

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SAMSUNG ELECTRONICS CO., LTD., and
SAMSUNG ELECTRONICS AMERICA, INC.,

Petitioners

v.

FOUR BATONS WIRELESS, LLC

Patent Owner

Case IPR2025-00495

U.S. Patent No. 8,239,671

**DECLARATION OF DR. NARAYAN B. MANDAYAM IN SUPPORT
OF PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 8,239,671**

Samsung v. Four Batons IPR2025-00495 Exhibit 1002

TABLE OF CONTENTS

I. INTRODUCTION1

II. BACKGROUND AND QUALIFICATIONS1

III. MATERIALS REVIEWED6

IV. LEVEL OF ORDINARY SKILL IN THE ART9

V. RELEVANT LEGAL STANDARDS10

VI. SUMMARY OF OPINIONS13

VII. OVERVIEW OF THE '671 PATENT13

VIII. OVERVIEW OF THE PRIMARY PRIOR ART REFERENCES17

 A. Background of Relevant Technology17

 B. Analogous Art19

 C. *Sood*20

 D. *Aboba*23

 E. *Lee*24

IX. CLAIM CONSTRUCTION25

 A. Lexicography25

 B. Claims 1 & 627

X. SPECIFIC GROUNDS FOR CHALLENGE30

 A. Ground 1: *Sood* Renders Obvious Claims 1-4, 7, 8, 12, 14, and 16 ...31

 1. Independent Claim 131

 a. 1[pre]: “A channel binding method based on parameter binding in a key derivation procedure for authentication of a mobile supplicant to an access network, comprising:”31

b.	1[a]: “cryptographically binding access network parameters to a key without needing to carry the parameters in authentication methods;”	38
c.	1[b]: “further including deriving a channel binding key from a channel binding master key bound to a key binding blob using a key derivation function; and”	42
d.	1[c]: “wherein said key binding blob is a string that is constructed from static parameters advertised from an authenticator.”	53
2.	Claim 2: “The method of claim 1, wherein said authentication methods include EAP methods.”	54
3.	Claims 3 and 4.....	56
a.	Claim 3: “The method of claim 1, wherein said parameters include parameters advertised by an authenticator to said supplicant.”	56
b.	Claim 4: “The method of claim 3, wherein the identity of the authenticator is one of said parameters.”.....	58
4.	Claim 7: “The method of claim 1, wherein said key binding blob is an octet-string that is constructed from static parameters advertised from an authenticator using an authenticator-suppliant protocol.”	59
5.	Claim 8: “The method of claim 1, including a network side authenticator and said supplicant using the channel binding master key for protecting an authenticator-suppliant protocol.”	59
6.	Claim 12: “The method of claim 1, further including creating multiple channel binding keys from a single channel binding master key for multiple authenticators.”	62
7.	Claims 14, 16: “The method of claim [1/12], further including creating multiple channel binding keys from a single channel binding master key for multiple authenticators using different key binding blobs for different authenticators.”	63

B.	Ground 2: <i>Sood</i> in Combination With <i>Aboba</i> Renders Obvious Claims 5 and 18	64
1.	Claim 5: “The method of claim 1, wherein said channel binding master key is at least 64 octets long.”	64
2.	Claim 18: “The method of claim 1, wherein said key derivation function is computed based on $CBK = kdf+(CBMK, KBB)$, where CBK represents channel binding key, CBMK represents channel binding master key, and KBB represents key binding blob.”	69
C.	Ground 3: <i>Sood</i> in Combination With <i>Lee</i> Renders Obvious Claims 6, 10, 11, 13, 15, 17, and 19	74
1.	Motivation to Combine <i>Sood</i> and <i>Lee</i>	74
2.	Claim 6	82
a.	6[pre]: “A channel binding method based on parameter binding in a key derivation procedure for authentication of a mobile supplicant to an access network, comprising:”	82
b.	6[a]: “using an extensible authentication protocol server to cryptographically bind access network parameters to a channel binding key and to transmit said channel binding key to an extensible authentication protocol authenticator for use by the extensible authentication protocol authenticator as an extensible authentication protocol master session key without said extensible authentication protocol authenticator needing to carry said access network parameters in extensible authentication protocol authentication methods;”	82
(i)	“using an extensible authentication protocol server to cryptographically bind access network parameters to a channel binding key and to transmit said channel binding key to an extensible authentication protocol authenticator ...”	82

(ii)	“... for use by the extensible authentication protocol authenticator as an extensible authentication protocol master session key ...”	84
(iii)	“... without said extensible authentication protocol authenticator needing to carry said access network parameters in extensible authentication protocol authentication methods;”	86
c.	6[b]: “including using said extensible authentication protocol server to derive said channel binding key from a channel binding master key bound to a key binding blob using a key derivation function;”	87
d.	6[c]: “wherein said key binding blob is a string that is constructed from static parameters advertised from said extensible authentication protocol authenticator.”	87
3.	Claim 10	88
a.	10[a]: “The method of claim 1, further including that: a) a server and the supplicant create a channel binding key used for an authenticator;”	88
b.	10[b]: “b) the server transfers the channel binding key to the authenticator; and”	90
c.	10[c]: “c) the supplicant and the authenticator verify proof of possession of the channel binding key over an authenticator-supplicant protocol.”	90
4.	Claim 11: “The method of claim 10, further including that the channel binding key is derived from a channel binding master key bound to a key binding blob associated with the authenticator using a key derivation function.”	92
5.	Claims 13, 15, 17: “The method of claim [12/14/16], further including using channel binding keys to form a hierarchical channel binding.”	93
6.	Claim 19: “The method of claim 3, wherein said authenticator is an EAP authenticator, and wherein the EAP authenticator	

	receives and processes the channel binding key as a Master Session Key (MSK).”.....	96
D.	Ground 4: <i>Sood</i> in Combination With <i>Aboba</i> and <i>Lee</i> Renders Obvious Claims 1-8 and 10-19.....	96
	1. “using an authenticator-suppliant protocol”	96
	2. “without needing to carry the parameters in authentication methods” / “without ... needing to carry said access network parameters in ... authentication methods”.....	99
XI.	SECONDARY CONSIDERATIONS	103

I. INTRODUCTION

1. I have been retained by Samsung Electronics Co., Ltd. and Samsung Electronics America, Inc. (“Samsung” or “Petitioners”) as an independent expert consultant in this proceeding before the United States Patent and Trademark Office (“PTO”).

2. I am being compensated at a rate of \$850/hour for my services in this proceeding, which is my regular and customary rate.

3. My compensation is in no way contingent on the nature of my finding, the presentation of my findings in testimony or this declaration, or the outcome of this or any other proceeding. I have no other interest in this proceeding.

4. I have been asked to consider whether certain references disclose and/or suggest the features recited in claims 1-8 and 10-19 of U.S. Patent No. 8,239,671 (“the ’671 patent”) (Ex. 1001).¹ My opinions are set forth below.

II. BACKGROUND AND QUALIFICATIONS

5. I am an independent consultant. All of my opinions stated in this declaration are based on my own personal knowledge and professional judgment. In forming my opinions, I have relied on my knowledge and extensive experience

¹ Where appropriate, I refer to exhibits which I understand will be attached to the petition for *inter partes* review of the ’671 patent (the “Petition”).

in the field of telecommunications networks, algorithms, and communication protocols.

6. I am over 18 years of age and, if I am called upon to do so, I would be competent to testify as to the matters set forth herein. A copy of my current curriculum vitae, which details my education and professional experience, is included as Ex. 1003 in this proceeding. The following provides an overview of some of my experience that is relevant to the matters set forth in this declaration.

7. I am a Distinguished Professor of Electrical and Computer Engineering at Rutgers University where I serve as the Director of the Wireless Information Network Laboratory (WINLAB), a world class wireless networking research center at Rutgers University. WINLAB has been serving as an industry-university cooperative research center since its inception in 1989. I have been writing, researching and teaching about wireless communications and networking since 1989.

8. I received my Bachelor's degree in Electrical Engineering from the Indian Institute of Technology (IIT) in Kharagpur, India in 1989 with honors. I received my M.S. in Electrical Engineering from Rice University in 1991 and my Ph. D in Electrical Engineering in 1994, also from Rice University. After completing my studies, I was hired as a Research Associate at WINLAB in 1994 before being hired as an Assistant Professor in 1996 in the Department of

Electrical and Computer Engineering at Rutgers University. I was promoted to Associate Professor in 2001, to Professor in 2003, and to Distinguished Professor in 2014.

9. I also served as the Peter D. Cherasia Endowed Faculty Scholar at Rutgers University from 2010 to 2014, the Chair of the Electrical and Computer Engineering Department from 2016 to 2022, and as an Associate Director at WINLAB since 1999. I was a visiting faculty fellow in the Department of Electrical Engineering at Princeton University in Fall 2002, and a visiting faculty at the Indian Institute of Science in Spring 2003.

10. I conduct research in various aspects of wireless systems and networks. I teach courses at Rutgers on the topics of Wireless System Design, Wireless Communication Technologies, Wireless Revolution, Detection and Estimation Theory and Introduction to Computing for Engineers. I have co-authored 2 books on wireless networks (Principles of Cognitive Radio, Cambridge (2012) and Wireless Networks: Multiuser Detection in Cross-Layer Design, Springer (2005)), 8 book chapters and published nearly 300 papers in prestigious international journals and conferences. I have also given numerous invited presentations at a variety of industry, government, and academic forums.

11. Over the last 30 years, I have published a wide range of articles on various aspects of wireless systems addressing PHY, MAC and Network layer issues including techniques for data transmission and multimedia communications, resource allocation strategies, algorithms for wireless network security, mathematical modeling, and performance analysis. Using constructs from game theory, communications, and networking, my work has focused on system modeling and performance, signal processing as well as radio resource management for enabling wireless technologies to support various applications.

12. I have made seminal research contributions to wireless data communications on issues ranging from the systems level (such as power control, base station assignment, capacity evaluation, protocol design, medium access control, and radio resource management) to the physical layer (such as data transmission and reception). I have also done seminal work in the area of the Physical-Layer security, where properties of the radio channel (PHY layer of the protocol stack) are exploited for the purposes of secret key generation, authentication and encryption. One of my papers on this topic is also featured in the Best Readings compilation of the IEEE Communications Society (COMSOC) as the top paper on Physical-Layer Authentication. My expertise includes cellular systems such as for 2G, 3G, 4G, 5G, and I have published papers on a wide variety of topics related to the design and operation of GSM/TDMA, CDMA and LTE and

5G based systems. During this time, I have also worked extensively on wireless local area network (WLAN) and Wi-Fi technologies as well as wireless ad-hoc and sensor networks, and as such, I am quite familiar with 802.11 and Bluetooth.

13. I have also served as a technical consultant since its inception in 2002 to the company Mojo Networks Inc. (recently acquired by Arista Networks), a world leader in enterprise network security for WLANs and Wi-Fi networks that offers the next generation of intelligent edge, secure, and flexible Wi-Fi solutions. I am familiar with the early developments in the area of cellular systems (2G, 3G, 4G and 5G) and have served on several graduate thesis defense committees pertaining to all aspects of wireless systems operation.

14. I have received several prestigious awards relating to my research on wireless networks and communications: the 2015 IEEE COMSOC Advances in Communications Award for seminal work on power control in wireless data networks (this is the highest paper award given by the IEEE Communications Society to the most impactful publication in the preceding 15 years), the 2014 IEEE Donald G. Fink Award (recognizes the most outstanding tutorial paper across all IEEE publications) for the paper titled “Frontiers of Wireless and Mobile Communications” that discusses the historical and future landscape of both WAN and LAN wireless technologies, and the Fred W. Ellersick Prize from the IEEE Communications Society in 2009 for work on dynamic spectrum access models

and spectrum policy. I also received the Peter D. Cherasia Faculty Scholar Award from Rutgers University in 2010, the National Science Foundation Career Award in 1998, the Institute Silver Medal from the Indian Institute of Technology, Kharagpur, in 1989, and its Distinguished Alumnus Award in 2018.

15. I have served as an Editor for the journals IEEE Communication Letters (1999- 2002) and IEEE Transactions on Wireless Communications (2002-2004). I have also served as a guest editor of the IEEE JSAC Special Issues on Adaptive, Spectrum Agile and Cognitive Radio Networks (2007) and Game Theory in Communication Systems (2008). I was elected Fellow of the IEEE for “contributions to wireless data transmission.” I have also served as a Distinguished Lecturer of the IEEE Communications Society.

16. My experience of over 30 years with networking and telecommunications in academic and practical situations as well as my hands on experience, has given me a detailed appreciation of the technology involved with the ‘671 patent.

III. MATERIALS REVIEWED

17. The opinions contained in this declaration are based on the documents I reviewed, my professional judgment, as well as my education, experience, and knowledge regarding the field of telecommunications networks, algorithms, and communication protocols.

18. In forming my opinions expressed in this declaration, I reviewed the following materials:

- U.S. Patent No. 8,239,671 (“’671Pat”) (Ex. 1001);
- File History of U.S. Patent No. 8,239,671 (Ex. 1004);
- U.S. Patent No. 7,787,627 (“Sood”) (Ex. 1005);
- Bernard Aboba et al., “Extensible Authentication Protocol (EAP) Key Management Framework Version 3” EAP Working Group INTERNET-DRAFT (Jul. 18, 2004) (“Aboba”) (Ex. 1006);
- U.S. Patent App. No. 2004/0242228 (“Lee”) (Ex. 1007);
- U.S. Patent No. 8,027,304 (“Forsberg”) (Ex. 1008);
- IEEE Std 802.11i (“802.11i”) (Ex. 1009);
- Declaration of Laura Nugent For IETF Administration LLC (Ex. 1010);
- RFC 5247 IETF Datatracker Page (Ex. 1011);
- C. Kaufman, “Internet Key Exchange (IKEv2) Protocol” Network Working Group Request for Comments: 4306 (Dec. 2005) (“IKEv2”) (Ex. 1012);
- Excerpts from Microsoft Computer Dictionary (5th ed. 2002) (“Computer Dictionary”) (Ex. 1013);
- Excerpts from IEEE Std 802.11-1997 (“802.11-1997”) (Ex. 1014);

- B. Aboba et al., “Extensible Authentication Protocol (EAP)” EAP Working Group Request for Comments: 3748 (Jun. 2004) (“*EAP*”) (Ex. 1015);
- Morris Dworkin, “Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication,” NIST Special Publication 800-38B (May 2005) (“*Dworkin*”) (Ex. 1016);
 - U.S. Patent No. 7,602,918 (“*Mizikovsky*”) (Ex. 1017);
 - U.S. Patent No. 8,621,201 (“*Costa*”) (Ex. 1018);
 - U.S. Patent No. 7,969,945 (“*Navali*”) (Ex. 1019);
 - Nut Taesombut et al., “A Secure Multimedia System in Emerging Wireless Home Networks,” CMS 2003, LNCS 2828, pp. 76-88 (2003) (“*Taesombut*”) (Ex. 2020);

and any other materials I refer to in this declaration in support of my opinions.

19. All of the opinions contained in this declaration are based on the documents I reviewed and my knowledge and professional judgment. My opinions have also been guided by my appreciation of how a person of ordinary skill in the art would have understood the claims and the specification of the '671 patent at the time of the alleged invention, which I have been asked to consider as early as April 20, 2006. My opinions reflect how one of ordinary skill in the art would have understood the '671 patent, the prior art to the patent, and the state of the art at the time of the alleged invention.

20. It is my opinion that certain references disclose and/or suggest all the features recited in claims 1-8 and 10-19 (“Challenged Claims”) of the ’671 patent, as I discuss in detail below.

IV. LEVEL OF ORDINARY SKILL IN THE ART

21. I have been informed and understand that, in the context of an invalidity analysis, a person having ordinary skill in the art is a hypothetical person who looks to prior art at the time of the invention. I further understand that the factors that may be considered in determining the level of ordinary skill include: (1) the problems encountered in the art; (2) the prior art solutions to the problems encountered in the art; (3) the rapidity of innovation; (4) the sophistication of the technology; and (5) the education level of individuals actively working in the field. I understand that these factors need not all be considered for the analysis and that one or more of these factors may control.

22. I was asked to provide my opinion on the level of one of ordinary skill in the art with respect to the alleged invention of the ’671 patent by April 20, 2006. I am familiar with the knowledge and capabilities of a POSITA in light of the experience noted above. Specifically, my experience working with industry, undergraduate and post-graduate students, colleagues from academia, and designers and engineers practicing in industry has allowed me to become directly and personally familiar with the level of skill of individuals and the general state of

the art. Based on my consideration of the factors above, I believe a person of ordinary skill in the art would have had at least a Bachelor's of Science Degree (or equivalent) in electrical engineering, computer science, or computer engineering, or a related technical field and three years of experience analyzing and/or designing network communication protocols. A greater amount of professional education could compensate for fewer years of work experience, and vice versa.

23. As of April 20, 2006, I met, and in fact exceeded, the qualifications of a person of ordinary skill in the art. To be clear, all of my opinions in this declaration are from the perspective of one of ordinary skill in the art as I have defined it here during the relevant timeframe.

V. RELEVANT LEGAL STANDARDS

24. I am not an attorney and offer no legal opinions, but in the course of my work, I have had experience studying and analyzing patents and patent claims from the perspective of a person skilled in the art.

25. For the purposes of this declaration, I have been informed about certain aspects of the law that are relevant to forming my opinions. My understanding of the law is as follows.

26. Petitioners' counsel has informed me that a patent claim can be considered to have been obvious to a person having ordinary skill in the art at the time the application was filed. I am informed that this means that, even if all of the

requirements of a claim are not found in a single prior art reference, the claim is not patentable if the differences between the subject matter in the prior art and the subject matter in the claim would have been obvious to a person having ordinary skill in the art at the time of the invention.

27. I have been informed by Petitioners' counsel that a determination of whether a claim would have been obvious should be based upon several factors, including, among others:

- the level of ordinary skill in the art at the time the application was filed;
- the scope and content of the prior art;
- what differences, if any, existed between the claimed invention and the prior art; and
- any "secondary indicia" of non-obviousness, if they are of record.

28. I have been informed by Petitioners' counsel that a single reference can render a patent claim obvious if any differences between that reference and the claims would have been obvious to a person having ordinary skill in the art.

Alternatively, I understand that the teachings of two or more references may be combined in the same way as disclosed in the claims, if such a combination would have been obvious to one having ordinary skill in the art. In determining whether a combination based on either a single reference or multiple references would have

been obvious, the following are examples of approaches and rationales that may be considered:

- whether the teachings of the prior art references disclose known concepts combined in familiar ways, and when combined, would yield predictable results;
- whether a person of ordinary skill in the art could implement a predictable variation, and would see the benefit of doing so;
- whether the claimed elements represent one of a limited number of known design choices, and would have a reasonable expectation of success by those skilled in the art;
- whether a person of ordinary skill would have recognized a reason to combine known elements in the manner described in the claim;
- whether there is some teaching or suggestion in the prior art to make the modification or combination of elements claimed in the patent;
and
- whether the innovation applies a known technique that had been used to improve a similar device or method in a similar way.

