



US 20060059347A1

(19) **United States**

(12) **Patent Application Publication**
Herz et al.

(10) **Pub. No.: US 2006/0059347 A1**

(43) **Pub. Date: Mar. 16, 2006**

(54) **SYSTEM AND METHOD WHICH EMPLOYS
A MULTI USER SECURE SCHEME
UTILIZING SHARED KEYS**

on Jul. 1, 2002. Provisional application No. 60/373,405, filed on Apr. 18, 2002.

(76) **Inventors: Frederick S.M. Herz**, Warrington, PA (US); **Yael Gertner**, Wayne, NJ (US); **Craig Martell**, Philadelphia, PA (US); **Sampath Kannan**, Philadelphia, PA (US)

Correspondence Address:

**WOODCOCK WASHBURN LLP
ONE LIBERTY PLACE, 46TH FLOOR
1650 MARKET STREET
PHILADELPHIA, PA 19103 (US)**

(21) **Appl. No.: 10/418,983**

(22) **Filed: Apr. 18, 2003**

Related U.S. Application Data

(60) Provisional application No. 60/444,743, filed on Feb. 4, 2003. Provisional application No. 60/392,253, filed

Publication Classification

(51) **Int. Cl.**
H04L 9/00 (2006.01)

(52) **U.S. Cl.** **713/176**

(57) **ABSTRACT**

We propose a multi user information theoretically secure scheme. Our scheme allows any two parties in a multi user system to exchange messages securely using encryption, and to sign messages. Our scheme achieves a significant saving in the number of total keys in the system and in the keys each user must store. The encryption, and signing algorithms proposed in the scheme are as efficient as possible. Our scheme is designed so that it is possible to easily and efficiently revoke and add membership of new users into the system. It is also designed so that authentication and security against man in the middle attacks can be added at low cost. In addition, we introduce a novel and efficient way to use steganography for key replenishment.

SYSTEM AND METHOD WHICH EMPLOYS A MULTI USER SECURE SCHEME UTILIZING SHARED KEYS

[0001] (Conversion of Provisional Applications Nos. 60,444,743, 60/392,253, and 60/373,405).

CROSS REFERENCE TO RELATED APPLICATIONS

[0002] William Arbaugh, David J. Farber, Angelos Keromytis, Jonathan M. Smith, U.S. Pat. No. 6,185,678, Feb. 6, 2001, "Secure and Reliable Bootstrap Architecture".

REFERENCE TO SEQUENCE LISTING, A TABLE

- [0003] Descriptive title of the invention page—page 1
- [0004] Cross Reference to Related Applications—page 1
- [0005] Field of the Invention—page—2
- [0006] Background of the Invention—page—2
- [0007] Brief Summary of the Invention—page—2
- [0008] Detailed Description—page—11
- [0009] Abstract—on a separate sheet of paper
- [0010] Claim—on a separate sheet of paper

FIELD OF THE INVENTION

[0011] The present novel disclosed system and method relates to the broad fields of art of secure communications, system security and cryptography, in general. A particular emphasis in this writing has placed on the application space of these fields which (at the time of the writing of this patent application) was substantively occupied by cryptographic solutions based upon Public Key Infrastructure (Pki) based methods.

BRIEF SUMMARY OF THE INVENTION

[0012] A system and method is herein proposed in which its preferred embodiment is described. The design of this preferred cryptographic scheme is based upon the use of random data strings for coding communications and it is thus capable of achieving an equivalency of data security which is theoretically unsurpassable and thus equal to that of a one time pad (i.e., the term herein used for this level of "information theoretic" security refers to the complete absence of data by a would-be interceptor). In addition, it requires an optimally minimal amount of computational overhead. Memory requirements associated with key distribution and replenishment are minimized as a consequence of the scheme's ability to utilize completely random strings for the message coding as a result of being able to maximize the quantity of the associated data, which constitute the key sets. This advantage is provided through a novel means through the sharing of its associated digital components between the various different keys residing with any given user in the community of potential correspondents. Some of the significant system advantages include efficiency in key revocation and an increasingly diminished risk of vulnerability to unauthorized key seizure by any potential rogue attempts on the part of colluding members (or previously revoked members) in direct proportion with the total size of the community. In this work (at a summary level) we explore the

general problem of data and data communication security within the context of addressing and provide a system and method to address certain important concerns and issues based upon the following system level objectives:

[0013] 1. Maximizing the inherent security of the cryptographic method which is information theoretic secure and thus meets the criteria for the strongest possible security theoretically achievable (equivalent to that of one time pad methods);

[0014] 2. Minimizing overhead costs such as computational overhead for which our preferred scheme offers the lowest possible computation requirements;

[0015] 3. Minimal storage of keys so as to maintain memory requirements within manageable limits. Accordingly our preferred scheme minimally reduces the size of the random number of keys (or pads) by sharing the data inherently contained within each unique key used to communicate with each unique member of a community. Thus the term "shared set keys" is applied.

[0016] 4. To minimize the inefficiency associated with key revocation. Unlike traditional one-time pads our preferred scheme enables each member who wishes to cancel the community membership of a revoked user to simply revoke the key for that user on his or her system.

