

Petition for *Inter Partes* Review of U.S. Patent No. 7,266,682

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

META PLATFORMS, INC.; INSTAGRAM, INC.; WHATSAPP LLC; META
PLATFORMS TECHNOLOGIES, LLC; AND GIPHY, INC.

Petitioners

v.

VL COLLECTIVE IP LLC

Patent Owner

U.S. Patent No. 7,266,682

Case IPR2023-00923

**DECLARATION OF DR. HENRY HOUH, PH.D.
IN SUPPORT OF PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 7,266,682**

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	QUALIFICATIONS	4
III.	RELEVANT LEGAL STANDARDS	8
IV.	BACKGROUND	12
	A. Summary of the '682 Patent.....	12
	B. Summary of the Specification	13
	C. Summary of the Prosecution History	18
	D. Claim Construction	19
	E. Priority Date of the Challenged Claims	19
	F. Person of Ordinary Skill in the Art	19
	G. State of the Art	20
V.	IDENTIFICATION OF CHALLENGES	26
	A. Challenged Claims	26
	B. Statutory Grounds for Challenges	26
VI.	OVERVIEW OF THE PRIOR ART	26
	A. Handley.....	27
	B. Basturk.....	28
	C. PacketCable	29
VII.	IDENTIFICATION OF HOW THE CHALLENGED CLAIMS ARE UNPATENTABLE.....	30
	A. Ground I: Claims 1-21, 23, and 25-28 Are Rendered Obvious over <i>Handley</i> in Combination with <i>Basturk</i>	30

B.	Ground II: Claims 1-28 Are Rendered Obvious Over <i>PacketCable</i> in Combination With <i>Handley</i>	72
C.	Ground III: Claims 1-28 Are Rendered Obvious Over <i>PacketCable</i> in Combination with <i>Basturk</i>	99
D.	Ground IV: Claims 22 and 24 Are Rendered Obvious Over <i>Handley</i> in Combination with <i>Basturk</i> and <i>PacketCable</i>	109
VIII.	SECONDARY CONSIDERATIONS	110
IX.	CONCLUSION AND DECLARATION	111

LIST OF EXHIBITS

Ex. No.	Description
1001	U.S. Patent No. 7,266,682 (“the ’682 Patent” or “the Patent”)
1002	Certified File History of U.S. Patent No. 7,266,682
1003	Declaration of Dr. Henry Houh, Ph.D. in Support of Petition for <i>Inter Partes</i> Review of U.S. Patent No. 7,266,682
1004	Curriculum Vitae of Dr. Henry Houh
1005	M. Handley, J. Crowcroft, & C. Bormann, Very Large Conferences on the Internet: The Internet Multimedia Conferencing Architecture and the MBONE by, https://www.cl.cam.ac.uk/~jac22/otalks/cnisdn-article.pdf (“Handley”)
1006	Erol Basturk, Mihir Bellare, Chee-Seng Chow, & Roch Guérin, Secure Transport Protocols for High-Speed Networks, https://cseweb.ucsd.edu/~mihir/papers/srtp.pdf (“Basturk”)
1007	PacketCable 1.0 Architecture Framework Technical Report, https://community.cablelabs.com/wiki/plugins/servlet/cablelabs/alfresco/download?id=bddae471-01dd-43c8-8e59-d6547fcf4ffd (“PacketCable”)
1008	RTP: A Transport Protocol for Real-Time Application, IETF RFC 1889, https://datatracker.ietf.org/doc/html/rfc1889
1009	Andrew S. Tanenbaum, Computer Networks (3 rd ed. 1996) (“Tanenbaum”)
1010	Declaration of Sylvia D. Hall-Ellis, Ph.D. in Support of Petition for <i>Inter Partes</i> Review of U.S. Patent No. 7,266,682
1011	Reserved
1012	Complaint, <i>VideoLabs, Inc., et al. v. Meta Platforms, Inc. et al.</i> , Civil Action No. 1-22-cv-00680 (D. Del.)
1013	Internet Protocol – RFC 791, DARPA Internet Program (September 1981)
1014	Carrier Sense Multiple Access with Collision Detection, IEEE Standard 802.3, IEEE Standards for Local Area Networks (1985)

1015	David D. Clark & David L. Tennenhouse, Architectural Considerations for a New Generation of Protocols, Laboratory for Computer Science, M.I.T. (1990), https://dl.acm.org/doi/10.1145/99517.99553 .
-------------	--

Unless otherwise noted, all emphasis added.

TABLE OF ABBREVIATIONS

Term	Abbreviation
Person of Ordinary Skill In The Art	POSITA
Real-time Transport Protocol	RTP
Protocol Data Unit	PDU
Open Systems Interconnection Model	OSI Model
Message Authentication Code	MAC
Voice over Internet Protocol	VoIP
Internet Protocol	IP
Message-Digest Algorithm	MD5
Data Encryption Standard	DES
Secure Hash Algorithm 1	SHA-1
Public Switched Telephone Network	PSTN

1. I, Henry Houh, declare as follows:

I. INTRODUCTION

2. I am over the age of eighteen (18) and otherwise competent to make this Declaration.

3. I have been retained by Meta Platforms, Inc, Instagram, Inc., WhatsApp LLC, Meta Platforms Technologies, LLC, and GIPHY, Inc. (“Petitioners”) as an independent, technical expert consultant in this proceeding before the United States Patent and Trademark Office (“USPTO”). I am being compensated at my rate of \$680 per hour for my services. No part of my compensation is dependent on my opinions or on the outcome of this proceeding. I have no financial interest in any of the parties in this proceeding.

4. This Declaration is in support of the Petition for *inter partes* review involving U.S. Patent 7,266,682 (“the ’682 Patent” or “the Patent”) (EX1001), entitled “Method and System for Transmitting Data From a Transmitter to a Receiver and Transmitter and Receiver Therefor” and listing Martin Euchner as the inventor.

5. For the purposes of this *inter partes* review, and as I discuss later, I have been instructed to assume that the effective filing date of the Claims of the ’682 Patent challenged by the Petitioner in this *inter partes* review is no earlier than January 18, 2000.

6. I understand that, according to the USPTO records, the '682 Patent is currently assigned to VL Collective IP LLC ("VideoLabs" or "Patent Owner"). I assume for the purposes of this proceeding that VideoLabs is the patent owner.

7. The '682 Patent is generally directed to data transmission, and specifically, a method and system for transmitting data from a transmitter to a receiver. The Challenged Claims purport to address a technique for transmitting data from a transmitter to a receiver with the goal of ensuring the data is not unwanted. The methods and systems claimed are based on, what the '682 Patent confirms are, known and conventional data transmission protocols and authentication techniques. The '682 Patent subsequently alleges to provide authentication from a transmitter to a receiver at a Real Time Transport Protocol (RTP) packet level as an application protocol on an application layer by inserting an authentication data at the transmitter at the end of a whole RTP packet payload. Based on the RTP packet level authentication, a receiver allegedly accepts the whole RTP packet payload if the receiver knows the transmitter based on the RTP packet level authentication data. At base, the receiver simply uses the authentication data to determine whether it knows the transmitter, and if the receiver does not, the data is rejected. As described herein, the '682 Patent's

techniques were known in the art (as admitted by the Patent itself), and the Challenged Claims¹ should be cancelled as obvious and unpatentable.

8. I am familiar with the technology described in the '682 Patent as of the earliest possible priority date of January 18, 2000, the filing date of German Application No. 10001855.6.

9. In preparing for this Declaration, I have reviewed the '682 Patent, as well as the patents and documents cited herein, and I have considered these documents in light of the general knowledge in the art as of January 18, 2000. In formulating my opinions, I have relied upon my experience in the relevant art. I have also considered the viewpoint of a person of ordinary skill in the art ("POSITA") in the relevant field, as of January 18, 2000, who I describe below in Section IV.F.

10. I have been asked to provide my technical expertise, analysis, insights and opinions regarding the '682 Patent and relevant references that form the basis of the arguments set forth in the accompanying Petition for *inter partes* review of the '682 Patent. As described in detail below, I offer the following opinions in this Declaration:

11. It is my opinion that Claims 1-21, 23, and 25-28 of the '682 Patent are rendered obvious over *Handley* in combination with *Basturk*.

¹ I understand the Challenged Claims to be 1-28 of the '682 Patent.

12. It also is my opinion that Claims 1-28 of the '682 Patent are rendered obvious over *PacketCable* in combination with *Handley*.

13. It also is my opinion that Claims 1-28 of the '682 Patent are rendered obvious over *PacketCable* in combination with *Basturk*.

14. It also is my opinion that Claims 22 and 24 of the '682 Patent are rendered obvious over *Handley* in combination with *Basturk* and *PacketCable*.

15. Unless otherwise indicated, the emphasis in quotations has been added.

II. QUALIFICATIONS

16. My qualifications for forming the opinions set forth in this declaration are summarized here and explained in more detail in my *curriculum vitae*, which is included in EX1004. EX1004 also includes a list of my publications and patents.

17. I received a Ph.D. in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology ("MIT") in 1998. I also received a Master of Science degree in Electrical Engineering and Computer Science in 1991, a Bachelor of Science degree in Electrical Engineering and Computer Science in 1989, and a Bachelor of Science Degree in Physics in 1990, all from MIT.

18. For the last 13 years I have been self-employed as an independent technical consultant which specializes in, among other things, Web Site

Development, Data Networking, Optical Networking, Telecommunications, and Media Streaming. My clients include almost every major law firm in the country.

19. My career in the area of telecommunications began in 1987 when I worked at AT&T Bell Laboratories as part of a five-year dual degree program at MIT. Later, as part of my doctoral research at MIT from 1991–1998, I was a research assistant in the Telemedia Network Systems (TNS) group at the Laboratory for Computer Science. The TNS group built a high-speed gigabit network and created applications that ran over the network. Example applications included ones for remote video capture, processing, and display of video on computer terminals. In addition to working on the design of core network components, designing and building the high-speed links, and designing and writing the device drivers for the interface cards, I also set up the group's web server. Furthermore, because our group focused on real-time software processing of live streamed data, some members of our group focused on research in software radios, which was thus commonly discussed in meetings and casual conversations. One of my lab mates, Vanu Bose, went on to found Vanu, Inc., a software radio company focused on low-cost base stations for rural cell carriers. Several members of our group joined Vanu's company and I continued to meet with Vanu and members of his company. Thus, I became aware of the architectures around software radios starting in the late 1990's.

20. I defended and submitted my Ph.D. thesis, titled “Designing Networks for Tomorrow’s Traffic,” in January 1998. As part of my thesis research, I analyzed local-area and wide-area flows to show a more efficient method for routing packets in a network, based on traffic patterns at the time.

21. From 1997 to 1999, I was a Senior Scientist and Engineer at NBX Corporation, a start-up that made business telephone systems for streaming packetized audio over data networks instead of using traditional telephone lines. NBX was later acquired by 3Com Corporation, and the phone system is still used today by numerous businesses. As part of my work at NBX, I designed the core audio reconstruction algorithms for the telephones, as well as the packet transmission algorithms. I also designed and validated the core packet transport protocol used by the phone system. The protocol was used for all signaling in the phone system. The NBX system also featured a computer interface for initiating phone calls, which could also initiate conference calls. The NBX system also supported TAPI, the Telephony Application Programming Interface, thus allowing other computer programs to integrate with our system telephony features. We obtained U.S. Patent No. 6,697,963, entitled “Telecommunication method for ensuring on-time delivery of packets containing time-sensitive data,” as a result of part of this work.

22. From 1999 to 2004, I was employed by Empirix or its predecessor company, Teradyne. Empirix was a leader in test tools for telecommunications protocols and systems, providing functional testing tools as well as load testing tools. From 2000 to 2001, I conceived and built a test platform for testing Voice-over-IP (VoIP). The first application on this new test platform was a cloud emulator for simulating the effects of transmitting VoIP over a busy network.

23. In 2006, as part of my role at BBN Technologies, I helped found PodZinger Inc., now known as RAMP Inc. PodZinger utilized BBN's speech recognition algorithms to search through the spoken words in audio and video segments. While I was Vice President of Operations and Technology, PodZinger followed its initial prototype with a full streaming audio and video search solution.

24. Furthermore, I have served as expert witness in over 140 patent-related cases where I wrote expert reports and testified in over 80 depositions on topics such as network communications and telecommunications, including Internet protocols, streaming media, and mobile wireless technologies. I have testified in Federal courts, before the PTAB, and at the ITC on behalf of leading companies such as Amazon, AT&T, Cisco Systems, Google, Microsoft, Comcast, Time Warner Cable, Samsung, and Apple.

25. I have been awarded several United States patents, including the following examples:

U.S. Pat. No.	Title
7,975,296	Automated security threat testing of web pages
7,877,736	Computer language interpretation and optimization for server testing
7,801,910	Method and apparatus for timed tagging of media content
7,590,542	Method of Generating Test Scripts Using a Voice-Capable Markup Language
7,860,999	Distributed computation in network devices
6,967,963	Telecommunication method for ensuring on-time delivery of packets containing time-sensitive data
7,359,993	Method and apparatus for interfacing external resources with a network element

26. Based on my above-described over three decades of experience in computer networking, VoIP, data transmission protocols cryptology, data authentication, I believe that I am qualified to be an expert in the field of data transmission and authentication.

III. RELEVANT LEGAL STANDARDS

27. I do not provide any legal opinions in this Declaration. I have been informed and understand that certain legal standards are to be applied by technical experts in forming opinions regarding the meaning and validity of patent claims. I have been asked to provide my opinions regarding whether the claims of the '682 Patent would have been obvious to a person having ordinary skill in the art at the time of the alleged invention, in light of the prior art.

28. Further, I have been informed and understand that a patent claim is not patentable as obvious (a requirement I understand to be governed by statute 35

U.S.C. § 103) if the differences between the patent claim and the prior art are such that the claimed subject matter as a whole would have been obvious to a person having ordinary skill in the relevant art at the time the claimed invention was made. Obviousness, as I have been informed and understand, is based on the scope and content of the prior art, the differences between the prior art and the claim, the level of ordinary skill in the art, and, to the extent that they exist, certain objective indicia of non-obviousness.

29. I understand that objective indicia can be important evidence regarding whether a patent is obvious or nonobvious, if it has an appropriate nexus to the claimed invention, i.e., is a result of the merits of a claimed invention (rather than the result of design needs, market-pressure advertising, or similar activities). Such indicia include: commercial success of products covered by the patent claims; long-felt need for the invention; failed attempts by others to make the invention; copying of the invention by others in the field; unexpected results achieved by the invention as compared to the closest prior art; praise of the invention by the infringer or others in the field; taking of licenses under the patent by others; expressions of surprise by experts and those skilled in the art at the making of the invention; patentee proceeded contrary to the accepted wisdom of the prior art, and contemporaneous development of the subject matter claimed by others.

30. I have been informed that whether there are any relevant differences between the prior art and the claimed invention is to be analyzed from the view of a person of ordinary skill in the relevant art at the time of the invention. As such, my opinions below as to a person of ordinary skill in the art are as of the time of the invention, even if not expressly stated as such; for example, even if stated in the present tense.

31. In analyzing the relevance of the differences between the claimed invention and the prior art, I have been informed that I must consider the impact, if any, of such differences on the obviousness or non-obviousness of the invention as a whole, not merely some portion of it. The person of ordinary skill faced with a problem is able to apply his or her experience and ability to solve the problem and also look to any available prior art to help solve the problem.

32. I have been informed that a precise teaching in the prior art directed to the subject matter of the claimed invention is not needed. I have been informed that one may take into account the inferences and creative steps that a person of ordinary skill in the art would have employed in reviewing the prior art at the time of the invention. For example, if the claimed invention combined elements known in the prior art and the combination yielded results that were predictable to a person of ordinary skill in the art at the time of the invention, then this evidence would make it more likely that the claim was obvious. On the other hand, if the

combination of known elements yielded unexpected or unpredictable results, or if the prior art teaches away from combining the known elements, then this evidence would make it more likely that the claim that successfully combined those elements was not obvious.

33. I have been informed and understand that there are recognized, exemplary rationales for combining or modifying references to show obviousness of claimed subject matter. Some of the rationales include the following: combining prior art elements according to known methods to yield predictable results; simple substitution of one known element for another to yield predictable results; use of a known technique to improve a similar device (method or product) in the same way; applying a known technique to a known device (method or product) ready for improvement to yield predictable results; choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success; known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces if the variations are predictable to one of ordinary skill in the art; and some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to modify the prior art reference or to combine prior art teachings to arrive at the claimed invention.

IV. BACKGROUND

A. Summary of the '682 Patent

34. The '682 Patent is titled “Method and System for Transmitting Data From a Transmitter to a Receiver and Transmitter and Receiver Therefor” and is generally directed to data transmission, and specifically, a method and system for transmitting data from a transmitter to a receiver.

