

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

Petition for *Inter Partes* Review of

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

TABLE OF CONTENTS

PETITIONERS’ EXHIBIT LIST vi

I. INTRODUCTION1

II. GROUNDS FOR STANDING.....1

III. STATEMENT OF PRECISE RELIEF REQUESTED FOR EACH CLAIM CHALLENGED1

IV. OVERVIEW OF THE ’671 PATENT2

V. OVERVIEW OF THE PRIOR ART5

 A. *Sood*.....6

 B. *Aboba*.....8

 C. *Lee*9

VI. LEVEL OF ORDINARY SKILL IN THE ART11

VII. CLAIM CONSTRUCTION11

 A. LEXICOGRAPHY12

 B. CLAIMS 1 & 6.....13

VIII. DETAILED EXPLANATION OF GROUNDS.....16

 A. GROUND 1: *SOOD* RENDERS OBVIOUS CLAIMS 1-4, 7, 8, 12, 14, AND 1617

 1. Claim 117

 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:”17

b.	1[a]: “cryptographically binding access network parameters to a key without needing to carry the parameters in authentication methods;”	21
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”	23
d.	1[c]: “wherein said key binding blob is a string that is constructed from static parameters advertised from an authenticator.”	32
2.	Claim 2: “The method of claim 1, wherein said authentication methods include EAP methods.”	32
3.	Claims 3 and 4.....	34
a.	Claim 3: “The method of claim 1, wherein said parameters include parameters advertised by an authenticator to said supplicant.”	35
b.	Claim 4: “The method of claim 3, wherein the identity of the authenticator is one of said parameters.”	37
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.”	38
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.”	38
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.”	40
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.”	41

B.	GROUND 2: <i>SOOD</i> IN COMBINATION WITH <i>ABOBA</i> RENDERS OBVIOUS CLAIMS 5 AND 18	42
1.	Claim 5: “The method of claim 1, wherein said channel binding master key is at least 64 octets long.”	42
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.”	46
C.	GROUND 3: <i>SOOD</i> IN COMBINATION WITH <i>LEE</i> RENDERS OBVIOUS CLAIMS 6, 10, 11, 13, 15, 17, AND 19	49
1.	Motivation to Combine <i>Sood</i> and <i>Lee</i>	49
2.	Claim 6	54
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:”	54
b.	6[a]	55
(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 ...”	55
(ii)	“... for use by the extensible authentication protocol authenticator as an extensible authentication protocol master session key ...”	56
(iii)	“... without said extensible authentication protocol authenticator needing to carry said access network parameters in extensible authentication protocol authentication methods;”	57

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;”	58
d.	6[c]: “wherein said key binding blob is a string that is constructed from static parameters advertised from said extensible authentication protocol authenticator.”	59
3.	Claim 10.....	59
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;”	59
a.	10[b]: “b) the server transfers the channel binding key to the authenticator; and”	61
a.	10[c]: “c) the supplicant and the authenticator verify proof of possession of the channel binding key over an authenticator-supplicant protocol.”	61
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.”	62
5.	Claims 13, 15, 17: “The method of claim [12/14/16], further including using channel binding keys to form a hierarchical channel binding.”	63
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).”.....	66
D.	GROUND 4: <i>SOOD</i> IN COMBINATION WITH <i>ABOBA</i> AND <i>LEE</i> RENDERS OBVIOUS CLAIMS 1-8 and 10-19	66
1.	“using an authenticator-supplicant protocol”	66

2.	“without needing to carry the parameters in authentication methods” / “without ... needing to carry said access network parameters in ... authentication methods”	68
IX.	THE DISCRETIONARY FACTORS FAVOR INSTITUTING TRIAL	71
A.	35 U.S.C. §314(a).....	71
1.	Stay.....	71
2.	Trial Date	72
3.	Diligence/Investment	73
4.	Overlap.....	74
5.	Parties.....	74
6.	Other considerations.	74
B.	35 U.S.C. §325(d).....	75
X.	MANDATORY NOTICES UNDER 37 C.F.R. §42.8.....	75
A.	Real Parties-in-Interest.....	75
1.	Related Matters	75
B.	Lead and Backup Counsel.....	76
C.	Service Information.....	76
D.	Power of Attorney	77
XI.	FEES	77

PETITIONERS' EXHIBIT LIST

Exhibit No.	DESCRIPTION
1001	U.S. Patent 8,239,671 (“’671Pat”)
1002	Declaration of Dr. Narayan B. Mandayam (“Mandayam”)
1003	Curriculum Vitae of Dr. Narayan B. Mandayam
1004	File History of U.S. Patent 8,239,671
1005	U.S. 7,787,627 (“Sood”)
1006	Bernard Aboba et al., “Extensible Authentication Protocol (EAP) Key Management Framework Version 3” EAP Working Group INTERNET-DRAFT (Jul. 18, 2004) (“Aboba”)
1007	U.S. 2004/0242228 (“Lee”)
1008	U.S. 8,027,304 (“Forsberg”)
1009	IEEE Std 802.11i (“802.11i”)
1010	Declaration of Laura Nugent For IETF Administration LLC
1011	RFC 5247 IETF Datatracker Page
1012	C. Kaufman, “Internet Key Exchange (IKEv2) Protocol” Network Working Group Request for Comments: 4306 (Dec. 2005) (“IKEv2”)
1013	Excerpts from Microsoft Computer Dictionary (5 th ed. 2002) (“Computer Dictionary”)
1014	Excerpts from IEEE Std 802.11-1997 (“802.11-1997”)
1015	B. Aboba et al., “Extensible Authentication Protocol (EAP)” EAP Working Group Request for Comments: 3748 (Jun. 2004) (“EAP”)

Exhibit No.	DESCRIPTION
1016	Morris Dworkin, “Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication,” NIST Special Publication 800-38B (May 2005) (“ <i>Dworkin</i> ”)
1017	U.S. 7,602,918 (“ <i>Mizikovsky</i> ”)
1018	U.S. 8,621,201 (“ <i>Costa</i> ”)
1019	U.S. 7,969,945 (“ <i>Navali</i> ”)
1020	Nut Taesombut et al., “A Secure Multimedia System in Emerging Wireless Home Networks,” CMS 2003, LNCS 2828, pp. 76-88 (2003) (“ <i>Taesombut</i> ”)
1021	Docket Control Order from <i>Four Batons Wireless, LLC v. Samsung Electronics Co., LTD, et al.</i> , No. 2:24-cv-0284-JRG (ECF 035) (E.D. Tex. Aug. 26, 2024)
1022	Cover Pleading of Plaintiff Four Batons Wireless, LLC’s Disclosures Pursuant to Local Patent Rules 3-1 and 3-2, Dated July 25, 2024

I. INTRODUCTION

Samsung Electronics Co., Ltd. and Samsung Electronics America, Inc. (“Samsung” or “Petitioners”) request *inter partes* review (“IPR”) of claims 1-8 and 10-19 (“Challenged Claims”) of U.S. Patent No. 8,239,671 (“the ’671 patent”).

II. GROUNDS FOR STANDING

Petitioners certify that the ’671 patent is available for IPR, and that Petitioners are not barred or estopped from requesting IPR to challenge the claims on the grounds herein.

III. STATEMENT OF PRECISE RELIEF REQUESTED FOR EACH CLAIM CHALLENGED

Petitioners respectfully request review and cancellation under 35 U.S.C. §311 of the Challenged Claims in view of:¹

¹ Petitioners do not concede that any Challenged Claims satisfy other requirements for patentability that cannot be raised in IPR, including 35 U.S.C. §§101 and 112.

Grounds	Claims	Basis
1	1-4, 7, 8, 12, 14, 16	§103: <i>Sood</i>
2	5, 18	§103: <i>Sood</i> + <i>Aboba</i>
3	6, 10, 11, 13, 15, 17, 19	§103: <i>Sood</i> + <i>Lee</i>
4	1-8, 10-19	§103: <i>Sood</i> + <i>Aboba</i> + <i>Lee</i>

As shown below, each reference pre-dates the '671 patent's earliest purported priority date, April 20, 2006.

Reference	Date	Pre-AIA Prior Art at Least Under
US 7,787,627 (EX1005) (" <i>Sood</i> ")	11/30/2005 (filing date) 08/31/2010 (publication date)	§102(e)
"Extensible Authentication Protocol (EAP) Key Management Framework Version 3" (EX1006) (" <i>Aboba</i> ")	07/18/2004 (publication date)	§102(b)
US 2004/0242229 (EX1007) (" <i>Lee</i> ")	12/02/2004 (publication date)	§102(b)

IV. OVERVIEW OF THE '671 PATENT

The '671 patent is directed 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.” ’671*Pat*, Abstract.

The ’671 patent defines the following terms:

- 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.

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 ASP.” *Id.*, 14:7-10. These steps, numbered (1) to (3), are depicted in figure 3. *Id.*, FIG. 3.

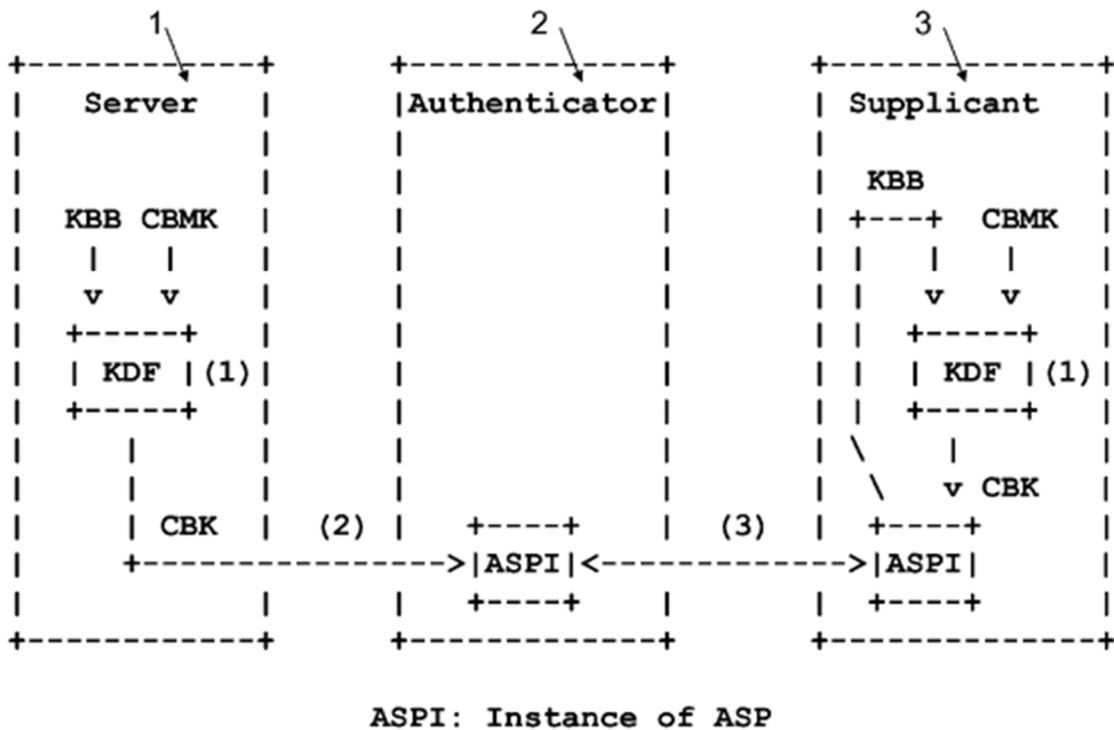


Figure 3: Basic Channel Binding Mechanism

'671Pat, FIG. 3.