[0017] 5. To minimize the likelihood of risk of collusion by rogue attempts of members (or previously revoked members) to attempt to combine their keys in order to identify (and thus break) the key pairs associated with those of other communicating members who would otherwise desire to communicate privately;

[0018] 6. To achieve information theoretic secure authentication and digital signatures as would be achievable using similar advantages and efficiencies associated with the preferred cryptographic scheme;

[0019] 7. To enable communities of users in which any user is able to very flexibly and efficiently communicate with any other user, any combination of users, or potentially all users in the community such that the establishment of the desired communicating members is absolutely secure with respect to those desired members;

[0020] 8. To provide a scheme which possesses substantial redundancy by virtue of its highly distributed nature and in this way is minimally vulnerable to any single point of attack whether it be physical or within the confines of the network environment;

[0021] 9. To provide efficient means for replenishing used keys. Although a method using steganographic techniques is provided in detail and possesses distinct advantages, any number of other possible techniques may exist each with a potentially unique suite of benefits and tradeoffs.

[0022] 10. To provide, within the present preferred scheme, a variety of tunable parameters which can be controlled and thus customized to advantageously meet the particular needs of security and efficiency at a variety of levels based upon the associated best match-

ing benefits and tradeoffs for the particular needs which characterize the context of the particular situation.

- [0023] 11. To create a scaleable system which can adapt to increasing numbers of users per community and/or where the community itself could potentially be as large as the totality of Internet users.

BACKGROUND OF THE INVENTION

[0024] In order to communicate securely over an insecure communication channel one needs to encrypt messages. Such encryption schemes fall into two broad classes—private key schemes and public key schemes. In a private key encryption scheme the Sender and the Receiver share a secret key *K*. This key is used both to encrypt and to decrypt the message. In a public key encryption scheme the Receiver has a pair of related keys—a secret key and a public key. He reveals his public key to everyone. To send a message to Receiver, sender encrypts it using Receiver's public key. Under certain assumptions (explained below), only the holder of the secret key, i.e., Receiver, can decrypt the message using a feasible computation.

[0025] When using a private key encryption scheme it is necessary for each pair of communicating parties to agree on a shared secret key before encrypted communication begins. In settings where the parties know each other and expect to communicate in the future this can be established quite easily. For example, such a setting might work for a communication network within a company where the messages are sent between the employees of the company. All the employees know each other in the sense that they are all listed in the company pay-roll. Each employee can establish private keys when joining the company. In contrast, a setting where private key encryption may not be appropriate, is a setting similar to the Internet where the parties do not have a way to get to know each other. In such settings, public key schemes offer immense practical advantages over private key schemes (with possibly only one novel exception, which is below described). In public key encryption each user simply has to publish his public key and keep his private key secret. No prior agreement is needed between the parties. Such considerations account for the enormous popularity of public key encryption schemes such as RSA over private key schemes.

[0026] However, from the standpoint of security, a particular private key encryption scheme called the one-time pad is the gold standard. In the one-time pad scheme, a secret key is a string of random bits and encryption is done by computing the exclusive-or of successive bits of the message with successive bits of the key. Decryption is done by computing the exclusive or of the bits of the cipher-text with the bits of the secret key. One-time pad schemes have the strongest possible security. As long as the secret key is not compromised, an eavesdropper who gets the cipher-text has no information that picks out the particular message transmitted over any other message. This is true even if we allow the attacker to have unlimited computational resources. Such a level of security is called perfect security or information-theoretic security and refers to the fact that it is the absence of information (about the message in the cipher-text) and not the computational prowess of the attacker that limits her. As the name implies, the key in a one-time pad is meant to be used just once and the key must

have as many bits as the message transmitted. This is well suited for applications where the users know in advance the approximate size and number of messages they plan to exchange in a give time. In such applications the users are able to determine the appropriate size of the secret key they need to share before communication begins. On the other hand, in applications where the size of the key cannot be determined ahead of time, using the one time pad scheme would require either exchanging a very large key with the possibility of never using much of it, or exchanging a shorter key with the possibility of ending the encrypted communication before the entire message got transmitted. Therefore, for these applications public-key schemes might be more appropriate.

[0027] Unfortunately, there are no perfectly secure public-key schemes. Public-key schemes are not secure if the attacker is given unlimited computational resources. An attacker can simply run through all possible secret keys, see which one is related to the public-key, and then use this secret key to decrypt the message. To obtain a public-key scheme one makes a computational assumption—that decryption is difficult for an attacker who can only perform probabilistic polynomial time computation. For example, in order to show that the RSA public key encryption scheme is secure, one needs to assume that factoring large integers is hard. (In fact, this assumption is necessary but not sufficient to ensure the security of RSA.) Thus, while public-key encryption offers logistic advantages, schemes based on this idea are vulnerable if the attacker has enormous resources at her disposal or the underlying assumption about the hardness of some computational problem turns out to be false. It has been shown that on a quantum computer the factoring problem can be solved efficiently. Thus the security of RSA rests on our inability to build quantum computers—not a satisfactory state of affairs for long-term, high-security communication.

[0028] A second drawback of public-key encryption is the computational cost of the encryption and decryption operations. Schemes such as RSA use large moduli (typically at least 500-bit) and have to compute expensive operations such as exponentiation with respect to such moduli. In contrast, a one-time pad uses the very fast operation of bitwise exclusive-or.

[0029] To summarize, the one-time pad private-key encryption scheme achieves the best security possible and essentially the fastest possible computation—bitwise exclusive-or. As mentioned above it is most appropriate for applications when the Sender and Receiver are able to establish a private-key. The one-time pad scheme can also be extended to scenarios where there are many pairs of senders and receivers. In such a scenario each pair of Sender and Receiver shares its own private key. This means that each Receiver must keep as many keys as there are parties that send messages to it which can become rather costly.

[0030] Therefore, when the one-time pad scheme is extended to applications where there are many parties it does not scale well.