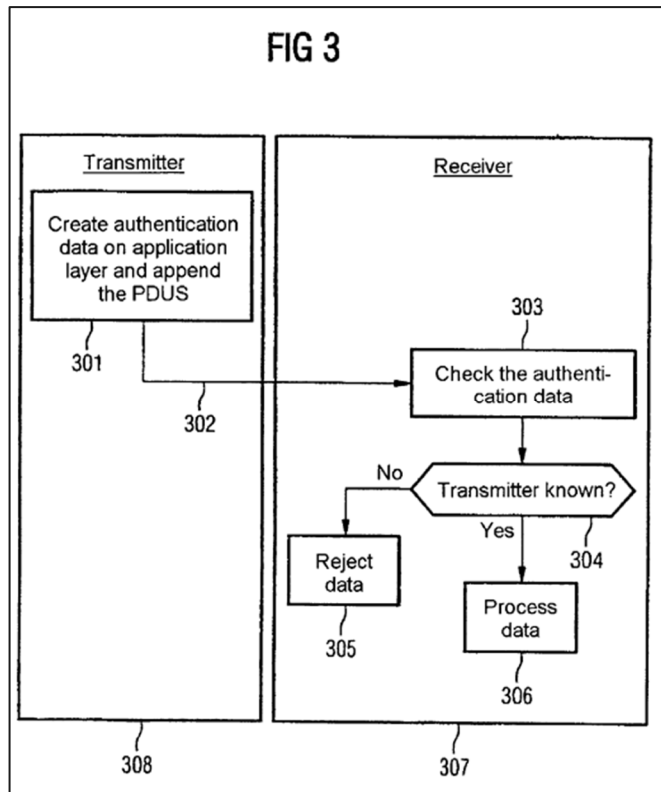
35. The '682 Patent describes a solution to transmitting data from a transmitter to a receiver with the goal of ensuring the data is not unwanted. The '682 Patent implements known methods of data transmission, data authentication, and a known data transmission protocol, Real-time Transport Protocol (“RTP”).

36. Claim 1 recites:

No.	Claim Language
Preamble [1]	A method for transmitting data from a transmitter to a receiver, comprising:
Element [1.1]	providing transmitter-to-receiver authentication at a Real Time Transport Protocol (RTP) packet level as an application protocol on an application layer
Element [1.2]	by inserting, at the transmitter, authentication data at end of a whole RTP packet payload;
Element [1.3]	ascertaining, by the receiver, whether the receiver knows the transmitter based on the RTP packet level authentication data; and
Element [1.4]	accepting, by the receiver, the whole RTP packet payload, if the receiver knows the transmitter, and otherwise rejecting the whole RTP packet payload.

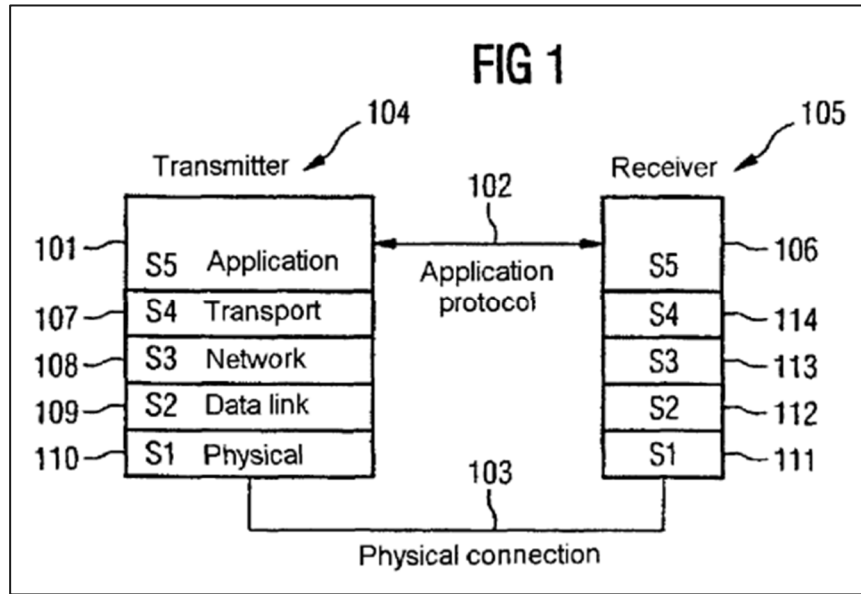
B. Summary of the Specification

37. The '682 Patent purports to disclose known methods and systems for transmitting data from a transmitter to a receiver while ensuring the transmitted data is not unwanted. EX1001, Abstract. The '682 Patent allegedly achieves this “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.” *Id.*, 3:49-52. 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.” *Id.*, 3:52-55. Data transmission is known in the art, and the claimed process is allegedly shown in Figure 3, reproduced below.



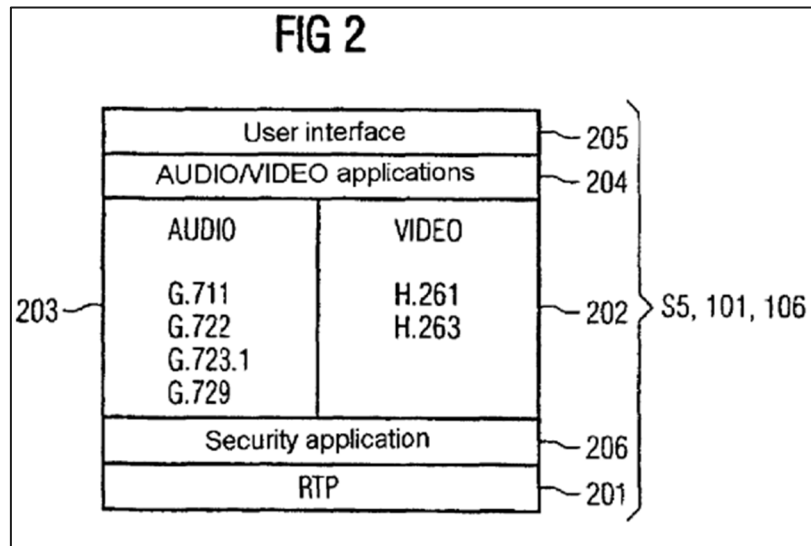
EX1001, Fig. 3

38. “The transmission of data is known, by way of example, on the basis of the OSI reference model as described [in the prior art].” EX1001, 1:19-21. The ’682 Patent adopts the use and framework of an OSI Model which “has seven layers, each of which has a different functionality (in terms of abstraction).” *Id.*, 1:22-25. This known OSI model is allegedly shown in Figure 1, reproduced below. Figure 1 shows five layers instead of seven, allegedly because for an application protocol, “the three layers above the transport layer 4 combine to form an ‘application layer.’” *Id.*, 1:40-45.



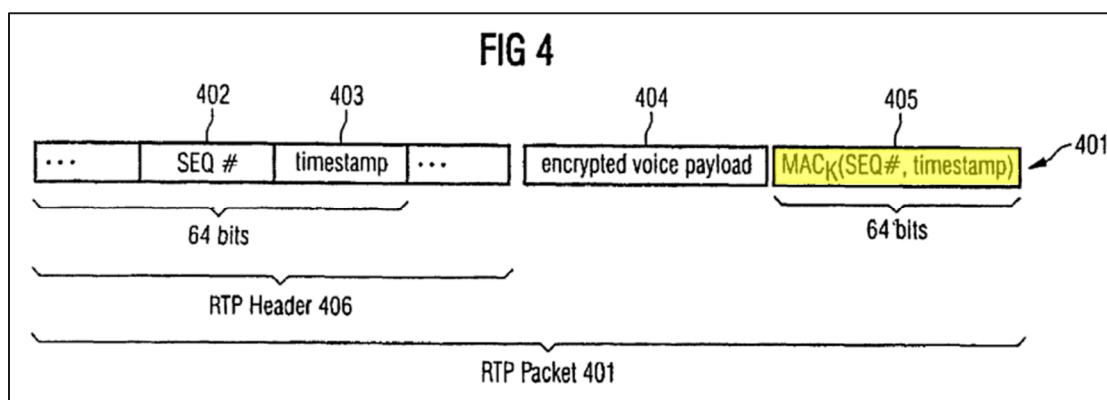
EX1001, Fig. 1

39. The known application layer comprises the user interface, audio/video applications, audio/video data, the security application, and a known networking protocol entitled Real-time Transport Protocol (“RTP”). *See* EX1001, Figure 2. The application layer is allegedly shown in Figure 2, reproduced below.



EX1001, Fig. 2

40. In using the known OSI model, the '682 Patent utilizes the known transmission protocol RTP to transmit media data “payloads,” *i.e.*, video or audio data. EX1001, 3:11-12. Included is an RTP Header (including a sequence number and timestamp), encrypted media payload, and inserted authentication data. *See id.*, Figure 4. An RTP packet is allegedly shown in Figure 4, reproduced below with the authentication data annotated in yellow.



41. The '682 Patent describes the alleged novel aspect as placing “authentication in the application layer” of the OSI model, which “has an advantageous effect on real time and availability of the entire communication system.” EX1001, 3:66-4:1-3. Namely, “authentication in the application layer is particularly advantageous, because the data packets themselves, in contrast to the layers beneath, are of significantly reduced size.” *Id.* The lower layers, including the TCP, IP, and link layers (such as Ethernet) add packet headers to packet data that are used by networking components and the network stacks to route data properly. For example, a TCP header consists of at least 20 bytes, an IP header

consists of at least 20 bytes, and an Ethernet header consists of 18 bytes. Thus, a packet contains at least 58 bytes of header information in addition to the packet's data contents. The use and benefits of appending the authentication data in the application layer were known in the prior art prior to the priority date of the '682 Patent. For example, Tanenbaum, a known networking text, devotes the first section of the chapter on "The Application Layer" (EX1009, 598-787) to "Network Security" (*Id.*, 598-643). Furthermore, the Network Security section covers "Authentication Protocols" and "Digital Signatures." *Id.*, 622-641. Tanenbaum discloses that it is an application layer process that performs authentication: "Authentication is the technique by which a process verifies that its communication partner is who it is supposed to be and not an imposter. Verifying the identity of a remote process in the face of a malicious, active intruder is surprisingly difficult and requires complex protocols based on cryptography." *Id.*, 622. According to the '682 Patent, the authentication data, applied 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." EX1001, 4:22-26. This authentication data is described as follows: "[b]oth 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 and the

timestamp.” *Id.*, 7:30-34. Because the MAC “requires a secret key in order to calculate it,” the receiver can check that the MAC is the same as the one the transceiver sent, and if so, “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.” *Id.*, 2:25-28, 7:42-45.

C. Summary of the Prosecution History

42. During prosecution, the examiner repeatedly rejected the ’682 Patent’s claims based on a “similar invention [U.S. Patent No. 6,918,034 (“Sengodan”)] of inserting an authentication data at the end of the whole RTP packet.” EX1002, 603-604. The applicant disagreed that Sengodan taught every claim element and argued that instead of adding an authenticator at the end of the *whole* RTP packet, Sengodan disclosed adding an authenticator to *each* mini-packet assembled into the RTP packet payload for authentication. *Id.*, 563. Initially, the examiner accepted that each mini-packet is a packet, but after an apparent interview with the examiner, the applicant convinced the examiner otherwise. *Id.*, 598, 603-604. While each element of the ’682 was found in the prior art, the claims were ultimately allowed because mini-packets were considered different than RTP packets by the examiner:

[I]t is interpreted that a mini-RTP packet is considered to be different from the RTP packet and at the time of Sengodan’s invention, there is no mechanism for authentication at the RTP level and the claimed invention of inserting the authentication data at the end of the whole

RTP packet is considered a different technique from inserting the authentication data at the end of the mini-RTP packets.

Id., 604. Accordingly, the examiner determined that “the cited prior art, either singularly or in combination fail to anticipate or render the claimed invention obvious.” *Id.*

D. Claim Construction

43. I understand that Petitioner proposes that each claim term in the Challenged Claims be given its plain and ordinary meaning in this proceeding, and that no specific construction of any claim term is required because the prior art relied on in this Petition meets each of the claim terms under any reasonable construction.

E. Priority Date of the Challenged Claims

44. The '682 Patent claims the benefit of U.S. Patent Application Ser. No. 10/181,564, based on a PCT filing date of January 5, 2001. Solely for the purposes of this IPR, I have been instructed to assume the priority date is January 18, 2000, based on the filing of a foreign German application bearing number 10001855.

F. Person of Ordinary Skill in the Art

45. A POSITA at the time of the '682 Patent would have had a Bachelor's degree in computer science, electrical engineering, or a related field, and two years of experience; or by obtaining a Master's degree in electrical engineering or a related field, but having only one to two years of experience; or by having no

formal education but experience in network data transmission and computer networking of at least eight years. Additional education might compensate for less experience, and vice-versa.

G. State of the Art

46. The following section describes the state of the art for data transmission, OSI Reference Model, RTP, and Authentication as of January 2000. The prior art references and the discussions of what was known to a POSITA provide the factual support for the general description of the state of the art at the time of the invention and additional motivation to modify or combine the references. Accordingly, these references should be considered by the Board.

1. Data Transmission

47. Data transmission is the process of sending either digital or analog data from one device to another, *e.g.*, from a transmitter to a receiver. In computer networking, digital data is transmitted in the form of packets or “short messages.” EX1009, 28, §1.2. The ’682 Patent admits that “the transmission of data is known, by way of example, on the basis of the OSI reference model [also known as an “OSI model”] from the International Standards Organization (ISO) has seven layers, each of which has a different functionality (in terms of abstraction).” EX1001, 1:19-25. Further, Tanenbaum, published in 1996, is a prior art textbook regularly used in undergraduate courses and teaches that “[t]wo computers are said

to be interconnected if they [can] exchange information” and notes that “high-resolution color photographs and even short video clips can now easily be transmitted over computer networks.” EX1009, 23, 27, §1.1.3. Various editions of the Tanenbaum textbook have been used in intermediate-level undergraduate courses in computer science, and as such, the material in this textbook would be known to a POSITA. For example, a Fourth or Fifth edition of this textbook was used in 2014 at California State University, Bakersfield.² I note that for the 5th edition of this textbook, Dr. Tanenbaum collaborated with David Wetherall, a professor at the University of Washington, who appears as a co-author. Dr. Wetherall and I spent many years working together in the Telemedia, Networks, and Systems Group at MIT, as we were among the five PhD students in the group. I also note that Nick Feamster, a professor at the University of Chicago, appears as a third co-author on the sixth edition of this textbook. I also previously worked on a project with Dr. Feamster.

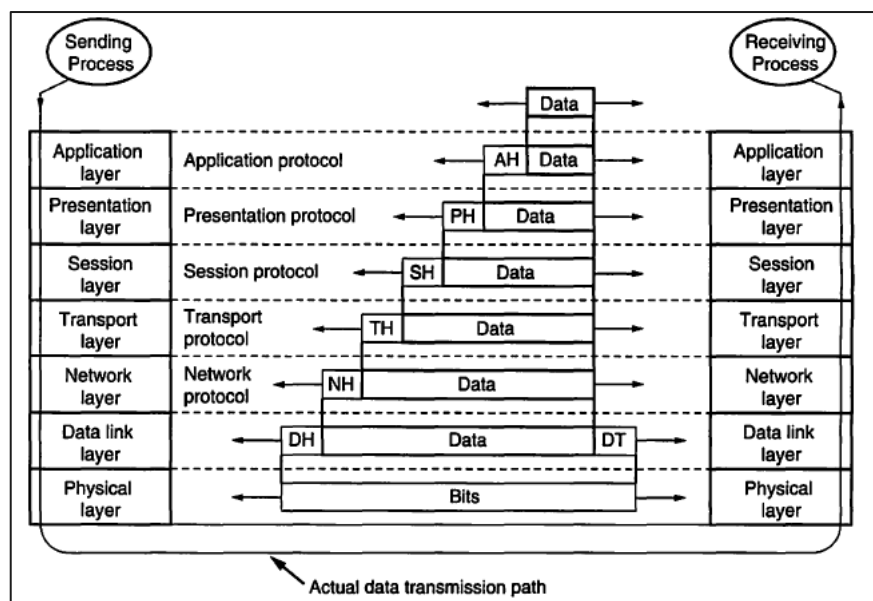
² For example, *see* <https://www.cs.csubak.edu/~melissa/courses.php?course=cs376&quarter=f14&category=info>. This course was required for a Computer Science Track and Computer Information Systems Track around 2008. *See* <https://web.archive.org/web/20081218170838/cs.csub.edu/CS/07-09Catalog.html>. Later, this course was required for a Computer Science Information Security Concentration in 2020. *See* http://www.cs.uccs.edu/~xzhou/teaching/CS4220/CS4220_syllabus_Spring20.pdf

2. OSI Reference Model

48. The OSI Model is “based on a proposal developed by the International Standards Organization (ISO) as a first step toward international standardization of the protocols used in the various layers.” EX1009, 49, §1.4.1. As described in Section IV.B, the ’682 Patent admits the OSI Model was known. Specifically, the ’682 Patent states that:

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. EX1001, 1:25-37.

49. The application layer was known and “contains a variety of protocols that are commonly needed,” *e.g.*, application protocols. EX1009, 54, §1.41. Below is an example of how the OSI model is used.



EX1009, Fig. 1-17

50. As shown, the “key idea throughout is that although actual data transmission is vertical in Fig. 1-17, each layer is programmed as though it were horizontal.” EX1009, 56, §1.4.1.

3. RTP

51. Likewise, RTP was known at the time of the '682 Patent. According to the original RTP Requests for Comments, RFC 1889³, “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.” EX1008, Abstract. The '682 Patent admits “an RTP protocol for transmitting media data (‘payload’), i.e., video or audio data, is known from H.

³ RFC 1889 is included as EX1008 to the Petition.