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.

V. OVERVIEW OF THE PRIOR ART

By April 2006, methods for channel binding that bound static parameters to a key used by an authenticator were well-known. *Mandayam* ¶¶41-44 (discussing *Mizikovsky*, *Costa*, and *Navali*). *Sood*, *Aboba*, and *Lee*—discussed further below—are just a few examples of such systems.

Moreover, the prior art references analyzed in the grounds below are analogous to the '671 patent because they are within the same field of endeavor as the '671 patent and reasonably pertinent to one or more problems addressed by the '671 patent. *Mandayam* ¶45. For example, like the '671 patent, *Sood* is directed to a “key management system” that generates keys bound to static parameters and used by authenticators. *Id.* ¶46. *Aboba* is directed to similar key generation techniques to generate “keying material used on different authenticators.” *Id.* ¶47. *Aboba*'s keying material is disclosed to be bound to static parameters and used at an authenticator. *Id.* *Lee* is also directed to a similar key management scheme for “providing a

proactive key,” which *Lee* discloses is bound to static parameters and used at an authenticator. *Id.* ¶48.

A. *Sood*

Sood is titled “Methods and Apparatus For Providing a Key Management System For Wireless Communication Networks.” *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.

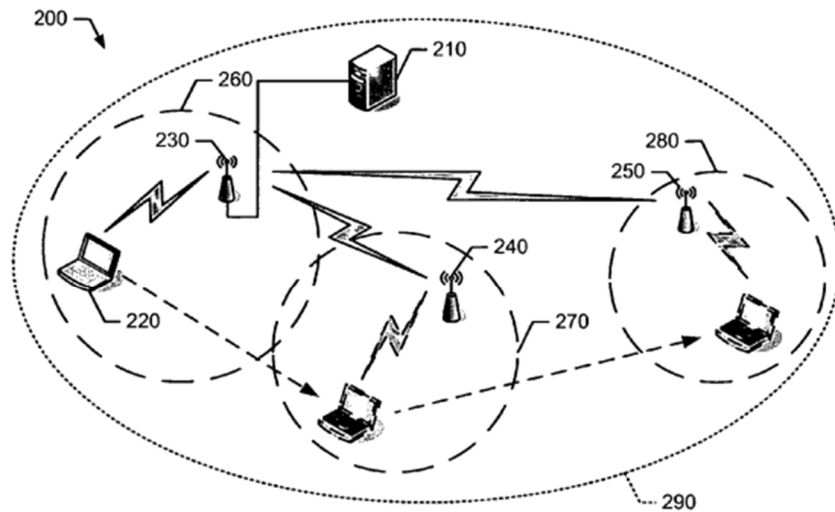


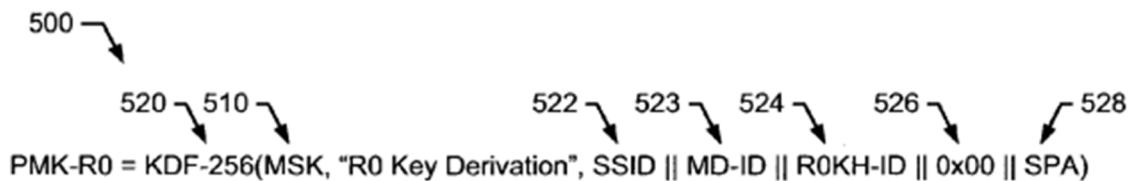
FIG. 2

Sood, FIG. 2.

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.

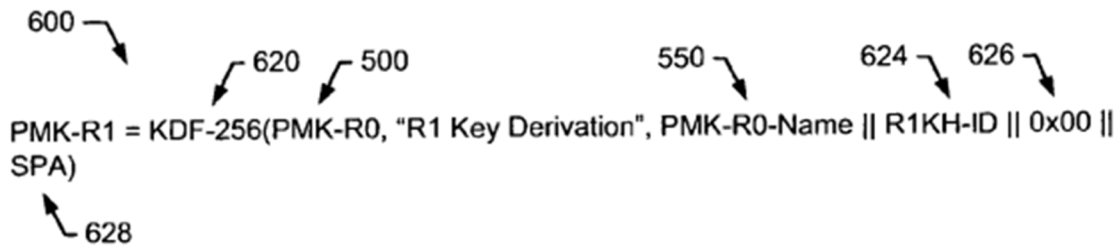
Sood's system allows a subscriber to "roam between coverage areas" of different access points. *Id.* at 4:19-22. *Sood* is particularly directed to providing a key hierarchy allowing for fast roaming so that "the subscriber station 220 may avoid performing a full authentication process with the authentication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

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:



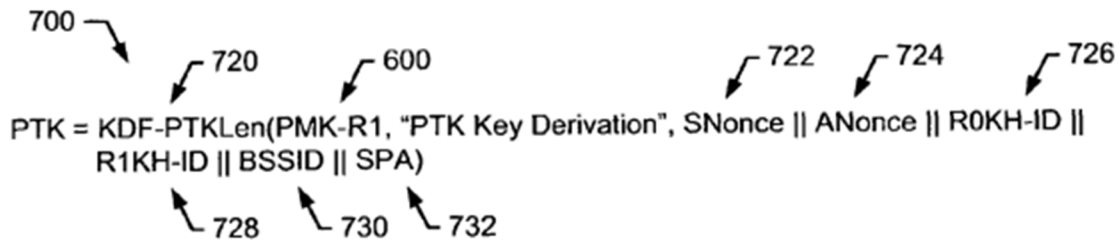
Sood, FIG. 5 (excerpted).

Sood discloses that the server transfers PMK-R0 to an access point (*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:



Sood, FIG. 6 (excerpted).

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:



Sood, FIG. 7 (excerpted).

B. *Aboba*

Aboba is version 3 of the draft “Extensible Authentication Protocol (EAP) Key Management Framework” standard, later finalized as RFC 5247. *Aboba*, 1; EX1011. *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.” *Aboba*, 1.

Aboba’s Appendix E discloses a technique for “fast handoff between authenticators” by providing “keying material to multiple authenticators in order to facilitate fast handoff.” *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).” *Aboba*, 70.

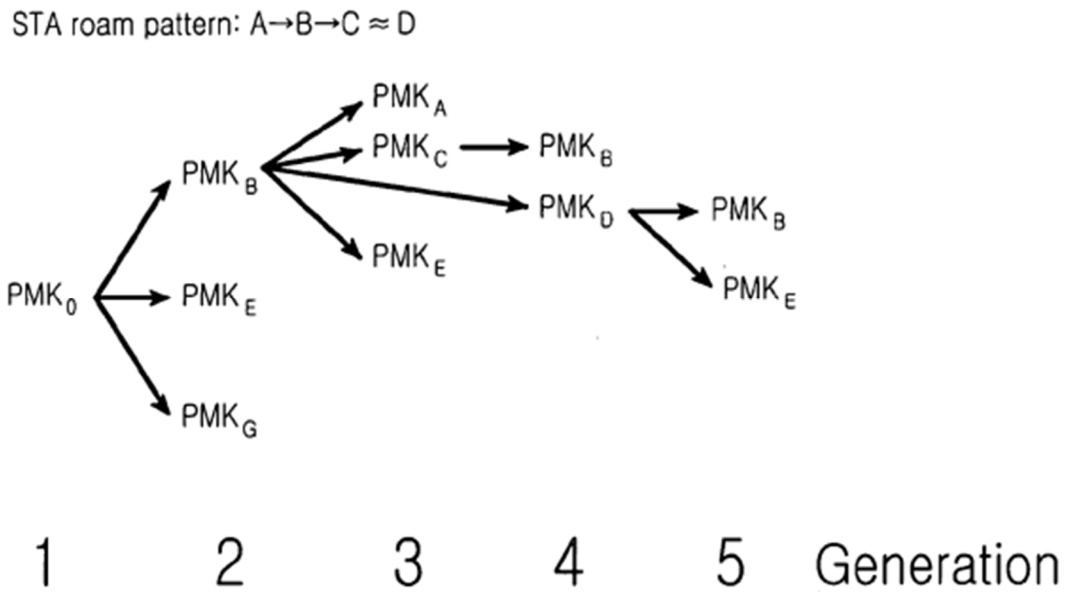
As established by the declaration of Laura Nugent, an announcement of publication for *Aboba* was made on July 19, 2004 and *Aboba* was available to the public via the IETF online directory within 24 hours of that announcement. EX1010, ¶¶7, 10. As part of the online directory, *Aboba* was indexed and searchable on the IETF website. *Id.*, ¶6. A POSITA would have been aware of IETF as a well-known standards organization and platform for the publication of proposed internet standards. *Mandayam* ¶57. Thus, *Aboba* qualifies as a printed publication. *See Weber, Inc. v. Provisur Technologies, Inc.*, 92 F.4th 1059, 1067 (Fed. Cir. 2024) (“The standard for public accessibility is whether interested members of the relevant public could locate the reference by reasonable diligence.”).

C. *Lee*

Lee is titled “Method For Fast Roaming in a Wireless Network.” *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].

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].

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:



Lee, FIG. 11.

VI. LEVEL OF ORDINARY SKILL IN THE ART

A person of ordinary skill in the art (“POSITA”) at the time of the ’671 patent’s purported priority date, which for purposes of the Petition is assumed to be April 20, 2006, 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. *Mandayam* ¶¶21-23. A greater amount of professional education could compensate for fewer years of work experience, and vice versa. *Id.*

VII. CLAIM CONSTRUCTION

Petitioners are unaware of any “prior claim construction determination” related to the ’671 patent. *See* 37 C.F.R. §42.100(b).