29. I understand that “secondary indicia” of non-obviousness may include certain objective factors, such as: commercial success of products practicing the claimed invention; long-felt but unsolved need; teaching away; unexpected results;

copying; and praise by others in the field. These factors are generally referred to as “secondary considerations” or “objective indicia” of non-obviousness. I understand, however, that for such objective evidence to be relevant to the (non)obviousness of a claim, there must be a causal relationship (called a “nexus”) between the claim and the evidence and that this nexus must be based on a novel element of the claim rather than something in the prior art. I also understand that even when they are present, secondary considerations may be unable to overcome primary evidence of obviousness (such as motivation to combine with predictable results) that is sufficiently strong.

30. I have been informed by Petitioners’ counsel that one of ordinary skill in the art has ordinary creativity and is not an automaton. Petitioners’ counsel has also informed me that in considering obviousness, obviousness may not be determined using the benefit of hindsight, including hindsight derived from the patent being considered.

VI. SUMMARY OF OPINIONS

31. For the reasons I discuss below, it is my opinion that claims 1-8 and 10-19 of the ’671 patent are rendered obvious by the prior art.

VII. OVERVIEW OF THE ’671 PATENT

32. The ’671 patent titled “Channel Binding Mechanism Based on Parameter Binding in Key Derivation” relates generally to a “channel binding

mechanism,” that is “based on parameter binding in the key derivation procedure” and “cryptographically binds access network parameters to a key without need[ing] to carry those parameters in EAP methods.” Ex. 1001 (*'671Pat*), Abstract.

33. The '671 patent includes a section titled “Terminology,” which states the following:

- A “Channel Binding Key (CBK)” is “[a] key that is derived from a Channel Binding Master Key (CBMK) and cryptographically bound to a Key Binding Blob (KBB) using a Key Derivation Function (KDF).” *Id.*, 13:18-21.
- A “Channel Binding Master Key (CBMK)” is “[a] key from which a CBK is derived using a KDF.” *Id.*, 13:23-24.
- A “Key Binding Blob (KBB)” is “[a]n octet-string that is constructed from static parameters advertised from an authenticator using an Authenticator-Supplicant Protocol (ASP).” *Id.*, 13:27-30.
- A “Server” is “[a]n entity that creates a CBK and transfers it to the authenticator.” *Id.*, 13:31-33.
- An “Authenticator” is “[a] network-side entity that uses a CBK for an Authenticator-Supplicant Protocol (ASP).” *Id.*, 13:36-38.
- A “Supplicant” is “[a] user-side entity that uses a CBK for an Authenticator-Supplicant Protocol (ASP).” *Id.*, 13:43-45.

- An “Authenticator-Suppliant Protocol (ASP)” is “[a] protocol that is executed between a supplicant and an authenticator and uses a CBK for protecting the protocol.” *Id.*, 13:49-51.

34. The method disclosed in the '671 patent involves a server and a supplicant “create[ing] a CBK used for an authenticator.” *Id.*, 13:64-66. Then, “the server [] transfers the CBK to the authenticator.” *Id.*, 14:4-6. Lastly, “the supplicant and authenticator verify proof of possession of the CBK over the [Authenticator Suppliant Protocol].” *Id.*, 14:7-10. These steps, numbered (1) to (3), are depicted in figure 3. *Id.*, FIG. 3.

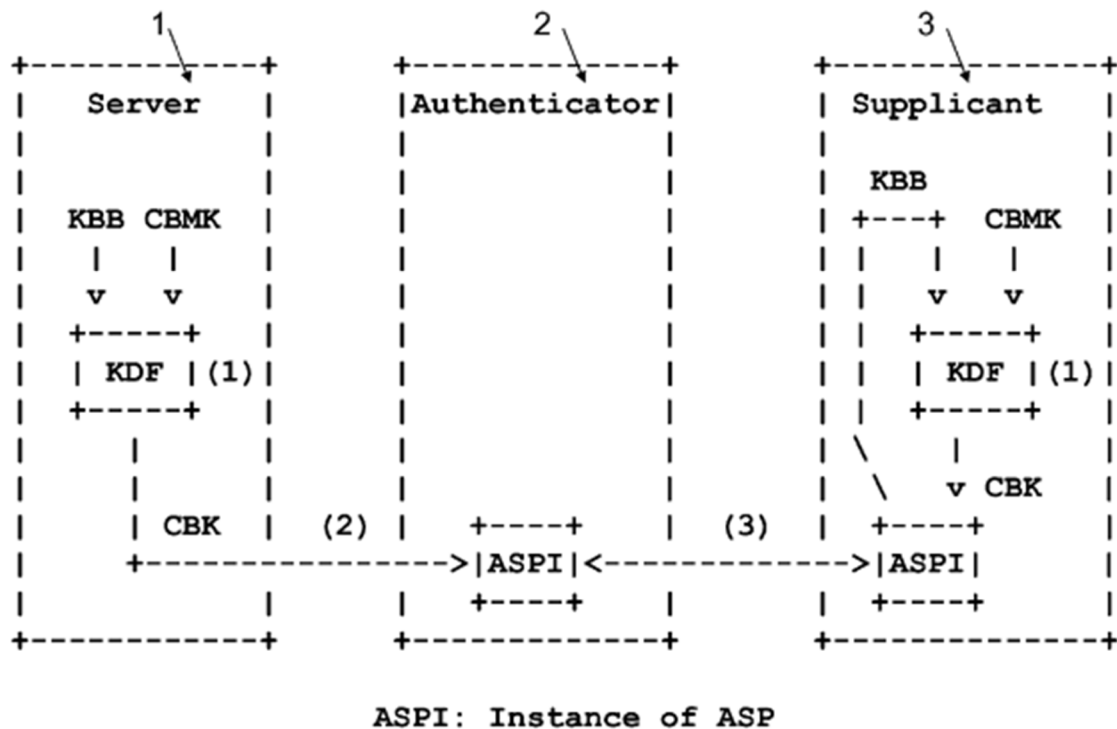


Figure 3: Basic Channel Binding Mechanism

Ex. 1001 ('671Pat), FIG. 3.

35. The '671 patent discloses several preferred embodiments of this method. In one preferred embodiment, “the KBB is pre-configured on the server.” *Id.*, 14:2-3. In another preferred embodiment, “after successful verification of proof of possession of the CBK, the supplicant and the authenticator are able to use the CBK in the ASP.” *Id.*, 14:10-12.

36. The '671 patent further discloses an embodiment that “allows CBKs to form a hierarchy.” *Id.*, 14:18-20. Such a hierarchy is depicted in figure 4:

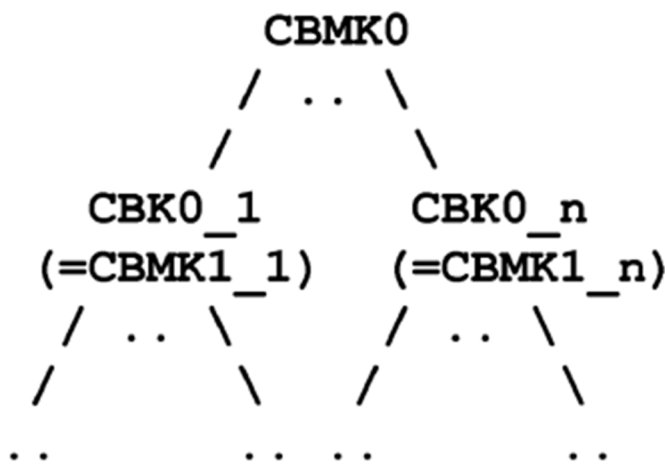


Figure 4: Hierarchical Channel Binding

Ex. 1001 ('671Pat), FIG. 4.

37. The '671 patent explains that “if there are a large number of authenticators within the administrative domain of a server, the domain can be partitioned into multiple sub-domains where each sub-domain has an authenticator.” *Id.*, 14:23-26. Thus “child CBKs” may be derived for “child

authenticators.” *Id.*, 14:26-29. The ’671 patent states that this hierarchical channel binding is also useful in a “roaming scenario” between different “administrative domains.” *Id.*, 14:30-35.

VIII. OVERVIEW OF THE PRIMARY PRIOR ART REFERENCES

A. Background of Relevant Technology

38. The Extensible Authentication Protocol (“EAP”), to which the ’671 patent is directed, is a well-known network protocol published in June 2004. Ex. 1015 (*EAP*), 1. EAP is an authentication protocol that “runs directly over data link layers” and thus is capable of authenticating network entities in environments where the higher-level Internet Protocol is not available. *Id.*

39. An important feature of EAP is “pass-through” authentication. *Id.*, 3. EAP explains that, “[r]ather than requiring the authenticator to be updated to support each new authentication method, EAP permits the use of a backend authentication server” with the authenticator passing communications through to the server. *Id.* Such a server is also referred to as an “AAA server,” where AAA refers to “Authentication, Authorization, and Accounting.” *Id.*, 4. AAA protocols used between the server and the authenticator, such as RADIUS, support the pass-through authentication procedure. *Id.*, 4, 14, 54.

40. As published, EAP discloses an optional security feature referred to as “Channel Binding.” *Id.*, 55-56. EAP explains that “[i]t is possible for a

compromised or poorly implemented EAP authenticator to communicate incorrect information to the EAP peer and/or server.” *Id.*, 55. While there existed a mechanism to detect an authenticator “attempt[ing] to impersonate another authenticator,” there would be a vulnerability if an authenticator “provide[d] correct information to the AAA server while communicating misleading information to the EAP peer via a lower layer protocol.” *Id.*, 55-56. To address this potential vulnerability, “EAP methods may support a protected exchange of channel properties.” *Id.*, 56. In other words, the server and the peer could use encrypted communications to verify that the authenticator provided the same information to both.

41. As detailed in §VII (Overview of the ’671 Patent), the ’671 patent purports to disclose an improvement to the channel binding method in EAP wherein, instead of exchanging parameters between the server and the peer, the parameters are bound to a key used by the authenticator. However, by April 2006, systems and technologies for channel binding that bound static parameters to a key used by an authenticator were well-known. *Sood*, *Aboba*, and *Lee*—discussed further below—are just a few examples of such systems.

42. As a further example, *Mizikovsky* discloses computing an “Authorization Key (AK)” which was bound to the identifier of a base station according to the formula “ $AK_i = \text{prf}(\text{PMK}, \text{BS_ID}, \text{MS_ID}, \dots)$.” Ex. 1017

(*Mizikovsky*), 2:62-3:1.² This key is “used by the supplicant and the authenticator” and “remains the same as long as the supplicant remains in contact with the same base station.” *Id.*, 3:3-6.

43. As a further example, *Costa* discloses computing “Fast-AAA-key” which is bound to “the identifier AS-Authid of the authenticator” according to the formula “Fast-AAA-key=H(Did,AS-Authid,RANDOM,AMSK’).” Ex. 1018 (*Costa*), 15:12-21. Fast-AAA-key is used “for the new communications session between the wireless user terminal [] and the new AP [].” *Id.*, 15:8-10.

44. As a further example, *Navali* discloses computing a “Transient Session Key (TSK)” which is bound to “the Foreign Agent Address.” Ex. 1019 (*Navali*), 8:51-54, 9:2-5.

B. Analogous Art

45. I have analyzed the prior art references used in the grounds below, and it is my opinion that they are analogous to the ’671 patent because they are within the same field of endeavor and reasonably pertinent to at least one problem addressed by the ’671 patent.

² I have added all **bold/italics/color** emphases and annotations unless I note otherwise.

46. In particular, *Sood* (like the '671 patent) is directed to a “key management system” that generates keys bound to static parameters and used by authenticators. Ex. 1005 (*Sood*), Abstract, 6:63-7:26, 8:3-42, 9:23-65.

47. *Aboba* (like the '671 patent) is directed to similar key generation techniques to generate “keying material used on different authenticators.” Ex. 1006 (*Aboba*), 70. *Aboba*'s keying material is disclosed to be bound to static parameters and used at an authenticator. *Id.*

48. *Lee* (like the '671 patent) is directed to a key management scheme for “providing a proactive key,” which *Lee* discloses is bound to static parameters and used at an authenticator. Ex. 1007 (*Lee*) ¶¶[0003], [0081].

C. *Sood*

49. *Sood* is titled “Methods and Apparatus For Providing a Key Management System For Wireless Communication Networks.” Ex. 1005 (*Sood*), (54). *Sood* discloses that its “key management system 200 may include an authentication server (AS) 210, a subscriber station (STA) 220, and two or more access points (APs).” *Id.*, 4:14-17. *Sood*'s figure 2 (below) illustrates *Sood*'s system.

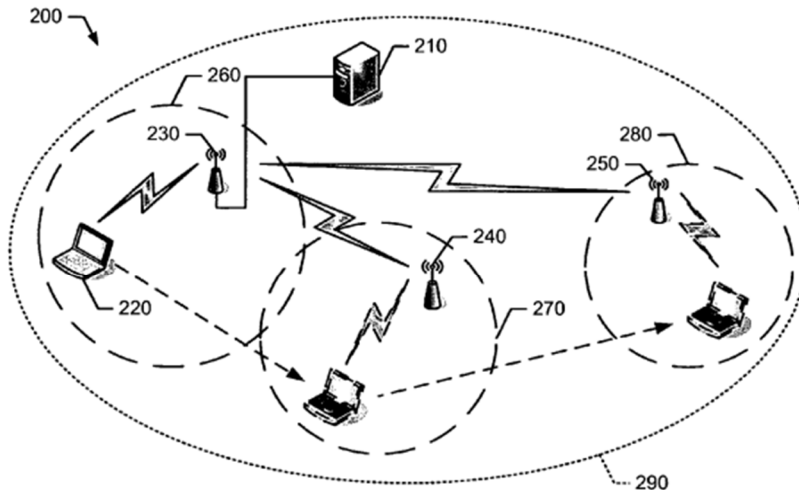


FIG. 2

Ex. 1005 (*Sood*), FIG. 2.

50. As shown, the authentication server 210 is connected to an access point 230. The access point 230 is in wireless communication with at least one subscriber station 220 within its coverage area and the other access points 240 and 250. *Id.*, 4:51-65, FIG. 2.

51. *Sood*'s system allows a subscriber to "roam between coverage areas" of different access points. *Id.* at 4:19-22. *Sood* is particularly directed to providing a key hierarchy allowing for fast roaming so that "the subscriber station 220 may avoid performing a full authentication process with the authentication server 210 when the subscriber station 220 roams from one coverage area to another." *Id.* at 6:40-45.

52. *Sood*'s key hierarchy contains three levels of key derivation. The process begins with a master secret key ("MSK"). *Id.* at 6:51-53. The MSK is

generated through communication between “the authentication server 210 and the subscriber station 220 (e.g., via a supplicant).” *Id.*, 6:53-56. Sood discloses that the server derives a new key, named “PMK-R0” from the master secret key and binds various parameters to PMK-R0 as shown in figure 5:

500 ↘
 520 ↘ 510 ↘
 522 ↘ 523 ↘ 524 ↘ 526 ↘ 528 ↘
 PMK-R0 = KDF-256(MSK, "R0 Key Derivation", SSID || MD-ID || R0KH-ID || 0x00 || SPA)

Ex. 1005 (*Sood*), FIG. 5 (excerpted).

53. *Sood* discloses that the server transfers PMK-R0 to an access point (Ex. 1005 (*Sood*), 7:64-67) and the access point then derives “PMK-R1” from PMK-R0 and again binds parameters to PMK-R1 as shown in figure 6:

600 ↘
 620 ↘ 500 ↘
 550 ↘ 624 ↘ 626 ↘
 628 ↗
 PMK-R1 = KDF-256(PMK-R0, "R1 Key Derivation", PMK-R0-Name || R1KH-ID || 0x00 || SPA)

Ex. 1005 (*Sood*), FIG. 6 (excerpted).

54. Lastly, Sood discloses that the access point derives a Pairwise Transient Key (“PTK”) from PMK-R1 and again binds parameters to PTK as shown in figure 7:

$$\begin{array}{c}
 \begin{array}{ccccccc}
 700 & & & & & & \\
 \swarrow & & \swarrow & \swarrow & & \swarrow & \swarrow & \swarrow & \swarrow \\
 & 720 & & 600 & & & 722 & 724 & 726 \\
 \text{PTK} = \text{KDF-PTKLen}(\text{PMK-R1}, \text{"PTK Key Derivation"}, \text{SNonce} \parallel \text{ANonce} \parallel \text{R0KH-ID} \parallel \\
 \text{R1KH-ID} \parallel \text{BSSID} \parallel \text{SPA}) \\
 & \swarrow & \swarrow & \swarrow & & & & & \\
 & 728 & & 730 & & & 732 & &
 \end{array}
 \end{array}$$

Ex. 1005 (*Sood*), FIG. 7 (excerpted).

D. *Aboba*

55. *Aboba* is version 3 of the draft “Extensible Authentication Protocol (EAP) Key Management Framework” standard, later finalized as RFC 5247. Ex. 1006 (*Aboba*), 1; Ex. 1011. *Aboba* is intended to supplement the Extensible Authentication Protocol defined in RFC 3748 by providing “a framework for the generation, transport and usage of keying material generated by EAP authentication algorithms.” Ex. 1006 (*Aboba*), 1.

56. Appendix E of *Aboba* discloses a technique for “fast handoff between authenticators” by providing “keying material to multiple authenticators in order to facilitate fast handoff.” Ex. 1006 (*Aboba*), 70. To generate this keying material, *Aboba* discloses a key derivation similar to *Sood*: “AAA-Key-B = PRF(EMSK(0,63), ‘EAP AAA-Key derivation for multiple attachments’, AAA-Key-A, B-Called-Station-Id, Calling-Station-Id, length).” Ex. 1006 (*Aboba*), 70.