[0031] The scheme proposed here is an information-theoretic (perfectly secure) private-key encryption scheme designed for a community of users with some commonality of interest. We achieve much greater efficiency than the one-time pad scheme when extended to many parties. This

is achieved while maintaining the information-theoretic security of the message with respect to an eavesdropper from outside the community but sacrificing security when there are large coalitions of colluding parties within the community. That is, within the community, if a coalition of sufficiently many members collude, they can together decrypt the cipher-texts exchanged by a pair of sender and receiver. We envision application scenarios where such malicious collusion between members of the community is extremely unlikely to occur and for such scenarios our scheme is perfectly secure. The level of security versus efficiency can be captured by the parameter S which is the threshold representing the minimum number of colluding members needed to cause this kind of insecurity.

[0032] (See appendix for exact calculation.)

[0033] This is a design parameter and can be chosen over a wide-range of values, representing a wide spectrum of trade-offs between efficiency and security. However, as is explained in further detail, regardless of what level of efficiency or security is deemed necessary, this overall level of compromise in security becomes relatively smaller in relation to the overall number of users on the system. As can be readily imagined, this specific feature of the scheme as proposed is an extremely important one for enabling it to be potentially more scaleable for purposes of extending the practical utility of at least certain implementations of the system to potentially increasingly large numbers of users. Such scalability advantages would represent a substantial advancement over pre-existing private key systems).

[0034] Assuming that there are no large coalitions, the security offered by our scheme is comparable to the security offered by the one-time pad scheme and does not rely on assumptions about the computational limitations of the attacker. We mention one specific important situation where collusion between many parties should not occur. Suppose we have an on-the-field communication network between several friendly agents operating in hostile territory. It is possible that one or more of the communication devices falls into enemy hands, but it is unlikely that a large number of them will fall into enemy hands before some kind of recovery action is taken. Each captured communication device can be viewed as a "colluding" party and the argument above says that there will not be too many colluding parties. This and other potential application scenarios are more fully described below.

[0035] We achieve significant savings over one-time pads in the number of keys to be remembered by each member of the community. These effects become enormous as the number of community members increases. In a community of N users, if we use a pure one-time pad scheme, each user has to remember $N-1$ keys, one for each other user. Overall, the system has about N -squared keys. In our scheme we reduce the total number of keys in the system to about N and the number of keys per user to about C times the square root of N where C is a tunable parameter.

[0036] Another advantage of our scheme to that of one-time pads or any other previously available cryptographic scheme, is that in our scheme it is possible to achieve security equivalent to one-time pads while also enabling the conditions by which it is possible for keys for any member in the group to be revoked by a coalition of other members. By contrast, one time pads lack this desirable property and

only allows either the Sender or Receiver of the message itself to revoke their own key.

BRIEF SUMMARY OF THE INVENTION

DETAILED DESCRIPTION

[0037] We now describe our scheme in detail. Our scheme allows M users to communicate with each other securely. A trusted server chooses N keys $K_1, K_2, \dots, K_N$ randomly such that all keys have the same length l . The trusted server assigns to each user a random subset of the N keys of size approximately C times the square root of N . To be more precise, the trusted server associates with each user a set of indices from 1 to N and assigns to the user the keys having those indices. It checks to see whether every pair of users shares some set of keys which are not shared by another user. The server then gives each user the particular random subset of size C times the square root of N assigned to it. In the unlikely event that two users do not share keys (See the appendix for extra details (*)), the server does not distribute the keys and starts the process of key assignments from the beginning to decide which new subset should go to which user.

[0038] Using probabilistic analysis, it is possible to show that any pair of users will share about C keys. However any other small subset of users is extremely unlikely to possess all C keys. Only large coalitions of colluding parties can obtain all C keys and break the security of the communication between two users. See the appendix for details (*).

[0039] After the key distribution is completed, the trusted server posts the indices of the keys (not the keys themselves) that each user receives. In order to post the information the trusted server can send the information to all the members. Alternatively the server can place it on a bulletin board or the web and all the users can access that information when they need it.

[0040] If users i and j want to communicate, they identify the keys that they share from the information posted about the indices that they share. They then take the bitwise exclusive—or of some of the keys they share to arrive at a key K which they use as a one-time pad. To encrypt a message they take the bitwise exclusive—or of the key K and the message m . Therefore, the length of the message must be equal to or smaller than the length of the key K .

[0041] If users want to exchange a message of size larger than the key K then they can split the message into several components each of the size of the key and encrypt each component using a different subset of the shared keys. If the message is smaller than the key then it can be padded with zeros before encryption.

[0042] After users i and j communicate using the key K constructed from their shared keys $K_{j1}, K_{j2}, \dots, K_{jc}$ they discard the keys with indices $j1, j2, \dots, jc$. They also post that the keys with indices $j1, j2, \dots, jc$ have been used and should be discarded by all users. This means that users i and j need to obtain new keys in order to communicate with one another. However, this should in most cases not affect the communication of other users. This is so, because as in the analysis of security the other users will normally have some other keys which they share. See the appendix for more details (*).

[0043] The number of keys N that the trusted server chooses can be set to be of the order of M the number of users. This number can be made larger depending on the level of security desired. It can also be set to be $2M$ if it is anticipated that additional users will be joining the system and the number of users is not likely to increase to more than double. This is described in more detail later below.

[0044] All the keys need to be of the same length. The length l can be determined by the typical message size of each particular application. The scheme can be augmented so that for each pair of users multiple one-time pad keys are available. The keys can also be of different sizes so that the users can choose the appropriate key for the appropriate message size.