A. LEXICOGRAPHY

Each of the following terms should be construed according to the “Terminology” section of the ’671 patent’s specification. *See Intel Corp. v. Parkervision, Inc.*, IPR2021-00346, Paper 33 at 20 (PTAB Jun. 30, 2022) (holding section titled “Terminology” “provides lexicographic definitions”).

- 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).” *’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.

B. CLAIMS 1 & 6

The term “[deriving a/derive said] channel binding key from a channel binding master key bound to a key binding blob using a key derivation function,” recited in claims 1 and 6 of the ’671 patent, requires construction. Consistent with the other claims and the specification, the Board should construe this term to require that the “key binding blob” is bound to the “channel binding key” (not the “channel binding master key”).

Reviewed in isolation, 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. In such an instance, the specification is instructive. *World Class Tech. Corp. v. Ormco Corp.*, 769 F.3d 1120, 1123-24 (Fed. Cir. 2014) (instructing that when claim language “does not by itself convey a clear, unambiguous meaning,” we look to the specification to resolve the uncertainty). The lexicography in the specification resolves this ambiguity, instructing that a channel binding key “is derived from a Channel Binding Master Key (CBMK) *and*

cryptographically bound to a Key Binding Blob (KBB).”² *'671Pat*, 13:18-21. A POSITA would have understood that 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. *Mandayam* ¶64. 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.

Additionally, 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. As the Federal Circuit has instructed, “we must not interpret an independent claim in a way that is inconsistent with a claim which depends from it.” *Wright Med. Tech., Inc. v. Osteonics Corp.*, 122 F.3d 1440, 1445 (Fed. Cir. 1997); *see also Baxalta Inc. v. Genentech, Inc.*, 972 F.3d 1341, 1345-46 (Fed. Cir. 2020) (looking to dependent claim to determine the scope of the independent claim).

² All **bold/italics/color** emphases and annotations added unless noted otherwise.

For example, claim 18 depends on claim 1 and further recites “wherein said key derivation function is computed based on $CBK = kdf+(CBMK, KBB) \dots$ ” *Id.*, Claim 18. Note that “said key derivation function” in claim 18 has antecedent basis in “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 “ $CBK = kdf+(CBMK, KBB)$.” This notation would indicate to a POSITA 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. *Mandayam* ¶66 (citing *IKEv2*, 28). There is no key derivation function that would have been known to a POSITA 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. *Id.*

Dependent claims 14 and 16 would also not have been intelligible to a POSITA unless the key binding blob is bound to the channel binding key. *Mandayam* ¶67. 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.” *Mandayam* ¶67. A POSITA 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. *Id.*, ¶67.

The remaining claim terms should be given their plain and ordinary meaning.³ *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc).

VIII. DETAILED EXPLANATION OF GROUNDS

The below, as supported by the Declaration of Dr. Narayan B. Mandayam demonstrates how the Challenged Claims are unpatentable. *See* 37 C.F.R. §42.104(b)(4)-(5).

³ Petitioners reserve all rights to raise claim construction arguments and other arguments in any parallel or future litigation concerning the '671 patent. For example, comparing the claims to the accused products in the litigation may raise controversies that require construction of additional claim terms.

A. GROUND 1: *SOOD* RENDERS OBVIOUS CLAIMS 1-4, 7, 8, 12, 14, AND 16

1. Claim 1

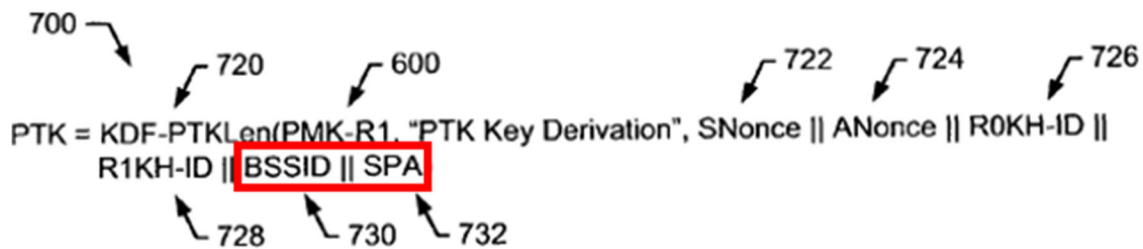
- a. 1[pre]: “A channel binding method based on parameter binding in a key derivation procedure for authentication of a mobile supplicant to an access network, comprising:”**

To the extent the preamble is limiting, *Sood* teaches this feature.⁴ *Mandayam* ¶¶74-84.

A POSITA 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. *Mandayam* ¶75 (discussing *Forsberg*). 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.” *’671Pat*, 5:5-6.

⁴ Petitioners use the term “teaches” as including both express teachings and those fairly suggested to a person of ordinary skill in the art. *In re Baird*, 16 F.3d 380, 383 (Fed. Cir. 1994); *In re Keller*, 642 F.2d 413, 425 (CCPA, 1981) (“The test for obviousness is ... what the combined teachings of the references would have suggested to those of ordinary skill in the art.” (citations omitted)).

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. *Mandayam* ¶77. 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:



Sood, FIG. 7 (excerpted and annotated).

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." *Sood*, 9:63-65. This would be consistent with a POSITA's understanding of "channel binding." *Mandayam* ¶78. 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 POSITA would have understood. *Id.* Thus, *Sood* discloses binding MAC-layer identifiers (parameter binding) in the derivation of PTK (a key derivation procedure).

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.” *Sood*, 1:55-57. One such network is illustrated in *Sood*’s figure 2:

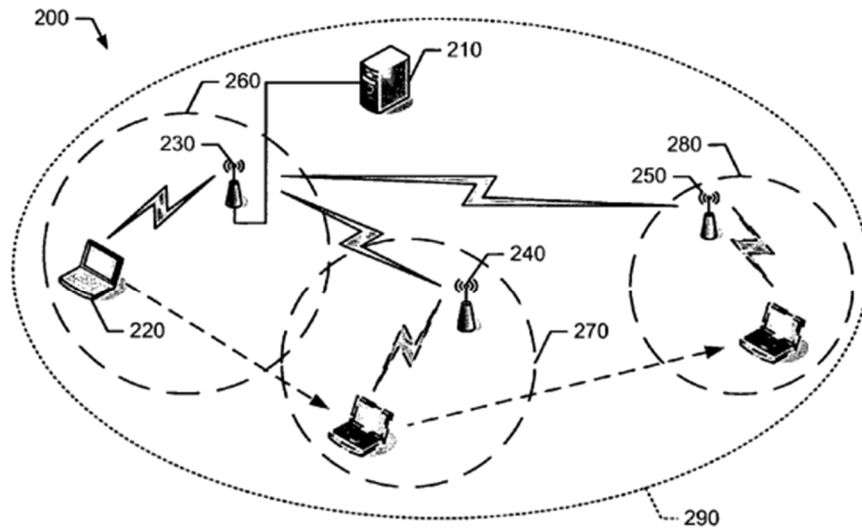


FIG. 2

Sood, FIG. 2.

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).” *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. *Sood*, 2:5-19.

As discussed in §VII.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, e.g., laptop computers or components thereof. These supplicants utilize the PMK-R1 key (“channel binding key”) to generate a session key for use in communications.

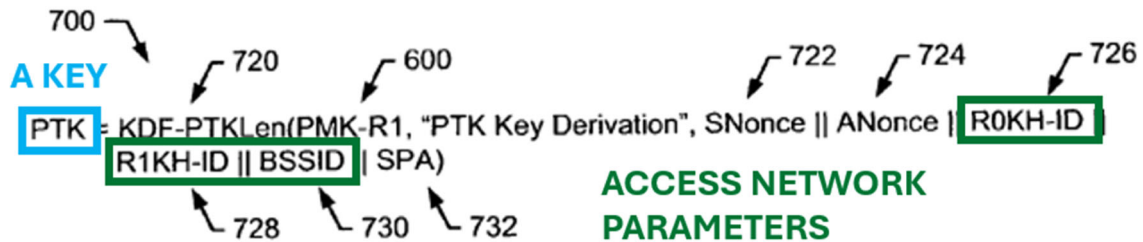
Mandayam ¶81; *Sood*, 9:13-17, FIG. 7 (showing the derivation of PTK from PMK-R1). *Sood* discloses using the IEEE 802.11i protocol to communicate (“authenticator-suppliant protocol”) because it states “the methods and apparatus disclosed herein may be applied to WPANs, *WLANs*, WMANs, and/or WWANs,” and gives as an example of a WLAN “the IEEE std. 802.11i” (ASP). *Sood*, 11:56-58, 2:37-44, 9:45-48. A POSITA 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 suppliant and an authenticator and uses a CBK for protecting the protocol.” ’671*Pat*, 13:49-51. In particular, 802.11i utilizes a “PMK” to generate session keys for use within the protocol. *Mandayam* ¶82; *see also 802.11i*, 90 (showing the derivation of session keys within the 802.11i protocol). A POSITA 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. *Compare 802.11i*, 90 with *Sood*, FIG. 7; *Mandayam* ¶¶82-83. Additionally, the ’671 patent identifies 802.11i as an example of an authenticator-suppliant protocol. *See* ’671*Pat*, 15:53-58.

Accordingly, *Sood* teaches 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 suppliant and an authentication server on a wireless

communications network. (“for authentication of a mobile supplicant to an access network”). *Mandayam* ¶84.

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

Sood teaches this feature. *Mandayam* ¶¶85-92. 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 **PTK** (“a key”):



Sood, FIG. 7 (excerpted and annotated).

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 POSITA. *Mandayam* ¶87 (citing *IKEv2*, 27-28). These functions are configured to take parameters and/or keys as input and output a new key cryptographically bound to the inputs. *Mandayam* ¶87. Here, **R0KH-ID**, **R1KH-ID**, and **BSSID** are inputs to the KDF-PTKLen function and, thus, are bound to **PTK** by the key derivation. *Id.*

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.” *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 POSITA would have understood to be another parameter that identifies an access point. *Id.*, 9:63-64; *Mandayam* ¶88 (citing *Aboba*, 70).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. *Mandayam* ¶88. This is consistent with the specification of the ’671 patent which discloses “the identity of the EAP authenticator” as an example of an access network parameter. *’671Pat*, 12:59-62.

