

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

APPLE INC.,

Petitioner

v.

HEADWATER PARTNERS I LLC,

Patent Owner

Inter Partes Review Case No. IPR2025-01572

U.S. Patent No. 10,064,055

DECLARATION OF DR. CHARLES D. CREUSERE, PH. D.

TABLE OF CONTENTS

I.	INTRODUCTION.....	8
A.	EDUCATIONAL BACKGROUND AND PROFESSIONAL EXPERIENCE	8
B.	MATERIALS CONSIDERED	12
II.	OVERVIEW AND LEGAL STANDARDS.....	16
A.	PERSON OF ORDINARY SKILL IN THE ART	17
B.	OBVIOUSNESS	17
C.	CLAIM CONSTRUCTION	22
III.	OVERVIEW OF THE '055 PATENT	23
A.	SUMMARY OF THE '055 PATENT.....	23
B.	FIELD OF ENDEAVOR.....	24
C.	PROBLEM SOLVED BY THE INVENTORS OF THE '055 PATENT	24
D.	FILE HISTORY OF THE '055 PATENT	25
IV.	LEVEL OF A PERSON OF ORDINARY SKILL.....	26
V.	OVERVIEW OF THE TECHNOLOGY	28
A.	DEVICE AND SOFTWARE AUTHENTICATION TO MAINTAIN NETWORK SECURITY	28
B.	DEVICE AUTHENTICATION FOR NETWORK ACCESS.....	32
C.	APPLICATION AUTHENTICATION FOR NETWORK ACCESS.....	36
VI.	SUMMARY OF THE PRIOR ART REFERENCES	39
A.	LUNDBLADE.....	39
B.	JOBST.....	46
C.	HARDJONO.....	50
D.	ZUCCHERATO.....	57
E.	CHEN	61
F.	LUNDBLADE '330.....	63
G.	FOSBACK	66
H.	SCARLATA	69
VII.	SUMMARY OF UNPATENTABILITY	71

VIII. GROUND 1: CLAIMS 1-2, 6-11, 13-15, AND 18-19 ARE OBVIOUS OVER LUNDBLADE, JOBST, AND HARDJONO	72
A. CLAIM 1	72
1. 1[Pre]: “A method of operating a network system, the method comprising:”	72
2. 1[a]: “receiving a first request from any first device in a plurality of similar wireless end-user devices, the first request for a secure device credential”	75
3. 1[b]: “performing a verification step of one or more device identifiers supplied by the first device, and, upon a successful verification in the verification step,”	82
a. Jobst’s Teachings	83
b. Motivation to Combine Lundblade and Jobst	85
4. 1[c]: “generating a secure device credential for the first device,”	88
5. 1[d]: “associating the secure device credential with one or more device identifiers of the first device, and”	90
a. Jobst’s Teachings	91
b. Motivation to Combine Lundblade and Jobst	92
6. 1[e]: “securely sending the secure device credential to the wireless end-user device;”	97
7. 1[f]: “based at least in part on the secure device credential, negotiating a secure message link between a message link server in the network system and a device link agent in the first device;”	102
8. 1[g]: “receiving, over the secure message link, a second request from the first device, the second request for a secured application credential, for any first application identified from a plurality of device applications registered to use wireless network communications, the second request comprising a general application credential for the first application; and”	110
a. Hardjono’s Teachings	113
b. Combination of Lundblade and Hardjono.....	123
c. Motivation to Combine Lundblade and Hardjono	127

9.	<i>1[h]: “in response to the second request, generating, based on the identified first application and the secure device credential, a first secured application credential unique to the first device, and”</i>	134
10.	<i>1[i]: “sending the first secured application credential to the first device over the secure message link.”</i>	137
B.	CLAIM 2: “THE METHOD OF CLAIM 1, WHEREIN GENERATING THE FIRST SECURED APPLICATION CREDENTIAL COMPRISES CREATING A COMBINATION OF THE GENERAL APPLICATION CREDENTIAL AND THE SECURE DEVICE CREDENTIAL.”	138
C.	CLAIM 6: “THE METHOD OF CLAIM 1, WHEREIN THE FIRST SECURED APPLICATION CREDENTIAL IS A RUN-TIME APPLICATION CREDENTIAL.”	139
D.	CLAIM 7: “THE METHOD OF CLAIM 6, FURTHER COMPRISING RECEIVING, FROM THE DEVICE LINK AGENT OVER THE SECURE MESSAGE LINK, THE RUN-TIME APPLICATION CREDENTIAL, AND EVALUATING THE RUN-TIME APPLICATION CREDENTIAL AT AN ELEMENT OF THE NETWORK SYSTEM.”	141
E.	CLAIM 8: “THE METHOD OF CLAIM 7, FURTHER COMPRISING, IN RESPONSE TO EVALUATING THE RUN-TIME APPLICATION CREDENTIAL, RETURNING A NETWORK POLICY SETTING OVER THE SECURE MESSAGE LINK TO THE DEVICE LINK AGENT, THE NETWORK POLICY SETTING APPLICABLE TO THE FIRST APPLICATION.”	143
F.	CLAIM 9: “THE METHOD OF CLAIM 7, WHEREIN THE ELEMENT OF THE NETWORK SYSTEM IS AN AUTHENTICATION SERVER, THE NETWORK SYSTEM FURTHER COMPRISING AN APPLICATION CREDENTIAL DATABASE COUPLED TO THE AUTHENTICATION SERVER AND STORING SECURED APPLICATION CREDENTIALS, INCLUDING THE FIRST SECURED APPLICATION CREDENTIAL.”	149
G.	CLAIM 10: “THE METHOD OF CLAIM 1, FURTHER COMPRISING BASED ON A FIRST DEVICE EVENT, REFRESHING THE FIRST SECURED APPLICATION CREDENTIAL TO CREATE A DIFFERENT SECURED APPLICATION CREDENTIAL UNIQUE TO THE FIRST DEVICE, AND SENDING THE DIFFERENT SECURED APPLICATION CREDENTIAL TO THE FIRST DEVICE, THE DIFFERENT CREDENTIAL SUPERSEDING THE FIRST SECURED APPLICATION CREDENTIAL.”	152

H.	CLAIM 11: “THE METHOD OF CLAIM 1, FURTHER COMPRISING, IN RESPONSE TO THE SECOND REQUEST, VERIFYING THAT THE GENERAL APPLICATION CREDENTIAL MATCHES A KNOWN-APPLICATION CREDENTIAL FOR THE FIRST APPLICATION.”	155
I.	CLAIM 13: “THE METHOD OF CLAIM 11, FURTHER COMPRISING OBTAINING THE KNOWN-APPLICATION CREDENTIAL FROM AN APP STORE.”	157
J.	CLAIM 14: “THE METHOD OF CLAIM 1, THE VERIFICATION STEP FURTHER COMPRISING EVALUATING A SUBSCRIBER IDENTIFIER ASSOCIATED WITH ONE OR MORE WIRELESS END-USER DEVICES, INCLUDING THE FIRST DEVICE.”	158
K.	CLAIM 15: “THE METHOD OF CLAIM 14, FURTHER COMPRISING GENERATING THE SECURE DEVICE CREDENTIAL BASED AT LEAST IN PART ON ONE OR MORE OF THE DEVICE IDENTIFIERS AND ALSO ON THE SUBSCRIBER IDENTIFIER.”	160
L.	CLAIM 18: “THE METHOD OF CLAIM 1, WHEREIN A FIRST SUBSCRIBER IS ASSOCIATED WITH BOTH THE FIRST DEVICE AND WITH A SECOND WIRELESS END-USER DEVICE, THE METHOD FURTHER COMPRISING THE NETWORK SYSTEM GENERATING, BASED ON THE IDENTIFIED FIRST APPLICATION AND A SECOND SECURE DEVICE CREDENTIAL ASSOCIATED WITH THE SECOND DEVICE, A SECOND SECURED APPLICATION CREDENTIAL UNIQUE TO THE SECOND DEVICE.”	160
M.	CLAIM 19: “THE METHOD OF CLAIM 2, WHEREIN THE COMBINATION COMPRISES A HASH.”	164
IX.	GROUND 2: CLAIM 3 IS OBVIOUS OVER LUNBLADE, JOBST, HARDJONO, AND ZUCCHERATO.....	164
A.	CLAIM 3: “THE METHOD OF CLAIM 1, FURTHER COMPRISING, IN RESPONSE TO RECEIVING THE FIRST REQUEST AND PRIOR TO PERFORMING THE VERIFICATION STEP, RETURNING TO THE FIRST DEVICE A NETWORK SYSTEM CREDENTIAL THAT IS VERIFIABLE THROUGH A TRUSTED CERTIFICATE AUTHORITY.”	164
	a. Zuccherato’s Teachings	165
	b. Motivation to Combine with Zuccherato	167
X.	GROUND 3: CLAIM 4 IS OBVIOUS OVER LUNDBLADE, JOBST, HARDJONO, AND CHEN.....	171

