamte RTP-Paket wird verworfen.

Literaturverzeichnis:

[1] Andrew S. Tanenbaum: Computer-Netzwerke; Wolfram's Fachverlag; Attenkirchen 1992; ISBN 3-925328-79-3; Seiten 17-32.

[2] Christoph Ruland: Informationssicherheit in Datennetzen; DATACOM-Verlag; Bergheim 1993; ISBN 3-89238-081-3; Seiten 61-63 und 68-79.

[3] [RFC1889] H. Schulzrinne: RTP: A Transport Protocol for Real-Time Applications; Internet Engineering Task Force, 1996.

[4] [RFC2409] The Internet Key Exchange (IKE), D. Harkins, D. Carrel, ; Internet Engineering Task Force, 1998.

[5] [RFC2402] IP Authentication Header, S. Kent, R. Atkinson; Internet Engineering Task Force, 1998.

[6] [RFC2406] IP Encapsulating Security Payload (ESP), S. Kent, R. Atkinson; Internet Engineering Task Force, 1998.

Patentansprüche

1. Verfahren zur Übermittlung von Daten von einem Sender zu einem Empfänger,
   a) bei dem von dem Sender die Daten mittels eines Anwendungsprotokolls auf der Anwendungsschicht um Authentisierungsdaten erweitert werden;
   b) bei dem anhand der Authentisierungsdaten von dem Empfänger ermittelt wird, ob er den Sender kennt;
   c) bei dem, falls der Empfänger den Sender kennt, die Daten entgegengenommen werden, ansonsten werden die Daten verworfen.

2. Verfahren nach Anspruch 1,
   bei dem die Übermittlung der Daten paketorientiert durchgeführt wird.

3. Verfahren nach einem der vorhergehenden Ansprüche,
   bei dem die Authentisierungsdaten bestimmt werden, indem mindestens ein Teil einer auf der Anwendungsschicht verfügbaren Protokolldateneinheit des Anwendungsprotokolls von dem Sender mit einem Geheimnis verknüpft wird.

4. Verfahren nach Anspruch 3,
   bei dem die Verknüpfung mit dem Geheimnis durch
   a) eine kryptographische Verknüpfung mit mindestens einem weiteren Parameter, insbesondere eine Verschlüsselungsoperation und/oder
   b) eine kryptographischen Prüfsumme und Schlüsselinformation
   durchgeführt wird.

5. Verfahren nach einem der vorhergehenden Ansprüche,
   bei dem durch eine Verifikation auf der Seite des Empfängers bestimmt wird, ob der Empfänger den Sender kennt oder nicht.

6. Verfahren nach Anspruch 5,
bei dem die Verifikation in Form einer kryptographischen Verifikation mit weiteren Parametern, insbesondere einer Entschlüsselungsoperation und/oder einer kryptographischen Prüfsummenüberprüfung durchgeführt wird.

7. Verfahren nach einem der Ansprüche 3 bis 6,
bei dem das Geheimnis zwischen Sender und Empfänger ein Schlüssel zur symmetrischen Verschlüsselung oder ein Schlüsselpaar zur asymmetrischen Verschlüsselung ist.

8. Verfahren nach einem der Ansprüche 3 bis 7,
bei dem der Teil der auf der Anwendungsschicht verfügbaren Protokolldateneinheit eine Sequenznummer und/oder einen Zeitstempel umfaßt.

9. Verfahren nach einem der Ansprüche 3 bis 8,
bei dem zur Verschlüsselung eine Einweg-Hash-Funktion oder ein Message-Authentication-Code mit einem Schlüssel, der nur Empfänger und zu authentifizierendem Sender bekannt ist, eingesetzt wird.

10. Verfahren nach einem der vorhergehenden Ansprüche,
bei dem vor der Übermittlung der Daten vom Sender zum Empfänger eine Authentifikation zwischen Sender und Empfänger durchgeführt wird.

11. Verfahren nach einem der vorhergehenden Ansprüche zum Einsatz in paketvermittelten Telefondiensten.

12. Verfahren nach Anspruch 11
eingesetzt in der Internet-Telefonie.

13. Verfahren nach einem der vorhergehenden Ansprüche eingesetzt in Vermittlungsknoten bzw. Vermittlungsanlagen.

14. System zur Übermittlung von Daten, bei dem ein Sender und ein Empfänger vorgesehen sind,
   a) bei dem der Sender die Daten mittels eines Anwendungsprotokolls auf der Anwendungsschicht um Authentisierungsdaten erweitert;
   b) bei dem der Empfänger anhand der Authentisierungsdaten bestimmt, ob ihm der Sender bekannt ist;
   c) bei dem, falls der Empfänger den Sender kennt, der Empfänger die Daten verwendet, ansonsten verwirft er die Daten.

15. Sender zur Versenden von Daten zu einem Empfänger, der die Daten mittels eines Anwendungsprotokolls auf der Anwendungsschicht um Authentisierungsdaten erweitert und an den Empfänger versendet.

16. Empfänger zum Empfangen von Daten eines Senders, insbesondere nach Anspruch 15,
   a) der anhand von Authentisierungsdaten innerhalb des Anwendungsprotokolls auf der Anwendungsschicht bestimmt, ob ihm der Sender bekannt ist;
   b) der, falls er den Sender kennt, die Daten verwendet, ansonsten verwirft er die Daten.

FIG 1

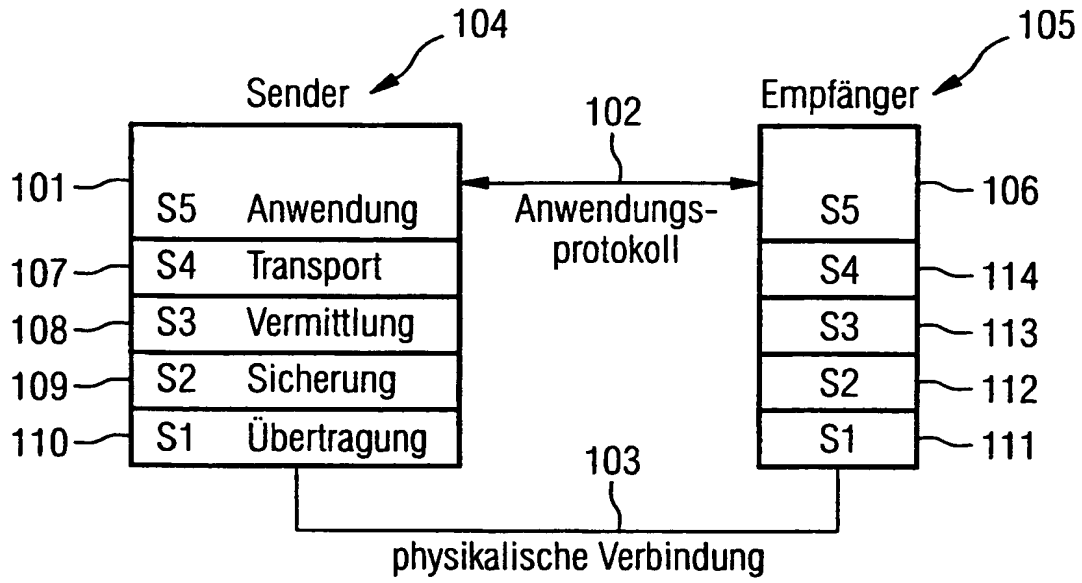
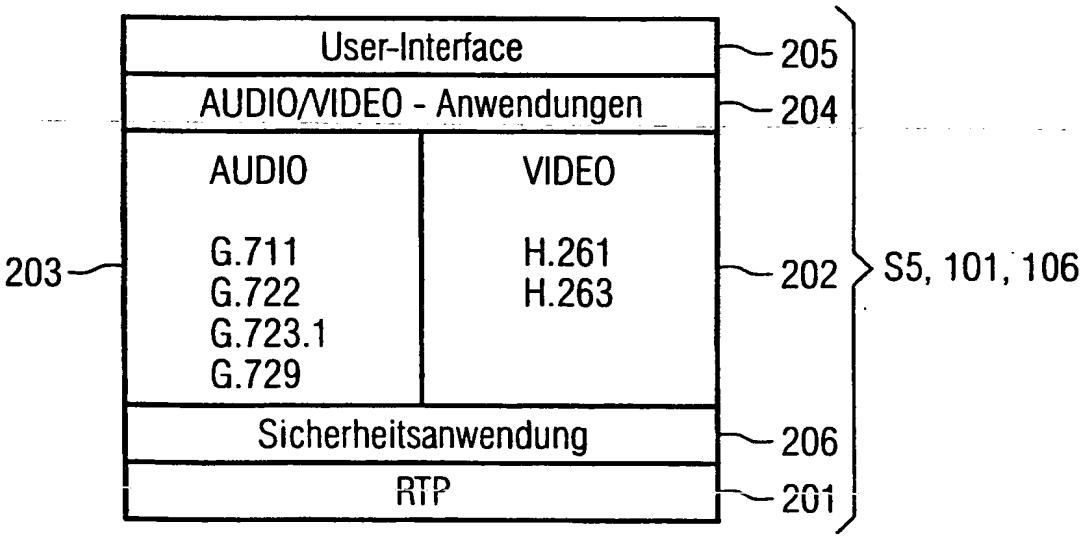
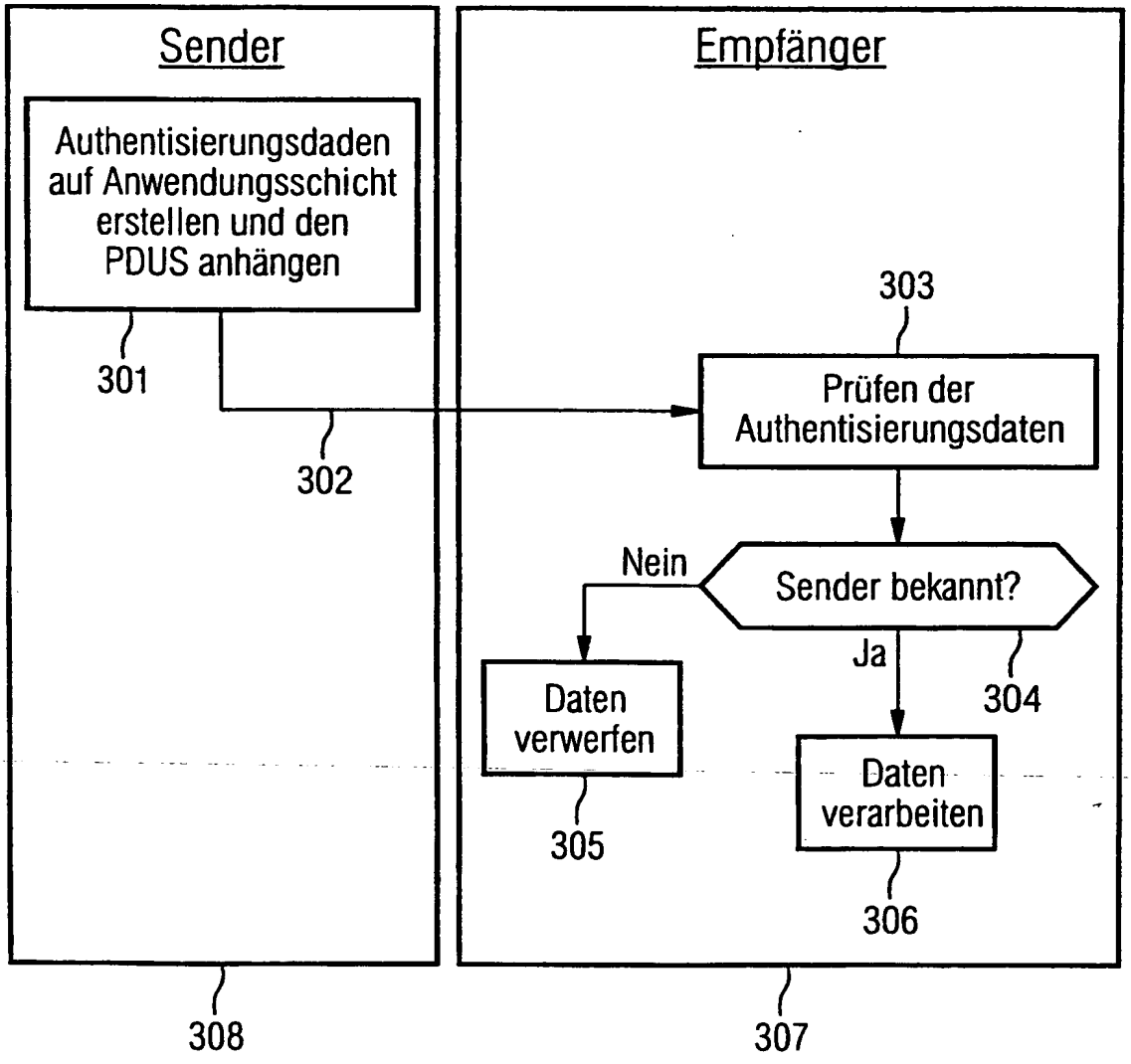

104
Sender
105
Empfänger
102
101
S5 Anwendung
Anwendungs-
protokoll
S5
106
107
S4 Transport
S4
114
108
S3 Vermittlung
S3
113
109
S2 Sicherung
S2
112
110
S1 Übertragung
S1
111
103
physikalische Verbindung

FIG 2


User-Interface
205
AUDIO/VIDEO - Anwendungen
204
AUDIO
VIDEO
203
G.711
G.722
G.723.1
G.729
H.261
H.263
202
S5, 101, 106
Sicherheitsanwendung
206
RTP
201

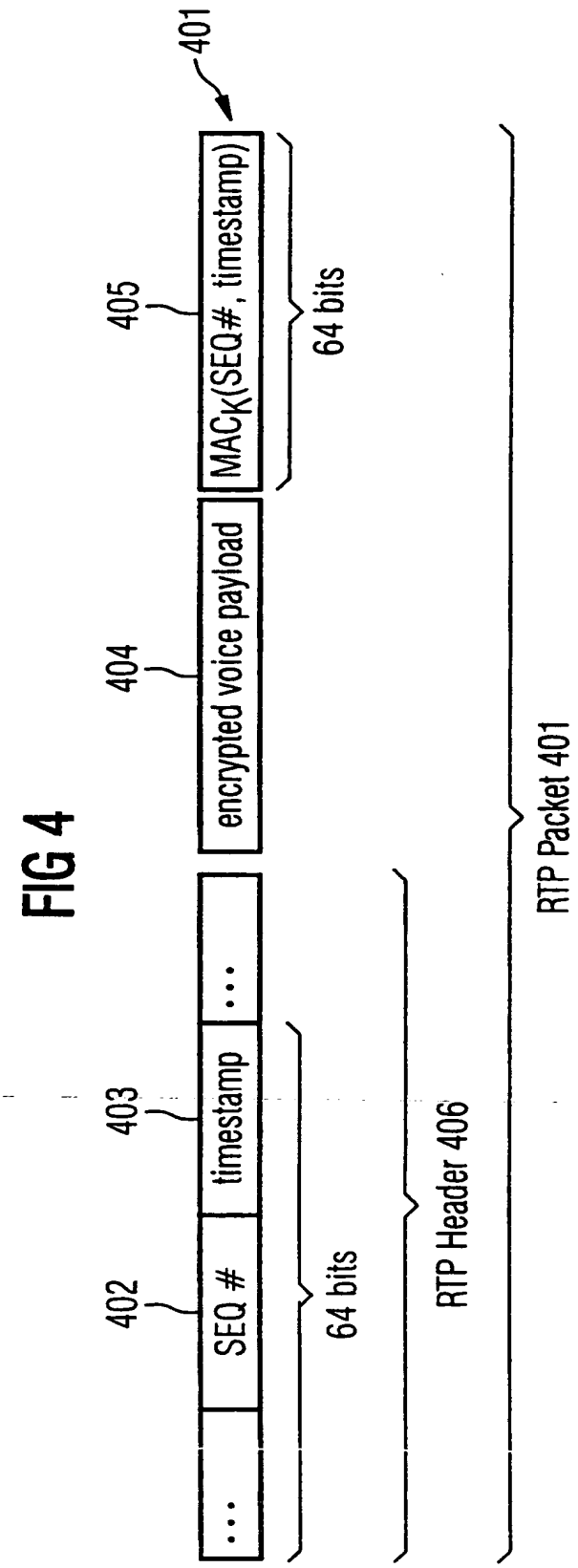
THIS PAGE BLANK (USPTO)

FIG 3


Sender
Empfänger
Authentisierungsdaden auf Anwendungsschicht erstellen und den PDUS anhängen
301
302
303
Prüfen der Authentisierungsdaten
Nein
Sender bekannt?
Ja
304
Daten verwerfen
305
Daten verarbeiten
306
308
307

THIS PAGE BLANK (USPTO)

FIG 4


401
405
404
403
402
MACK(SEQ#, timestamp)
encrypted voice payload
timestamp
SEQ #
...
...
64 bits
64 bits
RTP Header 406
RTP Packet 401

THIS PAGE BLANK (USPTO)

(12) NACH DEM VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS (PCT) VERÖFFENTLICHTE INTERNATIONALE ANMELDUNG

(19) Weltorganisation für geistiges Eigentum
Internationales Büro

(43) Internationales Veröffentlichungsdatum
26. Juli 2001 (26.07.2001)

PCT

(10) Internationale Veröffentlichungsnummer
WO 01/54371 A3

(51) Internationale Patentklassifikation[7]: H04L 29/06

(21) Internationales Aktenzeichen: PCT/DE01/00021

(22) Internationales Anmeldedatum:
5. Januar 2001 (05.01.2001)

(25) Einreichungssprache: Deutsch

(26) Veröffentlichungssprache: Deutsch

(30) Angaben zur Priorität:
100 01 855.6 18. Januar 2000 (18.01.2000) DE

(71) Anmelder *(für alle Bestimmungsstaaten mit Ausnahme von US)*: **SIEMENS AKTIENGESELLSCHAFT** [DE/DE]; Wittelsbacherplatz 2, 80333 München (DE).

(72) Erfinder; und

(75) Erfinder/Anmelder *(nur für US)*: **EUCHNER, Martin** [DE/DE]; Lorenzstrasse 2, 81737 München (DE).

(74) Gemeinsamer Vertreter: **SIEMENS AKTIENGESELLSCHAFT**; Postfach 22 16 34, 80506 München (DE).

(81) Bestimmungsstaaten *(national)*: CA, JP, US.

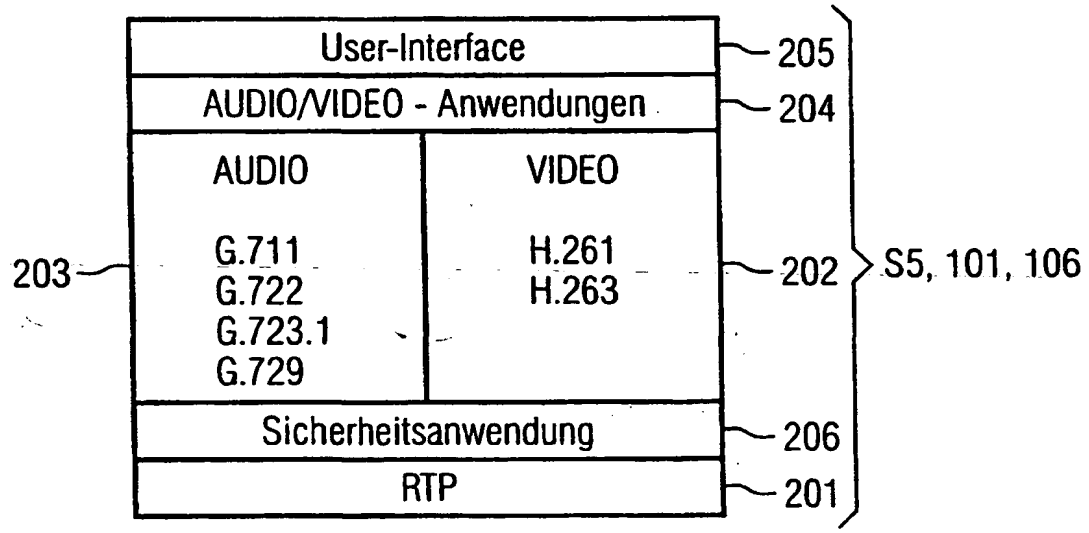
(84) Bestimmungsstaaten *(regional)*: europäisches Patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Veröffentlicht:
— *mit internationalem Recherchenbericht*

[Fortsetzung auf der nächsten Seite]

(54) Title: METHOD, SYSTEM FOR TRANSMITTING DATA FROM A TRANSMITTER TO A RECEIVER AND TRANSMITTER OR RECEIVER

(54) Bezeichnung: VERFAHREN, SYSTEM ZUR ÜBERMITTLUNG VON DATEN VON EINEM SENDER ZU EINEM EMPFÄNGER UND SENDER BZW. EMPFÄNGER HIERZU



204 ... APPLICATIONS
206 ... SECURITY APPLICATION

(57) Abstract: A method and system for transmitting data from a transmitter to a receiver, wherein data from a transmitter is extended to include authentication data on the application level by means an application protocol. Said authentication data is used by the receiver to determine whether the transmitter is known by the receiver. If the transmitter is known by the receiver, the data is accepted. If not, the data is rejected.

[Fortsetzung auf der nächsten Seite]

WO 01/54371 A3

(88) **Veröffentlichungsdatum des internationalen Recherchenberichts:** 28. März 2002

Zur Erklärung der Zweibuchstaben-Codes und der anderen Abkürzungen wird auf die Erklärungen ("Guidance Notes on Codes and Abbreviations") am Anfang jeder regulären Ausgabe der PCT-Gazette verwiesen.

(57) Zusammenfassung: Es wird ein Verfahren zur Übermittlung von Daten von einem Sender zu einem Empfänger angegeben, bei dem von dem Sender die Daten mittels eines Anwendungsprotokolls auf der Anwendungsschicht um Authentisierungsdaten erweitert werden. Anhand der Authentisierungsdaten wird von dem Empfänger ermittelt, ob er den Sender kennt. Falls der Empfänger den Sender kennt, werden die Daten entgegengenommen, ansonsten werden die Daten verworfen.

Meta 1002
p. 349

INTERNATIONAL SEARCH REPORT

International Application No
PC[redacted] 01/00021

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L29/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, PAJ, INSPEC, IBM-TDB, COMPENDEX

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 99 11019 A (V ONE CORP) 4 March 1999 (1999-03-04) abstract page 1, line 4 - line 10 page 11, line 3 - line 15 page 15, line 16 -page 16, line 22 --- -/--	1-16

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
10 October 2001	18/10/2001

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Adkhis, F

Form PCT/ISA/210 (second sheet) (July 1992)

Meta 1002
p. 350

INTERNATIONAL SEARCH REPORT

International Application No
PCT/DE 01/00021

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	HUNT R: "Internet/Intranet firewall security-policy, architecture and transaction services" COMPUTER COMMUNICATIONS, BUTTERWORTHS & CO. PUBLISHERS LTD, GB, vol. 21, no. 13, 1 September 1998 (1998-09-01), pages 1107-1123, XP004146571 ISSN: 0140-3664 abstract page 1113, right-hand column, line 1 -page 1114, left-hand column, line 23 page 1119, left-hand column, line 28 -page 1120, left-hand column, line 39 -----	1-16

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/[redacted]01/00021

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9911019 A	04-03-1999	US 6061796 A	09-05-2000
		AU 8915298 A	16-03-1999
		WO 9911019 A1	04-03-1999
		US 6158011 A	05-12-2000

INTERNATIONALER RECHERCHENBERICHT

Internationales Aktenzeichen
PCT/EP 01/00021

A. KLASSIFIZIERUNG DES ANMELDUNGSGEGENSTANDES
IPK 7 H04L29/06

Nach der Internationalen Patentklassifikation (IPK) oder nach der nationalen Klassifikation und der IPK

B. RECHERCHIERTE GEBIETE

Recherchierter Mindestprüfstoff (Klassifikationssystem und Klassifikationssymbole)
IPK 7 H04L

Recherchierte aber nicht zum Mindestprüfstoff gehörende Veröffentlichungen, soweit diese unter die recherchierten Gebiete fallen

Während der internationalen Recherche konsultierte elektronische Datenbank (Name der Datenbank und evtl. verwendete Suchbegriffe)
EPO-Internal, WPI Data, PAJ, INSPEC, IBM-TDB, COMPENDEX

C. ALS WESENTLICH ANGESEHENE UNTERLAGEN

Kategorie°	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
X	WO 99 11019 A (V ONE CORP) 4. März 1999 (1999-03-04) Zusammenfassung Seite 1, Zeile 4 - Zeile 10 Seite 11, Zeile 3 - Zeile 15 Seite 15, Zeile 16 -Seite 16, Zeile 22 --- -/--	1-16

[X] Weitere Veröffentlichungen sind der Fortsetzung von Feld C zu entnehmen

[X] Siehe Anhang Patentfamilie

° Besondere Kategorien von angegebenen Veröffentlichungen :
"A" Veröffentlichung, die den allgemeinen Stand der Technik definiert, aber nicht als besonders bedeutsam anzusehen ist
"E" älteres Dokument, das jedoch erst am oder nach dem internationalen Anmeldedatum veröffentlicht worden ist
"L" Veröffentlichung, die geeignet ist, einen Prioritätsanspruch zweifelhaft erscheinen zu lassen, oder durch die das Veröffentlichungsdatum einer anderen im Recherchenbericht genannten Veröffentlichung belegt werden soll oder die aus einem anderen besonderen Grund angegeben ist (wie ausgeführt)
"O" Veröffentlichung, die sich auf eine mündliche Offenbarung, eine Benutzung, eine Ausstellung oder andere Maßnahmen bezieht
"P" Veröffentlichung, die vor dem internationalen Anmeldedatum, aber nach dem beanspruchten Prioritätsdatum veröffentlicht worden ist
"T" Spätere Veröffentlichung, die nach dem internationalen Anmeldedatum oder dem Prioritätsdatum veröffentlicht worden ist und mit der Anmeldung nicht kollidiert, sondern nur zum Verständnis des der Erfindung zugrundeliegenden Prinzips oder der ihr zugrundeliegenden Theorie angegeben ist
"X" Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann allein aufgrund dieser Veröffentlichung nicht als neu oder auf erfinderischer Tätigkeit beruhend betrachtet werden
"Y" Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann nicht als auf erfinderischer Tätigkeit beruhend betrachtet werden, wenn die Veröffentlichung mit einer oder mehreren anderen Veröffentlichungen dieser Kategorie in Verbindung gebracht wird und diese Verbindung für einen Fachmann naheliegend ist
"&" Veröffentlichung, die Mitglied derselben Patentfamilie ist

Datum des Abschlusses der internationalen Recherche	Absendedatum des internationalen Recherchenberichts
10. Oktober 2001	18/10/2001

Name und Postanschrift der Internationalen Recherchenbehörde	Bevollmächtigter Bediensteter
Europäisches Patentamt, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Adkhis, F

Formblatt PCT/ISA/210 (Blatt 2) (Juli 1992)

Meta 1002
p. 354

Internationales Aktenzeichen
PCT/DE 01/00021

C.(Fortsetzung) ALS WESENTLICH ANGESEHENE UNTERLAGEN

Kategorie	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
X	HUNT R: "Internet/Intranet firewall security-policy, architecture and transaction services" COMPUTER COMMUNICATIONS, BUTTERWORTHS & CO. PUBLISHERS LTD, GB, Bd. 21, Nr. 13, 1. September 1998 (1998-09-01), Seiten 1107-1123, XP004146571 ISSN: 0140-3664 Zusammenfassung Seite 1113, rechte Spalte, Zeile 1 -Seite 1114, linke Spalte, Zeile 23 Seite 1119, linke Spalte, Zeile 28 -Seite 1120, linke Spalte, Zeile 39	1-16

INTERNATIONALER RECHERCHENBERICHT

Angaben zu Veröffentlic[redacted]gen, die [redacted]selben Patentfamilie gehören

In[redacted]ationales Aktenzeichen
P[redacted] 01/00021

Im Recherchenbericht angeführtes Patentdokument	Datum der Veröffentlichung	Mitglied(er) der Patentfamilie	Datum der Veröffentlichung
WO 9911019 A	04-03-1999	US 6061796 A	09-05-2000
		AU 8915298 A	16-03-1999
		WO 9911019 A1	04-03-1999
		US 6158011 A	05-12-2000

THIS PAGE BLANK (USPTO)

PCT/DE 01/00021 #2

BUNDESREPUBLIK DEUTSCHLAND

10/181564

DE01/21

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1 (a) OR (b)

REC'D 28 FEB 2001
WIPO PCT

EJU

Prioritätsbescheinigung über die Einreichung einer Patentanmeldung

Aktenzeichen: 100 01 855.6

Anmeldetag: 18. Januar 2000

Anmelder/Inhaber: Siemens Aktiengesellschaft, München/DE

Bezeichnung: Verfahren, System zur Übermittlung von Daten von einem Sender zu einem Empfänger und Sender bzw. Empfänger hierzu

IPC: H 04 L 9/32

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 7. Februar 2001
Deutsches Patent- und Markenamt
Der Präsident
Im Auftrag

Joost

A 9161
06/00
EDV-L

THIS PAGE BLANK (USPTO)

100[redacted]1855.6 vom 18.[redacted]00

REC'D 28 FEB 2001

WIPO PCT

Beschreibung

Verfahren, System zur Übermittlung von Daten von einem Sender zu einem Empfänger und Sender bzw. Empfänger hierzu

Die Erfindung betrifft ein Verfahren und ein System und zur Übermittlung von Daten von einem Sender zu einem Empfänger und den Sender bzw. den Empfänger hierzu.

Die Übermittlung von Daten ist z.B. bekannt nach dem OSI-Referenzmodell [1]. Das OSI-Referenzmodell (kurz: OSI-Modell) der International Standards Organisation (ISO) umfaßt sieben Schichten, deren jede eine (bezogen auf die Abstraktion) unterschiedliche Funktionalität aufweist. Im OSI-Modell entspricht Schicht 1 einer Bitübertragungsschicht, dort erfolgt von Sender zu Empfänger eine Übermittlung von Daten bzw. Nachrichten anhand eines Bitübertragungsprotokolls. In Schicht 2 ist eine Sicherungsschicht vorgesehen, zur Übertragung von Schicht 2 des Senders zur Schicht 2 des Empfängers wird ein Sicherungsprotokoll verwendet. Analog werden in Schicht 3, einer Vermittlungsschicht, zwischen Sender und Empfänger ein Vermittlungsprotokoll und in Schicht 4, einer Transportschicht, ein Transportprotokoll eingesetzt. Im OSI-Modell entsprechen die Schicht 5 einer Sitzungsschicht mit einem Sitzungsprotokoll, Schicht 6 einer Darstellungsschicht mit einem Darstellungsprotokoll und Schicht 7 einer Anwendungsschicht mit einem Anwendungsprotokoll. In der Praxis ist es im Rahmen vieler Anwendungen nicht immer möglich, strenge Grenzen gerade zwischen den oberen Protokollschichten, insbesondere den Schichten 5 bis 7, exakt zu benennen. Beispielsweise im Rahmen der Internet-Telefonie, d.h. der Wahrnehmung des Telefondienstes über das Medium Internet, verschmelzen die drei Schichten oberhalb der Transportschicht 4 zu einer "Anwendungsschicht", für das analog obiger Ausführung ein "Anwendungsprotokoll" eingesetzt wird.

Eine Besonderheit des OSI-Modells und damit aller an das OSI-Modell angelehnter Kommunikationsmodelle ist es, daß durch die Unterteilung in Schichten in jeder Schicht die unteren Schichten völlig transparent zu der aktuell betrachteten Schicht Funktionen wahrnehmen und an diese obere Schicht einen Dienst erbringen, der durch die Gesamtheit der Funktionalitäten der unteren Schichten bestimmt ist. "Transparent" bedeutet in diesem Zusammenhang, daß sich die aktuell betrachtete Schicht nicht um die Funktionalitäten der unteren Schichten kümmern muß. Im hier besprochenen Fall kann also das Anwendungsprotokoll zwischen einem Sender der Anwendungsschicht und einem Empfänger der Anwendungsschicht eingesetzt werden. Darunter mögen zahlreiche Dienste, beispielsweise zur Vermittlung, zur Sicherung oder zur tatsächlichen Übertragung über einen physikalischen Kanal, liegen, um die man sich aus Sicht des Anwendungsprotokolls, insbesondere als Anwender dieses Protokolls, nicht mehr kümmern muß.

Analog zu dieser Betrachtung gibt es zwischen den einzelnen Schichten sogenannte Protokolldateneinheiten (engl.: Protocol Data Unit, PDU), die für jedes Protokoll der jeweiligen Schicht eigens ausgeführt sein können. So kann für jede Schicht in dem jeweiligen Protokoll ein eigener Header vorgesehen werden, der administrative Informationen für das Protokoll der jeweiligen Schicht enthält, wobei gemäß OSI-Modell dieser Header auch nur für die Schicht des jeweiligen Protokolls sichtbar und verwendbar ist. Details zu der Konzeption des OSI-Modells finden sich in zahlreichen Literaturstellen, unter anderem der oben genannten.

Wird nachfolgend von Anwendungsschicht gesprochen, so sind damit die Schichten oberhalb der Transportschicht (Schichten größer 4) bezeichnet. Das Anwendungsprotokoll bezeichnet ferner das Protokoll für die Kommunikation zwischen einer oberhalb der Transportschicht liegenden Anwendungsschicht

eines Senders und einer oberhalb der Transportschicht angesiedelten Anwendungsschicht eines Empfängers.

Allgemein bekannt ist ein Message Authentication Code (MAC), der eine kryptographische Prüfsumme bezeichnet, anhand derer eine Veränderung in einer Nachricht bzw. in Daten erkannt werden soll (siehe [2], S.61ff.).

Eine Einweg-Hash-Funktion ist beispielsweise aus [2], S.68ff., bekannt. Mit solch einer Einweg-Hash-Funktion ist es nicht möglich, zu einem gegebenen Funktionswert den passenden Eingabewert zu berechnen. Ein anderes Merkmal ist die Kollisionsfreiheit, d.h. es darf nicht möglich sein, zu zwei verschiedenen Eingabewerten den gleichen Ausgabewert mittels der Einweg-Hash-Funktion zu erhalten.

Ein Unterschied zwischen dem Hash-Code und dem MAC besteht darin, daß der MAC zu seiner Berechnung einen geheimen Schlüssel erfordert, wohingegen die Hash-Funktion schlüsselunabhängig und öffentlich bekannt sein kann.

Weiterhin ist ein asymmetrisches kryptographisches Verfahren (auch: Public-Key-Verfahren) bekannt, z.B. aus [2], S.73ff. Jeder Teilnehmer an einem asymmetrischen kryptographischen Verfahren erhält zwei Schlüssel, einen öffentlichen und einen geheimen Schlüssel. Grundsätzlich ist der geheime Schlüssel aus dem öffentlichen Schlüssel ableitbar, wobei diese Aufgabe so komplex wie möglich sein soll. Mit dem asymmetrischen Verfahren kann weiterhin eine elektronische Unterschrift (Authentifikation) vorgenommen und/oder der Inhalt einer Nachricht (mit dem öffentlichen Schlüssel des Empfängers) so verschlüsselt werden, daß nur der Empfänger mit seinem geheimen Schlüssel die Nachricht wieder entschlüsseln kann.

Schließlich gibt es auch ein symmetrisches Verfahren, das einen (geheimen) Schlüssel erfordert, der sowohl zur Ver- als auch Entschlüsselung eingesetzt wird. Ein Beispiel für ein

symmetrisches Verschlüsselungsverfahren ist der DES-Algorithmus.

Ein Kommunikationssystem unterliegt einer Vielzahl von möglichen Angriffen, die u.a. auf den Inhalt der im Kommunikationssystem ausgetauschten Nachrichten oder auf die Verfügbarkeit des Kommunikationssystems abzielen können. Nimmt man das Beispiel der Internet-Telefonie, so ist es zum einen von Bedeutung, daß der Inhalt des Telefonats nicht von einem unberechtigten Dritten mitgehört werden kann, zum anderen ist es auch erforderlich, daß der Dritte nicht zahllose Anrufe initiieren kann und somit den Empfänger stört und das Kommunikationssystem unnötig belastet. Solche Angriffe werden auch als Denial-of-Service-Angriffe bezeichnet. Mögliche Beispiele für solche Angriffe sind Massendaten bzw. Massennachrichten, die automatisch erzeugt und zu einem oder mehreren Empfängern geschickt und dort Verfügbarkeit und Performanz in erheblichem Umfang belasten. Im Beispiel Internet-Telefonie ist es zumindest unerwünscht, daß unablässig ein Internet-Telefon klingelt und somit neben der unnötigen Netzlast auch noch die Aufmerksamkeit des Empfängers abgelenkt wird. Ebenso wird es als störend empfunden, wenn ein Angreifer durch das Senden unerlaubter Sprachdaten den Kommunikationsfluß zwischen Sender und Empfänger stört.

Zur Abwehr von Angriffen auf ein privates Kommunikationsnetz werden vielfach sogenannte "Firewalls" eingesetzt, die insbesondere eine Abgrenzung des privaten Kommunikationsnetzes von öffentlichen Netzen sicherstellen. Allerdings ist es für einen Angreifer leicht möglich, seine unsinnigen (Massen)Daten auch an Adressaten innerhalb des privaten Netzes, also hinter die Firewall, zu senden. Dort werden diese Daten decodiert und gegebenenfalls wiedergegeben. Dadurch erreicht der Angreifer, daß bspw. bei der Wiedergabe solcher unsinniger Daten nur lästige Tonstörungen oder Rauschen wahrgenommen und damit das private

Netz in seiner Bandbreite und Verfügbarkeit eingeschränkt wird. Im Extremfall ist auch ein Absturz eines Teils oder des gesamten privaten Kommunikationsnetzes möglich.

Ferner ist ein RTP-Protokoll für die Übertragung von Mediendaten (sog. "Payload"), d.h. von Bild- oder Tondaten, , bekannt [3].

Ein Ansatz zur Abhaltung von unerwünschten Daten liefert das IPSEC-Protokoll [5] bzw. [6]. Hier können Datenpakete des Internetprotokolls (IP-Datenpakete) eingekapselt und bezüglich Vertraulichkeit und/oder Integrität (implizite Absenderauthentifikation) geschützt werden. Ferner bietet das IPSEC-Protokoll ein Keymanagement-Verfahren mit einem sogenannten Cookie-Mechanismus (siehe [4]), anhand dessen die oben diskutierten Massendatenangriffe (Denial-of-Service-Angriffe) während der Keymanagement-Phase abgewehrt werden können. Bei dem Cookie-Mechanismus werden schnelle Einweg-Hash-Funktionen (z.B. SHA-1), Zufallszahlen und IP-Adressen miteinander verknüpft. Allerdings ist das erste "Cookie", das vom Sender zum Empfänger übertragen wird, nicht abgesichert, wodurch eine Sicherheitslücke entsteht. Ferner ist das IPSEC-Cookie-Verfahren nicht geeignet, um auf der Anwendungsschicht (für das Anwendungsprotokoll) auch unter Echtzeitbedingungen einen Schutz gegen derartige unerwünschte Massendaten zu gewährleisten. Nachteilig für IPSEC ist wegen der OSI Schichtengliederung allerdings unter anderem, daß eine Verknüpfung von IPSEC mit den Sicherheitsfunktionen einer Anwendungsschicht sich so nicht bewerkstelligen läßt.

Die **Aufgabe** der Erfindung besteht darin, Daten von einem Sender zu einem Empfänger zu übertragen, wobei bei dem Empfänger sichergestellt wird, daß es sich nicht um unerwünschte Daten, die beispielsweise von einem Denial-of-Service-Angriff herrühren, handelt.

Diese Aufgabe wird gemäß den Merkmalen der unabhängigen Patentansprüche gelöst.

Zunächst wird zur Lösung der Aufgabe ein Verfahren zur Übermittlung von Daten von einem Sender zu einem Empfänger angegeben, bei dem von dem Sender die Daten mittels eines Anwendungsprotokolls auf der Anwendungsschicht um Authentisierungsdaten erweitert werden. Anhand der Authentisierungsdaten wird von dem Empfänger ermittelt, ob er den Sender kennt. Falls der Empfänger den Sender kennt, werden die Daten entgegengenommen, ansonsten werden die Daten verworfen.

Wie oben ausgeführt, haben je nach Protokollstruktur (vgl. OSI-Modell) Daten in jeder Protokollschicht bezogen auf diese Protokollschicht ihre eigene Transparenz, d.h. die Dienste in darunterliegenden Protokollschichten werden (unsichtbar für die aktuelle Protokollschicht) wahrgenommen. Vorteilhaft sind insbesondere Datenpakete der Anwendungsschicht von den administrativen Daten darunterliegender Schichten befreit, d.h., die Datenpakete in der Anwendungsschicht weisen bis auf die administrative Information für die Anwendungsschicht lediglich die eigentlich zu kommunizierenden Daten auf. Demgemäß ist die Authentisierung in der Anwendungsschicht besonders vorteilhaft, weil die Datenpakete selbst, im Gegensatz zu darunterliegenden Schichten, eine deutlich reduzierte Größe haben. Diese geringe Größe wirkt sich vorteilhaft aus auf Echtzeit und Verfügbarkeit des gesamten Kommunikationssystems. Wenn der Empfänger den Sender kennt, d.h. insbesondere, wenn sich der Sender gegenüber dem Empfänger erfolgreich authentifiziert, wird die Nachricht des Senders entgegengenommen, insbesondere werden die Daten abgespeichert. Ansonsten, d.h. wenn der Empfänger den Sender nicht kennt, werden die Daten verworfen, d.h. eine Zwischenspeicherung entfällt. Dies ist besonders vorteilhaft, wenn eine solche Entscheidung automatisch getroffen wird.

Insbesondere können auf der Anwendungsschicht vorteilhaft vorhandene Sicherheitsfunktionen und Schlüssel den einzelnen Anwendungsbenutzern zugeordnet werden. Dies läßt sich vorteilhaft in einer oben skizzierten Filterfunktion mit den Anwendungsdaten verknüpfen. Anwendungsnahe Spezialfunktionen, wie Kompressionsverfahren für Bild und/oder Ton, können vorteilhaft mit Sicherheitsfunktionen in der Anwendungsschicht kombiniert werden, wodurch sich Performance der Endsysteme steigern läßt und der Implementierungsaufwand reduziert wird; dies trifft auch auf oben genannten Filterungsmechanismus zu.

Durch den beschriebenen Authentifikationsmechanismus in der Anwendungsschicht ist es möglich, Sender und Empfänger, die einander kennen, abzugrenzen gegenüber Sendern, die vom Empfänger nicht gekannt werden und von denen der Empfänger auch keine Nachrichten entgegennimmt. Somit läßt sich ein Angriff der eingangs beschriebenen Art ("Denial-of-Service"-Angriff) wirksam verhindern, d.h. unerwünschte Massendaten werden bereits mit dem Eintreffen beim Sender verworfen.