A POSITA would have understood that *Sood*’s method does not need to carry the access network parameters in authentication methods. *Mandayam* ¶89. *Sood* discloses that at least **R0KH-ID** and **R1KH-ID** are advertised by the access point to the subscriber station in a “beacon.” *Sood*, 7:10-20, 8:27-36; *Mandayam* ¶89. A POSITA 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. *Mandayam* ¶90 (citing *802.11-1997*, 13).

⁵ To be clear, here *Aboba* is being used to demonstrate a POSITA’s knowledge. *Aboba* is not part of unpatentability Ground 1.

Because at least these three access network parameters are available to the supplicant via beacon(s), there is no need to carry these parameters in authentication methods.

A POSITA would have understood that advertising parameters is not carrying parameters in authentication methods because advertising (e.g., in a beacon) occurs pre-authentication. *Mandayam* ¶91. 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.” '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. *Mandayam* ¶91.

Accordingly, *Sood* teaches 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”). *Mandayam* ¶92.

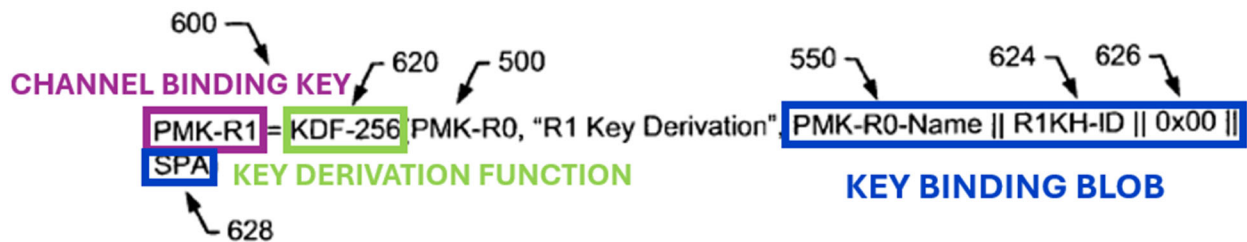
- c. 1[b]: “further including deriving a channel binding key from a channel binding master key bound to a key binding blob using a key derivation function; and”

Sood teaches this feature. *Mandayam* ¶¶93-114.

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-**

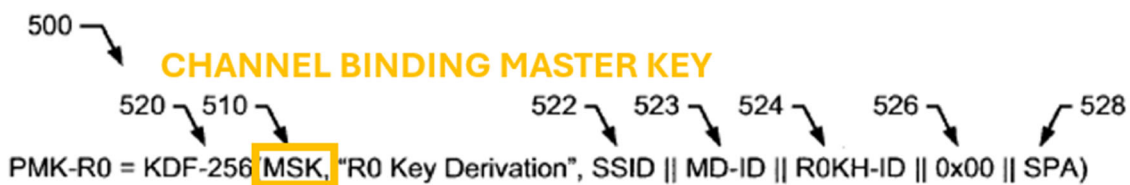
R0. This section first discuss limitation 1[b] with **MSK** as the channel binding master key. Next, the section discusses limitation 1[b] with **PMK-R0** as the channel binding master key.

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:



Sood, FIG. 6 (excerpted and annotated).

A POSITA 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. *Mandayam* ¶95. PMK-R0 is in turn derived from **MSK** using the same KDF-256 function, as shown in figure 5:



Sood, FIG. 5 (excerpted and annotated).

Thus, a POSITA would have understood that **PMK-R1** is ultimately derived from **MSK** (“channel binding master key”). *Mandayam* ¶96.

Additionally, *Sood*'s **PMK-R0** can 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:



Sood, FIG. 6 (excerpted and annotated).

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 §VII.A (Lexicography), a channel binding key 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).” **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 POSITA would have understood that inputting **the concatenation** to the key derivation function binds it to the output of the function, i.e., **PMK-R1**. *Mandayam* ¶98.

As also discussed in §VII.A (Lexicography), a key binding blob is “[a]n octet-string that is constructed from static parameters advertised from an authenticator using an Authenticator-Supplicant Protocol (ASP).” A POSITA would have recognized the “||” operators that form the key binding blob in figure 6 to be well-known notation for a concatenation operation. *Mandayam* ¶99 (citing *Dworkin*, 9, 10). Concatenation is an operation performed on two strings to join them into a single string. *Mandayam* ¶100 (citing *Computer Dictionary*, 4). Furthermore, *Sood* discloses that at least **R1KH-ID** is a “string octet.” *Sood*, 8:27-35. A POSITA would have understood “string octet” to be a typo of “octet-string.” *Mandayam* ¶101. Because the key binding blob of figure 6 is formed by a concatenation of **R1KH-ID**, it is an octet string.

The key binding blob of figure 6 is also constructed from static parameters advertised from an authenticator. In particular, for the reasons discussed below, a POSITA 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.

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.” *Sood*, 7:38-41. A POSITA would have understood that MD-ID is

static because it is defined by a network administrator. *Mandayam* ¶103. MD-ID is also advertised from an authenticator because *Sood*'s "access points" are authenticators. As discussed in §VII.A (Lexicography), an authenticator is "[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." *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:

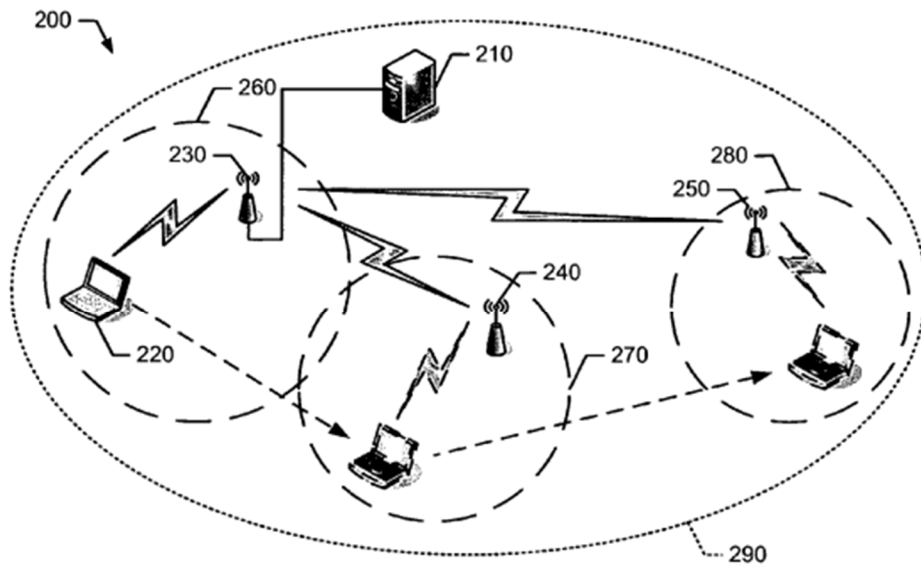
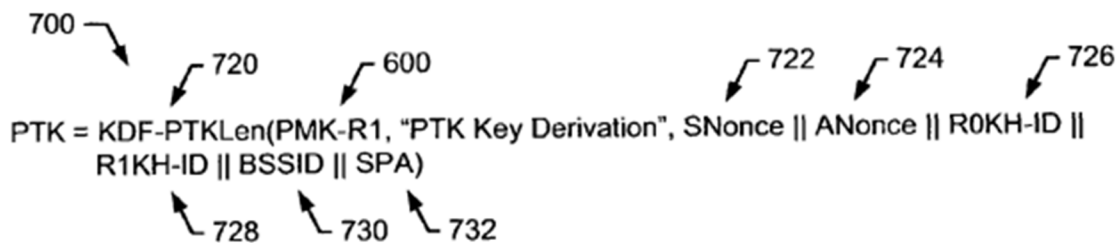


FIG. 2

Sood, FIG. 2.

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.” *Sood*, 11:38-40. A “session key” may be a “pairwise temporal key (PTK).” *Sood*, 9:5-7. As illustrated in figure 7, generating a PTK utilizes **PMK-R1** as an input (*Mandayam* ¶104):



Sood, FIG. 7.

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

R1KH-ID is also a static parameter advertised by an authenticator. *Sood* discloses that R1KH-ID “may be advertised in a beacon,” which a POSITA would have understood is a beacon of an access point. *Sood*, 8:27-36; *Mandayam* ¶106. R1KH-ID is static because it is an identifier of a “network entity.” *Sood*, 8:27-30; *Mandayam* ¶106.

SSID is also a static parameter advertised by an authenticator. A POSITA would have understood that advertising SSIDs is a well-known feature of wireless networks. *Mandayam* ¶107; *see also 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. *Mandayam* ¶107.

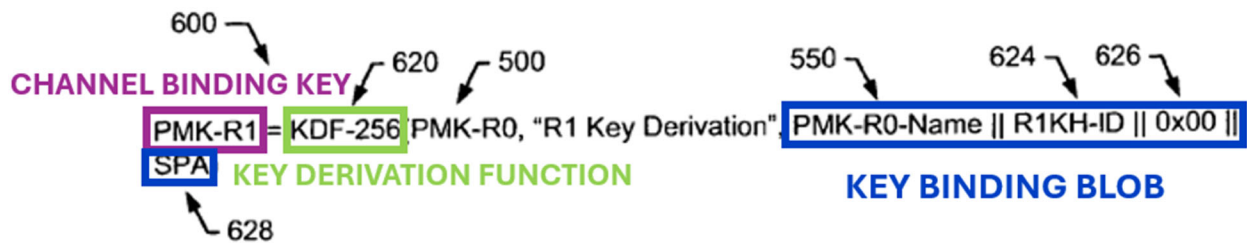
The static parameters SSID, MD-ID, and R1KH-ID are also advertised using the 802.11i protocol (ASP). *Sood* discloses that “the methods and apparatus disclosed herein may be applied to WPANs, *WLANs*, WMANs, and/or WWANs” and gives as an example of a WLAN “the IEEE std. 802.11i” (ASP). *Sood*, 11:56-58, 2:37-44, 9:45-48. A POSITA would have understood that the 802.11i protocol has advertising features. *Mandayam* ¶108 (citing *Aboba*, 25; *802.11i*, 38). Thus, a POSITA 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. *Mandayam* ¶108.

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:



Sood, FIG. 5 (excerpted and annotated).