A.	CLAIM 4: “THE METHOD OF CLAIM 1, WHEREIN GENERATING A SECURE DEVICE CREDENTIAL FOR THE FIRST DEVICE COMPRISES ENCRYPTING THE CREDENTIAL WITH A KEY KNOWN BY AT LEAST ONE ELEMENT OF THE NETWORK SYSTEM.”	171
a.	Chen’s Teachings	171
b.	Motivation to Combine with Chen.....	173
XI.	GROUND 4: CLAIMS 5 AND 16 ARE OBVIOUS OVER LUNDBLADE, JOBST, HARDJONO, AND LUNDBLADE ’330.....	176
A.	CLAIM 5: “THE METHOD OF CLAIM 1, FURTHER COMPRISING ROUTING A NETWORK MESSAGE TO THE IDENTIFIED FIRST APPLICATION OVER THE SECURE MESSAGE LINK, USING THE FIRST SECURED APPLICATION CREDENTIAL TO INDICATE THAT THE IDENTIFIED FIRST APPLICATION IS TO RECEIVE THE NETWORK MESSAGE.”	176
a.	Lundblade ’330’s Teachings	177
b.	Motivation to combine with Lundblade ’330.....	180
B.	CLAIM 16: “THE METHOD OF CLAIM 1, FURTHER COMPRISING FORWARDING NETWORK MESSAGES TO THE FIRST APPLICATION ON THE FIRST DEVICE OVER THE SECURE MESSAGE LINK, BASED ON THE FIRST SECURED APPLICATION CREDENTIAL.”	183
XII.	GROUND 5: CLAIM 12 IS OBVIOUS OVER LUNDBLADE, JOBST, HARDJONO, AND FOSBACK	183
A.	CLAIM 12: “THE METHOD OF CLAIM 11, WHEREIN THE KNOWN-APPLICATION CREDENTIAL IS UPLOADED TO THE NETWORK SYSTEM VIA AN APPLICATION DEVELOPER SERVICE DESIGN CENTER.”	183
a.	Fosback’s Teachings	184
b.	Motivation to combine with Fosback.....	186
XIII.	GROUND 6: CLAIM 17 IS OBVIOUS OVER LUNDBLADE, JOBST, HARDJONO, AND SCARLATA	189
A.	CLAIM 17: “THE METHOD OF CLAIM 1, WHEREIN THE SECURE DEVICE CREDENTIAL IS A CREDENTIAL ASSOCIATED WITH A SERVICE PROCESSOR ON THE FIRST DEVICE, THE METHOD FURTHER COMPRISING CONFIRMING THE IDENTITY OF THE SERVICE PROCESSOR BASED AT LEAST IN PART ON THE SECURE DEVICE CREDENTIAL.”	189

a.	Scarlata's Teachings.....	190
b.	Motivation to Combine with Scarlata	194
XIV.	CONCLUSION.....	199

I, Dr. Charles D. Creusere, declare as follows:

I. INTRODUCTION

1. I have been retained by counsel for Petitioner as a technical expert in the above-captioned case. Specifically, I have been asked to render certain opinions regarding the IPR petition with respect to U.S. Patent No. 10,064,055 (the “’055 Patent”). I understand that the Challenged Claims are 1-19, and my opinions are limited to those claims. A true and correct copy of my Curriculum Vitae, which provides further details about my background and experience, is appended to this Declaration.

A. Educational Background and Professional Experience

2. I received a Bachelor of Science degree in Electrical and Computer Engineering from the University of California at Davis in 1985. I received a Master of Science degree in Electrical and Computer Engineering from the University of California at Santa Barbara in 1990, and I received my Ph.D. in Electrical and Computer Engineering, also from the University of California at Santa Barbara, in 1993.

3. I am currently a Full Professor in the Klipsch School of Electrical & Computer Engineering at New Mexico State University, and I held the Frank Carden Endowed Chair in Telemetry and Telecommunications from 2010 to 2024. I was an Assistant Professor at New Mexico State from January 2000 until I became an

Associate Professor in 2004. I have been a Full Professor since August 2010. My research and teaching at New Mexico State have focused on digital signal processing and communications.

4. I have extensive experience in the technical areas of the '055 Patent, including more than 30 years of experience with digital systems and communications theory.

5. After receiving my B.S. from U.C. Davis in 1985, I went to work for the Naval Weapons Center, China Lake as a civilian Department of Defense (DoD) employee, where I was the lead designer for the guidance electronics of the Laser Guided Training Round. My work on this project included analog and digital circuit design, embedded software design, and front-end systems integration and testing. I was awarded a fellowship from the DoD to pursue graduate degrees at the University of California, Santa Barbara. During my Masters program at UCSB, I pursued a dual communications/digital signal processing coursework specialization. Several classes I took have relevance to the patent in suit, but ECE243 (Digital Communications) is particularly pertinent here given that digital communication is a critical part of any wireless network system.

6. While the focus of my Ph.D. research was on multirate filter banks and signal compression, I nonetheless followed the communications field closely during the six years after receiving my Ph.D. when I was a civilian working for the

department of defense at the Naval Air Warfare Center (now NAVAIR), China Lake, from 1993 through 1999. I was a Researcher and Team Leader at NAVAIR. As part of my work there, I developed efficient video compression and decompression algorithms for use with wireless communication channels. This work was presented at two conferences (the 1996 SPIE Conference and the 1996 Data Compression Conference) and it was published in the October 1997 issue of IEEE Transactions on Image Processing. In addition, the complete real-time internet streaming system was documented in NAWCWD Technical Paper, TP 8442.

7. Since joining the faculty of New Mexico State University in 2000, I have taught numerous classes at both the graduate and undergraduate levels. Many of the classes I have taught are relevant to the '055 Patent. These include EE497 (Digital Communications), EE573 (Signal Compression), and EE586 (Information Theory). In the course of teaching these various classes, I cover the breadth of the theory and practice of modern digital communication system—from data conditioning and modulation to receiver synchronization and network structures.

8. I have also done research in areas related to communications systems during my tenure at NMSU. The first was for a 2006 DARPA-funded project titled “ADAM Project” that we executed in collaboration with partners at Los Alamos National Laboratories. As part of this project, we implemented in the Mathwork’s Simulink environment a complete simulation of a satellite to ground

communications system to model the effects of specific compressive-sensing related processing techniques on signal recovery. From 2011-2014, I was a co-PI on a NASA-funded project entitled “Proximity Operations for Near Earth Asteroid Exploration.” My portion of the project dealt with the communications between the spacecraft orbiting the asteroid and multiple probes on its surface—specifically on how one might use the communications waveform along with orbital prediction to estimate relative location of the various parties.

9. More recently, I worked with the Air Force Research Lab to design a software-defined radio capable of fingerprinting satellites from the electronic signature of their radio. I have also done research involving communications at the network level, including assisting in Dr. Justin MacDonald, a professor at NMSU in the department of psychology, with his NIH-funded project on Gaze-Directed Hearing Assistance/Augmentation as well as leading an internal project studying the feedback dynamics of brain computer interfaces in a distributed processing computing environment. Both of these projects involved real-time network communications between multiple programs that were running simultaneously on multiple networked computer systems.

10. From 1990 through 1993, I worked as a Research Assistant in the Department of Electrical and Computer Engineering at the University of California, Santa Barbara. In this position, I worked on subband coding (compression) and

multirate filter bank theory. I also implemented real-time filter banks on a digital signal processor. In the summer of 1992, I worked at AT&T Bell labs where I developed and simulated new methods of extremely low bit rate video coding for video telephone applications.

11. A listing of the cases (including trials before the Patent Trial and Appeal Board) in which I have testified within the last four years is found following my CV in the attached Appendix A.

B. Materials Considered

12. I have relied upon my education, knowledge, and experience with data communications, as well as the other materials as discussed in this declaration in forming my opinions.

13. In developing my opinions, I have considered the following materials:

Exhibit 1001	U.S. Patent No. 10,064,055 (the “’055 Patent”)
Exhibit 1002	File History of the ’055 Patent (the “’055 Patent File History”)
Exhibit 1004	U.S. Patent No. 7,784,089 to Lundblade et al. (“Lundblade”)
Exhibit 1005	U.S. Patent No. 6,707,915 to Jobst et al. (“Jobst”)
Exhibit 1006	U.S. Patent Application Publication No. 2007/0143629 to Hardjono et al. (“Hardjono”)
Exhibit 1007	U.S. Patent Application Publication No. 2003/0014629 to Zuccherato (“Zuccherato”)
Exhibit 1008	U.S. Patent No. 7,958,544 to Chen et al. (“Chen”)
Exhibit 1009	U.S. Patent Application Publication No. 2005/097330 to Lundblade (“Lundblade ’330”)
Exhibit 1010	U.S. Patent Application Publication No. 2009/276433 to Fosback et al. (“Fosback”)
Exhibit 1011	U.S. Patent No. 7,587,595 to Scarlata et al. (“Scarlata”)
Exhibit 1012	U.S. Patent No. 7,203,967 to Chmaytelli et al. (“Chmaytelli”)
Exhibit 1013	U.S. Patent No. 6,980,660 to Hind et al. (“Hind”)