[0045] Depending on the desired security and reliability the trusted server can be replaced by several servers. For additional security, instead of storing the secret keys with one trusted server, shares of the secret can be distributed among several servers. For additional reliability, instead of storing the secret keys or the shares of the keys with one server those can be distributed among several servers. For further security the keys can be chosen and decided upon by a few servers rather than by a single trusted server.

[0046] When a new user is added to the scheme, the trusted server decides which random subset of size approximately C times square root of N of the existing N keys in the system will be given to this user. It then checks that this new user shares keys with each of the other existing users. If not, the server decides on a new subset of keys. This should happen only with negligible probability. The keys are then given to the new user. For security reasons we want the number of keys N to be larger than the number of users. As mentioned above N will be chosen to be larger than M based on the expected number of users that will join the system. In the case that more users join and M is bigger than N the server chooses another random key and distributes it to all the users such that each user will receive this key with probability C divided by square root of N . The server updates the posted information about which key indices are given to which user.

[0047] If a system decides to revoke the membership of a user for whatever reason, the keys (not just the indices) belonging to that user are posted so that they are no longer secret. By doing so this user can no longer send any private messages. This does not affect the security of the other users since they will have other keys to use which are still secure. Since a large coalition of users is able to obtain the keys of a particular user, in a case of emergency the members of the community can vote and decide to revoke the membership of a particular user. Note that this can only be done if the size of the coalition is large enough. Therefore, a user will not be revoked by some adversarial members but only by a legal vote.

[0048] After user i and user j used their shared keys $k_{j1}, \dots, k_{jc}$, the keys with indices $j1, \dots, jc$ are discarded. These keys need to be replenished. The trusted server chooses new values for the keys with indices $j1, \dots, jc$ in place of the old ones. The server distributes these new keys to all the users who received the keys with these indices. This key replenishment can be done in an incremental way, every time keys are discarded. Alternatively, the server can wait a period of time until a larger number of keys have been used and then

replenish all of these. It can also be done based on a need basis. If users wish to communicate and they are missing keys the server can choose new keys and distribute them. The replenishment policy can be decided based on what is most appropriate for the application. If the keys are always replaced, the size of the coalition S of adversaries always stays the same. Otherwise, the coalition size decreases as the keys get discarded. Again, the exact trade-off between security and frequency of key replenishment can be worked out based on the needs of the application.

[0049] The key distribution can be done by physical means such as distributing the keys in a disk, on a hard drive, or on various secure hardware depending on the size of the keys). The replenishment of keys can also be done by physical means. When keys need to be replenished as was elaborated upon above, a new disk or hard drive can be distributed to the parties that are in need of replenishment. This key replenishment may create a bottleneck in some applications in our otherwise efficient scheme. Therefore, the technical piece which follows we demonstrate in one exemplary embodiment provided for the sake of illustration how to replenish keys more efficiently. However, this would result per the below illustrative example in somewhat reducing the level of security. Depending upon the needs of the present user and/or community, one other useful idea for the key distribution process (whichever method is chosen) is to distribute a miniaturized version of the shared set keys, also referred to later in this specification as “mini-keys”. Of particular utility for this application could be a scenario in which there exist members of a large (or potentially large) group of communicants (which in its ultimate incarnation is comprised of potentially an Internet-wide community) in which there is a possibility (even potentially only a remote possibility) for the need to communicate with those parties, however, memory limitations restrict the practicality of distributing the standard size keys in conjunction with the present key distribution procedure. Subsequently, if a communication between any of these potentially communicating parties (possessing these shared-set mini-keys) becomes warranted, a key replenishment procedure could then be used. For example, implementation of a scheme such as the analog cryptography scheme, which we have herein proposed below, would, (depending upon the security needs and present scope of other available technical options) be one possible approach. It is, of course implicitly understood within this portion of the present disclosure that there may presently or eventually exist a range and diversity of other alternative methodologies for performing key replenishment (as well as potentially key distribution) in addition to the one described below which could achieve additional degrees of advantage in efficiency and/or security. In this way, the disclosure of the herein suggested key replenishment and distribution schemes are intended in no way to limit the range or scope of the presently inventive system and method as it is herein proposed. Therefore, the users can choose—depending on the particular application—whether they prefer to increase the efficiency or maintain the highest level of security.

[0050] Depending on the security requirements of the application, the information-theoretically secure method of physical replenishing of keys can be replaced by a pseudo-random generator which depends on a computational hardness assumption. The computational assumption it depends on is very weak and assumed by many to be reasonable. This

assumption is so weak that it is believed to hold even if quantum computers can be built. Under this replenishment scheme, the server generates N random seeds $S_1, S_2, \dots, S_N$, for the pseudorandom number generator and distributes these seeds to users exactly as in the earlier scheme. Each user uses the seeds s/he has to produce longer keys which s/he then uses in the same fashion as in the earlier scheme.

[0051] As another possibility for key replenishment we suggest a scheme that relies on an even weaker-assumption. This scheme would be beneficial when the cost of bandwidth is less expensive relative to the cost of computation. This scheme uses steganography in a novel way. Usually, steganography is the art and science of embedding a message in data so that an adversary will not be able to tell whether the data has a message embedded in it or not. Typically a user will use data available from an outside source to embed the message. Therefore, the user will not have the privilege to choose the type of data used. For example, a worker in an office might want to send personal notes to another worker embedding the messages in the data files already distributed at the work place.

[0052] For the purpose of our key replenishing scheme, we use steganography—embedding a message in data—in a novel way, as encryption. Therefore, unlike other steganographic schemes it is not relevant to our scheme whether the adversary will be able to detect the existence of a message in the data. In fact, the data is sent only for the purpose of embedding, so all data streams will have messages embedded in them. Moreover, the data in our scheme could be created especially for the purpose of embedding messages in it. Therefore, we can choose the data yielding the highest security and efficiency. As in all private key encryption schemes, the message will be embedded in the data so that an adversary who reads the data will not be able to learn information about the message without knowledge of the secret key.