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



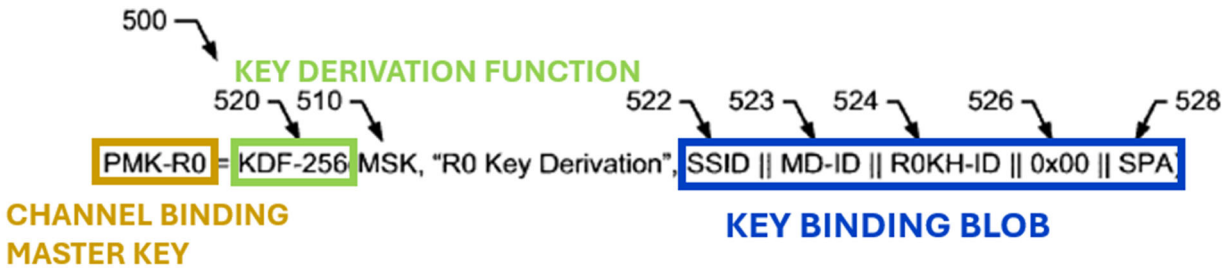
Sood, FIG. 6 (excerpted and annotated).

Accordingly, *Sood* teaches 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”). *Mandayam* ¶110.

* * *

As explained in §VII.B (Claim Construction) above, a POSITA would have understood this element to require that the key binding blob of claim 1 is bound to the channel binding key (not the channel binding master key). However, if PO argues and the Board determines that this claim element requires that the key binding blob be bound to the channel binding master key, *Sood* nevertheless teaches this limitation. The below analysis addresses this potential alternative claim construction, referred to hereinafter as *PO’s Potential Alternative Construction*.

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:



Sood, FIG. 5 (excerpted and annotated).

The concatenation of figure 5 meets the definition of a key binding blob: “[a]n octet-string that is constructed from static parameters advertised from an authenticator using an Authenticator-Supplicant Protocol (ASP).” *'671Pat*, 13:27-30. *Sood* discloses that **R0KH-ID** is an octet string. *Sood*, 7:10-18; *Mandayam* ¶113. Because the key binding blob of figure 5 is a concatenation of **R0HK-ID**, it is also an octet string. *Mandayam* ¶113. **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 POSITA would have understood is a beacon of an access point. *Sood*, 7:10-20; *Mandayam* ¶114. **R0KH-ID** is static because it is an identifier of a

“network entity.” *Sood*, 7:10-13; *Mandayam* ¶114. 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-supplicant protocol). *Mandayam* ¶108.

Accordingly, *Sood* teaches 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”). *Mandayam* ¶115.

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

This limitation recites the lexicography of “key binding blob” and is taught by *Sood* for the same reasons discussed in §VIII.A.1.c (Element 1[b]). *Mandayam* ¶¶93-114.

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

Sood teaches this claim, including under *PO’s Potential Alternative Construction*. *Mandayam* ¶¶117-122.

The “said authentication methods” of claim 2 refer back to the limitation “cryptographically binding access network parameters to a key without needing to

carry the [access network] parameters *in authentication methods*” of claim 1. For context, in the prosecution history of the ’671 patent, the applicant argued that a certain prior art reference 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.” EX1004, 56. Thus, 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. *Mandayam* ¶118.

There 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 explained in §VIII.A.1.c (Element 1[b]), there is no need in *Sood* to carry access network parameters in any kind of authentication method because the subscriber station receives the necessary parameters via advertisements from access points.

Accordingly, *Sood* teaches 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”). *Mandayam* ¶120.

To the extent that PO argues and the Board determines 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).” *Sood*, 6:56-62. These PMKs are used to derive a PTK. *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 explained in §VIII.A.1.b.

Accordingly, *Sood* teaches 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.”). *Mandayam* ¶122.

3. Claims 3 and 4

In claims 3 and 4, “said parameters” does not have clear antecedent basis, and Petitioners address two possible interpretations: (1) “said parameters” has antecedent basis to “access network parameters,” or (2) “said parameters” has antecedent basis to “static parameters.” Because claims 3 and 4 are unpatentable under either possibility, the Board can properly resolve their patentability despite the

ambiguity in the claim language.⁶ *See Realtime Data v. Iancu*, 912 F.3d 1368, 1375 (Fed. Cir. 2019); *Palo Alto Networks, Inc. v. BT Americas Inc.*, IPR2023-00889, Paper 32 at 87 n.17 (Nov. 7, 2024); *Maxlinear, Inc. v. Cresta Tech. Corp.*, IPR2015-00592, Paper 87 at 20 (Apr. 17, 2019); *Mandayam* ¶123.

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

Sood teaches this claim, including under *PO’s Potential Alternative Construction*. *Mandayam* ¶¶124-127.

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 as explained in §VIII.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 as explained in §VIII.A.1.c (Element 1[b]). As explained in §VIII.A.1.c (Element 1[b]), access points are authenticators. A POSITA 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. *Mandayam* ¶126. This understanding

⁶ Petitioners reserve the right to raise and address indefiniteness, enablement, and/or written description issues regarding Claims 3 and 4 in other proceedings.

of a POSITA is buttressed by *Sood*'s figure 2, which shows subscriber station 220 (“supplicant”) within the coverage area 260 of access point 230 (“authenticator”).

Mandayam ¶126; see also *Sood*, 4:57-65.

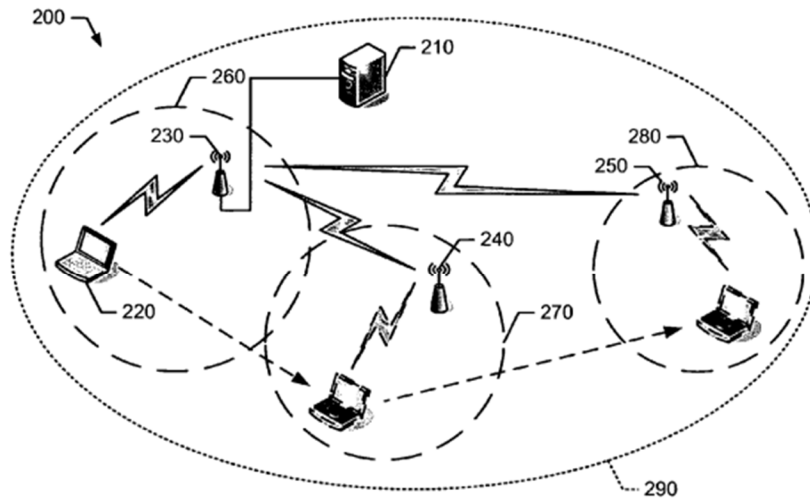


FIG. 2

Sood, FIG. 2.

Thus, a POSITA would have understood *Sood* to teach that parameters advertised by access point 230 are advertised to subscriber station 220. *Mandayam* ¶126.

Accordingly, *Sood* teaches 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”). *Mandayam* ¶127.

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

Sood teaches this claim, including under *PO’s Potential Alternative Construction. Mandayam* ¶¶128-131.

Petitioners again address both possible interpretations of “said parameters.” 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 discussed in §VIII.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 discussed in §VIII.A.1.c (Element 1[b]).

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).” *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 POSITA would have understood that R1KH-ID identifies an access point (authenticator) and its associated network access server. *Mandayam* ¶130; *see 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).

Accordingly, *Sood* teaches 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”). *Mandayam* ¶131.

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.”**

Claim 7 recites the lexicography of “key binding blob” and is taught by *Sood* for the same reasons discussed in §VIII.A.1.c (Element 1[b]). *Mandayam* ¶¶93-114.

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.”**

Sood teaches this claim, including under *PO’s Potential Alternative Construction*. *Mandayam* ¶¶133-138.

As explained in §VIII.A.1.c (Element 1[b]), *Sood’s* access point (“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.” *Sood*, 9:2-9. A POSITA would have understood that communicating “using session keys” involves using the keys to encrypt communications, thus protecting the communications. *Mandayam* ¶134. As explained in §VIII.A.1.a (Element 1[pre]), these session keys are used within the IEEE 802.11i protocol (“authenticator-suppliant 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. *Mandayam* ¶135.

Nothing in claim 8 requires the access point and the subscriber station to themselves derive the session keys from the channel binding master key. *Mandayam* ¶136. Indeed, the '671 patent explicitly teaches that “the server ... create[s] a CBK used for an authenticator” and “transfers the CBK to the authenticator.” '671*Pat*, 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. *Mandayam* ¶136.

However, to the extent that PO argues and the Board determines that claim 8 requires that the access point and the subscriber station themselves derive the session keys from the channel binding master key, *Sood* also teaches this feature. *Mandayam* ¶137. *Sood* teaches that “the authentication server 210 may forward the MSK to the access point 230, which in turn, may generate [PMK-R0].” *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. *Id.*, 6:51-56. A POSITA 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. *Mandayam* ¶137. 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. *Mandayam* ¶137.

Accordingly, *Sood* teaches 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”). *Mandayam* ¶138.

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.”

Sood teaches this claim, including under *PO’S Potential Alternative Construction*. *Mandayam* ¶¶139-143.

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

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). *Sood*, 8:3-6. As explained in §VIII.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 POSITA 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. *Mandayam* ¶141; *see also Sood*, 6:51-59. *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.” *Sood*, 8:10-13.

Accordingly, *Sood* teaches 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”). *Mandayam* ¶143.

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.”

Sood teaches these claims. *Mandayam* ¶¶144-147. As discussed in §VIII.A.6 (Claim 12), *Sood* discloses “creating multiple channel binding keys from a single channel binding master key for multiple authenticators.”

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

Accordingly, *Sood* teaches 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”). *Mandayam* ¶147.

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.”

Sood-Aboba teaches this claim, including under *PO’s Potential Alternative Construction*. *Mandayam* ¶¶148-160.

A POSITA would have understood that an octet is a unit of data that consists of exactly 8 bits. *Mandayam* ¶149 (citing *Computer Dictionary*, 5).

As explained in §VIII.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.” *Aboba*, 14. A POSITA would have understood that *Aboba*’s MSK is analogous to *Sood*’s MSK because both MSKs are “derived between the EAP peer and server” and used to derive lower level keys. *Aboba*, 14, 51; *Sood*, 6:51-59; *Mandayam* ¶150.

A POSITA 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. *Sood*, 10:58-61. A POSITA 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.” *Aboba*, 16; *EAP*, 51. Thus, a POSITA would have understood that 64 octets is a sufficient length to securely perform the similar derivations of PMKs and PTKs taught by *Sood*. *Mandayam* ¶151. Additionally, this modification would have been obvious to a POSITA 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). *Mandayam* ¶152. The modification would also have been obvious to a POSITA 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). *Id.*

Accordingly, the *Sood-Aboba* combination teaches an MSK (“said channel binding master key”) of at least 64 octets (“is at least 64 octets long”). *Mandayam* ¶153.