Exhibit 1014	U.S. Patent Application Publication No. 2004/0039924 to Baldwin et al. (“Baldwin”)
Exhibit 1015	U.S. Patent Application Publication No. 2006/0002556 to Paul (“Paul”)
Exhibit 1016	U.S. Patent No. 9,143,494 to Giles et al. (“Giles”)
Exhibit 1017	U.S. Patent Application Publication No. 2008/0022103 to Brown et al. (“Brown”)
Exhibit 1018	U.S. Patent No. 8,166,296 to Buer et al. (“Buer”)
Exhibit 1019	Trusted Computing Group: Controlling Network Access and Endpoints (Nov. 2007) (“Trusted Computing Group”)
Exhibit 1020	U.S. Patent Application Publication No. 2008/0060066 to Wynn et al. (“Wynn”)
Exhibit 1021	U.S. Patent Application Publication No. 2008/0301776 to Weatherford (“Weatherford”)
Exhibit 1022	3GPP Technical Report, Version 6.2.0, Release 6 (March, 2005), (“3GPP”)
Exhibit 1023	International Patent Publication No. 01/82037 to Moore et al. (“Moore”)
Exhibit 1024	U.S. Patent Application Publication No. 2007/0198834 to Ksontini et al. (“Ksontini”)
Exhibit 1025	U.S. Patent Application Publication No. 2009/0328207 to Patel (“Patel”)
Exhibit 1026	U.S. Patent No. 8,201,251 to Ngalle (“Ngalle”)
Exhibit 1027	U.S. Patent No. 8,522,025 to Lakshmeshwar et al. (“Lakshmeshwar”)
Exhibit 1028	U.S. Patent Application Publication No. 2006/0101507 to Camenisch (“Camenisch”)
Exhibit 1029	U.S. Patent No. 5,978,833 to Pashley et al. (“Pashley”)
Exhibit 1030	U.S. Patent No. 5,935,247 to Pai et al. (“Pai”)
Exhibit 1031	U.S. Patent No. 7,984,487 to Katsube et al. (“Katsube”)
Exhibit 1032	International Patent Publication No. 2007/128134 to Matta et al. (“Matta”)
Exhibit 1033	U.S. Patent Application Publication No. 2002/0071567 to Kurn et al. (“Kurn”)
Exhibit 1034	3GPP Technical Specification, Version 5.7.0 (Release 5) (December, 2003), (“3GPP TS 24.229”)
Exhibit 1035	U.S. Patent Application Publication No. 2007/0255957 to Arnone et al. (“Arnone”)

Exhibit 1036	An Introduction to Computer Security – The NIST Handbook, https://csrc.nist.gov/publications/nistpubs/800-12/800-12-html/chapter19-printable.html (July, 2007), (“An Introduction to Computer Security – The NIST Handbook”)
Exhibit 1037	Secure Programming for Linux and Unix HOWTO, <i>David A. Wheeler</i> , Version 3.50 (2004) (“Secure Programming”)
Exhibit 1038	U.S. Patent No. 7,979,912 to Roka (“Roka”)
Exhibit 1039	U.S. Patent Application Publication No. 2007/0214370 to Sato et al. (“Sato”)
Exhibit 1040	Introduction to Programming Using Java, Version 3.1, https://math.hws.edu/eck/cs124/javanotes3/java2cpp/s2.html (June 2004), (“Introduction to Programming”)
Exhibit 1041	Cell-Phone Security Far From Airtight, <i>Wired</i> , https://www.wired.com/1998/04/cell-phone-security-far-from-airtight/ (April 13, 1998), (“Cellphone Security Far From Airtight”)
Exhibit 1042	U.S. patent No. 7,925,214 to Weinans et al. (“Weinans”)
Exhibit 1043	U.S. Patent No. 5,579,394 to Waldron Jr. et al. (“Waldron”)
Exhibit 1044	Telecommunications Law in the Internet Age, <i>Sharon K. Black</i> , Academic Press (2002) (“Telecommunications”)
Exhibit 1045	The TLS Protocol Version 1.0, Network Working Group (January 1999), (“TLS Protocol Version 1.0”)
Exhibit 1046	Hypertext Transfer Protocol – HTTP/1.1, Network Working Group (June 1999), (“Hypertext Transfer Protocol”)
Exhibit 1047	Digital Signatures, <i>Gareth Branwyn</i> , https://www.wired.com/1994/06/digital-signatures/ (1994) (“Digital Signatures”)
Exhibit 1048	TCG Specification Architecture Overview, Revision 1.4 (August 2, 2007), (“TCG Specification Architecture Overview”)
Exhibit 1049	Banking on Mobiles: Why, How, for Whom?, CGAP, No. 48, Mas et al. (June 2008), (“Banking on Mobiles”)
Exhibit 1050	SMS Banking Services: A 21 st Century Innovation in Banking Technology, Rotimi et al., Issues in Informing Science and Information Technology, Volume 4 (2007), (“SMS Banking Services”)
Exhibit 1051	Form 10-K Netscout Systems INC, 10kWizard SEC Power Search (June 5, 2006), (“Netscout Systems Inc”)
Exhibit 1052	U.S. Patent Application Publication No. 2007/0287433 to Wollersheim et al. (“Wollersheim”)

Exhibit 1053	Internet and Email Use for Work, Madden et al., Pew Research Center, https://www.pewresearch.org/internet/2008/09/24/internet-and-email-use-for-work/ (Sept. 24, 2008), (“Internet and Email Use for Work”)
Exhibit 1054	Microsoft Reveals New Windows Mobile 6 Smartphone Software, Improves World’s Fastest-Growing Mobile Operating System, Microsoft Source, https://news.microsoft.com/source/2007/02/11/microsoft-reveals-new-windows-mobile-6-smartphone-software-improves-worlds-fastest-growing-mobile-operating-sy%E2%80%A6 (Feb. 12, 2007), (“Microsoft Reveals New Windows 6 Smartphone Software”)
Exhibit 1055	International Patent Publication No. 2005/098630 to Wilson (“Wilson”)
Exhibit 1056	An Initial Security Analysis of the IEEE 802.1X Standard, Mishra et al. (Feb. 6, 2002), (“An Initial Security Analysis of the IEEE 802.1X Standard”)
Exhibit 1057	Direct Anonymous Attestation Explained, Jan Camenisch (July 2007) (“Attestation Explained”)
Exhibit 1058	Enterprise Security: Putting the TPM to Work, Trusted Computing Group (2008), (“Putting the TPM to Work”)
Exhibit 1059	U.S. Patent No. 9,112,681 to Kotani et al. (“Kotani”)
Exhibit 1060	A Security Assessment of Trusted Platform Modules, Evan Sparks, Computer Science Technical Report TR2007-597 (June 28, 2007), (“Security Assessment”)
Exhibit 1061	The Role of TPM in Enterprise Security, IBM Research Report, Sailer et al. (Oct. 6, 2004), (“The Role of TPM in Enterprise Security”)

14. I have considered these materials from the viewpoint of a POSITA as of the priority date of the ’055 Patent. For the purposes of this declaration, I have been asked to assume that the priority date of the ’055 Patent is January 28, 2009. I note that my opinions provided in this Declaration are made from the perspective of

a POSITA as of this priority date of the '055 Patent unless expressly stated otherwise. To the extent that I use any verb tense in this Declaration that is present tense (e.g., “a POSITA would understand” instead of “a POSITA would have understood”), such verb tense should be understood to be my opinion as of the '055 Patent’s priority date (again, unless expressly stated otherwise). I merely use the present verb tense for ease of reading.

II. OVERVIEW AND LEGAL STANDARDS

15. In formulating my opinions, I have been instructed to apply certain legal standards. I am not a lawyer. I do not offer any testimony regarding what the law is. Instead, the following sections summarize the law as I have been instructed to apply it in formulating and rendering my opinions found later in this declaration. I understand that, in an *inter partes* review (“IPR”) proceeding, patent claims may be deemed unpatentable if it is shown that they are anticipated or rendered obvious in view of the prior art. I understand that prior art in an IPR review is limited to patents or printed publications that predate the priority date of the patent at issue. I understand that questions of claim clarity (definiteness) and enablement cannot be considered as a ground for considering the patentability of a claim in these proceedings.

A. Person of Ordinary Skill in the Art

16. I understand that the '055 Patent, the record of the proceedings at the Patent Office (which I understand is called the "File History" or "Prosecution History"), and the teachings of the prior art are evaluated from the perspective of a person of ordinary skill in the art ("POSITA"). I understand that the factors considered in determining the ordinary level of skill in the art may include: (i) the levels of education of the inventor; (ii) the types of problems encountered in the art; (iii) prior art solutions to those problems; (iv) the rapidity with which innovations are made; (v) the sophistication of the technology; and (vi) the educational level of persons working in the field.

17. I understand that a person of ordinary skill in the art is not a specific real individual, but rather a hypothetical individual having the qualities reflected by the factors above. The hypothetical person is presumed to have the same level of skill as the typical practitioner of the art and is presumed to have knowledge of all prior art in the relevant field. I understand that the inventor's actual knowledge or lack of knowledge of prior art reference is irrelevant to the obviousness determination.