Schulzrinne: RTP: A Transport Protocol for Real-Time Applications; Internet Engineering Task Force, RFC1889, 1996.” EX1001, 3:11-15.

52. Further, it was known that RTP operates on the session and application layer in the OSI Model, which are often combined with the application layer for purposes of describing an application layer. For example, the original RTP specification, RFC 1889, teaches that “RTP represents a new style of protocol following the principles of application level framing and integrated layer processing,” meaning some protocol functions are implemented in the application. EX1008, 3-4. Additionally, the Complaint in the associated litigation states “the design of RTP is based on the architectural principle known as application-layer framing, where protocol functions are implemented in the application as opposed to the operating system’s protocol stack.” EX1012, ¶63.

4. Authentication

53. Authentication was known in the art before the priority date of the ’682 Patent. In fact, Tanenbaum defined authentication as “[a] technique by which a process verifies that its communication partner is who it is supposed to be and not an imposter.” EX1009, 622, §7.1.5. It was also known in the art that authentication implements encryption techniques (converting plaintext into ciphertext), hash functions (taking input data and producing fixed-size output called a hash value or digest), as well as MAC (a cryptographic checksum). The

'682 Patent admits as much, stating “an asymmetric cryptographic method (also: public key method) is known,” “[a] one-way hash function is known,” and a “message authentication code (MAC) is known generally,” EX1001, 2:12-22, 29-30.

54. Describing one-way hash functions, the '682 Patent confirms that they “cannot be used to calculate the correct input value for a given function value” and that “it must not be possible to use the one-way hash function to obtain the same output value for two different input values.” EX1001, 2:19-24. For a MAC, it “denotes a cryptographic checksum which is intended to be used to identify an alteration in a message or in data.” *Id.*, 2:12-18. Notably, “[o]ne 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.” *Id.*, 2:25-28.

55. Authentication techniques implementing cryptography, *e.g.*, symmetric and asymmetric encryption, were also known in the art prior to the priority date of the '682 Patent. The '682 Patent teaches “[e]ach party involved in an asymmetric cryptographic method receives two keys, a public key and a secret (or private) key.” EX1001, 2:31-32. Using asymmetric cryptography “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.” *Id.*, 2:34-39. In contrast, symmetric encryption “requires a (secret) key used both for encryption and decryption. An example of a symmetric encryption method is the DES algorithm.” *Id.*, 2:40-43. Tanenbaum also devoted entire sections of their textbook to these techniques. *See, e.g.*, EX1009, 608-641.

V. IDENTIFICATION OF CHALLENGES

A. Challenged Claims

56. Claims 1-28 of the ’682 Patent are challenged in the Petition.

B. Statutory Grounds for Challenges

57. The Challenges are set forth in detail below and summarized as follows:

Ground	Challenged Claims	Proposed Statutory Rejection
I	1-21, 23, 25-28	Obvious under § 103 in view of <i>Handley</i> in combination with <i>Basturk</i>
II	1-28	Obvious under § 103 in view of <i>PacketCable</i> in combination with <i>Handley</i>
III	1-28	Obvious under § 103 in view of <i>PacketCable</i> in combination with <i>Basturk</i>
IV	22, 24	Obvious under § 103 in view of <i>Handley</i> in combination with <i>Basturk</i> and <i>PacketCable</i>

VI. OVERVIEW OF THE PRIOR ART

58. I have been informed that *Handley*, *Basturk*, and *PacketCable* qualify as prior art under 35 U.S.C. § 103 for the purposes of this IPR.

A. Handley

59. Handley (EX1005) – *Very Large Conferences on the Internet: The Internet Multimedia Conferencing Architecture and the MBONE*, was published February 11, 1999, in the Computer Networks journal. *Handley* provides an overview of synchronous “real-time” multimedia conferencing, including audio, video, shared whiteboards and other applications over the internet. EX1005, 1. *Handley* teaches that:

[T]he real interest in using the internet for audio and video should come from the prospect for a single ubiquitous communications network that not only allows traditional telephony services, but also video, shared collaboration tools, and through IP Multicast, multi-party conferences and multimedia sessions that scale from small group meetings through to television sized audiences.

Id., 1. Further, *Handley* evaluates how this technology was historically implemented and contemplates future cases by analyzing multicast traffic distribution, internet services models, transport protocols, RTP, conference control, conference setup, and security. For example, *Handley* teaches that secured broadcasts (*i.e.*, data transmissions) include authentication and “can be performed by encrypting a hash (digitally signing) of each packet with the senders private key of a public-private key pair. The public key is then given to the receivers, and they discard (and prune if possible) any packets that are unsigned.” *Id.*, 13.

B. Basturk

60. Basturk (EX1006) – *Secure Transport Protocols for High-Speed Networks*, was published in March 1994, in IBM Research Report RC 19981. I have reviewed and analyzed the Declaration of Dr. Hall-Ellis and understand that *Basturk* was publicly available no later than September 1, 1995. EX1010, ¶41. Stemming from the IBM Thomas J. Watson Research Center, this research report states “that the increasingly widespread use of networks is leading to new security concerns such as secure delivery,” “suggest[s] the design of secure transport protocols as the most effective way of filling all needs, and present[s] a general method to transform a given transport protocol into a secure one.” EX1006, Abstract, 3. *Basturk* teaches its “design puts particular emphasis on maintaining the level of performance of the original transport protocol, particularly in a high speed environment.” *Id.*, 3. *Basturk* teaches the “design of a secure transport protocol,” including authentication, integrity and signatures, confidentiality via encryption, and an “implementation overview.” *Id.*, 6, 15. Notably, *Basturk* teaches that “software encryption and decryption for secure video conferences may be feasible at low resolution. However, the provision of high resolution secure video conferences is likely to require hardware support. Our protocol should be able to operate in both environments.” *Id.*, 5. *Basturk* also teaches data transmission and authentication, where the sender attaches to each packet authentication data, *e.g.*, a

MAC that is generated based off the packet and a shared secret key. *Id.*, 11-12. If the receiver's generated MAC does not match that of the transmitter's, the packet is rejected. *Id.*

C. PacketCable

61. PacketCable (EX1007) – *PacketCable 1.0 Architecture Framework Technical Report*, was published December 1, 1999. Cable Television Laboratories, Inc. (“CableLabs”) is a non-profit Innovation and R&D lab funded by its cable industry member companies around the world, *e.g.*, Comcast, Cox, Vodafone, and Taiwan Broadband Communications. *About CableLabs*, Cable TV Labs, <https://www.cablelabs.com/about-cablelabs>, last visited May 24, 2023. The object of this technical report—PacketCable 1.0 Architecture Framework Technical Report—“is aimed at defining interface specifications that can be used to develop interoperable equipment capable of providing packet-based voice, video and other high-speed multimedia services over hybrid fiber coax (HFC) cable systems utilizing the DOCSIS protocol.” EX1007, 9. *PacketCable* teaches “the objective of the PacketCable Architecture Technical Report is to provide a high level reference framework that identifies the functional components and defines the interfaces necessary to implement the capabilities detailed in the individual PacketCable specifications.” *Id.* *PacketCable* has sections directed to security, RTP, functional components, protocol interfaces, network design considerations,

and future network considerations. *See generally id.* *PacketCable* discloses the transmission of data with authentication where “each media RTP packet is encrypted for privacy” and that “each RTP packet may include an optional message authentication code (MAC).” *Id.*, 47.

VII. IDENTIFICATION OF HOW THE CHALLENGED CLAIMS ARE UNPATENTABLE

A. Ground I: Claims 1-21, 23, and 25-28 Are Rendered Obvious over *Handley* in Combination with *Basturk*

62. In my opinion, Claims 1-21, 23, and 25-28 are rendered obvious over *Handley* in combination with *Basturk*.

1. Motivation to Combine

63. In my opinion, *Basturk*’s disclosure of secure transport protocols, and its known communication benefits, supplements *Handley*’s disclosure of internet conferencing architecture. *Basturk*, which shares the same field of internet data transmission as *Handley*, further informs a POSITA regarding implementing authentication over the internet. A POSITA would have been motivated to modify *Handley* with *Basturk* for many reasons, which I explain below. *Handley* discloses a “standard way to encrypt RTP and RTCP packets using private key encryption schemes,” but does not explicitly state how to perform these efficiently. EX1005, 12. *Basturk* teaches on-the-fly methods of performing the computations that “minimizes latency.” EX006, 7. A POSITA would have understood that minimizing latency was beneficial in processing and networking and would have

been motivated to adopt the teachings of *Basturk* for this purpose. Furthermore, *Basturk* teaches that their design “adopts various design principles that would allow” dedicated hardware “to be exploited in the best possible way.” *Id.* A POSITA would have recognized the desirability of a design that allowed dedicated hardware to be added to speed up the process required in the best possible way. Accordingly, a POSITA would have looked to the teachings in *Basturk* that could be used in the system of *Handley* to better prepare *Handley’s* system for acceleration using dedicated hardware.

64. Although *Handley* does not expressly disclose “inserting, at the transmitter, authentication data at end of a whole RTP packet payload,” combining *Handley’s* disclosure of internet conferencing architecture in combination with *Basturk’s* teaching of secure transport protocols would have been obvious to a POSITA. EX1001 cl. 1. A POSITA would have been motivated to use *Basturk’s* “on-the-fly” signature computation with *Handley’s* internet conferencing architecture due to the known communication benefits, which include faster data transmission, reduced jitter, and reductions in computing power. Furthermore, *Basturk* teaches that its “on-the-fly” computation “minimizes latency” and “facilitates the eventual off-loading to the hardware.” EX1006, 7. A POSITA would have recognized that reducing the latency in a live transmission provides a more natural conversational flow and reduces interruptions. For example,

conversations that happen over high latency connections, such as television interviews over satellite links, often seem awkward with long pauses. People are trained to wait for a reply, but those who try to have a normal conversation over such a link often speak over each other, with people wanting to talk in order to fill silent periods, then realizing the other person had started to speak. Furthermore, a POSITA would recognize that offloading certain computations to hardware allow for faster, uninterrupted process, which reduces jitter by having fewer interruptible periods where a process may be pre-empted, and that offloading computation to hardware would free up compute cycles.

65. For example, *Handley* recognized that receivers need to efficiently and securely process data and discloses that “[r]eal-time applications such as audio and video need to be able to buffer real-time data at the receiver for sufficient time to remove the jitter added by the network.” EX1005, 8. Similarly, *Basturk* discloses that “[i]deally, we want signatures to be computed and verified on-the-fly; i.e., MAC generation should be carried out as data transmissions proceeds rather than before these operations have completed” because “[t]he goal is to avoid slowing down the regular flow of data.” EX1006, 12 (cleaned up). For example, the data transmission occurs faster using on-the-fly signature computation because the signature computation is done in parallel with the transmission of the packet, and then the computed signature is placed at the end of the packet just as the

transmission is completing. Compare this to where the signature is computed before transmission, then only after the signature is computed does the packet start to be transmitted with the computed signature. The packet will be transmitted later than an on-the-fly signature computation, by the duration of time it takes to compute the signature. As *Basturk* states, “[t]his minimizes latency.” EX1006, 7.

66. A POSITA would have also appreciated that although *Basturk* does not explicitly disclose RTP, RTP’s secure transport protocol is compatible with *Handley’s* internet conferencing architecture, as *Handley* specifically considers security issues, devoting an entire section to security concerns. EX1005, 11-14. For example, *Basturk* discloses that its proposed protocol “should be able to operate in both environments,” including “software encryption and decryption for secure video conferences” at both low- and high-resolution environments. EX1006, 5. A POSITA would have understood that RTP is commonly used for video conferences, as “RTP is primarily designed to satisfy the needs of multi-participant multimedia conferences” and that the original RTP Request for Comments, RFC 1889, “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.” EX1008, 3. Further, *Basturk* teaches its security features “reflect[] the need to add security functionality in a way which is consistent with, and best

matches, the existing communication structure” and so that “[a]pplications using the regular transport protocol [will] not have to be modified at all.” EX1006, 3, 5.

67. A POSITA would have understood that the systems of *Handley* and *Basturk* are compatible and related because *Handley* describes a scenario in which authentication is needed in a multicast environment:

This is nicely illustrated by the multicast of the Rolling Stones concert in 1994 which was billed as being the first live concert on the Mbone. In fact, this honour goes to a little known band called Severe Tire Damage who had multicast an impromptu concert a year previously. To make their point, just before the Stones were due to go on stage, Severe Tire Damage suddenly (sic) started broadcasting one of their songs live to the same multicast group. ***Clearly commercial broadcasters want to avoid occurrences like this one.***

EX1005, 13.

68. *Basturk* describes a similar scenario in which “[t]wo parties, denoted *A* and *B*, communicate over a network. We will naturally view the parties as applications who use a common transport protocol. An adversary *E* monitors the parties’ communications, and is even capable of injecting or removing messages from the lines.” EX1006, 3. Both *Handley* and *Basturk* aim to remove the possibility of a third-party interfering with communication between a defined set of parties. Thus, a POSITA would have understood and found it obvious to implement an authentication solution for methods of secured data transmission presented in *Basturk* to the same problem presented in *Handley*. *Handley* specifically discusses unsecured transmission situations in which “will not be

acceptable for many broadcasters who require the ability to ensure that only they can send, perhaps in addition to ensuring that only their paid customers can receive.” EX1005, 13. Specifically, a POSITA would have understood and found obvious that *Basturk’s* authentication methods would have prevented unauthenticated interruption of video transmissions, like that in the Rolling Stones concert. *Handley* teaches that “[s]uch secured broadcasts can be performed by encrypting a hash (digitally signing) of each packet with the senders [sic] private key of a public-private key pair. The public key is then given to the receivers, and they discard (and prune if possible) any packets that are unsigned.” EX1005, 13. A POSITA would understand that a hash encrypted using the sender’s private key and decrypted using the sender’s public key could be checked against the packet received using the same hash algorithm. If the decrypted hash and the hash of the packet calculated at the receiver do not match, then either the packet was forged at the source, or the packet was altered in transit, which in either case indicates the packet should be discarded. Those who try to hijack the media session cannot send to those in a media session because the hijacker does not know the sender’s private key and thus, cannot sign the hash as the real sender would, meaning all their packets would be discarded by the receivers.

69. Using *Handley’s* internet conferencing architecture and *Basturk’s* secure transport protocol, a POSITA would have understood that hardware to

implement multi-media transport may be necessary to keep up with the high volume of packets generated in video applications and could include dedicated cryptographic hardware as taught by *Basturk*. EX1006, 7.

70. In my opinion, use of *Handley's* internet conferencing architecture in conjunction with *Basturk's* secure transport protocol would have made use of known multi-media data transport architecture and a known authentication method to improve data transmission security. The combination also applies a known technique to a known device or method ready for improvement to yield predictable results. Using an internet conferencing architecture in combination with a secure transport protocol involves the use of routine computers, equipment, and techniques, and a POSITA would have had a reasonable expectation of success in using "on-the-fly" data authentication with *Handley*. In my opinion, a POSITA would have found the use and implementation of such hardware to implement *Basturk's* authentication methods with the internet conferencing architecture obvious, as video conference applications in particular may generate large numbers of packets and signing packets at a high rate may require the use of hardware acceleration.

2. Independent Claim 1

a) **Preamble [1]**

71. As discussed above in Section IV.G.1, the '682 Patent concedes that data transmission, or the sending of data, was known in the art. Despite said admissions, and to the extent it is a limitation, it is my opinion that *Handley* disclosed, or at least rendered obvious, the preamble of Claim 1.

72. *Handley* teaches that “multimedia information must be transmitted in packets” and that traditionally, “the internet has provided so-called *best-effort* delivery of datagram traffic from senders to receivers.” EX1005, 2, 4. Specifically, *Handley* discloses synchronous or “real-time” conferencing “including audio, video, shared whiteboards and other applications.” EX1005, 1. In my opinion, a POSITA would have understood that the multimedia described in *Handley* can be transmitted over the internet in the form of packets and that transmitting data over the internet requires a transmitter and receiver. Further, a POSITA would have understood that a “sender” in the context of *Handley* is the equivalent to a transmitter because those terms are used interchangeably when describing multimedia conferencing networks. Although “sender” may sometimes be read more broadly than “transmitter,” it is my opinion that “sender” and “transmitter” in the context of computer networks is very close and may be used interchangeably. Thus, *Handley* discloses “a method for transmitting data from a transmitter to a

receiver” because multimedia information, *i.e.*, data, is transmitted by “senders to receivers” in packets over the internet. EX1001, cl. 1; EX1005 at 4.

73. The preamble is further obvious in view of *Handley* in combination with *Basturk*. *Basturk* discloses that “[t]wo parties, denoted *A* and *B*, communicate over a network...who use a common transport protocol.” EX1006, 3. In my opinion, a POSITA would have understood that networks support protocol hierarchies which are used to transmit data and that networks contain transmitters and receivers. Further, a POSITA would have understood that parties *A* and *B* in *Handley* may be transmitters or receivers because they both “communicate over a network” with one another, and such communications such as exchanging keys involve sending data back and forth to each other at least to acknowledge receipt of data. EX1006, 3. Thus, it is my opinion that *Handley* in combination with *Basturk* discloses “a method for transmitting data from a transmitter to a receiver.”

b) **Element [1.1]**

74. See Ground I, Preamble [1],

75. First, as discussed above in Section IV, the ’682 Patent admits that each claimed technology including RTP, authentication, and the OSI Model (which includes an application layer and application protocol), were known in the art. Second, despite said admissions, it is my opinion that *Handley* disclosed this element, or at least rendered it obvious.