57. I have reviewed the declaration of Laura Nugent, which indicates that an announcement of publication for *Aboba* was made on July 19, 2004 and *Aboba*

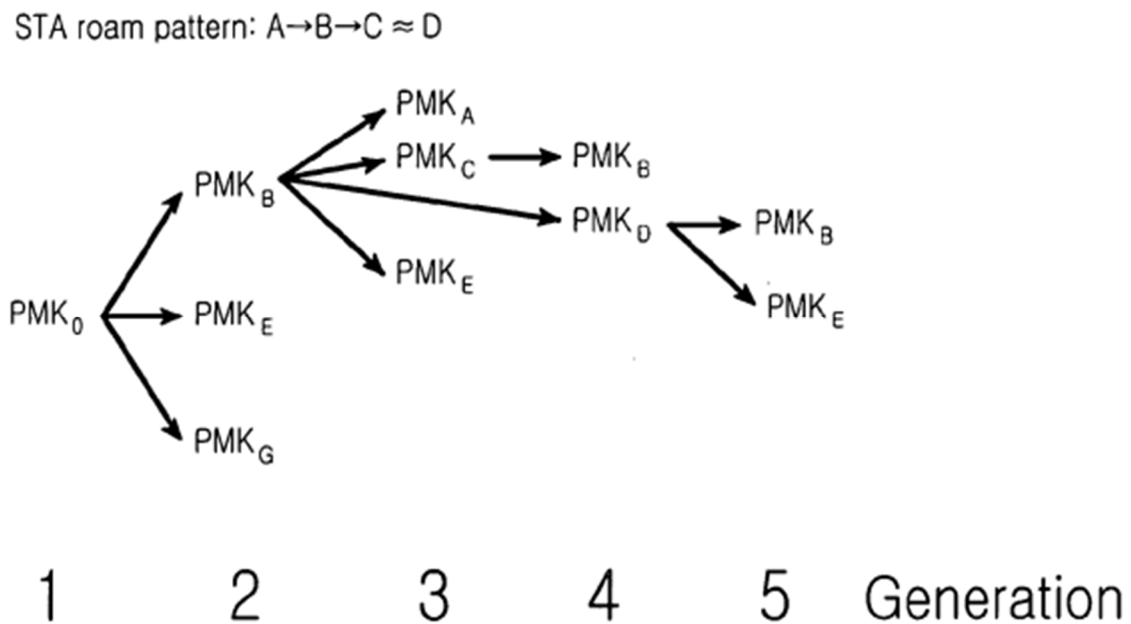
was available to the public via the IETF online directory within 24 hours of that announcement. Ex. 1010, ¶¶7, 10. As part of the online directory, *Aboba* was indexed and searchable on the IETF website. *Id.*, ¶6. IETF is a standards organization and platform for the publication of proposed internet standards that would have been well-known to a person of ordinary skill in the art at the priority date of the '671 patent.

E. *Lee*

58. *Lee* is titled “Method For Fast Roaming in a Wireless Network.” Ex. 1007 (*Lee*), (54). *Lee* discloses an authentication server and multiple access points. *Id.*, ¶[0040], FIG. 8A. A subscriber station connects to an access point and may be handed off between access points as it roams through the network. *See id.*, ¶¶[0090]-[0095], Fig. 8A-E. As in *Sood*, different keys are generated for different access points to facilitate this handoff. *Id.*, ¶[0062].

59. *Lee* is particularly directed to a system that generates keys according to an “AP-neighborhood graph.” *Id.*, ¶[0062]. As *Lee* explains, generating all necessary keys for all users requires “a large-capacity memory” at the access point. *Id.*, ¶[0060]. *Lee* instead teaches techniques for generating keys only for those access points “to which [a wireless station] may move,” i.e., a neighbor of the currently-connected access point. *Id.*, ¶[0063].

60. *Lee* additionally discloses that the derivation of keys for different access points may be performed by the server with the keys subsequently transferred to the access points. *Id.*, ¶¶[0040], [0108], [0109]. *Lee* also teaches that the pairwise master keys of different access points may be hierarchically derived from each other. *Id.*, ¶[0116]. This is shown in *Lee*'s figure 11, which shows an example key hierarchy:



Ex. 1007 (*Lee*), FIG. 11.

IX. CLAIM CONSTRUCTION

A. Lexicography

61. I understand from counsel that a patent applicant is permitted to define their own terms in a patent specification, and that this is commonly referred to as the applicant acting as his or her own lexicographer. The specification of the

'671 patent includes a section titled "Terminology." I understand from counsel that in this section the applicant should be understood to be acting as a lexicographer, and has defined certain terms as follows:

- A "Channel Binding Key (CBK)" is "[a] key that is derived from a Channel Binding Master Key (CBMK) and cryptographically bound to a Key Binding Blob (KBB) using a Key Derivation Function (KDF)." Ex. 1001 ('671Pat), 13:18-21.
- A "Channel Binding Master Key (CBMK)" is "[a] key from which a CBK is derived using a KDF." *Id.*, 13:23-24.
- A "Key Binding Blob (KBB)" is "[a]n octet-string that is constructed from static parameters advertised from an authenticator using an Authenticator-Supplicant Protocol (ASP)." *Id.*, 13:27-30.
- A "Server" is "[a]n entity that creates a CBK and transfers it to the authenticator. A server is a creator as well as a sender of the CBK." *Id.*, 13:31-33.
- An "Authenticator" is "[a] network-side entity that uses a CBK for an Authenticator-Supplicant Protocol (ASP)." *Id.*, 13:36-38.
- A "Supplicant" is "[a] user-side entity that uses a CBK for an Authenticator-Supplicant Protocol (ASP)." *Id.*, 13:43-45.

- An “Authenticator-Supplicant Protocol (ASP)” is “[a] protocol that is executed between a supplicant and an authenticator and uses a CBK for protecting the protocol.” *Id.*, 13:49-51.

62. In performing my analysis and forming my opinions, I have applied the definitions above.

B. Claims 1 & 6

63. Claims 1 and 6 state “[deriving a/derive said] channel binding key from a channel binding master key bound to a key binding blob using a key derivation function.”

64. Reviewed in isolation, I believe the language of this term is ambiguous because it could be read to require that the key binding blob be bound to either the channel binding key or the channel binding master key. However, in my opinion the definition of “Channel Binding Key” in the Terminology section of the specification resolves this ambiguity, because it states that a channel binding key “is derived from a Channel Binding Master Key (CBMK) *and* cryptographically bound to a Key Binding Blob (KBB).” Ex. 1001 (*'671Pat*), 13:18-21. A person of ordinary skill in the art would have understood the word “and” in the definition to mean that the key binding blob is bound to the channel binding key, not the channel binding master key.

65. Similarly, the “Channel Binding Mechanism” section of the specification, which discloses the core functionality of the invention, describes the “CBK derived from a CBMK *and* bound to a KBB.” *Id.*, 13:66-14:2. In contrast, the specification merely defines the channel binding master key as “[a] key from which a CBK is derived using a KDF” without mentioning the key binding blob. *Id.*, 13:23-24.

66. Additionally, in my opinion certain dependent claims of the ’671 patent would not be intelligible unless the key binding blob is bound to the channel binding key, not the channel binding master key. For example, claim 18 depends on claim 1 and further recites “wherein said key derivation function is computed based on $CBK = kdf+(CBMK, KBB)$, where CBK represents channel binding key, CBMK represents channel binding master key, and KBB represents key binding blob.” *Id.*, Claim 18. Note that “said key derivation function” in claim 18 refers back to “deriving a channel binding key from a channel binding master key bound to a key binding blob using *a key derivation function*” in claim 1. Thus, in claim 18, the key derivation function that binds the key binding blob in claim 1 has the form “ $CBK = kdf+(CBMK, KBB)$.” A person of ordinary skill in the art would have understood that this notation indicates that the key binding blob, as an input to the key derivation function, is being bound to the channel binding key as the output of the key derivation function. For example, the ’671 patent identifies the

key derivation function defined in the IKEv2 standard as a suitable key derivation function. Ex. 1001 (*'671Pat*), 14:59-15:27. Looking to the IKEv2 standard, the key derivation function defined therein outputs keying material “based on the inputs to the prf,” i.e., keying material bound to the inputs of the function. Ex. 1012 (*IKEv2*), 28. Moreover, there is no key derivation function that would have been known to a person of ordinary skill in the art that would take the channel binding master key as input, and would result in the key binding blob being bound to the same inputted channel binding master key.

67. In my opinion dependent claims 14 and 16 would also not have been intelligible to a person of ordinary skill in the art unless the key binding blob is bound to the channel binding key. These claims directly or indirectly depend on claim 1 and further recite “further including creating multiple channel binding keys from a single channel binding master key for multiple authenticators using different key binding blobs for different authenticators.” To bind different key binding blobs to different channel binding keys, the key binding blobs would be bound to the individual channel binding keys, not the “single channel binding master key.” A person of ordinary skill in the art would have understood that if the key binding blobs are bound to the “single channel binding master key,” then it would not be possible to use different key binding blobs for different channel

binding keys/authenticators because the channel binding keys would be bound to the same key binding blobs via the single channel binding master key.

68. I have given the remaining claim terms their plain and ordinary meaning, as would have been understood by a person having ordinary skill in the art at the time of the alleged invention, which I understand is April 20, 2006, having taken into consideration the language of the claims, the specification, and the prosecution history of record. I reserve the right to respond to any construction questions that may be raised in this matter.

X. SPECIFIC GROUNDS FOR CHALLENGE

69. In the sections below, I explain in detail how claims 1-8 and 10-19 of the '671 patent are not patentable over the prior art. More particularly:

70. **Ground 1** challenges claims 1-4, 7, 8, 12, 14, and 16 as obvious over *Sood*.

71. **Ground 2** challenges claims 5 and 18 as obvious over the combination of *Sood* and *Aboba*.

72. **Ground 3** challenges claims 6, 10, 11, 13, 15, 17, and 19 as obvious over the combination of *Sood* and *Lee*.

73. **Ground 4** challenges claims 1-8 and 10-19 as obvious over the combination of *Sood*, *Aboba*, and *Lee*.³

A. Ground 1: *Sood* Renders Obvious Claims 1-4, 7, 8, 12, 14, and 16

1. Independent Claim 1

a. 1[pre]: “A channel binding method based on parameter binding in a key derivation procedure for authentication of a mobile supplicant to an access network, comprising:”

74. I understand that “[a] channel binding method based on parameter binding in a key derivation procedure for authentication of a mobile supplicant to an access network, comprising:” is the preamble to claim 1 of the ’671 patent. I have been asked to assume that the preamble is a claim limitation. Under that assumption, in my opinion, *Sood* teaches the preamble.⁴

³ To the extent my declaration refers to prior art references other than those recited for each of the above-listed grounds 1-4, I am using such references as evidence supporting what I believe would have been known to a person of ordinary skill in the art at the time of the alleged invention, for example, about the state of the art and/or the benefits and advantages of certain design choices.

⁴ I have been informed by counsel that that the term “teaches” includes both express teaching as well as those fairly suggested to a person of ordinary skill in the art.

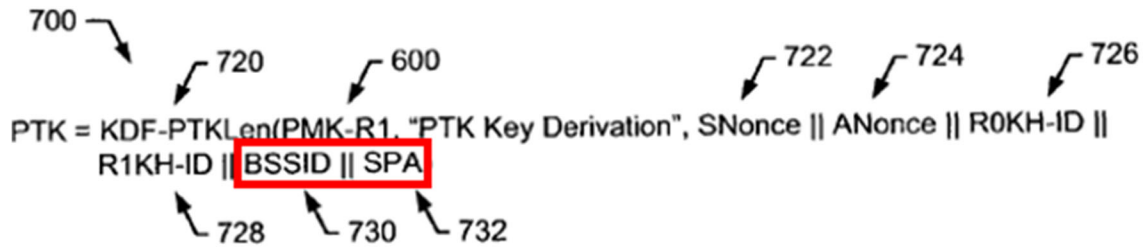
75. A person of ordinary skill in the art would have understood “channel binding” as used in the claims and specification of the ’671 patent to refer to the process of binding together different layers of a communications protocol. For example, *Forsberg* gives an example of channel binding:

At a high level, the KDF is fed with the KR key and AP identity information and the result is a session key that is bound to the AP's identity. This mechanism is called channel binding.

Ex. 1008 (*Forsberg*), 3:10-13. The channel binding described in *Forsberg* binds together a higher-layer KR key that is “is formed as a result of an authentication protocol run” and the lower level identity of an access point, thus binding the layers together. *Id.*, 2:20-23.

76. The ’671 patent’s specification is consistent with this understanding, explaining that “Channel Bindings include *lower layer parameters* that are verified for consistency between the EAP peer and server.” Ex. 1001 (*’671Pat*), 5:5-6.

77. A person of ordinary skill in the art would have understood that *Sood*’s three-level key derivation process is a channel binding method because this process involves binding lower-level MAC-layer identifiers to a higher level authentication key. For example, in the derivation of the Pairwise Transient Key (PTK), *Sood* discloses binding the values “BSSID” (basic service set identifier) and “SPA” (sender protocol address) to the key:



Ex. 1005 (*Sood*), FIG. 7 (excerpted and annotated).

78. *Sood* explains that BSSID “include[s] the **MAC address** of the access point” while the SPA “include[s] the **MAC address** ... of the subscriber station 220.” Ex. 1005 (*Sood*), 9:63-65. This would have been consistent with the understanding of a person of ordinary skill in the art regarding the term “channel binding.” Furthermore, *Sood*’s channel binding method is “based on parameter binding in a key derivation procedure” because the BSSID and SPA each include the address of a network entity, and are thus “parameters” as a person of ordinary skill in the art would have understood. Thus, *Sood* discloses binding MAC-layer identifiers (parameter binding) in the derivation of PTK (a key derivation procedure).

79. Regarding the limitation “authentication of a mobile supplicant to an access network,” *Sood*’s disclosures are directed to “a key management system *for wireless communication networks*.” Ex. 1005 (*Sood*), 1:55-57. One such network is illustrated in *Sood*’s figure 2:

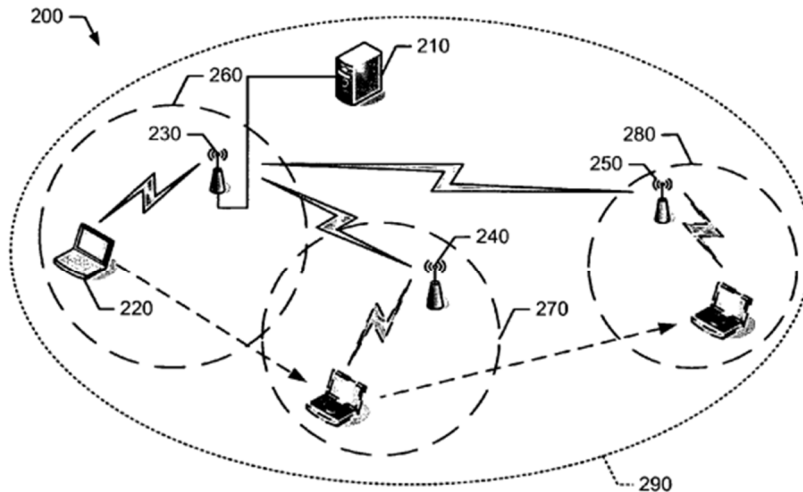


FIG. 2

Ex. 1005 (*Sood*), FIG. 2.

80. *Sood* further discloses that “the authentication server 210 and the subscriber station 220 (e.g., *via a supplicant*) may communicate with each other to generate the MSK (e.g., *a part of an authentication process*) (410).” Ex. 1005 (*Sood*), 6:53-56. As described in the specification and shown in figure 2, the subscriber station may be a mobile device such as a laptop:

The wireless communication system 100 may also include one or more subscriber stations, generally shown as 140, 142, 144, 146, and 148. For example, the subscriber stations 140, 142, 144, 146, and 148 may include wireless electronic devices *such as a desktop computer, a laptop computer, a handheld computer, a tablet computer, a cellular telephone, a pager, an audio and/or video player (e.g., an MP3 player or a DVD player), a gaming device, a video camera, a digital camera, a navigation device (e.g., a GPS device), a wireless peripheral (e.g., a printer, a scanner, a headset, a keyboard, a*

mouse, etc.), a medical device (e.g., a heart rate monitor, a blood pressure monitor, etc.), and/or other suitable fixed, portable, or mobile electronic devices. Although FIG. 1 depicts five subscriber stations, the wireless communication system 100 may include more or less subscriber stations.

Ex. 1005 (*Sood*), 2:5-19.

81. As discussed in §IX.A (Lexicography), the '671 patent defines a “supplicant” as “[a] user-side entity that uses a CBK for an Authenticator-Supplicant Protocol (ASP).” As shown in figure 2 above, *Sood*'s subscriber stations are user side entities, for example, laptop computers or components thereof. These supplicants utilize the PMK-R1 key (“channel binding key”) to generate a session key for use in communications:

To establish a session for communication services within the coverage area 270, the subscriber station 220 and the access point 240 may generate *a session key* associated with the access point 240 (e.g., PTK2).

Ex. 1005 (*Sood*), 9:13-17, FIG. 7 (showing the derivation of PTK from PMK-R1).

82. *Sood* discloses using the IEEE 802.11i protocol to communicate (“authenticator-supplicant protocol”) because it states “the methods and apparatus disclosed herein may be applied to WPANs, *WLANs*, WMANs, and/or WWANs,” and gives as an example of a WLAN “the IEEE std. 802.11i” (ASP). Ex. 1005

(*Sood*), 11:56-58, 2:37-44, 9:45-48. A person of ordinary skill in the art would have understood that the well-known 802.11i protocol meets the definition of an authenticator-suppliant protocol because it is “[a] protocol that is executed between a supplicant and an authenticator and uses a CBK for protecting the protocol.” Ex. 1001 (*'671Pat*), 13:49-51. In particular, the 802.11i protocol utilizes a “PMK” to generate session keys for use within the protocol. *See* Ex. 1009 (*802.11i*), 90 (showing the derivation of session keys within the 802.11i protocol). A person of ordinary skill in the art would have understood that the “PMK” of the 802.11i protocol is analogous to *Sood*’s PMK-R1 (“channel binding key”) because both keys are used to derive a “PTK” using similar methods. The 802.11i protocol specification derives PTK by using PMK as the input to a key derivation function, as shown below:

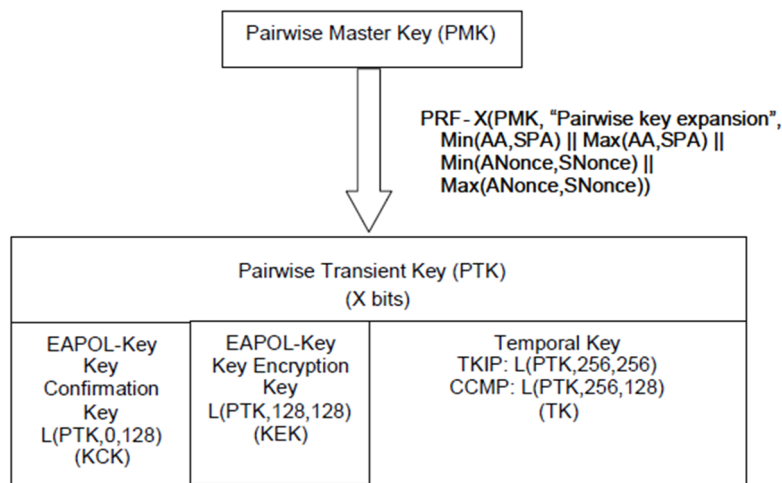


Figure 43s—Pairwise key hierarchy

Ex. 1009 (*802.11i*), 90.

83. *Sood* discloses a similar key derivation formula in figure 7 with similar inputs, including nonces:

In the example of FIG. 7, for example, the derivation of PTK 700 may be based on a key derivation function (KDF) 720, the PMK-R1 600 of FIG. 6, and concatenations of information elements in *a first nonce field 722, a second nonce field 724*, a first NAS identifier field 726, a second NAS identifier 728, a basic service set identifier (BSSID) field 730, and a SPA field 732.