[0053] The data we choose to use for our key replenishing scheme is data produced from scanning color pictures, or data produced from color pictures taken on a digital camera. The reason this data is most appropriate for our use is because in digital data encoding colors there are usually several data streams representing the same visual image. If these data streams representing the same image cannot be distinguished, the message can be embedded in on such a stream which is randomly chosen from the set. We are then exploiting the entropy available in such data to communicate specific messages without an eavesdropper being able to figure out what the messages are.

[0054] Indeed statistical tests which are typically used to break steganographic schemes do not perform well in breaking data scanned from color pictures. Moreover, since in our scheme the trusted party replenishing the keys can choose which pictures to scan, it can make sure to pick the pictures with the most variability in color which decreases the possibility of finding any statistical patterns in the data.

[0055] There are several ways to embed a message (fresh keys in our application) in data. One good way is to embed it in the low order bits of the data. The size of the message will be small relative to the data. Therefore, not all the low order bits will have keys embedded in them and only a small fraction of them will. This again reduces the possi-

bility of performing statistical attacks on the data, since most statistical attacks succeed only when a large fraction of the bits are used for embedding. In order to decide where in the data to embed the bits we can use a weak pseudo random generator.

[0056] The scheme works according to the following steps. The trusted party established an initial secret key with all the users in the system as was described above. The secret key will contain the keys a and b . Then the trusted party creates new keys of length l as described above to be distributed by being embedded in the data for the purpose of being used by our proposed information theoretic encryption scheme.

[0057] In order to distribute the keys, the trusted party first scans color pictures to create the data of size D to be used only once. Note that for the purposes of security the size D can be made as large as desirable relative to l the size of the key. For each bit i of the key from 1 to l , the trusted party computes $a+b*i$ and embeds the i 'th bit of the key in the $a+b*i$ modulo D location in the data.

[0058] This scheme relies on a much weaker pseudo random generator than the one which is obtained from the one way function assumption. The reason we are able to rely on a much weaker assumption is because the data itself has some randomness. Our scheme is computationally secure in the following sense. If D is the length of the data stream, we consider $O(D)$ to be feasible computation whereas $O(D^2)$ to be infeasible.

[0059] A user who knows a and b can look at the appropriate sequence of addresses and compute the message in $O(D)$ time whereas an adversary who does not know a and b , has to consider all D^2 possibilities for the pair (a,b) , look at the m messages that they give rise to and choose the "right" one. If the message itself is a random looking string, someone who doesn't know the right (a,b) pair cannot tell which of the D^2 possible messages is the right one. Moreover, in order to get more security, it is possible to replace D^2 by D^3 by using 3 seeds a,b,c and computing $a+i*b+i^2*c$ for $i=1,2,\dots$ and similarly for higher powers. Thus, there is a tradeoff between the security and the efficiency of computation.

[0060] This scheme requires less computation than the previous scheme, which uses standard pseudo random generators. However, it does rely on the ability to send large amounts of data in an efficient manner. This quite likely is a reasonable assumption since bandwidth is turning out to be inexpensive whereas computation is still costly. In addition, scanning pictures is a task, which is easy and inexpensive.

[0061] The shared-set-keys scheme we have described is secure against an adversary, which tries to find information about the message.

[0062] Therefore the adversary cannot learn any information about the message itself. However, the adversary can still disturb communication by changing the bits of the cipher-text. This could result in the Receiver obtaining the wrong message. It is also possible for an adversary to resend a message to the Receiver pretending to impersonate the Sender. In order to prevent the above two types of attacks on the communications and at the same time preserve the unique highly advantageous benefits of the presently proposed encryption scheme we advocate implementing a spe-

cial type of digital signature. In particular, we can utilize a variation of the present shared-set encryption methodology, which can be implemented as information theoretic digital signatures, which can be computed very efficiently and use a substantially minimized version of the present shared set key methodology herein referred to as mini keys. Depending on the security requirements, it is possible to design the signature scheme with keys as short as 1 bit.

[0063] The signature scheme is similar to the encryption scheme with a few modifications. In the key generation stage the signature scheme requires random keys as in the encryption scheme. However, the keys here can be shorter, as short as one bit. Their length depends on how much security the application requires. In order to sign a message the Sender computes some function h on the message and then encrypts $h(m)$ the output of the function. In order to verify Receiver decrypts the message from the encryption, computes the function h on m and checks whether the second encryption is equal to $h(m)$.

[0064] If they are the same the verification succeeded. Otherwise the message was altered. The function h depends on the security level required. The most secure would be to use an error correcting code as h . This way the Receiver can detect and maybe even fix up to a guaranteed number of errors. If h is an error correcting code its output might be relatively long. If a shorter string is desirable it is possible to use a random hash function which will decrease the size exponentially. Another good candidate for h is the parity function whose output is only a single bit.

[0065] A strong motivation for the use of digital signatures is to ensure that the correct message was received. It is especially important for applications where the messages sent are codes. In such cases if the message is altered it might result in a program that runs erroneously which acts like a virus. Using the digital signatures it is possible to prevent using such erroneous code.

[0066] A motivation for our signature scheme is its use as an authentication scheme. If the sender computes the function h as prescribed and then encrypts the output as well as the message using his secret keys, it is clear that no other user can perform the same task and the sender will thereby have established his identity.