Hierbei sei besonders angemerkt, daß es sich bei dem Empfänger keineswegs um einen Endempfänger oder Adressaten handeln muß. Vielmehr ist es möglich, daß der Empfänger selbst eine vermittelnde Instanz, z.B. ein Vermittlungsknoten oder eine Firewall mit Vermittlungsfunktionalität, ist und somit im Hinblick auf die Authentifikation des Senders für den Endempfänger handelt. So kann, bezogen auf das Beispiel "Firewall" bereits vor Eintritt in das private Kommunikationsnetzwerk eine Erzeugung von überflüssiger Netzlast durch Verwerfen der unerwünschten Massendaten unterbunden werden.
In diesem Szenario (Authentifikation in vermittelnder Instanz) ist es insbesondere von Vorteil, wenn die Authentisierungsdaten anhand eines asymmetrischen Verfahrens erzeugt werden, da somit auch von der vermittelnden Instanz, im Beispiel der Firewall, ermittelt werden kann, von wem die

Daten kommen, ohne daß dazu das Geheimnis für die Entschlüsselung (wie im symmetrischen Verschlüsselungsfall) bereits in der Firewall bekannt sein muß. Die Firewall verwirft unerwünschte Massendaten, wenn ihr der Sender nicht bekannt ist; im anderen Fall, also wenn der vermittelnden Instanz der Sender bekannt ist und die Daten keinesfalls unerwünscht sind, werden diese nach Verifikation des Senders an den Empfänger weitergeleitet. Der Empfänger kann immer noch anhand seines privaten Schlüssels unabhängig von der Überprüfung der Herkunft der Daten, was bereits in der Firewall geschehen ist, die Daten entschlüsseln und lokal weiterverarbeiten (darstellen, abspeichern, etc.). Es sei nochmals betont, daß die Entschlüsselung (im Hinblick auf die Vertraulichkeit) der Daten in dem beschriebenen asymmetrischen Verfahren vom Adressaten mit Hilfe seines privaten Schlüssels durchgeführt wird, wohingegen die Herkunft der Nachricht (die für diesen Adressaten bestimmt ist) auch in der vermittelnden Instanz mit Hilfe des öffentlichen Schlüssels des Absenders - unabhängig vom Inhalt der Nachricht - überprüft werden kann. So kann eine vorgeschaltete Filterfunktionalität von der vermittelnden Instanz erfolgreich wahrgenommen werden. Somit ergibt sich eine klare funktionale Trennung der Filterfunktion gegenüber der Anwendungsfunktion; die Komplexität der Endgeräte kann dadurch vereinfacht und der Netzverkehr im privaten Kommunikationetzwerk reduziert werden.

Eine Weiterbildung besteht darin, daß die Übermittlung der Daten paketorientiert durchgeführt wird.

Eine andere Weiterbildung ist es, daß die Authentisierungsdaten bestimmt werden, indem mindestens ein Teil einer auf der Anwendungsschicht verfügbaren Protokolldateneinheit des Anwendungsprotokolls von dem Sender mit einem Geheimnis verschlüsselt wird. Dabei kann das Geheimnis zwischen Sender und Empfänger ein Schlüssel zur symmetrischen Verschlüsseln oder ein Schlüsselpaar zur

asymmetrischen Verschlüsselung (Public-Key-Verfahren, siehe z.B. Beschreibung mit vermittelnder Instanz) sein. Durch das Entschlüsseln des Geheimnisses wird insbesondere auf der Seite des Empfängers bestimmt, ob der Empfänger den Sender kennt oder nicht. Dies ist vorteilhaft, wenn ohnehin auf Anwendungsebene eine Datenverschlüsselung vorhanden ist, die auch zu diesem Zweck herangezogen werden kann.

Eine Ausgestaltung besteht darin, daß die Authentisierungsdaten bestimmt werden anhand eines Teils der auf der Anwendungsschicht verfügbaren Protokolldateneinheit, insbesondere einer Sequenznummer und/oder eines Zeitstempels umfaßt.

Insbesondere ist es eine Weiterbildung, daß zur Sicherung eine Einweg-Hash-Funktion eingesetzt wird. Auch kann die Sicherung mittels eines Message Authentication Codes (MAC) erfolgen, wobei zusätzlich ein Schlüssel erforderlich ist, der nur dem Sender und dem Empfänger bekannt ist.

Eine andere Ausgestaltung besteht darin, daß vor der Übermittlung der Daten vom Sender zum Empfänger eine Authentifikation zwischen Sender und Empfänger durchgeführt wird.

Eine andere Ausgestaltung besteht darin, daß das beschriebene Verfahren in paketvermittelten Telefondiensten eingesetzt wird, insbesondere im Rahmen der Internet-Telefonie. Alternativ kann das Verfahren in Vermittlungsknoten bzw. Vermittlungsanlagen eingesetzt werden.

Eine andere Lösung der Aufgabe besteht darin, ein System zur Übermittlung von Daten anzugeben, bei dem ein Sender und ein Empfänger vorgesehen sind, wobei der Sender die Daten mittels eines Anwendungsprotokolls auf der Anwendungsschicht um Authentisierungsdaten erweitert. Der Empfänger überprüft anhand der Authentisierungsdaten, ob ihm der Sender bekannt

ist. Falls dem Empfänger der Sender bekannt ist, werden die Daten des Senders verwendet, ansonsten verwirft der Empfänger die Daten des ihm unbekannten Senders.

Auch wird zur Lösung der Aufgabe ein Sender zum Versenden von Daten zu einem Empfänger angegeben, der die Daten mittels eines Anwendungsprotokolls auf der Anwendungsschicht um Authentisierungsdaten erweitert und an den Empfänger versendet.

Schließlich wird zur Lösung der Aufgabe ein Empfänger zum Empfangen von Daten eines Senders angegeben, der insbesondere nach den obigen Ausführungen ausgestaltet ist, wobei der Empfänger anhand von Authentisierungsdaten innerhalb des Anwendungsprotokolls auf der Anwendungsschicht bestimmt, ob ihm der Sender bekannt ist. Falls er den Sender kennt, verwendet er die Daten, ansonsten verwirft der Empfänger die Daten.

Das System bzw. der Sender und der Empfänger sind insbesondere geeignet zur Durchführung des erfindungsgemäßen Verfahrens oder einer seiner vorstehend erläuterten Weiterbildungen.

Ausführungsbeispiele werden nachfolgend anhand der Zeichnungen dargestellt und erläutert.

Es zeigen

Fig.1 eine Protokollarchitektur (Protokoll-Stack) eines Kommunikationssystems im Schichtenmodell;

Fig.2 eine Anwendungsschicht mit möglichen Protokollen;

Fig.3 ein Flußdiagramm zwischen Sender und Empfänger zur Übertragung von Daten vom Sender zum Empfänger;

Fig.4 eine Protokolldateneinheit (PDU = Protocol Data Unit) des Anwendungsprotokolls (Schicht >5) mit Authentisierungsdaten.

In **Fig.1** ist eine Protokollarchitektur (Protokoll-Stack) eines Kommunikationssystems im Schichtenmodell, in Anlehnung an das eingangs erwähnte OSI-Modell, dargestellt. Dazu sind ein Sender 104 und ein Empfänger 105 gezeigt, die über eine physikalische Verbindung 103 Nachrichten austauschen. Sowohl Sender 104 als auch Empfänger 105, weisen die gleiche Protokollarchitektur auf: Schicht 1 110 bzw. 111 stellt eine Übertragungsschicht, Schicht 2 109, 112 eine Sicherungsschicht, Schicht 3 108, 113 eine Vermittlungsschicht, Schicht 4 107, 114 eine Transportschicht und Schicht 5 101, 106 eine Anwendungsschicht dar. Zwischen den einzelnen Schichten existiert von Sender 104 zu Empfänger 105 jeweils ein eigenes Protokoll, bei der Anwendungsschicht 101 bzw. 106 ist dies beispielhaft das Anwendungsprotokoll 102. Von Sender 104 zu Empfänger 105 werden aus Sicht der Schicht 5 101, 106 Protokolldateneinheiten übermittelt, die das in dem Anwendungsprotokoll 102 vorgeschriebene Format haben. Die unterhalb der Anwendungsschicht liegenden Schichten stellen Funktionalitäten dar, die transparent für die Anwendungsschicht erbracht werden. Nach dem OSI-Modell sind die Funktionalitäten zweckmäßig weitgehend voneinander unabhängig gewählt.

Fig.2 zeigt eine Anwendungsschicht mit möglichen Protokollen, bezogen auf eine Anwendung betreffend die Übertragung von Mediendaten (Audio- und Videodaten). So stellen 101 bzw. 106 die bereits aus Fig.1 bekannte Anwendungsschicht (Schicht 5) für Sender 104 und Empfänger 105 dar. Diese Anwendungsschicht ist wiederum unterteilt in mehrere Unterschichten. So gibt es unmittelbar oberhalb der Schicht 4 das Real Time Transport Protokoll (RTP) 201. Dieses Protokoll ermöglicht eine Übertragung von Sprach- und/oder Bilddaten in Echtzeit von

einem zu gegebenenfalls mehreren Punkten (Adressaten). In einer Schicht 206 ist eine Sicherheitsanwendung vorgesehen, die Sicherheitsfunktionen der Anwendungsschicht zusammenfaßt; zugehörige Sicherheitsprotokolle sorgen für Dienste wie

- Benutzerauthentifikation,
- Zugangskontrolle,
- Vertraulichkeit,
- Integrität,
- Verbindlichkeit der Anwendungsdaten und
- Abrechnung.

Je nach Anwendungsfall kommen Audiodaten 203 und/oder Videodaten 202 zum Einsatz. Für beide Alternativen gibt es unterschiedliche Komprimierungsstandards, von denen beispielhaft einige in Fig.2 angeführt sind. Für die Audiodaten sind dies die Standards G.711, G.722, G.723.1 und G.729. Für die Videodaten sind das die beiden Bildkompressionsstandards H.261 und H.263. In der nächsten Abstraktionsstufe (analog zum OSI-Modell) folgen nach der Komprimierung bzw. Dekomprimierung die unterschiedlichen Möglichkeiten einer Anwendung von Audio- bzw. Videodaten (siehe Block 204). Noch eine Abstraktionsebene darüber befindet sich beispielsweise ein User-Interface 205, das für einen Benutzer völlig transparent die darunterliegenden und eben beschriebenen Dienste zur Verfügung stellt. Beispielsweise kann somit ein Benutzer, unter Berücksichtigung sowohl der Audio- als auch der Video-Daten eine Bild-Telefonie, z.B. über das Internet, wahrnehmen, wobei er sich im einzelnen um keinerlei Details der Dienste, die die darunterliegenden Schichten erbringen, Gedanken machen muß. Er nutzt somit den für ihn sichtbaren Dienst der Bild-Telefonie über das Internet transparent. Dabei können unterhalb seines Dienstes verschiedene alternative Implementierungsformen liegen.

Fig.3 zeigt ein Flußdiagramm zwischen Sender und Empfänger zur Übertragung von Daten von dem Sender 308 zu dem Empfänger

307. In einem Schritt 301 werden Authentisierungsdaten auf der Anwendungsschicht erstellt und der für das Anwendungsprotokoll relevanten Protokolldateneinheit (PDU) angehängt. Gemäß den Vereinbarungen des Anwendungsprotokolls werden die Daten einschließlich der Authentisierungsdaten an den Empfänger 307 übermittelt (siehe Verbindung 302). Dies geschieht unter Wahrnehmung der unter dem Anwendungsprotokoll verfügbaren Funktionalitäten bzw. Dienste. In einem Block 303 werden bei dem Empfänger 307 die Authentisierungsdaten geprüft und abgefragt (in einem Block 304), ob der Sender 308 dem Empfänger 307 bekannt ist. Ist dies nicht der Fall, so wird zu einem Block, 305 verzweigt und die Daten werden verworfen. In dem Fall, in dem der Sender 308 dem Empfänger 307 bekannt ist, wird zu einem Block 306 verzweigt und die Daten werden weiter verarbeitet. Die Authentifikation zwischen Sender und Empfänger war erfolgreich, bei den übermittelten Daten handelt es sich nicht um unerwünschte Massendaten.

Fig.4 zeigt eine Protokolldateneinheit (PDU) des Anwendungsprotokolls mit Authentisierungsdaten. Dargestellt ist in Fig.4 eine Protokolldateneinheit (PDU) des Real Time Transport Protokolls (RTP, siehe 201 in Fig.2) 401. Ein solches RTP-Paket 401 umfaßt einen RTP-Header 406, verschlüsselte Mediendaten 404 und Authentisierungsdaten 405. Der RTP-Header 406 umfaßt unter anderem eine Sequenznummer 402 und einen Zeitstempel 403. Sowohl dem Sender als auch dem Empfänger ist ein gemeinsames Geheimnis K bekannt, hier angedeutet als ein Schlüssel, anhand dessen ein Message Authentication Code (MAC), abhängig von der Sequenznummer 402 und dem Zeitstempel 403 erzeugt wird (siehe Feld 405). Vorteilhaft ist im Hinblick auf die Verschlüsselung beispielsweise mittels DES-Algorithmus eine Feldlänge von 64 Bit.

Wie bereits beschrieben, wird ein solches vom Sender abgesandtes Paket auf Seiten des Empfängers derart

authentifiziert, daß (bezogen auf die Anwendungsschicht) abhängig von dem beim Empfänger bekannten Schlüssel (K) ein Message Authentication Code von Feld 402 (Sequenznummer) und Feld 403 (Zeitstempel) erzeugt wird. Ist dieser Message Authentication Code gleich dem Feld 405, so handelt es sich bei dem angekommenen RTP-Paket 401 um ein von einem bekannten Sender stammendes Datenpaket, die Daten werden verarbeitet, beispielsweise dargestellt oder abgespeichert. Im anderen Fall handelt es sich um nicht authentifizierte Daten, der Sender ist dem Empfänger unbekannt und das gesamte RTP-Paket wird verworfen.

Literaturverzeichnis:

[1] Andrew S. Tanenbaum: Computer-Netzwerke; Wolfram's Fachverlag; Attenkirchen 1992; ISBN 3-925328-79-3; Seiten 17-32.

[2] Christoph Ruland: Informationssicherheit in Datennetzen; DATACOM-Verlag; Bergheim 1993; ISBN 3-89238-081-3; Seiten 61-63 und 68-79.

[3] [RFC1889] H. Schulzrinne: RTP: A Transport Protocol for Real-Time Applications; Internet Engineering Task Force, 1996.

[4] [RFC2409] The Internet Key Exchange (IKE), D. Harkins, D. Carrel, ; Internet Engineering Task Force, 1998.

[5] [RFC2402] IP Authentication Header, S. Kent, R. Atkinson; Internet Engineering Task Force, 1998.

[6] [RFC2406] IP Encapsulating Security Payload (ESP), S. Kent, R. Atkinson; Internet Engineering Task Force, 1998.

Patentansprüche

1. Verfahren zur Übermittlung von Daten von einem Sender zu einem Empfänger,
   a) bei dem von dem Sender die Daten mittels eines Anwendungsprotokolls auf der Anwendungsschicht um Authentisierungsdaten erweitert werden;
   b) bei dem anhand der Authentisierungsdaten von dem Empfänger ermittelt wird, ob er den Sender kennt;
   c) bei dem, falls der Empfänger den Sender kennt, die Daten entgegengenommen werden, ansonsten werden die Daten verworfen.

2. Verfahren nach Anspruch 1,
   bei dem die Übermittlung der Daten paketorientiert durchgeführt wird.

3. Verfahren nach einem der vorhergehenden Ansprüche,
   bei dem die Authentisierungsdaten bestimmt werden, indem mindestens ein Teil einer auf der Anwendungsschicht verfügbaren Protokolldateneinheit des Anwendungsprotokolls von dem Sender mit einem Geheimnis verknüpft wird.

4. Verfahren nach Anspruch 3,
   bei dem die Verknüpfung mit dem Geheimnis durch
   a) eine kryptographische Verknüpfung mit mindestens einem weiteren Parameter, insbesondere eine Verschlüsselungsoperation und/oder
   b) eine kryptographischen Prüfsumme und Schlüsselinformation
   durchgeführt wird.

5. Verfahren nach einem der vorhergehenden Ansprüche,
   bei dem durch eine Verifikation auf der Seite des Empfängers bestimmt wird, ob der Empfänger den Sender kennt oder nicht.

6. Verfahren nach Anspruch 5,
bei dem die Verifikation in Form einer kryptographischen Verifikation mit weiteren Parametern, insbesondere einer Entschlüsselungsoperation und/oder einer kryptographischen Prüfsummenüberprüfung durchgeführt wird.

7. Verfahren nach einem der Ansprüche 3 bis 6,
bei dem das Geheimnis zwischen Sender und Empfänger ein Schlüssel zur symmetrischen Verschlüsselung oder ein Schlüsselpaar zur asymmetrischen Verschlüsselung ist.

8. Verfahren nach einem der Ansprüche 3 bis 7,
bei dem der Teil der auf der Anwendungsschicht verfügbaren Protokolldateneinheit eine Sequenznummer und/oder einen Zeitstempel umfaßt.

9. Verfahren nach einem der Ansprüche 3 bis 8,
bei dem zur Verschlüsselung eine Einweg-Hash-Funktion oder ein Message-Authentication-Code mit einem Schlüssel, der nur Empfänger und zu authentifizierendem Sender bekannt ist, eingesetzt wird.

10. Verfahren nach einem der vorhergehenden Ansprüche,
bei dem vor der Übermittlung der Daten vom Sender zum Empfänger eine Authentifikation zwischen Sender und Empfänger durchgeführt wird.

11. Verfahren nach einem der vorhergehenden Ansprüche zum Einsatz in paketvermittelten Telefondiensten.

12. Verfahren nach Anspruch 11
eingesetzt in der Internet-Telefonie.

13. Verfahren nach einem der vorhergehenden Ansprüche eingesetzt in Vermittlungsknoten bzw. Vermittlungsanlagen.

14. System zur Übermittlung von Daten, bei dem ein Sender und ein Empfänger vorgesehen sind,
    a) bei dem der Sender die Daten mittels eines Anwendungsprotokolls auf der Anwendungsschicht um Authentisierungsdaten erweitert;
    b) bei dem der Empfänger anhand der Authentisierungsdaten bestimmt, ob ihm der Sender bekannt ist;
    c) bei dem, falls der Empfänger den Sender kennt, der Empfänger die Daten verwendet, ansonsten verwirft er die Daten.

15. Sender zur Versenden von Daten zu einem Empfänger, der die Daten mittels eines Anwendungsprotokolls auf der Anwendungsschicht um Authentisierungsdaten erweitert und an den Empfänger versendet.

16. Empfänger zum Empfangen von Daten eines Senders, insbesondere nach Anspruch 15,
    a) der anhand von Authentisierungsdaten innerhalb des Anwendungsprotokolls auf der Anwendungsschicht bestimmt, ob ihm der Sender bekannt ist;
    b) der, falls er den Sender kennt, die Daten verwendet, ansonsten verwirft er die Daten.

Zusammenfassung

Verfahren, System zur Übermittlung von Daten von einem Sender zu einem Empfänger und Sender bzw. Empfänger hierzu

Es wird ein Verfahren zur Übermittlung von Daten von einem Sender zu einem Empfänger angegeben, bei dem von dem Sender die Daten mittels eines Anwendungsprotokolls auf der Anwendungsschicht um Authentisierungsdaten erweitert werden. Anhand der Authentisierungsdaten wird von dem Empfänger ermittelt, ob er den Sender kennt. Falls der Empfänger den Sender kennt, werden die Daten entgegengenommen, ansonsten werden die Daten verworfen.

99 E 6915

FIG 1

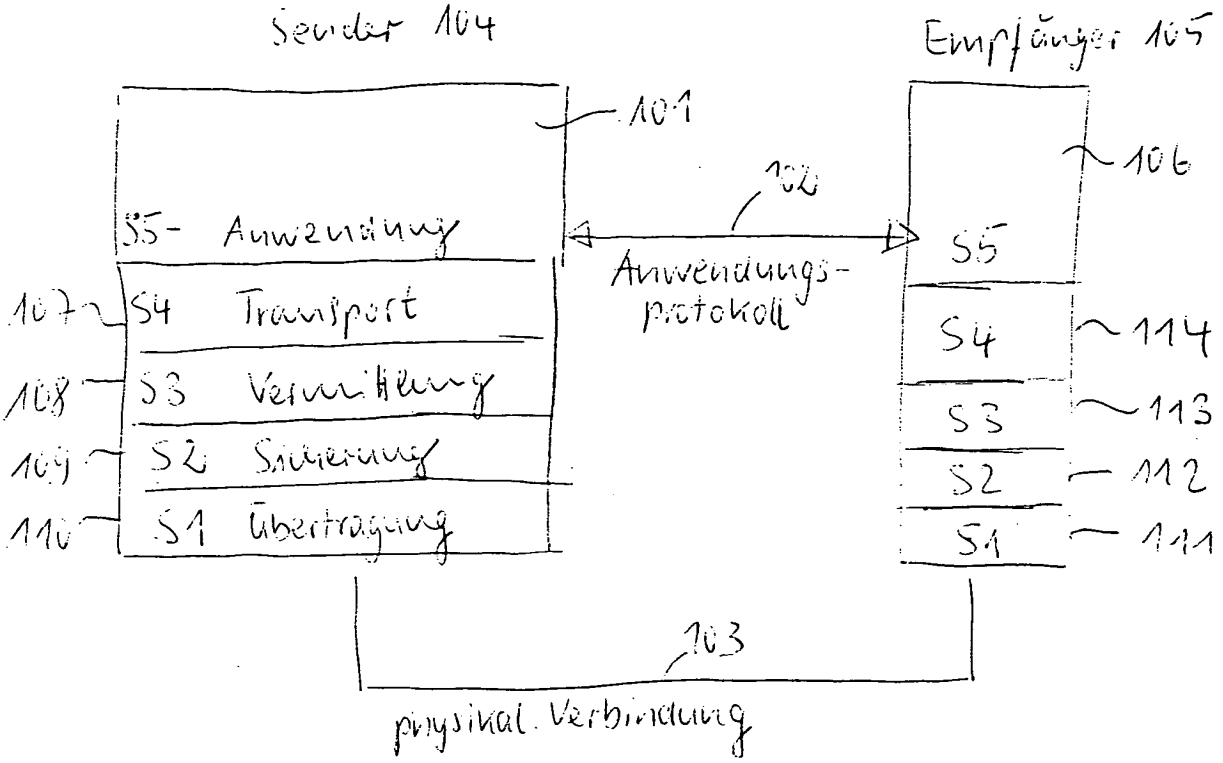


FIG 2

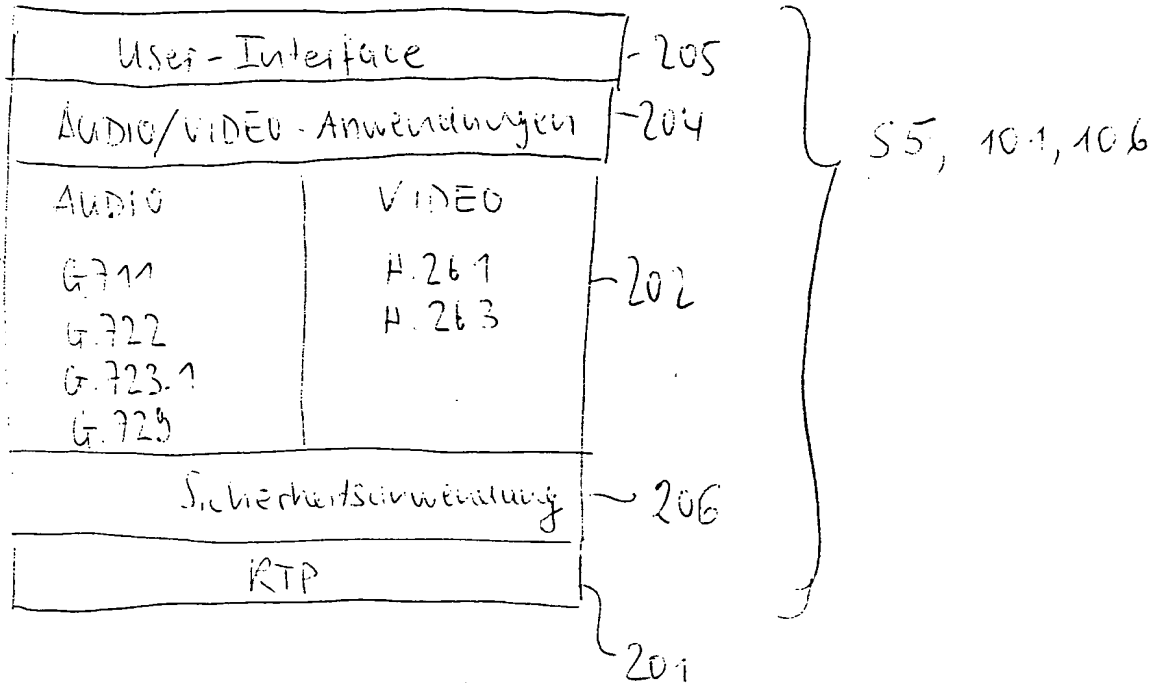

User - Interface
205
AUDIO/VIDEO - Anwendungen
204
AUDIO
G.711
G.722
G.723.1
G.729
VIDEO
H.261
H.263
202
Sicherheitsanwendung
206
RTP
201
55, 101, 106

FIG 3

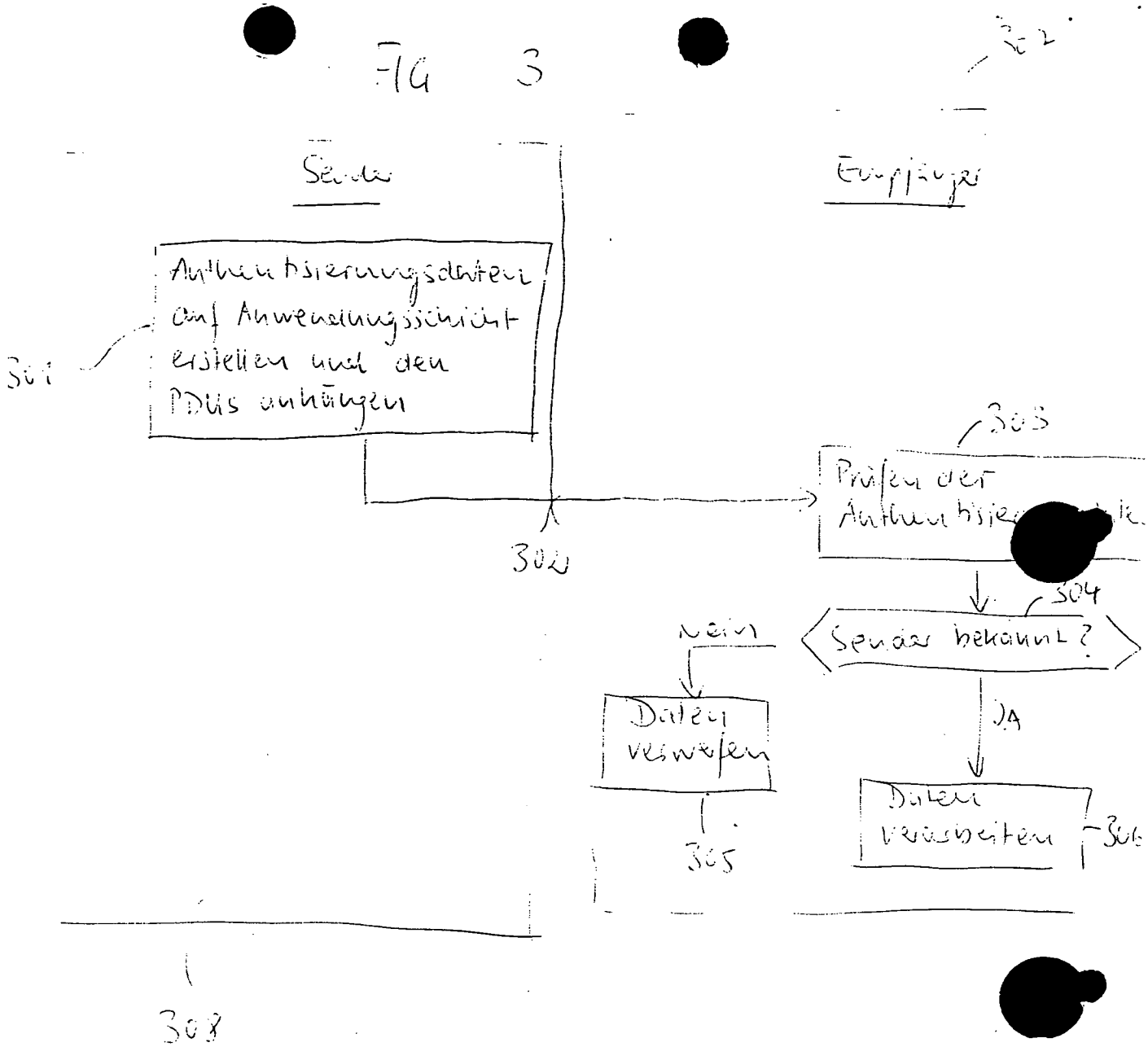
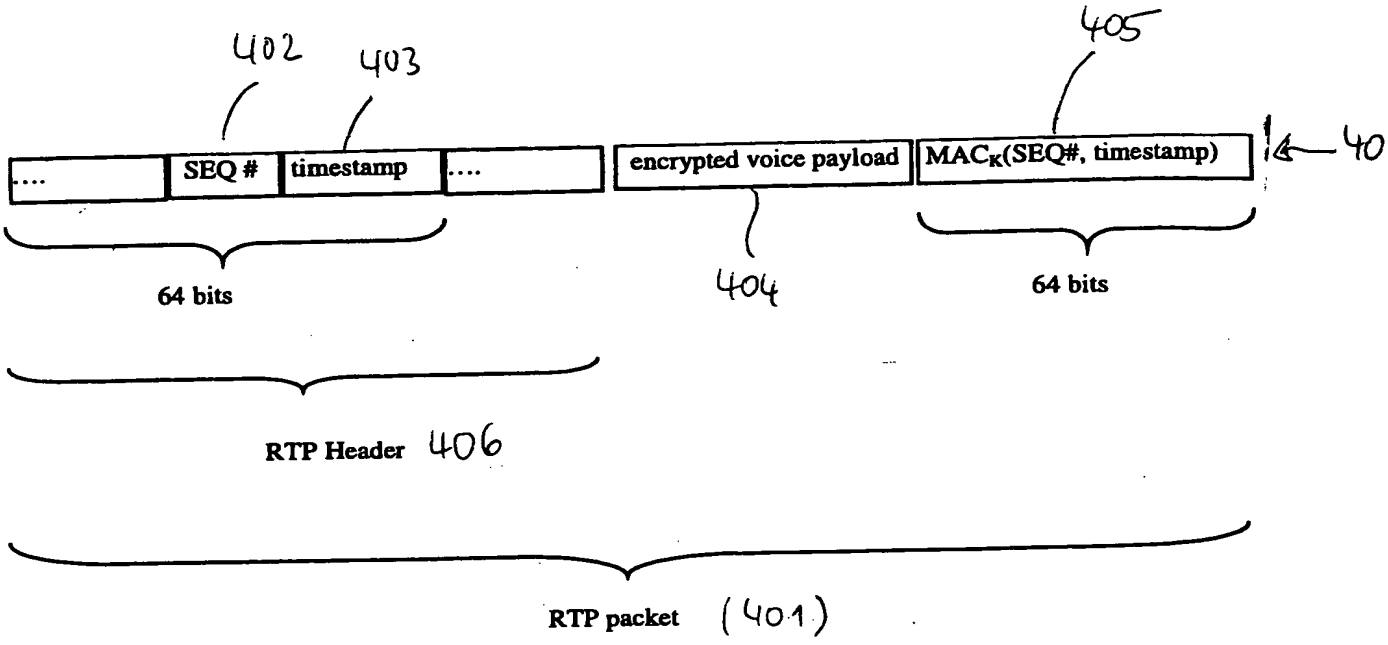

Sender
Empfänger
Authentisierungsdaten auf Anwendungsschicht erstellen und den PDUs anhängen
301
302
Prüfen der Authentisierungsdaten
303
Sender bekannt?
304
Nein
Ja
Daten verwerfen
305
Daten verarbeiten
306
308

FIG 4


402
403
405
....
SEQ #
timestamp
....
encrypted voice payload
MAC$_K$(SEQ#, timestamp)
40
64 bits
404
64 bits
RTP Header 406
RTP packet (401)

THIS PAGE BLANK (USPTO)

00 P1063

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 9/00

A1

(11) International Publication Number: WO 99/11019

(43) International Publication Date: 4 March 1999 (04.03.99)

(21) International Application Number: PCT/US98/17198

(22) International Filing Date: 24 August 1998 (24.08.98)

(30) Priority Data:
08/917,341 26 August 1997 (26.08.97) US

(71) Applicant: V-ONE CORPORATION [US/US]; Suite 300, 20250 Century Boulevard, Germantown, MD 20874 (US).

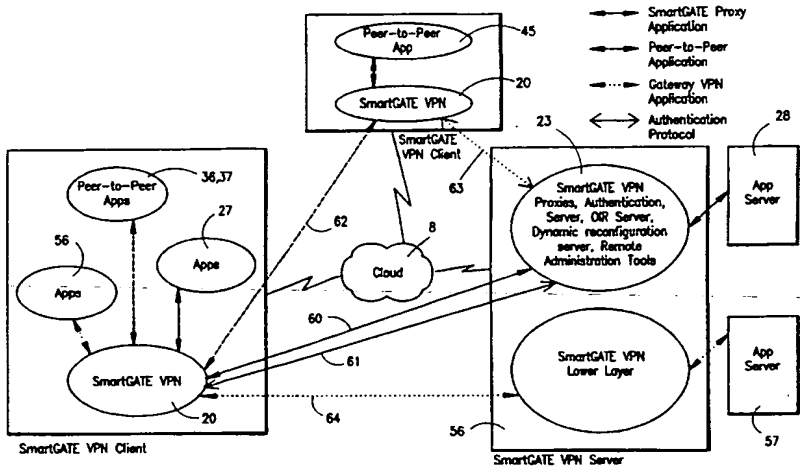
(72) Inventors: CHEN, James, F.; 12648 Tavilah Road, Potomac, MD 20854 (US). WANG, Jieh-Shan; 10903 Silent Wood Place, N. Potomac, MD 20878 (US). BROOK, Christopher, T.; 7308 Pomander Lane, Chevy Chase, MD 20815 (US). GARVEY, Francis; 2908 S. Buchanan Street, Arlington, VA 22206 (US).

(74) Agents: URCIA, Benjamin, E. et al.; Bacon & Thomas, PLLC, 4th floor, 625 Slaters Lane, Alexandria, VA 22314 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: MULTI-ACCESS VIRTUAL PRIVATE NETWORK



(57) Abstract

A virtual private network for communicating between a server and clients over an open network uses an applications level encryption and mutual authentication program (20) and at least one shim (50, 53) positioned above either the layers of a client computer to intercept function calls, communicate with the server and authenticate the parties to a communication and enable the parties to the communication to establish a common session key. Where the parties to the communication are peer-to-peer applications (36, 37, 45), the intercepted function calls, request for service, or data packets include the destination address of the peer application, which is supplied to the server so that the server can authenticate the peer and enable the peer to decrypt further direct peer-to-peer communications (62).

BNSDOCID: <WO___9911019A1_I_>

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

BNSDOCID: <WO___9911019A1_I_>

MULTI-ACCESS VIRTUAL PRIVATE NETWORK

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates a system and method for allowing private communications over an open network, and in particular to a virtual private network which provides data encryption and mutual authentication services for both client/server and peer-to-peer applications at the applications, transport driver, and network driver levels.

2. Discussion of Related Art

A virtual private network (VPN) is a system for securing communications between computers over an open network such as the Internet. By securing communications between the computers, the computers are linked together as if they were on a private local area network (LAN), effectively extending the reach of the network to remote sites without the infrastructure costs of constructing a private network. As a result, physically separate LANs

BNSDOCID: <WO__9911019A1_I_>

can work together as if they were a single LAN, remote computers can be temporarily connected to the LAN for communications with mobile workers or telecommuting, and electronic commerce can be carried out without the risks inherent in using an open network.

In general, there are two approaches to virtual private networking, illustrated in Figs. 1A and 1B. The first is to use a dedicated server 1, which may also function as a gateway to a secured network 2, to provide encryption and authentication services for establishment of secured links 3 between the server 1 and multiple clients 4-6 over the open network 7, represented in Fig. 1A as a cloud, while the second is to permit private communications links 8 to be established between any two computers or computer systems 9-12 on network 7, as illustrated in Fig. 1B.

The advantages of a client/server arrangement such as the one shown in Fig. 1A are that the server can handle functions requiring the majority of the computing resources, increasing the number of potential clients, and that management of the network, including key management is centralized. The disadvantage of a client/server network of this type is that peer-to-peer communications links between applications on the client computers cannot utilize the security and management functions provided by the server, leaving such communications unprotected. On the

BNSDOCID: <WO__9911019A1_I_>

other hand, the advantage of the direct peer-to-peer approach illustrated in Fig. 1B is that it permits secured links to be established between any computers capable of carrying out the required security functions, with the disadvantages being the cost of configuring each computer to carry-out encryption, authentication, and key management functions, and the lack of central control.

In both the client/server and peer-to-peer approaches, a virtual private network can in theory be based either on applications level technology or can operate at a lower level. Generally, however, peer-to-peer "tunneling" arrangements require modification of the lower layers of a computer's communications architecture, while client/server arrangements can use the applications level approach because less modification of the clients is required, and thus the two approaches are in practice mutually exclusive. The present invention, on the other hand, seeks to provide a virtual private network which utilizes a client/server approach, including centralized control of encryption, authentication, and key management functions, while at the same time enabling secured peer-to-peer communications between applications, by utilizing the server to provide authentication and session key generation functions for both client to server communications and peer-to-peer communications, providing a virtual private network capable of serving both as an extended intranet or wide area network (WAN), and as a commercial mass marketing network,

BNSDOCID: <WO___9911019A1_I_>

with high level mutual authentication and encryption provided for all communications.

In order to completely integrate the two approaches and maximize the advantage of each approach, the invention maintains the applications level infrastructure of prior client server private networking arrangements, while adding shims to lower levels in order to accommodate a variety of peer-to-peer communications applications while utilizing the applications level infrastructure for authentication and session key generation purposes. This results in the synergistic effect that not only are existing peer-to-peer tunneling schemes and applications level client server security arrangements combined, but they are combined in a way which greatly reduces implementation costs

In order to understand the present invention, it is necessary to understand a few basic concepts about computer to computer communications, including the concepts of "layers" and communications protocols, and of mutual authentication and file encryption. Further information about layers and protocols can be found in numerous sources available on the Internet, a few of which are listed at the end of this section, while a detailed description of a mutual authentication and encryption system and method suitable for use in connection with the present invention can be found in U.S. Patent No. 5,602,918, which is incorporated herein by reference. In general, the basic

BNSDOCID: <WO___9911019A1_I_>

communications protocols and architecture used by the present invention, as well as authentication, encryption, and key management schemes, are already well-known, and can be implemented as a matter of routine programming once the basic nature of the invention is understood. The changes made by the present invention to the conventional client server virtual private network may be thought of as, essentially, the addition of means, most conveniently implemented as shims, which add a secured mutual authentication and session key generation channel between the server and all parties to a communication, at all levels at which a communication can be carried out.

Having explained the key differences between the present invention and existing systems, the basic concepts of layers and so forth will now be briefly explained by way of background. First, the concept of "layers," "tiers," and "levels," which essential to an understanding of the invention, simply refers to libraries or sets of software routines for carrying out a group of related functions, and which can conveniently be shared or called on by different programs at a higher level to facilitate programming, avoiding duplication and maximizing computer resources. For example, the Windows NT device driver architecture is made up of three basic layers, the first of which is the Network Driver Interface Specification (NDIS 3.0) layer, the second of which is called the Transport Driver Interface (TDI) layer, and the third being the file

BNSDOCID: <WO___9911019A1_I_>

systems. These layers are generically referred to as the network driver layer, the transport or transport driver layer, and the applications layer.

In the Windows NT architecture, the TDI layer formats data received from the various file systems or applications into packets or datagrams for transmission to a selected destination over the open network, while the NDIS layer controls the device drivers that send the data, packets, or IP datagrams, for example by converting the stream of data into a waveform suitable for transmission over a telephone line or a twisted pair cable of the type known as an Ethernet.

By providing layers in this manner, an applications software programmer can design an application program to supply data to the TDI layer without having to re-program any of the specific functions carried out by that layer, and all of the transmission, verification, and other functions required to send a message will be taken care of the TDI layer without further involvement by the applications software. In a sense, each "layer" simply accepts data from the higher layer and formats it by adding a header or converting the data in a manner which is content independent, with retrieval of the data simply involving reverse conversion or stripping of the headers, the receiving software receiving the data as if the intervening layers did not exist.

BNSDOCID: <WO___9911019A1_I_>

In the case of Internet communications, the most commonly used set of software routines for the transport or TDI layer, which takes care of the data formatting and addressing, is the TCP/IP protocol, in which the transport control protocol (TCP) packages the data into datagrams and provides addressing, acknowledgements, and checksum functions, and the internet protocol (IP) further packages the TCP datagrams into packets by adding additional headers used in routing the packets to a destination address. Other transport protocols which can be included in the TDI layer include the user diagram protocol (UDP), the internet control message protocol (ICMP), and non-IP based protocols such as Netbeui or IPX.

Additional "protocols" are may be used at the applications level, although these protocols have nothing to do with the present invention except that they may be included in the applications programs served by the network. Common applications level protocols which utilize the TCP/IP protocol include hypertext transfer protocol (HTTP), simple mail transfer protocol (SMTP), and file transfer protocol (FTP), all of which operate at the layer above the transport layer.

Some applications are written to directly call upon the TCP functions. However, for most applications utilizing a graphical user interface conveniently rely on a set of software routines which are considered to operate

BNSDOCID: <WO___9911019A1_I_>

above the TDI layer, and are known as sockets. Sockets serve as an interface between the TCP set of functions, or stack, and various applications, by providing libraries of routines which facilitate TCP function calls, so that the application simply has to refer to the socket library in order to carry out the appropriate function calls. For Windows applications, a commonly used non-proprietary socket is the Windows socket, known as Winsock, although sockets exist for other operating systems or platforms, and alternative sockets are also available for Windows, including the Winsock 2 socket currently under development.

In order to implement a virtual private network, the encryption and authentication functions must be carried out at one of the above "levels," for example by modifying the network drivers to encrypt the IP datagrams, by inserting authentication headers into the TCP/IP stacks, or by writing applications to perform these functions using the existing drivers. If possible, it is generally desirable to minimize modification of the existing levels by adding a layer to perform the desired functions, calling upon the services of the layer below, while utilizing the same function calls so that the higher layer also does not need to be modified. Such a layer is commonly referred to as a "shim."

As indicated above, the preferred approach to implementing client/server virtual private networks is to

BNSDOCID: <WO___9911019A1_I_>

use an applications level security system to encrypt files to be transmitted, and to then utilize existing communications layers such as Winsock, or TCP/IP directly. This is the approach taken by the commercially available access control system known as SmartGATE™, developed by V-One Corp. of Germantown, Md., which provides both encryption and mutual authentication at the applications level utilizing a dedicated server known as an authentication server and authentication client software installed at the applications level on the client computers. A description of the manner in which encryption and mutual authentication is carried out may be found in the above-cited U.S. Patent No. 5,602,918. While the principles of the invention are applicable to other client/server based virtual private networks, SmartGATE™ is used as an example because it provides the most complete range of mutual authentication and encryption services currently available.

The present invention can be implemented using the existing SmartGATE™ system, but adds mutual authentication and encryption services to lower layers by intercepting function calls or data packets and, during initialization of a communications link, establishing separate channels between the party initiating the communication and the authentication server, and between the authentication server and the party which is to share in the communication, so as to mutually authenticate the parties

BNSDOCID: <WO__9911019A1_I_>

with respect to the server, and so as to establish a session key which can be used for further direct communications between the parties.