To the extent that *Sood*'s PMK-R0 is mapped to the claimed “channel binding master key,” a POSITA would have found it obvious to modify *Sood* to use an at least 64 octet PMK-R0 based on *Aboba*'s teachings. In *Aboba*, the keys derived from the MSK and transported from the server to the authenticator are named “AAA-Key.” *Aboba*, 7 (“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.*”). 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.* *Aboba* further teaches that an AAA-Key is 64 octets long, indicating that “AAA-Key-A = MSK(0,63).” *Aboba*, 70. A POSITA would have understood that this notation indicates that the AAA-Key-A is the first 64 octets of the MSK. *Mandayam* ¶156.

A POSITA 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. *Mandayam* ¶157; *see Sood*, 7:64-67. A POSITA 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*.” *Sood*, 7:4-5. Thus, *Sood* expressly teaches using other suitable KDFs (e.g., KDFs that would produce different length outputs) to derive PMK-R0. *Mandayam* ¶158. A POSITA would have found it obvious to implement *Aboba*’s suggestion to use 64 octet keys to implement PMK-R0 because, as discussed above, using a 64 octet key would have been sufficient to securely perform subsequent key derivations. *Mandayam* ¶158.

A POSITA would have had a reasonable expectation of success in incorporating *Aboba*’s teachings regarding 64 octet keys in *Sood* because a POSITA 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. *Mandayam* ¶159; *Aboba*, 71 (discussing “the PRF+ key expansion PRF from [IKEv2]” that “may produce up to 5100 bytes of key material”) (brackets in original).

Accordingly, the *Sood-Aboba* combination teaches a PMK-R0 (“said channel binding master key”) of at least 64 octets (“is at least 64 octets long”). *Mandayam* ¶160.

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

Sood-Aboba teaches this claim. *Mandayam* ¶¶161-168.

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.



Sood, FIG. 6 (excerpted and annotated).

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.” ’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:

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

wherein:

$$\text{T1} = \text{prf}(\text{K}, \text{S} | 0\text{x}01)$$

$$\text{T2} = \text{prf}(\text{K}, \text{T1} | \text{S} | 0\text{x}02)$$

$$\text{T3} = \text{prf}(\text{K}, \text{T2} | \text{S} | 0\text{x}03)$$

$$\text{T4} = \text{prf}(\text{K}, \text{T3} | \text{S} | 0\text{x}04)$$

'671Pat, 15:12-18.

A POSITA would have understood that this iterative behavior allows the prf+ of RFC 4306 to generate a variable length output. *Mandayam* ¶164. In fact, RFC 4306 explains this behavior allows the key derivation to continue “as needed to compute all required keys.” *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.*; *802.11i*, 90 (referring to “PRF-384” and “PRF-512”). Thus, a POSITA would have understood the notation kdf+ to refer to a key derivation function that generates a variable length output.

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 *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. *Mandayam* ¶165. Additionally, a POSITA would have understood that this key derivation binds AAA-Key-B to “B-Called-Station-Id.” *Mandayam* ¶165. *Aboba* indicates that B-Called-Station-Id is “AP B MAC address,” i.e., an identifier for the access point. *Aboba*, 70; *Mandayam* ¶165. Because AAA-Key-B is bound to the identity of the access point, a POSITA would have understood that this key is analogous to *Sood*’s PMK-R1 (“channel binding key”). *Mandayam* ¶165.

A POSITA 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*.” *Sood*, 8:26-27. A POSITA 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. *Mandayam* ¶166. A POSITA 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 Aboba*, 71 with *Sood*, Fig. 6; *Mandayam* ¶166. A POSITA 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. *Mandayam* ¶166.

A POSITA 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). *Mandayam* ¶167.

Accordingly, the *Sood-Aboba* combination teaches 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"). *Mandayam* ¶168.

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*

Sood teaches deriving and distributing authentication keys to access points wherein each key is bound to the identity of the access point. *Sood*, 8:3-20. 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. *Id.*, 8:21-32; *Mandayam* ¶171. *Sood*'s access points, which meet the '671 patent's definition of an authenticator as explained in §VIII.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); *Mandayam* ¶171.

Lee teaches a key derivation method where, as in *Sood*, authentication keys are generated for multiple access points. *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: “PMKnext = PRF(RK, PMKcurr , STAmac, nextAPmac).” *Id.*, ¶[0081]. A POSITA would have understood that this key derivation binds the PMK for an access point (here labeled PMKnext) to the identity of that access point (here labeled nextAPmac). *Mandayam* ¶172.

Lee additionally discloses **the server** deriving each access point's PMK proactively to provide fast roaming. *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...”); *Mandayam* ¶173. *Lee* explains that prior systems required access points to have “large-capacity” memory

to store the numerous roaming keys required for numerous users. *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. *Id.*, ¶[0062]; *see also* ¶[0040].

It would have been obvious to a POSITA 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. *Mandayam* ¶174. A POSITA 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. *Mandayam* ¶174. The combination would also have been obvious to a POSITA 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). *Mandayam* ¶175. The combination would further have been obvious to a POSITA 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). *Mandayam* ¶176.

A POSITA 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." *Sood*, 4:32-35. *Lee* discloses that the "higher-layer server" which performs the key derivations is "an existing AAA server." *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 within the capabilities of a POSITA. *Mandayam* ¶177.

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." *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:

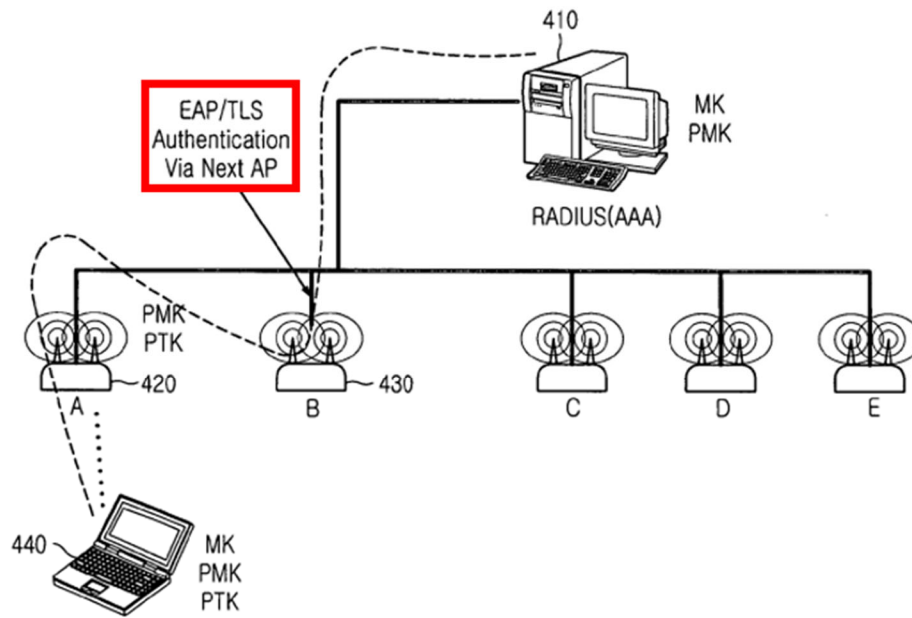


FIG. 4
(PRIOR ART)

Lee, FIG. 4 (annotated).

Thus, a POSITA 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. *Mandayam* ¶178.

A POSITA 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. *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

POSITA 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. *Mandayam* ¶179. Incorporating *Lee*'s teachings regarding access points automatically generating the AP-neighborhood graph into *Sood* would also have been obvious to a POSITA 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 having access points that automatically generate the AP-neighborhood graph). *Mandayam* ¶180. Additionally, a POSITA 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. *Mandayam* ¶181.

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:”

To the extent the preamble is limiting, *Sood* teaches this limitation for the reasons discussed above in §VIII.A.1.a (Element 1[pre]).

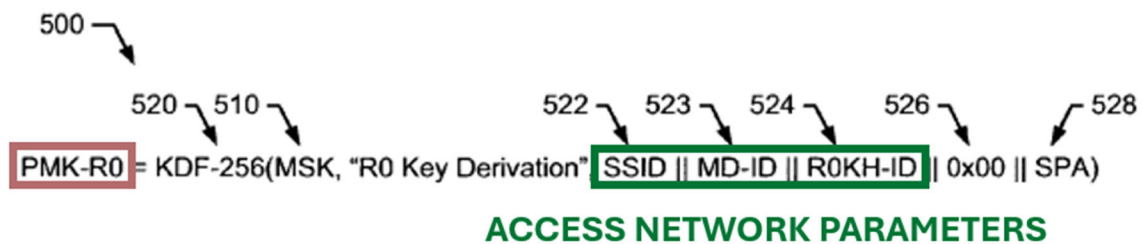
b. 6[a]

Sood-Lee teaches this feature. *Mandayam* ¶¶183-192.

- (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 ...”

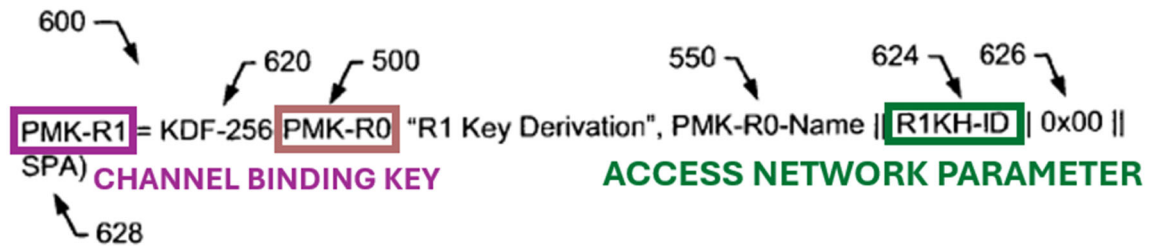
For the reasons discussed in §VIII.C.1 (Ground 3, Motivation to Combine), in the *Sood-Lee* combination the server generates PMK-R1 (“channel binding key”). That server is an “extensible authentication protocol server” because *Sood* discloses that its server implements “an authentication process such as Extensible Authentication Protocol-Transport Layer (EAP-TLS).” *Sood*, 6:59-65; *Mandayam* ¶184.

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:



Sood, FIG. 5 (excerpted and annotated).

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**:



Sood, FIG. 6 (excerpted and annotated).

As discussed in §VIII.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 as explained in §VIII.A.1.c (Element 1[b]). The access points of *Sood-Lee* are also “extensible authentication protocol authenticators” because, as explained in §VIII.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 ...”