B. Obviousness

18. I understand that a claim may be invalid under 35 U.S.C. § 103(a) if the subject matter described by the claim as a whole would have been "obvious" to a

POSITA in view of a single or combination of prior art references at the time the claimed invention was made. I further understand that a POSITA is assumed to know and to have all relevant prior art in the field of endeavor covered by the patent-in-suit and all analogous prior art. I understand that obviousness in an IPR review proceeding is evaluated using a preponderance of the evidence standard, which means that the claims must be more likely obvious than nonobvious.

19. I also understand that an obviousness determination includes the consideration of various factors including: (1) the scope and content of the prior art, (2) the differences between the prior art and the claim at issue, and (3) the level of ordinary skill in the pertinent art. I understand that secondary considerations of non-obviousness such as commercial success, long-felt but unresolved needs, failure of others, and so forth may be assessed as well. I have been informed that an obviousness analysis must consider the full scope of the claims to avoid impermissibly using hindsight to invalidate a patent.

20. In considering whether certain prior art renders a particular patent claim obvious, I have been informed that I can consider the scope and content of the prior art, including the fact that a POSITA would regularly look to the disclosures in patents, trade publications, journal articles, conference papers, industry standards, product literature and documentation, texts describing competitive technologies,

requests for comment published by standard setting organizations, and materials from industry conferences, as examples.

21. I have been informed that for a prior art reference to be proper for use in an obviousness analysis, the reference must be “analogous art” to the claimed invention. A reference is analogous art if: (1) the reference is from the same field of endeavor as the claimed invention (even if it addresses a different problem); or (2) the reference is reasonably pertinent to the problem faced by the inventor (even if it is not in the same field of endeavor as the claimed invention). For a reference to be “reasonably pertinent” to the problem, it must logically have commended itself to an inventor’s attention in considering the problem. In determining whether a reference is reasonably pertinent, one should consider the problem faced by the inventor, as reflected either explicitly or implicitly, in the specification. I believe that the documents I considered in forming my opinions in this IPR are well within the range of documents a POSITA would have consulted to address the type of problems described in the Challenged Claims.

22. I have been informed that to establish that a claimed invention was obvious based on a combination of prior art elements, an articulation of the reason(s) why a claimed invention would have been obvious must be provided. Specifically, I have been informed that the prior art, either as a single reference or a combination of multiple items of prior art, renders a patent claim obvious when there was an

apparent reason for a POSITA, at the time of the invention, to combine or modify the prior art. Rationales for combining or modifying the prior art include, but are not limited to, any of the following: (A) combining prior art methods according to known methods to yield predictable results; (B) substituting one known element for another to obtain predictable results; (C) using a known technique to improve a similar device in the same way; (D) applying a known technique to a known device ready for improvement to yield predictable results; (E) trying a finite number of identified, predictable potential solutions, with a reasonable expectation of success; (F) identifying that known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces if the variations are predictable to one of ordinary skill in the art; or (G) identifying an explicit teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to modify the prior art reference or to combine the prior art references to arrive at the claimed invention.

23. I have also been informed that where there is a motivation to combine, claims may be rejected as obvious provided a POSITA would have had a reasonable expectation of success regarding the proposed combination. I have also been informed that common sense may be considered. Common sense teaches that familiar items may have obvious uses beyond their primary purposes. I have been informed that if the combination was obvious to try (regardless of whether it was

actually tried) or leads to anticipated success, then it is likely the result of ordinary skill and common sense rather than non-obvious innovation.

24. I have been informed that the existence of an explicit teaching, suggestion, or motivation to combine known elements of the prior art is a sufficient, but not a necessary, condition to a finding of obviousness. In determining whether the subject matter of a patent claim is obvious, neither the particular motivation nor the avowed purpose described in the patent-in-suit controls. I have been further informed that the obviousness analysis may consider the effects of demands known to the technological community or present in the marketplace and the background knowledge possessed by a POSITA. These issues may be considered to determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent.

25. I have been informed that it is improper to combine references where the references teach away from their combination. A reference may be said to teach away when a POSITA, upon reading the reference, would be discouraged from following the path set out in the reference, or would be led in a direction divergent from the path that was taken by the patent applicant. I have also been informed that a reference does not teach away if it merely expresses a general preference for an alternative invention but does not criticize, discredit, or otherwise discourage investigation into the invention claimed.

26. I am informed that even if a case of obviousness is established, the final determination of obviousness must also consider “secondary considerations” if presented. Secondary considerations include: (a) commercial success of a product due to the merits of the claimed invention; (b) a long-felt, but unsatisfied need for the invention; (c) failure of others to find the solution provided by the claimed invention; (d) deliberate copying of the invention by others; (e) unexpected results achieved by the invention; (f) praise of the invention by others skilled in the art; (g) lack of independent simultaneous invention within a comparatively short space of time; and (h) teaching away from the invention in the prior art.

27. I have been further informed that secondary considerations evidence is only relevant if the offering party establishes a connection, or nexus, between the evidence and the claimed invention. The nexus cannot be based on prior art features. The establishment of a nexus is a question of fact. While I understand that Patent Owner here has not offered any secondary considerations at this time, I will supplement my opinions should Patent Owner raise secondary considerations during the course of this proceeding.

C. Claim Construction

28. I understand that the claim terms in an IPR proceeding are construed according to their plain and ordinary meaning as understood in light of the claim language, the patent’s description, and the prosecution history viewed from the

perspective of a POSITA. I further understand that where a patent defines claim language, the definition of the patent controls, even if there are other definitions that might be understood by those working in the art.

III. OVERVIEW OF THE '055 PATENT

A. Summary of the '055 Patent

29. The '055 Patent purports improvements to the security of mobile devices and a network. EX1001 (*'055 Patent*), Abstract, 1:53-55. Security of mobile devices and a network is improved by credentialing. For example, the '055 Patent describes a method of providing a device credential to an end-user device 100 via a network element. EX1001, 46:47-47:10. The method includes sending a request from the end-user device 100, with the request including “a device identifier.” EX1001, 46:47-58, 38:25-28. The network verifies the device identifier in the request to “a list of valid devices that are allowed to connect” to the network. EX1001, 46:60-65. The network then generates a device credential for the end-user device 100 and stores the credential in database 116 as associated with the “device identifier associated with end-user device 100.” EX1001, 47:2-8, 38:36-40. The network then sends the device credential to the end-user device 100. EX1001, 47:8-10. The device credential may be a “certificate, a key, a shared secret, a password, a hash, a shared algorithm, or any other item of information that allows” the network to confirm the identity of the device. EX1001, 45:57-62, 56:59-62. A device

identifier is “(e.g., any identifier associated with the end-user device).” EX1001, 38:28-31.

30. The '055 Patent also purports improving security on the network and for the device by validating an “application installed on an end-user device is authentic.” EX1001, 56:54-56, 57:40-42. Some methods of validating the installed application include “generat[ing] a signed application credential using the application credential and a device credential.” EX1001, 57:50-53. The network then sends the signed application credential to the end-user device. EX1001, 57:56-57.

B. Field of Endeavor

31. I have been informed that the field of endeavor of the claimed invention can be determined by reference to explanations of the invention’s subject matter in the patent application, including the embodiments, function, and structure of the claimed invention.

32. The '055 Patent generally relates to the security of network elements, mobile devices, and downloaded software. EX1001, 1:53-55, Abstract. For these reasons, a POSITA would understand that the field of endeavor of the '055 Patent includes security of devices and/or software communicating data via a network.

C. Problem Solved by the Inventors of the '055 Patent

33. I have been informed that a prior art reference is “reasonably pertinent” if a POSITA would have consulted it and applied its teachings when faced with the

problem(s) that the inventor was trying to solve. As such, I have been asked to analyze the '055 Patent and determine the problem(s) that the inventors were trying to solve.

34. The '055 Patent recognizes problems caused by device users attempting to, “or to successfully, spoof or hack end-user device components to fraudulently obtain access to data services” or “gain access to network elements.” EX1001, 1:45-51, 1:55-57. Additionally, the '055 Patent recognizes the need to ensure the authenticity of applications to ensure the application is not fraudulent. EX1001, 57:40-42, 56:54-56, 25:49-52. Thus, a POSITA would have understood the '055 Patent is directed to solving the problems of decreasing fraudulence at a device, application, and/or on a network.

D. File History of the '055 Patent

35. The '055 Application was issued a Non-final Office Action that rejected the claims filed in the preliminary amendment under 35 U.S.C. 112(a) as “failing to comply with the written description requirement.” EX1002 (*'055 Patent File History*), 563-565. Additionally, the Non-final Office Action rejected the claims under 103 based on “Lortz et al., US 8,146,142.” EX1002, 565-566. The Office Action also stated that “different application credentials as recited by Applicant used in a secure communication network has been common knowledge in the network

system art,” and accordingly Lortz renders obvious the claimed general application credential and secured application credential. EX1002, 565-566.