A number of protocols exist which can be used, in total or in part, to implement the mutual authentication and encryption services at the lower layers, using the same basic authentication and encryption scheme currently implemented by SmartGATE™ at the applications level. These include, by way of example, the SOCKS protocol, which places a shim between the TDI or transport layer and the applications, and the commercially available program, known as SnareNet, which operates at the network driver level and can be directly utilized in connection with the present invention.

On the other hand, a network level implementation such as the SKIP protocol, which operates below the TDI layer to encrypt the datagrams, and which in its description explicitly precludes the generation of session keys (see the above cited U.S. Patent No. 5,602,918), is fundamentally different in concept than the present invention. Similarly, alternative implementations such as Point-to-Point Tunneling Protocol (PPTP) which involve modifying the TCP/IP stack and/or hardware to provide encryption, as opposed to inserting shims, are not utilized by the preferred embodiment of the present invention, although individual aspects of the protocol could perhaps

BNSDOCID: <WO___9911019A1_I_>

be used, and the present system could be added to computers also configured to accept PPTP communications.

The SmartGATE™ system uses public key and DES encryption to provide two-way authentication and 56-bit encrypted communications between a server equipped with the SmartGATE program and client computers equipped with a separate program. Currently, SmartGATE™ operates at the highest level, or applications level, by using shared secret keys to generate a session key for use in further communications between the authentication server or gateway and the client program. Since the session key depends on the secret keys at the gateway and client sides of the communication, mutual authentication is established during generation of the session key, which can then be used to encrypt further communications.

When installed on a client system, the SmartGATE™ client software reads a request for communications by an applications program, such as a browser program, and then proceeds to establish its own communications link with the destination server to determine if the server is an authentication server. If it is not, control of communications is relinquished, but if it is, then the security program and the server carry out a challenge/response routine in order to generate the session key, and all further communications are encrypted by the security program. Although this program is placed between

BNSDOCID: <WO__9911019A1_I_>

the Winsock layer and the applications, it does not function as a shim, however, because it only affects communications directed to the authentication server.

Having briefly summarized the concepts used by the present invention, including the concepts of layers, protocols, and shims, and having described a specific applications level security program which is to be modified according to the present invention by adding shims in a way which enables secured authentication and session key generation channels to be set up from the lower layers, it should now be possible to understand the nature of the invention, and in particular how it integrates the two approaches to virtual private networking in a way which greatly expands the concept and yet can easily be implemented. More details will be given below, but as a final observation in this background portion of the patent specification, it should be noted that while the overall concept of the invention is in a sense very simple, it is fundamentally at odds with present approaches. For example, the literature is replete with references to conflicts between VPN standards and implementations, as exemplified by the title of an article from LAN Times On-Line, 9/96, (http://www.wcmh.com/), which reads *Clash Over VPN Supremacy*. Even a cursory search of the available literature indicates that the amount of information and choices available to those wishing to set up a virtual private network is overwhelming. One can choose between

BNSDOCID: <WO___9911019A1_I_>

Netscape Communications Secure Socket Layer, Open Market Inc.'s Secure HTTP, Microsoft's PPTP, among others. However, all of these approaches operate at a single level, and force a choice between establishing a network of the type shown in Fig. 1A and a network of the type shown in Fig. 1B. Only the present invention offer the advantages of both approaches, without the inflexibility of client/server arrangements or the costs of more distributed architectures.

For further information on the various competing VPN protocols and systems, see also *The Development of Network Security Technologies*, Internet Smartsec, 2/97 (http://www.smartsec.se), which compares SmartGATE™ to other application level security systems, including PPTP, SSL, and S-HTTP; *Point-To-Point Tunneling Protocol (PPTP) Frequently Asked Questions*, Microsoft Corp., date unknown, (http://www.microsoft.com), *Simple Key-Management for Internet Protocols (SKIP)*, Aziz *et al.*, date unknown, (http://skip.incog.com), and *SOCKS Protocol Version 5, RFC 1928*, Leech *et al.*, 3/96 (http://andrew2.andrew.cmu.edu) (this document describes a protocol involving a TDI shim). For more general information on security problems, Internet protocols, and sockets, see *Introduction to the Internet Protocols*, Charles L. Hedrick, Rutgers University, 1987 (http://oac3.hsc.uth.tmc.edu); *Windows Sockets - Where Necessity is the Mother of Reinvention*, Stardust

BNSDOCID: <WO__9911019A1_I_>

Technologies, Inc., 1996, (http//www.stardust.com), and *Secure Internet Connections*, LAN Times, 6/17/96 (Ibid).

SUMMARY OF THE INVENTION

It is accordingly a principal objective of the invention to provide a client/server virtual private network which is capable not only of carrying out authenticated secure communications over an open network between an authentication server and clients, but also authenticated secure peer-to-peer communications.

It is also an objective the invention to provide a virtual private network that provides data encryption and mutual authentication for both client/server and peer-to-peer communications for different-types of applications, using both the applications level and lower levels of a communications hierarchy.

It is a further objective of the invention to provide a client/server virtual private network which can provide both client/server and peer-to-peer encryption and authentication services for any application sharing a specified socket or sockets, whether or not the application is recognized by the encryption and authentication program.

It is a still further objective of the invention to provide a client/server virtual private network which can

BNSDOCID: <WO___9911019A1_I_>

provide encryption and authentication services at the applications level, transport driver interface level, and network interface level, without the need for modifying either the communication driver or network driver, or any sockets utilizing the communications driver interface.

It is yet another objective of the invention to provide a virtual private network which provides encryption and authentication services for peer-to-peer communications while maintaining centralized control of key distribution and management functions.

Finally, it is also an objective of the invention to provide a virtual private network which provides encryption and authentication services for peer-to-peer communications and in which registration is carried out by a central gateway server.

These objectives of the invention are accomplished by providing a virtual private network for communicating between a server and clients over an open network and in which the clients are equipped with an applications level encryption and mutual authentication program which includes at least one shim positioned above either the socket, transport driver interface, or network interface layers of a client computers communications hierarchy, and which intercepts function calls or data packets in order to authenticate the parties to the communication by

Meta 1002
p. 400

BNSDOCID: <WO___9911019A1_I_>

establishing secured channels between the server and the parties to the communication, prior to establishment of the secured communications link between the parties, in order to carry out mutual authentication and session key generation functions.

More particularly, according to the principles of a preferred embodiment of the invention, client communications software is provided which, at the socket or transport driver interface levels, intercepts function calls to the socket or transport driver and directs calls to the authentication server in order to perform encryption and authentication routines, and at the network driver interface, performs encryption and authentication functions by intercepting the datagrams or data portions of the packets transmitted by the transport driver interface based on communications between the authentication server and the client. According to this aspect of the invention, a system of providing authentication and encryption services for the purpose of establishing a virtual private network includes a plurality of shims arranged to operate at different protocol levels in order to establish a common secure communications link to an authentication server.

In one especially preferred embodiment of the invention, the client software includes a Winsock shim arranged to intercept function calls to the Winsock library on a client machine and redirect initial communications

BNSDOCID: <WO__9911019A1_I_>

through the authentication client software to the authentication server, so that any function calls to the Winsock library of programs are intercepted by the shim and carried out by the applications level security program. In this embodiment, the client authentication software substitutes its own function calls for the original function calls in order to establish a secured communications link to the authentication server over which such functions as mutual authentication between the client and server, indirect authentication of peer applications by the now trusted server, session key generation, are carried out, as well as ancillary functions such as on-line registration (OLR), utilizing the unmodified original Winsock library and TCP/IP communications stacks.

By inserting a shim at the Winsock level, an applications level client/server based security program such as SmartGATE™ can be used to provide secure communications for any application which utilizes the Winsock library. In addition, by including analogous shims at other levels, the invention can be used to secure virtually any communications application, including those which by-pass the TDI layer and communicate directly with the network driver level.

Instead of the current array of mutually exclusive alternative methods and systems of establishing secured communications over an open network, the invention thus

BNSDOCID: <WO___9911019A1_I_>

provides a single integrated method and system capable of carrying out both client/server communications and peer-to-peer communications between a wide variety of communications applications regardless of whether the applications use a socket or even commonly accepted internet protocols, with complete mutual authentication and encryption of data files at all levels and between all parties to the network.

It will be appreciated that the term "virtual private network" is not to be taken as limiting, and that the principles of the invention can be applied to any remote access schemes which utilize the Internet or other relatively insecure networks to provide access for remote users, corporate intranets, and electronic commerce.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1A is a schematic diagram of a client/server virtual private network.

Fig. 1B is a schematic diagram of an alternative virtual private network based on peer-to-peer communications.

Fig. 2 is a functional block diagram showing the operation of an applications level security program in a conventional communications network hierarchy.

BNSDOCID: <WO___9911019A1_I_>

Fig. 3 is a functional block diagram showing the communications network hierarchy of Fig. 1, modified to provide a second layer of service in accordance with the principles of a preferred embodiment of the invention.

Fig. 4 is a functional block diagram showing the communications network hierarchy of Fig. 2, modified to provide a third layer of service in accordance with the principles of the preferred embodiment.

Fig. 5 is a functional block diagram showing the communication network hierarchy of Fig. 3, modified to provide a fourth layer of service in accordance with the principles of the preferred embodiment.

Fig. 6 is a schematic diagram of a virtual private network utilizing the principles of the preferred embodiment of the invention.

Fig. 7 is a flowchart illustrating a method of implementing the system of the preferred embodiment.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Fig. 2 illustrates the operation of a client authentication program which is utilized in the present invention. An example of such a program is the SmartGATE™ program discussed briefly above, although other

BNSDOCID: <WO___9911019A1_I_>

applications level security programs, whether or not token based, could be modified in a manner similar to that discussed in the following description. The illustrated hierarchy is the Windows NT architecture, although versions of SmartGATE™ exist for other architectures, and the invention could easily be adapted for use with any version of SmartGATE™, including UNIX and MacIntosh versions, as well as for use with applications level security programs designed for communications architectures other than those supported by SmartGATE™. Conversely, it is intended that the present invention can be used with authentication and encryption schemes other than that used by SmartGATE™ and disclosed in U.S. Patent No. 5,602,918. For purposes of convenience, therefore, the software represented by SmartGATE™ is simply referred to as client authentication software.

In addition, it noted that the client computer architectures illustrated in Figs. 3-6, which are modified versions of the architecture of Fig. 2, is to be used with an overall network layout such as the one illustrated in Fig. 6, which includes an authentication server that may be a SmartGATE™ server, or another server depending on the client authentication software. The invention is not merely the addition of shims to the client software, but involves the manner in which the shims are used in the establishment of the authentications and key generation links to the server.

BNSDOCID: <WO___9911019A1_I_>

Turning to Fig. 2, which provides background for the description of the invention illustrated in Figs. 3-6, the client authentication software 20 is situated above the boundary of the transport or TDI layer 21 and is designed to utilize a socket 22, such as Winsock, to carry out communications with the authentication server 23 shown in Fig. 6 by means of a transport protocol such as TCP/IP, UDP, or the like, which in turn supply datagrams or packets to a hardware driver layer 24, such as NDIS 3.0, of a network or modem connection 25.

In operation, the client authentication software 20 intercepts interconnect calls 26 form client authentication software supported applications 27 and, if the calls are directed to the authentication server 23, or to a server 28 situated on a secured network whose access is controlled by the authentication server, establishes a secured communications link to the server by executing appropriate function calls 29 to the socket library, which in turn transmits function calls 30 to the TDI layer, causing the TDI layer to form datagrams or packets 31. Datagrams or packets 31 are then formatted over packaged for transmission by the hardware drivers 24 and sent to the communications network in the form of Ethernet packets or analog signals 32 containing the original datagrams from the TDI layer. Once the secured communications link has been established, client authentication software 20 encrypts all further data communications 34 from

Meta 1002
p. 406

BNSDOCID: <WO___9911019A1_I_>

applications 27, which are indicated by dashed lines, before handing them off to the next lower layer in the form of encrypted files 35. The dashed lines are shown in Fig. 2 as extending only to the TDI layer 21, because the datagrams formed by the TDI layer are indistinguishable as to content, but it is to be understood that datagrams or packets 31 carry both the communications used to establish the secure channel, and the encrypted files subsequently sent therethrough.

Finally, in the case of SmartGATE™, the authentication client software utilizes either a smart card or secured file to supply the secret keys used during authentication to generate a session key for encryption of further communications, and also to carry out certain other encryption and authentication functions, although it is of course within the scope of the invention to use key distribution and authentication methods which do not rely on smartcards or tokens, and the tokens are not involved in any of the basic communications functions of the client authentication software 20.

In addition to the applications 27 which communicate with the server via the authentication/encryption software 20, a typical system will have a number of additional software applications 36 and 37 capable of carrying out communications over the open network, but which the authentication client software is not configured to handle,

BNSDOCID: <WO___9911019A1_I_>

and which are not specifically adapted or intended to carry out communications with the authentication server. These are referred to herein as peer-to-peer applications, and can include applications which use the same sockets as the authentication client software, applications which directly call upon a transport driver interface stack, whether using the same protocol as the authentication client software or another protocol, all of which are intended to be represented by the TDI layer, and applications which are written to call directly upon the hardware drivers. These peer-to-peer applications may have their own encryption and authentication capabilities, but cannot utilize the services of the authentication server or client software, and therefore the function calls made by the applications and the files transmitted are indicated by separate reference numerals 40-43.

It will be appreciated by those skilled in the art that lower layer application programs which generate packets in forms other than those represented by the TDI layer are also possible, and should be considered within the scope of the invention, but at present virtually all open network applications use at least one of the TDI protocols, and thus while these programs may interact directly with the network driver layer, and require a network driver layer shim, as will be discussed below, are illustrated for purposes of convenience as part of the TDI layer applications.

BNSDOCID: <WO___9911019A1_I_>

Turning now to a preferred embodiment of the invention, the arrangement shown in Fig. 3 modifies the arrangement of Fig. 2 by adding a socket shim 50 between the socket 22 utilized by the authentication client software 20, the peer-to-peer applications 36 which also utilize the socket 20, and the authentication client software itself. The shim 50 operates by hooking or intercepting call initiation function calls 40 made to the socket and, in response thereto, having the authentication client software initiate communications with the authentication server 23, shown in Fig. 6, in order to carry out the authentication protocol, as will be discussed in more detail below. Shim 50 also causes files 41 intended for the TDI layer to be diverted to the authentication software for encryption based on the session keys generated during the initial communications with the authentication server, and transmission as encrypted files 51 addressed to the peer application, also shown in Fig. 6, which could also be an application on the application server 28.

Since the basic authentication client software is designed to send all communications directly to the authentication server, while the peer-to-peer applications are designed only to communicate with "peers" 45 and not with the authentication server, the principal function of shim 50 is to arrange for the destination of address of the communication to be supplied to both the authentication

BNSDOCID: <WO___9911019A1_I_>

client software and to authentication server, even though the peer application assumes that it is communicating only with the peer application. This function permits session key encrypted communications to be forwarded directly to the peer application, as illustrated in Fig. 6, while the latter function provides the authentication server with the client address so that the authentication server can establish a secured and authenticated link with the peer application, via authentication client software on the peer computer, and transmit the session key to the peer application or at least enable the peer application to recreate the session so that it can decrypt the encrypted files received directly from the client application.

Thus, while it is appreciated that the use of socket shims is well-known, as mentioned above, the socket shim shown in Fig. 2 has the unique function of enabling direct peer-to-peer communications with mediation by the authentication server, permitting the highest level of authentication service and collateral functions. In addition, because of the mediation by the key server, the peer applications do not need to have a shared secret key, allowing centralized key management, with only the authentication server having access to all of the client's secret keys.

Figs. 4 shows the variation of the client authentication software 20 in which a TDI shim 52 similar

BNSDOCID: <WO___9911019A1_I_>

in function to the socket shim 50 is provided above the TDI layer. Like the socket shim, implementation of the TDI shim essentially simply involves diverting certain information to the client software in order to establish a communications link with the authentication server, and subsequently perform encryption to obtain encrypted files 54 for transmission directly through the TDI layer in the usual manner. As with the socket shim, TDI shims are not new and can be implemented in known manner, by intercepting TDI service requests, but with the difference from prior TDI shims that the TDI shim works with the authentication software 20 and authentication server to authenticate communications and generate a session key.

Finally, as shown in Fig. 5, a further layer of authentication and encryption may be added by adding a network driver shim 55, either to the arrangement shown in Fig. 3 without the TDI shim, in combination with the TDI shim shown in Fig. 4, or in combination with the TDI shim of Fig. 4 but not the socket shim, to provide for authentication of communications at the network driver layer. At this layer, the shim 55 intercepts IP packets from applications 56, but instead of referring back to the applications level routine, checks the destination address (which can be in TCP format, UDP format, and so forth), establishes a session key by communications with the authentication server, converts the session key into a format which can be used to encrypt the IP packet, and

BNSDOCID: <WO___9911019A1_I_>

sends the IP packet towards the destination, all by carrying out the necessary operations at the network driver level, in a manner similar to that utilized by the above-mentioned SnareNet software program, but with the difference that the authenticating communications link and key generation is carried out by packets addressed to a corresponding layer 56 of the authentication server, which may be further connected to an applications server 57.

It will be noted that since the IP packets are not distinguishable by content, the network driver layer shim could be used as an additional level of security, rather than as an alternative to applications level encryption, with the encrypted files generated by software 20 being further encrypted by shim 55 before transmission to the authentication server or associated gateway.

The overall system utilizing the authentication client software illustrated in Figs. 3-5 is schematically illustrated in Fig. 6. The principal components of the overall system are the client computers containing software of the type illustrated in Figs. 2-5, including client authentication software 20 and shims 50, 53, and/or 55, and applications with communications capabilities (represented by applications 27, 36, 37, and 56 on one client, and application 45 on the other). For purposes of illustration, the client of Figs. 6 is thus depicted as including applications for communicating at the highest

BNSDOCID: <WO__9911019A1_I_>

levels, such as the SmartGATE™ proxy application, applications for communicating at the network driver level with corresponding applications connected to the lower layer of the authentication server, and peer-to-peer applications with no capability of communicating with SmartGATE™, but which use sockets or TDI protocols recognized by the shims.

In the case of the SmartGATE™ proxy application, communications are established in the same manner as in the currently available version of the SmartGATE™ authentication client software, and as described in U.S. Patent No. 5,602,918, the communications link being indicated by arrows 60 and 61, with arrow 60 representing the client/server response channel used to authenticate the parties and generate the session key.

In the case of a peer-to-peer application, in which the clients wish to communicate over a direct link 62, the invention provides for the function calls establishing the communications to be intercepted and the initialization procedure routed through channel 61 to the authentication server 23. Server 23 then opens a secured channel 63 to the authentication client software 20 associated with peer application 45 by performing the same mutual authentication procedure performed for the purpose of establishing channel 63, and once the channel is established with its own session key, transmits information using the channel 63

BNSDOCID: <WO__9911019A1_I_>

session key which allows the client to recreate the channel 60 session key for use in decrypting communications sent over channel 62. Alternatively, after establishing channel 63, the channel 60 session key could be used to transmit back to the original sending party information necessary to recreate the channel 63 session key. In either case, the authentication server is thus used to establish a fully authenticated "tunnel" between the peer applications without the need to modify any of the sockets, TDI protocols, or hardware drivers on either of the client computers. While the transmitting peer application has no way of directly authenticating the receiving peer, only a receiving peer authenticated by the authentication server will be able to generate the necessary session keys, and thus each of the parties to the communication is effectively authenticated.

For the lower layer application 56, a similar protocol may be employed, in which the attempted communication between lower layer applications is intercepted, and the communications link to the authentication server is used to generate a session key, which is then used to encrypt the packets or datagrams being sent. In this case, the destination must be the lower layer of the authentication server, and thus the communications link is indicated by a separate channel 67.

BNSDOCID: <WO__9911019A1_I_>

Finally, the procedures associated with the network illustrated in Fig. 6 are summarized in the flowchart of Fig. 7. For communications directly with the applications level portion of the server 23, steps 100-103 are used, while for peer-to-peer communications, steps 104-109 are used, and for network driver level communications, steps 110-114 are used.

In particular, step 100 by which the applications level authentication program 20 illustrated in Figs. 3-5 receives a call initiation request, either directly from a supported applications program 27 or from a programs 36 and 37 via one of the shims 50 and 53, step 101 is step by which the program 20 addresses the authentication server, step 102 is the step by which the client and server are mutually authenticated and the session keys generated using, for example, the procedure described in U.S. Patent No. 5,602,918, and step 103 is the step by which program 20 encrypts further communications received directly or via shims 50 and 53 from the applications programs 27, 36, and 37.

For peer-to-peer communications, step 105, which is part of step 100, is the step by which the peer address is supplied to program 20, steps 106 and 107 are identical to steps 101 and 102, step 108 is the step by which communications channel 63 shown in Figure 6 is established, step 109 is the step by which the destination computer

BNSDOCID. <WO__9911019A1_I_>

authenticated by the server is enabled to decrypt communications received over channel 62, and step 110 is the step by which program 20 encrypts the communications. It will of course be appreciated that these steps represent only a summary of the steps involved in carrying out the present invention, and that further steps will be apparent to those skilled in the art based on the above description of the apparatus and software portions of the preferred embodiment of the invention.

Having thus described various preferred embodiments of the invention, those skilled in the art will appreciate that variations and modifications of the preferred embodiment may be made without departing from the scope of the invention. It is accordingly intended that the invention not be limited by the above description or accompanying drawings, but that it be defined solely in accordance with the appended claims.

BNSDOCID: <WO___9911019A1_I_>

I claim:

1. Apparatus for carrying out communications over a multi-tier virtual private network, said network including a server and a plurality of client computers, the server and client computers each including means for transmitting data to and receiving data from an open network, comprising:

means for intercepting function calls and requests for service sent by an applications program on one of said client computers to a lower level set of communications drivers; and

means for causing an applications level authentication and encryption program in said one of said client computers to communicate with the server, generate said session key, and encrypt files sent by the applications program before transmittal over said open network.

2. Apparatus as claimed in claim 1, further comprising means for intercepting files packaged by a transport driver interface layer to form packets and encrypting the packets using a session key generated during communications with a lower layer of the server.

3. A method as claimed in claim 1, further comprising

means for intercepting a destination address during initialization of communications between said one of said

BNSDOCID: <WO___9911019A1_I_>

client computers and a second of said client computers on said virtual private network;

means for causing said applications level authentication and encryption program to communicate with the server to carry out functions a.) and b.);

means for transmitting said destination address to said server;

means for causing said server to carry-out functions a.) and b.) with respect to the second of said two client computers;

means for enabling said second of said two client computers to recreate the session key;

means for causing said authentication software to encrypt files to be sent to the destination address using the session key; and

means for transmitting the encrypted files directly to the destination address.

4. Apparatus as claimed in claim 3, wherein said means for intercepting the destination address is carried out by a shim positioned between a peer-to-peer applications program and a layer of a communications driver architecture of said one of the two client computers.

5. A multi-tier virtual private network, comprising:

a server and a plurality of client computers, the server and client computers each including means for

BNSDOCID: <WO___9911019A1_I_>

transmitting data to and receiving data from an open network,

wherein said means for transmitting data to and receiving data from the open network includes, in any client computer initiating communications with the server:

applications level encryption and authentication software arranged to communicate with the server in order to: a.) mutually authenticate the server and the client computer initiating communications with the server and b.) generate a session key for use by the client computer initiating communications to encrypt files;

at least one lower level set of communications drivers;

and a shim arranged to intercept function calls and requests for service sent by an applications program to the lower level set of communications drivers in order to cause the applications level authentication and encryption program to communicate with the server, generate said session key, and encrypt files sent by the applications program before transmittal over said open network.

6. A multi-tier virtual private network as claimed in claim 5, wherein said lower level set of communications drivers includes a network driver layer, a transport driver

BNSDOCID: <WO___9911019A1_I_>

interface layer arranged to package applications files as packets capable of being routed over the open network and supply the packets to the network driver layer for transmission to the open network, and an applications socket for facilitating service requests by said applications program to the transport driver interface layer, and wherein said shim is a socket shim positioned between the applications program and the socket to intercept function calls to the socket in order to cause the applications level authentication and encryption program to communicate with the server, generate said session key, and encrypt files sent by the applications program before the files are packaged by the transport driver interface layer.

7. A multi-tier virtual private network as claimed in claim 6, wherein said applications program is a peer-to-peer communications program, and wherein a peer application destination address, included in said function calls to the socket, is diverted by the socket shim and wherein a destination address including said intercepted function calls is supplied to the server during communications with the server, causing the service to establish a communications link with a peer application, mutually authenticate the peer application, and enable the peer application to reconstruct the session key in order to receive encrypted files sent by the peer-to-peer communications program over the open network.

BNSDOCID: <WO___9911019A1_I_>

8. A multi-tier virtual private network as claimed in claim 6, further including a transport driver interface shim positioned between the transport driver interface layer and a second applications program, for intercepting requests from the second applications program for service by the transport driver interface layer in order to cause the applications level authentication and encryption program to communicate with the server, generate said session key, and encrypt files sent by the applications program before the files are packaged by the transport driver interface layer.

9. A multi-tier virtual private network as claimed in claim 8, further comprising a network driver layer shim positioned between the network driver layer and the transport driver interface layer and arranged to intercept files packaged by the transport driver interface layer and encrypt the files using a session key generated during communications with a lower layer of the server.

10. A multi-tier virtual private network as claimed in claim 5, wherein said lower level set of communications drivers includes a network driver layer, and a transport driver interface layer arranged to package applications files as packets capable of being routed over the open network and supply the packets to the network driver layer for transmission to the open network, and wherein said shim is a transport driver interface layer shim positioned

BNSDOCID: <WO___9911019A1_I_>

between the applications program and the transport driver interface layer to intercept service requests by the applications program to the transport driver interface layer in order to cause the applications level authentication and encryption program to communicate with the server, generate said session key, and encrypt files sent by the applications program before the files are packaged by the transport driver interface layer.

11. A multi-tier virtual private network as claimed in claim 10, wherein said applications program is a peer-to-peer communications program, and wherein a peer application destination address, included in said intercepted requests for service, is diverted by the transport driver interface layer shim and supplied to the server during communications with the server, causing the service to establish a communications link with a peer application, mutually authenticate the peer application, and enable the peer application to reconstruct the session key in order to receive encrypted files sent by the peer-to-peer communications program over the open network.

12. A multi-tier virtual private network as claimed in claim 10, further comprising a network driver layer shim positioned between the network driver layer and the transport driver interface layer and arranged to intercept files packaged by the transport driver interface layer and

BNSDOCID: <WO___9911019A1_I_>

encrypt the files using a session key generated during communications with a lower layer of the server.

13. A multi-tier virtual private network, comprising:

a server and a plurality of client computers, the server and client computers each including means for transmitting data to and receiving data from an open network,

wherein said means for transmitting data to and receiving data from the open network includes, in any client computer initiating communications with the server:

applications level encryption and authentication software arranged to communicate with the server in order to: a.) mutually authenticate the server and the client computer initiating communications with the server and b.) generate a session key for use by the client computer initiating communications to encrypt files; and

at least one lower level set of communications drivers,

wherein said lower level set of communications drivers includes a network driver layer, a transport driver interface layer arranged to package applications files as packets capable of being routed over the open network and supply the packets to the network driver layer for transmission to the open network, and a

BNSDOCID: <WO___9911019A1_I_>

network driver layer shim positioned between the transport driver interface layer and the network driver layer and arranged to intercept files packaged by the transport driver interface layer and encrypt the files using a session key generated during communications with a lower layer of the server.

14. A multi-tier virtual private network, comprising:

a server and a plurality of client computers, the server and client computers each including means for transmitting data to and receiving data from an open network,

wherein said means for transmitting data to and receiving data from the open network includes, in any client computer initiating communications with the server:

applications level encryption and authentication software arranged to communicate with the server in order to: a.) mutually authenticate the server and the client computer initiating communications with the server and b.) generate a session key for use by the client computer initiating communications to encrypt files; and

further comprising means for securing peer-to-peer communications between applications on two of said client computers, said peer-to-peer communications securing means comprising:

BNSDOCID: <WO__9911019A1_I_>

means for intercepting a destination address during initialization of communications by a first of said two client computers;

means for causing said authentication software to communicate with the server to carry out functions a.) and b.);

means for transmitting said destination address to said server;

means for causing said server to carry-out functions a.) and b.) with respect to the second of said two client computers;

means for enabling said second of said two client computers to recreate the session key;

means for causing said authentication software to encrypt files to be sent to the destination address using the session key;

means for transmitting the encrypted files directly to the destination address.

15. A multi-tier virtual private network as claimed in claim 14, wherein said means for intercepting the destination address comprises a shim positioned between the peer-to-peer applications program and a layer of a communications driver architecture of said first of the two client computers.

16. A multi-tier virtual private network as claimed in claim 5, wherein said shim is positioned above a socket,

BNSDOCID: <WO___9911019A1_I_>

the socket being positioned above a transport driver layer of said communications driver architecture.

17. A multi-tier virtual private network as claimed in claim 5, wherein said shim is positioned above a transport driver layer of said communications driver architecture.

18. Computer software for installation on a client computer of a multi-tier virtual private network, said network including a server and a plurality of client computers, the server and client computers each including means for transmitting data to and receiving data from an open network,

wherein said computer software includes:

applications level encryption and authentication software arranged to communicate with the server in order to: a.) mutually authenticate the server and the client computer initiating communications with the server and b.) generate a session key for use by the client computer initiating communications to encrypt files;

and a shim arranged to intercept function calls and requests for service sent by an applications program to a lower level set of communications drivers in order to cause the applications level authentication and encryption program to communicate with the server, generate

BNSDOCID: <WO___9911019A1_I_>

said session key, and encrypt files sent by the applications program before transmittal over said open network.

19. Computer software as claimed in claim 18, wherein said lower level set of communications drivers includes a network driver layer, a transport driver interface layer arranged to package applications files as packets capable of being routed over the open network and supply the packets to the network driver layer for transmission to the open network, and an applications socket for facilitating service requests by said applications program to the transport driver interface layer, and wherein said shim is a socket shim positioned between the applications program and the socket to intercept function calls to the socket in order to cause the applications level authentication and encryption program to communicate with the server, generate said session key, and encrypt files sent by the applications program before the files are packaged by the transport driver interface layer.

20. Computer software as claimed in claim 19, wherein said applications program is a peer-to-peer communications program, and wherein a peer application destination address, included in said function calls to the socket, is diverted by the socket shim and wherein a destination address including said intercepted function calls is supplied to the server during communications with the

BNSDOCID: <WO___9911019A1_I_>

server, causing the service to establish a communications link with a peer application, mutually authenticate the peer application, and enable the peer application to reconstruct the session key in order to receive encrypted files sent by the peer-to-peer communications program over the open network.

21. Computer software as claimed in claim 19, further including a transport driver interface shim positioned between the transport driver interface layer and a second applications program, for intercepting requests from the second applications program for service by the transport driver interface layer in order to cause the applications level authentication and encryption program to communicate with the server, generate said session key, and encrypt files sent by the applications program before the files are packaged by the transport driver interface layer.

22. Computer software as claimed in claim 21, further comprising a network driver layer shim positioned between the network driver layer and the transport driver interface layer and arranged to intercept files packaged by the transport driver interface layer and encrypt the files using a session key generated during communications with a lower layer of the server.

23. Computer software as claimed in claim 18, wherein said lower level set of communications drivers includes a

BNSDOCID: <WO___9911019A1_I_>

network driver layer, and a transport driver interface layer arranged to package applications files as packets capable of being routed over the open network and supply the packets to the network driver layer for transmission to the open network, and wherein said shim is a transport driver interface layer shim positioned between the applications program and the transport driver interface layer to intercept service requests by the applications program to the transport driver interface layer in order to cause the applications level authentication and encryption program to communicate with the server, generate said session key, and encrypt files sent by the applications program before the files are packaged by the transport driver interface layer.

24. Computer software as claimed in claim 23, wherein said applications program is a peer-to-peer communications program, and wherein a peer application destination address, included in said intercepted requests for service, is diverted by the transport driver interface layer shim and supplied to the server during communications with the server, causing the service to establish a communications link with a peer application, mutually authenticate the peer application, and enable the peer application to reconstruct the session key in order to receive encrypted files sent by the peer-to-peer communications program over the open network.

Meta 1002
p. 429

BNSDOCID: <WO___9911019A1_I_>

25. Computer software as claimed in claim 23, further comprising a network driver layer shim positioned between the network driver layer and the transport driver interface layer and arranged to intercept files packaged by the transport driver interface layer and encrypt the files using a session key generated during communications with a lower layer of the server.

26. Computer software for installation on a client computer of a multi-tier virtual private network, said network including a server and a plurality of client computers, the server and client computers each including means for transmitting data to and receiving data from an open network,

wherein said computer software includes:

applications level encryption and authentication software arranged to communicate with the server in order to: a.) mutually authenticate the server and the client computer initiating communications with the server and b.) generate a session key for use by the client computer initiating communications to encrypt files; and

at least one lower level set of communications drivers,

wherein said lower level set of communications drivers includes a network driver layer, a transport driver interface layer

BNSDOCID: <WO__9911019A1_I_>

arranged to package applications files as packets capable of being routed over the open network and supply the packets to the network driver layer for transmission to the open network, and a network driver layer shim positioned between the transport driver interface layer and the network driver layer and arranged to intercept files packaged by the transport driver interface layer and encrypt the files using a session key generated during communications with a lower layer of the server.

27. Computer software for installation on a client computer of a multi-tier virtual private network, said network including a server and a plurality of client computers, the server and client computers each including means for transmitting data to and receiving data from an open network,

wherein said computer software includes:

applications level encryption and authentication software arranged to communicate with the server in order to: a.) mutually authenticate the server and the client computer initiating communications with the server and b.) generate a session key for use by the client computer initiating communications to encrypt files; and

further comprising means for securing peer-to-peer communications between applications on two of said client

BNSDOCID: <WO___9911019A1_I_>

computers, said peer-to-peer communications securing means comprising:

means for intercepting a destination address during initialization of communications by a first of said two client computers;

means for causing said authentication software to communicate with the server to carry out functions a.) and b.);

means for transmitting said destination address to said server;

means for causing said server to carry-out functions a.) and b.) with respect to the second of said two client computers;

means for enabling said second of said two client computers to recreate the session key;

means for causing said authentication software to encrypt files to be sent to the destination address using the session key;

means for transmitting the encrypted files directly to the destination address.

28. Computer software as claimed in claim 27, wherein said means for intercepting the destination address comprises a shim positioned between the peer-to-peer applications program and a layer of a communications driver architecture of said first of the two client computers.

BNSDOCID: <WO___9911019A1_I_>

29. Computer software as claimed in claim 27, wherein said shim is positioned above a socket, the socket being positioned above a transport driver layer of said communications driver architecture.

30. Computer software as claimed in claim 27, wherein said shim is positioned above a transport driver layer of said communications driver architecture.

31. A method of carrying out communications over a multi-tier virtual private network, said network including a server and a plurality of client computers, the server and client computers each including means for transmitting data to and receiving data from an open network, comprising the steps of:

intercepting function calls and requests for service sent by an applications program in one of said client computers to a lower level set of communications drivers;

causing an applications level authentication and encryption program said one of said client computers to communicate with the server, generate said session key, and encrypt files sent by the applications program before transmittal over said open network.

32. A method as claimed in claim 31, further comprising the step of intercepting files packaged by a transport driver interface layer to form packets and encrypting the

BNSDOCID: <WO__9911019A1_I_>

packets using a session key generated during communications with a lower layer of the server.

33. A method as claimed in claim 31, further comprising the step of intercepting a destination address during initialization of communications between said one of said client computers and a second of said client computers on said virtual private network;

causing said applications level authentication and encryption program to communicate with the server to carry out functions a.) and b.);

transmitting said destination address to said server;

causing said server to carry-out functions a.) and b.) with respect to the second of said two client computers;

enabling said second of said two client computers to recreate the session key;

causing said authentication software to encrypt files to be sent to the destination address using the session key; and

transmitting the encrypted files directly to the destination address.

34. A method as claimed in claim 33, wherein said step of intercepting the destination address is carried out by a shim positioned between a peer-to-peer applications program

BNSDOCID: <WO___9911019A1_I_>

and a layer of a communications driver architecture of said one of the two client computers.

Meta 1002
p. 435

BNSDOCID: <WO___9911019A1_I_>

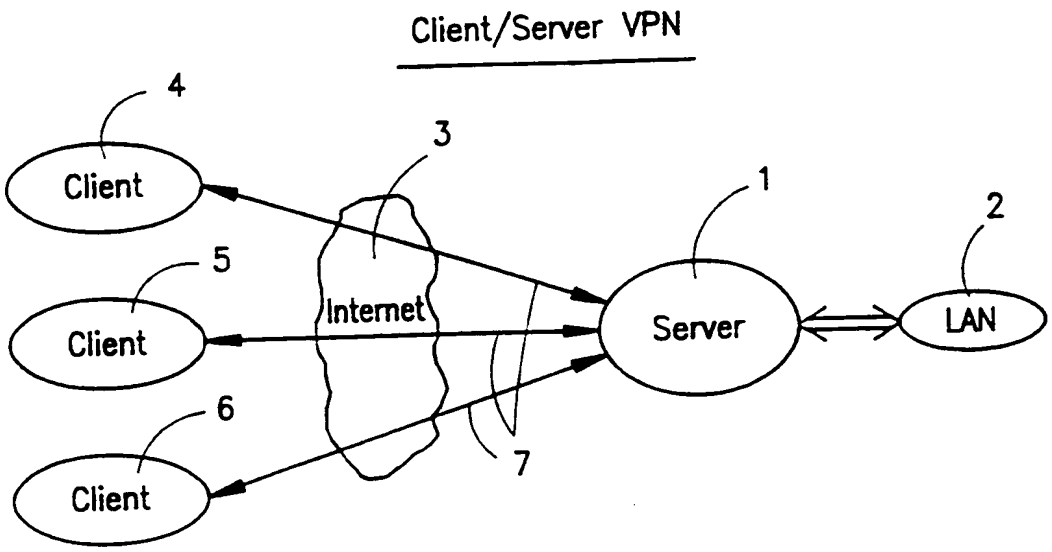


FIG. 1A
(PRIOR ART)

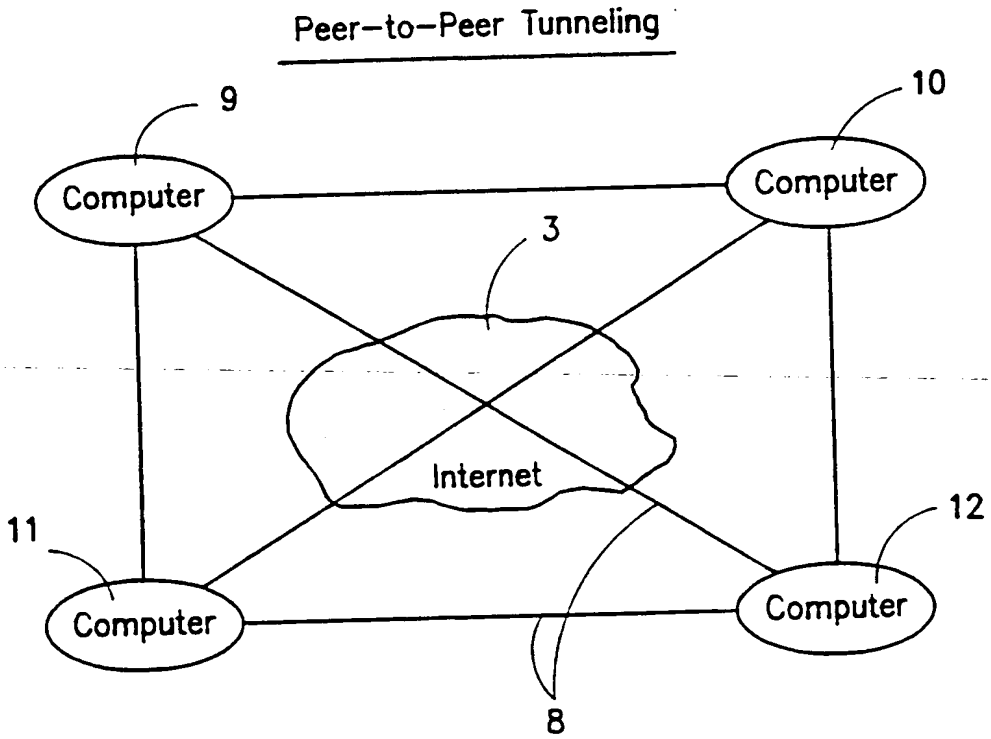


FIG. 1B
(PRIOR ART)

BNSDOCID: <WO___9911019A1_I_>

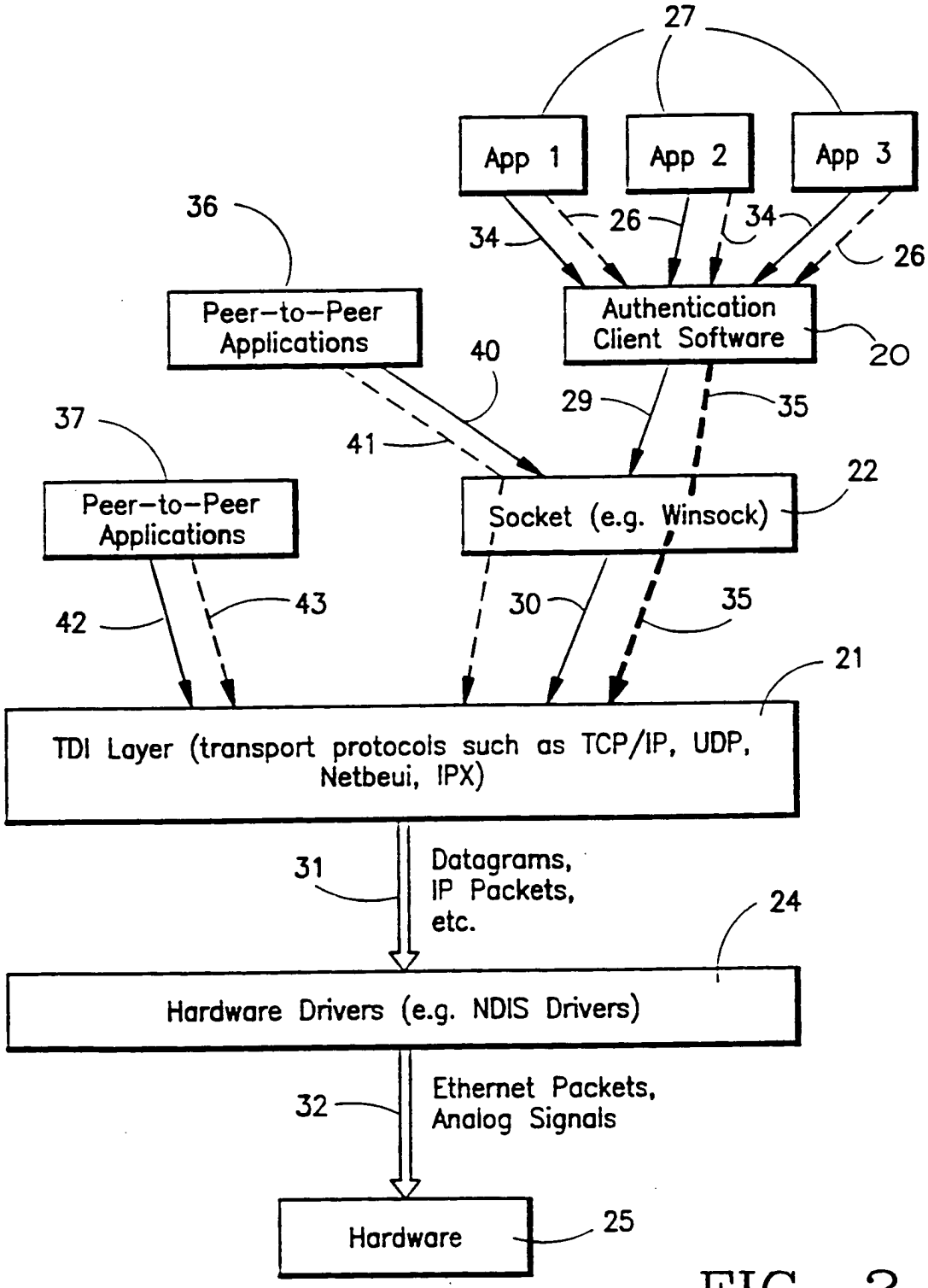


FIG. 2
(PRIOR ART)