Sood discloses that the access point (“extensible authentication protocol authenticator”) uses PMK-R1 (“channel binding key”) to generate a session key named PTK. *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 POSITA 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. *Mandayam* ¶188. 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.” *EAP*, 45. PMK-R1 is likewise used to generate PTK, which directly protects subsequent data. *Sood*, 9:17-22.

The ’671 Patent is consistent with the EAP specification, explaining 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)*.” ’671*Pat*, 12:43-50. A POSITA would have understood the term “bootstrap ciphersuits” refers to generating additional keys, not directly protecting data. *Mandayam* ¶190.

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

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 explained in §VIII.A.1.c (Element 1[b]), each of these

parameters are advertised from an authenticator. Advertised parameters need not be carried in authentication methods, as explained in §VIII.A.1.b (Element 1[a]).

* * *

Accordingly, the *Sood-Lee* combination teaches 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”). *Mandayam* ¶192.

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;”

Sood-Lee teaches this feature. *Mandayam* ¶¶193-194. As explained in §VIII.C.2.b (Element 6[a]), the server is used to derive the channel binding key. As explained in §VIII.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.”

Sood-Lee teaches this feature. *Mandayam* ¶¶195-196. This limitation mostly recites the lexicography of “key binding blob” and is disclosed by *Sood* for the same reasons given in §VIII.A.1.c (Element 1[b]). Additionally, the access points of *Sood-Lee* are “extensible authentication protocol authenticator[s]” as explained in §VIII.C.2.b (Element 6[a]).

3. Claim 10

Sood-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;”

Sood-Lee teaches this feature. *Mandayam* ¶¶198-205.

Sood-Lee discloses the server creating PMK-R1 (channel binding key) as explained in §VIII.C.2.b (Element 6[a]). *Sood* also discloses that the subscriber station “via a supplicant” generates an MSK and later “generate[s] a session key.”

Sood, 6:52-56, 7:49-56. Session keys (e.g., PTK) are generated from PMK-R1. *Sood*, 9:36-42. Thus, a POSITA would have understood that the subscriber station would generate PMK-R1 using its MSK as an intermediate step to generating a session key. *Mandayam* ¶200.

To the extent that PO argues and the Board determines that *Sood* alone does not teach this limitation, *Lee* explicitly discloses that the station generate[s] the corresponding PMK of an access point. *Lee*, ¶[0083] (“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_AP_B.***”). 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. *Lee*, ¶[0102]; *Mandayam* ¶203.

A POSITA 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. *Mandayam* ¶204. Accordingly, a POSITA would have found the combination obvious because it amounted to applying a known technique (*Lee* teaches 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). *Mandayam* ¶204.

Accordingly, the *Sood-Lee* combination teaches 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”). *Mandayam* ¶205.

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

Sood-Lee teaches the server transferring PMK-R1 (“channel binding key”) to an access point (“authenticator”) for the reasons discussed in §VIII.C.2.b (Element 6[a]).

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

Sood-Lee teaches this feature. *Mandayam* ¶¶207-209.

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).” *Sood*, 8:65-9:9. These session keys are, for example “a pairwise temporal key.” *Sood*, 9:5-7. A POSITA would have understood that using pairwise keys for communication requires the

keys to be identical. *Mandayam* ¶208 (citing *802.11i*, 19). Additionally, deriving identical pairwise keys requires using the same base PMK-R1, as confirmed by *Sood*'s disclosure that the PMK-R1s are “corresponding.” *Mandayam* ¶208; *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 §VIII.A.1.a (Element 1[pre]).

Accordingly, the *Sood-Lee* combination teaches 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”). *Mandayam* ¶209.

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.”**

Sood-Lee teaches this claim, including under *PO's Potential Alternative Construction*. *Mandayam* ¶¶210-214. 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 explained in §VIII.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). *Sood*, 8:27-32.

Accordingly, a POSITA 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”). *Mandayam* ¶212.

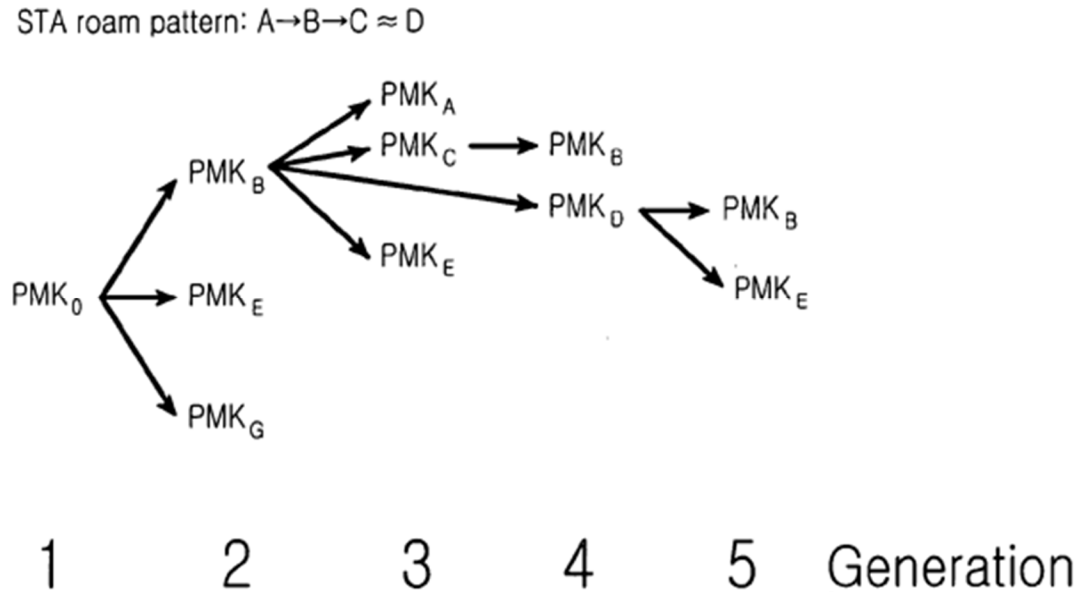
As also explained in §VIII.A.1.c (Element 1[b]), if the claim requires that the key binding blob be bound to the channel binding master key, the key binding blob is “SSID || MD-ID || R0KH-ID || 0x00 || SPA.” This key binding blob is also associated with the authenticator because it contains “SSID” and “MD-ID.” A POSITA 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. *Mandayam* ¶213; *see Sood* 7:5-10.

Accordingly, the *Sood-Lee* combination teaches a key binding blob containing the identity of the service set and the mobility domain of the associated access point (“associated with the authenticator”). *Mandayam* ¶214.

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

Sood in combination with *Lee* teaches these claims. Claim 13 is taught even under *PO’s Potential Alternative Construction*. *Mandayam* ¶¶215-220.

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



Lee, FIG. 11.

Lee’s figure 11 illustrates that, for example, the PMK associated with Access Point C may be derived from the PMK associated with Access Point B. 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.”); Mandayam ¶216. A POSITA would have understood this disclosure to describe using channel binding keys to form a hierarchical channel binding. Mandayam ¶216.

A POSITA would have found it obvious to implement Lee’s hierarchical key derivations in Sood because Sood itself suggests the combination. Sood, 11:50-53

(“the methods and apparatus disclosed herein may include additional levels of authentication keys.”).

Additionally, a POSITA 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. *Mandayam* ¶218. A POSITA 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. *Id.* The modification would also have been obvious to a POSITA 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). *Mandayam* ¶219.

Accordingly, the *Sood-Lee* combination teaches 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”). *Mandayam* ¶220.

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).”**

Sood in combination with *Lee* teaches this claim, including under *PO*'s *Potential Alternative Construction*, as explained in §VIII.C.2.b (Element 6[a]).

D. GROUND 4: *SOOD* IN COMBINATION WITH *ABOBA* AND *LEE* RENDERS OBVIOUS CLAIMS 1-8 and 10-19

The limitations of claims 1-8 and 10-19 are obvious for the reasons given in Grounds 1-3. However, to the extent that *PO* argues and the Board determines that one or both of the limitations discussed below is not taught for the reasons discussed in Ground 1, these limitations would have been obvious in view of the combination of *Sood* and *Aboba* (§VIII.D.1) or *Sood* and *Lee* (§VIII.D.2). Accordingly, a POSITA 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”

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-Suppliant Protocol (ASP)*.” '671Pat, 13:27-30. As explained in §VIII.A.1.c (Element 1[b]), *Sood* teaches this aspect of the

lexicography of “key binding blob.” However, to the extent that PO argues and the Board determines that *Sood* does not teach advertising parameters from an authenticator “using an authenticator-suppliant protocol,” this feature would have been obvious based on *Sood* in combination with *Aboba*.

Aboba discloses that the IEEE 802.11 protocol advertises parameters, including NAS identifiers. *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)**.”). *Aboba* confirms that its disclosures are compatible specifically with 802.11i. *Id.*, 11 (“EAP authentication can be run over ... **IEEE 802.11 wireless LANs [IEEE80211i]**”) (brackets in original). A POSITA would have understood that a “beacon” message is a message advertised by the authenticator. *Mandayam* ¶224.

A POSITA 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. *Compare Sood*, 8:3-6, 8:21-30 *with Aboba*, 70; *Mandayam* ¶225 (explaining that *Sood* and *Aboba* teach generating second-level keys in similar ways). Moreover, *Sood* teaches that its disclosures “may be applied to ... WLANs” and that WLAN is implemented

“in accordance with the 802.11 family of standards.” *Sood*, 11:56-58, 2:37-44. Thus, there is an explicit teaching in the prior art that would have lead a POSITA to implement *Sood* utilizing the features of 802.11i, as expressly taught in *Aboba*. *Mandayam* ¶226.

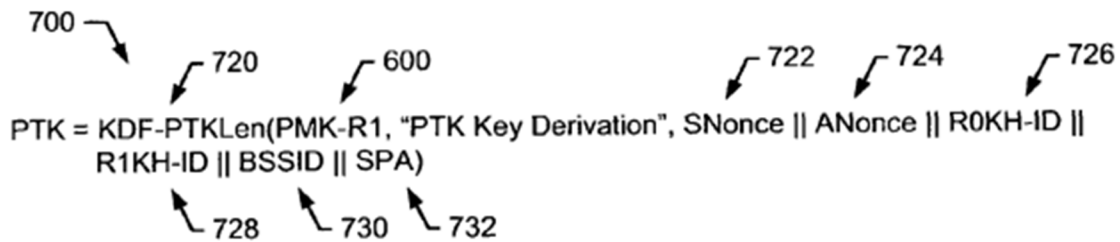
Furthermore, *Sood* already teaches that at least MD-ID and R1KH-ID are advertised, while it would have been well-known to a POSITA that SSIDs would be advertised. *Sood* 7:37-40, 8:27-36; *Mandayam* ¶227 (citing *802.11-1997*, 21). Advertising these parameters using IEEE 802.11i, as taught by *Aboba*, would have been obvious to a POSITA 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). *Mandayam* ¶227.