[0067] The information theoretic authentication is perfectly secure. This means that the probability that an adversary can successfully forge is very small $\sim 1/2^k$ —for a key of size k . For example, if the size of the key is 100 bits then the probability that an adversary can break the scheme is $1/2^{100}$ which is a very low probability. The size of the key k is flexible and can be chosen based on the security needs of the application. For example if for the purposes of the application it is enough to have a probability of breaking its security as 1 in 1,000,000 then the key can be chosen as size 20.

[0068] In our scheme, the size of the key can be chosen independently of the computing power of the adversary. On the other hand, in RSA like schemes, the size of the key depends on how powerful the adversary is. As the processing powers of an adversary increase, larger and larger keys will be needed to achieve a fixed level of security. Our scheme is faster to implement than RSA but has greater storage requirements. However, the storage needs of our scheme are well within the capabilities of present day computers.

[0069] In addition, our scheme is also compatible with various compression techniques. Thus it is possible to compress the message before encrypting to save on the number of key bits that are used up in each communication. We can also allow lossy compression whereby some information from the message is lost in the compression to further save on bits of the shared keys. In particular, if the current message is part of a sequence of messages between the two communicating parties, the receiver might be able to tolerate such information loss and recover the entire message from the context.

[0070] An important motivation for our scheme is that the encryption and decryption procedures are very efficient. Actually, they are as efficient as such procedures can be.

[0071] The system as described treats all the users as equals in the sense that they all can communicate with one another.

[0072] It is possible to modify the scheme so that the users in a hierarchy or onion community will be able to communicate according to the rules of the community.

[0073] In such a setting, users will have keys to communicate with members of their own group and people above them in the hierarchy.

[0074] It is possible to extend our scheme to much more general setting where we are told which pairs of users will potentially communicate. We can model this setting with a graph where the nodes are the users and edges are drawn between two nodes if the corresponding users are potential communicators. We refer to this as the communication graph. Before key distribution, the trusted server will perform an analysis of this graph.

[0075] Rather than distribute each key to a random subset of the users, the server will distribute each key randomly in a localized region within the communication graph. This produces a version of the scheme where a user who potentially communicates with k other users needs only about the square root of k keys. The total number of keys in the system will still be about N , the number of users. However, the above improvement in the number of keys each user needs to remember, opens up the possibility that our scheme could be adapted to work on much larger communities, such as the entire Internet. If N is the number of users on the Internet, it would be impractical for a user to have to remember about square root of N keys which need periodic replenishment. However, given that users on the Internet do not at present have a compelling need to securely communicate with everyone else, we can greatly reduce the number of keys each user needs to remember using the above approach. Other variations for optimally reducing the total number of keys in the system could, of course, be utilized without departing from the spirit of the claimed system and method for example, U.S. Pat. No. 5,754,939 describes a probabilistically based scheme which could be readily adapted for the end objective of predicting which users will likely communicate with which users and/or which servers and with what frequency in a probabilistic (and/or predictive fashion) and conversely which users and/or servers a given user with high level statistical confidence not likely to establish a communications connection. Thus the above approach could be replaced by that which is presently disclosed.

[0076] Undoubtedly, within the realm of conceivability would be an Internet wide implementation of the present scheme utilizing the above-described variation of mini-keys such as for implementing extremely high efficiency and extremely secure authentication and digital signature capabilities which are, of course, of an information theoretic secure nature.

[0077] One other application for such a widespread distribution of mini-keys is previously described and essentially leverages this particular variation of the present scheme to allow the shared mini-keys to instead (or in addition) assume the role of and thus achieve the initial key distribution itself. The only difference in this scenario to achieve that of standard shared sets is prior to establishing communication, connection with another party possessing appropriate matching shared-set mini-keys is that key replenishment would have to be performed (a distinct advantage for most applications than having to perform initial key distribution in de novo fashion).

EXAMPLE USES OF DISTRIBUTED SHARED KEYS TECHNOLOGY

[0078] Uses for this technology include, but are absolutely in no way limited to, situations where some fairly stable-sized group within a larger population wishes to have pair-wise secret communication among themselves. What follows are three examples: Private Communication within a Corporation/Organization's IT Infrastructure; Safer Virtual Private Networks (VPNs); and Mobile Military Communication Networks. In all of these cases the initial key-sets of each of the individuals in the communication group can be pre-distributed into the communication devices used.

EXAMPLE

[0079] Private Communication within a corporation/Organizations' IT Infrastructure. Currently, within an organization, computers on the same network can easily partake in peer-to-peer networking. The communications among the members of this network can be protected to some degree by having a firewall separate the organization as a whole from the rest of the network population (i.e., the Internet). However, there are at least two safety issues not addressed in this current set up. First, if the firewall is compromised, all interactions (including both data transfer and communications) become vulnerable. Second, Even assuming the inviolability of the firewall, internal communication and data transfer are vulnerable to interception by other members of the group.

[0080] Distributed Shared Keys solves the first of these problems by allowing for easy encryption of all data-transfers and communications. Thus allowing for another safety in the case of infiltration from the outside. It solves the second of these problems by allowing for pair-wise encrypted communication and data-transfer between any two members of the group. Thus, even if the firewall stands, communications and data transfers between Alice and Bob, who both work at the same company, can never be overheard by Eve, who also works there. This could all be done transparently by embedding the keys in each of the employee's computers. The IT staff can periodically update the key sets as new members arrive, or old members leave. The user can be totally unaware of the workings of the encryption—

unlike with public/private key schemes. Public key cryptography has not been the success that it was predicted to be, partly because it required the end user to manipulate a fairly large set of public keys before sending an encrypted message.