Ex. 1005 (*Sood*), 9:36-42. Additionally, the '671 patent identifies the 802.11i protocol as an example of an authenticator-supplicant protocol:

In the preferred embodiments, *any ASP* (e.g., PANA and *IEEE 802.11i*) that supports the mechanisms of embodiments of the present invention should define how a KBB is constructed from the parameters specific to the ASP, where the KBB construction mechanism should satisfy the following requirements: ...

Ex. 1001 ('671*Pat*), 15:53-58.

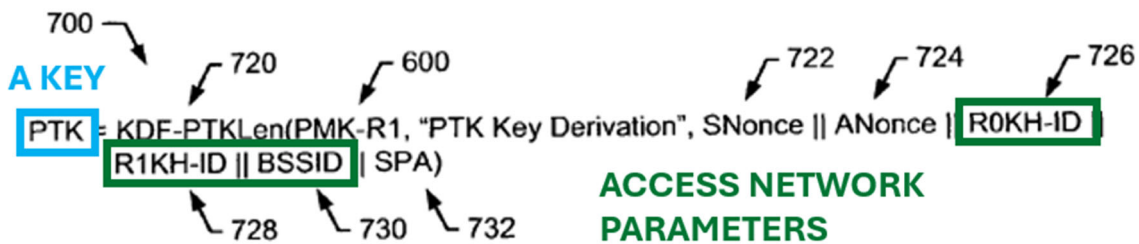
84. Accordingly, it is my opinion that a person of ordinary skill in the art would have understood *Sood* to teach a method for binding MAC-layer parameters in a higher-level key derivation (“channel binding method based on parameter binding in a key derivation procedure”) as part of an authentication process between a subscriber station’s supplicant and an authentication server on a wireless

communications network. (“for authentication of a mobile supplicant to an access network”).

b. 1[a]: “cryptographically binding access network parameters to a key without needing to carry the parameters in authentication methods;”

85. In my opinion, *Sood* teaches this feature.

86. For example, *Sood* discloses a key derivation in figure 7 that cryptographically binds at least **R0KH-ID**, **R1KH-ID**, and **BSSID** (“access network parameters”) to the **PTK** (“a key”):



Ex. 1005 (*Sood*), FIG. 7 (excerpted and annotated).

87. A key derivation function, such as the “KDF-PTKLen” function shown in *Sood*’s figure 7, is a kind of function that would be well-known to a person of ordinary skill in the art. For example, the well-known IKEv2 specification defines such a function, which *IKEv2* refers to as a “pseudo-random function.” Ex. 1012 (*IKEv2*), 27-28. This kind of function is configured to take parameters and/or keys as input and output a new key cryptographically bound to the inputs. *See Id.*, 27-28 (defining “ $prf^+(K,S) = T1 | T2 | T3 | T4 | \dots$ ” as a

function that takes input parameters K (key) and S (string) and outputs keying material T1-T4 generated from the inputs). Here, **R0KH-ID**, **R1KH-ID**, and **BSSID** are inputs to the KDF-PTKLen function and, thus, are bound to the **PTK** by the key derivation.

88. Furthermore, *Sood* discloses **R0KH-ID**, **R1KH-ID**, and **BSSID** are parameters of the access network. The **R0KH-ID** parameter identifies “the *network entity* holding the first-level derived authentication key,” for example, “the authentication server 210 of FIG. 2.” Ex. 1005 (*Sood*), 7:10-16. The **R1KH-ID** parameter also identifies a “network entity,” for example, an “access point.” *Id.*, 8:27-32. The **BSSID** “include[s] the MAC address of the access point,” which a person of ordinary skill in the art would have understood to be another parameter that identifies an access point. *Id.*, 9:63-64; Ex. 1006 (*Aboba*), 70 (indicating that a MAC address serves as an identifier of an access point: “B-Called-Station-Id = AP B MAC address”).⁵ Each of these parameters are access network parameters because they are associated with components (either authentication servers or access points) of the access network. This is consistent with the specification of the

⁵ To be clear, here I am using *Aboba* as evidence of a person of ordinary skill in the art’s knowledge. *Aboba* is not part of Ground 1.

'671 patent which discloses “the identity of the EAP authenticator” as an example of an access network parameter:

Each access network has its own set of parameters advertised by the EAP authenticator to EAP peers. *By way of example, the identity of the EAP authenticator is one of such parameters.*

Ex. 1001 ('671Pat), 12:59-62.

89. A person of ordinary skill in the art would have understood that *Sood's* method does not need to carry the access network parameters in authentication methods. *Sood* discloses that at least **R0KH-ID** and **R1KH-ID** are advertised by the access point to the subscriber station in a “beacon”:

The NAS identifier field 524 (e.g., R0KH-ID) may include a value to identify the network entity holding the first-level derived authentication key (e.g., a key holder of R0). In particular, the subscriber station 220 may establish full authentication with the NAS indicated by the NAS identifier field 524 (e.g., the authentication server 210 of FIG. 2). For example, the NAS identifier field 524 may be an Internet Protocol (IP) address or a string octet greater than three (3) octets. *The value of the NAS identifier field 524 may be advertised in a beacon*, a probe response, or a neighbor report.

The NAS identifier field 624 (e.g., R1KH-ID) may include a value to identify the network entity holding the second-level derived authentication key (e.g., a key holder of R1). In particular, the subscriber station 220 may roam to the coverage area of the access

point associated with the NAS indicated by the NAS identifier field 624. For example, the NAS identifier field 624 may be an Internet Protocol (IP) address or a string octet greater than three (3) octets.

The value of the NAS identifier field 624 may be advertised in a beacon, a probe response, or a neighbor report

Ex. 1005 (*Sood*), 7:10-20, 8:27-36.

90. A person of ordinary skill in the art would have also understood that the **BSSID** discussed in *Sood* would also have been available via a beacon, as this would have been a well-understood functionality of wireless networks. For example, starting with the very earliest 802.11 specification, the BSSID was baked into the header of all management frames, including beacon frames:

There are four address fields in the MAC frame format. These fields are used to indicate the **BSSID**, source address, destination address, transmitting station address, and receiving station address.

Ex. 1014 (*802.11-1997*), 13. Because at least these three access network parameters (**R0KH-ID, R1HK-ID, and BSSID**) are available to the supplicant via beacon(s), there is no need to carry these parameters in authentication methods.

91. A person of ordinary skill in the art would have understood that advertising parameters is not carrying parameters in authentication methods because advertising (e.g., in a beacon) occurs pre-authentication. Specifically, advertising occurs in the “discovery” phase before authentication begins, as a

person of ordinary skill in the art would have understood. *See* Ex. 1007 (*Lee*) ¶[0017] (“The complete roaming process can be divided into two distinct logical steps: discovery and re-authentication as described below.”). The ’671 patent confirms this understanding, disclosing that access network parameters may be bound to a key “without needing to carry the parameters in authentication methods,” but also that “[i]n some other examples, the parameters include parameters advertised by an authenticator to the supplicant.” Ex. 1001 (*’671Pat*), 10:11-19. Thus, the ’671 patent explicitly distinguishes between carrying parameters in authentication methods on one hand and advertising parameters on the other.

92. Accordingly, it is my opinion that a person of ordinary skill in the art would have understood *Sood* to teach cryptographically binding **R0KH-ID**, **R1KH-ID**, and **BSSID** (“access network parameters”) to **PTK** (“a key”) by advertising the parameters to the subscriber station (“without needing to carry the parameters in authentication methods”).

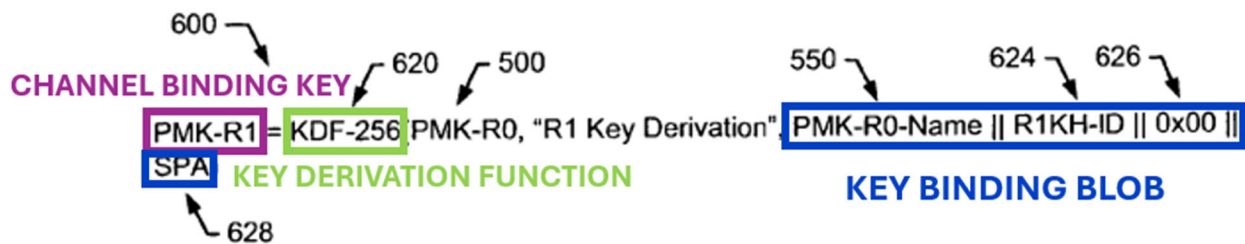
c. 1[b]: “further including deriving a channel binding key from a channel binding master key bound to a key binding blob using a key derivation function; and”

93. In my opinion, *Sood* teaches this feature.

94. *Sood* discloses two different keys, each of which independently meets the requirements of a “channel binding master key” in the ’671 patent: **MSK** and

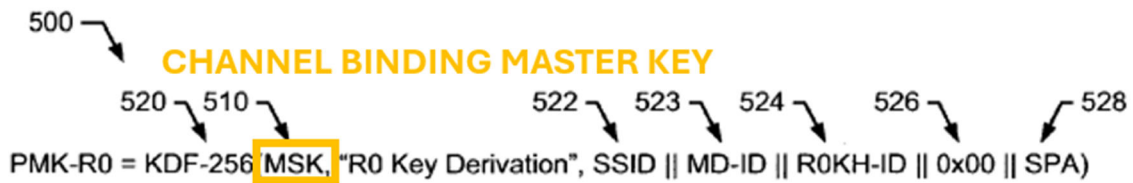
PMK-R0. Here, I first discuss how limitation 1[b] is met with **MSK** as the channel binding master key. Then, I discuss how limitation 1[b] is met with **PMK-R0** as the channel binding master key.

95. *Sood* discloses deriving **PMK-R1** (“channel binding key”) from **MSK** (“channel binding master key”) using a function named **KDF-256** (“key derivation function”) as shown in figure 6:



Ex. 1005 (*Sood*), FIG. 6 (excerpted and annotated).

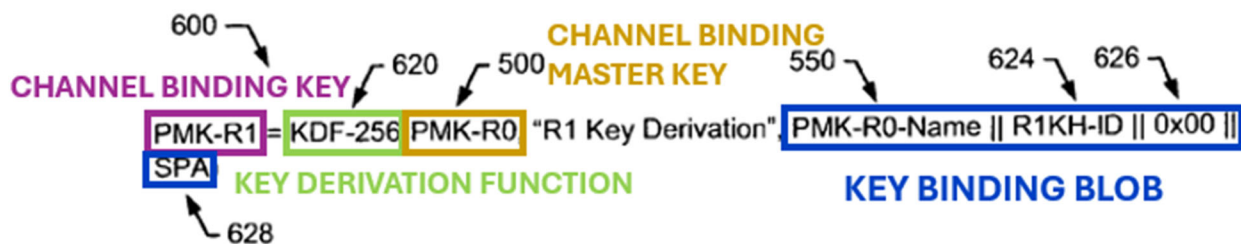
A person of ordinary skill in the art would have understood that, because PMK-R0 is an input to, and **PMK-R1** is the output of, **KDF-256** (“key derivation function”), *Sood* teaches deriving **PMK-R1** from PMK-R0. PMK-R0 is in turn derived from **MSK** using the same KDF-256 function, as shown in figure 5:



Ex. 1005 (*Sood*), FIG. 5 (excerpted and annotated).

96. It is therefore my opinion that a person of ordinary skill in the art would have understood that **PMK-R1** is ultimately derived from **MSK** (“channel binding master key”).

97. Additionally, *Sood*’s **PMK-R0** can separately and independently be mapped to the claimed channel binding master key. More particularly, *Sood* discloses deriving **PMK-R1** (“channel binding key”) from **PMK-R0** (“channel binding master key”) using the function **KDF-256** (“key derivation function”) as shown in figure 6:



Ex. 1005 (*Sood*), FIG. 6 (excerpted and annotated).

98. Regardless of whether **MSK** or **PMK-R0** is mapped to the claimed “channel binding master key,” *Sood* also teaches binding the channel binding key to a key binding blob as recited in 1[b]. As discussed in §IX.A (Lexicography), the ’671 Patent defines a channel binding key as “[a] key that is derived from a Channel Binding Master Key (CBMK) and cryptographically bound to a Key Binding Blob (KBB) using a Key Derivation Function (KDF).” **PMK-R1** (“channel binding key”) is cryptographically bound to **the concatenation of**

PMK-R0-Name, R1KH-ID, 0x00, and SPA (together the “key binding blob”) as shown in figure 6 because this concatenation is an input to the key derivation function **KDF-256**. A person of ordinary skill in the art would have understood that inputting **the concatenation** to the key derivation function binds it to the output of the function, i.e., **PMK-R1**.

99. As also discussed in §IX.A (Lexicography), the ’671 Patent defines a key binding blob as “[a]n octet-string that is constructed from static parameters advertised from an authenticator using an Authenticator-Supplicant Protocol (ASP).” A person of ordinary skill in the art would have recognized the “||” operators that form the key binding blob in figure 6 to be well-known notation for a concatenation operation. For example, a computer security publication from the National Institute of Standards and Technology titled “Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication” explains that the notation “X || Y” indicates “[t]he concatenation of two bit strings X and Y.” Ex. 1016 (*Dworkin*), 9. The publication further confirms that “[t]he concatenation operation on bit strings is denoted ||.” *Id.*, 10.

100. Additionally, *Sood* explains that its lists of elements joined with the “||” operators are “concatenations of information elements.”

In the example of FIG. 6, for example, the derivation of PMK-R1 600 may be based on a key derivation function (KDF) 620, the PMK-R0

500 of FIG. 5, and *concatenations of information elements* in the PMK-R0-Name 550 of FIG. 5, an NAS identifier field 624, a separator field 626, and a SPA field 628.

Ex. 1005 (*Sood*), 8:21-26. Concatenation is an operation performed on two strings to join them into a single string. Ex. 1013 (*Computer Dictionary*), 4 (defining “concatenate” as “[t]o join sequentially (for example, to combine the two strings ‘hello’ and ‘there’ into the single string ‘hello there’).”).

101. Furthermore, *Sood* discloses that at least **R1KH-ID** may be a “string octet greater than three (3) octets.”

The NAS identifier field 624 (e.g., R1KH-ID) may include a value to identify the network entity holding the second-level derived authentication key (e.g., a key holder of R1). In particular, the subscriber station 220 may roam to the coverage area of the access point associated with the NAS indicated by the NAS identifier field 624. For example, *the NAS identifier field 624 may be* an Internet Protocol (IP) address or *a string octet greater than three (3) octets*.

Ex. 1005 (*Sood*), 8:27-35. In my opinion a person of ordinary skill in the art would have understood “string octet” to be a typo of “octet-string.” Because the key binding blob of figure 6 is formed by a concatenation of **R1KH-ID**, it is an octet string.

102. The key binding blob of figure 6 is also constructed from static parameters advertised from an authenticator. In particular, for the reasons I discuss below, a person of ordinary skill in the art would have understood that at least each of SSID (service set identifier), MD-ID (mobility domain identifier), and R1KH-ID are static parameters advertised from an authenticator that are used to construct the key binding blob “**PMK-R0-Name || R1KH-ID || 0x00 || SPA**” shown in figure 6.

103. More particularly, *Sood* discloses that MD-ID “include[s] a name defined by a network administrator and advertised by one or more access points within a mobility domain.” Ex. 1005 (*Sood*), 7:38-41. A person of ordinary skill in the art would have understood that MD-ID is static because it is defined by a network administrator. MD-ID is also advertised from an authenticator because *Sood*’s “access points” are authenticators. As I discuss in §IX.A (Lexicography), the ’671 patent defines an authenticator as “[a] network-side entity that uses a CBK for an Authenticator-Supplicant Protocol (ASP).” *Sood* discloses that access points are network-side. For example, *Sood* describes a “network entity,” and gives as an example, an “access point.”

The NAS identifier field 624 (e.g., R1KH-ID) may include a value to identify the network entity holding the second-level derived authentication key (e.g., a key holder of R1). In particular, the

subscriber station 220 may roam to the coverage area of *the access point associated with the NAS indicated by the NAS identifier field 624*.

Id., 8:27-32. Additionally, figure 2 shows access points 230, 240, and 250 providing network access to subscriber stations, making them network-side entities:

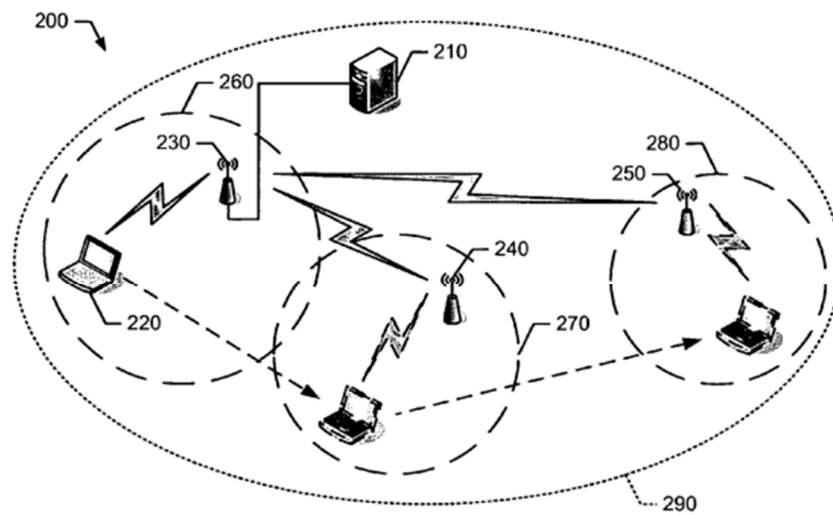
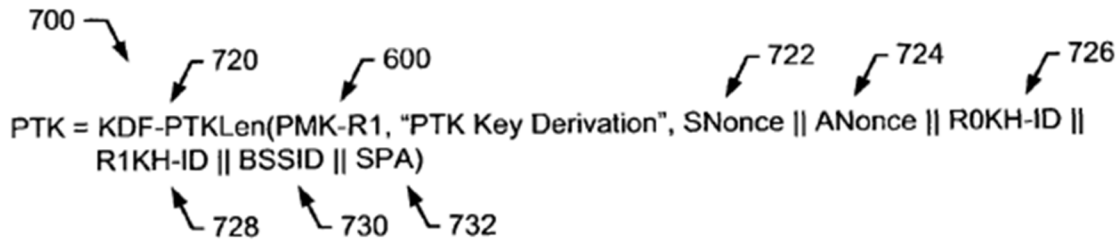


FIG. 2

Ex. 1005 (*Sood*), FIG. 2.

104. The access points also use **PMK-R1** (CBK) in an IEEE 802.11i protocol (authenticator-suppliant protocol). *Sood* discloses that “the access point 250 may generate a session key for a session between the subscriber station 220 and the access point 250.” Ex. 1005 (*Sood*), 11:38-40. A “session key” may be a “pairwise temporal key (PTK).” *Id.*, 9:5-7. As illustrated in figure 7, generating a PTK utilizes **PMK-R1** as an input:



Ex. 1005 (*Sood*), FIG. 7.