36. The Applicant did not amend the claims but only addressed the written description rejection by pointing to support in the '055 specification and argued Lortz does not teach claim 2 because “Lortz focuses on an identification of an ‘enrollee’ and thus does not ‘associate the secure device credential with one or more device identifiers of the first device,’ as claimed.” EX1002, 727-731. The Applicant also argued Lortz fails to teach the secured application credential generating step and the second request step of claim 1 because Lortz’s “application merely seeks and receives configuration data and keys for its own secure sessions.” EX1002, 730-731. The Examiner then issued a Notice of Allowance stating, “Claims 2-20 are allowed as all the cited prior art, either alone or in combination, does not disclose Applicant’s inventive claim language pursuant to Applicant’s May 6, 2018 Amendment and REMARKS.” EX1002, 990.

IV. LEVEL OF A PERSON OF ORDINARY SKILL

37. Based on my review and analysis of the '055 Patent, the prior art cited herein, and the ordinary skill factors described in this section, a POSITA in the field of the '055 Patent at the time of the earliest possible priority date (January 28, 2009) would have been knowledgeable regarding the field of data communications. In my experience working in this field, most workers of ordinary skill in the art as of the

earliest possible priority date of January 28, 2009, would have had a master's degree in computer engineering, computer science, electrical engineering, or a related field, with at least two years of experience in the field of data communications, or a bachelor's degree in the same field with at least four years of experience in the field of data communications. Additional education or experience might substitute for the above requirements. *See, e.g.,* EX1001 ('055 *Patent*), 1:44-57, Abstract, 4:23-25. When I refer to the understanding of a POSITA, I am referring to the understanding of such a person as of January 28, 2009.

38. As of January 28, 2009, I had more than ordinary skill in the art. I am, however, familiar with the skills and knowledge possessed by those I would have considered to be of ordinary skill in the art as of that date. In the late 1990s, for example, my team and I at China Lake implemented a demonstration version of our wavelet-based video communications system that used network communications (TCP/IP and UDP over the Internet) to transfer the compressed video in real-time from the encoder to the decoder and to send camera control signals back the other way. Later on after arriving at NMSU, I worked with a number of people on various projects where we used network IP protocols to facilitate the real-time transmission of data between different programs running on multiple networked computers. In addition, I also worked closely with colleagues at NMSU whose research specialties

focused on secure network communications prior to 2009, including my former colleague, the late Dr. Hong Huang.

39. My opinions provided in this declaration would not change in view of minor modifications to this level of ordinary skill.

V. OVERVIEW OF THE TECHNOLOGY

40. I was asked to briefly summarize the background of the prior art from the standpoint of a person having ordinary skill in the art prior to January 28, 2009. As explained below, providing secure network access and communications by authenticating a device and its software was well-known. Thus, the techniques discussed in the '055 Patent were well-known and conventional prior to the invention of the '055 Patent.

A. Device and Software Authentication to Maintain Network Security

41. By at least 2003, mobile computing devices (e.g., phones, laptops, personal digital assistants (PDAs), etc.) were commonly used. EX1012 (*Chmaytelli*), 1:20-24; EX1013 (*Hind*), 1:24-31 (describing the “proliferation of wireless devices in computer networks,” with such wireless devices including, e.g., a laptop, cell phone, and pager). These mobile computing devices were commonly connected to “various networks,” including public networks like the internet, and thus, securely authenticating devices became increasingly important. EX1014 (*Baldwin*), [0003]; EX1015 (*Paul*), [0001], [0003]; EX1016 (*Giles*), 1:15-20 (describing allowing

computing devices to “remotely connect to an enterprise or other private computer network over the public Internet”). Indeed, wireless communication devices were known to enable users to communicate via various networks, including the internet via a web browser, connecting to applications servers via installed applications to access resources, and communicating with an enterprise network, such as via email. EX1017 (*Brown*), [0002]; EX1012, 1:13-30; EX1018 (*Buer*), 1:20-25. Given the mobile nature of these devices and the ability of such devices to connect to a variety of private and public networks, the risk of devices becoming “infected” or compromised when moving between networks has increased. EX1019 (*Trusted Computing Group*), 2. Another known risk is unauthorized users accessing sensitive data on a private network or mobile device. EX1017, [0004] (describing “as more people have access to the network and more data becomes available on the network, the risk of the wrong people accessing sensitive data increases”).

42. Because of the security risks above, the importance of authenticating a device prior to providing network access was known. EX1014, [0003] (describing the increasing importance of securely “authentica[ing] the entities involved in transactions using these devices”); EX1016, 1:17-23 (generally describing controlling “who has access to enterprise networks” to prevent “unauthorized users from remotely accessing their computer networks” via authentication); EX1018, 1:26-30. Indeed, mobile devices were often used to access confidential information

via various networks (e.g., conducting financial transactions and/or obtaining private information), further affirming the need to authenticate the mobile device prior to allowing access to a network. EX1015, [0003]; EX1017, [0004] (describing “it is desirable” to limit “devices that access” a network via device authentication using a “device certificate”). Thus, performing device authentication to allow a device to access a network was well-known prior to 2009, and was often accomplished by checking a device credential. EX1017, [0004]; EX1020 (*Wynn*), [0010] (describing provisioning network credentials to a device to provide communication access); EX1021 (*Weatherford*), [0003]; EX1022 (*3GPP*), 8 (describing issuing authentication credentials (e.g., shared secret, certificate) to a client to provide secure communications with an application server on a network); EX1018, 1:44-60 (describing checking a key of a device to authenticate the device and accordingly allow network access). Authentication was commonly used to obtain secure access. EX1023 (*Moore*), 1:12-16; EX1019, 6 (generally describing authenticating a device at an authentication server prior to providing network access via 802.1X authentication protocols).

43. Furthermore, authentication of applications run on the wireless devices, as well as the device itself was known to be important. EX1024 (*Ksontini*), [0005] (“A verification becomes thus necessary on at least two levels: on the one hand at the level of the mobile equipment itself and on the other hand at the level of software

applications allowing the functioning of the different services”). Indeed, authentication was known to ensure that an application that accesses confidential information during execution is legitimate. EX1024, [0005]-[0006]; EX1025 (*Patel*), [0006]. For example, nefarious individuals selling knock-off or tampered applications and making such appear “trusted” to a user was a known risk. EX1026 (*Ngalle*), 1:56-61. A user may unknowingly download an “unverified” application, which could cause the network to be compromised or information on the network to be compromised when a user executes the unverified application. EX1025, [0006]; EX1009 (*Lundblade* ’330), [0006] (describing before providing a resource on the network (e.g., data) requested from an application, the network “needs to verify that the application is authorized to receive the data,” and thus it is “necessary for the application to authenticate to the data server”). Thus, authenticating the application prior to providing network resource access (e.g., executing the request for a network resource) was a well-known fix to such a “rogue software application.” EX1025, [0006]. Methods of authenticating a software application were known to include generating and checking a credential unique to an application on a particular device. EX1026, 1:4-27 (generally describing a device browser verifying signed application code and an issued certificate); EX1024, Abstract (generally describing generating an application-specific cryptogram and verifying the cryptogram to provide the application access to resources); EX1027 (*Lakshmeshwar*), 2:17-25 (describing an

application on a mobile phone authenticating to a network application server by obtaining an application specific shared secret (i.e., application specific credential)).

44. Thus, authenticating devices and applications on devices for accessing networks and network resources was known prior to 2009.

B. Device Authentication for Network Access

45. Device authentication was known to start with a device request, e.g., requesting access to a network or registration/activation of the device onto a network. EX1016, 2:65-67, 4:5-7 5:36-40 (generally describing a user requesting access to an internal enterprise network to initiate authentication); EX1017, [0017], [0037] (generally describing activating a mobile device with a data network by requesting a device certificate); EX1028 (*Camenisch*), [0007] (generally describing requesting registering a device and accordingly receiving a “device related credential” in response). A request for a “device related credential” was known to form a part of such network access/registration requests. EX1017, [0036]-[0037] (device requesting a device certificate as part of device activation); EX1020, [0023], [0027] (device requesting a credential to access a network via a hotspot).

46. Such device credential requests were known to commonly include an identifier of the user and/or subscriber of the device, such as a phone number, username, password, IMSI, etc. EX1016, 4:31-40, 5:67-6:3; EX1017, [0037]-[0041], [0051] (describing that the device certificate request could include a PIN and

the user's email or could include a phone number). It was also common to include device identifiers within such authentication requests, with these device identifiers known to include a MAC address, serial number, IMEI, "a unique identifier, or any other identifier that identifies the digital device 102," which is separate from a "user identifier," e.g., username. EX1020, [0027], [0043]; *see also*, EX1017, [0052] (disclosing using a device "serial number" to identify the device within the device certificate request); EX1016, 3:14-19, 4:23-26 (describing the computing device having predetermined parameters "relate[d] to the physical computing device itself, such as unique hardware identifier (such as an international mobile equipment identity or IMEI, where the device is a mobile telephone), a hardware serial number, media access controller (MAC) address, and the like" and sending the parameters to the network when requesting network access). Well-known and common device identifiers included IMEI, MAC address, electronic serial number (ESN), system identification numbers (SID), among others. EX1016, 3:14-19, 4:23-26; EX1029 (*Pashley*), 7:14-17, 7:53-59 (describing a handheld computer having a "system identification number" that "uniquely identifies the particular handheld system"); EX1030 (*Pai*), 1:15-17, 1:29-31 (describing a "security code" commonly employed in computers being a "system identification number" and the SID being "unique to the computer system").