76. Beginning with RTP, *Handley* teaches RTP as a “transport protocol for real-time flows” which “provides a standard format packet header which gives media specific timestamp data, as well as payload format information and sequence numbering amongst other things.” EX1005, 8. According to the original RTP Requests for Comments, RFC 1889, “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.” EX1008, 4. A POSITA would have understood RFC 1889 to mean RTP packets are created at the application layer and use an application protocol because RTP is integrated into the application processing, and the media application generating the audio or video stream is creating the RTP packets as different applications require different packet sizes and different packet generation rates. For example, live conversational audio applications (e.g., a telephone call or teleconference) require low packetization latency, and packets may contain as little as 10 to 20 milliseconds (82 to 164 bytes) of audio per packet, but applications that stream pre-recorded audio do not require such small packets and can use the full payload’s worth of data (around 1500 bytes) per packet. As such, in my opinion a POSITA would have understood this to mean the media application generating the audio or video stream is creating the RTP packets at the application layer in order that an application create the RTP packets according to the application’s needs.

77. Next, RTP utilizes the known OSI Model and, as described above in Section IV.G.1, RFC 1889 describes RTP as “represent[ing] a new style of protocol following the principles of application level framing and integrated layer processing.” EX1008, 4. A POSITA would have understood and found obvious that because RTP follows the principles of application level framing and integrated layer processing and that the RTP packets are created at the application layer and RTP is an application protocol. *See* EX1015.⁴ Thus, *Handley* teaches “Real Time Transport Protocol (RTP) packet level as an application protocol on an application layer” due to its disclosure of RTP. EX1001, cl. 1.

78. *Handley* discloses “transmitter-to-receiver authentication.” *Handley* discloses that “security measures may be required to actually enforce the conference policy, e.g., to control who is listening and to ***authenticate*** contributions as actually originating from a specific person.” EX1005, 2. Further, *Handley* discloses that if an application requirement, e.g., conference policy, “is to communicate between some defined set of users” then “strict privacy can [] be enforced... through adequate end-to-end encryption.” *Id.*, 12. *Handley* also discloses multiple types of transmitter-to-receiver authentication techniques that

⁴ This is the seminal Application Layer Framing paper, and I was a member of the research group headed by both Dr. Clark and Dr. Tennenhouse. The research group later split into two research groups, and I remained with Dr. Tennenhouse, who was also my Ph.D. Thesis advisor.

take place at the RTP packet level. For example, *Handley* teaches that “RTP specifies a standard way to encrypt RTP and RTCP packets using private key encryption schemes such as DES,”⁵ and “also specifies a standard mechanism to manipulate plain text keys using MD5.”⁶ *Id.*, 12. In my opinion, a POSITA would have understood that *Handley* discloses the implementation of authentication at the RTP packet level because using encryption schemes like DES or MD5 to manipulate the RTP packet directly occurs at the RTP packet level since *Handley* teaches that the encryption is specified by RTP to encrypt RTP packets and not at any other layer in the stack. *Id.*

79. To the extent *Handley* does not expressly disclose Element [1.1], it is my opinion that this element would have been obvious in view of *Handley* in combination with *Basturk*.

80. As described above in Section VII.A.1, a POSITA would have found it obvious to implement *Basturk*’s secure transport protocol and its authentication techniques to *Handley*’s internet conferencing architecture. Regarding said authentication, *Basturk* discloses:

⁵ DES is a symmetric-key block cipher encryption algorithm used to secure electronic data by converting plaintext to ciphertext.

⁶ MD5 is a cryptographic hash function that generates a fixed-size output from an input message of any size, used for verifying the integrity of the data and detecting any changes or tampering with the original message.

secure transport protocol has two phases, the initialization phase and the steady state. In the former, an entity authentication and key exchange protocol is executed. This authenticates the parties and yields them a session key α . In the steady state, if the integrity option is selected, both data integrity and secure delivery are provided by attaching to **each packet a signature**, computed under α , and taken over some portions of the header and the data payload.

EX1006, 5.

81. Further, *Basturk* teaches that when machine A (“ M_A ”) and machine B (“ M_B ”) wish to send and receive data with the integrity option selected, then “data is **signed** by M_A under the session key α (a signature is also called a MAC, or message authentication code). The signature covers not only the data payload but also the transport header (as explained later). On receiving a signed packet, M_B uses α to verify the signature.” EX1006, 9.

82. Additionally, *Basturk* discloses that “[a]pplication level security features are easily (and usually) provided by **application level security**. Here the sender applies appropriate cryptographic functions to his data **before** passing it to the transport for conveyance” and that “[i]ntegrity is usually guaranteed by **signing** the data before passing it to the transport level.” EX1006, 3-4. A POSITA would have understood that key elements of *Basturk’s* authentication technique takes place as an application protocol on an application layer because even in “transport level security” described in *Basturk*, e.g., cryptographic functions such as key exchange and selection of a key to use for a communication, occur at the

application layer, since such keys are passed from the application layer to the transport layer. *Id.*, 4-5.

c) **Element [1.2]**

83. See Ground I, Preamble [1] and Element [1.1].

84. It is my opinion that Element [1.2] would have been obvious in view of *Handley* in combination with *Basturk*. *Basturk* discloses that, for performance, it is “essential that the steady state overhead involved in computing and checking signatures and counters be kept to a minimum.” EX1006, 12. To keep computing to a minimum, which is a necessity in multi-media transport for delay and buffering purposes, *Basturk* states “[i]deally, we want signatures to be computed and verified on-the-fly; i.e., MAC generation should be carried out as data transmissions proceeds rather than before these operations have completed” because the “goal is to avoid slowing down the regular flow of data.” *Id.* (cleaned up). *Basturk* further explains that:

[c]onfidentiality means user data should not be readable by the adversary. The secure transport protocol will encrypt user data, under the session key, and use the ciphertexts as the actual payload. As encryption is done prior to packetization, the signature (if present) of an outgoing packet is computed after data encryption and the signature of an incoming packet is computed before decryption. This complies with the requirement of a signature computed ‘on the fly’. If the signature checking at reception fails, it means the the [sic] packet is corrupted, and the packet is simply dropped: decryption is avoided.”

Id., 14.

85. As an example of “on-the-fly” signature implementation, *Basturk* discloses that “[a] security segment has been added to include information needed or authentication and confidentiality” and that “[f]ollowing data, a trailer has been added that contains the packet signature.” EX1006, 16. A POSITA would have understood and found obvious that a “trailer” is data that can be added the end of the packet, much as the header is data added at the beginning of a packet, and that a trailer may be defined to contain authentication data, such as the packet signature, much like a header can be defined to include data fields such as address information. Although *Basturk* does not explicitly disclose RTP, a POSITA would have understood and found obvious that when combined with *Handley’s* RTP internet conferencing architecture, the trailer, *i.e.*, authentication data or MAC from *Basturk*, would be placed at the end of the whole RTP packet payload to enable “on-the-fly” signature computation and verification, allowing the transmitter to transmit data while simultaneously computing the authentication data. EX1005, 12; EX1006, 12. A POSITA would understand that the data transmission occurs faster using on-the-fly signature computation because the signature computation is done in parallel with the transmission of the packet, and then the computed signature is placed at the end of the packet just as the transmission is completing. Compare this to where the signature is computed before transmission, then only after the signature is computed does the packet start

to be transmitted with the computed signature. The packet will be transmitted later than an on-the-fly signature computation, by the duration of time it takes to compute the signature. As *Basturk* states, “[t]his minimizes latency.” EX1006, 7. For example, the Ethernet standard, first published in 1985, employs the same design choice reasoning to include a “Frame Check Sequence” (“FCS”) at the end of the Ethernet frame, so that it can be computed during transmission and appended to the end of the frame, and it can be computed upon reception to check against the FCS received and discarded if there is not a match. See EX1014, 43, 46, 41, 23-24, 26, 27, Fig 3-1.

d) **Element [1.3]**

86. See Ground I, Preamble [1], Element [1.1] and [1.2].

87. In my opinion, *Handley* disclosed this element or at least rendered it obvious. A POSITA would have understood and found obvious that a receiver would ascertain whether it “knows” if the transmitter is using either symmetric or asymmetric key encryption and, further, that for symmetric key encryption, the transmitter and receiver exchange a secret key prior to data transmission for data encryption and decryption. Thus, in order to exchange symmetric keys securely, the two parties would have to have knowledge of each other before any encrypted data can be exchanged. For symmetric encryption, *Handley* discloses that “RTP specifies a standard way to encrypt RTP and RTCP packets using private key

encryption schemes such as DES.” EX1005, 12. As described above in Ground I; Element [1.1], a POSITA would have understood that RTP encryption occurs at the packet level because DES and other encryption schemes encrypt the RTP packets directly. A POSITA would understand that in real time streaming uses for which RTP was designed, packets would be decrypted on a packet-by-packet basis as waiting to aggregate packets before decryption would only cause unnecessary delays to access data without any real benefit. In addition, if data were encrypted in higher layer bundles of data, errors or losses in the network affecting just one packet would cause the loss of data of the whole bundle of data, since all the encrypted data would be needed in order to decrypt the data or check its signature. A POSITA would have further understood that DES is a symmetric-key algorithm, meaning both the transmitter and the receiver must know and use the same secret key to both encrypt and decrypt the data, and this can only be done securely by exchanging the secret key before the data transmission. Further, a POSITA would have understood and found obvious that if the receiver knows the transmitter’s secret key for the sessions, then the receiver knows the transmitter’s identity because the transmitter and receiver would have previously established a secret key before they could encrypt and decrypt the data. Since the receiver needs to know which secret key to use for encrypted data received from a different transmitter, each party stores secret keys in association with information identifying the

transmitter with which the secret key is associated, and thus, the receiver already knows of the transmitter when it receives the first encrypted data from the transmitter.

88. For asymmetric encryption, a POSITA would have understood it to involve the use of public and private keys, whereby the transmitter encrypts the data using the public key of the recipient, such that only the recipient can decrypt the data using its private key. Private keys are a closely held secret and thus are only known by the entity using its private key. The corresponding public key can be used to decrypt data using the private key. The public key can be distributed to anyone, and to send data securely to a recipient, the public key is used to encrypt the data. Because the public key and corresponding private key are cryptographically related, the private key can be used to decrypt data encrypted with the corresponding public key. It follows that only the party with the corresponding private key can decrypt data using the corresponding private key, and private keys are a closely held secret; only the intended recipient can decrypt a message encrypted using a public key. *Handley* discloses that “secured broadcasts can be performed by encrypting a hash (digitally signing) of each packet with the senders private key of a public-private key pair. The public key is then given to the receivers and they discard (and prune if possible) any packets that are unsigned.” EX1005, at 13. In this asymmetric encryption digital signature scenario, a POSITA

would have understood and found obvious that if the receiver already has the transmitter's public key stored, *e.g.*, in its key cache, then the receiver knows who the transmitter is because the receiver would have stored the transmitter's public key to avoid the overhead of retrieving the public key every time a secure communication is initiated. Because entities can exchange encrypted data with numerous other entities, the key cache needs to store public keys in association with information identifying the entity with which the public key is associated. In some public/private key systems, public keys are stored in a certificate which also contains the identifying information, and the locally cached certificates alleviate the need to locate the certificate each time it is needed. Certificates also often comprise expiration dates, so certificates can be cached locally until the expiry time.

89. In both the symmetric or asymmetric encryption implementation, a POSITA would have understood and found obvious that the receiver is ascertaining, based on RTP-packet-level authentication data, whether it knows the transmitter because it either has a shared key to verify the RTP level authentication data in the symmetric implementation or, in the asymmetric implementation, has stored the transmitter's public key in its key cache to check the authentication data. In either case, the receiver needs to retrieve the proper key to use a symmetric key associated with the transmitter, or in an asymmetric implementation, the public key

associated with the transmitter. If the appropriate key cannot be located, then the receiver did not know the transmitter prior to the receipt of the signed packet, and the packet's signature cannot be verified.

90. To the extent *Handley* does not expressly disclose this element, this element would have been obvious in view of *Handley* in combination with *Basturk*. *Basturk* first discloses a symmetric key implementation, whereby “parties wish to be able to communicate ‘securely’” and use a “shared secret key a (e.g., a DES key)” to do so. EX1006, at 3. Further, *Basturk* discloses that for a message authentication code under session key “ α ”, denoted as “ MAC_α ,” a transmitter “attaches to each packet p the value of $\text{MAC}_\alpha(\bar{p})$ where $\bar{p}$ is the subset of p which includes the transport header, the full data payload, and the trailer of the packet p ” and that “the transport header contains information which identifies the sender so that the identity of the sender is covered by the MAC as well.” EX1006, 12. A POSITA would have understood and found obvious that the MAC described in *Basturk*, when applied to *Handley*'s RTP internet conferencing architecture, would be generated based off of the RTP packet because the MAC is generated from $\bar{p}$ “which includes the transport header, the full data payload, and the trailer of the packet p ” and, therefore, it would have been obvious that packet p would be an RTP packet. Further, a POSITA would have understood and found obvious that the MAC would be based on RTP packet level authentication data because, as I

described in Ground I; Elements [1.1] and [1.2], the MAC is generated from the RTP packet data and is inserted at the end of the whole RTP packet payload. A POSITA would understand that in real time streaming uses for which RTP was designed, packets would be decrypted on a packet-by-packet basis since waiting to aggregate packets before decryption would only cause unnecessary delays to access data without any real benefit. In addition, if data were encrypted in higher layer bundles of data, errors or losses in the network affecting just one packet would cause the loss of data of the whole bundle of data, since all the encrypted data would be needed in order to decrypt the data or check its signature.

91. As described above, a receiver knows a transmitter if the two share a secret key or if the receiver has the transmitter's public key stored. *Basturk* discloses that "data is *signed* by M_A under the session key α (a signature is also called a MAC)" and "[o]n receiving a signed packet, M_B uses α to verify the signature." EX1006, 9. *Basturk* describes this as "[o]n receiving p , τ the receiver checks that $\tau = \text{MAC}_\alpha(\bar{p})$ " whereby p is the packet and τ is the MAC signature. EX1006, 12. As such, a POSITA would have understood and found obvious said teaching to mean that the receiver uses the session key α (e.g., a symmetric DES key) is used to verify the MAC signature (i.e., compute a MAC based off the data received and use the same session key α to see if the computed MAC matches the received MAC) which, as described above, is generated based of RTP packets and

the session key α , making it RTP packet level authentication data. *Basturk* clarifies that said teaching is not limited to symmetric encryption but states that “for simplicity we stick to this symmetric case. Our methods can be modified to work in the public key scenario as well.” EX1006, 3 n.2.

e) **Element [1.4]**

92. See Ground I, Preamble [1], Elements [1.1], [1.2], and [1.3].

93. In my opinion, *Handley* disclosed this element or at least rendered it obvious. As discussed in Element [1.3], a POSITA would have understood and found obvious that, in using symmetric encryption, a receiver knows a transmitter based on the shared secret key and, in using asymmetric encryption, a receiver knows the transmitter if the receiver has stored the transmitter’s public key. To exchange symmetric keys securely, the two parties would have to have knowledge of each other before any encrypted data can be exchanged. *Handley* discloses that, in the asymmetric encryption use case, “[t]he public key is [] given to the receivers, and they *discard* (and prune if possible) any packets that are unsigned.” EX1005, 13. As I describe below, a POSITA would have understood and found obvious that if the receiver discards unsigned packets, or if the signature does not match, then the receiver does not know the transmitter and the whole RTP packet payload is rejected. If the signatures do not match, then the packet was either forged or altered in transit, and in any case should be discarded. Further, a POSITA

would have understood and found obvious that if the packets are not “discard[ed],” then the receiver is “accepting” the whole RTP packet payload because the packets are then decrypted and processed. EX1005, 13. If the signatures do match, a POSITA would have understood that the data came from the transmitter and was not altered in transit and is thus authentic and should be accepted. A POSITA would understand these basic concepts. For example, a POSITA receiving bills in the mail should not expect to pay a bill from an entity they did not recognize as providing a service to which they subscribe (unknown sender or unsigned packet) or was noticeably forged (unsigned packet or signature does not match). These basic concepts were known even to lay persons.

94. To the extent *Handley* does not explicitly disclose this element, this element would have been obvious in view of *Handley* in combination with *Basturk*. As described in Ground I, Elements [1.1] and [1.3], *Basturk* discloses the MAC is computed using a DES session key α in addition to the packet’s contents. EX1006, 12. *Basturk* further discloses that “[o]n receiving p , τ the receiver checks that $\tau = \text{MAC}_{\alpha}(\bar{p})$ and **rejects** the packet unless this is true.” EX1006, 12. As described above, p is the packet and τ is the MAC signature. This signature “prevents the adversary from making the receiver accept packets which were not in fact produced by the sender.” *Id.* Further, *Basturk* discloses “[i]f the signature checking at reception fails, it means the the [sic] packet is corrupted, and the

packet is simply dropped: decryption is avoided.” *Id.*, 14. A POSITA would have understood and found obvious that when combining *Basturk’s* authentication techniques with *Handley’s* RTP internet conferencing architecture that the receiver “rejects” the whole RTP packet payload because it would not accept a packet from a transmitter it does not know, or a packet that failed the authentication and thus would have been forged in the first instance or modified en route. EX1005, 11-12. A POSITA would have understood that a failed signature match indicates that the packet did not come from the other party using the appropriate key or was altered in transit, causing the hash of the data to be different from the signed hash received.