105. The PTK is used in the 802.11i protocol as explained in §X.A.1.a (Element 1[pre]).

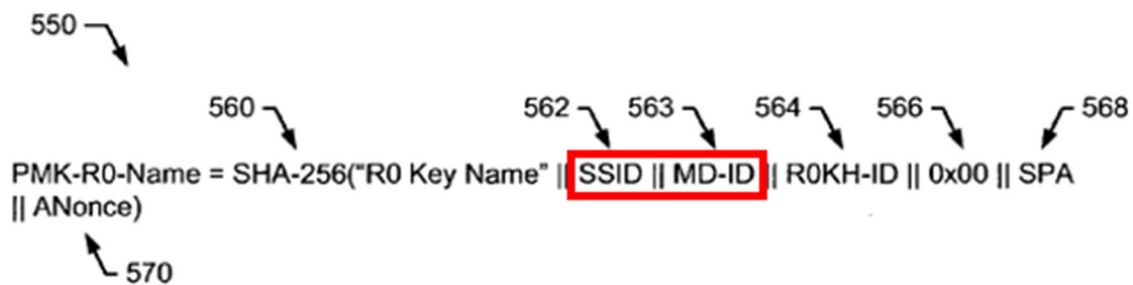
106. R1KH-ID is also a static parameter advertised by an authenticator. *Sood* discloses that R1KH-ID “may be advertised in a beacon,” which a person of ordinary skill in the art would have understood is a beacon of an access point. Ex. 1005 (*Sood*), 8:27-36. **R1KH-ID** is static because it is an identifier of a “network entity.” *Id.*, 8:27-30.

107. SSID is also a static parameter advertised by an authenticator. A person of ordinary skill in the art would have understood that advertising SSIDs is a well-known feature of wireless networks. *See* Ex. 1009 (*802.11i*), 79 (“The STA selects an authorized ESS by selecting among APs that advertise an appropriate SSID.”). An SSID identifies the wireless network service, and thus is static.

108. The static parameters SSID, MD-ID, and R1KH-ID are also advertised using the 802.11i protocol (ASP). *Sood* discloses that “the methods and apparatus disclosed herein may be applied to WPANs, *WLANs*, WMANs, and/or

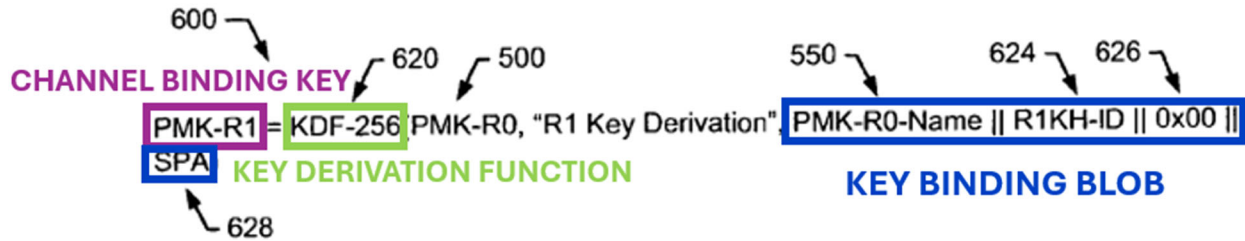
WWANs” and gives as an example of a WLAN “the IEEE std. 802.11i” (ASP). Ex. 1005 (*Sood*), 11:56-58, 2:37-44, 9:45-48. A person of ordinary skill in the art would have understood that the 802.11i protocol has advertising features. Ex. 1006 (*Aboba*), 25 (“In order to ensure that all parties can agree on the authenticator name this requires the authenticator to advertise its name (typically using a lower layer mechanism, such as the 802.11 Beacon/Probe Response)”; Ex. 1009 (*802.11i*), 38 (referring to the format for a “Beacon frame” used for advertising). Thus, a person of ordinary skill in the art would have understood that *Sood*’s disclosures regarding advertising parameters were performed in accordance with the well-known advertising capabilities of the 802.11i protocol.

109. Lastly, the static parameters SSID, MD-ID, and R1KH-ID are used to construct the concatenation of PMK-R0-Name, R1KH-ID, 0x00, and SPA (“key binding blob”) shown in *Sood*’s figure 6. First, SSID and MD-ID are inputs to a hash function that is used to generate **PMK-R0-Name**, as shown in figure 5:



Ex. 1005 (*Sood*), FIG. 6 (excerpted and annotated).

Then **PMK-R0-Name** is concatenated with other strings, including **R1KH-ID**, to form the key binding blob:



Ex. 1005 (*Sood*), FIG. 6 (excerpted and annotated).

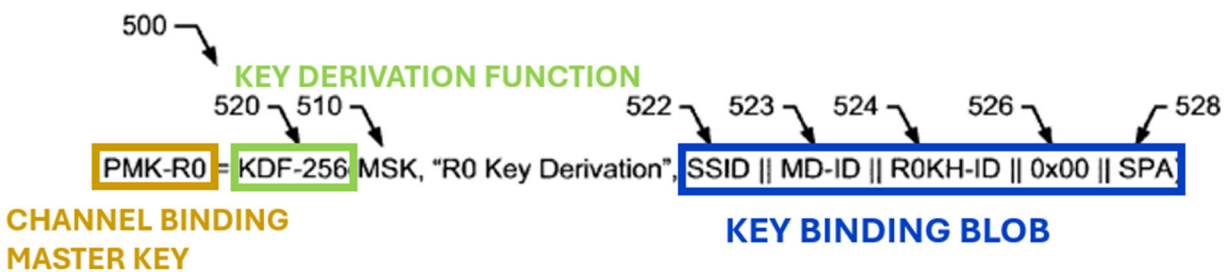
110. Accordingly, it is my opinion that a person of ordinary skill in the art would have understood *Sood* to teach deriving **PMK-R1** (“deriving a channel binding key”) from **MSK** via **PMK-R0**, either of which may be considered a channel binding master key (“from a channel binding master key”), **PMK-R1** being bound to **the concatenation of PMK-R0-Name, R1KH-ID, 0x00, and SPA** (“bound to a key binding blob”) using **KDF-256** (“using a key derivation function”).

* * *

111. As I discussed in §IX.B (Claim Construction), a person of ordinary skill in the art would have understood this element to require that the key binding blob is bound to the channel binding key (not the channel binding master key). However, to the extent that the Patent Owner (PO) argues and the Board agrees that this element requires that the key binding blob be bound to the channel

binding master key, it is my opinion that *Sood* still teaches this limitation. Here, I address this potential alternative construction of this element, which I will refer to as *PO's Potential Alternative Construction* as shorthand.

112. *Sood* discloses that **PMK-R0** (“channel binding master key”) is bound to **the concatenation of SSID, MD-ID, R0KH-ID, 0x00, and SPA** (“key binding blob”) using the function **KDF-256**, as shown in figure 5:



Ex. 1005 (*Sood*), FIG. 5 (excerpted and annotated).

113. **The concatenation of figure 5** meets the definition of a key binding blob, which is: “[a]n octet-string that is constructed from static parameters advertised from an authenticator using an Authenticator-Supplicant Protocol (ASP).” Ex. 1001 (*’671Pat*), 13:27-30. *Sood* discloses that **R0KH-ID** is an octet string. Ex. 1005 (*Sood*), 7:10-18. Because the key binding blob of figure 5 is a concatenation of **R0HK-ID**, it is also an octet string, as previously explained.

114. **The concatenation of figure 5** is constructed from at least three static parameters advertised from an authenticator using an authenticator-suppliant protocol: **SSID**, **MD-ID**, and **R0KH-ID**. **SSID** and **MD-ID** meet this criteria as

explained above. **R0KH-ID** is also a static parameter advertised by an authenticator. *Sood* discloses that **R0KH-ID** is “advertised in a beacon,” which a person of ordinary skill in the art would have understood is a beacon of an access point. Ex. 1005 (*Sood*), 7:10-20. **R0KH-ID** is static because it is an identifier of a “network entity.” Ex. 1005 (*Sood*), 7:10-13. As explained above, *Sood* teaches that values such as **R0KH-ID** were advertised in accordance with the well-known advertising capabilities of the 802.11i protocol (authenticator-suplicant protocol).

115. Accordingly, it is my opinion that a person of ordinary skill in the art would have understood *Sood* to teach deriving **PMK-R1** (“deriving a channel binding key”) from **PMK-R0** (“from a channel binding master key”), **PMK-R0** being bound to **the concatenation of SSID, MD-ID, R0KH-ID, 0x00, and SPA** (“bound to a key binding blob”) using **KDF-256** (“using a key derivation function”).

d. 1[c]: “wherein said key binding blob is a string that is constructed from static parameters advertised from an authenticator.”

116. This limitation recites the lexicography of “key binding blob” and is taught by *Sood* for the same reasons I discussed in §X.A.1.c (Element 1[b]).

2. Claim 2: “The method of claim 1, wherein said authentication methods include EAP methods.”

117. In my opinion, *Sood* teaches this claim, including under *PO’s Potential Alternative Construction*.

118. A person of ordinary skill in the art would have understood the “said authentication methods” recited in claim 2 to refer back to the limitation “cryptographically binding access network parameters to a key without needing to carry the [access network] parameters *in authentication methods*” in claim 1. For context, in the prosecution history of the ’671 patent, the applicant argued that the EAP specification (RFC 3748), which the Examiner had cited as prior art, did not disclose the limitation “without needing to carry the parameters in authentication methods” because that reference disclosed “*communicating* the access network parameters over a protected channel of an EAP method.” Ex. 1004, 56. In my opinion, therefore, the limitation “without needing to carry the parameters in authentication methods” should be understood to encompass circumstances in which there is no need to communicate the parameters using an authentication method.

119. In my opinion, here would be no need to carry (i.e., communicate) access network parameters in EAP methods for the same reason there would be no need to carry access network parameters in any other kind of method. As I explain

in §X.A.1.c (Element 1[b]), *Sood* does not need to carry access network parameters in any kind of authentication method because the subscriber station receives the necessary parameters via advertisements from access points.

120. Accordingly, it is my opinion that a person of ordinary skill in the art would have understood *Sood* to teach access points advertising parameters to a subscriber station, a procedure that does not involve EAP methods (“without needing to carry the parameters in authentication methods ... wherein said authentication methods include EAP methods”).

121. To the extent that Patent Owner argues and the Board agrees that the EAP methods of claim 2 must cryptographically bind access network parameters to a key, *Sood* discloses deriving pairwise master keys “from an authentication process such as Extensible Authentication Protocol-Transport Layer (EAP-TLS) or Protected EAP (PEAP).” Ex. 1005 (*Sood*), 6:56-62. These PMKs are used to derive a PTK. Ex. 1005 (*Sood*), 9:36-42 (“the derivation of PTK 700 may be based on ... PMK-R1 600”), 8:21-26 (“the derivation of PMK-R1 600 may be based on ... PMK-R0 500”). The PTK is bound to access network parameters, as I explained in §X.A.1.b.

122. Accordingly, it is my opinion that a person of ordinary skill in the art would have understood *Sood* to teach deriving PTKs bound to access network parameters (“cryptographically binding access network parameters to a key”) from

PMKs generated by EAP-TLS or PEAP (“in authentication methods ... wherein said authentication methods include EAP methods.”).

3. Claims 3 and 4

123. In claims 3 and 4, it is not clear what the term “said parameters” refers back to, and I address two possible interpretations: (1) “said parameters” refers back to “access network parameters,” or (2) “said parameters” refers back to “static parameters.” For the purposes of this declaration, I do not take a position on which interpretation is correct. But as I discuss below, it is my opinion that *Sood* teaches claims 3 and 4 regardless of which interpretation is applied.

a. Claim 3: “The method of claim 1, wherein said parameters include parameters advertised by an authenticator to said supplicant.”

124. In my opinion, *Sood* teaches this claim, including under *PO*’s *Potential Alternative Construction*.

125. To the extent “said parameters” refers back to “access network parameters,” *Sood*’s access network parameters (e.g., R0KH-ID, R1KH-ID, and BSSID) are advertised by the access point 230 for the reasons I discussed in §X.A.1.b (Element 1[a]). Alternatively, to the extent “said parameters” refers back to “static parameters,” *Sood*’s static parameters (e.g., SSID, MD-ID, and R1KH-ID) are advertised by the access point 230 for the reasons I discussed in §X.A.1.c (Element 1[b]).

126. As I also explained in §X.A.1.c (Element 1[b]), access points are authenticators. A person of ordinary skill in the art would have understood that when parameters are advertised by the access point, they are available to any station within the access point’s coverage area. This understanding is buttressed by *Sood*’s figure 2, which shows subscriber station 220 (“supplicant”) within the coverage area 260 of access point 230 (“authenticator”). See Ex. 1005 (*Sood*), 4:57-65.

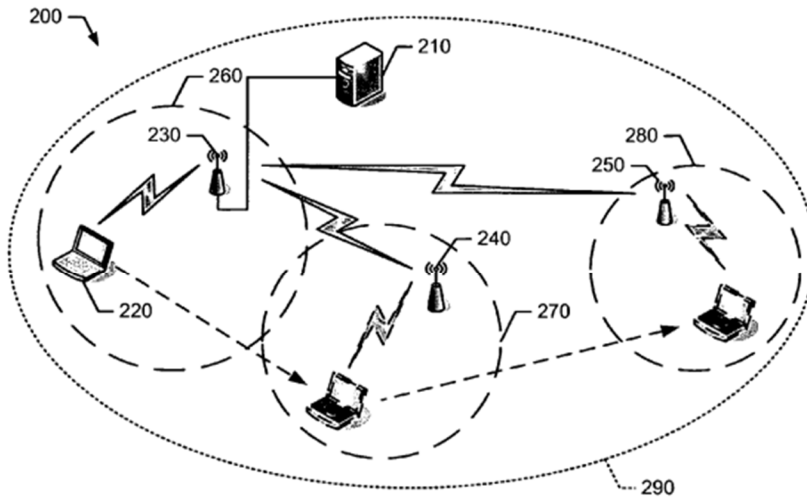


FIG. 2

Ex. 1005 (*Sood*), FIG. 2.

Thus, a person of ordinary skill in the art would have understood *Sood* to teach that parameters advertised by access point 230 are advertised to subscriber station 220.

127. Accordingly, it is my opinion that a person of ordinary skill in the art would have understood *Sood* to teach that both the access network parameters and the static parameters (one or the other, “said parameters”) are parameters

advertised by the access point (“include parameters advertised by an authenticator”) to a subscriber station (“to said supplicant”).

b. Claim 4: “The method of claim 3, wherein the identity of the authenticator is one of said parameters.”

128. In my opinion, *Sood* teaches this claim, including under *PO*’s *Potential Alternative Construction*.

129. To the extent “said parameters” refers back to “access network parameters” in claim 1, *Sood*’s access network parameters include R1KH-ID for the reasons I explained in §X.A.1.b (Element 1[a]). To the extent “said parameters” refers back to “static parameters” in claim 1, *Sood*’s static parameters also include R1KH-ID for the reasons I explained in §X.A.1.c (Element 1[b]).

130. *Sood* explains that R1KH-ID “include[s] a value to identify the network entity holding the second-level derived authentication key (e.g., a key holder of R1).” Ex. 1005 (*Sood*), 8:27-30. *Sood* further describes a subscriber station roaming “to the coverage area of the access point associated with the NAS indicated by [R1KH-ID].” *Id.*, 8:30-32. A person of ordinary skill in the art would have understood that R1KH-ID identifies an access point (authenticator) and its associated network access server. *See* Ex. 1005 (*Sood*), 5:21-25 (“a communication node 300 (e.g., the AP 230 of FIG. 2) may include ... a network access server

(NAS) 330...”), FIG. 3 (showing a NAS as a component of a communication node).

131. Accordingly, it is my opinion that a person of ordinary skill in the art would have understood *Sood* to teach R1KH-ID (“the identity of the authenticator”) is one of (“is one of”) both the access network parameters and the static parameters of claim 1 (one or the other, “said parameters”).

4. Claim 7: “The method of claim 1, wherein said key binding blob is an octet-string that is constructed from static parameters advertised from an authenticator using an authenticator-suppliant protocol.”

132. Claim 7 recites the lexicography of “key binding blob” and is taught by *Sood* for the same reasons I discussed in §X.A.1.c (Element 1[b]), including under *PO’s Potential Alternative Construction*.

5. Claim 8: “The method of claim 1, including a network side authenticator and said suppliant using the channel binding master key for protecting an authenticator-suppliant protocol.”

133. In my opinion, *Sood* teaches this claim, including under *PO’s Potential Alternative Construction*.

134. As I explained in §X.A.1.c (Element 1[b]), *Sood’s* access point 230 (“authenticator”) is network side. Additionally, *Sood* explains that “[t]he subscriber station 220 and the access point 230 mutually derive session keys” and “communicate with each other using session keys.” Ex. 1005 (*Sood*), 9:2-9. A

person of ordinary skill in the art would have understood that communicating “using session keys” involves using the keys to encrypt communications, thus protecting the communications.

135. As I explained in §X.A.1.a (Element 1[pre]), these session keys are used within the IEEE 802.11i protocol (“authenticator-supplicant protocol”). These session keys are derived from PMK-R1 (shown in figure 7) which in turn is derived from PMK-R0 (shown in figure 6), which in turn is derived from MSK (shown in figure 5). Thus, whether the channel binding master key is mapped to PMK-R0 or MSK, it is used by the access point and the subscriber station as the source of the derived session keys that protect the IEEE 802.11i protocol communications.

136. In my opinion, a person of ordinary skill in the art would have understood that nothing in claim 8 requires the access point and the subscriber station to themselves derive the session keys from the channel binding master key. Indeed, the '671 patent explicitly teaches that “the server ... create[s] a CBK used for an authenticator” and “transfers the CBK to the authenticator.” Ex. 1001 (*'671Pat*), 13:64-66, 14:4-6. The '671 patent does not teach any embodiment in which the authenticator itself derives the channel binding key from the channel binding master key.

137. However, to the extent that Patent Owner argues and the Board agrees that claim 8 requires that the access point and the subscriber station themselves derive the session keys from the channel binding master key, it is my opinion that *Sood* teaches this feature. More particularly, *Sood* teaches that “the authentication server 210 may forward the MSK to the access point 230, which in turn, may generate [PMK-R0].” Ex. 1005 (*Sood*), 7:67-8:2. *Sood* also teaches that the subscriber station “generate[s] the MSK” prior to the key derivations in figures 5-7:

The key hierarchy of the key management system 200 ***may begin with a master secret key (MSK)*** or a master authentication key. Turning to FIG. 4, for example, the authentication server 210 and ***the subscriber station 220*** (e.g., via a supplicant) ***may communicate with each other to generate the MSK*** (e.g., a part of an authentication process) (410).

Id., 6:51-56. A person of ordinary skill in the art would have understood that the subsequent key derivations in figures 5-7 that ultimately generate session keys would be performed on the access point and the subscriber station respectively. Thus, whether the channel binding master key is mapped to PMK-R0 or MSK, it is used by the access point and the subscriber station to derive session keys that protect the IEEE 802.11i protocol.