47. Once the request is received by the network, the device and/or user identifier is often verified by comparing the received identifier with a list in a database on the network. EX1013, 9:14-23; EX1016, 4:53-63; EX1020, [0052]. This ensures the device should actually have access to the network. EX1016, 4:53-63; EX1021, [0116]-[0117]. This verification step would often be performed when a device requests registration on a subscriber network, such as a cellular network or hot spot, to ensure the device is subscribed to use the network. EX1021, Abstract, [0116]-[0117]; EX1020, [0023] (describing a user requesting access to a hot spot and automatically requesting a device credential).

48. Based on receiving the authentication request, the network then generates a device credential, such as, e.g., a device certificate, often via a certification authority. EX1028, [0016]; EX1013, 9:29-41; EX1017, [0042]-[0043]. A device credential in the form of a certificate is often generated via a certificate authority, and the device certificate is unique to the device. EX1013, 9:29-41; EX1017, [0042]-[0043]. Device credentials other than certificates were known to be generated, including keys, passwords, etc. EX1008 (*Chen*), 8:16-41 (generally describing generating a password as a device credential after the client requests a credential); EX1014, [0076] (generally describing a device master key (DMK) being a “shared secret between the device and one or more device authorities”). A device credential is unique to the requesting device because it is generated based on a device

identifier. EX1028, [0016] (describing generating the “device related credential” based on trusted platform module (TPM) information (i.e., identification information unique to the device) so the device credential “cannot be used on any other computing device”); EX1014, [0102] (Stating the device master key is associated with the “machine so that DMK cannot be transferred by unauthorized means from one machine to another”). The network may store the credential with information identifying the user device (e.g., device ID), for later use in comparing with the received credential to validate the device should be provided access. EX1020, [0052]-[0053]; EX1031 (*Katsube*), 20:50-66 (describing a web authentication proxy server that accesses a database storing a “hash value stored corresponding to the device ID”).

49. The device credentials, when sent from the network to the devices, are often sent securely to remove the risk of snooping on the signal to learn the device credential. EX1013, 11:23-29, 9:23-55 (describing using an “encryption key” to establish a temporary secure link for sending the device certificate back to the device); EX1008, 8:52-59. This secure sending may be accomplished through encryption, which was well-known. EX1013, 9:23-55; EX1020, [0046] (Describing the “credential request response” being sent to device 102 via encryption). For example, private/public key encryption was commonly used to secure transmissions, including transmitting credentials. EX1028, [0017].

50. Once the device receives the device credential, the device credential is used to establish a secure communication channel, such as SSL/TLS. EX1013, 10:39-65 (generally describing the device exchanging certificates to provide “mutual authentication” under SSL/TLS protocol and being passed a “link key” that establishes secure SSL/TLS communication); EX1032 (*Matta*), 16:1-6 (describing certificate exchange between a RADIUS server and device as “enable[ing] the creation of a secure link”). It was known that information must be provided to the server side of a network (i.e., credentials exchanged) to negotiate a secure message link, so the server can determine the legitimacy of the device and accordingly allow or deny communication between two devices. *Id.*

51. Thus, the steps of authenticating a device for providing network access was known prior to 2009, including: (1) receiving a request for a credential from a device that includes a device ID, (2) verifying the device ID, (3) generating the credential, (4) associating the credential with the device ID in memory, (5) securely sending the credential, and (6) establishing a secure link based on the credential.

C. Application Authentication for Network Access

52. Much like device authentication to access a network, steps for authenticating an application trying to access a network resource (e.g., data on a server) were also known to start with a request for an application credential. EX1009, [0034]; EX1027, 6:39-47 (describing an application requesting registration to

authenticate to a server); EX1024, [0072] (describing, during a verification process, an application on a user device requesting an application credential (e.g., cryptogram)). For example, Ksontini's application authentication system requests a cryptogram (e.g., application credential) used in verifying the application should be provided access to confidential resources. EX1024, [0055]. The request often includes an application identifier to identify the application requesting access, which may be verified against a trusted source. EX1009, [0034]-[0035] (request containing the "application identifier (ID)"), [0042]-[0043] (describing checking the application ID via a signature checker to ensure the "application is truly the one assigned the particular application ID"); EX1027, 6:39-47 (describing the request to authenticate containing an application identifier (i.e., "application instance identifier")). The application authentication request also often includes information about the device, such as the device credential. EX1033 (*Kurn*), [0118] (describing a user requesting to execute a "business application," with such a request including "information necessary to authenticate the consumer," such as a "digital certificate"). Indeed, Kurn teaches sending "such a request" to execute a business application, including a device credential (i.e., certificate) to a key repository process to authenticate the application with the enterprise network. EX1033, [0118]. The request also includes application identification and authentication information so the

network is “able to determine with assurance which application is making the request.” EX1033, [0118].

53. Once the request for authenticating the application is received, the network then generates an application credential that indicates the authentication of the application on the device. EX1009, [0049]; EX1027, 7:17-27 (describing deriving and sending back an application key so the application can authenticate itself and the application can be added “into its trusted applications list”); EX1024, [0055]-[0056], [0072] (generally describing the control server generating the cryptogram from the application that includes a “digest” of the application). The authenticated application credential was known to be generated based on the application ID and the device credential. EX1009, [0048]-[0049] (describing generating the application credential from the checked application ID and the “master credential 204” of the device); EX1014, [0033] (describing using the device master key (i.e., device credential) to compute “application keys” at a server to “establish trust and authentication of devices and device-bound applications”), [0039], [0063].

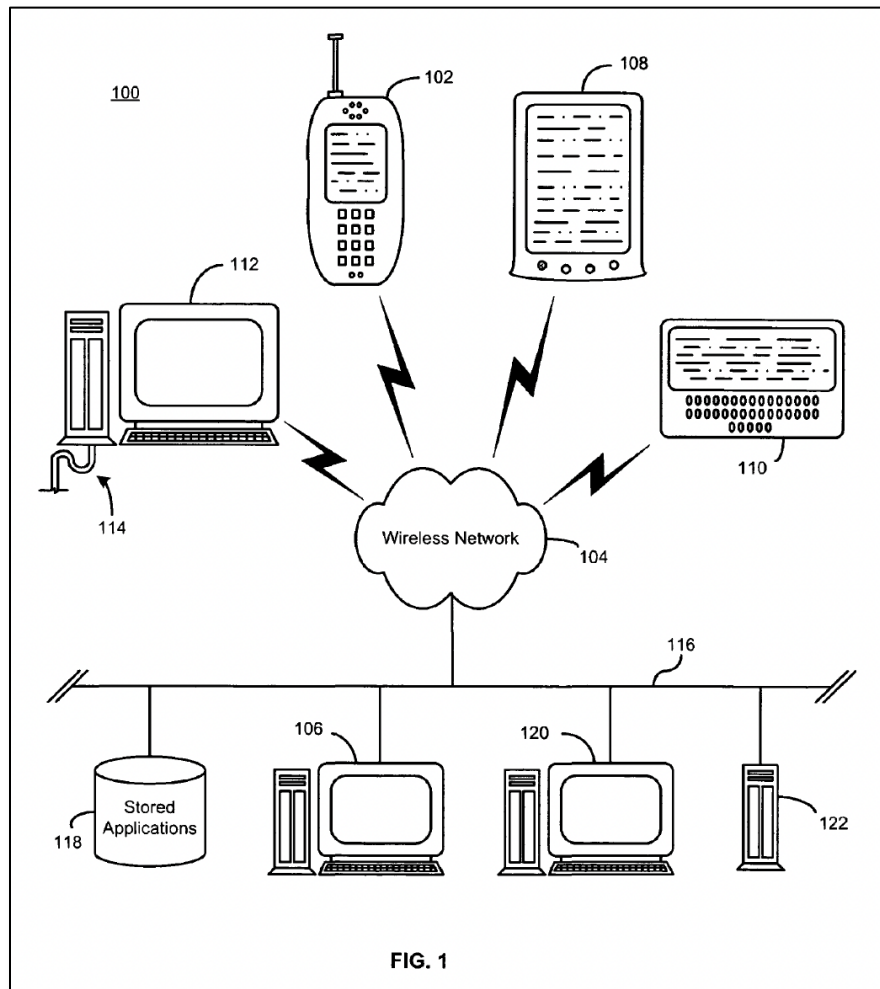
54. Once the application credential indicating authentication is generated, the credential is sent back to the user device, so the user can apply the credential to indicate the application is authentic/trusted. EX1024, [0072]; EX1027, 7:17-27; EX1014, [0107].

55. Thus, the process of authenticating applications by provisioning application credentials was well-known prior to 2009, including (1) requesting an application credential (by request access to a network resource, such as, via executing an application), (2) generating an application credential based on the device credential and an application identifier, and (3) sending the generated application credential to the device.