3. Dependent Claim 2

95. See Ground I, Claim 1.

96. In my opinion, *Handley* disclosed this Claim or at least rendered it obvious. A POSITA would have understood and found obvious that a PDU is a single unit of information at a particular OSI layer (*e.g.*, an RTP packet at the application layer). A POSITA would have understood that a media application, such as in a conference application, generates RTP packets at a steady rate, and the protocol data units are the RTP data in each packet. *Handley* discloses that “secured broadcasts can be performed by encrypting a hash (digitally signing) of each packet with the senders private key of a public-private key pair. The public

key is then given to the receivers and they discard (and prune if possible) any packets that are unsigned.” EX1005, at 13. A POSITA would appreciate that the PDU, e.g., an RTP packet, is linked to a secret in two possible ways. First, the RTP packet may be encrypted, and is linked to the secret via the keys used to encrypt and decrypt the RTP packet, since the correct key must be used to properly decrypt the packet. Second, the RTP packet may be used as the basis of an authentication code, e.g., digital signature, and is linked to the secret when the hash is encrypted since the correct key must be used to decrypt the signature and retrieve the correct hash. *Handley*, by way of example, states, “[n]ote that a conference policy that restricts conference membership can be implemented using encryption and restricted distribution of encryption keys.” *Id.*, 9 n.2. Further, *Handley* discloses that if “the application requirement (conference policy) is to communicate between some defined set of users, then strict privacy can only be enforced in any case through adequate end-to-end encryption.” *Id.*, 12. A POSITA would have understood and found obvious that because the distribution of keys is handled as a conference policy, it is, therefore, handled at the application layer, since the layers below the application layer (transport layer and below) do not track the specifics of any particular application’s policies; the lower layers are simply to transport and deliver the data provided by the applications. Basturk confirms that the applications track the keys which are then used by the lower layer network

modules to perform encryption as needed: “TRANSPORT LEVEL SECURITY. In this design, the application passes its key *a* to the transport module and leaves to the latter the task of applying any cryptographic functions or taking other security steps.” EX1006, 4.

97. To the extent *Handley* does not expressly disclose this element, this element is disclosed in view of *Handley* in combination with *Basturk*. *Basturk* discloses that:

the application passes its key *a* to the transport module, and leaves to the latter the task of applying any cryptographic functions or taking other security steps. The advantage is that the transport module can compute a signature which covers not only the data but also (relevant parts of) the transport header which it creates, particularly the byte sequence numbers. It can also sign the acknowledgements messages which exist only at the transport level. As we will see, these possibilities are key to the provision of secure delivery.

EX1006, 4-5.

98. A POSITA would have understood the key is “protocol data unit available at the application layer” because the key is passed from the application to the transport module using an appropriate protocol so the data is recognized as a key and not just as data. Further, as discussed in Ground I, Claim 1, a POSITA would have understood and found obvious that because the key is a secret key used by a transmitter to sign each packet in the communication, it discloses “the application protocol being linked to a secret by the transmitter” because the key *a* itself is the secret itself that is used to perform appropriate decryption or encryption

operations at a lower layer. Alternatively, the secret is a signing process comprising generating a hash of data then encrypting the hash with the secret key a that is known only to the receiver and the transmitter to be authenticated.

4. Dependent Claim 3

99. See Ground I, Elements [1.1], [1.3], and Claim 2.

5. Dependent Claim 4

100. See Ground I, Claim 2.

101. In my opinion, *Handley* disclosed this Claim or at least rendered it obvious. *Handley* discloses the “transport protocol for real-time flows is RTP” and that RTP “provides a standard format packet header which gives media specific timestamp data, as well as payload format information and sequence numbering.” EX1005, 8. A POSITA would have understood that RTP inherently includes sequence numbers and timestamps, as adherence to the RTP RFC 1889 require these fields to be in the header. For example, RFC 1889 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.” EX1008, 3, §1. RFC 1889 further discloses that “the RTP header contains timing information and a sequence number that allow the receivers to reconstruct the timing produced by the source” and “RTP Fixed Header Fields”

including “sequence number” and “timestamp” as these are required header fields as specified by the RTP RFC 1889. EX1008, 5, §2.1, 10, §5.1. From this, a POSITA would have understood and found obvious that the sequence and timestamp are included in the RTP packet header and that the whole RTP packet payload is a PDU on the application layer. Further, a POSITA would have understood that a media application, such as in a conference application, generates RTP packets at a steady rate, and the protocol data units are the RTP data in each packet. As discussed in Element [2.1], a POSITA would have understood that a protocol data unit would be any part of a protocol exchange and, because RTP is available at the application layer in order for the application to decode and reassemble any real time data, *Handley* discloses this element.

102. To the extent *Handley* does not disclose this element, it would have been obvious in view of *Handley* in combination with *Basturk*. *Basturk* discloses that “[w]e will see that of particular importance is the fact that the transport header [of the packet] includes the sequence number of the packet which is thereby covered by the MAC.” A POSITA would have understood and found obvious that a packet on the application layer is a PDU, and the MAC that is included with the packet includes a sequence number.

6. Dependent Claim 5

103. See Ground I, Claim 2.

104. In my opinion, *Handley* disclosed this Claim or at least rendered it obvious. A POSITA would have understood and found obvious that “a key known only to the receiver and the transmitter to be authenticated” is implementing symmetric encryption where the transmitter and receiver share a secret key. *Handley* discloses that a hash “secured broadcasts can be performed by encrypting a hash (digitally signing) of each packet with the senders [sic] private key of a public-private key pair” and that “[s]ession keys can be distributed using encrypted Session Descriptions carried in SIP session invitations, or in encrypted session announcements as described below,” which describes a key known only to the receiver and transmitter. EX1005, 12, 13. A POSITA would have understood this to mean the secret is the use of both one-way hash function, as well as encryption of the resulting hash with the session key that is known only to the receiver and the transmitter to be authenticated by decrypting the hash.

105. *Handley* further discloses that “RTP specifies a standard way to encrypt RTP and RTCP packets using private key encryption schemes such as DES. It also specifies a standard mechanism to manipulate plain text key using MD5 so that the resulting bit string can be used as a DES key.” EX1005, 12. A POSITA would have understood that the secret would be the combination of MD5, a known one-way hash function, and DES, a known symmetric encryption algorithm, that uses keys known only to the parties involved and keeping the

underlying data encrypted in this manner secret from all parties who do not possess the secret key, since the proper secret key must be used to decrypt any data or signature.

106. To the extent *Handley* does not disclose this element, it would have been obvious in view of *Handley* in combination with *Basturk*. *Basturk* discloses that “[b]oth signature computation and encryption/decryption rely on the use of secret information (key) shared by the two communicating parties.” EX1006, 8. Regarding one-way hash functions, *Basturk* discloses “[a] library of fast implementation of cryptographic functions has been used in order to implement the encryption algorithm and the signature scheme. Currently, MD5, SHA, DES, Double DES are the functions the upper layer can choose from.” *Id.*, 16. A POSITA would have understood that the secret is the encryption operation of DES or Double DES (which utilizes a secret key known only to the transmitter and receiver and were known in the art), and the one-way hash function is MD5 or SHA (which were each known in the art at the relevant time).

107. *Basturk* also discloses the use of message authentication codes. For example, if “the integrity option is selected then data is signed by M_A under the session key α (a signature is also called a MAC, or message authentication code). The signature covers not only the data payload but also the transport header (as explained later). On receiving a signed packet, M_B uses α to verify the signature.”

EX1006, 9. A POSITA would have understood this to mean α is a secret key known only to the transmitter and receiver which is used to generate a MAC. Further, *Basturk* discloses that “MAC computations are kept simple and can start prior to context retrieval. In particular, we rely on MAC implementations of the form $MAC_{\alpha}(m) = f_{\alpha}(h(m))$, where h is a (hash) function depending neither on α nor on the cryptographic primitive chosen in the initial options.” *Id.*, 13. A POSITA would have understood that “ h is a (hash) function,” m is “a message m ” and accordingly, the message is placed through a one-way hash function like MD5 or SHA, and the resulting hash is then encrypted using f_{α} , which may be a number of “cryptographic functions.” *Id.*, 8, 13. A POSITA would recognize this the textbook definition of a signing operation.

7. Dependent Claim 6

108. See Ground I, Claim 2.

109. In my opinion, *Handley* disclosed this Claim or at least rendered it obvious. As discussed in Ground I, Elements [1.1], [1.3], and [5.1], *Handley* discloses the use of symmetric and asymmetric encryption/decryption as well as the use of MD5. A POSITA would have understood that the ascertaining in a pure encryption/decryption scenario utilizing only keys and no cryptographic checksum would stem from whether the receiver knows the transmitter’s secret key or has cached the receiver’s public key. In a scenario using both an encryption/decryption

operation and a cryptographic checksum, a POSITA would have understood that (1) the transmitter runs data through a hash algorithm to create a message digest; (2) the digest is encrypted to create a digital signature using either a secret key or private key; (3) the data and digital signature are sent to the receiver; (4) the receiver decrypts the digital signature using a secret key or the transmitter's public key to obtain the message digest; (5) the data is run through the same hash algorithm as the transmitter did; and (6) the output of the hash algorithm is compared to the decrypted message digest. If the decrypted hash and generated hash match, a POSITA would have understood that the packet has been cryptographically verified and thus is authentic.

110. To the extent *Handley* does not expressly disclose this element it is my opinion that this element would have been obvious over *Handley* in combination with *Basturk*. As discussed in Ground I, Elements [1.1], [1.3], and Claim 5, *Basturk* discloses the implementation of MAC signatures, hash functions, cryptographic checksums, as well as asymmetric or symmetric encryption and decryption that are used to authenticate and ascertain whether the receiver knows the transmitter. A POSITA would have understood that the signature must also be decrypted because the signature was originally encrypted and, in the symmetric scenario, utilizes the same key it was encrypted with, and that if this decrypted

value matches a locally generated hash, the packet will have been cryptographically verified.

8. Dependent Claim 7

111. See Ground I, Claims 6 and 2.

9. Dependent Claim 8

112. See Ground I, Claim 2.

113. In my opinion, *Handley* disclosed this Claim or at least rendered it obvious. As discussed in Ground I, Elements [1.1], [1.3] and Claim 5, *Handley* discloses the use of MD5 and asymmetric and symmetric encryption, *e.g.*, DES. As described in Ground I, Claim 6, a POSITA would have understood that the linking to the secret is the use of MD5 to create a hash or cryptographic checksum and DES with the secret key to encrypt the data because only the parties that have the secret key can properly decrypt the data, keeping the data secret from other non-parties.

114. To the extent *Handley* does not expressly disclose this element, this element would have been obvious in view of *Handley* in combination with *Basturk*. As I described in Ground I, Elements [1.1], [1.3] and Claim 6, *Handley* discloses the use of MAC signatures, MD5, SHA, and asymmetric and symmetric encryption. EX1005, 13, 16. A POSITA would have understood and found obvious the linking to the secret is the use of MD5/SHA to create a hash or cryptographic

checksum and a MAC with secret keys to encrypt the data because only the parties that have the secret key can properly decrypt the data, keeping the data secret from other non-parties, since non-parties will not have access to the proper key necessary to decrypt the data.

10. Dependent Claim 9

115. See Ground I, Elements [1.1], [1.3], and Claim 8.

11. Dependent Claim 10

116. See Ground I, Claims 4 and 8.

12. Dependent Claim 11

117. See Ground I, Claims 5 and 8.

13. Dependent Claim 12

118. See Ground I, Element [1.3] and Claim 8.

14. Dependent Claim 13

119. See Ground I, Elements [1.1], [1.3], and Claim 12.

15. Dependent Claim 14

120. See Ground I, Claims 4 and 12.

16. Dependent Claim 15

121. See Ground I, Claims 5 and 12.

17. Dependent Claim 16

122. See Ground I, Claims 6 and 12.

18. Dependent Claim 17

123. See Ground I, Elements [1.1], [1.3], and Claim 16.

19. Dependent Claim 18

124. See Ground I, Claims 4 and 17.

20. Dependent Claim 19

125. See Ground I, Claims 5 and 18.

21. Dependent Claim 20

126. See Ground I, Claim 19.

127. In my opinion, *Handley* disclosed this Claim or at least rendered it obvious. As discussed in Ground I, Element [1.1] and [1.3], *Handley* discloses the use of DES, a symmetric key encryption algorithm. A POSITA would have understood that authentication is carried out between the transmitter and the receiver before the data is transmitted from the transmitter to the receiver because exchanging of a secret key needs to occur prior to data transmission in order for a transmitter to encrypt the data with a secret key that the receiver can use to decrypt the data. A POSITA would understand that exchanging the secret key involves authentication carried out between the transmitter and the receiver to ensure that the secret key is exchanged with the proper entities and not with an imposter.

128. To the extent *Handley* does not expressly disclose this element, this element would have been obvious in view of *Handley* in combination with *Basturk*. As I described in Ground I, Element [1.1] and [1.3], *Basturk* discloses

DES, a symmetric key encryption algorithm. A POSITA would have understood that authentication is carried out between the transmitter and the receiver before the data is transmitted from the transmitter to the receiver because any exchanging of a secret key needs to occur prior to data transmission in order for a transmitter to encrypt the data with a secret key that the receiver can use to decrypt the data. A POSITA would understand that exchanging the secret key involves authentication carried out between the transmitter and the receiver, to ensure that the secret key is exchanged with the proper entities and not with an imposter.

22. Dependent Claim 21

129. *See* Ground I, Claim 20.

130. In my opinion, *Handley* disclosed this Claim or at least rendered it obvious. *Handley* discloses that “the real interest in using the internet for audio and video should come from the prospect for a single ubiquitous communications network that not only allows traditional telephony services, but also video, shared collaboration tools, and through IP Multicast, multi-party conferences and multimedia sessions that scale from small group meetings through to television sized audiences.” EX1005, 1. A POSITA would have known that the internet is a packet-switched network and any services provided over the internet, such as “traditional telephony services,” are internet telephony, since packets containing voice data are being used. As such, a POSITA would have known and understood

that the receiver and transmitter are used in Internet telephony because traditional telephony services involve the exchange of packetized voice data.

23. Dependent Claim 23

131. See Ground I, Claims 20 and 21.

24. Independent Claims 25

a) **Preamble [25]**

132. See Ground I, Preamble [1].

b) **Element [25.1]**

133. See Ground I, Elements [1.1], [1.2], [1.3], and Preamble [25].

c) **Element [25.2]**

134. See Ground I, Elements [1.1], [1.3], [1.4], and [25.1].

d) **Element [25.3]**

135. See Ground I, Elements [1.1], [25.2] and Claim 4.

136. A POSITA would have understood and found obvious that the whole RTP packet payload comprises a whole RTP packet payload header made up of at least one sequence number and a time stamp because RFC 1889 discloses that “the RTP header contains timing information and a sequence number that allow the receivers to reconstruct the timing produced by the source” and “RTP Fixed Header Fields” including a “sequence number” and “timestamp.” as these are required header fields as specified by the RTP RFC 1889. EX1008, 5, §2.1, 10, §5.1.

e) **Element [25.4]**

137. See Ground I, Elements [1.2] and [25.3].

f) **[Element [25.5]**

138. See Ground I, Elements [1.1], [1.3], [25.4], and Claim 4.

139. In my opinion, this element would have been obvious in view of *Handley* in combination with *Basturk*. As I described in Ground I, Elements [1.1] and [1.3], *Basturk* discloses a MAC that “includes the transport header, the full data payload, and the trailer of the packet” as part of the MAC computation. EX1006, 12. The common secret between the transmitter and the receiver is a session key α that is used to generate MAC_α and a non-party attacker will be unable to forge an authentic MAC since a non-party does not have access to the session key. EX1006, 12-13.