138. Accordingly, it is my opinion that a person of ordinary skill in the art would have understood *Sood* to teach the access point (“a network side

authenticator”) and the subscriber station (“and said supplicant”) using MSK or PMK-R0, either of which may be considered the channel binding master key (“using the channel binding master key”) as the source of session keys used within an IEEE 802.11i protocol (“for protecting an authenticator-supplicant protocol”).

6. Claim 12: “The method of claim 1, further including creating multiple channel binding keys from a single channel binding master key for multiple authenticators.”

139. In my opinion, *Sood* teaches this claim, including under *PO*’s *Potential Alternative Construction*.

140. As I explained in §X.A.1.c (Element 1[b]), either MSK or PMK-R0 (which *Sood* refers to as a first-level authentication key (Ex. 1005 (*Sood*), 6:63-65)) can be mapped to the claimed “channel binding master key.” As I also explained in §X.A.1.c (Element 1[b]), PMK-R1 (which *Sood* refers to as a second-level authentication key (Ex. 1005 (*Sood*), 11:3-6)) can be mapped to the claimed “channel binding key.”

141. *Sood* discloses generating “one or more second-level derived authentication keys (e.g., PMK-R1-1, PMK-R1-2, PMK-R1-3, etc.)” from a single “first-level derived authentication key” (e.g., PMK-R0). Ex. 1005 (*Sood*), 8:3-6. As explained in §X.A.1.c (Element 1[b]), PMK-R0 is in turn derived from a single MSK. Thus, regardless of whether MSK or PMK-R0 is mapped to the claimed “channel binding master key,” a person of ordinary skill in the art would have

understood *Sood* to teach that each of PMK-R1-1, PMK-R1-2, PMK-R1-3, etc. is created from a single channel binding master key. *See* Ex. 1005 (*Sood*), 6:51-59.

142. *Sood* further discloses that “PMK-R1-1 may be associated with the access point 230, PMK-R1-2 may be associated with the access point 240, and the PMK-R1-3 may be associated with the access point 250.” Ex. 1005 (*Sood*), 8:10-13. Thus, PMK-R1-1, PMK-R1-2, and PMK-R1-3 (multiple channel binding keys) are used for access points 230, 240, and 250 (“multiple authenticators”).

143. Accordingly, it is my opinion that a person of ordinary skill in the art would have understood *Sood* to teach deriving PMK-R1-1, PMK-R1-2, and PMK-R1-3 (“creating multiple channel binding keys”) from MSK via PMK-R0, either of which may be considered a channel binding master key (“from a single channel binding master key”) and using PMK-R1-1, PMK-R1-2, and PMK-R1-3 for access points 230, 240, and 250, respectively (“for multiple authenticators”).

7. Claims 14, 16: “The method of claim [1/12], further including creating multiple channel binding keys from a single channel binding master key for multiple authenticators using different key binding blobs for different authenticators.”

144. In my opinion, *Sood* teaches these claims.

145. As I discussed in §X.A.6 (Claim 12), *Sood* discloses “creating multiple channel binding keys from a single channel binding master key for multiple authenticators.”

146. Regarding the further limitation, “using different key binding blobs for different authenticators,” as I explained in §X.A.1.c (Element 1[b]), *Sood* generates each PMK-R1 (“channel binding key”) using the key binding blob “PMK-R0-Name || R1KH-ID || 0x00 || SPA.” This blob includes R1KH-ID, which identifies an associated access point, as I explained in §X.A.3.b (Claim 4). Because the key binding blob contains the identity of the associated access point, a person of ordinary skill in the art would have understood that the key binding blob used in the key derivation is different for each access point (“authenticator”).

147. Accordingly, it is my opinion that a person of ordinary skill in the art would have understood *Sood* to teach using concatenations containing R1KH-ID which represents the identity of the associated authenticator and is different when deriving each of PMK-R1-1, PMK-R1-2, and PMK-R1-3 (“using different key binding blobs”) for each access point 230, 240, and 250 (“for different authenticators”).

B. Ground 2: *Sood* in Combination With *Aboba* Renders Obvious Claims 5 and 18

1. Claim 5: “The method of claim 1, wherein said channel binding master key is at least 64 octets long.”

148. In my opinion, *Sood* in combination with *Aboba* teaches this claim, including under *PO’s Potential Alternative Construction*.

149. A person of ordinary skill in the art would have understood that an octet is a unit of data that consists of exactly 8 bits. Ex. 1013 (*Computer Dictionary*), 5 (defining “octet” as “unit of data that consists of exactly 8 bits...”). Thus, a 64 octet key is $64*8=512$ bits long.

150. As I explained in §X.A.1.c (Element 1[b]), either MSK or PMK-R0 in *Sood* can be mapped to the claimed “channel binding master key.” To the extent that *Sood*’s MSK is mapped to the claimed “channel binding master key,” *Aboba* teaches that the “Master Session Key (MSK)” is “at least 64 octets in length”:

Master Session Key (MSK)

Keying material that is derived between the EAP peer and server and exported by the EAP method. ***The MSK is at least 64 octets in length.***

Ex. 1006 (*Aboba*), 14. A person of ordinary skill in the art would have understood that *Aboba*’s MSK is analogous to the MSK described in *Sood* because both MSKs are “derived between the EAP peer and server” and used to derive lower level keys. Ex. 1006 (*Aboba*), 14, 51; Ex. 1005 (*Sood*), 6:51-59.

151. A person of ordinary skill in the art would have found it obvious to use an at least 64 octet MSK in *Sood*, as taught by *Aboba*, because *Sood* teaches that the MSK is generated “during a mutual authentication process” between a server and a subscriber station, but does not further describe or discuss the length

of the resulting MSK. Ex. 1005 (*Sood*), 10:58-61. A person of ordinary skill in the art would, thus, have been motivated to look for additional details regarding an appropriate length of the MSK to use in the method and would have readily found *Aboba*. *Aboba*, in explaining the length requirement for MSKs, cites to section 7.10 of the EAP specification which explains that 64 octet MSKs “are of sufficient size to enable derivation of a AAA-Key subsequently used to derive Transient Session Keys.” Ex. 1006 (*Aboba*), 16; Ex. 1015 (*EAP*), 51. In my opinion, therefore, a person of ordinary skill in the art would have understood that 64 octets is a sufficient length to perform the similar derivations of PMKs and PTKs taught by *Sood*.

152. Additionally, this modification would have been obvious to a person of ordinary skill in the art because it would have amounted to use of a known technique (*Aboba* uses an at least 64 octet MSK) to improve similar methods (*Sood* uses a similar MSK in its method) in the same way (using an at least 64 octet MSK in *Sood* to, for example, improve security). The modification would also have been obvious to a person of ordinary skill in the art because it would have amounted to applying a known technique (*Aboba* discloses MSKs being at least 64 octets) to a known method (*Sood* has MSKs that are similar to those in *Aboba*) ready for improvement (the length of *Sood*'s MSK could readily be made at least 64 octets)

to yield predictable results (to improve the security of *Sood*'s method by using an at least 64 octet MSK).

153. Accordingly, it is my opinion that a person of ordinary skill in the art would have understood the *Sood-Aboba* combination to teach an MSK (“said channel binding master key”) of at least 64 octets (“is at least 64 octets long”).

154. To the extent that *Sood*'s PMK-R0 is mapped to the claimed “channel binding master key,” a person of ordinary skill in the art would have found it obvious to modify *Sood* to use an at least 64 octet PMK-R0 based on *Aboba*'s teachings.

155. In *Aboba*, the keys derived from the MSK and transported from the server to the authenticator are named “AAA-Key”:

An additional step (phase 1b) is required in deployments which include a backend authentication server, in order to ***transport keying material (known as the AAA-Key) from the backend authentication server to the authenticator.***

Ex. 1006 (*Aboba*), 7. The AAA-key derived “during the initial EAP authentication between the peer and authenticator A” is called AAA-Key-A. *Id.*, 70. This AAA-Key-A is subsequently used to derive other AAA-Keys. *Id.*

156. *Aboba* further teaches that an AAA-Key is 64 octets long, indicating that “AAA-Key-A = MSK(0,63).” Ex. 1006 (*Aboba*), 70. A person of ordinary

skill in the art would have understood that this notation indicates that the AAA-Key-A is the first 64 octets of the MSK.

157. A person of ordinary skill in the art would have understood that *Aboba's* AAA-Key-A key is analogous to *Sood's* PMK-R0 key because, in *Sood*, it is PMK-R0 that is transported from the server to the authenticator and used to derive further keys:

Referring back to FIG. 4, the authentication server 210 may encrypt or wrap the first-level derived authentication key and ***forward the first-level derived authentication key to the access point 230*** (420).

See Ex. 1005 (*Sood*), 7:64-67.

158. A person of ordinary skill in the art would have found it obvious to modify *Sood* to use a 64 octet PMK-R0 because *Sood* discloses that the function used to derive PMK-R0 “may be a 256-bit KDF (KDF-256) ***or other suitable KDFs.***” Ex. 1005 (*Sood*), 7:4-5. Thus, *Sood* expressly teaches using other suitable KDFs to derive PMK-R0, and does not limit such KDFs to any particular length. In other words, *Sood* expressly teaches using KDFs that would produce different length outputs in its methods. A person of ordinary skill in the art would have found it obvious to implement *Aboba's* suggestion to use 64 octet keys to implement PMK-R0 because, as I discuss above, using a 64 octet key would have been sufficient to securely perform subsequent key derivations.

159. A person of ordinary skill in the art would have had a reasonable expectation of success in incorporating *Aboba*'s teachings regarding 64 octet keys in *Sood* because a person of ordinary skill in the art would have been aware of numerous options for KDFs used to generate a 64 octet PMK-R0, including the iterative KDFs disclosed in *Aboba* that are capable of generating variable length outputs:

The EAP key derivation function is taken from the PRF+ key expansion PRF from [IKEv2]. This KDF takes 4 parameters as input: secret, label, application data, and output length. It is only defined for 255 iterations so *it may produce up to 5100 bytes of key material*.

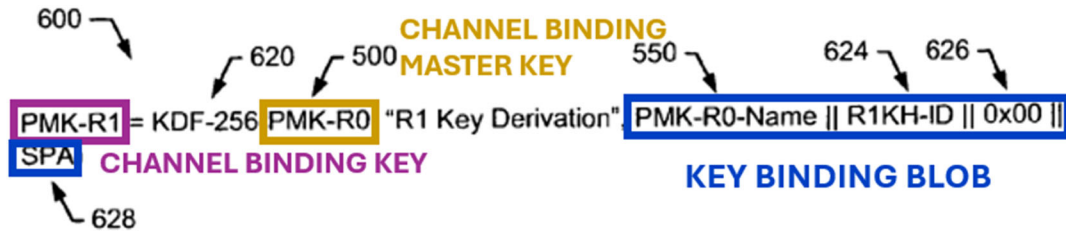
Ex. 1006 (*Aboba*), 71 (brackets in original).

160. Accordingly, it is my opinion that a person of ordinary skill in the art would have understood the *Sood-Aboba* combination to teach a PMK-R0 (“said channel binding master key”) of at least 64 octets (“is at least 64 octets long”).

2. **Claim 18: “The method of claim 1, wherein said key derivation function is computed based on CBK =kdf+(CBMK, KBB), where CBK represents channel binding key, CBMK represents channel binding master key, and KBB represents key binding blob.”**

161. In my opinion, *Sood* in combination with *Aboba* teaches claim 18.

162. *Sood*'s key derivation function, shown in figure 6, takes as input **PMK-R0** (“CBMK”) and **the concatenation of PMK-R0-Name, R1KH-ID, 0x00, and SPA** (“KBB”) and outputs **PMK-R1** (“CBK”) as shown below.



Ex. 1005 (*Sood*), FIG. 6 (excerpted and annotated).

163. The '671 Patent does not explicitly define “kdf+.” However, the specification reproduces an excerpt from RFC 4306 that refers to “prf+” as a “function that outputs a pseudo-random stream”:

We will use the terminology prf+ to describe the function that outputs a pseudo-random stream based on the inputs to a prf as follows: (where | indicates concatenation) ...

Ex. 1001 ('671Pat), 15:7-10. This is possible because prf+ “use[s] the prf iteratively” to repeatedly generate more keying material. *Id.*, 15:5-7. In particular, prf+ generates blocks of keying material T1, T2, T3, etc., and concatenates the blocks together as shown:

$$\text{prf}^+(K,S)=T1|T2|T3|T4| \dots$$

wherein:

$$T1=\text{prf}(K, S |0x01)$$

$$T2=\text{prf}(K, T1|S|0x02)$$

$$T3=\text{prf}(K, T2|S|0x03)$$

$$T4=\text{prf}(K, T3|S|0x04)$$

Ex. 1001 (*'671Pat*), 15:12-18.

164. A person of ordinary skill in the art would have understood that this iterative behavior allows the prf^+ of RFC 4306 to generate a variable length output. In fact, RFC 4306 explains this behavior allows the key derivation to continue “as needed to compute all required keys.” Ex. 1012 (*IKEv2*), 28. This contrasts with other derivation functions that feature fixed-size output, such as the 160-bit prf referred to in RFC 4306 and the 384 and 512-bit prfs referred to in the 802.11i standard. *Id.*; Ex. 1009 (*802.11i*), 90 (referring to “PRF-384” and “PRF-512”). Thus, a person of ordinary skill in the art would have understood the notation kdf^+ to refer to a key derivation function that generates a variable length output.

165. *Aboba* discloses using a key derivation function with a variable length output to derive channel binding keys. For example, *Aboba* discloses a derivation of AAA-Key-B that, unlike the base AAA-Key-A, is bound to the identity of an access point. *See* Ex. 1006 (*Aboba*), 70. In particular, AAA-Key-B is derived as

follows: “AAA-Key-B = PRF(EMSK(0,63), ‘EAP AAA-Key derivation for multiple attachments’, AAA-Key-A, B-Called-Station-Id, Calling-Station-Id, *length*).” *Id.* The length variable indicates that the PRF function used in *Aboba* has a variable length output. Additionally, a person of ordinary skill in the art would have understood that this key derivation binds AAA-Key-B to “B-Called-Station-Id.” *Aboba* indicates that B-Called-Station-Id is “AP B MAC address,” i.e., an identifier for the access point. *Id.* Because AAA-Key-B is bound to the identity of the access point, a person of ordinary skill in the art would have understood that this key is analogous to *Sood*’s PMK-R1 (“channel binding key”).

166. A person of ordinary skill in the art would have found it obvious to use *Aboba*’s key derivation function that has a variable length output to generate PMK-R1 (“channel binding key”) because *Sood* teaches that the KDF used to derive PMK-R1 “may be a 256-bit KDF (KDF-256) *or other suitable KDFs*.” Ex. 1005 (*Sood*), 8:26-27. A person of ordinary skill in the art would have understood *Aboba*’s function to be suitable because *Aboba* teaches using it to derive the key AAA-Key-B, which is analogous to *Sood*’s PMK-R1. A person of ordinary skill in the art would have additionally recognized the suitability of *Aboba*’s key derivation function because, just like *Sood*’s KDF-256 function, it takes a key, a label, and data as input and derives a new key. *Compare* Ex. 1006 (*Aboba*), 71 with Ex. 1005 (*Sood*), Fig. 6. A person of ordinary skill in the art would have

additionally been motivated to use *Aboba*'s iterative KDF function in *Sood* because it would have provided the added flexibility to generate a variable amount of keying material.

167. A person of ordinary skill in the art would have additionally found the combination obvious because it would have amounted to a simple substitution of one known element (*Aboba*'s variable length output key derivation function) for another (*Sood*'s KDF-256 function) to obtain predictable results (*Sood-Aboba* having a variable length key derivation function capable of generating any desired quantity of keying material).

168. Accordingly, it is my opinion that a person of ordinary skill in the art would have understood the *Sood-Aboba* combination to teach a key derivation procedure that outputs **PMK-R1** ("CBK="), utilizes *Aboba*'s variable length key derivation function ("kdf+"), and takes as input **PMK-R0** and **the concatenation of PMK-R0-Name, R1KH-ID, 0x00, and SPA** ("(CBMK, KBB), where CBK represents channel binding key, CBMK represents channel binding master key, and KBB represents key binding blob").

C. Ground 3: *Sood* in Combination With *Lee* Renders Obvious Claims 6, 10, 11, 13, 15, 17, and 19

1. Motivation to Combine *Sood* and *Lee*

169. In my opinion, a person of ordinary skill in the art would have been motivated to combine *Sood* and *Lee*.

170. *Sood* teaches deriving and distributing authentication keys to access points wherein each key is bound to the identity of the access point:

Based on the first-level derived authentication key, the access point 230 may generate one or more second-level derived authentication keys (e.g., PMK-R1-1, PMK-R1-2, PMK-R1-3, etc.). ***Each of the second-level derived authentication key may be associated with an access point*** that may provide communication services to the subscriber station 220 when the subscriber station 220 may roam into a corresponding coverage area. In one example, PMK-R1-1 may be associated with the access point 230, PMK-R1-2 may be associated with the access point 240, and the PMK-R1-3 may be associated with the access point 250. Initially, the access point 230 may generate PMK-R1-1 to establish full authentication with the subscriber station 220. The access point 230 (e.g., via the encryptor 355) may encrypt or securely wrap PMK-R1-2 and PMK-R1-3. Accordingly, ***the access point 230 may forward the encrypted and/or wrapped PMK-R1-2 and PMK-R1-3 to controllers of the access points 240 and 250*** (430 and 440, respectively).

Ex. 1005 (*Sood*), 8:3-20.

171. In particular, *Sood* discloses deriving “second-level derived authentication keys” that are bound to R1KH-ID, the identifier of an access point which will hold the corresponding second-level key:

In the example of FIG. 6, for example, the derivation of PMK-R1 600 may be based on a key derivation function (KDF) 620, the PMK-R0 500 of FIG. 5, and concatenations of information elements in the PMK-R0-Name 550 of FIG. 5, *an NAS identifier field 624*, a separator field 626, and a SPA field 628. For example, the KDF 620 may be a 256-bit KDF (KDF-256) or other suitable KDFs. *The NAS identifier field 624 (e.g., R1KH-ID) may include a value to identify the network entity holding the second-level derived authentication key (e.g., a key holder of R1)*. In particular, the subscriber station 220 may roam to the coverage area of *the access point associated with the NAS indicated by the NAS identifier field 624*.

Id., 8:21-32. *Sood*'s access points, which meet the '671 patent's definition of an authenticator as I explained in §X.A.1.c (Element 1[b]), are EAP-capable. *Sood*, 6:56-62 (“the PMK ... derive[s] from an authentication process such as Extensible Authentication Protocol-Transport Layer (EAP-TLS)”), 7:67-8:6 (teaching that the access point derives PMKs).