BNSDOCID: <WO___9911019A1_I_>

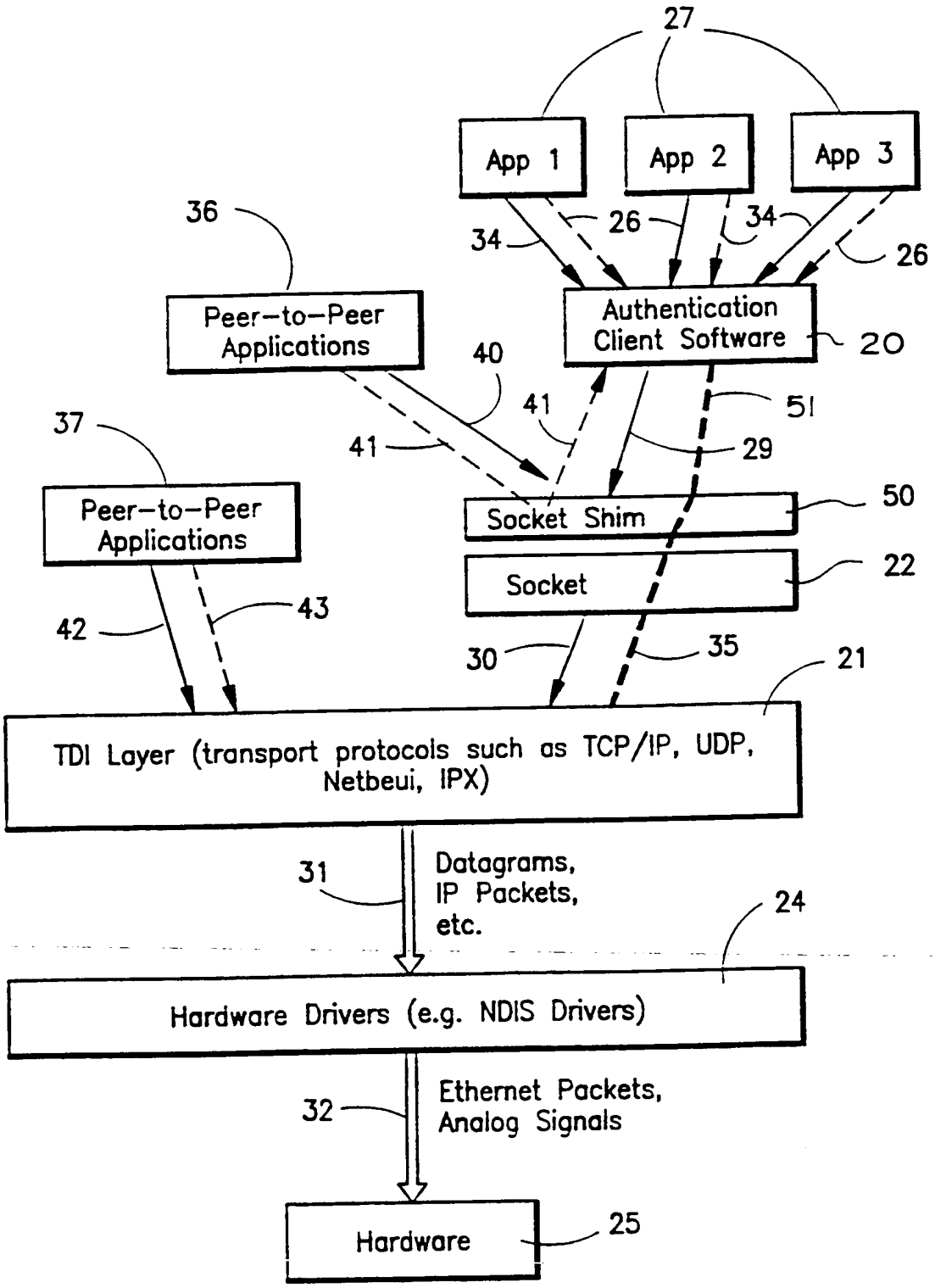


FIG. 3

BNSDOCID: <WO___9911019A1_I_>

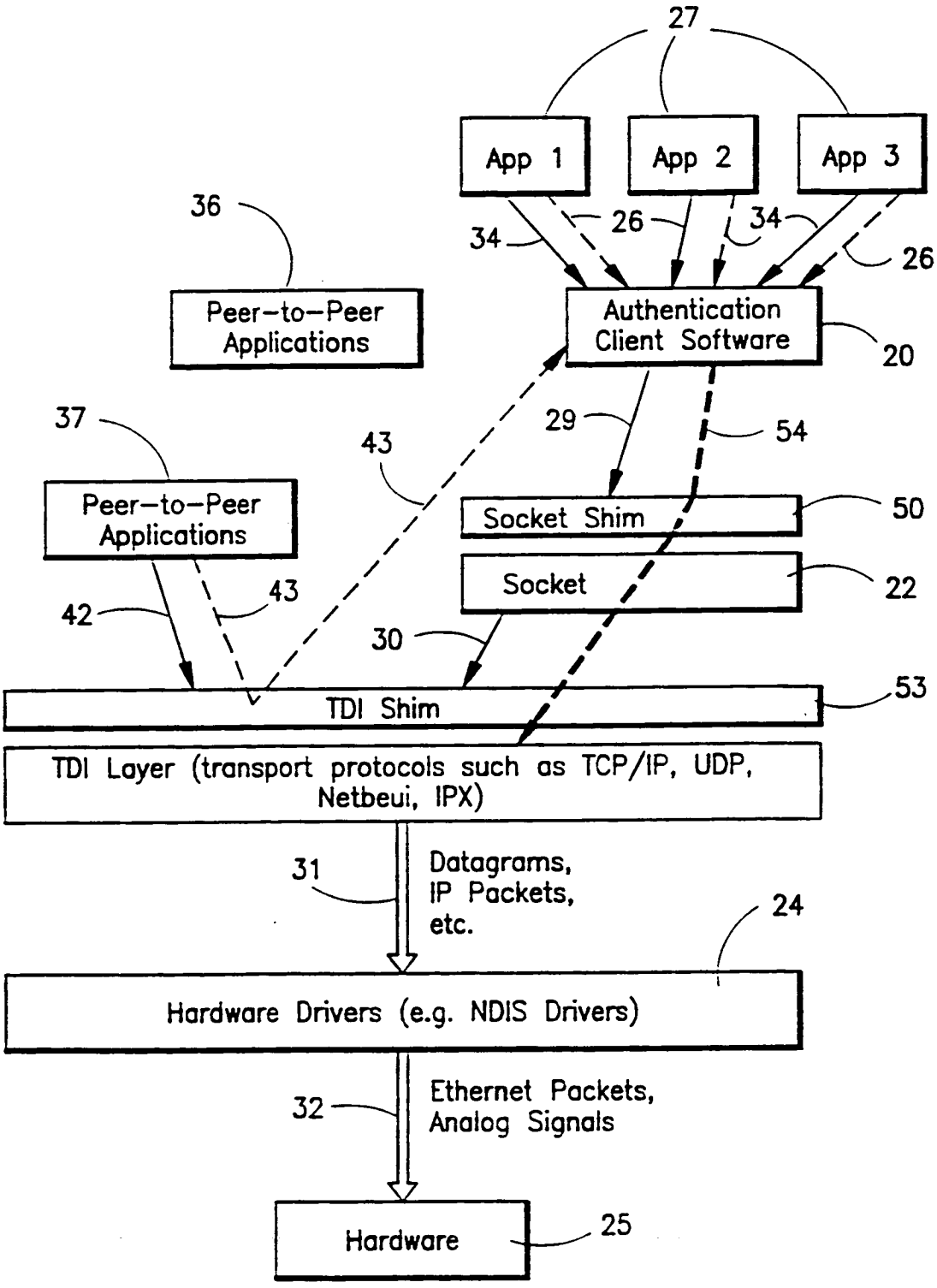


FIG. 4

BNSDOCID: <WO___9911019A1_I_>

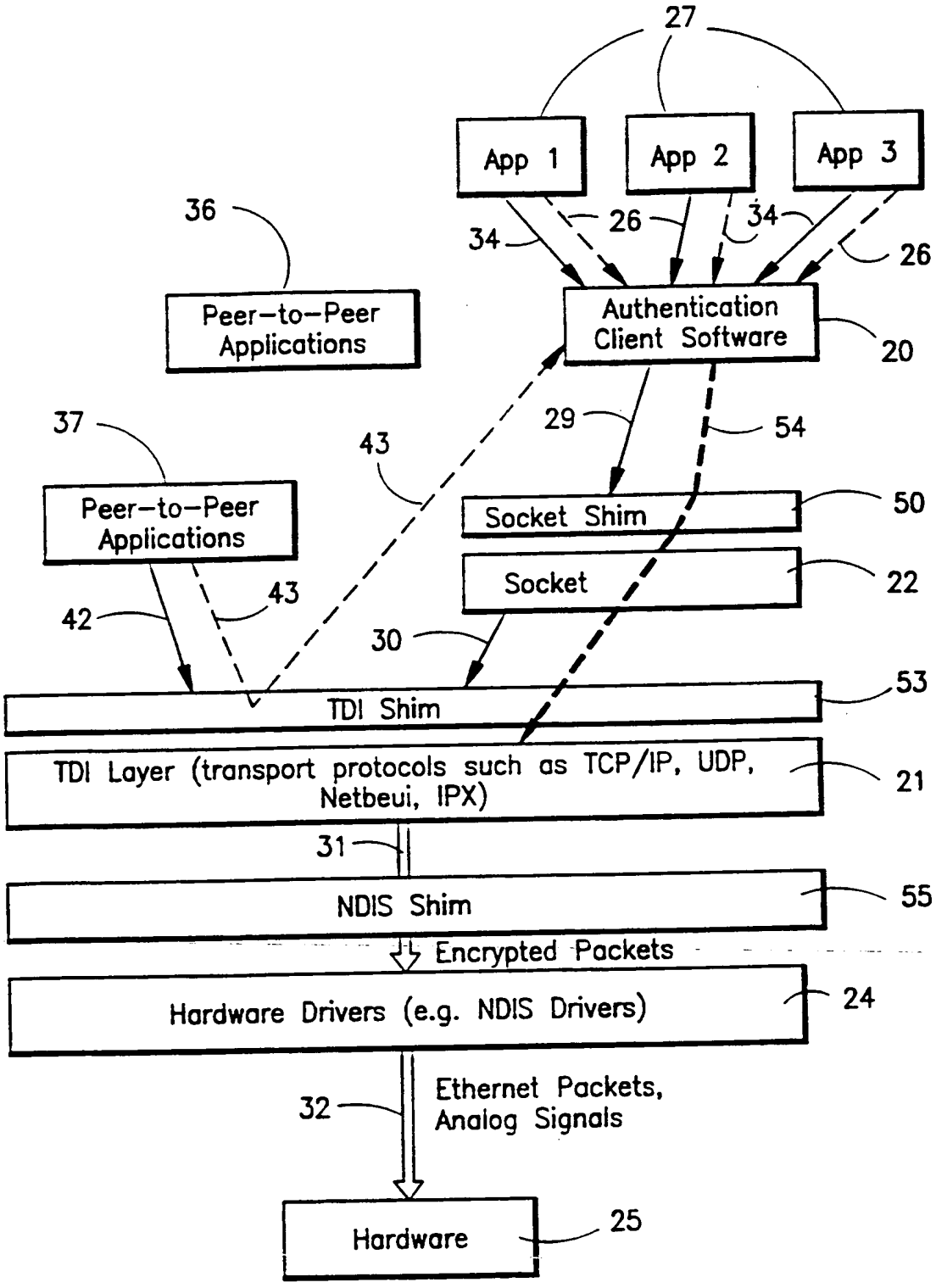


FIG. 5

BNSDOCID: <WO___9911019A1_I_>

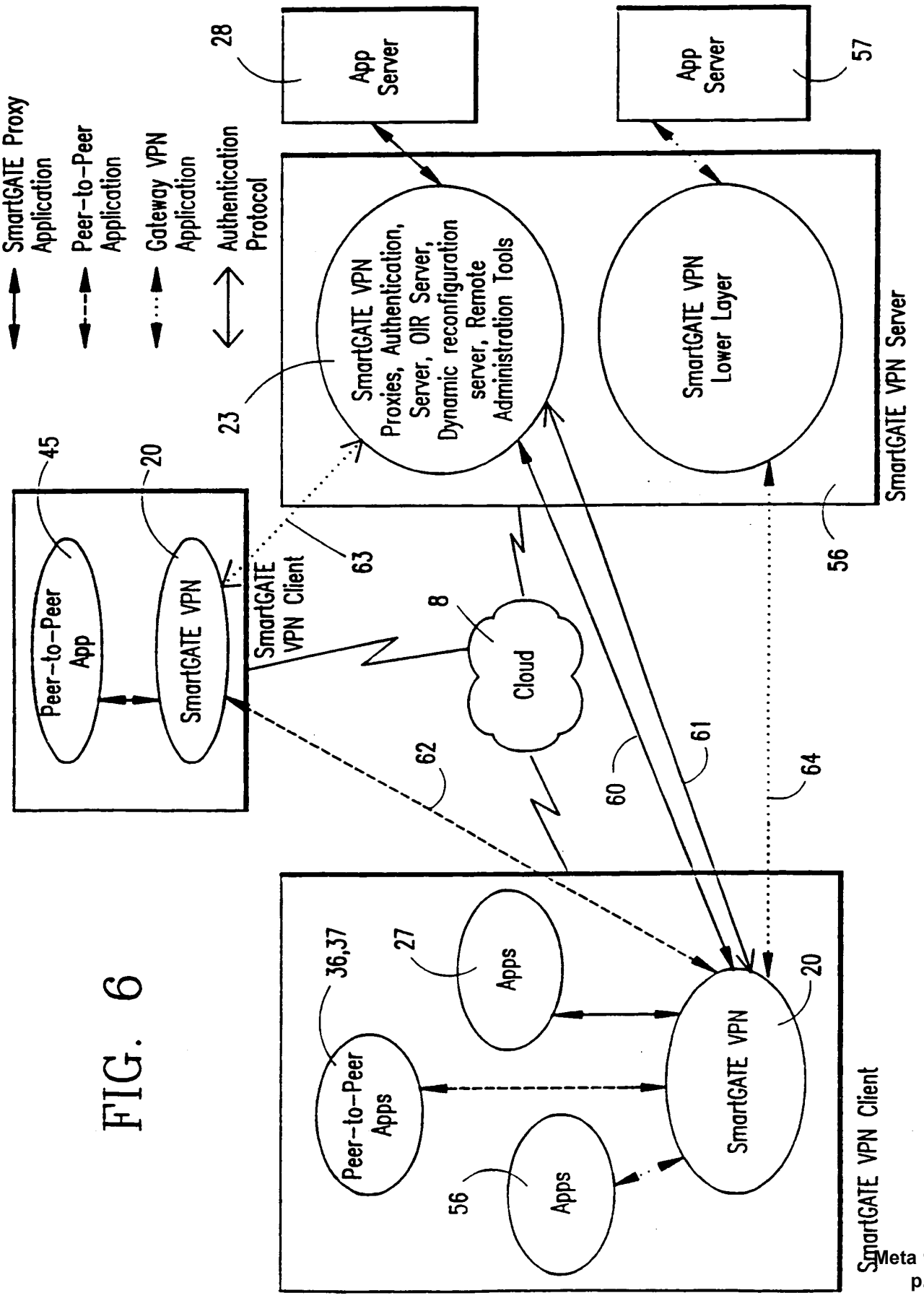


FIG. 6

BNSDOCID: <WO___9911019A1_I_>

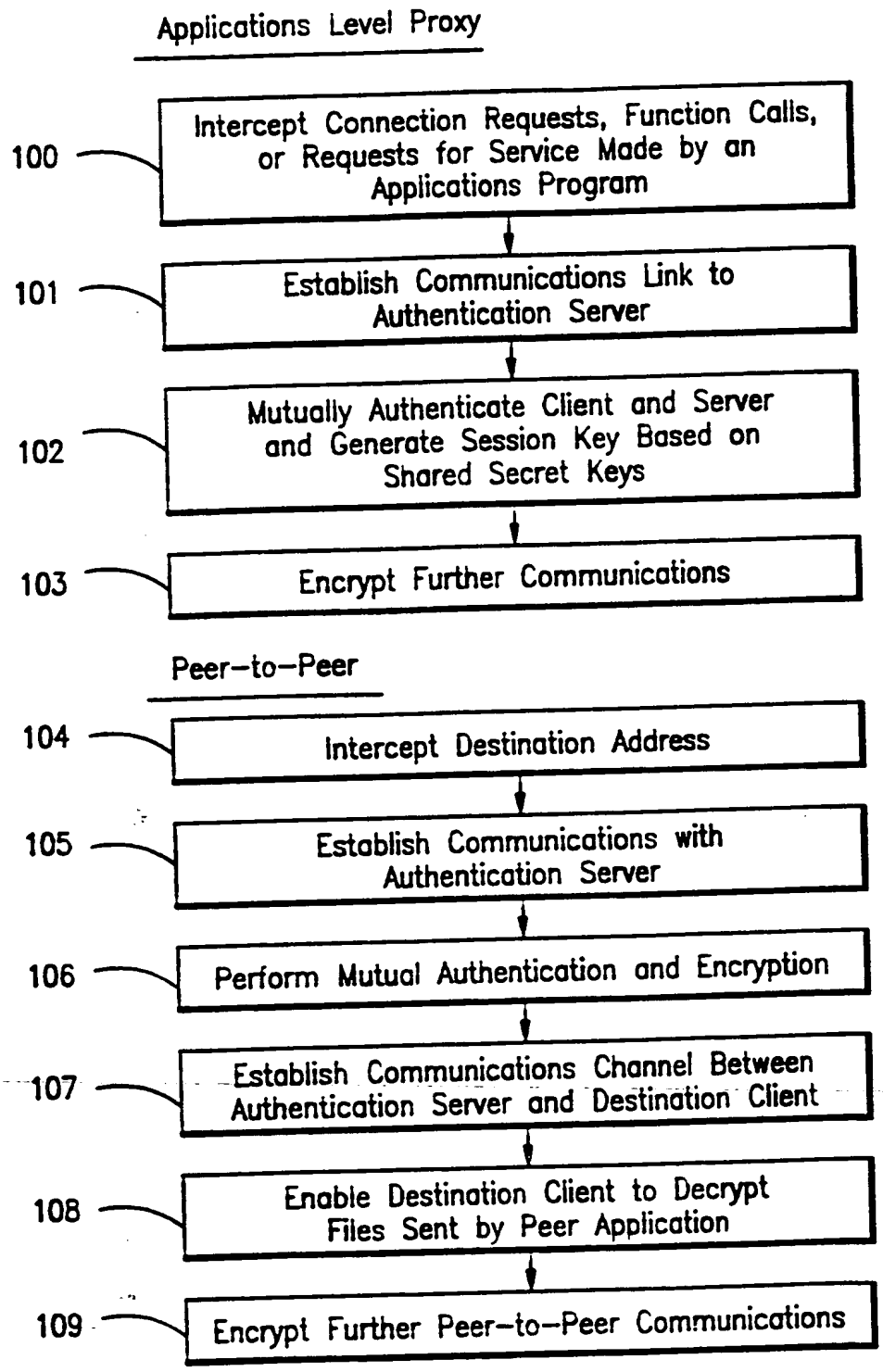


FIG. 7

BNSDOCID: <WO___9911019A1_I_>

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/17198

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :H04L 9/00
US CL :395/187.01

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 395/187.01, 186, 188.01, 200.17, 200.12; 380/49, 21, 25, 4

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS, STN, IEEE ProQuest
search terms: virtual private network, shims, DLLs, protocol layers, Winsock, sockets, encryption, authentication.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,657,390 A (ELGAMAL ET AL) 12 AUGUST 1997, FIGURES 1-8, COL. 3, LINES 20-55, COL. 5, LINE 15 TO COL. 8, LINE 32, COL. 11, LINE 1 TO COL. 16, LINE 49.	1,5,6,16,17,18,19,23,31
A	US 5,602,918 A (CHEN ET AL) 11 FEBRUARY 1997, SEE ENTIRE PATENT.	1-34
A	US 5,550,984 A (GELB) 27 AUGUST 1996, ABSTRACT, COL. 3, LINE 52 TO COL. 4, LINE 45, COL.6, LINES 27-55.	1-34
Y	HURWICZ, A VIRTUAL PRIVATE AFFAIR, BYTE MAGAZINE, JULY 1997, PAGES 79-87.	1,5,6,16,17,18,19,23,31

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 22 OCTOBER 1998	Date of mailing of the international search report 12 NOV 1998
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer JOSEPH PALYS Telephone No. (703) 305-9600

Form PCT/ISA/210 (second sheet)(July 1992)*

Meta 1002
p. 443

BNSDOCID: <WO___9911019A1_I_>

00P1063

B2

ELSEVIER

Computer Communications 21 (1998) 1107–1123

computer communications

Tutorial

Internet/Intranet firewall security—policy, architecture and transaction services

Ray Hunt*

Department of Computer Science, University of Canterbury, Private Bag 4800, Christchurch, New Zealand

Received 22 September 1997; received in revised form 1 April 1998; accepted 3 April 1998

Abstract

The development of Internet/Intranet security is of paramount importance to organisations that plan to gain the economic benefits from interconnection with the Internet. This paper commences by examining firewall policy, focusing on both network service access policy and firewall design policy. Various firewall architectures, ranging from simple packet filters through to screened subnets and proxy gateways, are then discussed. Finally, the various mechanisms by which transactions can be secured over the Internet/Intranet are covered. These include encrypted tunnelling, IPv6, point-to-point tunnelling protocol, secure sockets layer, secure electronic transactions and secure multipart Internet mail encoding. © 1998 Elsevier Science B.V.

Keywords: Firewall design policy; Network service access policy; Packet filter/screening router; Dual-homed gateway; Screened host/subnet; Proxy gateway; Encrypted tunnelling; IPv6; PPTP; SSL; SET; S/MIME

1. Firewall policy

A firewall is a method of achieving security between trusted and untrusted networks and the choice, configuration and operation of a firewall is defined by the policy. The policy defines the services and type of access permitted between trusted and untrusted domains. Therefore, a firewall can be viewed as both a policy and the implementation of that policy in terms of network configuration, host systems, routers, encryption tunnels, authentication procedures and applications systems.

The definition of a firewall policy requires a clear explication of the security perimeter, since different firewall architectures provide different levels of guarantee against attack. Another important term is "zone of risk" that generally applies to TCP/IP capable networks, although networks using other protocols such as Netware/IPX, DECnet and SNA can also be vulnerable.

In principle, the zone of risk covers all networks and servers connected to the Internet, including the Internet backbone and related network infrastructures. The objective of the firewall policy is to minimise the organisation's zone of risk by removing the possibility of attack from an external network. In other words the firewall becomes the zone of risk for the trusted network.

It is widely accepted that there is a real risk from insider threats, and it is often stated that there are more insider attacks (for which a firewall is of little value) than external attacks [1]. Insiders usually have more direct access to the systems and the opportunity to abuse privileges. For example, many workstations can be easily reconfigured to grant privileged access and it is then a simple task to run a protocol analyser or decode software. Also, most standard TCP/IP applications, such as Telnet, FTP, rlogin, etc., have weak authentication control and passwords are transmitted in cleartext.

There are two levels of policy that influence the design, installation and use of a firewall:

- network service access policy (NSAP)
- firewall design policy (FDP).

1.1. NSAP

The NSAP defines which services are to be explicitly allowed or denied between trusted and untrusted networks, together with the way in which these services are to be used as well as any conditions for exception to this policy.

* Tel.: +64 3 3642347; fax: +64 3 3642569; e-mail: ray@cosc.canterbury.ac.nz

0140-3664/98/$19.00 © 1998 Elsevier Science B.V. All rights reserved
PII S0140-3664(98)00173-X

Meta 1002
p. 444

BNSDOCID: <XP___4146571A__I_>

The NSAP should be an extension to existing business policy that will have already addressed the following issues:

- information value—what value does management places on information?
- responsibility—who is responsible for ensuring the protection of the organisation's information from untrusted networks?
- commitment—what is the organisation's commitment to protecting its information resources?
- domains—what domains should or should not be protected?

Further, business policy should already have implemented controls on such systems as

- virus scanning[1]
- physical security access
- floppy disk controls
- RAID back-up systems.

At the highest level the organisational policy might state:

- information is the strategic resource for the organisation;
- the availability, integrity, authenticity and confidentiality of the information will be protected by every cost-effective measure possible;
- ensuring the availability, integrity, authenticity and confidentiality of the information is a priority for all users at all levels of the company.

Below this level, specific policies are implemented which cover issues such as

- access to services (dial-in, dial-out)
- version controls
- user authentication
- trusted/untrusted network access.

It is at this level that the firewall's NSAP is formulated. The NSAP must be drafted before the firewall is implemented. It must provide a balance between protecting the trusted network from known risks while providing users with convenient access to the untrusted network. Further, if a firewall denies access to certain services on an untrusted network, it is essential that the NSAP ensures that these controls are not circumvented or disabled. A typical NSAP might

- allow no access to applications or services on the trusted network from the Internet;
- as above, but allow access to a subset of applications or services by way of a secure server (e.g. bastion host);
- allow access from the Internet to selected applications on the trusted network (e.g. e-mail) in conjunction with strict authentication procedures (e.g. challenge/response and one-time password controls).

[1] Contrary to popular belief, firewalls can scan for viruses. This may require scanning at the application layer (mail or file headers). Most common antivirus products such as McAfee, F-Prot, Dr Solomon, Symantecs and Norton's AntiVirus can be configured to achieve firewall virus control.

1.2. FDP

FDP defines how the firewall implements restricted access and service filtering specified by the NSAP and addresses issues such as

- IP address filtering
- encryption tunnelling,
- secure socket control to facilitate application access
- audit and accounting control.

This policy is specific to the firewall and defines the rules and procedures necessary to implement the NSAP, but it must take account of the capabilities and limitations of the particular firewall platform as well as the threats and vulnerabilities associated with TCP/IP. For example, if the NSAP forbids access to all applications on the trusted network, then implementing a firewall by way of a packet filtering router is extremely risky.

In principle a firewall can

- permit any service unless it is specifically disallowed
- deny any service unless it is specifically permitted.

However, in practice, only the latter option is used. The first option might unintentionally allow denied services to run on non-standard TCP/UDP ports. Further, some services such as FTP, RPC and X-Windows are difficult to filter [2].

Depending upon the various security and flexibility requirements, some firewalls are more appropriate than others, which means that the NSAP must be carefully designed before the firewall is implemented. For example, dual-homed gateways (Section 2.2.1) and screened subnets (Section 2.2.3) can both be used to implement a "deny all" firewall. However, the dual-homed gateway is cheaper but also less flexible than the screened subnet.

In order to arrive at a successful design policy, together with a platform that implements this policy, it is usual to start by restricting all access from the untrusted to the trusted network, and then to specify the following [3].

- What Internet services will the organisation use (e.g. e-mail, Telnet, FTP, World Wide Web (WWW))?
- Where will these services be used from (intra-company, between branches, on a mobile or dial-in basis, by subsidiary organisations, etc.)?
- What additional security features will be needed (e.g. one-time password control, authentication procedures, encryption tunnels, secure sockets, point-to-point encryption, dial-in/dial-back procedures, etc.)?
- What risks result from the provision of these services? E.g. is a 40-bit RSA [4] encryption key adequate for certain government or banking applications? Is dial-in access without formal authentication procedures an acceptable risk?
- What is the cost (e.g. financial, inconvenience) of providing these services? For example, how is key distribution handled? What is the cost of managing a dedicated authentication server?

Meta 1002
p. 445

BNSDOCID: <XP___4146571A__I_>

- What is the balance between usability and security (e.g. if a particular service is too expensive or risky to use should its use be forbidden, thus creating great inconvenience)?

Some services that are inherently insecure may, with the addition of certain technologies, be secured to pose little or no risk. For example, a remote Telnet session can be very vulnerable to packet sniffing for passwords, and would pose a high risk when connecting a machine to a trusted network over an untrusted network such as the Internet. However, with the addition of encryption or strong authentication techniques this risk can be dramatically reduced.

Implementation of the firewall based upon these considerations requires careful use of risk analysis so that the calculated level of risk can be compared with that deemed to be acceptable according to overall company policy [5]. This may result in a change to the initial policy. For example, if the original NSAP denied all dial-in access, certain exceptions to this rule may need to be considered so as to achieve some overall organisational objective.

1.3. Sample policies

1.3.1. Remote access policy

As a specific example a "remote user advanced authentication policy" might address dial-in user access from the Internet as well as authorised users on travel or working from home. All such connections should use the advanced authentication service of the firewall to access systems at the site. Policy should reflect that remote users might not access systems through unauthorised modems placed behind the firewall, as it takes only one captured password or one uncontrolled modem line to enable a backdoor around the firewall.

Authorised users may also wish to have a dial-out capability to access those systems that cannot be reached through the Internet. These users need to recognise the vulnerabilities they may be creating if they are careless with modem access. A dial-out capability may easily become a dial-in capability if proper precautions are not taken.

Therefore, both dial-in and dial-out capabilities should be incorporated into the design of the firewall. Forcing outside users to go through the advanced authentication of the firewall should be strongly reflected in policy. Policy might also prohibit the use of unauthorised modems attached to host systems if the modem capability is offered through the firewall.

Since users could run point-to-point protocol (PPP) to create new network connections into a site protected by a firewall, it needs to be considered as part of the overall access policy. Such connections are potentially a backdoor around the firewall, and may be an even greater danger than a simple dial-in connection.

1.3.2. Information server policy

A site providing public access to an information server may wish to incorporate appropriate access controls into the firewall design and policy should reflect the idea that the security of the site will not be compromised in the provisioning of an information service. For example, a Web server that is intended to provide access for Internet users may not need to be behind the firewall at all, as the information provided by this server resides on that machine rather than being drawn from systems on the internal network. As long as the machine is regularly backed up it can operate unencumbered by a firewall and simply be restored if attacked.

It is useful to make a distinction between two fundamentally different types of traffic:

- information-server traffic (traffic concerned with retrieving information from an organisation's information server);
- business traffic, such as e-mail, file transfer, transaction services, etc.

The two types of traffic have their own risks and do not necessarily need to be mixed with each other. Screened subnet firewalls (Section 2.2.3) allow information servers to be located on a subnet and, therefore, to be isolated from other site systems. This reduces the chance that an information server could be compromised and then used to attack site systems.

1.4. Policy evolution

Two considerations drive the formation of an FDP with respect to Internet connections:

- the risk to the organisation's internal information and systems from external threats, e.g. denial of service attacks, IP spoofing, etc.;
- the risk of sensitive organisational information being disclosed as it is transmitted across the Internet, e.g. password file capture, information leakage attacks (e.g. finger), etc.

Once the FDP has been drafted, maintenance and review are important ongoing activities.

1.4.1. Maintenance of the FDP

Unlike many organisational policies, the FDP is not static and may need to change on a day-by-day basis depending upon new vulnerabilities which arise. For example JAVA was considered to be a great invention and the industry was assured by SUN that it was not a security risk. Therefore, as browsers evolved to become JAVA aware, JAVA code (applets) passed through firewalls. It is likely that JAVA never appeared in any company's firewall policy as it was probably considered to be part of the WWW. One large multinational decreed from the highest level that JAVA be disabled on all browsers, thus demonstrating the dynamic nature of an FDP. Other examples of policy maintenance

Meta 1002
p. 446

BNSDOCID: <XP__4146571A__I_>

include changes to a network's filtering rules as well as rule changes resulting from the introduction of new services.

1.4.2. Review of the FDP

It is most important that the FDP remains under constant review to ensure that the policy reflects the current state of play. As a result of FDP maintenance, the original policy can become unrepresentative of reality and this can introduce security holes. Examples include: change of the systems expert; wrong versions of software being loaded following a system crash. In many of these cases problems may not be detected until after a security breach has occurred.

1.5. Installing and operating a firewall

Once the decision is made to use firewall technology to implement an organisation's security policy, it is then necessary to install a cost-effective firewall that provides an appropriate level of protection. In general, a firewall should provide the following levels of protection:

- support and not impose a security policy;
- support a "deny all services except those specifically permitted" design policy (even if this policy is not initially implemented);
- accommodate new facilities and services should an organisation's security policy change;
- contain advanced authentication measures, such as encryption, challenge/response systems, and should contain the hooks for installing these facilities;
- employ filtering techniques to permit or deny services to specified hosts as needed;
- use flexible and user-friendly IP filtering and be able to filter on as many attributes as possible, including source and destination IP address, source and destination TCP/UDP port, protocol type, and inbound/outbound interfaces;
- use proxy services for applications such as FTP and Telnet, so that advanced authentication measures can be utilised at the firewall.

It will also assist if the firewall supports proxies for services such as NTP, NNTP, X-Windows, Finger, HTTP, and certain web browser software (Section 2.2.4). The firewall should also have the ability to centralise simple mail transfer protocol (SMTP) access, thus reducing direct SMTP connections between site and remote systems. This results in centralised handling of site e-mail.

The firewall should have the ability to concentrate and filter dial-in access as well as logging suspicious activity. If the firewall requires an operating system such as UNIX or Windows NT, then a secured version of the operating system should be part of the firewall, with other security tools as necessary to ensure firewall host integrity. The operating system should have all patches installed and be developed in a manner that its strength and correctness is verifiable. It should be simple in design so that it can be understood and maintained.

Some organisations have the capability to put together their own firewalls, using available software components and equipment or by writing a firewall from scratch. Trusted Information Systems (TIS) Internet Firewall Toolkit [6] is a good example of a company that offers "firewall construction kits". At the same time, there are many vendors[2] [7] offering a wide range of services in firewall technology which include

- provision of the necessary hardware and software
- development of security policy and carrying out risk assessments
- security reviews and security training.

Consideration of the following questions may help an organisation decide whether or not it has the resources to install and operate a successful firewall.

- How will the firewall be tested?
- Who will verify that the firewall performs as expected?
- Who will perform general maintenance of the firewall, such as backups and repairs?
- Who will install updates to the firewall, such as for new proxy servers, new patches, and other enhancements?
- Can security-related patches and problems be corrected in a timely manner?
- Who will perform user support and training?

As a general rule it is desirable that sites:

- standardise operating system versions and software to make installation of patches and security fixes more manageable;
- institute a program for efficient, site-wide installation of patches and new software;
- use services to assist in centralising system administration, if this will result in better administration and better security;
- perform periodic scans and checks of host systems to detect common vulnerabilities and errors in configuration.

2. Firewall architecture

There are many different interpretations of the term firewall and this can be a source of confusion. One basis for defining a firewall is the OSI 7-layer model (Fig. 1) which provides a clearer picture than does the TCP/IP model.

[2] Examples include Digital's Alta Vista Firewall, Secure Computing's Firewall for NT, Eagle NMS (Raptor Systems), ANS Interlock Service 3.06 (ANS CO + RE Systems), Borderware Firewall Server, Firewall-1 v2.0 (Checkpoint Software), Black Hole 3.0 (Milkyway Networks Corp.), Sidewinder, Gauntlet (Data General), GFX Internet Firewall (Global Technology Associates).

BNSDOCID: <XP___4146571A__I_>

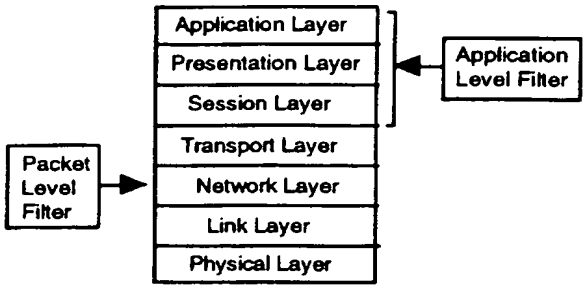


Fig. 1. Firewall architecture in relation to the OSI model.

It can be seen that firewall architecture consists of two levels:

- Packet-level firewalls that operate at the network (IP) and transport (TCP) layers. These are commonly referred to as screening routers or packet filters and block transmission of certain classes of traffic;
- application-level firewalls which operate at the session, presentation and application layers. They are usually implemented using dedicated hosts running specialised software and can also be referred to as bastion hosts or proxy servers, usually running under UNIX or Windows NT. They can also provide relay services to compensate for the effects of the filter(s).

Another important term often used in conjunction with a firewall is "gateway", and Internet firewalls are often referred to as secure Internet gateways. However, there is a more specific use of this term, as can be seen in Fig. 2. The network occupied by the gateway is often referred to as the demilitarised zone (DMZ) [8].

The gateway in the DMZ may consist of both an internal and external machine, as shown in Fig. 3. Normally these two gateways will have more open communication through the inside packet filter than the outside gateway has to other internal hosts. The outside packet filter can be used to protect the gateway from attack, while the inside gateway can be used to guard against the consequences of a compromised gateway.

2.1. Packet filters/screening routers

As packets pass through the router their filtering is based upon a set of rules established by the NSAP. Filtering based upon one of more of the following criteria are commonly applied:

- source IP address
- destination IP address
- TCP/UDP source port
- TCP/UDP destination port.

Not all packet filters can filter on TCP/IP port numbers. Some can examine which of the network interfaces a packet arrived at and then use this as further filtering criterion.

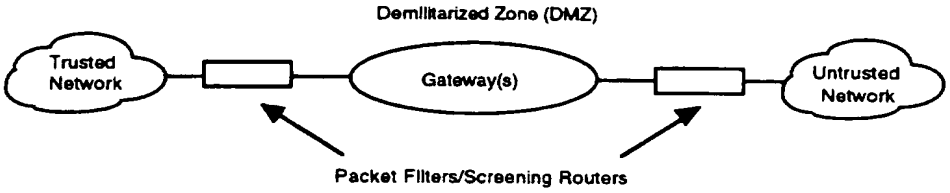


Fig. 2. Firewall design (filter/router and gateway).

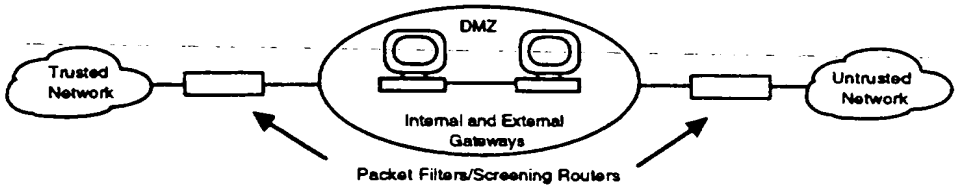


Fig. 3. Firewall design (filter/router with internal and external gateways).

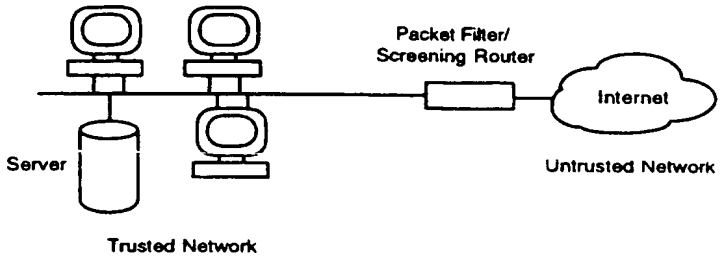


Fig. 4. Firewall using a packet filter/screening router.

BNSDOCID: <XP___4146571A_I_>

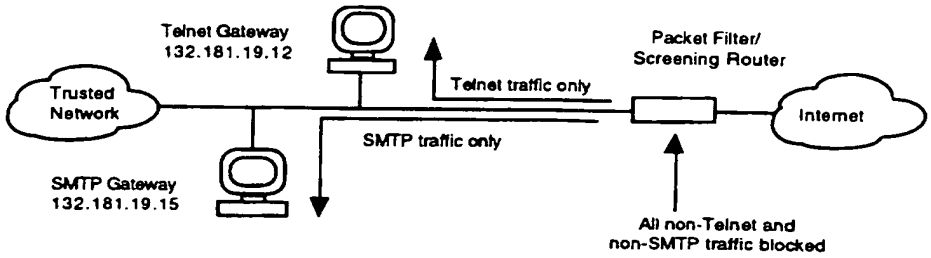


Fig. 5. Packet filtering on Telnet and SMTP.

Filtering can be used to block connections to or from specific hosts or networks (Fig. 4), as well as to block connections to specific ports. A site might wish to block connections from certain addresses, such as from hosts or sites that it considers being untrusted. Alternatively, a site may wish to block connections from all addresses external to the site (with certain exceptions, such as SMTP for receiving e-mail).

Adding TCP or UDP port filtering to IP address filtering results in a great deal of flexibility. Servers such as the Telnet daemon usually reside at specific ports (e.g. port 23). If a packet filter can block TCP or UDP connections to or from specific ports, then it is possible to implement policies that call for certain types of connection to be made to specific hosts, but not others. For example, a site may wish to block all incoming connections to all hosts except for several firewalls-related systems. At those systems, the site may wish to allow only specific services, such as SMTP for one system and Telnet or FTP connections to another system. By filtering on TCP or UDP ports, this policy can be implemented in a simple manner using a packet filtering router or a host with packet filtering capability.

Fig. 5 shows an example of packet filtering with a policy that permits only certain connections to a network of address 132.181.*.*:

- Telnet connections will only be allowed to the Telnet application machine (Telnet gateway) with an IP address of 132.181.19.12
- SMTP connections will only be allowed to the SMTP application machine (e-mail gateway) with an IP address of 132.181.19.15.

All other services and packets are to be blocked. This is a very basic example of packet filtering, and actual rules can become very complicated in practice.

2.1.1. Policing protocols

It is the NSAP that determines which protocols and fields are filtered, i.e. which systems should have Internet access and the type of access to permit. The following services are inherently vulnerable to abuse and are usually blocked at the firewall [3]:

- TFTP (port 69), trivial FTP, used for booting diskless workstations, terminal servers and routers, can also be used to read any file on the system if set up incorrectly;
- X-Windows (ports 6000 +) and OpenWindows (port 2000), can leak information from X-Window displays, including all keystrokes (intruders can even gain control of a server through the X-server);
- RPC (port 111), remote procedure call services including NIS and NFS, which can be used to steal system information such as passwords and read and write to files;
- rlogin, rsh, and rexec (ports 513, 514, and 512) services which, if improperly configured, can permit unauthorised access to accounts and commands.

Other services, whether inherently dangerous or not, are usually filtered and possibly restricted to only those systems that need them. These could include:

- Telnet (port 23), often restricted to certain systems;
- FTP (ports 20 and 21), like Telnet (port 23), often restricted to certain systems;
- SMTP (port 25), often restricted to a central e-mail server;
- RIP (port 520), routing information protocol, can be spoofed to redirect packet routing;
- DNS (port 53) domain names service, contains names of hosts and information about hosts that could be helpful to attackers and it can be spoofed;
- UUCP (port 540) UNIX-to-UNIX CoPy, if improperly configured can be used for unauthorised access;
- NNTP (port 119) network news transfer protocol, used for reading network news;
- NTP (port 123) network time protocol, used for setting system clocks;
- gopher (port 70) and HTTP (port 80) information servers and client programs for gopher and WWW clients, should be restricted to an application gateway that contains proxy services.