140. Further, as I described in Ground I, Element [4.1], *Handley* discloses RTP and, pursuant to RFC 1889, “the RTP header contains timing information and a sequence number that allows the receivers to reconstruct the timing produced by the source” and “RTP Fixed Header Fields” including “sequence number” and “timestamp” as these are required header fields as specified by the RTP RFC 1889. EX1008, 5, §2.1, 10, §5.1. A POSITA would have understood that the RTP packet header includes a sequence number and timestamp, as adherence to the RTP RFC 1889 require these fields to be in the header. A POSITA would have appreciated

when combining *Basturk's* secure transport protocol to *Handley's* RTP internet conferencing architecture, that to generate a MAC the transmitter would use for computation, the contents of the RTP which “includes the transport header, the full data payload, and the trailer of the packet,” as well as the sequence number and time stamp in the RTP header, as adherence to the RTP RFC 1889 require these fields to be in the header.

g) **[Element 25.6]**

141. See Ground I, Elements [1.1], [1.2], [1.3], and [25.5].

h) **Element [25.7]**

142. See Ground I, Elements [1.1], [1.3], [1.4], and [25.6].

25. Independent Claim 26

a) **Preamble [26]**

143. See Ground I, Preamble [1].

b) **Element [26.1]**

144. See Ground I, Element [1.2]. A POSITA would have understood and found obvious that a transmitter contains a processor for transmitting data and computing the authentication data because in networking a transmitter is a type of computer that contains a processor for functions like encoding and managing the transmission protocol. A POSITA would understand that these cryptographic algorithms are implemented as software running on a processor as cryptographic

algorithms are generally part of the curriculum studied in software engineering and algorithm courses.

c) **Element [26.2]**

145. See Ground I, Preamble [1], Elements [1.1], [1.2], and [26.1].

d) **Element [26.3]**

146. See Ground I, Element [26.3] and Claim 4.

e) **Element [26.4]**

147. See Ground I, Elements, [25.3], [25.4], [25.5], [25.6], [26.1] and Claim 4.

26. Independent Claim 27

a) **Preamble [27]**

148. See Ground I, Preamble [1].

b) **Element [27.1]**

149. See Ground I, Preamble [1], Preamble [27], and Element [26.1].

150. A POSITA would have understood and found obvious that a receiver, as with a transmitter, would also have a processing unit to decrypt data received from the transmitter and manage the protocol of the packets received from the transmitter. Further, a POSITA would have understood and found obvious that the processing unit of the receiver is communicably connecting to the transmitter as the transmitter is either directing the communications to the receiver through a unicast transmission, or the transmitter is sending multicast communication to a

group to which the receiver belongs. A POSITA would understand that the receiver's network interface and device driver will accept and process these communications, thus communicably connecting the receiver with the transmitter through the use of device driver software that instructs the processor how to utilize the packets received from the network. A POSITA would understand that the device driver passes the received packets through the network stack which processes packets using the header information in the received packet (that was transmitted from a transmitter) utilizing the port number information in the TCP or UDP packet header to determine to which application to deliver the packet. Thus, the appropriate application on the receiver received the data that was transmitted because the destination port numbers of the packets carrying the data identify the respective applications to which the packets are addressed.

c) **Element [27.2]**

151. See Ground I, Elements [1.1], [1.2], [1.3], and [27.1].

d) **Element [27.3]**

152. See Ground I, Elements [1.4] and [27.2].

e) **Element [27.4]**

153. See Ground I, Elements [25.3], [27.3] and Claim 4.

f) **Element [27.5]**

154. See Ground I, Elements [1.1], [1.2], [1.3], [25.5], [27.4], and Claim 4.

155. A POSITA would have understood and found obvious that the receiver, obtains the authentication data after the transmitter transmits data because it was added to the end of the whole RTP packet payload as a MAC which travels with the whole RTP packet payload. A POSITA would understand that these cryptographic algorithms are implemented as software running on a processing unit such as a general-purpose CPU, as cryptographic algorithms are generally part of the curriculum studied in software engineering and algorithm courses.

27. Independent Claim 28

a) Preamble [28]

156. See Ground I, Preamble [1].

b) Element [28.1]

157. See Ground I, Elements [1.1], [1.2] and [28.1].

c) Element [28.2]

158. See Ground I, Elements [1.3] and [28.1].

d) Element [28.3]

159. See Ground I, Elements [1.4] and [28.2].

e) Element [28.4]

160. See Ground I, Element [28.3] and Claim 4.

f) Element [28.5]

161. See Ground I, Elements [1.2] and [28.4].

g) **Element [28.6]**

162. See Ground I, Elements [1.1], [1.2], [1.3], [25.5], [28.5], and Claim 4.

h) **Element [28.7]**

163. See Ground I, Elements [1.1], [1.2], [1.3], [28.6] and Claim 4.

i) **Element [28.8]**

164. See Ground I, Elements [1.1], [1.3], [1.4], and [25.7].

B. Ground II: Claims 1-28 Are Rendered Obvious Over *PacketCable* in Combination With *Handley*

1. Motivation to Combine

165. In my opinion, a POSITA would have been motivated to modify *PacketCable* in view of *Handley* for a number of reasons. Both references are in the field of data transport over the internet and, more specifically, directed to standardizing the transport of multimedia. For example, *PacketCable* discloses that the “PacketCable project is aimed at defining interface specifications that can be used to develop interoperable equipment capable of providing packet-based voice, video and other high-speed multimedia services over hybrid fiber coax (HFC) cable systems utilizing the DOCSIS protocol.” EX1007, 9. Similarly, and as discussed above in Section VII.A.1, *Handley* discloses that “the real interest in using the internet for audio and video should come from the prospect for a single ubiquitous communications network” that “not only allows traditional telephony services, but also video, shared collaboration tools, and through IP multicast,

multi-party conferences and multimedia sessions that scale from small group meetings through to television sized audiences.” EX1005, 1. A POSITA would have looked to *Handley* that discusses specific types of packet-based voice, video, and other high-speed multimedia services such as the shared collaboration tools discussed in *Handley* that can be used in the system of *PacketCable*. *PacketCable* also discloses using “[t]he IETF standard RTP (RFC 1899 - Real-Time Transport Protocol) [] to transport all media streams in the PacketCable network[32]” (EX1007, 31). This mirrors the disclosure of *Handley*: “The transport protocol for real-time flows is RTP[22]” referring to H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson ‘RTP: A Transport Protocol for Real-Time Applications’ RFC 1889. EX1005, 8.

166. Although *PacketCable* does not expressly disclose acceptance/rejection of the whole RTP packet payload, the use of *PacketCable*’s disclosure of its PacketCable project which provides high-speed multimedia data transport using RTP in combination with *Handley*’s RTP internet conferencing architecture and its teaching of discarding of packets would have been obvious to a POSITA in view of *PacketCable*.

167. In my opinion, a POSITA would have also been motivated to modify *PacketCable* in view of *Handley* because both references understood and describe the importance of authentication in networks. For example, *PacketCable* discloses

that “[d]esign considerations include confidentiality, authentication, integrity, non-repudiation and access control.” EX1007, 17. Similarly, *Handley* discloses that “*security* measures may be required to actually enforce the conference policy, *e.g.* to control who is listening and to authenticate contributions as actually originating from a specific person.” EX1005, 2. *Handley* was published in February 1999, is prior art, relates to the field of computer networking, and therefore, a POSITA would have known of and looked to its teachings and motivations. *PacketCable* provides overviews of various protocol interfaces, for example, Call Signaling and Media Streams. *Handley* discusses providing not only traditional telephony services that would be initiated using call signaling protocols and delivered using media streams, but also video, shared collaboration tools, multi-party conferences, and multimedia sessions. EX1005, 1.

168. The similarities between the references are substantial, and a POSITA would have had a reasonable expectation of success in combining *PacketCable* with *Handley*. For example, both references contemplate the use of RTP, a relatively new protocol at the time.⁷ EX1007, 31; EX1005, 8, 15. *PacketCable* discloses that “*PacketCable* core security services include a mechanism for providing end-to-end encryption of RTP media streams, thus substantially reducing

⁷ The original RTP specification, RFC 1889, was published in 1996. *PacketCable* was published in 1999.

the threat to privacy.” EX1007, 42-43. Similarly, *Handley* discloses that “[i]f the application requirement (conference policy) is to communicate between some defined set of users, then strict privacy can only be enforced in any case through adequate end-to-end encryption. RTP specifies a standard way to encrypt RTP and RTCP packets using private key encryption schemes such as DES.” EX1005, 12.

169. A POSITA would have understood that the hardware and systems to implement authentication that accepts and rejects packets would be necessary and included in view of *PacketCable*’s multimedia services architecture and *Handley*’s internet conferencing architecture, as both *PacketCable* and *Handley* disclose ample concerns about security and devote large portions of their disclosures to security issues. EX1007, 17, 42-48; EX1005, 11-14. Furthermore, *Handley* discloses that “many broadcasters [] require the ability to ensure that only they can send, perhaps in addition to ensuring that only their paid customers can receive” and thus provides motivation for broadcasters to ensure proper authentication. EX1005, 13.

170. Thus, use of *PacketCable*’s multimedia services architecture with *Handley*’s teaching of discarding unwanted packets would have made use of a known data transmission architecture and known hardware in a known application to improve a similar method in the same way. The combination also applies a known technique to a method ready for improvement to yield predictable results.

Using a data authentication technique in combination with data transmission involved the use of routine computers, equipment, protocols, and programming techniques, and a POSITA would have had a reasonable expectation of success in using the disclosed internet conferencing architecture with *Handley*. Both *PacketCable* and *Handley* utilize RTP as the transport protocol for real-time data flows. EX1007, 31; EX1005, 8, 15. Both *PacketCable* and *Handley* are directed to providing audio, video, and other high-speed applications such as shared collaboration tools for users. EX1007, 9; EX1005, 1.

2. Independent Claim 1

a) Preamble [1]

171. To the extent the preamble is limiting, it is my opinion that *PacketCable* disclosed, or at least rendered obvious, the preamble. As described in Ground I, Preamble[1] and Element [1.1], data transmission over the internet and the OSI Model were known. *PacketCable* discloses “interoperable equipment capable of providing packet-based voice, video and other high-speed multimedia services.” EX1007, 9. Further, *PacketCable* discloses the implementation of multimedia terminal adapters (“MTA”), which are “a PacketCable client device that contains a subscriber-side interface to the subscriber’s CPE (e.g., telephone) and a network-side signaling interface to call control elements in the network.” *Id.*, 18. MTAs are responsible for providing “codecs and all signaling and

encapsulations functions required for media transport and call signaling.” *Id.* Further, *PacketCable* discloses that the “pkt-s6” interface is “end-to-end media packets between two MTAs.” *Id.*, 45. A POSITA would have understood that the MTAs in *PacketCable* are transmitting data between one another and are the equivalent to a transmitter and receiver because the “pkt-6” interface connects two MTAs to send end-to-end media packets.

172. To the extent *PacketCable* does not expressly disclose this element, it would have been obvious in view of *PacketCable* in combination with *Handley*. See Ground I, Preamble [1].

b) **Element [1.1]**

173. See Ground II, Preamble [1].

174. In my opinion, *PacketCable* disclosed this element or at least rendered it obvious. *PacketCable* explicitly discloses RTP, an application protocol whose packets are created at the application layer. EX1007, 45. For example, *PacketCable* discloses that “RTP packets are encrypted directly with RC4, without any additional security layers. An MMH-based MAC (Message Authentication Code) optionally provides message integrity. Keys are distributed by the CMS to the two endpoints.” *Id.* As described in Ground I, Element [1.1], a POSITA would have understood (1) that manipulating RTP packets, *e.g.*, implementing RC4 or adding a MAC, results in authentication at the packet level and (2) RTP packets are

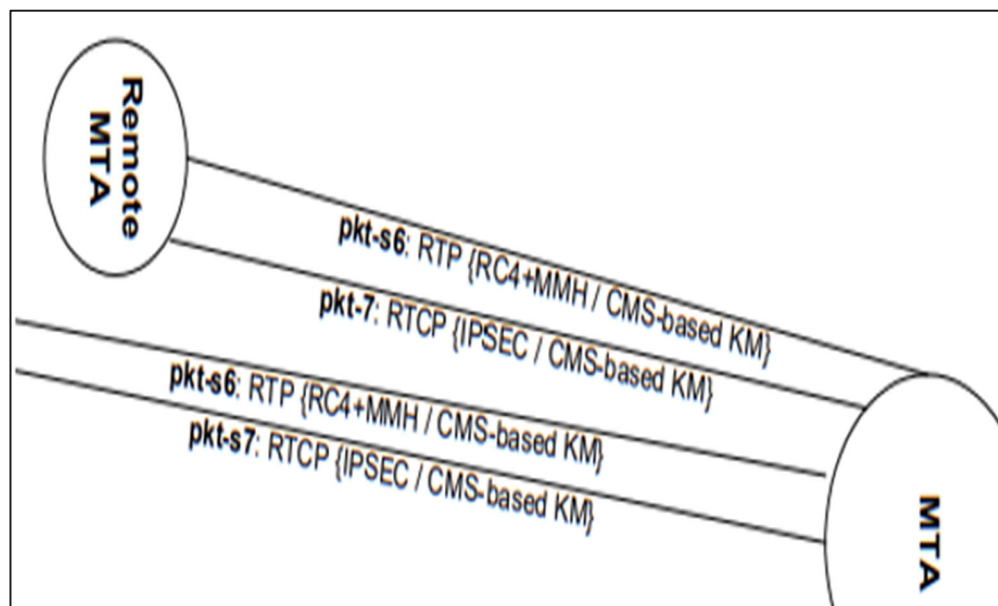
created at the application layer because RTP is an application protocol. As such, *PacketCable* discloses this element.

175. To the extent *PacketCable* does not expressly disclose this element, it would have been obvious in view of *PacketCable* in combination with *Handley*. See Ground I, Element [1.1].

c) **Element [1.2]**

176. See Ground II, Preamble [1] and Element [1.1].

177. In my opinion, *PacketCable* disclosed this element or at least rendered it obvious. *PacketCable* discloses MAC utilization for message integrity and illustrates in Figure 13 (reproduced below) that the “RC4 + MMH / CMS-based KM” encrypts and adds authentication to the whole RTP packet in a “pkt-s6” interface. EX1007, 44.



EX1007, Fig. 13

178. Further, *PacketCable* discloses that the “MAC computation spans the packet’s unencrypted header and encrypted payload.” EX1007, 47. A POSITA would have understood and found obvious that placement of authentication data in the middle of the packet payload would negatively impact performance, cause packet fragmentation issues, and complicate the implementation of networking software, since the data in the packet would be interrupted by the authentication data, and extra data fields would be needed to identify the exact location of the authentication data within the packet data. As such, a POSITA would have avoided such placement. With this knowledge, a POSITA would have understood available authentication data placement options to include either at the beginning or the end of the payload, either in a packet header (at the beginning as ubiquitous in packets) or in packet trailer (at the end which are already present in certain types of packets such as in Ethernet frames). Further, a POSITA would have understood that the authentication data would go at the end of the whole RTP packet payload because it would reduce the latency of delivering the packet since placing the authentication data at the beginning of the payload requires storing the packet, computing the authentication data, transmitting the packet to the network, and placing the authentication data at the end of the payload allows the packet to be transmitted to the network while calculating the authentication data and then adding the authentication data to the end of the payload while the packet

transmission is being completed. A POSITA would have understood and found obvious that placing authentication at the end of the payload would allow the transmitter to transmit data while simultaneously computing the checksum and preventing any unnecessary transmission delay by having to first compute the checksum and then transmit the payload. Thus, a POSITA would have understood and found obvious that placing authentication at the end of the whole RTP packet payload would be the best solution to increase transmission speeds and decrease memory usage needed to encrypt and decrypt the data, since computations can be performed as the packet is being received. A POSITA would have appreciated that the transmitter is the device to insert authentication data for the receiver to use and verify that it knows the transmitter, and the data was not manipulated while in transit as is already commonly done with checksums and other data integrity checks. Other protocols similarly put check sequences at the end of a packet. For example, the Ethernet standard, first published in 1985, employs the same design choice reasoning to include a “Frame Check Sequence” (“FCS”) at the end of the Ethernet frame so that it can be computed during transmission and appended to the end of the frame, and it can be computed upon reception to check against the FCS received and discarded if there is not a match. *See* EX1014, 43, 46, 41, 23-24, 26, 27, Fig 3-1.

179. To the extent *PacketCable* does not expressly disclose this element, it would have been obvious in view of *PacketCable* in combination with *Handley*. See Ground I, Element [1.2].

d) **Element [1.3]**

180. See Ground II, Preamble [1], Elements [1.1] and [1.2].

181. In my opinion, *PacketCable* disclosed this element or at least rendered it obvious. First, a POSITA would have understood that *PacketCable* discloses the use of RTP packet level authentication data and that this data is inserted by the transmitter, since the transmitter has access to the secret key or private key necessary to add authentication data. For example, *PacketCable* discloses that in a “pkt-s6” interface there is “end-to-end media packets between two MTAs, or between MTA and MG. RTP packets are encrypted *directly* with RC4, without any additional security layers. An MMH-based MAC (Message Authentication Code) optionally provides message integrity. Keys are distributed by the CMS to the two endpoints.” EX1007, 45. A POSITA would have understood this disclosure to mean RTP packet level authentication because “RTP packets are encrypted directly with RC4” at the RTP packet level. *Id.* Further, *PacketCable* discloses that “[e]ach media RTP packet is encrypted for privacy” where “[e]ach RTP packet may include an optional message authentication code (MAC)” and that the “MAC algorithm can also be negotiated.” *Id.*, 47. A POSITA would have understood that

this use of MAC is RTP packet level authentication because each RTP packet includes a MAC at the RTP packet level, and thus, the MAC is decrypted and verified on a packet-by-packet basis.

182. Further, a POSITA would have understood and found obvious that the receiver is ascertaining whether it knows the transmitter on the RTP packet level authentication described above because the authentication data is generated using symmetric or asymmetric encryption methods. Since the receiver needs to know which secret key to use for encrypted data received from a different transmitter, each party stores secret keys in association with information identifying the transmitter with which the secret key is associated, and thus, the receiver already knows of the transmitter when it receives the first encrypted data from the transmitter. Private keys are a closely held secret and thus, are only known by the entity using its private key. The corresponding public key can be used to decrypt data using the private key. The public key can be distributed to anyone, and to send data securely to a recipient, the public key is used to encrypt the data. Because the public key and corresponding private key are cryptographically related, the private key can be used to decrypt data encrypted with the corresponding public key. It follows that only the party with the corresponding private key can decrypt data using the corresponding private key, and since private keys are a closely held secret, only the intended recipient can decrypt a message using a public key.