VI. SUMMARY OF THE PRIOR ART REFERENCES

A. Lundblade

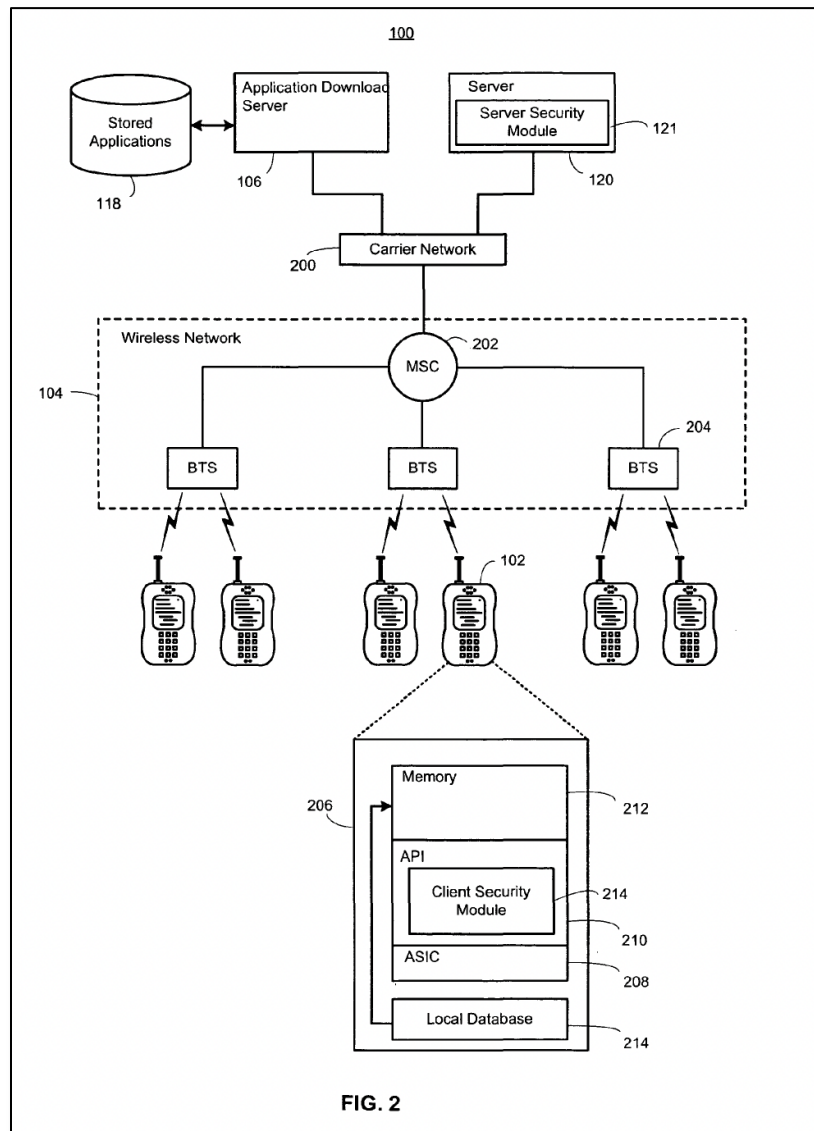
56. Lundblade teaches a system and method for provisioning authentication credentials to a device through a network system to provide “secure communications between remote computing devices and servers.” EX1004 (*Lundblade*), Abstract, 9:59-67 (disclosing the patent “describes methods, systems, software and apparatus used to provide secure communications between client computing devices and servers”). Lundblade understands the importance of “strong authentication between client computing devices and servers.” EX1004, 2:13-15. The system 100 includes various wireless devices 102, 108, 110, 112 and servers 106, 120, 122 that communicate with one another across a wireless network 104. EX1004, 10:19-26, 10:38-43. Figure 1 below shows system 100:



EX1004, FIG. 1.

57. The wireless devices are, e.g., a cell phone 102, a PDA 108, a pager 110, or a computer 112 “that has a wireless communication portal.” EX1004, 10:26-32. Any of servers 106, 120, 122 may carry out logic for performing secure communications with the wireless devices 102, 108, 110, 112, and each server 106, 120, 122 provides services and processes to the wireless devices 102, 106, 108, 110, 112 across wireless network 104. EX1004, 10:41-50.

58. FIG. 2 shows system 100 in further detail, including application download server 106 and server 120 (that is “needed to provide cellular telecommunication services”) communicating with a carrier network 200, such as through the Internet, LAN, WAN, or other network:



EX1004, FIG. 2, 10:60-65.

59. Though only wireless device 102 is shown in FIG. 2, Lundblade teaches system 100 “can include any system whereby remote modules, such as wireless client computing devices 102, 108, 110, 112 communicate over-the-air between and among each other and/or between and among components connected via a wireless network 104, including, without limitation, wireless network carriers and/or servers.” EX1004, 10:54-60. As shown in FIG. 2 above, server 120 includes a server security module 121 that “contains logic configured to provide for secure communications over the carrier network 200.” EX1004, 10:65-11:1. The client computing devices 102, 108, 110, 112 contain a client security module 214 that operates with the server security module 121 to provide secure communications over network 104. EX1004, 11:1-4, 11:37-39. The wireless client computing device receives and executes software applications “transmitted from the application download server 106.” EX1004, 11:20-23.

60. Lundblade teaches how the network provides secure communications between a client computing device 300 and a remote server 400. EX1004, 11:59-61, 14:49-51. The client computing device 300 may be any of wireless devices 102, 108, 110, 112. EX1004, 15:63-65. The server 400 may be any of servers 106, 120, 122. EX1004, 14:63-65. To initiate secure communications with server 400, Lundblade teaches a client 300 “detect[ing] activation of itself,” and accordingly retrieving and sending characteristics 310 of client 300, via the network, to server 400 (steps 504-

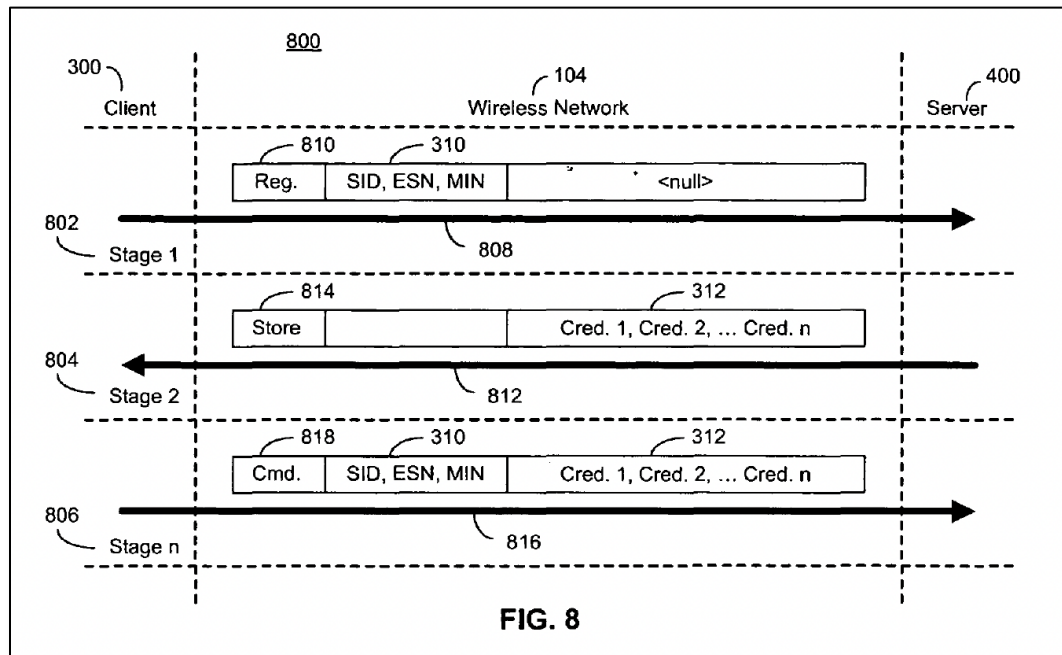
508 of FIG. 5). EX1004, 16:49-61. Characteristics 310 may include one or more of: a phone number, SID, hardware ID (e.g., ESN), carrier ID, removable user identity module (RUIM), mobile directory number (MDN), international mobile subscriber identity (IMSI), among other identifiers. EX1004, 12:29-42. The characteristics 310 include identifiers “useful in uniquely identifying the particular client computing device 300.” EX1004, 12:49-51.

61. After receiving the characteristics 310, the server 400 generates two device credentials 320, 322: one based on the characteristics 310 and a unique client key 409, and another based on the characteristics 310 and a generic key 410 (steps 510-514). EX1004, 16:61-17:3. The server 400 then sends the generated credentials (referred to together as 312) to client 300 (step 516). EX1004, 17:4-6. Lundblade teaches one or more of the steps in method 500 of providing secure communication are “performed in a secure mode” to “prevent unwanted snooping of data by unauthorized programs and unauthorized entities.” EX1004, 17:10-17.