At most sites, not all systems require access to all services. Although some of these services, such as Telnet or FTP, are inherently risky, blocking access to these services completely may be too drastic a policy. For example, restricting Telnet or FTP access from the Internet to only those systems that require this type of access can improve security at no cost to user convenience. Services such as NTP and NNTP may seem to pose little threat, but restricting these services to only those systems that need them helps to create a cleaner network environment and helps

Metal 4002
p. 449

BNSDOCID: <XP___4146571A__I_>

the likelihood of exploitation from yet-to-be-discovered vulnerabilities and threats.

Forged or spoofed packets can cause a real threat, as indicated above in relation to RIP and DNS. IP spoofing can be used to make a host look as though it is a trusted host by changing the IP number for example. Also, if source routing is used, the IP header contains the route that the packet will take to reach its destination and thus override routing table instructions. However, attackers who are attempting to circumvent security can generate source-routed packets with Telnet clients to ensure packets follow specific paths. Worse still, reply packets are intended to use the inverse of the original route and this can now permit a two-way connection which was supposed to be blocked.

2.1.2. Non-IP protocols

Most policing protocols outlined in Section 2.1.1 focus on IP. However, other protocols at the same level as IP—such as IPX, NetBeui and AppleTalk—have different packet header formats and filtering characteristics. These non-IPs are rarely used between organisations across the Internet. Some packet filters offer limited filtering support for non-IPs, but they are usually considerably less flexible than their IP equivalent. Many packet filters are configured to just drop non-IP packets. Most non-IPs can be handled by encapsulating them within IP packets. Packets filters then usually either permit or deny encapsulated protocols in their entirety [9].

Another reason to disable non-IP packets is because their escape from LAN domains can reveal important information, such as the LAN's primary architecture. Many of the older protocols, such as RIPX, NetBios and Digital's LAVC (VAX cluster keep alive packets), broadcast the presence of devices that support these protocols across large parts of the network and misconfigured routers may not limit this dangerous traffic to a single LAN segment.

2.1.3. Stateful inspection

Stateful inspection, sometimes also known as dynamic packet filtering, is a recent development by some firewall manufacturers and represents yet another approach towards firewalling. Stateful inspection intercepts packets at the network layer and examines these packets based on their communication state. This mandates the storing of state information from one or more packets as well as buffering and reassembling the datagram before an access decision can be made.

The advantage of this approach lies in its diversity and ability to support a variety of protocols and services. By not relying on the native stack of the firewall host for processing, as well as its ability to look back at past information related to the session, stateful inspection can apply any rule based on the communication content itself. Stateful inspection is usually implemented with support of the entire TCP/IP suite.

2.2. Application-level firewalls

Unfortunately, packet filtering routers have limitations and are frequently difficult to configure and update. Packet filtering rules are complex to specify and usually no testing facility exists for verifying the correctness of the rules (other than by exhaustive testing by hand). Some routers do not provide any audit capability, so that if a router's rules still let through dangerous packets, this may remain undetected until a break-in has occurred.

Exceptions to rules will often need to be made to allow certain types of access that normally would be blocked. However, exceptions to packet filtering rules can make the filtering rules so complex as to be unmanageable. For example, it is relatively straightforward to specify a rule to block all inbound connections to port 23 (the Telnet service). If exceptions are made and certain systems need to accept Telnet connections directly, then a rule for each system may need to be added (some packet filtering systems allow the sequential order of the filter rules to be significant). Sometimes the addition of certain rules can complicate the entire filtering scheme and open up further holes.

Advantages of application-level firewalls or gateways include:

- information hiding, in which the names of internal systems need not necessarily be made known via the DNS to outside systems. The application firewall may be the only host whose name must be made known to outside systems;
- robust authentication and logging, in which the application traffic can be pre-authenticated before it reaches internal hosts and can be audited more effectively;
- cost-effectiveness, since third-party software or hardware for authentication or auditing needs to be located only at the application gateway;
- less-complex filtering rules, in which the rules at a packet filtering router will be less complex than they would if the router needed to filter application traffic and direct it to a number of specific systems.

Application firewalls or gateways can make some set-up and operational procedures more complex. For example, Telnet requires ports for both inbound and outbound traffic and the Telnet user has to remember to connect to the firewall and not the destination host. Sometimes client software has to be reconfigured to operate through the firewall, e.g. the set-up of Netscape requires proxy addresses to be allocated for common services such as HTTP and FTP.

Application firewalls also act as important filters for other protocols such as SMTP, FTP and X-Windows. For example, some application firewalls provide the capability to deny FTP get and put commands which prevents the uploading of a file to an anonymous FTP server and thus provides a higher degree of assurance than just relying on correct file setting permissions at the anonymous FTP server.

Meta 1002
p. 450

BNSDOCID: <XP___4146571A__I_>

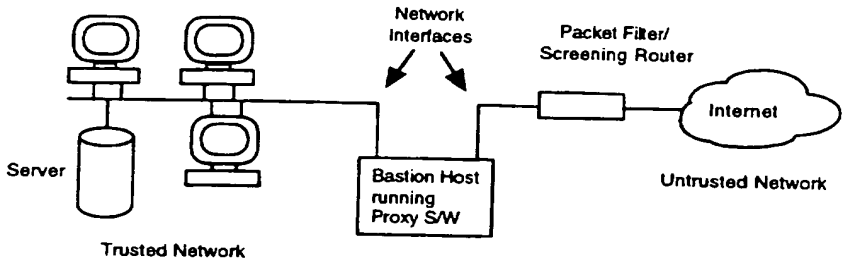


Fig. 6. Dual-homed gateway with two network interfaces.

Some Internet firewalls use a combination of a packet-filter screening computer or a hardware-router for controlling the lower layers of communication, and application gateways for the enabled applications. But even this combination may provide only limited transparency, flexibility and connectivity. It may also be expensive in terms of set-up, management, and expertise.

Another approach is for the firewall to employ an inspection module, applicable to a range of protocols. This can provide an interpretation of encapsulated data destined for higher layers. This has the potential to provide context-sensitive security for complex applications, which may be more effective than technologies which only have data in some of the layers available to them. For example, while application gateways have access only to the application layer and routers have access only to the lower layers, this inspection approach integrates the information gathered from all layers into a single inspection point.

In general, packet filters/screening routers and application gateways can form a powerful combination. Packets filtering via the router controls the lower architectural layers whereas application gateways offer fine-tuned controls appropriate for the applications.

2.2.1. *Dual-homed gateway*

The dual-homed gateway is an application-level firewall implemented without the use of a screening router. It has two network connection ports connected to each of the internal and external networks (Fig. 6). With IP forwarding disabled, a complete block of traffic between the two networks is possible.

There are two ways a user can access the external network from the internal network. The first is by direct logon to the gateway. This is not advisable as it makes this gateway directly vulnerable to password cracking, and can provide access directly to the firewall through software vulnerabilities, such as compiler bugs at the host. The biggest threat to the security of a dual-homed gateway arises when an attacker gains login access. Therefore, login should always be through application proxies on the dual home gateway.

The second and safest way for a connection to be made is through the application layer using a proxy service. A proxy service is an application which routes IP traffic from one port to another. Such an application can provide user authentication, auditing, and logging facilities. These features are a great improvement over packet filters/screening routers, which generally provide no more than rudimentary facilities.

Proxy software has to be written for each service, although basic proxy servers for standard TCP/IP services, such as Telnet, FTP, WWW, etc., are generally available for the Unix and Windows NT environment.

Once an attacker obtains login access to a dual-homed gateway the internal network is subject to intrusions. The zone of risk has been extended from the dual-homed host, to include the entire internal network.

The following is a list of sources from which an attack could be mounted [10]:

- weak permissions on the file system;
- internal network NFS-mounted volumes;
- permissions granted to Berkley r-utilities (e.g. rlogin, rcp (remote copy program), rdist (remote distribution)) through host equivalent files, such as .rhosts (list of available hosts), in users' home directories from user accounts that have been compromised;
- network backup programs that could restore excessive permissions;

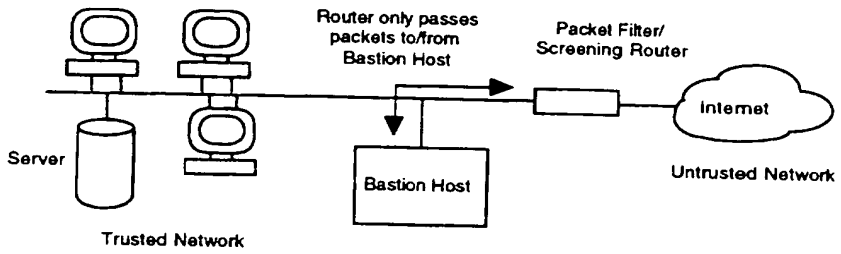


Fig. 7. Screened host gateway.

Meta 1002
p. 451

BNSDOCID: <XP___4146571A__I_>

- use of administrative shell scripts that have not been properly secured;
- learning about the system from older software revision levels and release notes that have not been properly secured;
- installing older operating system kernels that have IP forwarding enabled, or installing an older version of the operating system kernel.

The key for the attacker is to gain enough system privileges to be able to change the Unix kernel variable "ipforwarding", which controls IP forwarding. Once this variable has been enabled the firewall has been completely subverted. Thus, the zone of risk is increased from including only the dual-homed gateway, to including all areas of the network.

Apart from disabling IP forwarding, there are a number of items that can be checked to ensure the security of a dual-homed gateway:

- remove all programming tools, including compilers, linkers, utilities, and services not specifically required for the operation of the dual-homed gateway;
- ensure programs that have SUID (Set User ID) and SGID (Set Group ID) permissions are removed if not required. Check that no excessive permissions on files and programs exist;
- use disk partitions so that denial-of-service attacks designed to fill all available disk space on a partition are confined to that partition only;
- remove unneeded system and special accounts, e.g. disable guest accounts, and maintenance accounts found on some proprietary systems (including screening routers);
- delete network services that are not required.

2.2.2. *Screened host gateway*

The screened host gateway is implemented using a packet filter and a bastion host. It is one of the most popular firewall architectures. The bastion host is usually placed on the internal network, with the packet filter configured such that the bastion host is the only machine reachable from the Internet (Fig. 7). To restrict Internet access further, the packet filter is generally configured to block all traffic not destined to specifically authorised ports on the bastion host. This has the effect of controlling the number of available services.

Major benefits of screened host gateways include reduction of router programming complexity, and improved connectivity for local users. As all traffic is passed through the bastion host, the rules for configuring the router table need only consider the bastion host's IP address. All other packets arriving at the inbound or outbound ports of the packet filter can be discarded, which greatly simplifies packet filter rules.

The zone of risk incorporates only the packet filter and bastion host. The security of this firewall architecture is determined by the accuracy of the packet filter rules in relation to the security policy, and the level of assurance regarding the software running on the bastion host. If an attacker gains entry to the bastion host then the threats to the internal network are similar to those of the dual-homed gateway.

There is, however, a major problem with this architecture which relates to the positioning of the bastion host on the internal network, and relying on the packet filter to control the traffic flow to and from it. If the packet filter is compromised, either through mis-configuration or through an attacker gaining access—such as via a proprietary maintenance account—then the entire internal network is at risk. Compromising the packet filter effectively subverts the bastion host and, once an attacker has control of the packet filter, all traffic can be routed to the external network.

A more secure implementation is to use a packet filter connected to a dual-homed gateway. This architecture ensures that the bastion host is not circumvented in the event of the packet filter being compromised. The attacker has to overcome the dual-homed gateway before the internal network is at risk. Of course, this architecture offers no improvement if an attacker is able to enter through the packet filter and compromises the bastion host directly.

2.2.3. *Screened subnet*

A screened subnet firewall architecture, consists of an isolated network positioned between the external and internal networks. This configuration allows non-critical hosts, such as WWW servers and anonymous FTP sites, to be placed outside the internal network. Bastion hosts are also

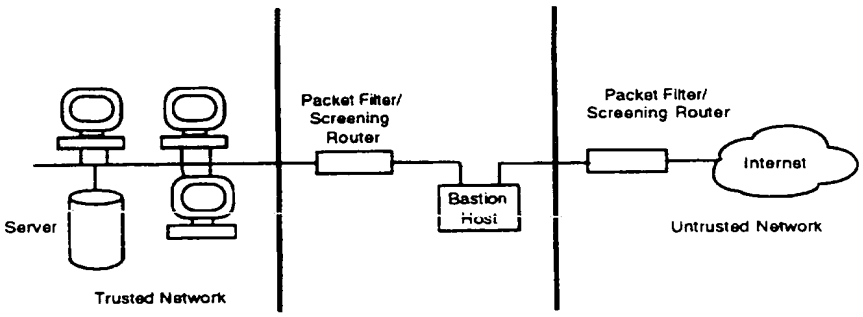


Fig. 8. Screened subnet employing dual-homed gateway.

Meta 1002
p. 452

BNSDOCID: <XP___4146571A__I_>

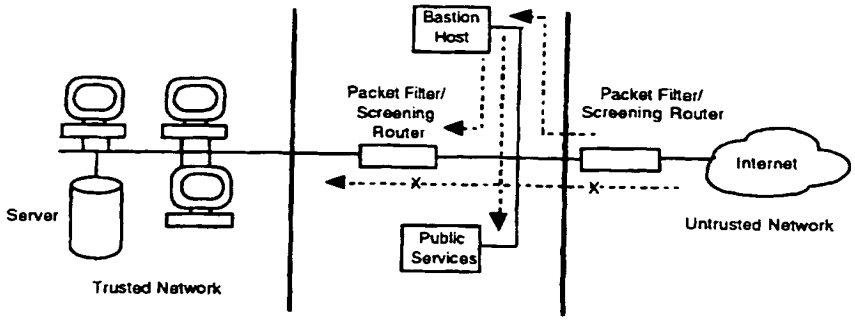


Fig. 9. Screened subnet employing bastion host and public services server.

placed on the subnet to provide interactive terminal sessions, or application-level firewalls. The screened subnet is generally considered to be the most secure firewall architecture. Two possible screened subnet scenarios are shown in Figs. 8 and 9.

A dual-homed gateway or bastion host provides the sole point of access to machines on the internal network, and forces all services through the firewall to be provided by application gateways. Also providing protection are two packet filters, one between the external network and subnet (known as the external router), the other between the subnet and internal network (known as the internal router). Therefore, the zone of risk for this configuration consists of only the two routers, and the dual-homed gateway or bastion host, as well as any other hosts placed on the subnet.

The strength of this firewall architecture comes from the fact that an attacker must subvert the external router, followed by the dual-homed gateway or bastion host, and finally the internal router. If the packet filters are configured so they cannot be managed remotely from the network, then subverting this firewall architecture without setting off alarms and appearing in audit logs would be very difficult.

As with the screened host gateway, if the packet filters can be compromised by logging in and reconfiguring their routing tables, the bastion host can be negated and the internal network put at risk. A drawback with this configuration is the extra level of complexity added to the definition of packet filter rules, especially if there are additional public hosts on the subnet.

2.2.4. Proxy gateway

A proxy server is a specific type of application firewall or gateway designed to mediate traffic between a protected network and the Internet. This application is called a proxy service, and the host running the proxy service is often referred to as an application gateway. Proxies are often used instead of router-based traffic controls to prevent traffic from passing directly between trusted networks and the Internet.

Users on an untrusted network, such as the Internet, can only see the proxy server and are therefore shielded from the trusted network's services and applications. The proxy allows only those services through for which a proxy has been specifically specified. For example if the application gateway specifies proxies for FTP and Telnet, then only those services will be allowed into the trusted network and all other services are disabled.

Many proxies contain additional auditing facilities to support user authentication and protocol specific security, for example an FTP proxy might be configured to permit incoming but block outgoing FTP. For some sites, this degree of security is important, as it guarantees that only those services that are deemed trustworthy are allowed through the firewall. It also prevents other untrusted services from being implemented behind the backs of the firewall administrators.

Proxy servers are application specific and software must be installed for each protocol for which a proxy is required. One popular set of proxy servers is the TIS Internet Firewall Toolkit that includes proxies for protocols such as Telnet, rlogin, FTP, X-Windows, HTTP/Web, and NNTP/Usenet news.

SOCKS is a generic proxy system that can be compiled into a client application to make it operate through a firewall. It consists of a set of near-compatible replacements for various system calls such as: socket, connect, bind, etc. Converting an application simply requires the replacement of the standard calls with the SOCKS versions and existing applications can run unchanged [11]. Its advantage is that it is easy to use, but it is not portable and is therefore unlikely to support additional authentication hooks or protocol-specific logging.

As an example, consider a site that blocks all incoming Telnet and FTP connections using a packet filtering router. The router allows Telnet and FTP packets to go to one host only—the Telnet/FTP proxy gateway. A user who wishes to connect inbound to a site system would have to connect first to the proxy gateway, and then to the destination host in the following sequence (Fig. 10):

- user first Telnets to the proxy gateway and enters the name of an internal host;
- the proxy checks the user's source IP address—probably via some internal host—and accepts or rejects it according to the access policy;
- the user may need to be authenticated (possibly using a challenge/response device);

Meta 1002
p. 453

BNSDOCID: <XP___4146571A__I_>

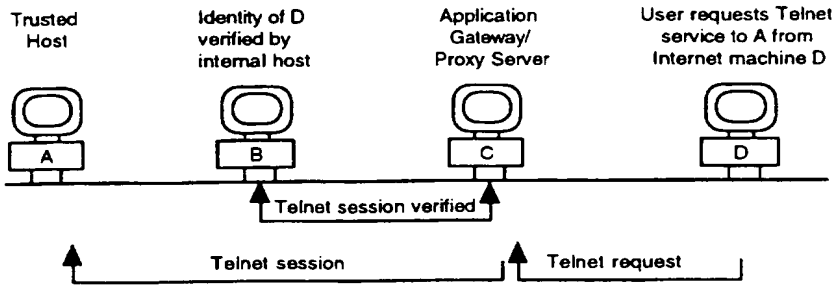


Fig. 10. Proxy used to log and filter Telnet session between Internet and trusted host.

- the proxy service creates a Telnet connection to the internal host;
- the proxy service then passes data between the two connections;
- the proxy gateway audits the connection.

2.2.5. Circuit-level gateway

Circuit-level gateways are similar to application-level gateways, with the distinction that instead of connections being mediated by the firewall, a virtual circuit is created between the external and internal hosts, which results in creating a hole in the firewall. Generally, circuit-level gateways are used to relay TCP connections from internal to external hosts. A certain amount of trust must be placed in the host opening the circuit, as the level of control the firewall has over the connection once it is established is less than that of the application-level gateway. For example, it is possible to piggyback restricted protocols over those already permitted.

3. Secure transactions on the Internet/Intranet

3.1. Security challenges and solutions

There are many challenges in building a full-service Intranet that provides safe communications and collaboration. As the exponential growth of the public Internet demonstrates, TCP/IP solves many problems in a remarkable way. However, TCP/IP was not designed to offer secure communication services. Because TCP/IP was not designed with security in mind, additional technology and policies must be brought to bear to solve, for example, the following typical security problems.

- How can users be authenticated to make sure they are who they claim to be? Standard Web protocols such as TCP/IP and HTTP make impersonating a person or an organisation relatively simple.
- How can authentication be performed without sending usernames and passwords across the network in cleartext?
- How can a single-user login service be provided to avoid costly account maintenance for all the servers (Web, proxy, directory, mail, news, etc.) across the enterprise?
- How is it possible ensure that these services not only work on an Intranet but also scale to the Internet? In other words, how is it possible to avoid managing a separate security scheme inside the firewall and a completely different scheme outside the firewall?
- How can privacy of communications be enforced for both real-time (e.g. data flowing between a Web client and server) and those with store-and-forward applications such as e-mail?
- How is it possible to ensure that messages have not been tampered with between the sender and the recipient?
- How can confidential documents be safeguarded to ensure that only authorised individuals have access to them?

There are a number of technologies available which provide the foundation for a wide variety of security services, including encryption, message integrity verification, authentication, and digital signatures, all of which are based upon cryptography. A variety of authentication and encryption techniques have become available in recent times to provide secure Intranet operation out of basic Internet services.

These techniques include encrypted tunnelling, IP next generation (IPng) and now called IPv6, point-to-point tunnelling protocol (PPTP) and layer 2 forwarding (L2F), secure sockets layer (SSL), secure electronic transactions (SET), secure multipart Internet mail encoding (S/MIME) which can be seen in the protocol stack in Fig. 11. The following sections provide a brief overview of how secure Internet transactions can be achieved.

3.2. Encrypted tunnelling

Tunnelling is a technique for encapsulating data for transparent transmission across an interconnecting network. For example, IPX traffic can be encapsulated in TCP/IP for transport across a network that does not recognise the IPX protocol. Such a link is transparent to the interconnected applications.

Encrypted tunnels take standard IP packets, encrypt and encapsulate them in TCP/IP packets for transport. Source and destination IP-based applications operate as normal, but the data between the two tunnel servers is encrypted

Meta 1002
p. 454

BNSDOCID: <XP___4146571A__I_>

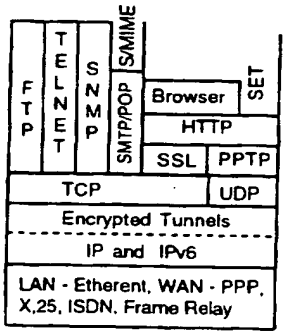


Fig. 11. Protocol architecture for secure Internet transaction services.

Frequently, users initiate connections to the Internet via the firewall, but considerable restrictions usually occur in the reverse direction. When setting up an Intranet using encrypted tunnels, the firewall can be configured to permit encrypted and authenticated data from Internet tunnel clients to pass through into the trusted network. The tunnelled data remains encrypted past the firewall inside the trusted network until it reaches the tunnel server, which may be on the same LAN as the firewall—or in fact anywhere in the trusted network.

Some vendors provide host-to-host or LAN-to-LAN encryption tunnels, whereas others permit the tunnel to terminate on the workstation. For example, Digital's Internet Personal Tunnel [12] makes use of client software running on a Windows 97/NT workstation. In contrast, the tunnel server can handle multiple workstation tunnel clients simultaneously. Authentication keys are stored at the workstation in encrypted form and are accessible by the use of a PIN. Different keys are used for each tunnel set up between the tunnel client and server.

Mobile, home or office-based PC users can connect to an Internet service provider (ISP) and establish a secure tunnel to the trusted network. Authorisation is based upon a PIN or challenge/response system (and not an IP address) followed by an encryption key pair used to authenticate the PC with the trusted tunnel server. This allows a user to roam the network, without being tied to a specific hardware address. Once an encryption tunnel has been set up to the trusted network, the PC can communicate anywhere within this network or be restricted by specific host policies. No special software is required within the trusted network since normal routing is used once the packets leave the tunnel and are decrypted. Neither is any special software required at the ISP, even though all traffic passing through is encrypted as can be seen in Fig. 12.

3.2.1. *Operation of encrypted tunnels*

When a tunnel client (group or personal) wishes to initiate a connection with an Internet Group Tunnel server, a connection request is sent over the network. The request may actually be addressed to a firewall, which relays the connection to the proper Internet Group Tunnel server. The connection request message contains an identification message that is encrypted by the client with the server's public key, and then decrypted by the server with its own private key.

The server's database contains a list of clients that are authorised to build tunnels. If the request is to be granted, the tunnel server sends an encrypted response using the client's public key, which is then decrypted by the client using its own private key. After this authentication sequence, the two parties exchange portions of a session key, which are combined to form a secret session key (Fig. 13). Session keys are periodically changed to enhance security. Each tunnel pair uses a public/private RSA key pair to authenticate each other. Once the tunnel is established, secret RC4 session keys are exchanged for use in encrypting user data. Within the USA a 128-bit key is used, whereas for international operation a 40-bit key is used.

The tunnel client is issued an IP address that is valid for the network of the Group Tunnel server. The tunnel client in effect has two addresses. One is appropriate for the network the client is directly connected to, and the other is appropriate for the network that has been tunnelled into. This will prevent a backdoor option for other workstations on the ISP's LAN from establishing a connection with the Intranet.

The clients routing tables are set up so that all communication intended for hosts on the tunnelled network is sent through the tunnel. All communication intended for other hosts is sent normally through the directly connected network.

Most firewalls can be set up to permit the passing of encrypted tunnel traffic into a trusted network and on to the tunnel server. The usual approach is to relay traffic

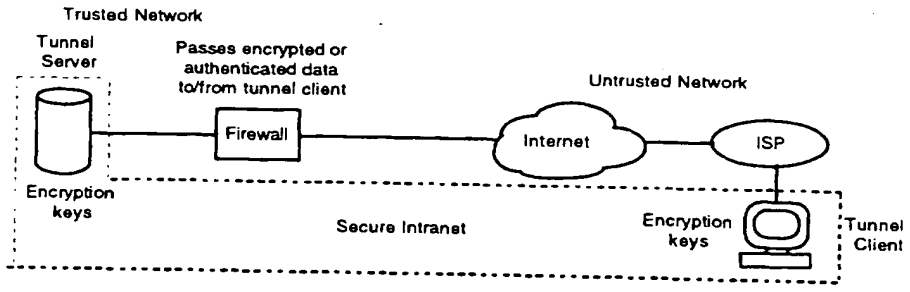


Fig. 12. Intranet architecture with encrypted tunnels.

Meta 1002
p. 455

BNSDOCID: <XP___4146571A__I_>

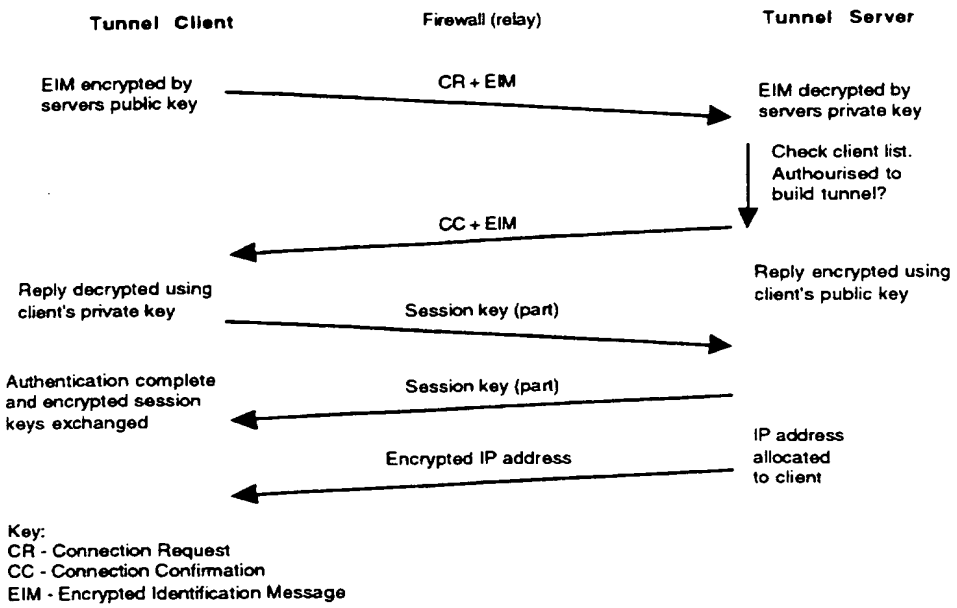


Fig. 13. Authentication sequence for encrypted tunnel establishment.

addressed to a specified port on the firewall on to a specified port on the tunnel server, therefore hiding the actual address of the tunnel server as well as logging all connection attempts at both the firewall and tunnel server.

3.3. PPTP and L2F

Microsoft's PPTP [13] is an extension to the standard point-to-point protocol (PPP) that is used to create multiprotocol virtual private networks (VPNs) via local access to an ISP. Cisco's equivalent is L2F. Use of the Internet or Intranets in conjunction with secure encrypted transactions creates VPNs at a fraction of the cost of conventional network services.

The VPN can be viewed as an application, whereas PPTP or L2F are the protocols in the same way that e-mail is an application and SMTP is the protocol used. As an extension to the PPP, PPTP can only handle point-to-point and not multipoint links.

The result can be global networking with substantial economies of scale. In addition, no changes need be made to existing LAN applications. PPTP/L2F is designed to tackle the twin concerns of cost and security, using the Internet as an inter-LAN technology and opening up the benefits of low-cost and secure communications. PPTP is supported by Microsoft, 3Com, Ascend, Telematics and US Robotics. The competitor to PPTP is Cisco's PIX Private Link product for VPNs and is based upon L2F. Suggestions have been made to combine the best features of L2F (Cisco) and PPTP (Microsoft) into a new protocol called L2TP.

3.4. SSL

SSL provides a method for adding security to existing applications by incorporating RSA data security technology. SSL is application protocol independent and provides

- encryption (which creates a secured channel)
- authentication
- uses X.509 certificates [14] and digital signatures to verify the identity of parties
- message integrity (which ensures that messages cannot be altered en route).

The RSA encryption technology used by both SSL and encryption tunnelling is similar, although the encryption is done at different layers of the architectural stack. SSL encrypts data at the application layer while tunnels establish a link for all logical connections between the two networks. Applications which need to encrypt a specific session, such as a web browser, Telnet or FTP session, must be modified to enable the request for an encrypted link. With tunnelling, the applications do not have to be modified as all traffic between tunnelled networks is encrypted.

Netscape Communications has designed the SSL [15] protocol for providing data security between application protocols (such as HTTP, Telnet, NNTP or FTP) and TCP/IP. SSL is designed to provide privacy between client and server applications on the Internet. It provides data encryption (using a 40-bit key for the RC4 stream encryption algorithm), server authentication, message integrity, and optional client authentication for a TCP/IP connection. Further, SSL requires a reliable transport protocol such TCP for data transmission and reception.

One advantage of the SSL protocol is that it is application independent. Higher-level application protocols, such as HTTP, FTP, Telnet, layer transparently on top of SSL which can negotiate an encryption algorithm and session key as well as authenticate a server before the application

Meta 1002
p. 456

BNSDOCID: <XP___4146571A_I_>

protocol transmits or receives data. All transmitted application protocol data is encrypted, thus ensuring privacy.

SSL provides a security "handshake" that is used to initiate the TCP/IP connection. This handshake results in the client and server agreeing on the level of security they will use, and fulfils any authentication requirements for the connection. Thereafter, SSL's only role is to encrypt and decrypt the application data being used (e.g. HTTP, NNTP, or Telnet). This means that all the information in both the HTTP request and response are fully encrypted, including the requested URL, any submitted form contents (including credit card numbers), any HTTP access authorisation information (usernames and passwords), and all the data returned from the server to the client.

SSL provides three fundamental security services, all of which use public-key encryption:

Service	*Underlying technology*	*Protection against*
Message privacy	Encryption	Eavesdroppers
Message integrity	Message authentication codes (keyed hash functions)	Vandals
Mutual authentication	X.509 certificates	Impostors

SSL therefore provides channel security which has three basic properties:

- the channel is private—encryption is used for all messages after a simple handshake is used to define a secret key;
- the channel is authenticated—the server endpoint of the conversation is always authenticated, whereas the client endpoint is optionally authenticated;
- the channel is reliable—the message transport includes a message integrity check.

In addition the latest version of SSL has a number of protocol enhancements including:

- fewer and faster handshake messages;
- support for more key-exchange and encryption algorithms (in particular, Diffie–Hellman and Fortezza);
- support for hardware tokens in the form of Fortezza cards. This is the first step towards more general support for cryptography-capable smart cards;
- an improvement to the client certificate request protocol that allows a server to specify a list of certificate authorities that it trusts to issue client certificates. This frees the user from having to choose a certificate for each connection.

SSL has been submitted to the W3C* working group on security for consideration as a standard security approach for WWW browsers and servers on the Internet.

Netscape Navigator supports a new URL access method, HTTPS, for connecting to HTTP servers using SSL. HTTPS is simply SSL underneath HTTP (Fig. 11). HTTP + SSL (or HTTPS) and HTTP are different protocols and typically reside on different ports (443 and 80 respectively). The same server system can run both secure and insecure HTTP servers simultaneously. For example the "storefront" and merchandise catalogue could be insecure, while the ordering and payment documents could be secure.

Another variation on the same theme is S/HTTP—a security-enhanced variant of HTTP and also proposed to the W3C security working group. SSL/HTTP and S/HTTP have different motivations. Whereas SSL layers security beneath application protocols like HTTP, Telnet and FTP, S/HTTP adds message-based security to HTTP specifically, drawing on the approaches of privacy enhanced mail (PEM) and multipurpose Internet mail extensions (MIME). SSL and S/HTTP are not mutually exclusive—rather they can coexist in a complementary fashion by layering S/HTTP on top of SSL.

3.5. Fortezza identification and authentication

Fortezza [16] is a PCMCIA type II card and is a key component of the National Security Agency's ongoing program for secure Department of Defence messaging. It is an integrated security solution that meets the needs of civilian and military Government agencies who require the protection of sensitive but unclassified information.

Fortezza combines a user authentication technique, which defies attempts at password sniffing, with the strength of the Digital Signature Algorithm, proposed by the National Institute of Standards and Technology (NIST). It ensures that remote identification is not susceptible to eavesdropping. Fortezza follows the guidelines of Federal Information Processing Standard (FIPS) 185. Encryption and key management are automatic, and never require user intervention.

3.6. IPv6 security, IP security (IPSec)

The Internet has a number of security problems and lacks effective privacy and authentication mechanisms below the application layer. IPv6 or IPSec [17] remedies these shortcomings by providing two security options that may be used singly or together to provide differing levels of security to different users.

The first mechanism, called the IPv6 Authentication Header, is an extension header that provides authentication and integrity (without confidentiality) to IPv6 packets. Although the extension is algorithm independent and will support many different authentication techniques, the use of MD5 is proposed to help ensure interoperability within the worldwide Internet. This can be used to eliminate a

*The World Wide Web Consortium (W3C) is an international collaboration between research organisations, whose goal is to guide the evolution of the World Wide Web. The Consortium works with member companies to refine and revise the technical specifications of the Web as well as to create pre-competitive joint projects that extend the application of the Web into new domains.

Meta 1002
p. 457

BNSDOCID: <XP___4146571A__I_>

significant class of network attacks and, in particular, host masquerading. Its placement at the IP layer can help provide host origin authentication to those upper layer protocols and services that currently lack good protection. This mechanism provides authentication and integrity, but specifically does not provide confidentiality.

The second security extension header provided with IPv6 is the IPv6 Encapsulating Security Header. This mechanism provides integrity and confidentiality to IPv6 packets. It is simpler and more flexible than some similar security protocols and is algorithm independent. To achieve interoperability within the global Internet, the use of DES is being used as the standard algorithm for use with the IPv6 Encapsulating Security Header.

3.7. Secure wide area networking (S/WAN)

S/WAN describes the concept of secure IP wide area networking, the standards for which are based upon IPv6 and IPSec developed by the IETF (Internet Engineering Task Force). S/WAN is also designed to ensure interoperability among firewall and TCP/IP-based products. S/WAN also uses the IETF's security architecture for the Internet protocol, RFC 1825–1829 [18]. It supports encryption at the IP level using RSA (asymmetric) or DES (symmetric) encryption with keys ranging from 40 to 128 bits. This provides more fundamental, lower-level security than higher-level protocols, such as SSL and S/HTTP. However, SSL and S/HTTP can be layered on top of S/WAN implementations so that these security specifications can work together.

S/WAN can be implemented using RSA's BSAFE [19] cryptography engine which provides software developers with a selection of public-key, secure-key and cryptographic hashing algorithms for authentication applications. BSAFE includes modules for popular encryption techniques, such as RSA, Triple DES, RC2, RC4, and RC5, and also supports digital signatures and certificates.

S/WAN's goal is to use a selection of these security features in conjunction with TCP/IP to build Internet-based VPNs. VPNs provide privacy and authentication on public networks by encrypting all data packets between communicants. VPNs can be formed between two or more firewalls across the Internet and, since it is common for organisations to have firewalls at multiple sites, it is a natural evolution for these firewalls to become points for encrypting communications. In the past, users have often been locked into a single-vendor solution, because vendors have been unable to agree upon the implementation details. This S/WAN effort should remove a major obstacle to the widespread deployment of secure VPNs.

Key vendors supporting the S/WAN initiative include RSA, Bay Networks, Cisco, CheckPoint Software Technologies, Digital Pathways, IBM, Raptor Systems, Secure Computing Corporation, Sun Microsystems, Trusted Information Systems, and VeriSign.

3.8. Secure electronic transactions (SETs)

In response to the demand for electronic commerce, MasterCard, Visa, Netscape, and a small group of key technology partners jointly developed the SET protocol as a method to secure credit card transactions over untrusted networks [20]. SET defines both the electronic payment protocol and the certificate management process.

SET will support electronic payment systems that

- provide for confidential transmissions
- authenticate the parties involved
- ensure the integrity of payment instructions
- authenticate the identity of the cardholder and merchant to each other.

The SET specification addresses the following requirements.

- Confidentiality of information—ensures that cardholder information is accessible only by the intended recipient. The SET specification uses message encryption to ensure confidentiality.
- Integrity of data—guarantees that message content is not altered during transmission. SET ensures data integrity with the use of digital signatures.
- Cardholder account authentication—verifies that the cardholder is the legitimate user of the account. SET ensures account authentication by using digital signatures and cardholder certificates.
- Merchant authentication—confirms to the cardholder that a merchant can accept bank card payments. SET uses digital signatures and merchant certificates to provide merchant authentication.
- Interoperability—ensures that any cardholder with compliant software can communicate with any merchant running compliant software. SET provides specific protocols and message formats to ensure interoperability on a variety of hardware and software platforms.

Fig. 14 shows payment system participants and their interactions that are supported by SET.

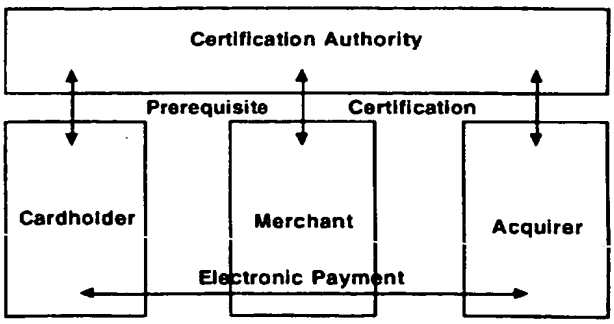


Fig. 14. Payment system participants and their interactions supported by SET.

Meta 1002
p. 458

BNSDOCID: <XP__4146571A_I_>

3.9. S/MIME

S/MIME is a specification for secure electronic mail and was designed to add security to e-mail messages in MIME format [21]. The security services offered are authentication (using digital signatures) and privacy (using RSA public key encryption). More importantly, it was designed to be interoperable, so that any two packages that implement S/MIME can communicate securely.

S/MIME uses a hybrid cryptographic approach to providing security—often referred to as a digital envelope. The bulk message encryption is done with a symmetric cipher, and a public-key algorithm is used for both key exchange and digital signatures. S/MIME also uses X.509-based digital certificates and recommends three symmetric encryption algorithms: DES, Triple-DES, and RC2. The adjustable keysize of the RC2 algorithm makes it especially useful for applications intended for export outside the USA.

S/MIME, PGP (Pretty Good Privacy) and PEM all specify methods for securing electronic mail. All offer privacy and authentication services. Since PGP and PEM are quite different, they need to be compared with S/MIME individually.

PGP can be thought of as both a specification and an application. It relies on users to exchange keys and establish trust in each other. This informal method works well for small workgroups, but can become unmanageable for large numbers of users. S/MIME, on the other hand, utilises hierarchies in which the roles of the user and the certifier are formalised. This means that S/MIME is both more secure and more scalable than PGP implementations. S/MIME can also be well integrated into many e-mail applications, making it simple for users.

PEM was an early standard for securing e-mail that specified a message format and a hierarchy structure. The PEM message format is based on 7-bit text messages, whereas S/MIME is designed to work with MIME binary attachments as well as text. The guidelines for hierarchies are also more flexible in S/MIME. This should allow for both easy set-ups for small workgroups that do not need to be part of an all-encompassing hierarchy, and an easy path to move workgroups to the hierarchy that best suits their needs.

4. Conclusions

This paper has examined a range of issues that need to be addressed in firewall policy, technological choices for hardware and software platforms as well as the protocols currently being designed to secure transactions on untrusted networks.

Any corporate security policy which deals with Internet access and services, has to be flexible because the Internet itself is in a state of flux. Furthermore, an organisation's needs may change as the Internet offers new services and methods for doing business, particularly with the evolution of secure transaction services.

New protocols and services are emerging on the Internet. Using them may be of benefit to the organisation, but they may result in new security concerns. Thus policy needs flexibility to reflect and incorporate new concerns, particularly given the rapid nature of many changes in business today. New partnerships and alliances may bring new network connections and new risks. Few organisations are likely to remain entirely static.

The developments with secure transaction services have been remarkably successful in a short time. In part, this success has resulted from the open nature of protocol development, where companies which would otherwise be in competition work together to develop and fine-tune industry standards.

There is no doubt that the shape of firewall policy, the hardware and software platforms upon which these policies are implemented, as well as the protocols that drive secure transaction services on the Internet/Intranet, will continue to evolve rapidly over the next few years.

References

[1] C. Hare, K. Siyan, Internet Firewalls and Network Security, 2nd edition, New Riders Publishing, 1996, p. 128.
[2] W.R. Cheswick, S.M. Bellovin, Firewalls and Internet Security, 1994, pp. 54–74.
[3] National Computer Security Association (NCSA) Firewall Policy V 1.01, 1996. Internet: www.ncsa.com/store.
[4] Public-Key Cryptography Standards, RSA Data Security, Inc., Version 1.5, 1993.
[5] G.B. White, Computer System and Network Security, CRC Press, 1996, pp. 9–21.
[6] F. Avolio, M. Ranum, A network perimeter with secure external access, Proc. Internet Society Symposium on Network and Distributed System Security, San Diego, 1994 (also available from ftp.tis.com as / pub/firewalls/isoc94.ps.z).
[7] Computer Security Institute, 1996 Firewall Product Matrix, Computer Security Journal XII (1) (1996) 40–45.
[8] W.R. Cheswick, S.M. Bellovin, Firewalls and Internet Security, 1994, pp. 51–52.
[9] D.B. Chapman, E.D. Zwicky, Building Internet Firewalls, O'Reilly and Associates, 1995, pp. 153–154.
[10] C. Hare, K. Siyan, Internet Firewalls and Network Security, 2nd edition, New Riders Publishing, 1996, pp. 334–335.
[11] Internet: anonymous FTP from ftp.nec.com:/pub/security/socks.cstc.
[12] Virtual Private Networking over the Internet, Digital Equipment Corporation, White Paper, October 1995.
[13] Internet: Anonymous FTP from ftp://ftp.microsoft.com/developr/drg/pptp, 1996.
[14] ITU Rec. X.509 (1993) | ISO/IEC 9594-8: 1995, including Draft Amendment 1: Certificate Extensions (Version 3 certificate). Also available from: http://info.itu.ch/itudoc/itu-t/rec/x/x500up/s_x509_30222.html.
[15] Internet: http://home.netscape.com/newsref/ref/128bit.html, 1996.
[16] Internet: www.sctc.com/lockout/HTML/fortezza.html, 1996.
[17] S. Deering, R., Hinden, Internet Protocol, Version 6 (IPv6) Specification, IETF RFC 1883, 1995.