EXAMPLE

[0081] Corporate Virtual Private Networks (VPNs) "Virtual Private Network" is a fairly ambiguous phrase to describe the attempt to organize spatially disparate computers into a network for secure communications using existing public, and therefore insecure, networks (i.e., the Internet). An example would be using encrypted communication between the notebook of a traveling salesman and the home network. Another would be the series of connections necessary to allow a group of telecommuters to act as a work-group and securely share communications and data. However, many of these technologies are dependent upon public/private key schemes for the security of their communications. The problems with public/private key schemes are described above. Of note is that they will be susceptible to cracking with the advent of quantum computation. Since Distributed Shared Keys is comparable to the security of one-time pads, a VPN based on it is far more secure than current implementations. The key-sets could easily be embedded in the mobile users' computer to allow for seamless connection to the VPN.

EXAMPLE

[0082] Mobile Military Communications Network A very important possible use of Distribute Shared Keys is an on-the-field military communications network. It is not difficult to imagine groups of soldiers engaged in action needing fast, absolutely secure communications among themselves. In addition, it would be advantageous for the chain of command structure if sometimes many of those communications were not able to be intercepted even by other soldiers in the group. For example, a captain may want to securely talk to each of the lieutenants without the sergeants or privates being privy to the discussions. Often, there will be discussions of different options, each with their own trade-offs concerning loss of life, etc. It could be demoralizing to the foot soldier to have such discussion to be overheard. And, obviously, the entire communication of the group should be kept secret from the enemy. Using Distributed Shared Keys, pre-embedded in the communication devices of the group, there could be save group communication, as well as secure pair-wise communication. As aforementioned, this would give mobile military communications the security of one-time pads with the easier key distribution. As long as no more than C (see above) communicators fell into enemy hands, the communications would remain secure.

[0083] The three examples given are all of the format such that the group using Distributed Keys is a stable-sized subset of the whole population and there are, of course, numerous other example application domains in which the advantageous properties of the present scheme are highly desirable (e.g., government organizations in which different types of information are frequently passed between different individuals or groups each possessing very uniquely different criteria for accessibility and secrecy with regards to any given communication or series thereof). However, this is not

to rule out the usefulness of Distributed Shared Keys as a cryptographic technique used for the entirety of a population or for a subgroup whose size is dynamically changing and even the rapid reductions continuing to occur in costs for memory resources, the present scheme certainly could eventually become appropriate for incorporation into standard encryption and even security protocols for the Internet.

[0084] Additional Potential Emergent Application which Leverages a Highly Secure Network Infrastructure

[0085] The network infrastructure can be built to include a subnetwork comprised of highly secure links. If this is done, the key replenishment can be solved using this subnetwork. A user who needs his/her keys replenished accesses the highly secure subnetwork, authenticates him/herself to the trusted server and obtains a fresh set of keys. As a possible embodiment of this abstract idea access points for a highly secure subnetwork made up of optical fiber links (such as could eventually become achievable through present advances in all optical networking technologies) could be provided at points of high public traffic such as grocery stores or shopping malls such as visa vie kiosk terminals through which highly secure access to downloadable keys could be provided in asynchronous fashion to their use at a later time, i.e., over networks possessing an inherently lower degree of security. Given a satisfactory user identification means for the present level of security required the present approach could be used as well for performing the actual distribution of keys. (See Appendix for reference)

[0086] Other Applications Relevant to the Emergent Technical Advances in Quantum Cryptography

[0087] As briefly explained above the presently proposed scheme's ability to provide quantum secure cryptographic protection will be potentially a much more critical and indispensable advantage than those, which it provides in the present pre-quantum world. However, arguments have been made within the technical community that once the parallel but related field of quantum cryptography matures to the point of practical engineering feasibility the need for data transmitted over highly secure (quantum cryptography secured) optical connections, there will be less of a need to actually encrypt the communicated messages themselves in as much as it will then be possible to detect an eavesdropper anywhere along the line of the transmission. With regards to the above assertion, there is an issue on which our scheme impacts. Although the detection of an eavesdropper becomes possible over a quantum cryptographically secure communication channel, the fact remains that certain data is still likely to become intercepted by the eavesdropper in the process of the detection. Particularly if all of the data is very sensitive, risk of such a potential event may be unacceptable. To prevent this, it is possible to sue the secure channel to first pre-send the keys, validate that the keys had not been intercepted, then send the message itself in the one time pad encrypted form which we have presently herein disclosed. In order to counteract any undesirable latency effects (such as might be encountered particularly noticeably in conjunction with transmissions from live data feeds) such key/message transmissions could be broken up into smaller sub-segments and which, thus could be transmitted in very frequent sequentially alternating periodic intervals in which these smaller key segments if allowed by their appropriately corresponding encrypted message segments.

[0088] Additional Future Considerations as is Relevant to Potential Quantum Cryptanalytic Attacks

[0089] In accordance with almost all other types of encryption methods (i.e., which include all cypher based encryption techniques) the high computational overhead associated with such security measures may influence the sender not to encrypt the message at all. While the increasing computational power in the future will tend to reduce the computational burden of decrypting a given size cipher text, it is anticipated that the above concern will only increase as cryptanalytic tools in the hands of would-be attackers increases and accordingly drives up the overall relative computational overhead burden. This effect is expected to become particularly significant as quantum processing becomes a practical reality. By contrast, our scheme is not subject to the above concerns.

CONCLUSION

[0090] In conclusion, we summarize some of the benefits of our scheme. The encryption scheme proposed here achieves perfect security (information theoretic). It achieves a significant saving in the number of keys used relative to the extension of the one-time pad scheme. The encryption and decryption algorithms proposed in the scheme are as efficient as possible. Our scheme is designed so that it is possible to easily and efficiently revoke and add membership of new users into the system. It is also designed so that authentication and security against man in the middle attacks can be added at low cost.