Further, as discussed in Ground I, Elements [1.1] and [1.3], a POSITA would have understood that a receiver knows whether the transmitter is using symmetric or asymmetric encryption. If using symmetric key encryption, the transmitter and receiver will exchange secret keys prior to data transmission so the transmitter can use the secret key to encrypt the data, and the receiver can use the secret key to decrypt the data. *PacketCable* states that “[k]eys for the encryption and MAC calculation are derived from the End-End secret, which is exchanged between sending and receiving MTA as part of the call signaling.” EX1007, 47. *PacketCable* discloses that a secret key is “the cryptographic key used in a symmetric key algorithm, which results in the secrecy of the encrypted data depending solely upon keeping the key a secret, also known as a symmetric key.” *Id.*, 66. A POSITA would have understood and found obvious that a receiver MTA knows the shared secret key, which it then uses with the received message to see if the computed MAC matches the MAC received from the transmitter MTA.

183. To the extent *PacketCable* does not expressly disclose this element, it would have been obvious in view of *PacketCable* in combination with *Handley*. See Ground I, Element [1.1] and [1.3].

e) **Element [1.4]**

184. See Ground II, Preamble [1], Elements [1.1], [1,2], and [1.3].

185. In my opinion, *PacketCable* disclosed this element or at least rendered it obvious. A POSITA would have understood that if the authentication data, *i.e.*, the MAC, is not verified and accepted by the receiver, then the receiver would reject the data, meaning it would not process or decrypt the data.

186. Further, this element would have been obvious in view of *PacketCable* in combination with *Handley*. See Ground I, Element [1.4].

3. Dependent Claim 2

187. See Ground II, Claim 1.

188. In my opinion, *PacketCable* disclosed this Claim or at least rendered it obvious. A POSITA would have understood and found obvious that a PDU is a single unit of information at a particular OSI layer (*e.g.*, an RTP packet at the application layer). A POSITA would have understood that a media application, such as in a conference application, generates RTP packets at a steady rate, and the protocol data units are the RTP data in each packet. *PacketCable* discloses that “[k]eys for the encryption and MAC calculation are derived from the End-End secret, which is exchanged between sending and receiving MTA as part of the call signaling. Thus, the key exchanges for media stream security are secured themselves by the call signaling security.” EX1007, 47. A POSITA would appreciate that the PDU in *PacketCable*, *e.g.*, an RTP packet, is linked to a secret in two possible ways. First, the RTP packet may be encrypted and is linked to the

secret via the keys used to encrypt and decrypt the RTP packet since the correct key must be used to properly decrypt the packet. Second, the RTP packet may be used as the basis of an authentication code, e.g., the MAC, and is linked to the secret when the hash is encrypted since the correct key must be used to decrypt the signature and retrieve the correct hash.

189. To the extent *PacketCable* does not expressly disclose this element, it would have been obvious in view of *PacketCable* in combination with *Handley*. See Ground I, Element [2.1].

4. Dependent Claim 3

190. See Grounds I-II, Elements [1.1], [1.3] and Claim 2.

5. Dependent Claim 4

191. See Grounds I-II, Claims 2 and 4.

192. In my opinion, *PacketCable* disclosed this Claim or at least rendered it obvious. *PacketCable* discloses that “[t]he IETF standard RTP (RFC 1899 – Real-Time Transport Protocol) is used to transport all media streams in the PacketCable network.” EX1007, 31. A POSITA would have understood RTP inherently includes sequence numbers and timestamps. For example, RFC 1889 teaches that “[t]his 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.” EX1008, 3, §1. RFC 1889 further discloses “the RTP header contains timing information and a sequence number that allow the receivers to reconstruct the timing produced by the source” and “RTP Fixed Header Fields” including “sequence number” and “timestamp” as these are required header fields as specified by the RTP RFC 1889. *Id.*, 5, §2.1, 10, §5.1. As discussed in Grounds I-II, Claim 2, a POSITA would have understood that a protocol data unit would be anything part of a protocol exchange and, therefore, *PacketCable* discloses this element because RTP (including the sequence number and time stamp) is available at the application layer in order for the application to decode and reassemble any real time data.

6. Dependent Claim 5

193. See Grounds I-II, Claim 2.

194. In my opinion, *PacketCable* disclosed this Claim or at least rendered it obvious. *PacketCable* discloses the use of a Hashed Message Authentication Code (“HMAC”), which is an encryption operation with a secret key that uses a one-way hash function like MD5, or SHA-1. For example, *PacketCable* discloses “HMAC – Hashed Message Authentication code – a message authentication algorithm, based on either SHA-1 or MD5 hash.” EX1007, 62. As described in Ground II, Element [1.3], *PacketCable* discloses that “keys for the encryption and

MAC calculation are derived from the End-End secret, which is exchanged between sending and receiving MTA as part of the call signal.” *Id.*, 47. A POSITA would have understood that this end-end secret may be a secret key in a symmetric encryption scenario and known only to the transmitter and receiver since the proper secret key must be used to decrypt any data or signature.

195. *PacketCable* describes one-way hash functions as “a hash function that has an insignificant number of collisions upon output” and notes that “Message Digest 5 [is] a one-way hash algorithm which maps variable length plaintext into fixed length (16 byte) ciphertext).” EX1007, 63, 64. A POSITA would have understood and found obvious that MD5 is a one-way hash algorithm that may be implemented in a symmetric key scenario such as HMAC, where keys are known only to the parties involved, i.e., the transmitter and receiver.

196. To the extent *PacketCable* does not expressly disclose this element, it would have been obvious in view of *PacketCable* in combination with *Handley*. See Ground I, Claim 5.

7. Dependent Claim 6

197. See Grounds I-II, Claim 2.

198. In my opinion, *PacketCable* disclosed this Claim or at least rendered it obvious. As described in Ground II, Elements [1.1], [1.3] and [5.1], *PacketCable* discloses the use of symmetric and asymmetric encryption/decryption as well as

the use of MD5. A POSITA would have understood that the ascertaining would stem from whether the receiver knows the transmitter's secret key or has cached the receiver's public key in an encryption/decryption scenario that uses only keys and no cryptographic checksum. In a scenario using both an encryption/decryption operation and a cryptographic checksum, e.g., MD5, a POSITA would have understood that (1) the data is ran through a hash algorithm to create a message digest; (2) the digest is encrypted to create a digital signature using a either a secret key or private key; (3) the data and digital signature are sent to the receiver; (4) the receiver decrypts the digital signature using a secret key or the transmitter's public key to obtain the message digest; (5) running the data through the same hash algorithm as the transmitter did; and (6) comparing the output of the hash algorithm to the decrypted message digest.

199. As discussed in Ground II, Elements [1.2] and [5.1], *PacketCable* discloses the implementation of MAC signatures, as well as asymmetric or symmetric encryption and decryption. A POSITA would have understood that because the signature is encrypted it must also be decrypted and, in the symmetric scenario, uses the same key it was encrypted with.

200. To the extent *PacketCable* does not expressly disclose this element, it would have been obvious in view of *PacketCable* in combination with *Handley*. See Ground I, Elements [1.1], [1.2], [1.3], and Claim 6.

8. Dependent Claim 7

201. See Grounds I-II, Claims 2 and 6.

9. Dependent Claim 8

202. See Grounds I-II, Claim 2.

203. In my opinion, *PacketCable* disclosed this Claim or at least rendered it obvious. As I discussed in Ground II, Elements [1.1], [1.3], and [5.1], *PacketCable* discloses the use of MD5 and asymmetric and symmetric encryption. A POSITA would have understood that linking to the secret is carried out by either asymmetric or symmetric encryption as well as a cryptographic checksum and key information, since only the parties that have the secret key can properly decrypt the data. Thus, it keeps the data secret from other non-parties.

204. To the extent *PacketCable* does not expressly disclose this element, this element would have been obvious in view of *PacketCable* in combination with *Handley*. As I discussed in Ground I, Elements [1.1], [1.3], and [5.1], *Handley* discloses the use of MAC signatures, MD5, SHA, and the use of asymmetric and symmetric encryption.

10. Dependent Claim 9

205. See Grounds I-II, Elements [1.1], [1.3] and Claim 8.

11. Dependent Claim 10

206. See Grounds I-II, Claims 4 and 8.

12. Dependent Claim 11

207. *See* Grounds I-II, Claims 5 and 8.

13. Dependent Claim 12

208. *See* Grounds I-II, Elements [1.1], [1.3] and Claim 8.

14. Dependent Claim 13

209. *See* Grounds I-II, Elements [1.1], [1.3], and Claim 12.

15. Dependent Claim 14

210. *See* Grounds I-II, Claims 4 and 12.

16. Dependent Claim 15

211. *See* Grounds I-II, Claims 5 and 12.

17. Dependent Claim 16

212. *See* Grounds I-II, Claims 6 and 1.

18. Dependent Claim 17

213. *See* Grounds I-II, Elements [1.1], [1.3], and Claim 16.

19. Dependent Claim 18

214. *See* Grounds I-II, Claims 4 and 17.

20. Dependent Claim 19

215. *See* Grounds I-II, Claims 5 and 19.

21. Dependent Claim 20

216. *See* Grounds I-II, Claim 19.

217. In my opinion, *PacketCable* disclosed this Claim or at least rendered it obvious. As I discussed in Ground II, Element [1.3], *PacketCable* discloses the use of symmetric keys. A POSITA would have understood that authentication is carried out between the transmitter and the receiver before the data is transmitted from the transmitter to the receiver because exchanging of symmetric keys occurs prior to data transmission.

218. To the extent *PacketCable* does not expressly disclose this element, this element would have been obvious in view of *PacketCable* in combination with *Handley*. As I discussed in Ground I, Elements [1.1] and [1.3], *Handley* discloses DES, a symmetric key encryption algorithm and a POSITA would have understood and found obvious that DES keys need to be exchanged prior to encryption such that the transmitter can use the secret key to encrypt the data, and the receiver can use the secret key to decrypt the data.

22. Dependent Claim 21

219. See Grounds I-II, Claim 20.

220. In my opinion, *PacketCable* disclosed this Claim or at least rendered it obvious. *PacketCable* discloses that “MTA – Media Terminal Adapter – contains the interface to subscribers’ CPE, a network interface, CODECs, and all signaling and encapsulation functions required for VoIP transport, class features signaling, and QoS signaling.” EX1007, 64. A POSITA would have understood that “VoIP”

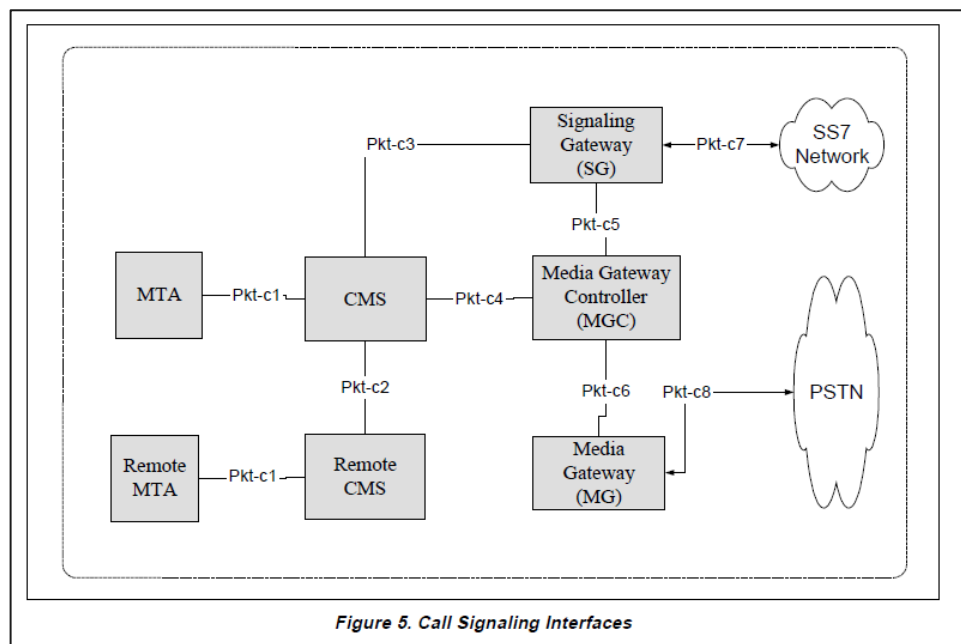
stands for “Voice over Internet Protocol,” a technology that uses packets to carry telephone traffic for Internet telephony, as the Internet Protocol is packet based. See EX1013, 7.

221. To the extent *PacketCable* does not expressly disclose this element, this element would have been obvious in view of *PacketCable* in view of *Handley*. See Ground I, Claim 21.

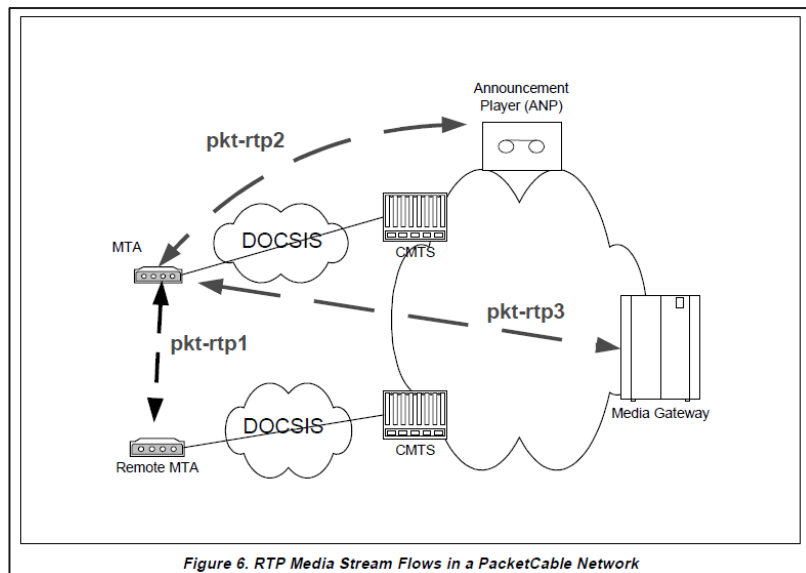
23. Dependent Claim 22

222. See Ground II, Claim 21.

223. In my opinion, *PacketCable* disclosed this Claim or at least rendered it obvious. *PacketCable* discloses the use of media gateways in Figures 5 and 6.



EX1007, Figure 5.



EX1007, Figure 6.

224. A POSITA would have understood the media gateways act to switch data between a particular IP flow and a particular PSTN port on the gateway and the receiver and transmitter would be a switch, as the gateway must perform a switching function in order to map specific IP flows to particular PSTN ports. Since flows and ports change as calls are initiated and terminated, the gateway must include a switching function in order to make the appropriate connections between IP flows and PSTN ports on an ongoing basis. Further, a POSITA would have understood and found obvious to include this functionality in a media gateway, which terminates the IP data side into the PSTN and ultimately acts as a switch between a particular IP flow and a particular PSTN port. Inclusion of a gateway would have been obvious to a POSITA due to the benefits of having data

network devices communicate with all pre-existing telephones on the PSTN such that the data network devices can conduct a telephone call with users beyond just those on the data network. Adding a media gateway would thus allow users of packet-based telephones to reach all the traditional landlines in the Public Switched Telephone Network, rather than being confined to calling only packet-based telephones.

24. Dependent Claim 23

225. *See* Grounds I-II, Claims 21 and 23.

25. Dependent Claim 24

226. *See* Ground II, Claim 22 and Grounds I-II, Claim 23.

26. Independent Claims 25

a) **Preamble [25]**

227. *See* Grounds I-II, Preamble [1].

b) **Element [25.1]**

228. *See* Grounds I-II, Elements [1.1], [1.2], [1.3], and Preamble [25].

c) **Element [25.2]**

229. *See* Grounds I-II, Elements [1.1], [1.3], [1.4], [25.1], and [25.2].

d) **Element [25.3]**

230. *See* Grounds I-II, Elements [1.1], [25.2], [25.3], and Claim 4.