Meta-1002
p. 459

BNSDOCID: <XP___4146571A__I_>

[18] R. Atkinson, P. Metzger, Security Architecture for the Internet Protocol, RFCs 1825-9, 1995.
[19] Internet: www.rsa.com/rsalabs/newfaq/q173.html, 1997.
[20] Internet: Secure Electronic Transaction (SET) Specification, Version 1, 1997. www.mastercard.com/set/setdraft2, www.visa.com/cgi-bin/vee/nt/ecomm/set/main.html and www.setco.org.
[21] Internet: www.rsa.com/rsalabs/newfaq/q131.html, 1997.

Meta 1002
p. 460

BNSDOCID: <XP___4146571A__I_>

THIS PAGE BLANK (USPTO)

PTO/SB/06 (08-03)
Approved for use through 7/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD
Substitute for Form PTO-875

Application or Docket Number: 10/184,564

CLAIMS AS FILED – PART I

	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY RATE	SMALL ENTITY FEE	OR	OTHER THAN SMALL ENTITY RATE	OTHER THAN SMALL ENTITY FEE
BASIC FEE (37 CFR 1.16(a))				$	OR		$
TOTAL CLAIMS (37 CFR 1.16(c))	minus 20 =	*	x $ =		OR	x $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(b))	minus 3 =	*	x $ =		OR	x $ =	
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(d))			+ $ =		OR	+ $ =	
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL		OR	TOTAL	

CLAIMS AS AMENDED – PART II

7/18/02

		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE	SMALL ENTITY ADDITIONAL FEE	OR	OTHER THAN SMALL ENTITY RATE	OTHER THAN SMALL ENTITY ADDITIONAL FEE
AMENDMENT A	Total (37 CFR 1.16(c))	* 28	Minus	** 20	= 8	x $ =		OR	x $ =	144
	Independent (37 CFR 1.16(b))	* 4	Minus	*** 3	= 1	x $ =		OR	x $ =	84
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(d))					+ $ =		OR	+ $ =	
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	228
AMENDMENT B	Total (37 CFR 1.16(c))	*	Minus	**	=	x $ =		OR	x $ =	
	Independent (37 CFR 1.16(b))	*	Minus	***	=	x $ =		OR	x $ =	
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(d))					+ $ =		OR	+ $ =	
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	
AMENDMENT C	Total (37 CFR 1.16(c))	*	Minus	**	=	x $ =		OR	x $ =	
	Independent (37 CFR 1.16(b))	*	Minus	***	=	x $ =		OR	x $ =	
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(d))					+ $ =		OR	+ $ =	
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/181,564	07/18/2002	Martin Euchner	1454.1302	4619

21171 7590 10/05/2005

STAAS & HALSEY LLP
SUITE 700
1201 NEW YORK AVENUE, N.W.
WASHINGTON, DC 20005

EXAMINER	
TRAN, TONGOC	
ART UNIT	**PAPER NUMBER**
2134	

DATE MAILED: 10/05/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/18[illegible],564	CHARPENTIER, FREDERIC JEAN-LUC	
	Examiner	Art Unit	
	Tongoc Tran	2134	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *18 July 2002*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-16* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-16* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *7/18/2002*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

Meta 1002
p. 464

DETAILED ACTION

1. This office action is in response to Applicant's Application Serial No. 10/181564 filed on September 18, 2002. Claims 1-16 are pending.

Information Disclosure Statement

2. The information disclosure statement (IDS) submitted on July 18, 2002 has been considered by the Examiner.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1-26 are rejected under 35 U.S.C. 102(b) as being anticipated by Chen et al. (WO 99/11019, hereinafter Chen).

In respect to claim 1, Chen discloses the method for transmitting data from a transmitter to a receiver, a) in which the transmitter extends the data by authentication data using an application protocol on the application layer; b) in which the authentication data are used by the receiver to ascertain whether it knows the transmitter; c) in which, if the receiver knows the transmitter, the data are accepted, otherwise the data are rejected (see Abstract, page 1, 1st paragraph, page 8, 3rd paragraph – page 9, 1st

paragraph and page 11, 2nd paragraph).

In respect to claim 2, Chen discloses the method as claimed in claim 1, in which the data are transmitted on a packet-oriented basis (see page 9, 2nd paragraph).

In respect to claim 3, Chen discloses the method as claimed in one of the preceding claims, in which the authentication data are determined by virtue of at least part of a protocol data unit, available on the application layer, of the application protocol being linked to a secret by the transmitter (see page 11, 2nd paragraph).

In respect to claim 4, Chen discloses the method as claimed in claim 3, in which the linking to the secret is carried out by virtue of a) cryptographical linking to at least one further parameter, particularly by virtue of an encryption operation and/or b) a cryptographic checksum and key information (see page 11, 2nd paragraph).

In respect to claim 5, Chen discloses the method as claimed in one of the preceding claims, in which verification at the receiver end determines whether or not the receiver knows the transmitter (see page 11, 2nd paragraph).

In respect to claim 6, Chen discloses the method as claimed in claim 5, in which the verification is carried out in the form of cryptographic verification using further parameters, particularly a decryption operation and/or a cryptographic checksum check

(see page 11, 2nd paragraph).

In respect to claim 7, Chen discloses the method as claimed in one of claims 3 to 6, in which the secret between the transmitter and the receiver is a key for symmetric encryption or a key pair for asymmetric encryption (see page 11, 2nd paragraph, DES).

In respect to claim 8, Chen discloses the method as claimed in one of claims 3 to 7, in which the part of the protocol data unit available on the application layer comprises a sequence number and/or a timestamp (see page 11, 2nd paragraph, timestamp is inherently required in packet heading).

In respect to claim 10, Chen discloses the method as claimed in one of the preceding claims, in which, before the data are transmitted from the transmitter to the receiver, authentication is carried out between the transmitter and the receiver (see page 11, 2nd paragraph).

In respect to claims 14-16, the claim limitations are substantially similar to claim 1. Therefore, claims 14-16 are rejected based on the similar rationale.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claim 9, 11-13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chen (WO 99/11019).

In respect to claim 9, Chen discloses the method as claimed in one of claims 3 to 8 in which the encryption is performed using a one-way hash function or a message authentication code with a key which is known only to the receiver and to the transmitter which is to be authenticated. However, Applicant admits that authentication through one way hash or MAC is old and well known (Specification, page 3, 2nd and 3rd paragraph). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to apply one way hash or MAC as authenticate method with the mutual authentication between the transmitter and the receiver taught by Chen to enhance the security of providing accessing control.

In respect to claims 11-13, Chen discloses the method as claimed in one of the preceding claims. Chen does not explicitly disclose the authentication is for use in packet-switched telephone services; used in Internet telephone; used in switching nodes and switching installations. Official Notice is taken that providing authentication through packet switch Internet telephone services is old and well known. It would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the authentication taught by Chen through packet-switch, Internet telephone

services to enhance security protection in the IP telephone service.

Conclusion

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tongoc Tran whose telephone number is (571) 272-3843. The examiner can normally be reached on 8:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Gregory Morse can be reached on (571) 272-3838. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Examiner: Tongoc Tran
Art Unit: 2134

October 3, 2005

GREGORY MORSE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER

10/181564

Rec'd PCT 18 JUL 2002

#3

FORM PTO-1449 U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE	ATTORNEY DOCKET NO. 1454.1302	APPLICATION NO.
LIST OF REFERENCES CITED BY APPLICANT	FIRST NAMED INVENTOR Martin EUCHNER	
(Use several sheets if necessary)	FILING DATE	GROUP ART UNIT

U.S. PATENT DOCUMENTS

*EXAMINER INITIAL		DOCUMENT NO.	DATE	NAME	CLASS	SUB-CLASS	FILING DATE
	AA						
	AB						
	AC						

FOREIGN PATENT DOCUMENTS

		DOCUMENT NO.	DATE	COUNTRY	CLASS	SUB-CLASS	TRANSLATION YES	TRANSLATION NO
M	AD	99/11019	3/4/99	WIPO				
	AE							

OTHER REFERENCES (INCLUDING AUTHOR, TITLE, DATE, PERTINENT PAGES, ETC.)

M	AF	Ray Hunt, "Internet/Intranet Firewall Security-Policy, Architecture and Transaction Services, Computer Communications, vol. 21, No. 13, September 1998, pages 1107-1123
No English Translation	AG	~~Andrew S. Tanenbaum, "Computer-Netzwerke", Wolfram's Fachverlag, Attenkirchen 1992, pages 17-32~~
J	AH	~~Christoph Ruland, "Informationssicherheit in Datennetzen", DATACOM-VERLAG, Bergheim 1993, pages 61-63 and 68-79~~
M	AI	H. Schulzrinne: "RTP: A Transport Protocol for Real-Time Applications; Internet Engineering Task Force", RFC 1889, 1996, pages 1-75
M	AJ	D. Harkins, et al., "The Internet Key Exchange (IKE)" Internet Engineering Task Force, RFC 2409, 1998, pages 1-41
M	AK	S. Kent et al. "Internet Engineering Task Force", RFC 2402, 1998, pages 1-22
M	AL	S. Kent et al. "Internet Task Force", RFC 2406, 1998, pages 1-22

EXAMINER M	DATE CONSIDERED 9/30/05

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

©2001 Staas & Halsey LLP

Notice of References Cited	Application/Control No. 10/18[illegible],564	Applicant(s)/Patent Under Reexamination CHARPENTIER, FREDERIC JE	
	Examiner Tongoc Tran	Art Unit 2134	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-6,061,796	05-2000	Chen et al.	726/15
	B	US-6,158,011	12-2000	Chen et al.	726/15
	C	US-5,602,918	02-1997	Chen et al.	713/153
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Search Notes

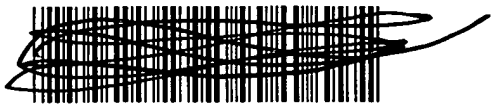
Application/Control No.	Applicant(s)/Patent under Reexamination
10/183,564	CHARPENTIER, FREDERIC JEAN-LUC
Examiner	**Art Unit**
Tongoc Tran	2134

SEARCHED			
Class	Subclass	Date	Examiner
713	152-153, 160-162,168, 178	10/2/2005	TT
726	4,27	10/2/2005	TT

INTERFERENCE SEARCHED			
Class	Subclass	Date	Examiner

SEARCH NOTES (INCLUDING SEARCH STRATEGY)		
	DATE	EXMR
West	10/2/2005	TT
Plus	10/2/2005	TT

Index of Claims



Application/Control No.	Applicant(s)/Patent under Reexamination
10/18[illegible],564	CHARPENTIER, FREDERIC JEAN-LUC
Examiner	**Art Unit**
Tongoc Tran	2134

√	Rejected	–	(Through numeral) Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

Claim		Date								
Final	Original	10/3/05								
	1	√								
	2	√								
	3	√								
	4	√								
	5	√								
	6	√								
	7	√								
	8	√								
	9	√								
	10	√								
	11	√								
	12	√								
	13	√								
	14	√								
	15	√								
	16	√								

Claim		Date								
Final	Original									

Claim		Date								
Final	Original									

UNITED STATES PATENT AND TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON, D.C. 20231
www.uspto.gov

Bib Data Sheet

CONFIRMATION NO. 4619

SERIAL NUMBER	FILING DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
10/181,564	07/18/2002 RULE	713	2131	1454.1302

APPLICANTS

Martin Euchner, Muenchen, GERMANY;

**** CONTINUING DATA ************************

THIS APPLICATION IS A 371 OF PCT/DE01/00021 01/05/2001

**** FOREIGN APPLICATIONS ********************

GERMANY 100 01 855.6 01/18/2000

Foreign Priority claimed	35 USC 119 (a-d) conditions met	Verified and Acknowledged	STATE OR COUNTRY	SHEETS DRAWING	TOTAL CLAIMS	INDEPENDENT CLAIMS
☑ yes ☐ no	☑ yes ☐ no ☐ Met after Allowance	Examiner's Signature / Initials	GERMANY	3	28	4

ADDRESS

21171

TITLE

Method, system for transmitting data from a transmitter to a receiver and transmitter or receiver

FILING FEE RECEIVED	FEES: Authority has been given in Paper No. _____ to charge/credit DEPOSIT ACCOUNT No. _____ for following:
1118	☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other _____ ☐ Credit

WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Sunday, October 02, 2005

Hide?	Set Name	Query	Hit Count
		DB=USPT; PLUR=YES; OP=OR	
☐	L29	L28 and internet near10 telephon$4	14
☐	L28	L26 and authenticat$4	24
☐	L27	L26 and PDU$1	0
☐	L26	20000101	31
☐	L25	application near2 level near2 firewall	37
☐	L24	internet adj telephone adj protocol	12
☐	L23	internet adj telephone adj protocol	12
☐	L22	L21 and internet near2 telephon$4	13
☐	L21	20000101	71
☐	L20	(ftp or http or imap or irc or nntp or pop3 or sip or smtp or snmp or ssh or telnet or bittorrent or websphere or mq)and packet and (prototol adj data adj unit or PDU$1) and (sequen$4 adj number or timestamp) and authenticat$5	96
☐	L19	(ftp or http or imap or irc or nntp or pop3 or sip or smtp or snmp or ssh or telnet or bittorrent or websphere or mq)and packet with (prototol adj data adj unit or PDU$1) with (sequen$4 adj number or timestamp) and authenticat$5	5
☐	L18	(ftp or http or imap or irc or nntp or pop3 or sip or smtp or snmp or ssh or telnet or bittorrent or websphere or mq) with packet with (prototol adj data adj unit or PDU$1) with (sequen$4 adj number or timestamp) and authenticat$5	0
☐	L17	(ftp or http or imap or irc or nntp or pop3 or sip or smtp or snmp or ssh or telnet or bittorrent or websphere or mq) with packet with (prototol adj data adj unit or PDU$1) with (sequen$4 adj number or timestamp) and authenticat$5	0
☐	L16	l15 and transmi$4	48
☐	L15	20000101	51
☐	L14	l7 and packet and (application or session or presentation) adj layer$1 and authenticat$5 Internet adj telephon$4 and timestamp and sequence adj number and (protocol adj data adj unit or PDU$1)	71
☐	L13	l7 and packet and (application or session or presentation) adj layer$1 and authenticat$5 and (key or encrypt$5 or enciph$5 or encod$5) and (check adj sum or hash or mac) and internet and telephon$4 and timestamp and sequence and (protocol adj data adj unit or PDU$1)	2
☐	L12	L7 and (application or session or presentation) adj layer with protocol with authenticat$4	8
☐	L11	20000101	8
☐	L10	L7 and (application or session or presentation) adj layer with authenticat$4	14

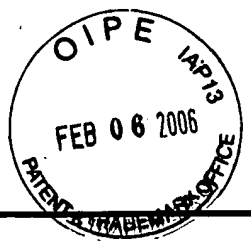
☐	L9	L7 and (application or session or presentation) adj layer$1	110
☐	L8	L7 and (application or session or presentation) with layer$1	215
☐	L7	L6 or l2 or l5	1913
☐	L6	726/4,27.ccls.	327
☐	L5	713/151,152,153,160,161,162,168,178.ccls.	1333
☐	L4	(6731609 6731625).pn.	2
☐	L3	(6061796 6158011).pn.	2
☐	L2	(L1).pn. (5487120 5491463 5528621 5581707 5592556 5668946 5724343 5727006 5768301 5799040 5850526 5949552 5956168 RE36310 5954810 5995512 6011977 6026150 6111910 6125207 6128338 6166496 6173189 6175316 6240088 4322576 4340961 4343042 4346440 4353128 4358825 4368512 4383312 4384307 4417320 4429382 4433378 4437182 4438511 4447908 4450558 4451827 4464658 4466107 4472712 4489416 4499609 4511887 4514824 4519083).pn. (4524440 4536876 4542517 4542380 4558409 4561092 4564838 4590468 4601035 4608559 4622682 4633462 4759041 4761763 4761800 4763357 4787083 4792648 4797879 4797898 4803485 4807223 4809269 4809362 4811279 RE32900 4818987 4821292 4823305 4823312 4829421 4829511 4831624 4839891 4852088 4853962 4856081 4857898 4857915 4864617 4866431 4868558 4868562 4868860 4870410 4872006 4875039 4876538 4878051 4881073).pn. (4897835 4905234 4905273 4905282 4907225 4910791 4914688 4918690 4924308 4933672 4942598 4942591 4945806 4947484 4949175 4951280 4953210 4955018 4956841 4958344 4964138 4969147 4969146 4977593 4977559 4979095 4985895 4987572 5001471 5007045 5007051 5008952 5010330 5018138 5020051 5021890 5058132 5049875 5047927 5047764 5046182 5185833 5189413 5191326 5192947 5197083 5202940 5206909 5208693 5212715).pn. (5212808 5214705 5216519 5223707 5231646 5241160 5241660 5245553 5245606 5245617 5247381 5247702 5251020 5258868 5258995 5259003 5267244 5267263 5272558 5272702 5276684 5278992 5282028 5282205 5283781 5285409 5285449 5285512 5286908 5287353 5287384 5287458 5289582 5296936 5297181 5309153 5311508 5311595 5323426 5325372 5331315 5331318 5331431 5331634 5333198 5335221 5341377 5341426 5342408 5347268).pn. (5347547 5359446 5359594 5363402 5371734 5371736 5377021 5379289 5386435 5387981 5387993 5390038 5390039 5390041 5396228 5398115 5398258 5406403 5410417 5410536 5412498 5416695 5418496 5418624 5420583 5422733 5426513 5428638 5434684 5434859 5434860 5440405 5442348 5444718 5446759 5448554 5450071 5450616 5455570 5455703 5455917 5461621 5469285 5469432 5471469 5471503 5473459 5475381 5475687 5477550).pn.	300
☐	L1	(5553094 5673252 5963146 6172616 5280584 5257257 5394391 5845085 4400778 4491916 4774707 5022046 5541957 5946348 6038252 6078632 6114990 4516248 4536877 4568930 4584679 4612653 4623886 4769839 4796256 5293400 5331450 5488653 5574585 5642350 5649122 5721725 5784408 6034948 6040770 6167515 6249216 4409593 4420842 4453228 4534025 4626995 4794589 4949333 5249220 5280498 5297203 5371783 5461618 5479441)	1147

END OF SEARCH HISTORY

PLUS Search Results for S/N 10181564, Searched September 29, 2005

The Patent Linguistics Utility System (PLUS) is a USPTO automated search system for U.S. Patents from 1971 to the present. PLUS is a query-by-example search system which produces a list of patents that are most closely related linguistically to the application searched. This search was prepared by the staff of the Scientific and Technical Information Center, SIRA.

5553094	5487120	4524440	4897835	5212808	5347547
5673252	5491463	4536876	4905234	5214705	5359446
5963146	5528621	4542517	4905273	5216519	5359594
6172616	5581707	4542380	4905282	5223707	5363402
5280584	5592556	4558409	4907225	5231646	5371734
5257257	5668946	4561092	4910791	5241160	5371736
5394391	5724343	4564838	4914688	5241660	5377021
5845085	5727006	4590468	4918690	5245553	5379289
4400778	5768301	4601035	4924308	5245606	5386435
4491916	5799040	4608559	4933672	5245617	5387981
4774707	5850526	4622682	4942598	5247381	5387993
5022046	5949552	4633462	4942591	5247702	5390038
5541957	5956168	4759041	4945806	5251020	5390039
5946348	RE36310	4761763	4947484	5258868	5390041
6038252	5954810	4761800	4949175	5258995	5396228
6078632	5995512	4763357	4951280	5259003	5398115
6114990	6011977	4787083	4953210	5267244	5398258
4516248	6026150	4792648	4955018	5267263	5406403
4536877	6111910	4797879	4956841	5272558	5410417
4568930	6125207	4797898	4958344	5272702	5410536
4584679	6128338	4803485	4964138	5276684	5412498
4612653	6166496	4807223	4969147	5278992	5416695
4623886	6173189	4809269	4969146	5282028	5418496
4769839	6175316	4809362	4977593	5282205	5418624
4796256	6240088	4811279	4977559	5283781	5420583
5293400	4322576	RE32900	4979095	5285409	5422733
5331450	4340961	4818987	4985895	5285449	5426513
5488653	4343042	4821292	4987572	5285512	5428638
5574585	4346440	4823305	5001471	5286908	5434684
5642350	4353128	4823312	5007045	5287353	5434859
5649122	4358825	4829421	5007051	5287384	5434860
5721725	4368512	4829511	5008952	5287458	5440405
5784408	4383312	4831624	5010330	5289582	5442348
6034948	4384307	4839891	5018138	5296936	5444718
6040770	4417320	4852088	5020051	5297181	5446759
6167515	4429382	4853962	5021890	5309153	5448554
6249216	4433378	4856081	5058132	5311508	5450071
4409593	4437182	4857898	5049875	5311595	5450616
4420842	4438511	4857915	5047927	5323426	5455570
4453228	4447908	4864617	5047764	5325372	5455703
4534025	4450558	4866431	5046182	5331315	5455917
4626995	4451827	4868558	5185833	5331318	5461621
4794589	4464658	4868562	5189413	5331431	5469285
4949333	4466107	4868860	5191326	5331634	5469432
5249220	4472712	4870410	5192947	5333198	5471469
5280498	4489416	4872006	5197083	5335221	5471503
5297203	4499609	4875039	5202940	5341377	5473459
5371783	4511887	4876538	5206909	5341426	5475381
5461618	4514824	4878051	5208693	5342408	5475687
5479441	4519083	4881073	5212715	5347268	5477550



TFW 2134

S&H Form: (02/05)

REPLY/AMENDMENT FEE TRANSMITTAL

Attorney Docket No.	1454.1302
Application Number	10/181,564
Filing Date	July 18, 2002
First Named Inventor	Martin EUCHNER
Group Art Unit	2134
Examiner Name	Tongoc Tran

AMOUNT ENCLOSED	420.00

FEE CALCULATION (fees effective 12/08/04)

CLAIMS AS AMENDED	Claims Remaining After Amendment	Highest Number Previously Paid For		Number Extra	Rate	Calculations
TOTAL CLAIMS	29	-27	=	2	X $ 50.00 =	$ 100.00
INDEPENDENT CLAIMS	5	-4	=	1	X $ 200.00 =	200.00

Since an Official Action set an original due date of January 5, 2006, petition is hereby made for an extension to cover the date this reply is filed for which the requisite fee is enclosed (1 month ($120). **Additional 7 claim fees, including 1 independent claim fee, were paid in the preliminary Amendment filed July 18, 2002**	120
If Notice of Appeal is enclosed, add ($500.00)	
If Statutory Disclaimer under Rule 20(d) is enclosed, add fee ($130.00)	
Information Disclosure Statement (Rule 1.17(p)) ($180.00)	
Total of above Calculations =	$ 420.00
Reduction by 50% for filing by small entity (37 CFR 1.9, 1.27 & 1.28)	
TOTAL FEES DUE =	$ 420.00

(1) If entry (1) is less than entry (2), entry (3) is "0".
(2) If entry (2) is less than 20, change entry (2) to "20".
(4) If entry (4) is less than entry (5), entry (6) is "0".
(5) If entry (5) is less than 3, change entry (5) to "3".

METHOD OF PAYMENT

☒ Check enclosed as payment.

☐ Charge "TOTAL FEES DUE" to the Deposit Account No. below.

☐ No payment is enclosed.

GENERAL AUTHORIZATION

☒ If the above-noted "AMOUNT ENCLOSED" is not correct, the Commissioner is hereby authorized to credit any overpayment or charge any additional fees necessary to:

Deposit Account No. 19-3935
Deposit Account Name STAAS & HALSEY LLP

☒ The Commissioner is also authorized to credit any overpayments or charge any additional fees required under 37 CFR 1.16 (filing fees) or 37 CFR 1.17 (processing fees) during the prosecution of this application, including any related application(s) claiming benefit hereof pursuant to 35 USC § 120 (e.g., continuations/divisionals/CIPs under 37 CFR 1.53(b) and/or continuations/divisionals/CPAs under 37 CFR 1.53(d)) to maintain pendency hereof or of any such related application.

SUBMITTED BY: STAAS & HALSEY LLP

Typed Name	Mehdi D. Sheikerz	Reg. No.	41,307
Signature		Date	February 6, 2006

©2005 Staas & Halsey LLP



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of:

Martin EUCHNER

Serial No. 10/181,564 Group Art Unit: 2134

Confirmation No. 4619

Filed: July 18, 2002 Examiner: Tongoc Tran

For: METHOD, SYSTEM FOR TRANSMITTING DATA FROM A TRANSMITTER TO A RECEIVER, AND TRANSMITTER AND RECEIVER THEREFOR

AMENDMENT

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

This is in response to the Office Action mailed October 5, 2005, and having a period for response set to expire on January 5, 2006. A Petition for a one-month extension of time, together with the requisite fee for same, is submitted herewith, thereby extending the period for response to February 6, 2006 (February 5, 2006 falling on a Sunday).

The following amendments and remarks are respectfully submitted. Reconsideration of the claims is respectfully requested.

If there are any additional fees associated with filing of this Amendment, please charge the same to our Deposit Account No. 19-3935.

02/07/2006 HALI11 00000159 10181564

01 FC:1202 100.00 OP
02 FC:1201 200.00 OP
03 FC:1251 120.00 OP

IN THE SPECIFICATION

Pursuant to 37 CFR 1.125(b), please **REPLACE** the pending specification with the substitute specification, including the Abstract, but excluding the claims, attached hereto. The substitute specification includes no new matter. According to 37 CFR 1.125(c) a marked up copy of the substitute specification is submitted showing all the changes relative to the immediate prior version of the specification of record.

IN THE CLAIMS

The text of all pending claims, (including withdrawn claims) is set forth below. Cancelled and not entered claims are indicated with claim number and status only. The claims as listed below show added text with underlining and deleted text with ~~strikethrough~~. The status of each claim is indicated with one of (original), (currently amended), (cancelled), (withdrawn), (new), (previously presented), or (not entered).

1-16 (CANCELLED)

17. (NEW) A method for transmitting data from a transmitter to a receiver, comprising:

extending the data in the transmitter by authentication data using an application protocol on an application layer;

ascertaining, by the receiver, whether the receiver knows the transmitter based on the authentication data; and

accepting the data at the receiver, if the receiver knows the transmitter, and otherwise rejecting the data,

wherein the application protocol comprises a Real Time Transport Protocol (RTP) and the authentication data is inserted at end of an RTP payload.

18. (NEW) The method as claimed in claim 17, wherein the data are transmitted on a packet-oriented basis.

19. (NEW) The method as claimed in claim 17, wherein the authentication data are determined based on at least part of a protocol data unit available on the application layer, and the application protocol being linked to a secret by the transmitter.

20. (NEW) The method as claimed in claim 19, wherein the linking to the secret is carried out by at least one of cryptographical linking to at least one further parameter, particularly by virtue of an encryption operation and a cryptographic checksum and key information.

21. (NEW) The method as claimed in claim 20, wherein said ascertaining includes

verification at the receiver to determine whether the receiver knows the transmitter.

22. (NEW) The method as claimed in claim 21, wherein the verification is cryptographic verification using parameters and including at least one of a decryption operation and a cryptographic checksum check.

23. (NEW) The method as claimed in claim 22, wherein the secret between the transmitter and the receiver is one of a key for symmetric encryption and a key pair for asymmetric encryption.

24. (NEW) The method as claimed in claim 23, wherein the part of the protocol data unit available on the application layer includes at least one of a sequence number and a timestamp.

25. (NEW) The method as claimed in claim 24, wherein the encryption operation is performed using one of a one-way hash function and a message authentication code with a key known only to the receiver and the transmitter to be authenticated.

26. (NEW) The method as claimed in claim 25, wherein the authentication is carried out between the transmitter and the receiver before the data are transmitted from the transmitter to the receiver.

27. (NEW) The method as claimed in claim 26, wherein the receiver and transmitter provide packet-switched telephone services.

28. (NEW) The method as claimed in claim 27, wherein the receiver and transmitter are each disposed in at least one of a switching node and a switching installation.

29. (NEW) The method as claimed in claim 26, wherein the receiver and transmitter are used in Internet telephony.

30. (NEW) The method as claimed in claim 29, wherein the receiver and transmitter are each disposed in at least one of a switching node and a switching installation.

31. (NEW) The method as claimed in claim 21, wherein the secret between the transmitter and the receiver is one of a key for symmetric encryption and a key pair for asymmetric encryption.

32. (NEW) The method as claimed in claim 21, wherein the part of the protocol data unit available on the application layer includes at least one of a sequence number and a timestamp.

33. (NEW) The method as claimed in claim 21, wherein the encryption operation is performed using one of a one-way hash function and a message authentication code with a key known only to the receiver and the transmitter to be authenticated.

34. (NEW) The method as claimed in claim 20, wherein the secret between the transmitter and the receiver is one of a key for symmetric encryption and a key pair for asymmetric encryption.

35. (NEW) The method as claimed in claim 20, wherein the part of the protocol data unit available on the application layer includes at least one of a sequence number and a timestamp.

36. (NEW) The method as claimed in claim 20, wherein the encryption operation is performed using one of a one-way hash function and a message authentication code with a key known only to the receiver and the transmitter to be authenticated.

37. (NEW) The method as claimed in claim 19, wherein the secret between the transmitter and the receiver is one of a key for symmetric encryption and a key pair for asymmetric encryption.

38. (NEW) The method as claimed in claim 19, wherein the part of the protocol data unit available on the application layer includes at least one of a sequence number and a timestamp.

39. (NEW) The method as claimed in claim 19, wherein the secret is an encryption operation performed using one of a one-way hash function and a message authentication code with a key known only to the receiver and the transmitter to be authenticated.

40. (NEW) The method as claimed in claim 19, wherein said ascertaining includes cryptographic verification using parameters and including at least one of a decryption operation and a cryptographic checksum check.

41. (NEW) The method as claimed in claim 19, wherein said ascertaining includes cryptographic verification using parameters and including at least one of a decryption operation and a cryptographic checksum check.

42. (NEW) A system for transmitting data, comprising:

a transmitter to extend the data by authentication data using an application protocol on an application layer; and

a receiver to determine whether the transmitter is known based on the authentication data, to accept the data if the transmitter is known and to reject the data if the transmitter is not known,

wherein the application protocol comprises a Real Time Transport Protocol (RTP) having an RTP header comprising at least one of a sequence number and a time stamp, and

wherein the extending the data by the authentication data using the application protocol comprises generating a message authentication code (MAC) according to a common secret between the transmitter and the receiver and using, as the application protocol, the RTP header sequence number and timestamp, and appending the MAC to an RTP packet, and if a MAC obtained by a receiver is same as a MAC contained in a transmitted RTP packet received by the receiver, the receiver accepts the transmitted data.

43. (NEW) A transmitter for sending data to a receiver, comprising:

a processor to extend the data by authentication data using an application protocol on an application layer; and

a transmitter to send the data and the authentication data to the receiver,

wherein the application protocol comprises a Real Time Transport Protocol (RTP) having an RTP header comprising at least one of a sequence number and a time stamp, and

wherein the processor extends the data by the authentication data using the application protocol, by generating a message authentication code (MAC) according to a common secret between the transmitter and the receiver and using, as the application protocol, the RTP header sequence number and timestamp, and appending the MAC to an RTP packet.

44. (NEW) A receiver for receiving data from a transmitter, comprising:

a processing unit to control the receiver according to a process comprising:

communicably connecting with the transmitter,

determining whether the transmitter is known based on authentication data added by the transmitter according to an application protocol on an application layer, and

accepting the data if the transmitter is known, and rejecting the data if the transmitter is not known,

wherein the application protocol comprises a Real Time Transport Protocol (RTP) having an RTP header comprising at least one of a sequence number and a time stamp, and

wherein the processing unit obtains the authentication data added by the transmitter according to the application protocol, by generating a message authentication code (MAC) according to a common secret between the transmitter and the receiver and using, as the application protocol, the RTP header sequence number and timestamp.

45. (NEW) A method of transmitting data from a transmitter to a receiver, comprising:

extending the data in the transmitter by authentication data using an application protocol on an application layer;

ascertaining, by the receiver, whether the receiver knows the transmitter based on the authentication data; and

accepting the data at the receiver, if the receiver knows the transmitter, and otherwise rejecting the data,

wherein the application protocol comprises a Real Time Transport Protocol (RTP) having an RTP header comprising at least one of a sequence number and a time stamp, and

wherein the extending the data by the authentication data using the application protocol comprises generating a message authentication code (MAC) according to a common secret between the transmitter and the receiver and using, as the application protocol, the RTP header sequence number and timestamp, and appending the MAC to an RTP packet, and if a MAC obtained by a receiver is same as a MAC contained in a transmitted RTP packet received by the receiver, the receiver accepts the transmitted data.

REMARKS

STATUS OF CLAIMS

Claims 1-16 have been pending.

Office Action page 2, item 3 rejects claims 1-26 (understood to be claims 1-16 because only 16 claims appear to be pending for consideration by the Office Action) under 35 USC 102(b) as being anticipated by Chen (WO 99/11019).

Office Action page 5, item 4 rejects claims 9 and 11-13 under 35 USC 103(a) as being unpatentable over Chen (WO 99/11019).

According to the foregoing, pending claims 1-16 have been cancelled without disclaimer or prejudice.

New claims 17-45 have been added.

Thus, claims 17-45 remain pending for reconsideration, which is respectfully requested.

No new matter has been added in this Amendment.

STATUS OF PRELIMINARY AMENDMENT OF JULY 18, 2002

Applicants submitted a Preliminary Amendment, including claim amendments and a Substitute Specification, on July 18, 2002 concurrent with filing/entry into national stage of the present pending US patent application. The USPTO's Patent Application Information Retrieval (PAIR) including file image wrapper confirm filing of the same by the Applicant on July 18, 2002. Enclosed is also a USPTO stamped postcard evidencing filing of the same by the Applicant on July 18, 2002.

However, the outstanding first Office Action of October 5, 2005 does not confirm entry of the July 18, 2002 Preliminary Amendment, which submitted a Substitute Specification and claim amendments, canceling claims 1-16 and adding new claims 17-44, so that the Office Action appears to only examine and consider original claims 1-16. The July 18, 2002 Preliminary Amendment new claims 17-44 generally improved form of original claims 1-16 according to US practice.

According to the forgoing, the July 18, 2002 Preliminary Amendment new claims 17-44 are submitted herewith as new claims 17-44 and as further amended to clarify a patentably

distinguishing feature of the claimed present invention. According to the foregoing a new claim 45 is added. Therefore, this Amendment is fully responsive pursuant to 37 CFR 1.111 to the Office Action.

Further, another copy of the Substitute Specification is submitted herewith, entry of which is respectfully requested.

IN THE SPECIFICATION

Pursuant to 37 CFR 1.125(b), please replace the pending specification with the substitute specification, including Abstract, but excluding the claims, attached hereto. The substitute specification includes no new matter. According to 37 CFR 1.125(c) a marked up copy of the substitute specification is submitted showing all the changes relative to the immediate prior version of the specification of record.

35 USC 102 REJECTION

The independent claims are 17, 42, 43, 44 and 44. In contrast to Chen, the claimed present invention, using new claim 17 as an example, provides:

> 17. (NEW) A method for transmitting data from a transmitter to a receiver, comprising:
>
> ***extending the data*** in the transmitter ***by authentication data using an application protocol on an application layer***;
>
> ascertaining, by the receiver, whether the receiver knows the transmitter based on the authentication data; and
>
> accepting the data at the receiver, if the receiver knows the transmitter, and otherwise rejecting the data,
>
> wherein ***the application protocol comprises a Real Time Transport Protocol (RTP) and the authentication data is inserted at end of an RTP payload***.

In contrast to Chen, the claimed present invention provides "***extending the data*** in the transmitter ***by authentication data using an application protocol on an application layer*** ..., wherein ***the application protocol comprises a Real Time Transport Protocol (RTP) and the authentication data is inserted at end of an RTP payload***." In other words, Chen is silent on using an application protocol, such as "***RTP***," based upon which to provide authentication data for a transmitter and a receiver. The present substitute specification paragraphs 36, 37 and 38 and FIGS. 3-4 support claims 17, 42, 43 and 44.

Independent claims 42, 43 and 44 are apparatus type claims of similar scope to independent claim 17, and are patentably distinguishing over Chen.

The claimed present invention takes into consideration the following: A special class of application protocols is communication protocols with real-time properties. Such a protocol, such as for example the RTP (Real Time Transport Protocol – see also the literature citation ([3]) on the audio, speech and/or video transmission of data packets – must satisfy, in addition to the pure data transfer, also a series of additional special properties regarding real-time requirements. This includes a treatment, as free of delays as possible, of the data to be sent and received in an interaction with the corresponding application, such as speech/video compression or a processing of application-specific signaling information within the data packet. Furthermore, such a real-time protocol has a data volume requirement, especially to keep the packet size as small as possible, and to keep as small as possible the overhead specific to the application level, such as data fields with address, additional and control information. In this way, requirements regarding the available transmission bandwidth have to be met.

For the fast processing of the data packet to application level, heed must also be paid to the fact that a pipeline-like processing of the headers and payload data fields is possible in the logical stream-based order, which can be enforced by an appropriate arrangement of the data fields in the packet. In addition, certain data fields must be present in the clear text in order to assure early processing steps, such as multiplexing or quality of service control, before security functions can be completed. On the other hand, certain security-sensitive data fields are not to be perceived in the clear text, while other security data fields are allowed to be present in the clear text, but are not needed until security functions are performed.

Likewise, various marginal conditions are to be heeded, for example, that various applications (i.e., speech, video, control protocol information) require different application-specific fields in the data packet. Some of these application-specific factors are not represented in other applications. Thus, speech packets usually do not contain video parameters, and control protocol data contain no speech and video data. The consequence is that there are both common established data fields for all applications, as well as specific, variable and/or optimum data fields in the data packet, which occur only in specific cases. While data protocols incapable of real time data, such as IPsec, TLS, etc., for example, use a number of complex security

mechanisms and security algorithms, real-time capable data transfer protocols use only certain simple security mechanisms and security algorithms to assure the real-time requirement. Thus, no complex security real-time protocols, which include additional handshakes or comprehensive additional security fields, are applicable to real-time capable communication protocols. Accordingly, a problem targeted by the claimed present invention is to expand real-time capable data with additional security data, such that the base properties of the data transmission protocol are not negatively affected.

Regarding the security functions that are to be expanded, a compromise must be found between achievable security and the necessary performance and the available volume of data for maintaining bandwidth. Furthermore, for the reasons given above, not all available security techniques can equally be used as they are otherwise common in plain data transfer protocols incapable of real time. In those cases in which the given application protocol cannot be expanded for reasons of upward/downward compatibility with older versions of the protocol, it is necessary to connect the existing data fields of the application with security data such that effective protection is assured. Where this is not sufficient, the problem is to find suitable expansions of the data protocol with additional data fields and security processing steps so that the said requirements regarding performance, data volume and upward/downward compatibility are satisfied. Since not all security mechanisms and security algorithms of a general real-time capable data protocol equally satisfy real-time requirements, an additional task consists in finding suitable security mechanisms and corresponding security algorithms which satisfy the above-named real-time requirements. Additionally, in the expansion of the real-time capable data protocol, the problem is to preserve of the upward/downward compatibility of earlier versions of the data transfer protocol with older versions of the data transfer protocol, which have additional security functions. This means that the additional security functions must able to be integrated easily into existing hardware and/or software systems of the data transmission protocol without necessitating a complete re-implementation. This requirement is the result of economic considerations involved in the production of a practical product.

This problem is solved in each case by the features of the new independent patent claims, using claim 17 as an example, by "***extending the data*** in the transmitter ***by authentication data using an application protocol on an application layer*** ..., wherein ***the application protocol comprises a Real Time Transport Protocol (RTP)*** and ***the authentication data is inserted at end of an RTP payload***." In other words, although Chen

pages 8-9, which is relied upon in the Office Action, discusses "encryption and mutual authentication at the applications level utilizing a dedicated server known as an authentication server and authentication client software installed at the application level on the client computers" (Chen page 9, 1st paragraph), Chen's "mutual authentication and encryption services to lower layers by intercepting function calls or data packets" based upon the authentication server to establish a session key, differs from the claimed present invention's "***extending the data*** in the transmitter ***by authentication data using an application protocol on an application layer*** ..., wherein ***the application protocol comprises a Real Time Transport Protocol (RTP)*** and ***the authentication data is inserted at end of an RTP payload***." In other words, Chen is silent on using an application protocol, such as "***RTP***," based upon which to provide authentication data for a transmitter and a receiver. The present substitute specification paragraphs 36, 37 and 38 and FIGS. 3-4 support claims 17, 42, 43 and 44. A benefit of the claimed present invention is all applications can be equally secured by the invented generic arrangement of the authentication fields in the data packet.

A benefit of the chosen arrangement of the authentication fields at the end of the payload permits an advantageous expandability of the present RPT protocol. Thus, not only symmetrical security algorithms are employed, but also asymmetrical security algorithms by means of private/public keys.

An additional advantage of the claimed present invention's solution is that additional time-consuming operations, such as decoding, can be spared if an erroneous authentication has been recognized. In this way denial-or-service attacks can better be warded off.

It is an additional advantage of the present invention that network and transport transfer crews can be relieved of complicated security functions, and in addition both the volume of data and the protocol overload can be reduced. Thus, attacks by unauthorized data packets can be efficiently detected and warded off.

In particular, the method according to the claimed present invention as recited in independent claims 17, 42, 43, 44, and 45 (claim 45 discussed below) has the advantage that this arrangement of the authentication fields within the data transmission protocol assures that the authentication data are attached to the data packet with upward and downward compatibility, and thus a good ability of integration into existing systems is assured. Furthermore, with this system a logical, flow-based processing sequence of high performance is assured.

Chen cannot anticipate the claimed present invention as recited in independent claims 17, 42-44, because Chen fails to disclose, either expressly or inherently, each and every element of the claimed present invention's "***extending the data*** in the transmitter ***by authentication data using an application protocol on an application layer*** ..., wherein ***the application protocol comprises a Real Time Transport Protocol (RTP)*** and ***the authentication data is inserted at end of an RTP payload***." In other words, Chen is silent on using an application protocol, such as "***RTP***," based upon which to provide authentication data for a transmitter and a receiver. The present substitute specification paragraphs 36, 37 and 38 and FIGS. 3-4 support claims 17, 42, 43 and 44.

NEW CLAIM 45

In contrast to Chen, the claimed present invention as recited in independent claim 45 provides:

> 45. (NEW) A method of transmitting data from a transmitter to a receiver, comprising:
>
> extending the data in the transmitter by authentication data using an application protocol on an application layer;
>
> ascertaining, by the receiver, whether the receiver knows the transmitter based on the authentication data; and
>
> accepting the data at the receiver, if the receiver knows the transmitter, and otherwise rejecting the data,
>
> wherein the application protocol comprises a Real Time Transport Protocol (RTP) having an RTP header comprising at least one of a sequence number and a time stamp, and
>
> wherein the extending the data by the authentication data using the application protocol comprises generating a message authentication code (MAC) according to a common secret between the transmitter and the receiver and using, as the application protocol, the RTP header sequence number and timestamp, and appending the MAC to an RTP packet, and if a MAC obtained by a receiver is same as a MAC contained in a transmitted RTP packet received by the receiver, the receiver accepts the transmitted data.