[0091] In addition, we introduce a novel and efficient way to use steganography for key replenishment.

APPENDIX

[0092] 1.

[0093] In this section we calculate the probability that two users in the scheme U_i and U_j share m keys after the trusted party distributes the keys are described in the document.

[0094] Consider two users U_i and U_j ,

[0095] Let X_i be a random variable which is equal to 1 when both users U_i and U_j receive the i th key, and 0 otherwise.

[0096] $\Pr[X_i=1] = (c^2)/n$, since each key is given to the users with probability $c/(n^{1/2})$

[0097] Let $X = \sum X_i$, which is the number of keys both parties have in common.

[0098] The expected value of X

$$E[X] = E[\sum X_i] = \sum E[X_i] = \sum c^2/n = c^2.$$

[0099] We want to calculate the probability that the users share at least m keys,

$$\Pr[X \geq m] = 1 - \Pr[X < m]$$

[0100] The Chernoff bound gives us:

$$\Pr[X < (1-k) * c^2] < e^{-(-c^2 k^2 / 2)}$$

$$\text{To get } m > (1-k) * c^2$$

we set

$$e^{-(-c^2 k^2 / 2)} < 1/\text{poly}(n)$$

$$c^2 k^2 / 2 > \log(\text{poly}(n))$$

solving for k we get

$$k^2 > 2 \log \text{poly}(n)/c^2.$$

[0101] 2.

[0102] In this section we calculate the size of a coalition which cannot break the encryption.

$$\text{Prob}[A \text{ coalition of size } k \text{ does not have all } m \text{ of } m \text{ keys}] =$$

$\text{Prob}[\text{Each member of a coalition of size } k \text{ does not have at least one of } m \text{ keys}] = m^* \text{Prob}[\text{Each member of a coalition of size } k \text{ does not have a key}] = m^* (\text{Prob}[A \text{ member does not have a key}])^k = m(1 - c/n^{1/2})^k$

[0103] Solve this for k :

$$m(1 - c/n^{1/2})^k = 1 \quad \{\text{Using } 1 + x < e^x\}$$

$$m * e^{(-kc/n^{1/2})} > 1.$$

$$\log m + -kc/n^{1/2} > 0$$

$$k < n^{1/2} * (\log m)/c$$

[0104] 3.

[0105] In this section we calculate the size of a coalition which cannot break the encryption after a set of keys was thrown away?

[0106] The analysis is similar.

[0107] We now have

$$k < n^{1/2} * (\log m - 1)/c$$

where l is the number of keys that were thrown away.

REFERENCE

[0108] D. Stucki, N. Gisin, O. Guinnard, G. Ribordy, H. Zbinden "Quantum Key Distribution over 67 km with a plug and play system". arXiv:quant-ph/0203118V1, 22 Mar. 2002.

[0109] The trusted party chooses N keys $K_1, K_2, K_3, \dots, K_N$ and distributes them to the M users $U_1, U_2, U_3, \dots, U_M$. Each user gets a subset of the N keys. Any two users share a subset of keys.

[0110] Below is an example of how the keys can be distributed for a community of 4 users.

	K1	k2	K3	K4
U1	x	x	x	
U2	x		x	x
U3		x	x	x
U4	x	x		x

[0111] In this example the trusted party generates the random keys k_1, k_2, k_3, k_4 .

[0112] It then distributes the keys as follows: the user U_1 gets the keys k_1, k_2, k_3 ; the second user U_2 gets the keys k_1, k_3, k_4 ; the third user U_3 gets the keys k_2, k_3, k_4 ; the fourth user U_4 gets the keys k_1, k_2, k_4

[0113] If U_1 and U_2 wish to communicate they create a key computing the exclusive—or the keys they share k_1 and k_3 . Note that the other users U_3 and U_4 alone cannot find out the key used by U_1 and U_2 because each lacks information. U_3 does not know k_1 and U_4 does not know K_3 . Only if U_3 and U_4 collude then they will be able to find both k_1 and k_3 and compute the key used by U_1 and U_2 . It is easy to see that this is the case for keys computed between any pair of users.

[0114] In this example the size of an illegal coalition is small (two users). This is so because the number of users in this example is so small. However, when the number of users M increases the coalition size increases as well.

[0115] Our scheme requires $N=M$ keys. (When $M=4$ the number of total keys is 4). The number of keys each user keeps is roughly square root of M (When $M=4$ the number of keys is 3). In this example the number of keys each user keeps is relatively close to the total number of keys. This is so because the number of users in this example is so small. However, when the number of users increases the number of keys is proportionally smaller.

[0116] To illustrate the efficiency of this scheme, we compare this with what can be achieved for the same number of users $M=4$ when using an extension of the one-time pad scheme. The one-time pad scheme would require $N=M(M-1)/2$ keys (When $M=4$ the number of keys is 6). The number of keys each user keeps is $M-1$ (When $M=4$ the number of keys is 3)

User\Keys	K1	K2	K3	K4	K5	K6
U1	x	x	x			
U2	x			x	x	
U3		x		x		x
U4			x		x	x

[0117] In our scheme there are less overall keys (M compared with $M(M-1)/2$). In our scheme each user stores less keys (square root of M compared with $M-1$). As M gets larger the savings become more significant.

We claim:

1. A system and method for providing a secure encryption scheme based upon keys which, are shared among multiple users in a community of potentially communicating members

* * * * *