172. *Lee* teaches a key derivation method where, as in *Sood*, authentication keys are generated for multiple access points. Ex. 1007 (*Lee*) ¶[0089] (“as many PMKs, PMKnext as the number of the neighbor APs are generated”). Also as in

Sood, Lee's keys are distributed to the access points and each bound to the identity of the access point using the key derivation: "PMK_{next} = PRF(RK, PMK_{curr}, STAMac, nextAPmac)":

Meanwhile, *AP_A generates a PMK for a neighbor AP*, AP_B, managed by its AP-neighborhood graph. Let the PMK for the neighbor AP be PMK_{next}. PMK_{next} is generated using the RK, the current PMK, PMK_{curr}, the MAC address of the STA, STAMac, and ***the MAC address of a new-AP, nextAPmac*** by a PRF (Pseudo-Random Function), expressed as $PMK_{next} = PRF(RK, PMK_{curr}, STAMac, nextAPmac)$ (1)

Id., ¶[0081]. A person of ordinary skill in the art would have understood that this key derivation binds the PMK for an access point (here labeled PMK_{next}) to the identity of that access point (here labeled nextAPmac).

173. *Lee* additionally discloses ***the server*** deriving each access point's PMK proactively to provide fast roaming. Ex. 1007 (*Lee*) ¶¶[0102] ("the servers generate PMKs for APs neighboring to a particular AP and transmits them to the neighbor APs."), [0104] ("fast roaming through proactive key distribution ..."). *Lee* explains that prior systems required access points to have "large-capacity" memory to store the numerous roaming keys required for numerous users:

First, APs each preserve all necessary proactive keys for roaming. Each AP reserves memory space for the roaming service, stores all proactive keys needed for the roaming service in the memory, and

retrieves one proactive key from the memory when necessary. *A distinctive shortcoming of this scheme is the requirement of a large-capacity memory.*

Lee, ¶[0060]. By outsourcing key generation to the server, the server is able to “manage the AP-neighborhood graph for each AP” so that only necessary keys (i.e., keys for access points that neighbor the connected access point) are generated and distributed:

The third scheme is that a higher-layer server (accounting server) manages neighbor APs for each AP and provides the neighbor APs with proactive keys necessary for roaming when an STA accesses the AP. To implement this scheme, the higher-layer server is provided to manage the AP-neighborhood graph for each AP. The higher-layer server may be an existing AAA server or a separately procured server. Depending on the amount of information regarding the managed AP-neighborhood graphs, a plurality of higher-layer servers can be used.

Id., ¶[0062]; *see also* ¶[0040].

174. It would have been obvious to a person of ordinary skill in the art to incorporate *Lee*'s teachings into *Sood* such that *Sood*'s PMK-R1 keys (i.e., the keys bound to particular access points), are centrally derived by the server and transferred to the respective access points. A person of ordinary skill in the art would have found it obvious to implement *Lee*'s teachings in *Sood* to gain the

benefit of this centralized key management and the resulting reduced memory requirements for access points.

175. The combination would also have been obvious to a person of ordinary skill in the art because it would have amounted to applying a known technique (*Lee*'s central key derivation) to a known device (*Sood*'s key management system) ready for improvement to yield predictable results (the *Sood-Lee* combination having access points with lower memory requirements).

176. The combination would further have been obvious to a person of ordinary skill in the art because it would have amounted to applying a known technique (*Lee*'s central key derivation) to similar devices (*Sood*'s key derivations procedures and access points which are similar to *Lee*'s) in the same way (by implementing central key derivation functionality on *Sood-Lee*'s server).

177. A person of ordinary skill in the art would also have had a reasonable expectation of success in implementing this combination because *Sood*'s system already contains an "authentication server 210" which is "an authentication, authorization, and accounting (AAA) server." Ex. 1005 (*Sood*), 4:32-35. *Lee* discloses that the "higher-layer server" which performs the key derivations is "an existing AAA server":

To implement this scheme, the higher-layer server is provided to manage the AP-neighborhood graph for each AP. The higher-layer server *may be an existing AAA server* or a separately procured server.

Ex. 1007 (*Lee*) ¶[0062]. Thus, implementing *Lee*'s central key derivation in *Sood* would have amounted to adding additional known functionality to *Sood*'s existing server and access points which would have been well within the capabilities of a person of ordinary skill in the art.

178. Additionally, despite *Lee*'s teaching that PMKs for an access point are proactively derived at the server, *Lee* further teaches that its access points retain the capability to perform a “conventional roaming process”:

The third method is that the AP-neighborhood graph is automatically generated for each AP and automatically updated each time the AP layout is changed. ***According to this method, however, roaming is carried out by the conventional roaming process until the AP-neighborhood graph is generated.*** In other words; a procedure for checking connections to, each AP is needed. For example, if an STA associated to AP_A attempts to initially roam to AP_B that the STA has never moved to, AP_B performs an IAPP procedure to receive a context corresponding to the STA from AP_A. AP_A and AP_B then confirm that there is a connection between them for roaming and thus can update their AP-neighborhood graphs. After the updating, the STA can roam from AP_A to AP_B or vice versa without the IAPP procedure.

Ex. 1007 (*Lee*) ¶[0066]. This “conventional roaming process” is disclosed to be “a re-authentication procedure performed by an *EAP-TLS protocol*.” *Id.*, ¶[0021].

The process is illustrated in figure 4:

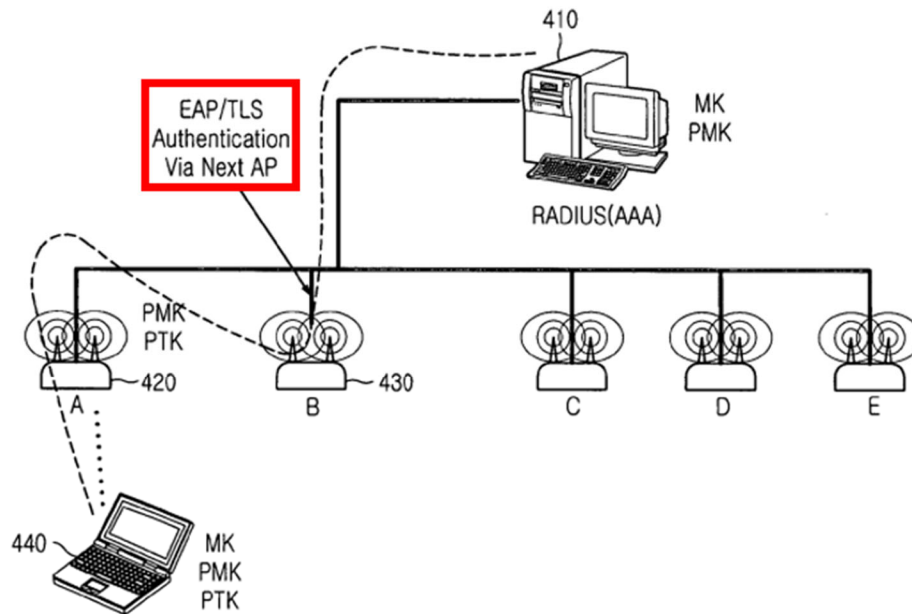


FIG.4
(PRIOR ART)

Ex. 1007 (*Lee*), FIG. 4 (annotated).

Thus, a person of ordinary skill in the art would have understood that *Lee*'s teachings can be implemented in access points that retain the ability to perform EAP authentication in the absence of a proactively distributed key.

179. A person of ordinary skill in the art would have been motivated to use EAP-capable access points as disclosed in *Lee* because EAP-capable access points can “automatically generate[]” the “AP-neighborhood graph” subsequently used by

the higher-layer server. Ex. 1007 (*Lee*) ¶[0066]. *Lee* explains that, when a device roams from one access point to another conventionally, the two access points “confirm that there is a connection between them for roaming and thus can update their AP-neighborhood graphs.” *Id.* A person of ordinary skill in the art would have been motivated to use EAP-capable access points to automatically generate the AP-neighborhood graph to reduce the network configuration burden of enabling server-managed key derivation and distribution.

180. Incorporating *Lee*’s teachings regarding access points automatically generating the AP-neighborhood graph into *Sood* would also have been obvious to a person of ordinary skill in the art because it would have amounted to applying a known technique (*Lee*’s access points automatically generate the AP-neighborhood graph) to a known device (*Sood*’s access points) ready for improvement to yield predictable results (the *Sood-Lee* combination access points that automatically generate the AP-neighborhood graph).

181. Additionally, a person of ordinary skill in the art would have had a reasonable expectation of success in incorporating *Lee*’s teachings into *Sood* because, as noted above, *Sood*’s authenticators are already disclosed to be EAP-capable.

2. **Claim 6**

- a. **6[pre]: “A channel binding method based on parameter binding in a key derivation procedure for authentication of a mobile supplicant to an access network, comprising:”**

182. To the extent the preamble is limiting, *Sood* teaches this feature for the reasons I explained in §X.A.1.a (Element 1[pre]).

- b. **6[a]: “using an extensible authentication protocol server to cryptographically bind access network parameters to a channel binding key and to transmit said channel binding key to an extensible authentication protocol authenticator for use by the extensible authentication protocol authenticator as an extensible authentication protocol master session key without said extensible authentication protocol authenticator needing to carry said access network parameters in extensible authentication protocol authentication methods;”**

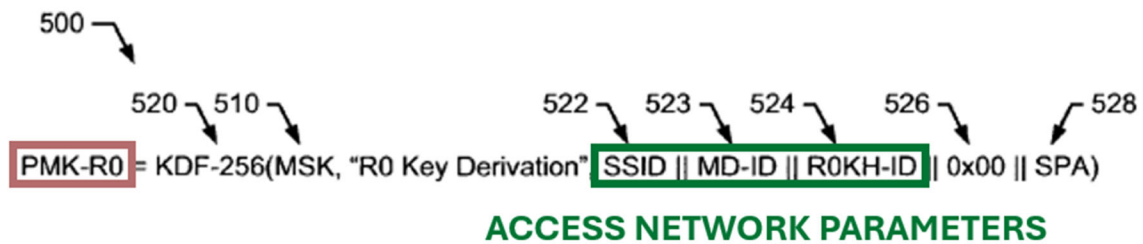
183. In my opinion, *Sood* teaches this feature.

- (i) **“using an extensible authentication protocol server to cryptographically bind access network parameters to a channel binding key and to transmit said channel binding key to an extensible authentication protocol authenticator ...”**

184. For the reasons I explained in §X.C.1 (Ground 3, Motivation to Combine), in the *Sood-Lee* combination the server generates PMK-R1 (“channel binding key”). That server is an “extensible authentication protocol server” because *Sood* discloses that its server implements “an authentication process such

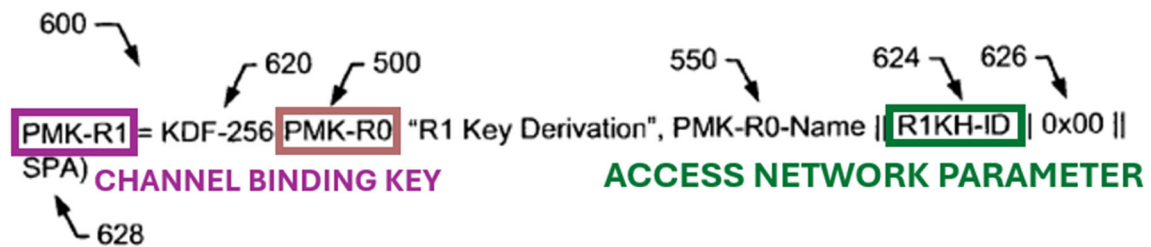
as Extensible Authentication Protocol-Transport Layer (EAP-TLS).” Ex. 1005 (*Sood*), 6:59-65.

185. The parameters **SSID**, **MD-ID**, **R0KH-ID**, and **R1KH-ID** (“access network parameters”) are bound to **PMK-R1** (“channel binding key”) as follows. First, **SSID**, **MD-ID**, and **R0KH-ID** are bound to **PMK-R0** because each of these parameters are components of the concatenation of figure 5:



Ex. 1005 (*Sood*), FIG. 5 (excerpted and annotated).

186. Second, **PMK-R1** is derived from **PMK-R0** (as shown in figure 6), thereby inheriting the above bindings. In the same derivation of figure 6, **R1KH-ID** is also bound to **PMK-R1**:



Ex. 1005 (*Sood*), FIG. 6 (excerpted and annotated).

187. As I discussed in §X.C.1 (Ground 3, Motivation to Combine), in the *Sood-Lee* combination the server derives **PMK-R1** (“channel binding key”) and

transmits it to an access point (“extensible authentication protocol authenticator”). The access point is an authenticator for the reasons I explained in §X.A.1.c (Element 1[b]). The access points of *Sood-Lee* are also “extensible authentication protocol authenticators” because, as I explained in §X.C.1 (Ground 3, Motivation to Combine), the *Sood-Lee* combination uses EAP-capable access points.

(ii) “... for use by the extensible authentication protocol authenticator as an extensible authentication protocol master session key ...”

188. *Sood* discloses that the access point (“extensible authentication protocol authenticator”) uses PMK-R1 (“channel binding key”) to generate a session key named PTK. Ex. 1005 (*Sood*), 9:36-42 (“the derivation of PTK 700 may be based on ... PMK-R1 600 of FIG. 6”); *see also* 7:51-55 (“a session key (e.g., PTK1)”). A person of ordinary skill in the art would have understood that using PMK-R1 to generate session keys is using it “as an extensible authentication protocol master session key” because this disclosure is consistent with the EAP specification. In particular, the EAP specification discloses that “[t]he MSK is used only for further key derivation, not directly for protection of the EAP conversation or subsequent data.” Ex. 1015 (*EAP*), 45. PMK-R1 is likewise used to generate PTK, which directly protects subsequent data. Ex. 1005 (*Sood*), 9:17-22 (“In particular, the subscriber station 220 and the access point 240 may mutually generate session keys for the session based on a

corresponding second-level derived authentication key (e.g., PMK-R1-2).

Accordingly, the subscriber station 220 and the access point 240 may communicate with each other using session keys (460).”).

189. The disclosure of the '671 Patent is consistent with the EAP specification. The 671 patent explains that peers and authenticators “use *a key* generated and exported by an EAP method to *bootstrap ciphersuites used for protecting their access network*” and “[t]his key is referred to as *MSK (Master Session Key)*.” Ex. 1001 (*'671Pat*), 12:43-50.

190. A person of ordinary skill in the art would have understood the term “bootstrap ciphersuits” refers to generating additional keys, not directly protecting data. In the computing field, the term “bootstrap” originally referred to executing simple software that loads and starts a computer’s more complicated operating system. Ex. 1013 (*Computer Dictionary*), 3. Thus, the computer is said to “pull itself up by its own bootstraps.” *Id.* In the specific context of network ciphersuites, the term “bootstrap” takes on a different but related meaning of using an internal process to use existing ciphering data (e.g., a single key) to create new more capable or voluminous ciphering data, e.g., session keys. *See* Ex. 2020 (*Taesombut*), 2, 7 (describing a method of “secure bootstrap registration” in a wireless network comprising using a master key to generate session keys for use in an encryption algorithm). In both contexts, a system uses a relatively small

program or data item as a starting point to enable the system to execute complex tasks.

(iii) “... without said extensible authentication protocol authenticator needing to carry said access network parameters in extensible authentication protocol authentication methods;”

191. In the *Sood-Lee* combination there would be no need to carry SSID, MD-ID, R0KH-ID, and R1KH-ID (access network parameters) in extensible authentication protocol methods because, as I explained in §X.A.1.c (Element 1[b]), each of these parameters are advertised from an authenticator. Advertised parameters need not be carried in authentication methods for the reasons I explained in §X.A.1.b (Element 1[a]).

* * *

192. Accordingly, it is my opinion that a person of ordinary skill in the art would have understood the *Sood-Lee* combination to teach a server used (“using an extensible authentication protocol server”) to cryptographically bind (“to cryptographically bind”) SSID, MD-ID, R0KH-ID, and R1KH-ID (“access network parameters”) to PMK-R1 (“to a channel binding key”) and to transmit PMK-R1 to an access point (“and to transmit said channel binding key to an extensible authentication protocol authenticator”) to generate session keys (“for use by the extensible authentication protocol authenticator as an extensible

authentication protocol master session key”) while the access point advertises the parameters to the subscriber station (“without said extensible authentication protocol authenticator needing to carry said access network parameters in extensible authentication protocol authentication methods”).

c. 6[b]: “including using said extensible authentication protocol server to derive said channel binding key from a channel binding master key bound to a key binding blob using a key derivation function;”

193. In my opinion, *Sood* in combination with *Lee* teaches this feature.

194. As I explained in §X.C.2.b (Element 6[a]), the server is used to derive the channel binding key. As I also explained in §X.A.1.c (Element 1[b]), the channel binding key is derived “from a channel binding master key bound to a key binding blob using a key derivation function” regardless of whether the key binding blob must be bound to the channel binding key or the channel binding master key (i.e., whether Petitioner’s proposed construction or *PO’s Potential Alternative Construction* applies).

d. 6[c]: “wherein said key binding blob is a string that is constructed from static parameters advertised from said extensible authentication protocol authenticator.”

195. In my opinion, *Sood* in combination with *Lee* teaches this feature.

196. This limitation mostly recites the lexicography of “key binding blob” and is disclosed by *Sood* for the same reasons I discussed in §X.A.1.c (Element

1[b]). Additionally, the access points of the *Sood-Lee* combination are “extensible authentication protocol authenticator[s]” for the reasons I explained in §X.C.2.b (Element 6[a]).

3. Claim 10

197. In my opinion, *Sood* in combination with *Lee* teaches this claim, including under *PO’s Potential Alternative Construction*.

a. 10[a]: “The method of claim 1, further including that: a) a server and the supplicant create a channel binding key used for an authenticator;”

198. In my opinion, *Sood* in combination with *Lee* teaches this feature.

199. For the reasons I explained in §X.C.2.b (Element 6[a]), the *Sood-Lee* combination teaches the server creating PMK-R1 (channel binding key).

200. *Sood* also discloses that the subscriber station “via a supplicant” generates an MSK and later “generate[s] a session key.” Ex. 1005 (*Sood*), 6:52-56, 7:49-56. Session keys (e.g., PTK) are generated from PMK-R1. Ex. 1005 (*Sood*), 9:36-42. Thus, a person of ordinary skill in the art would have understood that the subscriber station would generate PMK-R1 using its MSK as an intermediate step to generating a session key.

201. To the extent that Patent Owner argues and the Board agrees that *Sood* alone does not teach this limitation, it is my opinion that it would have been obvious based on the combination of *Sood* and *Lee*.

202. More particularly, *Lee* explicitly discloses that the station generates the corresponding PMK of an access point to connect to the access point:

For example, AP_A or AP_B provides the STA with PMKnext. ***Or the STA can directly generate PMKnext from next APmac received from AP_A or AP_B.***

Ex. 1007 (*Lee*) ¶[0083].