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

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 explained in §VIII.A.1.b (Element 1[a]), *Sood* teaches these limitations.

However, to the extent that PO argues and the Board finds that *Sood* alone does not teach this feature, this feature would have been obvious based on the combination of *Sood* and *Lee*.

In the *Sood-Lee* combination, there would have been no need to carry at least SSID, MD-ID, R0KH-ID and R1KH-ID (“access network parameters”) in authentication methods because, as explained in §VIII.C.2.b(i) (Element 6[a] Part i), the server would have already bound these values to PMK-R1. These bindings would have been performed by the server in the *Sood-Lee* combination, as explained in §VIII.C.1 (Ground 3, Motivation to Combine). 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:



Sood, FIG. 7.

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. *Mandayam* ¶232. A POSITA 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. *Mandayam* ¶232. 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. *Mandayam* ¶232; *see Sood*, 7:52-55 (indicating that the subscriber station generates the PTK).

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. '671*Pat*, 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. *Mandayam* ¶233; '671*Pat*, 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. *Mandayam* ¶233.

Accordingly, a POSITA 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. *Mandayam* ¶234.

IX. THE DISCRETIONARY FACTORS FAVOR INSTITUTING TRIAL

A. 35 U.S.C. §314(a)

To the extent PO asks the Board to exercise its discretion to deny institution despite the strong invalidity showing on the merits, the Board should decline to do so because the weight of the factors articulated in *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 at 6 (PTAB Mar. 20, 2020) (precedential) favors institution.⁷

1. Stay

Factor 1 is neutral. Petitioners have not requested a stay but intend to do so. The PTAB has explained that it will not speculate on how any such motion would be resolved, before one is filed. *Google LLC v. Parus Holdings, Inc.*, IPR2020-00847, Paper 9 at 12 (PTAB Oct. 21, 2020).

⁷ The *Fintiv* framework should not be followed because it is legally invalid as (1) exceeding the Director’s authority, (2) arbitrary and capricious, and (3) adopted without notice-and-comment rulemaking.

2. Trial Date

The district court in the parallel litigation held a scheduling conference on July 18, 2024, and set jury selection for February 9, 2026, approximately 13 months away. EX1021, 1. Any comparison of a projected FWD date against the scheduled trial date is speculative. *See Dish Network LLC v. Broadband iTV, Inc.*, IPR2020-01280, Paper 17 at 16 (PTAB Feb. 4, 2021) (“We cannot ignore the fact that the currently scheduled trial date is more than nine months away and much can change during this time”).

Setting that aside, this Petition is filed on January 22, 2025, so a FWD would be expected by July 2026. Although this is approximately five months after the currently scheduled (speculative) trial date, this factor is not determinative or considered in isolation. *Facebook, Inc. v. USC IP P’Ship*, IPR2021-00034, Paper 13 at 11 (PTAB Apr. 13, 2021) (“[T]his factor is not considered in isolation, but holistically along with other factors”) (citation omitted). Moreover, the Board has instituted IPR and found, on similar facts, that this factor weighs only minimally in favor of denial. *See, e.g., Google LLC, et al. v. Multimodal Media LLC*, IPR2024-00056, Paper 9 at 8 (PTAB Apr. 12, 2024) (a period of about six months between trial and the expected FWD “weighs only marginally in favor” of denial); *NetNut Ltd. v. Bright Data Ltd.*, IPR2021-01492, Paper 12 at 9-16 (PTAB Mar. 21, 2022) (instituting IPR without stipulation and copending trial date six months before

FWD); *Facebook*, IPR2021-00034, Paper 13 at 11 (a period of five months between trial and expected FWD “slightly favors denial”); *CoolIT Systems, Inc. v. Asetek Danmark A/S*, IPR2021-01195, Paper 10 at 11 (PTAB Dec. 28, 2021) (a period of five months between trial and expected FWD “weighs slightly in favor” of denial); *Equipmentsshare.com Inc. v. Ahern Rentals, Inc.*, IPR2021-00834, Paper 19 at 13 (PTAB Nov. 16, 2021) (a period of seven months between trial and expected FWD “weighs somewhat in favor” of denial).

3. Diligence/Investment

Factor 3 weighs strongly against discretionary denial. No substantive orders have been issued by the court in the underlying litigation, and investment in the parallel litigation will have remained low at the time of institution. *See, e.g., Hulu, LLC v. SITO Mobile R&D IP, LLC*, IPR2021-00298, Paper 11 at 12-14 (PTAB May 19, 2021) (holding that this factor supports instituting IPR given the early stage of the district court proceedings, the lack of substantial discovery related to invalidity claims, and the petitioner’s diligent filing of the petition after receiving preliminary infringement contentions). Assuming that a Decision on Institution is issued by July 2025, much work in district court will remain. The *Markman* hearing is scheduled for August 11, 2025, opening expert reports are due September 24, 2025, and dispositive motions are due November 3, 2025. EX2021, 3-4.

4. Overlap

Factor 4 weighs against discretionary denial. Petitioners challenge claims 1-8 and 10-19, whereas Patent Owner has alleged infringement of only claims 1-4, 6-8, 10, 11, 18, and 19 (EX1022, 2), and will likely reduce the number of asserted claims before trial. Accordingly, a material number of the challenged claims will not be addressed by the district court. *See, e.g., Precision Planting LLC v. Maschio Gaspardo S.p.A.*, IPR2024-00008, Paper 12 at 18 (PTAB Mar. 26, 2024) (holding that despite substantial overlap in issues between the IPR petition and parallel district court action, the inclusion of claims in the petition not contested in court argues against discretionary denial due to incomplete overlap).

5. Parties

Petitioners and PO are also parties to the parallel litigation.

6. Other considerations.

The merits are compelling here. The Office did not consider a wealth of prior art during prosecution that discloses and/or suggests the alleged invention.

Petitioners therefore respectfully submit that the *Fintiv* factors favor institution and that discretionary denial of this Petition would be neither appropriate nor equitable.

B. 35 U.S.C. §325(d)

The Board should likewise not exercise its discretion under §325(d) to deny institution of Petitioner’s petition. There are no references presented in the unpatentability grounds that were previously considered by the Office.

X. MANDATORY NOTICES UNDER 37 C.F.R. §42.8

A. Real Parties-in-Interest

Petitioners identify themselves as the real parties-in-interest.

1. Related Matters

To the best of Petitioners’ knowledge, the ’671 patent has only been involved in the following district court litigation: *Four Batons Wireless, LLC v. Samsung Electronics Co., Ltd. et al.*, 2:24-cv-00284 (E.D. Tex.), filed April 26, 2024 (“District Court Litigation”).

To the best of Petitioners’ knowledge, the ’671 patent has not been challenged in any other *inter partes* review or post-grant review prior to this proceeding.

B. Lead and Backup Counsel

Lead Counsel	Back-Up Counsel
<p>Ali R. Sharifahmadian (Reg. No. 48,202) ali.sharifahmadian@arnoldporter.com</p> <p>Arnold & Porter Kaye Scholer LLP 601 Massachusetts Ave., NW Washington, DC 20001-3743 Tel: 202-942-5000 Fax: 202-942-5999</p>	<p>Jeffrey Miller (Reg. No. 35,287) jeffrey.miller@arnoldporter.com David Caine (Reg. No. 52,683) david.caine@arnoldporter.com</p> <p>Arnold & Porter Kaye Scholer LLP 3000 El Camino Real Five Palo Alto Square, Suite 500 Palo Alto, CA 94306-3807 Tel: 650-319-4500 Fax: 650-319-4700</p> <p>Kevin Cosgrove (pro hac vice to be filed) kevin.cosgrove@arnoldporter.com</p> <p>Arnold & Porter Kaye Scholer LLP Three Embarcadero Center 10th Floor San Francisco, CA 94111-4024 Tel: 415-471-3100 Fax: 415-471-3400</p>

C. Service Information

Please address all correspondence to lead and back-up counsel at the addresses shown above. Petitioners consent to electronic service by email at the following addresses:

ali.sharifahmadian@arnoldporter.com
jeffrey.miller@arnoldporter.com
david.caine@arnoldporter.com
kevin.cosgrove@arnoldporter.com

xSamsungFourBatonsAP@arnoldporter.com

D. Power of Attorney

A power of attorney is filed herewith according to 37 C.F.R. §42.10(b).

XI. FEES

Petitioners concurrently electronically submits the required fees for this Petition. The Board is authorized to charge Arnold & Porter Kaye Scholer LLP's deposit account, No. 50-2387, for any fee deficiency.

Date: January 24, 2025

Respectfully submitted,

/Ali R. Sharifahmadian/

Ali R. Sharifahmadian (Reg. No. 48,202)
Counsel for Petitioners

CERTIFICATE OF COMPLIANCE

The undersigned hereby certifies that the foregoing Petition for *Inter Partes* Review contains 13,894 words, excluding those portions identified in 37 C.F.R. §42.24(a), as measured by the word-processing system used to prepare this paper.

/Ali R. Sharifahmadian/

Ali R. Sharifahmadian (Reg. No. 48,202)
Counsel for Petitioners

CERTIFICATE OF SERVICE

I certify that on January 24, 2025, I caused a true and correct copy of the foregoing Petition for *Inter Partes* Review of U.S. Patent No. 8,239,671 and supporting exhibits to be served via overnight delivery on the Patent Owner at the following correspondence address of record as listed on PAIR:

Stephen B Parker
Westerman, Hattori, Daniels & Adrian, LLP
8500 Leesburg Pike
SUITE 7500
Tysons, VA 22182-2435

A courtesy copy was also sent via electronic mail to Patent Owner's litigation counsel listed below:

Allen Hernandez (ahernandez@susmangodfrey.com)
Joseph Grinstein (jgrinstein@susmangodfrey.com)
Rohit Nath (rnath@susmangodfrey.com)
Shawn Blackburn (sblackburn@susmangodfrey.com)
Meng Xi (mxi@susmangodfrey.com)
Corey Lipschutz (clipschutz@susmangodfrey.com)
Andrew Leigh Fair (andrea@millerfairhenry.com)

/Ali R. Sharifahmadian/
Ali R. Sharifahmadian (Reg. No. 48,202)
Counsel for Petitioners