The present substitute specification paragraphs 36, 37 and 38 and FIGS. 3-4 support new claim 45. Although Chen pages 8-9, which is relied upon in the Office Action, discusses "encryption and mutual authentication at the applications level utilizing a dedicated server known as an authentication server and authentication client software installed at the application level on

the client computers" (Chen page 9, 1st paragraph), Chen's "mutual authentication and encryption services to lower layers by intercepting function calls or data packets" based upon the authentication server to establish a session key, differs from the claimed present invention (as also discussed with respect to independent claim 17), "***extending*** the data in the transmitter by ***authentication data using an application protocol*** on an application layer; ... wherein the application protocol comprises ***a Real Time Transport Protocol (RTP) having an RTP header comprising at least one of a sequence number and a time stamp***, and wherein the ***extending*** the data ***by the authentication data using the application protocol comprises generating a message authentication code (MAC) according to a common secret between the transmitter and the receiver and using, as the application protocol, the RTP header sequence number and timestamp***, and ***appending the MAC to an RTP packet***, and if a ***MAC obtained*** by a receiver ***is same as a MAC contained*** in a transmitted RTP packet received by the receiver, the receiver accepts the transmitted data" (claim 45). In other words, Chen is silent on using an application protocol, such as "***RTP***," based upon which to provide authentication data for a transmitter and a receiver, where the authentication data using RTP application protocol is provided by "***generating a message authentication code (MAC) according to a common secret between the transmitter and the receiver and using, as the application protocol, the RTP header sequence number and timestamp***." Chen cannot anticipate the claimed present invention as recited in independent claim 45, because Chen fails to disclose, either expressly or inherently, each and every element of the claimed present invention.

In view of the forgoing remarks and claim amendments, withdrawal of the rejection of pending claims and allowance pending claims is respectfully requested.

CONCLUSION

There being no further outstanding objections or rejections, it is submitted that the application is in condition for allowance. An early action to that effect is courteously solicited.

Finally, if there are any formal matters remaining after this response, the Examiner is requested to telephone the undersigned to attend to these matters.

Respectfully submitted,
STAAS & HALSEY LLP

Date: February 6, 2006

By: _____
Mehdi D. Sheikerz
Registration No. 41,307

1201 New York Avenue, NW, 7th Floor
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501

OIPE IAP13 FEB 06 2006 PATENT & TRADEMARK OFFICE

Please Date Stamp and return

PCT TRANSMITTAL; FEE TRANSMITTAL, GERMAN APPLICATION WITH 3 SHEETS OF DRAWINGS ATTACHED; ENGLISH TRANSLATION WITH 3 SHEETS OF FORMAL DRAWINGS; PRELIMINARY AMENDMENT; SUBSTITUTE SPECIFICATION AND MARKED-UP SPECIFICATION; INFORMATION DISCLOSURE STATEMENT WITH 7 REFERENCES; COPIES OF INT'L SEARCH REPORT, IPER AND FIRST PAGE OF PUBLISHED INT'L APPLICATION; DECLARATION; ASSIGNMENT WITH ASSIGNMENT COVERSHEET AND CHECK FOR $1,140.00

APPLICANT(S): Martin EUCHNER

SERIAL NO: NEW PCT NATIONAL PHASE APPLICATION

CONFIRMATION NO.

TITLE: METHOD, SYSTEM FOR TRANSMITTING DATA FROM A TRANSMITTER TO A RECEIVER, AND TRANSMITTER OR RECEIVER

FILING DATE: JULY 18, 2002

10/181564
533 Rec'd PCT/PTO 18 JUL 2002

DOCKET NO: 1454.1302/RAG:fc

DUE DATE: JULY 18, 2002

EARLY NOTIFICATION OF SERIAL NUMBER

Please Date Stamp and return

PCT TRANSMITTAL; FEE TRANSMITTAL, GERMAN APPLICATION WITH 3 SHEETS OF DRAWINGS ATTACHED; ENGLISH TRANSLATION WITH 3 SHEETS OF FORMAL DRAWINGS; PRELIMINARY AMENDMENT; SUBSTITUTE SPECIFICATION AND MARKED-UP SPECIFICATION; INFORMATION DISCLOSURE STATEMENT WITH 7 REFERENCES; COPIES OF INT'L SEARCH REPORT, IPER AND FIRST PAGE OF PUBLISHED INT'L APPLICATION; DECLARATION; ASSIGNMENT WITH ASSIGNMENT COVERSHEET AND CHECK FOR $1,140.00

APPLICANT(S): Martin EUCHNER

SERIAL NO: NEW PCT NATIONAL PHASE APPLICATION

CONFIRMATION NO.

TITLE: METHOD, SYSTEM FOR TRANSMITTING DATA FROM A TRANSMITTER TO A RECEIVER, AND TRANSMITTER OR RECEIVER

FILING DATE: JULY 18, 2002

DOCKET NO: 1454.1302/RAG:fc

DUE DATE: JULY 18, 2002

OIPE JC98 JUL 18 2002 PATENT & TRADEMARK OFFICE

U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE

TRANSMITTAL LETTER TO THE UNITED STATES DESIGNATED/ELECTED OFFICE (DO/EO/US) CONCERNING A FILING UNDER 35 U.S.C. 371

ATTORNEY'S DOCKET NUMBER
1454.1302

COPY

INTERNATIONAL APPLICATION NO.	INTERNATIONAL FILING DATE	PRIORITY DATE CLAIMED
PCT/DE01/00021	5 January 2001	18 January 2000

TITLE OF INVENTION
METHOD, SYSTEM FOR TRANSMITTING DATA FROM A TRANSMITTER TO A RECEIVER, AND TRANSMITTER OR RECEIVER

APPLICANT(S) FOR DO/EO/US
Martin EUCHNER

Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. ☒ This is a FIRST submission of items concerning a filing under 35 U.S.C. 371.
2. ☒ This is an express request to immediately begin national examination procedures (35 U.S.C. 371(f)).
3. ☒ The US has been elected by the expiration of 19 months from the priority date (PCT Article 31).
4. ☒ A copy of the International Application as filed (35 U.S.C. 371(c)(2))
   a. ☒ is transmitted herewith (required only if not transmitted by the International Bureau).
   b. ☐ has been transmitted by the International Bureau.
   c. ☐ is not required, as the application was filed in the United States Receiving Office (RO/US).
5. ☒ A translation of the International Application into English (35 U.S.C. 371(c)(2)).
6. ☐ Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371(c)(3))
   a. ☐ are transmitted herewith (required only if not transmitted by the International Bureau).
   b. ☐ have been transmitted by the International Bureau.
   c. ☐ is not required, as the application was filed in the United States Receiving Office (RO/US)
7. ☐ A translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)).
8. ☒ An oath or declaration of the inventor (35 U.S.C. 371(c)(4)).
9. ☐ A translation of the Annexes to the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371(c)(5)).

Items 10-15 below concern document(s) or information included:

10. ☒ An Information Disclosure Statement Under 37 CFR 1.97 and 1.98.
11. ☒ An assignment document for recording.
    Please mail the recorded assignment document to:
    a. ☒ the person whose signature, name & address appears at the bottom of this document.
    b. ☐ the following:
12. ☒ A preliminary amendment.
13. ☒ A substitute specification
14. ☐ A change of power of attorney and/or address letter.
15. ☒ Other items or information:

Copies of PCT EASY forms filed with International Application, cover page of International Application as published, International Search Report, and International Preliminary Examination Report.

©2001 Staas & Halsey LLP

☒ The U.S. National Fee (35 U.S.C. 371(c)(1)) and other fees as follows:

CLAIMS	(1) FOR	(2) NUMBER FILED	(3) NUMBER EXTRA	(4) RATE	(5) CALCULATIONS
	TOTAL CLAIMS	27 -20=	7	x $ 18.00	126.00
	INDEPENDENT CLAIMS	4 -3=	1	x $ 84.00	84.00
	MULTIPLE DEPENDENT CLAIM(S) (if applicable)			+$280.00	0.00
	BASIC NATIONAL FEE (37 CFR 1.492(a)(1)-(4): ☐ Neither international preliminary examination fee (37 CFR 1.482) nor international search fee (37 CFR 1.445(a)(2)) paid to USPTO .....$1,040 ☐ International preliminary examination fee (37 C.F.R. 1.482) not paid to USPTO but International Search Report prepared by the EPO or JPO.....$ 890 ☐ International preliminary examination fee (37 C.F.R. 1.482) not paid to USPTO but international search fee (37 C.F.R. 1.445(a)(2) paid to USPTO...$ 740 ☐ International preliminary examination fee paid to USPTO (37 CFR 1.482) but all claims did not satisfy provision of PCT Article 33(1)-(4).....$ 710 ☐ International preliminary examination fee paid to USPTO (37 CFR 1.482) and all claims satisfied provisions of PCT Article 33(2) to (4) .....$ 100				890.00
	Surcharge of $130 for furnishing the National fee or oath or declaration later than ☐ 20 ☐ 30 mos. from the earliest claimed priority date (37 CFR 1.482(e)).				0.00
			TOTAL OF ABOVE CALCULATIONS		1,100.00
	Reduction by 1/2 for filing by small entity, if applicable. Affidavit must be filed also. (Note 37 CFR 1.9, 1.27, 1.28.)				
			SUBTOTAL		1,100.00
	Processing fee of $130 for furnishing the English Translation later than [ ] 20 [ ] 30 mos. from the earliest claimed priority date (37 CFR 1.482(f)).				
			TOTAL NATIONAL FEE		1,100.00
	Fee for recording the enclosed assignment (37 CFR 1.21(h)).				+ 40.00
			TOTAL FEES ENCLOSED		1,140.00

a. ☒ A check in the amount of $1,140.00 to cover the above fees is enclosed.
b. ☐ Please charge my Deposit Account No. 19-3935 in the Amount of $ to cover the above fees. A duplicate copy of this sheet is enclosed.
c. ☒ The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account No. 19-3935. A duplicate copy of this sheet is enclosed.



21171
PATENT TRADEMARK OFFICE

SUBMITTED BY: STAAS & HALSEY LLP

Type Name	Richard A. Gollhofer	Reg. No.	31,106
Signature	Richard A. Gollhofer	Date	7/18/02

©2001 Staas & Halsey LLP

OIPE IAP13 FEB 06 2006 PATENT & TRADEMARK OFFICE

Docket No.: 1454.1302

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of:

Martin EUCHNER

Serial No. Group Art Unit:

COPY

Confirmation No.

Filed: (concurrently) Examiner:

For: METHOD AND SYSTEM FOR TRANSMITTING DATA FROM A TRANSMITTER TO A RECEIVER, AND TRANSMITTER AND RECEIVER THEREFOR (as amended)

PRELIMINARY AMENDMENT

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

Before examination of the above-identified application, please amend the application as follows:

IN THE TITLE:

Please change "METHOD," to --METHOD AND-- and "OR RECEIVER" to --AND RECEIVER THEREFOR--.

IN THE SPECIFICATION:

Please REPLACE the pending specification with the substitute specification attached hereto.

IN THE CLAIMS:

Please cancel without prejudice or disclaimer claims 1-16 and ADD new claims 17-43 in accordance with the following:

16. (NEW) A method for transmitting data from a transmitter to a receiver, comprising:

extending the data in the transmitter by authentication data using an application protocol on an application layer;

ascertaining, by the receiver, whether the receiver knows the transmitter based on the authentication data; and

accepting the data at the receiver, if the receiver knows the transmitter, and otherwise rejecting the data.

17. (NEW) The method as claimed in claim 16, wherein the data are transmitted on a packet-oriented basis.

18. (NEW) The method as claimed in claim 16, wherein the authentication data are determined based on at least part of a protocol data unit available on the application layer, and the application protocol being linked to a secret by the transmitter.

19. (NEW) The method as claimed in claim 18, wherein the linking to the secret is carried out by at least one of cryptographical linking to at least one further parameter, particularly by virtue of an encryption operation and a cryptographic checksum and key information.

20. (NEW) The method as claimed in claim 19, wherein said ascertaining includes verification at the receiver to determine whether the receiver knows the transmitter.

21. (NEW) The method as claimed in claim 20, wherein the verification is cryptographic verification using parameters and including at least one of a decryption operation and a cryptographic checksum check.

22. (NEW) The method as claimed in claim 21, wherein the secret between the transmitter and the receiver is one of a key for symmetric encryption and a key pair for asymmetric encryption.

23. (NEW) The method as claimed in claim 22, wherein the part of the protocol data unit available on the application layer includes at least one of a sequence number and a timestamp.

24. (NEW) The method as claimed in claim 23, wherein the encryption operation is performed using one of a one-way hash function and a message authentication code with a key known only to the receiver and the transmitter to be authenticated.

25. (NEW) The method as claimed in claim 24, wherein the authentication is carried out between the transmitter and the receiver before the data are transmitted from the transmitter to the receiver.

26. (NEW) The method as claimed in claim 25, wherein the receiver and transmitter provide packet-switched telephone services.

27. (NEW) The method as claimed in claim 26, wherein the receiver and transmitter are each disposed in at least one of a switching node and a switching installation.

28. (NEW) The method as claimed in claim 25, wherein the receiver and transmitter are used in Internet telephony.

29. (NEW) The method as claimed in claim 28, wherein the receiver and transmitter are each disposed in at least one of a switching node and a switching installation.

30. (NEW) The method as claimed in claim 20, wherein the secret between the transmitter and the receiver is one of a key for symmetric encryption and a key pair for asymmetric encryption.

31. (NEW) The method as claimed in claim 20, wherein the part of the protocol data unit available on the application layer includes at least one of a sequence number and a timestamp.

32. (NEW) The method as claimed in claim 20, wherein the encryption operation is performed using one of a one-way hash function and a message authentication code with a key known only to the receiver and the transmitter to be authenticated.

33. (NEW) The method as claimed in claim 19, wherein the secret between the transmitter and the receiver is one of a key for symmetric encryption and a key pair for asymmetric encryption.

34. (NEW) The method as claimed in claim 19, wherein the part of the protocol data unit available on the application layer includes at least one of a sequence number and a timestamp.

35. (NEW) The method as claimed in claim 19, wherein the encryption operation is performed using one of a one-way hash function and a message authentication code with a key known only to the receiver and the transmitter to be authenticated.

36. (NEW) The method as claimed in claim 18, wherein the secret between the transmitter and the receiver is one of a key for symmetric encryption and a key pair for asymmetric encryption.

37. (NEW) The method as claimed in claim 18, wherein the part of the protocol data unit available on the application layer includes at least one of a sequence number and a timestamp.

38. (NEW) The method as claimed in claim 18, wherein the secret is an encryption operation performed using one of a one-way hash function and a message authentication code with a key known only to the receiver and the transmitter to be authenticated.

39. (NEW) The method as claimed in claim 18, wherein said ascertaining includes cryptographic verification using parameters and including at least one of a decryption operation and a cryptographic checksum check.

40. (NEW) The method as claimed in claim 18, wherein said ascertaining includes cryptographic verification using parameters and including at least one of a decryption operation and a cryptographic checksum check.

41. (NEW) A system for transmitting data, comprising:

a transmitter to extend the data by authentication data using an application protocol on an application layer; and

a receiver to determine whether the transmitter is known based on the authentication data, to accept the data if the transmitter is known and to reject the data if the transmitter is not known.

42. (NEW) A transmitter for sending data to a receiver, comprising:

a processor to extend the data by authentication data using an application protocol on an application layer; and

a transmission unit to send the data and authentication data to the receiver.

43. (NEW) A receiver for receiving data from a transmitter, comprising:

a processing unit, coupled to the transmitter, to determine whether the transmitter is known based on authentication data added by the transmitter according to an application protocol on an application layer, to accept the data if the transmitter is known, and to reject the data if the transmitter is not known.

IN THE ABSTRACT:

Please DELETE the Abstract in its entirety and replace with the attached Substitute Abstract.

REMARKS

This Preliminary Amendment is submitted to improve the form of the English translation as filed. It is respectfully requested that this Preliminary Amendment be entered in the above-referenced application.

In accordance with the foregoing, claims 1-16 have been canceled and claims 17-43 have been added. Thus, claims 17-43 are pending and are under consideration.

A substitute specification is also being filed herewith. The substitute specification is accompanied by a marked-up copy of the original specification.

If there are any questions regarding these matters, such questions can be addressed by telephone to the undersigned. Otherwise, an early action on the merits is respectfully solicited.

If any further fees are required in connection with the filing of this Preliminary Amendment, please charge same to our Deposit Account No. 19-3935.

Respectfully submitted,

STAAS & HALSEY LLP

Date: 7/18/02

By: Richard A. Gollhofer
Richard A. Gollhofer
Registration No. 31,106

700 Eleventh Street, NW, Suite 500
Washington, D.C. 20001
(202) 434-1500

OIPE IAP13 FEB 06 2006 PATENT & TRADEMARK OFFICE

SUBSTITUTE SPECIFICATION

TITLE OF THE INVENTION

METHOD AND SYSTEM FOR TRANSMITTING DATA FROM A TRANSMITTER TO A RECEIVER, AND TRANSMITTER AND RECEIVER THEREFOR

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application is based on and hereby claims priority to German Application No. 10001855.6 filed on 18 January 2000, the contents of which are hereby incorporated by reference.

BACKGROUND OF THE INVENTION

[0002] The invention relates to a method and a system and for transmitting data from a transmitter to a receiver, and to the transmitter and the receiver therefor.

[0003] The transmission of data is known, by way of example, on the basis of the OSI reference model as described in Andrew S. Tanenbaum, Computer-Netzwerke; Wolfram's Fachverlag, Attenkirchen 1992, pages 17 - 32. The OSI reference model (OSI model for short) from the International Standards Organization (ISO) has seven layers, each of which has a different functionality (in terms of abstraction). In the OSI model, layer 1 corresponds to a physical layer, where data and messages are transmitted from the transmitter to the receiver using a physical protocol. In layer 2, a data link layer is provided, and a data link protocol is used for transmission from layer 2 of the transmitter to layer 2 of the receiver. In a similar manner, layer 3, a network layer, uses a network protocol between transmitter and receiver, and layer 4, a transport layer, uses a transport protocol. In the OSI model, layer 5 corresponds to a session layer using a session protocol, layer 6 corresponds to a presentation layer using a presentation protocol, and layer 7 corresponds to an application layer using an application protocol. In practice, many applications do not always permit exact nomination of strict boundaries specifically between the upper protocol layers, particularly layers 5 to 7. By way of example, within the context of Internet telephony, i.e. use of the telephone service using the medium of the Internet, the three layers above the transport layer 4 combine to form an "application layer" for which, in a similar way to in the implementation above, an "application protocol" is used.

[0004] A special feature of the OSI model and hence of all communication models following the OSI model is that, as a result of the division into layers, in each layer the bottom layers perform functions fully transparently with regard to the currently considered layer and provide this upper layer with a service which is determined by all of the functionalities of the bottom layers. In this context, "transparent" means that currently considered layer does not need to concern itself with the functionalities of the bottom layers. In the case discussed here, the application protocol can thus be used between an application layer transmitter and an application layer receiver. This may cover numerous services, for example for switching, for protection or for actual transmission via a physical channel; these need no longer be a concern from the point of view of the application protocol, particularly as a user of this protocol.

[0005] Similarly with respect to this consideration, the individual layers have "protocol data units" (PDUs) between them which can be designed specifically for each protocol of the respective layer. Thus, for each layer, the respective protocol can comprise a dedicated header containing administrative information for the respective layer's protocol, with this header also being able to be seen and used only by the respective protocol's layer in accordance with the OSI model. Details about the design of the OSI model can be found in numerous pieces of literature, inter alia in those cited above.

[0006] When referring to application layer below, this denotes the layers above the transport layer (layers above 4). The application protocol also denotes the protocol for communication between a transmitter application layer situated above the transport layer and a receiver application layer situated above the transport layer.

[0007] A message authentication code (MAC) is known generally and denotes a cryptographic checksum which is intended to be used to identify an alteration in a message or in data (see Christoph Ruland, Informationssicherheit in Datennetzen (which translates as Information Security in Data Networks), DATACOM-Verlag; Bergheim, 1993, pp.61 - 63 and 68 - 79).

[0008] A one-way hash function is known from Ruland, pp. 68 - 79, for example. Such a one-way hash function cannot be used to calculate the correct input value for a given function value. Another feature is the absence of collision, i.e. it must not be possible to use the one-way hash function to obtain the same output value for two different input values.

[0009] One difference between the hash code and the MAC is that the MAC requires a secret key in order to calculate it, whereas the hash function can be independent of a key and can be known publicly.

[0010] In addition, an asymmetric cryptographic method (also: public key method) is known, e.g. from Ruland, pp. 73 - 79. Each party involved in an asymmetric cryptographic method receives two keys, a public key and a secret (or private) key. In principle, the secret key can be derived from the public key, with this task needing to be as complex as possible. The asymmetric method can also be used to produce an electronic signature (authentication) and/or to encrypt the content of a message (using the receiver's public key) such that only the receiver can decrypt the message again using its secret key.

[0011] Finally, there is also a symmetric method, which requires a (secret) key used both for encryption and decryption. An example of a symmetric encryption method is the DES algorithm.

[0012] A communication system is subject to a multiplicity of possible attacks which can target, inter alia, the content of the messages interchanged in the communication system or the availability of the communication system. Taking the example of Internet telephony, then it is firstly important for the content of the telephone conversation not to be able to be monitored by an unauthorized third party, and secondly it is also necessary for the third party not to be able to initiate countless calls, and hence for him not to interfere with the receiver and unnecessarily encumber the communication system. Such attacks are also called denial-of-service attacks. Possible examples of such attacks are mass data or mass messages which are automatically generated and sent to one or more receivers, where they significantly encumber availability and performance. In the Internet telephony example, it is at least unwanted for an Internet telephone to ring incessantly and hence not only to load the network unnecessarily but also to distract the receiver's attention. It is likewise perceived to be disruptive if an attacker interferes with the flow of communication between transmitter and receiver by sending unauthorized voice data.

[0013] To fend off attacks on a private communication network, "firewalls" are often used which ensure, in particular, that the private communication network is separated from public networks. However, it is a simple matter for an attacker to send his meaningless (mass) data to addressees within the private network as well, that is to say behind the firewall. There, these data are decoded and possibly reproduced. Thus, the effect achieved by the attacker is that, by

way of example, the reproduction of such meaningless data produces nothing but annoying audio interference or noise and hence restricts the bandwidth and availability of the private network. In the extreme case, part or all of the private communication network can even crash.

[0014] In addition, an RTP protocol for transmitting media data ("payload"), i.e. video or audio data, is known from H. Schulzrinne: RTP: A Transport Protocol for Real-Time Applications; Internet Engineering Task Force, RFC1889, 1996.

[0015] One approach to holding off unwanted data is provided by the IPSEC protocol described S. Kent, R. Atkinson; IP Authentication Header, Internet Engineering Task Force, RFC2402, 1998 and S. Kent, R. Atkinson; IP Encapsulating Security Payload (ESP), Internet Engineering Task Force, RFC2406, 1998. In this case, data packets of the Internet protocol (IP data packets) can be encapsulated and can be protected in terms of confidentiality and/or integrity (implicit sender authentication). In addition, the IPSEC protocol affords a key management method using a "cookie" mechanism (see D. Harkins, D. Carrel, The Internet Key Exchange (IKE), Internet Engineering Task Force, RFC2409, 1998), which can be used to fend off the mass data attacks (denial-of-service attacks) discussed above during the key management phase. The cookie mechanism involves linking fast one-way hash functions (e.g. SHA-1), random numbers and IP addresses to one another. However, the first "cookie" transmitted from the transmitter to the receiver is not protected, which results in a security gap. In addition, the IPSEC cookie method is not suitable for ensuring protection against such unwanted mass data on the application layer (for the application protocol) under real-time conditions as well. However, the OSI layer structure means that a drawback for IPSEC is, among other things, that it is not possible to link IPSEC to the security functions of an application layer in this manner.

SUMMARY OF THE INVENTION

[0016] An object of the invention is to transmit data from a transmitter to a receiver, with the receiver ensuring that the data are not unwanted data arising from a denial-of-service attack, for example.

[0017] First, the object is achieved by specifying a method for transmitting data from a transmitter to a receiver, in which the transmitter extends the data by authentication data using an application protocol on the application layer. The authentication data are used by the

receiver to ascertain whether it knows the transmitter. If the receiver knows the transmitter, the data are accepted, otherwise the data are rejected.

[0018] As stated above, depending on the protocol structure (cf. OSI model), data in each protocol layer have their own transparency with regard to this protocol layer, i.e. the services in protocol layers beneath are performed (without being seen by the current protocol layer). Advantageously, data packets on the application layer, in particular, are free of the administrative data associated with layers beneath, i.e. the data packets in the application layer have, besides the administrative information for the application layer, only the data which are actually to be communicated. Accordingly, authentication in the application layer is particularly advantageous, because the data packets themselves, in contrast to layers beneath, are of significantly reduced size. This small size has an advantageous effect on real time and availability of the entire communication system. If the receiver knows the transmitter, i.e. particularly if the transmitter is successfully authenticated with respect to the receiver, the message from the transmitter is accepted, in particular the data are stored. Otherwise, i.e. if the receiver does not know the transmitter, the data are rejected, i.e. buffering does not take place. This is particularly advantageous if such a decision is made automatically.

[0019] In particular, available security functions and keys on the application layer can advantageously be allocated to the individual application users. This can advantageously be linked to the application data in a filter function outlined above. Special functions at the application level, such as picture and/or sound compression methods, can advantageously be combined with security functions in the application layer, which makes it possible to increase the end systems' performance and reduces the implementation complexity; this also applies to the aforementioned filtering mechanism.

[0020] The described authentication mechanism in the application layer allows transmitters and receivers which know one another to be separated from transmitters which are not known by the receiver and from which the receiver does not accept any messages either. This allows an attack of the type described in the introduction ("denial-of-service" attack) to be effectively prevented, i.e. unwanted mass data are actually rejected when arriving at the transmitter.

[0021] In this context, special note will be made that the receiver in no way has to be an end receiver or addressee. Instead, the receiver itself can be a switching entity, e.g. a switching node or a firewall with switching functionality, and hence can act with regard to authentication of

the transmitter for the end receiver. Thus, in relation to the "firewall" example, it is possible to prevent superfluous network load from being produced by rejecting the unwanted mass data before they even enter the private communication network.

[0022] In this scenario (authentication in a switching entity), it is advantageous, in particular, if the authentication data are generated using an asymmetric method, since it is also possible for the switching entity, in the example the firewall, to ascertain from whom the data are coming, without the decryption secret already needing to be known in the firewall (as in the case of symmetric encryption) for this purpose. The firewall rejects unwanted mass data if it does not know the transmitter; in the other case, that is to say if the switching entity knows the transmitter and the data are by no means unwanted, these data are forwarded to the receiver after the transmitter has been verified. The receiver can still use its private key to decrypt the data and process them further locally (display them, store them, etc.) irrespective of the check on the data's origin, which has already taken place in the firewall. It will be stressed once again that the decryption (in terms of confidentiality) of the data is performed by the addressee using its private key in the asymmetric method described, whereas the origin of the message (which is determined for this addressee) can also be checked in the switching entity using the public key of the sender – irrespective of the content of the message. Thus, an upstream filter functionality can be provided successfully by the switching entity. This results in a clear functional division between the filter function and the application function; the complexity of the terminals can be simplified as a result and the network traffic in the private communication network can be reduced.

[0023] One development is that the data are transmitted on a packet-oriented basis. Another development is that the authentication data are determined by virtue of at least part of a protocol data unit, available on the application layer, of the application protocol being encrypted by the transmitter using a secret. In this case, the secret between the transmitter and the receiver can be a key for symmetric encryption or a key pair for asymmetric encryption (public key method, see description with switching entity, for example). Decryption of the secret determines, particularly at the receiver end, whether or not the receiver knows the transmitter. This is advantageous if data encryption is available on the application level anyway and can also be used for this purpose.

[0024] One refinement is that the authentication data are determined using part of the protocol data unit available on the application layer, in particular a sequence number or a timestamp .

[0025] In particular, one development is that a one-way hash function is used for protection. Protection can also be effected using a message authentication code (MAC), with a key which is known only to the transmitter and to the receiver additionally being necessary.

[0026] Another refinement is that, before the data are transmitted from the transmitter to the receiver, authentication is carried out between the transmitter and the receiver.

[0027] Another refinement is that the described method is used in packet-switching telephone services, particularly within the context of Internet telephony. Alternatively, the method can be used in switching nodes and switching installations.

[0028] The object is achieved in another way by specifying a system for transmitting data, in which a transmitter and a receiver are provided, the transmitter extending the data by authentication data using an application protocol on the application layer. The receiver uses the authentication data to check whether it knows the transmitter. If the receiver knows the transmitter, the data from the transmitter are used, otherwise the receiver rejects the data from the transmitter which is unknown to it.

[0029] The object is also achieved by specifying a transmitter for sending data to a receiver, which extends the data by authentication data using an application protocol on the application layer and sends them to the receiver.

[0030] Finally, the object is achieved by specifying a receiver for receiving data from a transmitter designed on the basis of the statements above, in particular, the receiver using authentication data within the application protocol on the application layer to determine whether it knows the transmitter. If it knows the transmitter, it uses the data, otherwise the receiver rejects the data. The system or the transmitter and the receiver are particularly suitable for carrying out the inventive method or one of its developments explained above.

[0031] Reference will now be made in detail to the preferred embodiments of the present invention, examples of which are illustrated in the accompanying drawings, wherein like reference numerals refer to like elements throughout.

BRIEF DESCRIPTION OF THE DRAWINGS

[0032] These and other objects and advantages of the present invention will become more apparent and more readily appreciated from the following description of the preferred embodiments, taken in conjunction with the accompanying drawings of which:

Figure 1 is a block diagram of a protocol architecture (protocol stack) for a communication system in the layer model;

Figure 2 is a block diagram of an application layer with possible protocols;

Figure 3 is a flowchart between the transmitter and the receiver for transmitting data from the transmitter to the receiver;

Figure 4 is a protocol data unit (PDU) of the application protocol (layer > 5) with authentication data.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0033] Figure 1 is a block diagram of a protocol architecture (protocol stack) for a communication system in the layer model, following the OSI model mentioned above. A transmitter 104 and a receiver 105 interchange messages via a physical connection 103. Both the transmitter 104 and the receiver 105 have the same protocol architecture: layer 1 (110 and 111) represents a physical layer, layer 2 (109, 112) represents a data link layer, layer 3 (108, 113) represents a network layer, layer 4 (107, 114) represents a transport layer and layer 5 (101 106) represents an application layer. Between the individual layers there is a respective dedicated protocol from a transmitter 104 to a receiver 105; for the application layer 101 or 106, this is the application protocol 102, for example. From the point of view of layer 5 (101, 106), protocol data units having the format prescribed in the application protocol 102 are transmitted from the transmitter 104 to the receiver 105. The layers situated below the application layer represent functionalities which are provided transparently for the application layer. On the basis of the OSI model, the functionalities have expediently been chosen to be largely independent of one another.

[0034] Figure 2 is a block diagram of an application layer with possible protocols, based on an application concerning the transmission of media data (audio and video data). Thus, 101 and 106 represent the application layer (layer 5) shown in Fig. 1 for the transmitter 104 and the receiver 105. This application layer is in turn divided into a plurality of sublayers. Thus, directly above the layer 4 is the real time transport protocol (RTP) 201. This protocol allows voice

and/or picture data to be transmitted in real time from one to possibly a plurality of points (addressees). A layer 206 contains a security application which compiles security functions of the application layer; associated security protocols provide for services such as

- User authentication,
- Access control
- Confidentiality,
- Integrity
- Binding nature of the application data, and
- Accounting.

[0035] Depending on the instance of application, audio data 203 and/or video data 202 are used. For both alternatives, there are different compression standards, a few of which are shown by way of example in Fig. 2. For audio data, these are the standards G.711, G.722, G.723.1 and G.729. For video data, these are the two picture compression standards H.261 and H.263. In the next abstraction stage (similarly with respect to the OSI model) come the different possibilities for an application of audio or video data after compression and decompression (see block 204). One more abstraction level above, there is, by way of example, a user interface 205, which fully transparently provides a user with the services beneath which have just been described. By way of example, a user, taking into account both the audio and the video data, can thus make use of video telephony, e.g. over the Internet, without specifically needing to concern himself with any details of the services provided by the layers beneath. He thus uses the video telephony service, visible to him, over the Internet transparently. In this context, various alternative implementation forms can be situated below his service.

[0036] Figure 3 is a flowchart for transmitting data from the transmitter 308 to the receiver 307. In block 301, authentication data are produced on the application layer, and the protocol data unit (PDU) relevant to the application protocol is appended. In accordance with the arrangements of the application protocol, the data, including the authentication data, are transmitted to the receiver 307 (see connection 302). This is done using the functionalities or services available under the application protocol. In block 303, the authentication data are checked in the receiver 307 and there is a test (in block 304) to determine whether the transmitter 308 is known to the receiver 307. If this is not the case, the procedure branches to

block 305 and the data are rejected. If the transmitter 308 is known to the receiver 307, the procedure branches to block 306 and the data are processed further. The authentication between transmitter and receiver was successful, and the transmitted data are not unwanted mass data.

[0037] Figure 4 shows a protocol data unit (PDU) of the application protocol with authentication data. What is shown in Fig. 4 is a protocol data unit (PDU) of the real time transport protocol (RTP, see 201 in Fig. 2) 401. Such an RTP packet 401 comprises an RTP header 406, encrypted media data 404 and authentication data 405. The RTP header 406 comprises, inter alia, a sequence number 402 and a timestamp 403. Both the transmitter and the receiver know a shared secret K, in this case indicated as a key, which is used to generate a message authentication code (MAC), on the basis of the sequence number 402 and the timestamp 403 (see field 405). An advantage with respect to encryption using a DES algorithm, for example, is a field length of 64 bits.

[0038] As already described, such a packet sent by the transmitter is authenticated at the receiver end such that (in relation to the application layer) a message authentication code is generated by field 402 (sequence number) and field 403 (timestamp) on the basis of the key (K) known to the receiver. If this message authentication code is the same as the field 405, then the arriving RTP packet 401 is a data packet coming from a known transmitter, and the data are processed, for example displayed or stored. In the other case, unauthenticated data are involved, the transmitter is unknown to the receiver and the entire RTP packet is rejected.

[0039] The invention has been described in detail with particular reference to preferred embodiments thereof and examples, but it will be understood that variations and modifications can be effected within the spirit and scope of the invention.

ABSTRACT

METHOD AND SYSTEM FOR TRANSMITTING DATA FROM A TRANSMITTER TO A RECEIVER, AND TRANSMITTER AND RECEIVER THEREFOR

Data from a transmitter is extended to include authentication data on the application level by an application protocol. The authentication data is used by the receiver to determine whether the transmitter is known by the receiver. If the transmitter is known by the receiver, the data is accepted. If not, the data is rejected.

OIPE IAP13 FEB 06 2006 PATENT & TRADEMARK OFFICE

MARKED-UP COPY OF SUBSTITUTE SPECIFICATION

[Description] TITLE OF THE INVENTION

METHOD[,] AND SYSTEM FOR TRANSMITTING DATA FROM A TRANSMITTER TO A RECEIVER, AND TRANSMITTER AND RECEIVER THEREFOR

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application is based on and hereby claims priority to German Application No. 10001855.6 filed on 18 January 2000, the contents of which are hereby incorporated by reference.

BACKGROUND OF THE INVENTION

[0002] The invention relates to a method and a system and for transmitting data from a transmitter to a receiver, and to the transmitter and the receiver therefor.

[0003] The transmission of data is known, by way of example, on the basis of the OSI reference model [1] as described in Andrew S. Tanenbaum, Computer-Netzwerke; Wolfram's Fachverlag, Attenkirchen 1992, pages 17 - 32. The OSI reference model (OSI model for short) from the International Standards Organization (ISO) [comprises] has seven layers, each of which has a different functionality (in terms of abstraction). In the OSI model, layer 1 corresponds to a physical layer, where data and messages are transmitted from the transmitter to the receiver using a physical protocol. In layer 2, a data link layer is provided, and a data link protocol is used for transmission from layer 2 of the transmitter to layer 2 of the receiver. In a similar manner, layer 3, a network layer, uses a network protocol between transmitter and receiver, and layer 4, a transport layer, uses a transport protocol. In the OSI model, layer 5 corresponds to a session layer using a session protocol, layer 6 corresponds to a presentation layer using a presentation protocol, and layer 7 corresponds to an application layer using an application protocol. In practice, many applications do not always permit exact nomination of strict boundaries specifically between the upper protocol layers, particularly layers 5 to 7. By way of example, within the context of Internet telephony, i.e. use of the telephone service using the medium of the Internet, the three layers above the transport layer 4 combine to form an "application layer" for which, in a similar way to in the implementation above, an "application protocol" is used.

[0004] A special feature of the OSI model and hence of all [communications] communication models following the OSI model is that, as a result of the division into layers, in each layer the bottom layers perform functions fully transparently with regard to the currently considered layer and provide this upper layer with a service which is determined by all of the functionalities of the bottom layers. In this context, "transparent" means that currently considered layer does not need to concern itself with the functionalities of the bottom layers. In the case discussed here, the application protocol can thus be used between an application layer transmitter and an application layer receiver. This may cover numerous services, for example for switching, for protection or for actual transmission via a physical channel; these need no longer be a concern from the point of view of the application protocol, particularly as a user of this protocol.

[0005] Similarly with respect to this consideration, the individual layers have "protocol data units" (PDUs) between them which can be designed specifically for each protocol of the respective layer. Thus, for each layer, the respective protocol can comprise a dedicated header containing administrative information for the respective layer's protocol, with this header also being able to be seen and used only by the respective protocol's layer in accordance with the OSI model. Details about the design of the OSI model can be found in numerous pieces of literature, inter alia in those cited above.

[0006] When referring to application layer below, this denotes the layers above the transport layer (layers above 4). The application protocol also denotes the protocol for communication between a transmitter application layer situated above the transport layer and a receiver application layer situated above the transport layer.

[0007] A message authentication code (MAC) is known generally and denotes a cryptographic checksum which is intended to be used to identify an alteration in a message or in data (see [2] Christoph Ruland, Informationssicherheit in Datennetzen (which translates as Information Security in Data Networks), DATACOM-Verlag; Bergheim, 1993, pp.61[ff] - 63 and 68 - 79).

[0008] A one-way hash function is known from [2] Ruland, pp. 68[ff.] - 79, for example. Such a one-way hash function cannot be used to calculate the correct input value for a given function value. Another feature is the absence of collision, i.e. it must not be possible to use the one-way hash function to obtain the same output value for two different input values.

[0009] One difference between the hash code and the MAC is that the MAC requires a secret key in order to calculate it, whereas the hash function can be independent of a key and can be known publicly.

[0010] In addition, an asymmetric cryptographic method (also: public key method) is known, e.g. from [2] Ruland, pp. 73 [ff] - 79. Each party involved in an asymmetric cryptographic method receives two keys, a public key and a secret (or private) key. In principle, the secret key can be derived from the public key, with this task needing to be as complex as possible. The asymmetric method can also be used to produce an electronic signature (authentication) and/or to encrypt the content of a message (using the receiver's public key) such that only the receiver can decrypt the message again using its secret key.

[0011] Finally, there is also a symmetric method, which requires a (secret) key used both for encryption and decryption. An example of a symmetric encryption method is the DES algorithm.

[0012] A [communications] communication system is subject to a multiplicity of possible attacks which can target, inter alia, the content of the messages interchanged in the [communications] communication system or the availability of the [communications] communication system. Taking the example of Internet telephony, then it is firstly important for the content of the telephone conversation not to be able to be monitored by an unauthorized third party, and secondly it is also necessary for the third party not to be able to initiate countless calls, and hence for him not to interfere with the receiver and unnecessarily encumber the [communications] communication system. Such attacks are also called denial-of-service attacks. Possible examples of such attacks are mass data or mass messages which are automatically generated and sent to one or more receivers, where they significantly encumber availability and performance. In the Internet telephony example, it is at least unwanted for an Internet telephone to ring incessantly and hence not only to load the network unnecessarily but also to distract the receiver's attention. It is likewise perceived to be disruptive if an attacker interferes with the flow of communication between transmitter and receiver by sending unauthorized voice data.

[0013] To fend off attacks on a private [communications] communication network, "firewalls" are often used which ensure, in particular, that the private [communications] communication network is separated from public networks. However, it is a simple matter for an attacker to send his meaningless (mass) data to addressees within the private network as well, that is to

say behind the firewall. There, these data are decoded and possibly reproduced. [By this means] Thus, the effect achieved by the attacker is that, by way of example, the reproduction of such meaningless data produces nothing but annoying audio interference or noise and hence restricts the bandwidth and availability of the private network. In the extreme case, part or all of the private [communications] communication network can even crash.

[0014] In addition, an RTP protocol for transmitting media data ("payload"), i.e. video or audio data, is known [3] from H. Schulzrinne: RTP: A Transport Protocol for Real-Time Applications; Internet Engineering Task Force, RFC1889, 1996.

[0015] One approach to holding off unwanted data is provided by the IPSEC protocol [5] described S. Kent, R. Atkinson; IP Authentication Header, Internet Engineering Task Force, RFC2402, 1998 and [6] S. Kent, R. Atkinson; IP Encapsulating Security Payload (ESP), Internet Engineering Task Force, RFC2406, 1998. In this case, data packets of the Internet protocol (IP data packets) can be encapsulated and can be protected in terms of confidentiality and/or integrity (implicit sender authentication). In addition, the IPSEC protocol affords a key management method using a "cookie" mechanism (see [4] D. Harkins, D. Carrel, The Internet Key Exchange (IKE), Internet Engineering Task Force, RFC2409, 1998), which can be used to fend off the mass data attacks (denial-of-service attacks) discussed above during the key management phase. The cookie mechanism involves linking fast one-way hash functions (e.g. SHA-1), random numbers and IP addresses to one another. However, the first "cookie" transmitted from the transmitter to the receiver is not protected, which results in a security gap. In addition, the IPSEC cookie method is not suitable for ensuring protection against such unwanted mass data on the application layer (for the application protocol) under real-time conditions as well. However, the OSI layer structure means that a drawback for IPSEC is, among other things, that it is not possible to link IPSEC to the security functions of an application layer in this manner.

SUMMARY OF THE INVENTION

[0016] [The] An object of the invention is to transmit data from a transmitter to a receiver, with the receiver ensuring that the data are not unwanted data arising from a denial-of-service attack, for example. [This object is achieved in accordance with the features of the independent patent claims.]

[0017] First, the object is achieved by specifying a method for transmitting data from a transmitter to a receiver, in which the transmitter extends the data by authentication data using an application protocol on the application layer. The authentication data are used by the receiver to ascertain whether it knows the transmitter. If the receiver knows the transmitter, the data are accepted, otherwise the data are rejected.

[0018] As stated above, depending on the protocol structure (cf. OSI model), data in each protocol layer have their own transparency with regard to this protocol layer, i.e. the services in protocol layers beneath are performed (without being seen by the current protocol layer). Advantageously, data packets on the application layer, in particular, are free of the administrative data associated with layers beneath, i.e. the data packets in the application layer have, besides the administrative information for the application layer, only the data which are actually to be communicated. Accordingly, authentication in the application layer is particularly advantageous, because the data packets themselves, in contrast to layers beneath, are of significantly reduced size. This small size has an advantageous effect on real time and availability of the entire [communications] communication system. If the receiver knows the transmitter, i.e. particularly if the transmitter is successfully authenticated with respect to the receiver, the message from the transmitter is accepted, in particular the data are stored. Otherwise, i.e. if the receiver does not know the transmitter, the data are rejected, i.e. buffering does not take place. This is particularly advantageous if such a decision is made automatically.

[0019] In particular, available security functions and keys on the application layer can advantageously be allocated to the individual application users. This can advantageously be linked to the application data in a filter function outlined above. Special functions at the application level, such as picture and/or sound compression methods, can advantageously be combined with security functions in the application layer, which makes it possible to increase the end systems' performance and reduces the implementation complexity; this also applies to the aforementioned filtering mechanism.