203. The generated PMK is “used for an authenticator” because the PMK is transmitted to an authenticator (e.g., access point) to enable the subscriber station to authenticate with that authenticator:

Higher-layer servers manage neighbor APs for individual APs by their AP-neighborhood graphs. When necessary, ***the servers generate PMKs for APs neighboring to a particular AP and transmits them to the neighbor APs.*** Thus when an STA roams to one of the neighbor APs, a security system operates using the already known PMK, thereby providing fast roaming.

Ex. 1007 (*Lee*) ¶[0102].

204. A person of ordinary skill in the art would have found it obvious to have *Sood*'s subscriber station directly generate PMK-R1 as taught by *Lee* because generating PMK-R1 locally (i.e., without needing to transmit PMK-R1 over the wireless network), would have improved security by foreclosing the possibility that the PMK-R1 could be intercepted in transit by a malicious actor/device. A person of

ordinary skill in the art would have also found the combination obvious because it amounted to applying a known technique (*Lee* generating PMKs at the subscriber station) to a known method ready for improvement (*Sood*'s key management system) to yield predictable results (*Sood-Lee* populating the subscriber station with necessary keys without transmitting the keys over the air).

205. Accordingly, it is my opinion that a person of ordinary skill in the art would have understood the *Sood-Lee* combination to teach the server and the subscriber station (“a server and the supplicant”) creating, and the server transmitting, PMK-R1 (“create a channel binding key”) to an authenticator to enable authentication to that authenticator (“used for an authenticator”).

b. 10[b]: “b) the server transfers the channel binding key to the authenticator; and”

206. In my opinion, *Sood* in combination with *Lee* teaches this feature for the reasons I discussed in §X.C.2.b (Element 6[a]).

c. 10[c]: “c) the supplicant and the authenticator verify proof of possession of the channel binding key over an authenticator-supplicant protocol.”

207. In my opinion, *Sood* in combination with *Lee* teaches this feature.

208. *Sood* discloses that its subscriber station (“supplicant”) and access point (“authenticator”) “mutually derive session keys for the session based on a corresponding second-level derived authentication key (e.g., PMK-R1-1)” and

“communicate with each other using session keys (450).” Ex. 1005 (*Sood*), 8:65-9:9. These session keys are, for example “a pairwise temporal key.” Ex. 1005 (*Sood*), 9:5-7. A person of ordinary skill in the art would have understood that using pairwise keys for communication requires the keys to be identical. See Ex. 1009 (802.11i), 19 (defining “pairwise” to refer to “keys that are shared only between the two entities in a pairwise association.”). Additionally, deriving identical pairwise keys requires using the same base PMK-R1, as confirmed by *Sood*’s disclosure that the PMK-R1s are “corresponding.” Ex. 1005 (*Sood*), 9:2-5. Thus, using pairwise keys for communication verifies that both parties have possession of the relevant PMK-R1 (channel binding key). This communication may be conducted over a 802.11i WLAN network (authenticator-supplicant protocol) as explained in §X.A.1.a (Element 1[pre]).

209. Accordingly, it is my opinion that a person of ordinary skill in the art would have understood the *Sood-Lee* combination to teach the subscriber station and access point (“the supplicant and the authenticator”) communicate using pairwise keys (“verify proof of possession of the channel binding key”) via an 802.11i WLAN network (“over an authenticator-supplicant protocol”).

4. Claim 11: “The method of claim 10, further including that the channel binding key is derived from a channel binding master key bound to a key binding blob associated with the authenticator using a key derivation function.”

210. In my opinion, *Sood* in combination with *Lee* teaches this claim, including under *PO’s Potential Alternative Construction*.

211. The only new limitation in claim 11, as compared to claim 1, is the requirement that the key binding blob be “associated with the authenticator.” As I explained in §X.A.1.c (Element 1[b]), the key binding blob is “PMK-R0-Name || R1KH-ID || 0x00 || SPA.” This key binding blob is associated with the authenticator because it contains “R1KH-ID” which identifies the associated access point (authenticator). Ex. 1005 (*Sood*), 8:27-32.

212. Accordingly, it is my opinion that a person of ordinary skill in the art would have understood the *Sood-Lee* combination to teach a key binding blob containing the identity of the associated access point (“associated with the authenticator”).

213. As I also explained in §X.A.1.c (Element 1[b]), if the Patent Owner argues and the Board agrees that the claim requires that the key binding blob be bound to the channel binding master key, the key binding blob is “SSID || MD-ID || R0KH-ID || 0x00 || SPA.” This key binding blob is also associated with the authenticator because it contains “SSID” and “MD-ID.” A person of ordinary skill

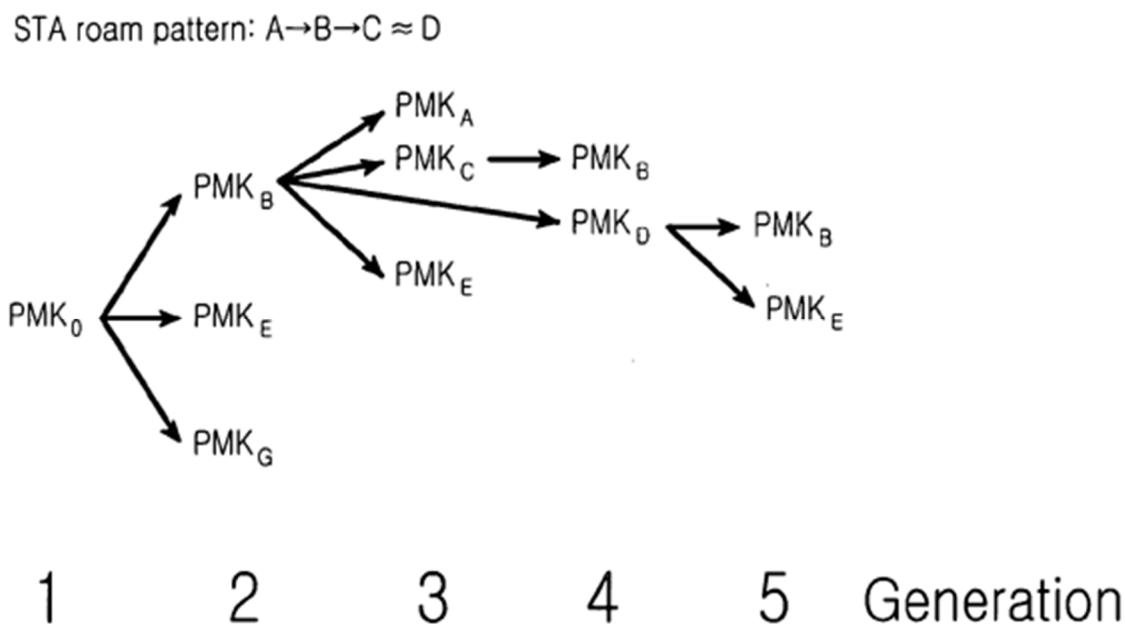
would understand that the SSID identifies the service with which the access point is associated and the MD-ID identifies the mobility domain with which the access point is associated. *See Sood* 7:5-10.

214. Accordingly, it is my opinion that a person of ordinary skill in the art would have understood the *Sood-Lee* combination to teach a key binding blob containing the identity of the service set and the mobility domain of the associated access point (“associated with the authenticator”).

5. Claims 13, 15, 17: “The method of claim [12/14/16], further including using channel binding keys to form a hierarchical channel binding.”

215. In my opinion, *Sood* in combination with *Lee* teaches these claims. With respect to claim 13, this would be true even under *PO’s Potential Alternative Construction*.

216. *Lee* discloses a hierarchy between different channel binding keys, as shown in figure 11:



Ex. 1007 (*Lee*), FIG. 11.

A person of ordinary skill in the art would have understood *Lee*'s figure 11 to illustrate that, for example, the PMK associated with Access Point C may be derived from the PMK associated with Access Point B. Ex. 1007 (*Lee*) ¶[0116] (“As the STA roams to AP_B, *the AS generates PMKC for AP_C from PMKB* in preparation for roaming of the STA to AP_C in a third generation stage.”). A person of ordinary skill in the art would have understood this disclosure to describe using channel binding keys to form a hierarchical channel binding.

217. A person of ordinary skill in the art would have found it obvious to implement *Lee*'s hierarchical key derivations in *Sood* because *Sood* teaches that “the methods and apparatus disclosed herein may include additional levels of

authentication keys.” Ex. 1005 (*Sood*), 11:50-53. In other words, *Sood* itself suggests the combination.

218. Additionally, a person of ordinary skill in the art would recognize that *Lee*, by deriving keys hierarchically, causes the channel binding key to be bound to the particular path the subscriber station takes through the network. A person of ordinary skill in the art would have been motivated to bind the channel binding key to the subscriber station’s path because this binding increases the security of the network by preventing an attacker from reusing old keys.

219. The modification would also have been obvious to a person of ordinary skill in the art because it would have amounted to applying a known technique (*Lee* discloses binding an authentication key to a subscriber station’s path through the network) to a known method (*Sood*’s method for generating PMK-R1s) ready for improvement (*Sood*’s PMK-R1 could readily be bound to a subscriber station’s path through the network) to yield predictable results (to improve the security of *Sood*’s method by preventing an attacker from reusing old keys).

220. Accordingly, it is my opinion that a person of ordinary skill in the art would have understood the *Sood-Lee* combination to teach generating a PMK-R1 for an access point from a previous PMK-R1 generated for a different access point (“using channel binding keys to form a hierarchical channel binding”).

6. Claim 19: “The method of claim 3, wherein said authenticator is an EAP authenticator, and wherein the EAP authenticator receives and processes the channel binding key as a Master Session Key (MSK).”

221. In my opinion, *Sood* in combination with *Lee* teaches this claim, including under *PO’s Potential Alternative Construction*, for the reasons I explained in §X.C.2.b (Element 6[a]).

D. Ground 4: *Sood* in Combination With *Aboba* and *Lee* Renders Obvious Claims 1-8 and 10-19

222. In my opinion, the limitations of claims 1-8 and 10-19 are obvious for the reasons I explained in Grounds 1-3. However, to the extent that Patent Owner argues and the Board agrees that one or both of the limitations discussed below is not taught for the reasons I explained in Ground 1, these limitations would have been obvious in view of the combination of *Sood* and *Aboba* or *Sood* and *Lee*, for the reasons I discuss below in §X.D.1 and §X.D.2, respectively. Accordingly, it is my opinion that a person of ordinary skill in the art would have found claims 1-8 and 10-19 obvious based on the combination of *Sood*, *Aboba* and *Lee* for the reasons discussed above and the additional reasons discussed in this Ground 4.

1. “using an authenticator-suppliant protocol”

223. All challenged claims recite “a key binding blob” either directly or via dependency on claim 1. The ’671 patent defines a “Key Binding Blob” as “[a]n octet-string that is constructed from static parameters advertised from an

authenticator *using an Authenticator-Supplicant Protocol (ASP)*.” Ex. 1001 (*'671Pat*), 13:27-30. As I explained in §X.A.1.c (Element 1[b]), *Sood* teaches this aspect of the lexicography of “key binding blob.” However, to the extent that Patent Owner argues and the Board agrees that *Sood* does not teach advertising parameters from an authenticator “using an authenticator-suppliant protocol,” it is my opinion that this feature would have been obvious based on *Sood* in combination with *Aboba*.

224. *Aboba* discloses that the IEEE 802.11 protocol advertises parameters, including NAS identifiers:

For the purpose of identifying the authenticator, the contents of the NAS-Identifier attribute is recommended. In order to ensure that all parties can agree on the authenticator name this requires the authenticator to advertise its name (*typically using a lower layer mechanism, such as the 802.11 Beacon/Probe Response*).

Ex. 1006 (*Aboba*), 25. *Aboba* confirms that its disclosures are compatible specifically with 802.11i:

One of the goals of EAP is to allow EAP methods to function on *any lower layer meeting the criteria outlined in [RFC3748]*, Section 3.1. For example, as described in [RFC3748], EAP authentication can be run over PPP [RFC1661], IEEE 802 wired networks [IEEE8021X], *and IEEE 802.11 wireless LANs [IEEE80211i]*

Id., 11 (brackets in original). A person of ordinary skill in the art would have understood that a “beacon” message is a message advertised by the authenticator.

225. A person of ordinary skill in the art would have found it obvious to advertise the SSID, MD-ID, and R1KH-ID (“static parameters”) discussed in *Sood* using the IEEE 802.11i protocol’s beacon functionality as expressly taught by *Aboba* because *Sood* and *Aboba* are directed to similar methods for generating keys bound to static parameters. *Sood* discloses generating second-level keys bound to a R1KH-ID value that identifies the authenticator that will hold the second-level key. Ex. 1005 (*Sood*), 8:3-6, 8:21-30. *Aboba* similarly discloses generating AAA-Keys bound to a Called-Station-Id that identifies the authenticator which the AAA-Key will be transferred to. Ex. 1006 (*Aboba*), 70.

226. Moreover, *Sood* teaches that its disclosures “may be applied to WPANs, WLANs, WMANs, and/or WWANs” and that WLAN is implemented “in accordance with the 802.11 family of standards.” Ex. 1005 (*Sood*), 11:56-58, 2:37-44. Thus, there is an explicit teaching in the prior art that would have lead a person of ordinary skill in the art to implement *Sood* utilizing the features of 802.11i, as expressly taught in *Aboba*.

227. Furthermore, *Sood* already teaches that at least MD-ID and R1KH-ID are advertised, while it would have been well-known to a person of ordinary skill in the art that SSIDs would be advertised. *Sood* 7:37-40, 8:27-36; Ex. 1014

(802.11-1997), 21 (showing SSID field in the “Beacon frame body”). Advertising these parameters using IEEE 802.11i, as taught by *Aboba*, would have been obvious to a person of ordinary skill in the art because it would have amounted to using a known technique (*Aboba* teaches advertising parameters using IEEE 802.11i) to improve a similar method (*Sood* teaches to advertise certain parameters) in the same way (to make necessary parameters available to supplicants prior to authentication).

2. “without needing to carry the parameters in authentication methods” / “without ... needing to carry said access network parameters in ... authentication methods”

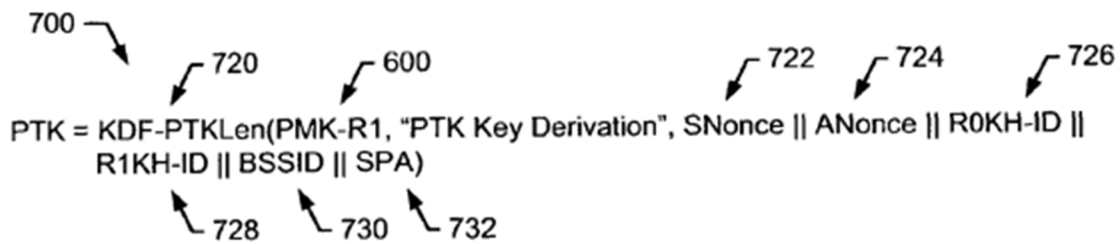
228. All challenged claims recite “without needing to carry the parameters in authentication methods” (claim 1 and its dependents) or “without ... needing to carry said access network parameters in ... authentication methods” (claim 6). As I explained in §X.A.1.b (Element 1[a]), *Sood* teaches these limitations.

229. However, to the extent that Patent Owner argues and the Board agrees that *Sood* alone does not teach this feature, this feature would have been obvious based on the combination of *Sood* and *Lee*. This would be true even though in the *Sood-Lee* combination the derivation of second-level keys is performed by the server and not the access point.

230. In the *Sood-Lee* combination, there would still have been no need to carry at least SSID, MD-ID, R0KH-ID and R1KH-ID (“access network

parameters”) in authentication methods because, as I explained in §X.C.2.b(i) (Element 6[a] Part i), the server would have already bound these values to PMK-R1.

231. For the reasons I explained in §X.C.1 (Ground 3, Motivation to Combine *Sood* and *Lee*), these bindings would also have been performed by the server in the *Sood-Lee* combination. Because the access network parameters are bound to PMK-R1, they will also be bound to PTK (“a key”) when PTK is derived from PMK-R1 in figure 7:



Ex. 1005 (*Sood*), FIG. 7.

232. Because in the *Sood-Lee* combination the access network parameters are bound to PMK-R1 by the server, there is no need to separately carry the access network parameters in authentication methods. A person of ordinary skill in the art would have understood that, with the access network parameters bound to PMK-R1 by the server, there is no risk that a rogue authenticator could advertise incorrect access network parameters to the subscriber station without detection and, thus, no need to verify the access network parameters by carrying them in

authentication methods. If the authenticator were to advertise incorrect access network parameters, the subscriber station's different parameters would not allow it to generate the corresponding PTK for communication with the access point. *See* Ex. 1005 (*Sood*), 7:52-55 (indicating that the subscriber station generates the PTK).

233. The '671 Patent also describes a similar technique for obviating the need to carry access network parameters in authentication methods. Parameters may need to be carried in authentication methods in order to verify lower layer parameters for consistency between the EAP peer and the server. Ex. 1001 (*'671Pat*), 5:5-6. The '671 Patent describes "an alternative Channel Binding mechanism" (*Id.*, 13:5-8) which, instead of carrying the access network parameters in authentication methods, has both the server and the supplicant bind a key binding blob to channel binding keys:

STEP 1 (CBK Creation): *In this first step, depicted using reference numerals (1) in FIG. 3, the server 1 and the supplicant 3 create a CBK used for an authenticator 2. In the preferred embodiments, the CBK is derived from a CBMK and bound to a KBB associated with the authenticator using a KDF.* In the preferred embodiments, the KBB is pre-configured on the server.

STEP 2 (CBK Transfer): In this second step, depicted using reference numeral (2) in FIG. 3, ***the server 1 transfers the CBK to the authenticator 2.***

STEP 3 (CBK Verification): In this third step, depicted using reference numeral (3) in FIG. 3, ***the supplicant and authenticator verify proof of possession of the CBK over the ASP.*** In the preferred embodiments, after successful verification of proof of possession of the CBK, ***the supplicant and the authenticator are able to use the CBK in the ASP.***

Id., 13:64-14:12. Because the server and supplicant independently generate the keys, communication will only be possible if both the server and the supplicant have consistent parameters.

234. Accordingly, it is my opinion that a person of ordinary skill in the art would have understood that in the *Sood-Lee* combination SSID, MD-ID, R0KH-ID and R1KH-ID (“access network parameters”) are cryptographically bound to PTK (“a key”) without needing to carry the parameters in authentication methods because any discrepancy between the parameters on the server and at the subscriber station would prevent the authenticator and the subscriber station from deriving symmetric PTKs.

XI. SECONDARY CONSIDERATIONS

235. I am not aware of any evidence in the '671 patent's prosecution history or elsewhere supporting any secondary considerations arguments, or evidence of nexus of such alleged evidence to the challenged claims. *See generally* Ex. 1004. To the extent Patent Owner asserts the existence of any secondary considerations in its responses, I reserve the right to address any such evidence.

236. I declare that all statements made herein of my knowledge are true, 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.

Executed on: 1-22-2025



Narayan B. Mandayam