231. A POSITA would have understood and found obvious that the whole RTP packet payload comprises a whole RTP packet payload header comprising at

least one of a sequence number and a time stamp because RFC 1889 discloses that “the RTP header contains timing information and a sequence number that allow the receivers to reconstruct the timing produced by the source” and “RTP Fixed Header Fields” including “sequence number” and “timestamp” as these are required header fields as specified by the RTP RFC 1889. EX1008, 5, §2.1, 10, §5.1.

e) **Element [25.4]**

232. See Grounds I-II, Elements [1.2], [25.3], and [25.4].

f) **Element [25.5]**

233. See Grounds I-II, Elements [1.1], [1.3], [25.4], [25.5], and Claim 4.

234. In my opinion, *PacketCable* disclosed this element or at least rendered it obvious. As I described in Ground II, Elements [1.1], [1.2], [1.3], and [1.4], *PacketCable* discloses that “[t]he MAC computation spans the packet’s encrypted header and encrypted payload.” EX1007, 47. A POSITA would have understood and found obvious that RTP inherently includes a sequence number and timestamp as described in Grounds I-II, Claim 4. Further, because “RTP packets are encrypted directly with RC4,” a symmetric encryption scheme with secret keys, that there is a common secret between the transmitter and receiver. *Id.*, 45.

g) **Element [25.6]**

235. See Grounds I-II, Elements [1.1], [1.2], [1.3], [25.5], and [25.6].

h) **Element [25.7]**

236. See Ground I, Elements [1.1], [1.3], [1.4], [25.6], and [25.7].

27. Independent Claim 26

a) **Preamble [26]**

237. See Ground I, Preamble [1].

b) **Element [26.1]**

238. See Ground I-II, Elements [1.2], [26.1].

c) **Element [26.2]**

239. See Ground I-II, Preamble [1], Elements [1.1], [1.2], [26.1], and [26.2]

d) **Element [26.3]**

240. See Grounds I-II, Elements [26.2], [26.3] and Claim 4.

e) **Element [26.4]**

241. See Grounds I-II, Elements [1.1], [1.2], [1.3], [25.3], [25.4], [25.5], [25.6], [26.1] and Claim 4.

28. Independent Claim 27

a) **Preamble [27]**

242. See Grounds I-II, Preamble [1], Preamble [27], and Element [26.1].

b) **Element [27.1]**

243. A POSITA would have understood and found obvious that, similar to a transmitter, a receiver would also have a processing unit to decrypt data received

from the transmitter and manage the protocol of the packets received from the transmitter. Further, a POSITA would have understood and found obvious that the processing unit of the receiver is communicably connecting to the transmitter as the transmitter is either directing the communications to the receiver through a unicast transmission, or the transmitter is sending multicast communication to a group to which the receiver belongs. A POSITA would understand that the receiver's network interface and device driver will accept and process these communications and thus, communicably connecting the receiver with the transmitter through the use of device driver software that instructs the processor how to utilize the packets received from the network. A POSITA would understand that the device driver passes the received packets through the network stack which processes packets using the header information in the received packet (that was transmitted from a transmitter) utilizing the port number information in the TCP or UDP packet header to determine to which application to deliver the packet. Thus, the appropriate application on the receiver received the data that was transmitted.

c) **Element [27.2]**

244. See Grounds I-II, Elements [1.1], [1.2], [1.3], [27.1], [27.2], and [27.3].

d) **Element [27.3]**

245. See Grounds I-II, Elements [1.4], [27.2], and [27.3].

e) **Element [27.4]**

246. See Ground I, Elements [25.3], [27.3] and Claim 4.

f) **Element [27.5]**

247. See Grounds I-II, Elements [1.1], [1.2], [1.3], [25.5], [27.4], [27.5] and Claim 4.

248. A POSITA would have understood and found obvious that the receiver and its processing unit would obtain the authentication data after the transmitter transmits data because it was added to the end of the whole RTP packet payload as a MAC, that the MAC travels with the whole RTP packet payload as any trailer continues as part of a packet, and that any packet fragmentation that occurs in the network would have reassembly performed at the endpoint to reconstitute the whole packet.

29. Independent Claim 28

a) **Preamble [28]**

249. See Grounds I-II, Preamble [1], Preamble [28].

b) **Element [28.1]**

250. See Grounds I-II, Elements [1.1], [1.2] and [28.1]

c) **Element [28.2]**

251. See Grounds I-II, Elements [1.3], [28.1], and [28.2].

d) **Element [28.3]**

252. See Grounds I-II, Elements [1.4], [28.2], and [28.3].

e) **Element [28.4]**

253. See Grounds I-II, Elements [28.3], [28.4] and Claim 4.

f) **Element [28.5]**

254. See Grounds I-II, Elements [1.2] and [28.4].

g) **Element [28.6]**

255. See Grounds I-II, Elements [1.1], [1.2], [1.3], [25.5], [28.5], and Claim 4.

h) **Element [28.7]**

256. See Grounds I-II, Elements [1.1], [1.2], [1.3], [28.6] and Claim 4.

i) **Element [28.8]**

257. See Grounds I-II, Elements [1.1], [1.3], [1.4], [25.7], [28.7], and [28.8].

**C. Ground III: Claims 1-28 Are Rendered Obvious Over
PacketCable in Combination with *Basturk***

1. Motivation to Combine

258. Although *PacketCable* does not expressly disclose accepting/rejecting the whole RTP packet payload, using *PacketCable*'s architecture framework which provides high-speed multimedia data transport using RTP in combination with *Basturk*'s secure transport protocols would have been obvious to a POSITA in view of *PacketCable*. A POSITA would have been motivated to use *Basturk*'s "on-the-fly" signature computation with *PacketCable*'s architecture framework due to the known communication benefits, which include faster data transmission,

reduced jitter, and reduced computing power. Furthermore, Basturk teaches that its “on-the-fly” computation “minimizes latency” and “facilitates the eventual off-loading to the hardware.” EX1006, 7. A POSITA would recognize that reducing the latency in a live transmission provides a more natural conversational flow and reduces interruptions. For example, conversations that happen over high latency connections, such as television interviews over satellite links, often seem awkward, with long pauses. People are trained to wait for a reply, but those who try to have a normal conversation over such a link often speak over each other, with people wanting to talk in order to fill silent periods, then realizing the other person had started to speak. Furthermore, a POSITA would recognize that offloading certain computations to hardware allow for faster, uninterrupted process, which reduces jitter by have fewer interruptible periods where a process may be pre-empted, and that offloading computation to hardware would free up compute cycles.

259. For example, *PacketCable* aimed to “enable real-time multimedia communications” in a high-speed environment. EX1007, 14. Similarly, *Basturk* discloses that “[i]deally, we want signatures to be computed and verified on-the-fly; i.e., MAC generation should be carried out as data transmissions proceeds rather than before these operations have completed” because “[t]he goal is to avoid slowing down the regular flow of data.” EX1006, 12 (cleaned up).

260. A POSITA would have also appreciated that, although *Basturk* does not explicitly disclose RTP, its secure transport protocol is compatible with *PacketCable*'s architectural framework. For example, *Basturk* discloses that its proposed protocol "should be able to operate in both environments," including "software encryption and decryption for secure video conferences" at both low and high resolution. EX1006, 5. A POSITA would have understood that RTP is commonly used for video conferences, as "RTP is primarily designed to satisfy the needs of multi-participant multimedia conferences" and that the original RTP Request for Comments, RFC 1889, "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." EX1008, 3. Further, *Basturk* teaches its discussion of security features "reflects the need to add security functionality in a way which is consistent with, and best matches, the existing communication structure" and so that "[a]pplications using the regular transport protocol should not have to be modified at all." EX1006, 3, 5.

261. A POSITA would have understood that the systems of *PacketCable* and *Basturk* are compatible or related because both implement MACs for authentication. EX1007, 45, 47; EX1006, 9, 11-13. In addition to the reasons described above in Sections VII.A.1 and VII.A.2, a POSITA would have understood that *PacketCable* recognized the need for authentication and discloses

that the “media stream may be vulnerable to malicious eavesdropping, resulting in a loss of communications privacy.” EX1007, 42. PacketCable core security services “include a mechanism for providing end-to-end encryption of RTP media streams, thus substantially reducing the threat to privacy” and the core security services are “authentication, access control, integrity, confidentiality and non-repudiation.” *Id.*, 42-43. Further, *PacketCable* aimed to “defin[e] interface specifications that can be used to develop interoperable equipment capable of providing packet-based voice, video and other high-speed multimedia services over hybrid fiber coax (HFC) cable systems utilizing the DOCSIS protocol.” *Id.*, 9.

262. In using *PacketCable*’s architectural framework and *Basturk*’s secure transport protocol, a POSITA would have understood that hardware to implement multi-media transport may be necessary to keep up with the high volume of packets generated in video applications and could include dedicated cryptographic hardware as taught by *Basturk*. EX1006, 7.

263. Thus, use of *PacketCable*’s architectural framework in conjunction with *Basturk*’s secure transport protocol, would have made use of known multi-media data transport architecture and a known authentication method to improve data transmission security. Said combination also applies a known technique to a known device or method ready for improvement to yield predictable results. Using an internet conferencing architecture in combination with a secure transport

protocol involves the use of routine computers, equipment, and techniques, and a POSITA would have had a reasonable expectation of success in using “on-the-fly” data authentication with *PacketCable*. A POSITA would have found the use and implementation of such hardware to implement *Basturk’s* authentication methods with the internet conferencing architecture obvious, as video conference applications in particular may generate large numbers of packets, and signing packets at a high rate may require the use of hardware acceleration.

2. Independent Claim 1

a) Preamble [1]

264. See Grounds I-III, Preamble [1].

b) Element [1.1]

265. See Grounds I-III, Preamble [1], Element [1.1].

c) Element [1.2]

266. See Grounds I-III, Preamble [1], Elements [1.1] and [1.2].

d) Element [1.3]

267. See Grounds I-III, Preamble [1], Elements [1.2] and [1.3].

e) Element [1.4]

268. See Grounds I-III, Preamble [1], [1,3], and [1.4].

3. Dependent Claim 2

269. See Grounds I-III, Claim 2.

4. Dependent Claim 3

270. *See* Grounds I-III, Elements [1.1], [1.3] and Claim 2, 3.

5. Dependent Claim 4

271. *See* Grounds I-III, Claims 2 and 4.

6. Dependent Claim 5

272. *See* Grounds I-III, Claims 2 and 5.

7. Dependent Claim 6

273. *See* Grounds I-III, Claim 2 and 6.

8. Dependent Claim 7

274. *See* Grounds I-III, Claims 2, 6, and 7.

9. Dependent Claim 8

275. *See* Grounds I-III, Claims 2, and 8.

10. Dependent Claim 9

276. *See* Grounds I-III, Elements [1.1], [1.3] and Claims 8, 9.

11. Dependent Claim 10

277. *See* Grounds I-III, Claims 4, 8 and 10.

12. Dependent Claim 11

278. *See* Grounds I-III, Claims 5, 8 and 11.

13. Dependent Claim 12

279. *See* Grounds I-III, Elements [1.1], [1.3] and Claims 8, 12.

14. Dependent Claim 13

280. *See* Grounds I-III, Elements [1.1], [1.3], and Claims 12, 13.

15. Dependent Claim 14

281. *See* Grounds I-III, Claims 4, 12 and 14.

16. Dependent Claim 15

282. *See* Grounds I-III, Claims 5, 12 and 15.

17. Dependent Claim 16

283. *See* Grounds I-III, Claims 6, 1 and 16.

18. Dependent Claim 17

284. *See* Grounds I-III, Elements [1.1], [1.3], and Claims 16, 17.

19. Dependent Claim 18

285. *See* Grounds I-III, Claims 4, 17, 18.

20. Dependent Claim 19

286. *See* Grounds I-III, Claims 5 and 19.

21. Dependent Claim 20

287. *See* Grounds I-III, Claims 19 and 20.

22. Dependent Claim 21

288. *See* Grounds I-III, Claims 20 and 21.

23. Dependent Claim 22

289. *See* Ground II-III, Claims 21 and 22.

24. Dependent Claim 23

290. *See* Grounds I-III, Claims 21 and 23.

25. Dependent Claim 24

291. *See* Ground II-III, Claims 22, 24 and Grounds I-II, Claims 22 and 23.

26. Independent Claims 25

a) Preamble [25]

292. See Grounds I-III, Preamble [1].

b) Element [25.1]

293. See Grounds I-III, Elements [1.1], [1.2], [1.3], and Preamble [25].

c) Element [25.2]

294. See Grounds I-III, Elements [1.1], [1.3], [1.4], [25.1], and [25.2].

d) Element [25.3]

295. See Grounds I-III, Elements [1.1], [25.2], [25.3], and Claim 4.

e) Element [25.4]

296. See Grounds I-III, Elements [1.2], [25.3], and [25.4].

f) Element [25.5]

297. See Grounds I-III, Elements [1.1], [1.3], [25.4], [25.5], and Claim 4.

g) Element [25.6]

298. See Grounds I-III, Elements [1.1], [1.2], [1.3], [25.5], and [25.6].

h) Element [25.7]

299. See Grounds I-III, Elements [1.1], [1.3], [1.4], [25.6], and [25.7].

27. Independent Claim 26

a) Preamble [26]

300. See Grounds I-III, Preamble [1].

b) Element [26.1]

301. See Ground I-III, Elements [1.2], [26.1].

c) **Element [26.2]**

302. *See* Ground I-III, Preamble [1], Elements [1.1], [1.2], [26.1], and [26.2].

d) **Element [26.3]**

303. *See* Grounds I-III, Elements [26.2], [26.3] and Claim 4.

e) **Element [26.4]**

304. *See* Grounds I-III, Elements [1.1], [1.2], [1.3], [25.3], [25.4], [25.5], [25.6], [26.1] and Claim 4.

28. Independent Claim 27

a) **Preamble [27]**

305. *See* Grounds I-III, Preamble [1]. .

b) **Element [27.1]**

306. *See* Grounds I-III, Preamble [1], Preamble [27], and Element [26.1].

c) **Element [27.2]**

307. *See* Grounds I-III, Elements [1.1], [1.2], [1.3], [27.1], and [27.2].

d) **Element [27.3]**

308. *See* Grounds I-III, Elements [1.4], [27.2], and [27.3].

e) **Element [27.4]**

309. *See* Grounds I-III, Elements [25.3], [27.3] and Claim 4.

f) **Element [27.5]**

310. *See* Grounds I-III, Elements [1.1], [1.2], [1.3], [25.5], [27.4], [27.5] and Claim 4.

29. Independent Claim 28

a) **Preamble [28]**

311. *See* Grounds I-III, Preamble [1], Preamble [28].

b) **Element [28.1]**

312. *See* Grounds I-III, Elements [1.1], [1.2] and [28.1].

c) **Element [28.2]**

313. *See* Grounds I-III, Elements [1.3], [28.1], and [28.2].

d) **Element [28.3]**

314. *See* Grounds I-III, Elements [1.4], [28.2], and [28.3].

e) **Element [28.4]**

315. *See* Grounds I-III, Elements [28.3], [28.4] and Claim 4.

f) **Element [28.5]**

316. *See* Grounds I-III, Elements [1.2] and [28.4].

g) **Element [28.6]**

317. *See* Grounds I-III, Elements [1.1], [1.2], [1.3], [25.5], [28.5], and Claim 4.

h) **Element [28.7]**

318. *See* Grounds I-III, Elements [1.1], [1.2], [1.3], [28.6] and Claim 4.

i) **Element [28.8]**

319. See Grounds I-III, Elements [1.1], [1.3], [1.4], [25.7], [28.7], and [28.8].

D. Ground IV: Claims 22 and 24 Are Rendered Obvious Over *Handley* in Combination with *Basturk* and *PacketCable*

1. Motivation to Combine

320. Although *Handley* does not disclose the use of switching nodes and switching installations, in addition to the reasons described above in Sections VII.A.1, VII.A.2, and VII.A.3, a POSITA would have been motivated to add the media gateway of *PacketCable* to the combination of *Handley* and *Basturk*. A POSITA would have had a reasonable expectation of success because *PacketCable*, *Handley*, and *Basturk* all involve communications in a digital environment, in which interfaces between different types of communications networks can be easily created by a POSITA using existing hardware interfaces combined with straightforward software programming techniques to convert between different data formats and protocols. Gateways were known in the art, and a gateway was known “to mean any device that connects two or more dissimilar networks.” EX1009, 418. A POSITA would have understood and found obvious that a media gateway allows for telephony communications between devices on the data network and devices on the Public Switched Telephone Network, such as land lines and cell phones. Further, a POSITA would have understood and found

obvious the benefits of having data network devices communicate with all pre-existing telephones on the PSTN so that the data network devices can conduct a telephone call with users beyond just those on the data network. Adding the media gateway of *PacketCable* to the combination of *Handley* and *Basturk* would thus allow users of the packet-based *Handley* system to reach all the traditional landlines in the Public Switched Telephone Network, rather than be confined to calling only packet-based telephones.

2. Dependent Claim 22

321. See Grounds I-III, Element [1.2], Claims 21 and 22.

3. Dependent Claim 24

322. See Grounds I-III, Claim 23.

323. See Ground IV, Claim 22.

VIII. SECONDARY CONSIDERATIONS

324. I have seen no evidence of, and I am not aware of, any secondary considerations that would tend to show non-obviousness of the Challenged Claims of the '682 Patent, including commercial success, long-felt need, failure of others, skepticism, praise, teaching away, recognition of a problem, or copying by competitors. However, should Patent Owner assert any secondary considerations of non-obviousness, I reserve the right to respond.

IX. CONCLUSION AND DECLARATION

325. My findings and opinions set forth in this Declaration are based on my work and examination to date. I may continue my examinations in view of additional documents or other factual evidence over the course of this proceeding that may necessitate supplementing and/or refining my opinions. I reserve the right to add, alter, or delete my opinions and my Declaration upon discovery of additional information. I reserve the right to make such changes as necessary.

326. In signing this Declaration, I understand that the Declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I acknowledge that I may be subject to cross-examination in this case and that cross-examination will take place within the United States. If cross-examination is required of me, I will appear for cross-examination within the United States during the time allotted for cross-examination.

327. I declare that all statements made herein of my knowledge are true, and that all statements made on information and belief are believed to be true, and 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.

Date: May 24, 2023

By:

Henry H. Houh.

Henry H. Houh