[0020] The described authentication mechanism in the application layer allows transmitters and receivers which know one another to be separated from transmitters which are not known by the receiver and from which the receiver does not accept any messages either. This allows an attack of the type described in the introduction ("denial-of-service" attack) to be effectively prevented, i.e. unwanted mass data are actually rejected when arriving at the transmitter.

[0021] In this context, special note will be made that the receiver in no way has to be an end receiver or addressee. Instead, the receiver itself can be a switching entity, e.g. a switching node or a firewall with switching functionality, and hence can act with regard to authentication of the transmitter for the end receiver. Thus, in relation to the "firewall" example, it is possible to prevent superfluous network load from being produced by rejecting the unwanted mass data before they even enter the private [communications] communication network.

[0022] In this scenario (authentication in a switching entity), it is advantageous, in particular, if the authentication data are generated using an asymmetric method, since it is also possible for the switching entity, in the example the firewall, to ascertain from whom the data are coming, without the decryption secret already needing to be known in the firewall (as in the case of symmetric encryption) for this purpose. The firewall rejects unwanted mass data if it does not know the transmitter; in the other case, that is to say if the switching entity knows the transmitter and the data are by no means unwanted, these data are forwarded to the receiver after the transmitter has been verified. The receiver can still use its private key to decrypt the data and process them further locally (display them, store them, etc.) irrespective of the check on the data's origin, which has already taken place in the firewall. It will be stressed once again that the decryption (in terms of confidentiality) of the data is performed by the addressee using its private key in the asymmetric method described, whereas the origin of the message (which is determined for this addressee) can also be checked in the switching entity using the public key of the sender – irrespective of the content of the message. Thus, an upstream filter functionality can be provided successfully by the switching entity. This results in a clear functional division between the filter function and the application function; the complexity of the terminals can be simplified as a result and the network traffic in the private [communications] communication network can be reduced.

[0023] One development is that the data are transmitted on a packet-oriented basis. Another development is that the authentication data are determined by virtue of at least part of a protocol data unit, available on the application layer, of the application protocol being encrypted by the transmitter using a secret. In this case, the secret between the transmitter and the receiver can be a key for symmetric encryption or a key pair for asymmetric encryption (public key method, see description with switching entity, for example). Decryption of the secret determines, particularly at the receiver end, whether or not the receiver knows the transmitter. This is advantageous if data encryption is available on the application level anyway and can also be used for this purpose.

[0024] One refinement is that the authentication data are determined using part of the protocol data unit available on the application layer, in particular a sequence number [and/or] or a timestamp [lacuna] [comprises] [lacuna].

[0025] In particular, one development is that a one-way hash function is used for protection. Protection can also be effected using a message authentication code (MAC), with a key which is known only to the transmitter and to the receiver additionally being necessary.

[0026] Another refinement is that, before the data are transmitted from the transmitter to the receiver, authentication is carried out between the transmitter and the receiver.

[0027] Another refinement is that the described method is used in packet-switching telephone services, particularly within the context of Internet telephony. Alternatively, the method can be used in switching nodes and switching installations.

[0028] The object is achieved in another way by specifying a system for transmitting data, in which a transmitter and a receiver are provided, the transmitter extending the data by authentication data using an application protocol on the application layer. The receiver uses the authentication data to check whether it knows the transmitter. If the receiver knows the transmitter, the data from the transmitter are used, otherwise the receiver rejects the data from the transmitter which is unknown to it.

[0029] The object is also achieved by specifying a transmitter for sending data to a receiver, which extends the data by authentication data using an application protocol on the application layer and sends them to the receiver.

[0030] Finally, the object is achieved by specifying a receiver for receiving data from a transmitter designed on the basis of the statements above, in particular, the receiver using authentication data within the application protocol on the application layer to determine whether it knows the transmitter. If it knows the transmitter, it uses the data, otherwise the receiver rejects the data. The system or the transmitter and the receiver are particularly suitable for carrying out the inventive method or one of its developments explained above.

[0031] [Exemplary embodiments are illustrated and explained below with reference to the drawings, in which:] Reference will now be made in detail to the preferred embodiments of the present invention, examples of which are illustrated in the accompanying drawings, wherein like reference numerals refer to like elements throughout.

BRIEF DESCRIPTION OF THE DRAWINGS

[0032] These and other objects and advantages of the present invention will become more apparent and more readily appreciated from the following description of the preferred embodiments, taken in conjunction with the accompanying drawings of which:

[figure] Figure 1 [shows] is a block diagram of a protocol architecture (protocol stack) for a [communications] communication system in the layer model;

[figure] Figure 2 [shows] is a block diagram of an application layer with possible protocols;

[figure] Figure 3 [shows] is a flowchart between the transmitter and the receiver for transmitting data from the transmitter to the receiver;

[figure] Figure 4 [shows] is a protocol data unit (PDU) of the application protocol (layer > 5) with authentication data.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0033] Figure 1 [shows] is a block diagram of a protocol architecture (protocol stack) for a [communications] communication system in the layer model, following the OSI model mentioned [in the introduction] above. [To this end, it shows a] A transmitter 104 and a receiver 105 [which] interchange messages via a physical connection 103. Both the transmitter 104 and the receiver 105 have the same protocol architecture: layer 1 (110 and 111) represents a physical layer, layer 2 (109, 112) represents a data link layer, layer 3 (108, 113) represents a network layer, layer 4 (107, 114) represents a transport layer and layer 5 (101 106) represents an application layer. Between the individual layers there is a respective dedicated protocol from a transmitter 104 to a receiver 105; for the application layer 101 or 106, this is the application protocol 102, for example. From the point of view of layer 5 (101, 106), protocol data units having the format prescribed in the application protocol 102 are transmitted from the transmitter 104 to the receiver 105. The layers situated below the application layer represent functionalities which are provided transparently for the application layer. On the basis of the OSI model, the functionalities have expediently been chosen to be largely independent of one another.

[0034] Figure 2 [shows] is a block diagram of an application layer with possible protocols, based on an application concerning the transmission of media data (audio and video data). Thus, 101 and 106 represent the application layer (layer 5) [already known from figure] shown in Fig. 1 for the transmitter 104 and the receiver 105. This application layer is in turn divided into a

plurality of sublayers. Thus, directly above the layer 4 is the real time transport protocol (RTP) 201. This protocol allows voice and/or picture data to be transmitted in real time from one to possibly a plurality of points (addressees). A layer 206 contains a security application which compiles security functions of the application layer; associated security protocols provide for services such as

- User authentication,
- Access control
- Confidentiality,
- Integrity
- Binding nature of the application data, and
- Accounting.

[0035] Depending on the instance of application, audio data 203 and/or video data 202 are used. For both alternatives, there are different compression standards, a few of which are shown by way of example in [figure] Fig. 2. For [the] audio data, these are the standards G.711, G.722, G.723.1 and G.729. For [the] video data, these are the two picture compression standards H.261 and H.263. In the next abstraction stage (similarly with respect to the OSI model) come the different possibilities for an application of audio or video data after compression and decompression (see block 204). One more abstraction level above, there is, by way of example, a user interface 205, which fully transparently provides a user with the services beneath which have just been described. By way of example, a user, taking into account both the audio and the video data, can thus make use of video telephony, e.g. over the Internet, without specifically needing to concern himself with any details of the services provided by the layers beneath. He thus uses the video telephony service, visible to him, over the Internet transparently. In this context, various alternative implementation forms can be situated below his service.

[0036] Figure 3 [shows] is a flowchart [between the transmitter and the receiver] for transmitting data from the transmitter 308 to the receiver 307. In [a step] block 301, authentication data are produced on the application layer, and the protocol data unit (PDU) relevant to the application protocol is appended. In accordance with the arrangements of the application protocol, the data, including the authentication data, are transmitted to the receiver 307 (see connection 302). This is done using the functionalities or services available under the

application protocol. In [a] block 303, the authentication data are checked in the receiver 307 and there is a test (in [a] block 304) to determine whether the transmitter 308 is known to the receiver 307. If this is not the case, the procedure branches to [a] block 305 and the data are rejected. If the transmitter 308 is known to the receiver 307, the procedure branches to [a] block 306 and the data are processed further. The authentication between transmitter and receiver was successful, and the transmitted data are not unwanted mass data.

[0037] Figure 4 shows a protocol data unit (PDU) of the application protocol with authentication data. What is shown in [figure] Fig. 4 is a protocol data unit (PDU) of the real time transport protocol (RTP, see 201 in [figure] Fig. 2) 401. Such an RTP packet 401 comprises an RTP header 406, encrypted media data 404 and authentication data 405. The RTP header 406 comprises, inter alia, a sequence number 402 and a timestamp 403. Both the transmitter and the receiver know a shared secret K, in this case indicated as a key, which is used to generate a message authentication code (MAC), on the basis of the sequence number 402 and the timestamp 403 (see field 405). An advantage with respect to encryption using a DES algorithm, for example, is a field length of 64 bits.

[0038] As already described, such a packet sent by the transmitter is authenticated at the receiver end such that (in relation to the application layer) a message authentication code is generated by field 402 (sequence number) and field 403 (timestamp) on the basis of the key (K) known to the receiver. If this message authentication code is the same as the field 405, then the arriving RTP packet 401 is a data packet coming from a known transmitter, and the data are processed, for example displayed or stored. In the other case, unauthenticated data are involved, the transmitter is unknown to the receiver and the entire RTP packet is rejected.

[0039] [Literature references]

[1] [Andrew S. Tanenbaum: Computer-Netzwerke;] [Computer networks] [Wolfram's Fachverlag; Attenkirchen 1992; ISBN 3-925328-79-3; pages 17 – 32.]

[2] [Christoph Ruland: Informationssicherheit in Datennetzen;] [Information security in data networks] [DATACOM-Verlag; Bergheim 1993; ISBN 3-89238-081-3; pages 61 – 63 and 68 – 79.]

[3] [RFC1889] [H. Schulzrinne: RTP: A Transport Protocol for Real-Time Applications; Internet Engineering Task Force, 1996]

[4] [RFC2409] [The Internet Key Exchange (IKE), D. Harkins, D. Carrel,; Internet Engineering Task Force, 1998.]

[5] [RFC2402] [IP Authentication Header, S. Kent, R. Atkinson; Internet Engineering Task Force, 1998.]

[6] [RFC2406] [IP Encapsulating Security Payload (ESP), S.Kent, R. Atkinson; Internet Engineering Task Force, 1998.]

[0040] The invention has been described in detail with particular reference to preferred embodiments thereof and examples, but it will be understood that variations and modifications can be effected within the spirit and scope of the invention.

PATENT APPLICATION FEE DETERMINATION RECORD
Effective October 1, 2001

Application or Docket Number: 10/181564

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS		
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	28 minus 20=	* 08
INDEPENDENT CLAIMS	4 minus 3 =	* 1
MULTIPLE DEPENDENT CLAIM PRESENT		☐

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE ☐		OR	OTHER THAN SMALL ENTITY	
RATE	FEE		RATE	FEE
BASIC FEE	370.00	OR	BASIC FEE	~~740.00~~ 840
X$ 9=		OR	X$18=	144
X42=		OR	X84=	84
+140=		OR	+280=	
TOTAL		OR	TOTAL	

CLAIMS AS AMENDED - PART II

2-6-06

AMENDMENT A

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 29	Minus	** 28	= 1
Independent	* 5	Minus	*** 4	= 1
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$~~18~~ 50 =	50
X42=		OR	X~~84~~ 200 =	200
+140=		OR	+280=	0
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	250

AMENDMENT B

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/181,564	07/18/2002	Martin Euchner	1454.1302	4619

21171 7590 04/28/2006

STAAS & HALSEY LLP
SUITE 700
1201 NEW YORK AVENUE, N.W.
WASHINGTON, DC 20005

EXAMINER	
TRAN, TONGOC	

ART UNIT	PAPER NUMBER
2134	

DATE MAILED: 04/28/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/181,564	EUCHNER, MARTIN	
	Examiner	Art Unit	
	Tongoc Tran	2134	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *06 February 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *17-45* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-45* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

U.S. Patent and Trademark Office
PTOL-326 (Rev. 7-05) Office Action Summary Part of Paper No./Mail Date 20060426

DETAILED ACTION

1. This Office Action is in response to Applicant's Amendment filed on 2/6/2006. Claims 1-16 were previously canceled and claims 17-44 were added by preliminary amendment filed on 7/18/2002. Examiner regrets that the original claims 1-16 were examined instead of the preliminary amendment claims 17-44. Claims 17-44 are amended and claim 45 is added on 2/6/2006 and are pending for examination. Substitute of Specification and Abstract submitted on 7/18/2002 and resubmitted on 2/6/2006 have been entered.

Response to Arguments

2. Applicant's arguments with respect to amended claims 17-45 have been considered but are moot in view of the new ground(s) of rejection.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 17-45 are rejected under 35 U.S.C. 102(e) as being anticipated by Sengodan et al. (U.S. Patent No. 6,918,034, hereinafter Sengodan).

In respect to claims 17, 24-25, 32-33, 35-36 and 38-45, Sengodan discloses a method for transmitting data from a transmitter to a receiver, comprising:

Extending the data in the transmitter by authentication data using an application protocol on an application layer (Abstract, Fig. 6, col. 5, lines 44-47);

Ascertaining, by the receiver, whether the receiver knows the transmitter based on the authentication data; and accepting the data at the receiver, if the receiver knows the transmitter based on the authentication data wherein the application protocol comprises a Real Time Protocol (RTP) and the authentication data is inserted at end of an RTP payload Wherein the application protocol comprising: a sequence number or timestamp; generating a message authentication code (MAC) and verifying (see col. 6, lines 47-55 and col. 8, lines 22-37).

In respect to claim 18, Sengodan discloses wherein the data are transmitted on a packet-oriented basis (see col. 1, lines 9-13).

In respect to claim 19, Sengodan discloses wherein the authentication data are determined based on at least part of a protocol data unit available on the application layer, and the application protocol being linked to a secret by the transmitter (see col. 8, lines 22-37, HMAC-SHA1).

particularly by virtue of an encryption operation and a cryptographic checksum and key information (see col. 8, lines 22-37).

In respect to claim 21, Sengodan discloses wherein said ascertaining includes verification at the receiver to determine whether the receiver knows the transmitter (see col.8, lines 31-37).

In respect to claim 22, Sengodan discloses wherein the verification is cryptographic verification using parameter and including at least one of a decryption operation and a cryptographic checksum check (see col. 8, lines 22-37).

In respect to claims 23, 31, 34 and 37, Sengodan discloses wherein the secret between the transmitter and the receiver is one of a key for symmetric encryption and a key pair for asymmetric encryption (see col. 3, lines 54-63).

In respect to claim 26, Sengodan discloses wherein the authentication is carried out between the transmitter and the receiver (see Col. 8, lines 22-37).

In respect to claim 27, Sengodan discloses wherein the receiver and transmitter provide packet-switched telephone services (see col. 2, lines 39-50).

In respect to claims 28 and 30, Sengodan discloses wherein the receiver and transmitter are disposed in at least one of a switching node and a switching installation (see col. 2, lines 39-50).

In respect to claim 29, Sengodan discloses wherein the receiver and transmitter are used in Internet telephony (see col. 2, lines 39-50).

Conclusion

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tongoc Tran whose telephone number is (571) 272-3843. The examiner can normally be reached on 8:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Jacques Louis-Jacques can be reached on (571) 272-3962. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Examiner: Tongoc Tran
Art Unit: 2134

April 26, 2006

Notice of References Cited	Application/Control No. 10/181,564	Applicant(s)/Patent Under Reexamination EUCHNER, MARTIN	
	Examiner Tongoc Tran	Art Unit 2134	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-6,918,034	07-2005	Sengodan et al.	713/160
*	B	US-6,229,821	05-2001	Bharucha et al.	370/471
*	C	US-6,396,840	05-2002	Rose et al.	370/401
*	D	US-4,375,097	02-1983	Ulug, Mehmet E.	370/400
*	E	US-6,327,660	12-2001	Patel, Baiju V.	713/193
*	F	US-5,917,830	06-1999	Chen et al.	370/487
*	G	US-5,345,507	09-1994	Herzberg et al.	380/28
*	H	US-6,320,869	11-2001	Van Driel et al.	370/443
*	I	US-6,366,961	04-2002	Subbiah et al.	709/238
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	Perkins et al., "RTP Payload for Redundant Audio Data", Network Working Group, September 1997. pages 1-10.
	V	Http://msdn2.microsoft.com/en-us/library/system.security.cryptography.hmacsha1; 2006.
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Network Working Group
Request for Comments: 2198
Category: Standards Track

C. Perkins
I. Kouvelas
O. Hodson
V. Hardman
University College London
M. Handley
ISI
J.C. Bolot
A. Vega-Garcia
S. Fosse-Parisis
INRIA Sophia Antipolis

September 1997

RTP Payload for Redundant Audio Data

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Abstract

This document describes a payload format for use with the real-time transport protocol (RTP), version 2, for encoding redundant audio data. The primary motivation for the scheme described herein is the development of audio conferencing tools for use with lossy packet networks such as the Internet Mbone, although this scheme is not limited to such applications.

1 Introduction

If multimedia conferencing is to become widely used by the Internet Mbone community, users must perceive the quality to be sufficiently good for most applications. We have identified a number of problems which impair the quality of conferences, the most significant of which is packet loss. This is a persistent problem, particularly given the increasing popularity, and therefore increasing load, of the Internet. The disruption of speech intelligibility even at low loss rates which is currently experienced may convince a whole generation of users that multimedia conferencing over the Internet is not viable. The addition of redundancy to the data stream is offered as a solution [1]. If a packet is lost then the missing information may be reconstructed at the receiver from the redundant data that arrives in the following packet(s), provided that the average number

of consecutively lost packets is small. Recent work [4,5] shows that packet loss patterns in the Internet are such that this scheme

typically functions well.

This document describes an RTP payload format for the transmission of audio data encoded in such a redundant fashion. Section 2 presents the requirements and motivation leading to the definition of this payload format, and does not form part of the payload format definition. Sections 3 onwards define the RTP payload format for redundant audio data.

2 Requirements/Motivation

The requirements for a redundant encoding scheme under RTP are as follows:

- o Packets have to carry a primary encoding and one or more redundant encodings.
- o As a multitude of encodings may be used for redundant information, each block of redundant encoding has to have an encoding type identifier.
- o As the use of variable size encodings is desirable, each encoded block in the packet has to have a length indicator.
- o The RTP header provides a timestamp field that corresponds to the time of creation of the encoded data. When redundant encodings are used this timestamp field can refer to the time of creation of the primary encoding data. Redundant blocks of data will correspond to different time intervals than the primary data, and hence each block of redundant encoding will require its own timestamp. To reduce the number of bytes needed to carry the timestamp, it can be encoded as the difference of the timestamp for the redundant encoding and the timestamp of the primary.

There are two essential means by which redundant audio may be added to the standard RTP specification: a header extension may hold the redundancy, or one, or more, additional payload types may be defined.

Including all the redundancy information for a packet in a header extension would make it easy for applications that do not implement redundancy to discard it and just process the primary encoding data. There are, however, a number of disadvantages with this scheme:

- o There is a large overhead from the number of bytes needed for the extension header (4) and the possible padding that is needed at the end of the extension to round up to a four byte boundary (up to 3 bytes). For many applications this overhead is unacceptable.
- o Use of the header extension limits applications to a single redundant encoding, unless further structure is introduced into the extension. This would result in further overhead.

For these reasons, the use of RTP header extension to hold redundant

audio encodings is disregarded.

The RTP profile for audio and video conferences [3] lists a set of payload types and provides for a dynamic range of 32 encodings that may be defined through a conference control protocol. This leads to two possible schemes for assigning additional RTP payload types for redundant audio applications:

1. A dynamic encoding scheme may be defined, for each combination of primary/redundant payload types, using the RTP dynamic payload type range.

2. A single fixed payload type may be defined to represent a packet with redundancy. This may then be assigned to either a static RTP payload type, or the payload type for this may be assigned dynamically.

It is possible to define a set of payload types that signify a particular combination of primary and secondary encodings for each of the 32 dynamic payload types provided. This would be a slightly restrictive yet feasible solution for packets with a single block of redundancy as the number of possible combinations is not too large. However the need for multiple blocks of redundancy greatly increases the number of encoding combinations and makes this solution not viable.

A modified version of the above solution could be to decide prior to the beginning of a conference on a set a 32 encoding combinations that will be used for the duration of the conference. All tools in the conference can be initialized with this working set of encoding combinations. Communication of the working set could be made through the use of an external, out of band, mechanism. Setup is complicated as great care needs to be taken in starting tools with identical parameters. This scheme is more efficient as only one byte is used to identify combinations of encodings.

It is felt that the complication inherent in distributing the mapping of payload types onto combinations of redundant data preclude the use of this mechanism.

A more flexible solution is to have a single payload type which signifies a packet with redundancy. That packet then becomes a container, encapsulating multiple payloads into a single RTP packet. Such a scheme is flexible, since any amount of redundancy may be encapsulated within a single packet. There is, however, a small overhead since each encapsulated payload must be preceded by a header indicating the type of data enclosed. This is the preferred solution, since it is both flexible, extensible, and has a relatively low overhead. The remainder of this document describes this solution.

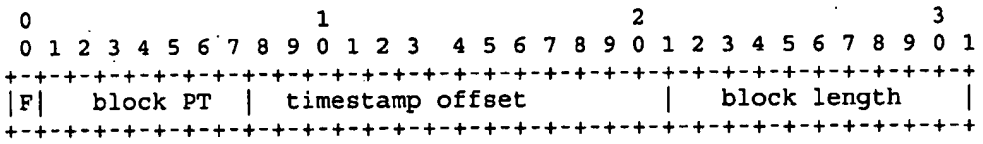
3 Payload Format Specification

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [7].

The assignment of an RTP payload type for this new packet format is outside the scope of this document, and will not be specified here. It is expected that the RTP profile for a particular class of applications will assign a payload type for this encoding, or if that is not done then a payload type in the dynamic range shall be chosen.

An RTP packet containing redundant data shall have a standard RTP header, with payload type indicating redundancy. The other fields of the RTP header relate to the primary data block of the redundant data.

Following the RTP header are a number of additional headers, defined in the figure below, which specify the contents of each of the encodings carried by the packet. Following these additional headers are a number of data blocks, which contain the standard RTP payload data for these encodings. It is noted that all the headers are aligned to a 32 bit boundary, but that the payload data will typically not be aligned. If multiple redundant encodings are carried in a packet, they should correspond to different time intervals: there is no reason to include multiple copies of data for a single time interval within a packet.



The bits in the header are specified as follows:

F: 1 bit First bit in header indicates whether another header block follows. If 1 further header blocks follow, if 0 this is the last header block.

block PT: 7 bits RTP payload type for this block.

timestamp offset: 14 bits Unsigned offset of timestamp of this block relative to timestamp given in RTP header. The use of an unsigned offset implies that redundant data must be sent after the primary data, and is hence a time to be subtracted from the current timestamp to determine the timestamp of the data for which this block is the redundancy.

block length: 10 bits Length in bytes of the corresponding data block excluding header.

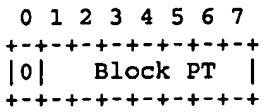
It is noted that the use of an unsigned timestamp offset limits the use of redundant data slightly: it is not possible to send redundancy before the primary encoding. This may affect schemes where a low bandwidth coding suitable for redundancy is produced early in the encoding process, and hence could feasibly be transmitted early. However, the addition of a sign bit would unacceptably reduce the range of the timestamp offset, and increasing the size of the field above 14 bits limits the block length field. It seems that limiting redundancy to be transmitted after the primary will cause fewer problems than limiting the size of the other fields.

The timestamp offset for a redundant block is measured in the same units as the timestamp of the primary encoding (ie: audio samples, with the same clock rate as the primary). The implication of this is that the redundant encoding MUST be sampled at the same rate as the primary.

It is further noted that the block length and timestamp offset are 10 bits, and 14 bits respectively; rather than the more obvious 8 and 16 bits. Whilst such an encoding complicates parsing the header information slightly, and adds some additional processing overhead, there are a number of problems involved with the more obvious choice: An 8 bit block length field is sufficient for most, but not all, possible encodings: for example 80ms PCM and DVI audio packets comprise more than 256 bytes, and cannot be encoded with a single byte length field. It is possible to impose additional structure on the block length field (for example the high bit set could imply the lower 7 bits code a length in words, rather than bytes), however such schemes are complex. The use of a 10 bit block length field retains

simplicity and provides an enlarged range, at the expense of a reduced range of timestamp values.

The primary encoding block header is placed last in the packet. It is therefore possible to omit the timestamp and block-length fields from the header of this block, since they may be determined from the RTP header and overall packet length. The header for the primary (final) block comprises only a zero F bit, and the block payload type information, a total of 8 bits. This is illustrated in the figure below:



The final header is followed, immediately, by the data blocks, stored in the same order as the headers. There is no padding or other delimiter between the data blocks, and they are typically not 32 bit aligned. Again, this choice was made to reduce bandwidth overheads, at the expense of additional decoding time.

The choice of encodings used should reflect the bandwidth requirements of those encodings. It is expected that the redundant encoding shall use significantly less bandwidth that the primary encoding: the exception being the case where the primary is very low-bandwidth and has high processing requirement, in which case a copy of the primary MAY be used as the redundancy. The redundant encoding MUST NOT be higher bandwidth than the primary.

The use of multiple levels of redundancy is rarely necessary. However, in those cases which require it, the bandwidth required by each level of redundancy is expected to be significantly less than that of the previous level.

4 Limitations

The RTP marker bit is not preserved for redundant data blocks. Hence

if the primary (containing this marker) is lost, the marker is lost. It is believed that this will not cause undue problems: even if the marker bit was transmitted with the redundant information, there would still be the possibility of its loss, so applications would still have to be written with this in mind.

In addition, CSRC information is not preserved for redundant data. The CSRC data in the RTP header of a redundant audio packet relates to the primary only. Since CSRC data in an audio stream is expected to change relatively infrequently, it is recommended that

applications which require this information assume that the CSRC data in the RTP header may be applied to the reconstructed redundant data.

5 Relation to SDP

When a redundant payload is used, it may need to be bound to an RTP dynamic payload type. This may be achieved through any out-of-band mechanism, but one common way is to communicate this binding using the Session Description Protocol (SDP) [6]. SDP has a mechanism for binding a dynamic payload types to particular codec, sample rate, and number of channels using the "rtpmap" attribute. An example of its use (using the RTP audio/video profile [3]) is:

```
m=audio 12345 RTP/AVP 121 0 5
a=rtpmap:121 red/8000/1
```

This specifies that an audio stream using RTP is using payload types 121 (a dynamic payload type), 0 (PCM u-law) and 5 (DVI). The "rtpmap" attribute is used to bind payload type 121 to codec "red" indicating this codec is actually a redundancy frame, 8KHz, and monaural. When used with SDP, the term "red" is used to indicate the redundancy format discussed in this document.

In this case the additional formats of PCM and DVI are specified. The receiver must therefore be prepared to use these formats. Such a specification means the sender will send redundancy by default, but also may send PCM or DVI. However, with a redundant payload we additionally take this to mean that no codec other than PCM or DVI will be used in the redundant encodings. Note that the additional payload formats defined in the "m=" field may themselves be dynamic payload types, and if so a number of additional "a=" attributes may be required to describe these dynamic payload types.

To receive a redundant stream, this is all that is required. However to send a redundant stream, the sender needs to know which codecs are recommended for the primary and secondary (and tertiary, etc) encodings. This information is specific to the redundancy format, and is specified using an additional attribute "fmtp" which conveys format-specific information. A session directory does not parse the values specified in an fmtp attribute but merely hands it to the media tool unchanged. For redundancy, we define the format parameters to be a slash "/" separated list of RTP payload types.

Thus a complete example is:

```
m=audio 12345 RTP/AVP 121 0 5
a=rtpmap:121 red/8000/1
```

```
a=fmtp:121 0/5
```


This specifies that the default format for senders is redundancy with PCM as the primary encoding and DVI as the secondary encoding. Encodings cannot be specified in the fmtp attribute unless they are also specified as valid encodings on the media ("m=") line.

6 Security Considerations

RTP packets containing redundant information are subject to the security considerations discussed in the RTP specification [2], and any appropriate RTP profile (for example [3]). This implies that confidentiality of the media streams is achieved by encryption. Encryption of a redundant data stream may occur in two ways:

1. The entire stream is to be secured, and all participants are expected to have keys to decode the entire stream. In this case, nothing special need be done, and encryption is performed in the usual manner.

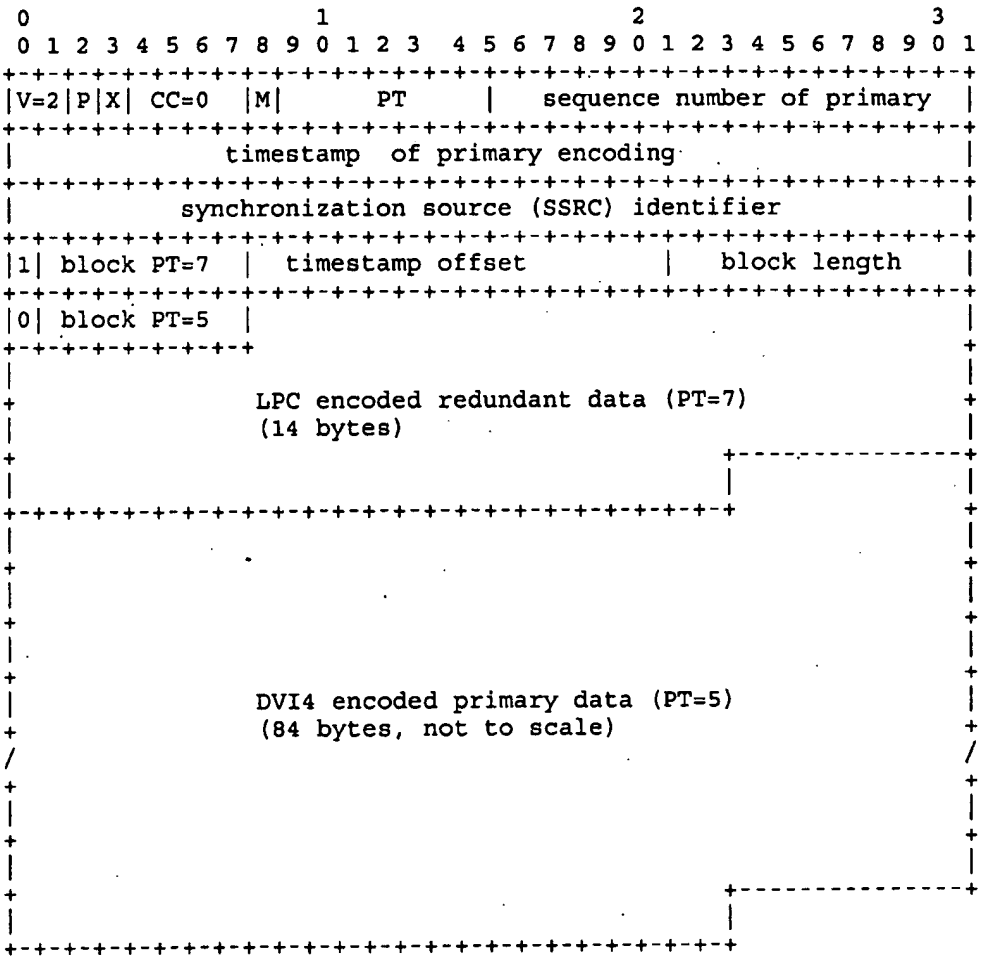
2. A portion of the stream is to be encrypted with a different key to the remainder. In this case a redundant copy of the last packet of that portion cannot be sent, since there is no following packet which is encrypted with the correct key in which to send it. Similar limitations may occur when enabling/disabling encryption.

The choice between these two is a matter for the encoder only. Decoders can decrypt either form without modification.

Whilst the addition of low-bandwidth redundancy to an audio stream is an effective means by which that stream may be protected against packet loss, application designers should be aware that the addition of large amounts of redundancy will increase network congestion, and hence packet loss, leading to a worsening of the problem which the use of redundancy was intended to solve. At its worst, this can lead to excessive network congestion and may constitute a denial of service attack.

7 Example Packet

An RTP audio data packet containing a DVI4 (8KHz) primary, and a single block of redundancy encoded using 8KHz LPC (both 20ms packets), as defined in the RTP audio/video profile [3] is illustrated:



8 Authors' Addresses

Colin Perkins/Isidor Kouvelas/Orion Hodson/Vicky Hardman
Department of Computer Science
University College London
London WC1E 6BT
United Kingdom

EMail: {c.perkins|i.kouvelas|o.hodson|v.hardman}@cs.ucl.ac.uk

Mark Handley
USC Information Sciences Institute
c/o MIT Laboratory for Computer Science
545 Technology Square
Cambridge, MA 02139, USA

EMail: mjh@isi.edu

Jean-Chrysostome Bolot/Andres Vega-Garcia/Sacha Fosse-Parisis
INRIA Sophia Antipolis
2004 Route des Lucioles, BP 93
06902 Sophia Antipolis
France

EMail: {bolot|avega|sfosse}@sophia.inria.fr

9 References

[1] V.J. Hardman, M.A. Sasse, M. Handley and A. Watson; Reliable Audio for Use over the Internet; Proceedings INET'95, Honalulu, Oahu, Hawaii, September 1995. http://www.isoc.org/in95prc/

[2] Schulzrinne, H., Casner, S., Frederick R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC1889 January 1996.

[3] Schulzrinne, H., "RTP Profile for Audio and Video Conferences with Minimal Control", RFC1890 January 1996.

[4] M. Yajnik, J. Kurose and D. Towsley; Packet loss correlation in the MBone multicast network; IEEE Globecom Internet workshop, London, November 1996

[5] J.-C. Bolot and A. Vega-Garcia; The case for FEC-based error

control for packet audio in the Internet; ACM Multimedia Systems, 1997

[6] Handley, M., and V. Jacobson, "SDP: Session Description Protocol (draft 03.2)", Work in Progress.

[7] Bradner, S., "Key words for use in RFCs to indicate requirement levels", RFC2119 March 1997.

Meta 1002
p. 543

msdn

©2006 Microsoft Corporation. All rights reserved.

.NET Framework Class Library

HMACSHA1 Class

Computes a Hash-based Message Authentication Code (HMAC) using the SHA1 [http://msdn2.microsoft.com/en-us/library/system.security.cryptography.sha1.aspx] hash function.

Namespace: System.Security.Cryptography
Assembly: mscorlib (in mscorlib.dll)

Syntax

Visual Basic (Declaration) — Copy Code

```
<ComVisibleAttribute(True)> _
Public Class HMACSHA1
    Inherits HMAC
```

Visual Basic (Usage) — Copy Code

```
Dim instance As HMACSHA1
```

C# — Copy Code

```
[ComVisibleAttribute(true)]
public class HMACSHA1 : HMAC
```

C++ — Copy Code

```
[ComVisibleAttribute(true)]
public ref class HMACSHA1 : public HMAC
```

J# — Copy Code

```
/** @attribute ComVisibleAttribute(true) */
public class HMACSHA1 extends HMAC
```

JScript — Copy Code

```
ComVisibleAttribute(true)
public class HMACSHA1 extends HMAC
```

Remarks

HMACSHA1 is a type of keyed hash algorithm that is constructed from the SHA1 hash function and used as an HMAC, or hash-based message authentication code. The HMAC process mixes a secret key with the message data, hashes the result with the hash function, mixes that hash value with the secret key again, then applies the hash function a second time. The output hash is 160 bits in length.

An HMAC can be used to determine whether a message sent over an insecure channel has been tampered with, provided that the sender and receiver share a secret key. The sender computes the hash value for the original data and sends both the original data and hash value as a single message. The receiver recalculates the hash value on the received message and checks that the computed HMAC matches the transmitted HMAC.

Any change to the data or the hash value will result in a mismatch, because knowledge of the secret key is required to change the message and reproduce the correct hash value. Therefore, if the original and computed hash values match, the message is authenticated.

The SHA-1 (Secure Hash Algorithm, also called SHS, Secure Hash Standard) is a cryptographic hash algorithm published by the United States Government. It produces a 160-bit hash value from an arbitrary length string.

HMACSHA1 accepts keys of any size, and produces a hash sequence of length 160 bits.

this page

Example

The following code example shows how to encode a file using **HMACSHA1** and then how to decode the file.

C# Copy Code

```
using System;
using System.IO;
using System.Security.Cryptography;

public class HMACSHA1example
{
    // Computes a keyed hash for a source file, creates a target file with the keyed hash
    // prepended to the contents of the source file, then decrypts the file and compares
    // the source and the decrypted files.
    public static void EncodeFile(byte[] key, String sourceFile, String destFile)
    {
        // Initialize the keyed hash object.
        HMACSHA1 myhmacsha1 = new HMACSHA1(key);
        FileStream inStream = new FileStream(sourceFile, FileMode.Open);
        FileStream outStream = new FileStream(destFile, FileMode.Create);
        // Compute the hash of the input file.
        byte[] hashValue = myhmacsha1.ComputeHash(inStream);
        // Reset inStream to the beginning of the file.
        inStream.Position = 0;
        // Write the computed hash value to the output file.
        outStream.Write(hashValue, 0, hashValue.Length);
        // Copy the contents of the sourceFile to the destFile.
        int bytesRead;
        // read 1K at a time
        byte[] buffer = new byte[1024];
        do
        {
            // Read from the wrapping CryptoStream.
            bytesRead = inStream.Read(buffer,0,1024);
            outStream.Write(buffer, 0, bytesRead);
        } while (bytesRead > 0);
        myhmacsha1.Clear();
        // Close the streams
        inStream.Close();
        outStream.Close();
        return;
    } // end EncodeFile


    // Decrypt the encoded file and compare to original file.
    public static bool DecodeFile(byte[] key, String sourceFile)
    {
        // Initialize the keyed hash object.
        HMACSHA1 hmacsha1 = new HMACSHA1(key);
        // Create an array to hold the keyed hash value read from the file.
        byte[] storedHash = new byte[hmacsha1.HashSize/8];
        // Create a FileStream for the source file.
        FileStream inStream = new FileStream(sourceFile, FileMode.Open);
        // Read in the storedHash.
        inStream.Read(storedHash, 0, storedHash.Length);
        // Compute the hash of the remaining contents of the file.
        // The stream is properly positioned at the beginning of the content,
        // immediately after the stored hash value.
        byte[] computedHash = hmacsha1.ComputeHash(inStream);
        // compare the computed hash with the stored value
        for (int i =0; i < storedHash.Length; i++)
        {
            if (computedHash[i] != storedHash[i])
            {
                Console.WriteLine("Hash values differ! Encoded file has been tampered with!
                return false;
            }
        }
        Console.WriteLine("Hash values agree -- no tampering occurred.");
        return true;
    } //end DecodeFile

    private const string usageText = "Usage: HMACSHA1 inputfile.txt encryptedfile.hsh\nYou
    public static void Main(string[] Fileargs)
    {
        //If no file names are specified, write usage text.
        if (Fileargs.Length < 2)
        {
```

```
            Console.WriteLine(usageText);
        }
        else
        {
            try
            {
                // Create a random key using a random number generator. This would be the
                //  secret key shared by sender and receiver.
                byte[] secretkey = new Byte[64];
                //RNGCryptoServiceProvider is an implementation of a random number generatc
                RNGCryptoServiceProvider rng = new RNGCryptoServiceProvider();
                // The array is now filled with cryptographically strong random bytes.
                rng.GetBytes(secretkey);

                // Use the secret key to encode the message file.
                EncodeFile(secretkey, Fileargs[0], Fileargs[1]);

                // Take the encoded file and decode
                DecodeFile(secretkey, Fileargs[1]);
            }
            catch (IOException e)
            {
                Console.WriteLine("Error: File not found",e);
            }
        } //end if-else

    }  //end main
} //end class
```

C++ Copy Code

```
using namespace System;
using namespace System::IO;
using namespace System::Security::Cryptography;

// Computes a keyed hash for a source file, creates a target file with the keyed hash
// prepended to the contents of the source file, then decrypts the file and compares
// the source and the decrypted files.
void EncodeFile( array<Byte>^key, String^ sourceFile, String^ destFile )
{

   // Initialize the keyed hash object.
   HMACSHA1^ myhmacsha1 = gcnew HMACSHA1( key );
   FileStream^ inStream = gcnew FileStream( sourceFile,FileMode::Open );
   FileStream^ outStream = gcnew FileStream( destFile,FileMode::Create );

   // Compute the hash of the input file.
   array<Byte>^hashValue = myhmacsha1->ComputeHash( inStream );

   // Reset inStream to the beginning of the file.
   inStream->Position = 0;

   // Write the computed hash value to the output file.
   outStream->Write( hashValue, 0, hashValue->Length );

   // Copy the contents of the sourceFile to the destFile.
   int bytesRead;

   // read 1K at a time
   array<Byte>^buffer = gcnew array<Byte>(1024);
   do
   {

      // Read from the wrapping CryptoStream.
      bytesRead = inStream->Read( buffer, 0, 1024 );
      outStream->Write( buffer, 0, bytesRead );
   }
   while ( bytesRead > 0 );

   myhmacsha1->Clear();

   // Close the streams
   inStream->Close();
   outStream->Close();
   return;
} // end EncodeFile


// Decrypt the encoded file and compare to original file.
```

```
bool DecodeFile( array<Byte>^key, String^ sourceFile )
{

   // Initialize the keyed hash object.
   HMACSHA1^ hmacsha1 = gcnew HMACSHA1( key );

   // Create an array to hold the keyed hash value read from the file.
   array<Byte>^storedHash = gcnew array<Byte>(hmacsha1->HashSize / 8);

   // Create a FileStream for the source file.
   FileStream^ inStream = gcnew FileStream( sourceFile,FileMode::Open );

   // Read in the storedHash.
   inStream->Read( storedHash, 0, storedHash->Length );

   // Compute the hash of the remaining contents of the file.
   // The stream is properly positioned at the beginning of the content,
   // immediately after the stored hash value.
   array<Byte>^computedHash = hmacsha1->ComputeHash( inStream );

   // compare the computed hash with the stored value
   for ( int i = 0; i < storedHash->Length; i++ )
   {
      if ( computedHash[ i ] != storedHash[ i ] )
      {
         Console::WriteLine( "Hash values differ! Encoded file has been tampered with!" );
         return false;
      }

   }
   Console::WriteLine( "Hash values agree -- no tampering occurred." );
   return true;
} //end DecodeFile


int main()
{
   array<String^>^Fileargs = Environment::GetCommandLineArgs();
   String^ usageText = "Usage: HMACSHA1 inputfile.txt encryptedfile.hsh\nYou must specify t

   //If no file names are specified, write usage text.
   if ( Fileargs->Length < 3 )
   {
      Console::WriteLine( usageText );
   }
   else
   {
      try
      {

         // Create a random key using a random number generator. This would be the
         //  secret key shared by sender and receiver.
         array<Byte>^secretkey = gcnew array<Byte>(64);

         //RNGCryptoServiceProvider is an implementation of a random number generator.
         RNGCryptoServiceProvider^ rng = gcnew RNGCryptoServiceProvider;

         // The array is now filled with cryptographically strong random bytes.
         rng->GetBytes( secretkey );

         // Use the secret key to encode the message file.
         EncodeFile( secretkey, Fileargs[ 1 ], Fileargs[ 2 ] );

         // Take the encoded file and decode
         DecodeFile( secretkey, Fileargs[ 2 ] );
      }
      catch ( IOException^ e )
      {
         Console::WriteLine( "Error: File not found", e );
      }

   }
} //end main
```

J# Copy Code

```
import System.*;
import System.IO.*;
```