62. Once client 300 receives and stores the credentials 312 (steps 518, 520), the client 300 then sends the credentials 312 to the server 400 in subsequent communication messages. EX1004, 17:46-48, 17:56-61. The server 400 then compares the credentials 312 to “expected credentials” stored on server 400 and, if a match occurs, the server 400 processes the signal “as an authorized signal.” EX1004, 17:61-18:5. Thus, client 300 and server 400 use credentials 312 to

authenticate the communicated signals to ensure the device is authorized to communicate with server 400 on the network.

63. FIG. 8 illustrates a process 800 of providing secure communications between client 300 and server 400 by establishing secure transmissions over network 104. EX1004, 18:7-13. Process 800 provides exemplary transmissions between client 300 and server 400 that implement method 500. For example, in process 800, client 800 transmits a signal 808 containing a registration command 810 and client characteristics 310. EX1004, 18:13-16. In response to signal 808, server 400 generates the credentials 312 and transmits a signal 812 containing the credentials 312 to client 300. EX1004, 18:16-20. After storing the credentials 312, client 300 generates a subsequent message 816 containing the credentials 312, so server 400 compares the received credentials 312 with the expected credentials to authorize the communication as secure, as discussed above. EX1004, 18:26-30. Figure 8, below, shows such exemplary transmissions for carrying out secure communications:



EX1004, FIG. 8.

64. I have been informed that for a prior art reference to be proper for use in an obviousness analysis, the reference must be “analogous art” to the claimed invention. I have been informed that a prior art reference is analogous to the claimed invention if the reference is from the same field of endeavor as the claimed invention or if it is reasonably pertinent to the particular problem that the inventor was trying to solve.

65. As discussed above, the field of endeavor of the claimed invention of the '055 Patent includes security of devices and/or software communicating data via a network. See ¶¶ 31-32. As demonstrated above, Lundblade provisions device-specific credentials to provide secure communications within a network between a device and server. Thus, Lundblade is directed to security of devices and/or software

communicating data via a network and accordingly is in the same field of endeavor as the claimed invention of the '055 Patent.

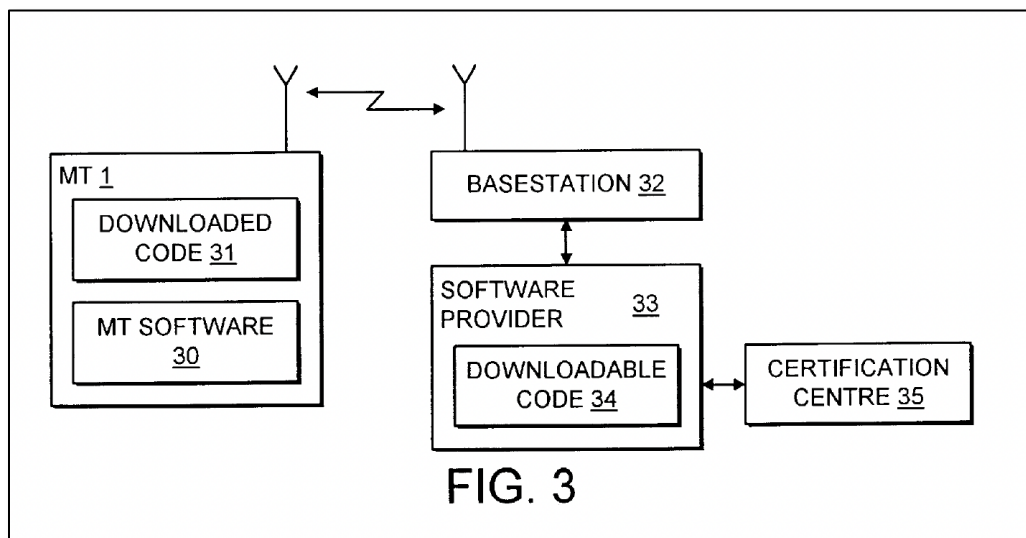
66. Lundblade is also reasonably pertinent to the problems solved by the inventors of the '055 Patent. As discussed above, the '055 Patent is directed to solving the problem of decreasing fraudulence at a device, application, and/or on a network. *See* ¶¶ 33-34. As demonstrated above, Lundblade describes methods for providing secure communication between a client and server over a network to ensure the device (and signals sent by such device) is authentic and thus is concerned with the problem of fraud devices. Thus, a POSITA would have consulted Lundblade and applied its teachings when faced with the problem of decreasing fraudulence at a device and/or on a network.

B. Jobst

67. Jobst teaches securely communicating data (e.g., software or components of software) on a network by verifying the communications and securely sending the data. EX1005 (*Jobst*), Abstract (generally describing “requesting communication terminal” and “providing communication terminal” verifying each other’s communications using unique codes). The secure communication verification is performed to ensure “communication terminals such as cellular phones” do not interact “with the network or other types of electronic equipment in an unintended or unfavorable.” EX1005, 1:41-45. Additionally, the

secure communication method solves a “need for securing the terminal against unauthorized software loading into the phone.” EX1005, 1:46-49, 4:35-46. Jobst’ invention thus provides both a requester and provider the ability to verify each other’s identity, as well as the requester the ability to check whether the communication comes from “an authorized provider.” EX1005, 2:1-13.

68. Jobst teaches the method being performed between a mobile terminal (e.g., a phone 1), software provider 33, and certification center 35, with communication occurring via a “base station 32 in the communication network.” EX1005, 6:31-33, 7:5-8, 7:11-14, FIG. 3. FIG. 3 below shows such components:



EX1005, FIG. 3

69. Jobst teaches providing secure communication across the network, for example, when phone 1 “wants to download and application or a program segment for use in an already available application.” EX1005, 6:57-59. The method of

verifying communicating entities starts when phone 1 sends a request for software (or a software component) to the software provider 33 via the communication network (i.e., via base station 32). EX1005, 6:57-62, 7:5-8, 10:26-32. The request includes a device identifier of the phone (i.e., IMEI) and identifies the “requested software.” EX1005, 6:63-7:1.

70. The software provider 33 (also referred to as service provider 33) sends the IMEI portion of the request to certification center 35, which checks “the validity of the request” by checking the IMEI against a database table of IMEIs to ensure the “phone 1 has an open account for paying for the requested software code or file.” EX1005, 7:15-20, 9:26-37. In validating the IMEI, the certification center 35 also finds the phone number based on the IMEI. EX1005, 7:20-22. Once the request is found valid, certification center 35 “calculates a phone password” using the IMEI via a secure hash algorithm. EX1005, 7:24-32. The certification center 35 also calculates a first signature 43 unique for the specific software transfer that includes the phone password. EX1005, 7:37-48. The first signature 43 is unique to the specified software and the requesting phone. EX1005, 7:48-52.

71. The certification center 35 then forwards the first signature 43 to the software provider 33, and software provider 33 transfers the software code and first signature in a message to the phone via the network (i.e., via base station 22). EX1005, 7:53-61, 9:56-61. The phone then extracts the software code and first

signature 43 from the message and calculates its own second signature 46. EX1005, 7:62-65, 8:9-16, 10:43-57. The same signature generating algorithm used to form the first signature 42 is used by phone 1 to generate the second signature, and the phone 1 uses the same phone password. EX1005, 8:11-17, 8:1-3. The phone 1 then compares the second signature 46 with the first signature 42, and if they match, the phone 1 “deems the response message to be coming from an authorized software provider,” indicating to the phone 1 that the software code is authentic. EX1005, 8:17-25. The phone then downloads the code. EX1005, 8:23-25. The software provider message may include “an instruction that, when authenticated by the phone, activates an application that has been present in the phone as software code,” rather than including the actual software code for download. EX1005, 8:45-49, 9:5-10.

72. The certification center 35 may store the phone password in a record “that may be addressed by the IMEI code.” EX1005, 9:11-14. The software provider 33 is, for example, a server that is authorized by the phone manufacturer and communicating on the network phone 1 is connected to. EX1005, 9:17-24. The technique for verifying software code/instructions may include using public key encryption. EX1005, 11:24-31.

73. As discussed above, the field of endeavor of the claimed invention of the '055 Patent includes security of devices and/or software communicating data via a network. *See* ¶¶ 31-32. As demonstrated above, Jobst teaches authenticating

software and/or instructions over a communication network to activate software on a phone to ensure the signals come from an authorized software provider, as well as ensuring software/instruction requests come from an authorized phone. Thus, Jobst is directed to security of devices and/or software communicating data via a network and accordingly is in the same field of endeavor as the claimed invention of the '055 Patent.

74. Jobst is also reasonably pertinent to the problems solved by the inventors of the '055 Patent. As discussed above, the '055 Patent is directed to solving the problem of decreasing fraudulence at a device, application, and/or on a network. *See ¶¶ 33-34.* As demonstrated above, Jobst solves the issue of decreasing fraudulence of an application and phone by verifying the requesting phone and software provider are authorized to perform communications. Thus, a POSITA would have consulted Jobst and applied its teachings when faced with the problem of decreasing fraudulence at a device and/or on a network.

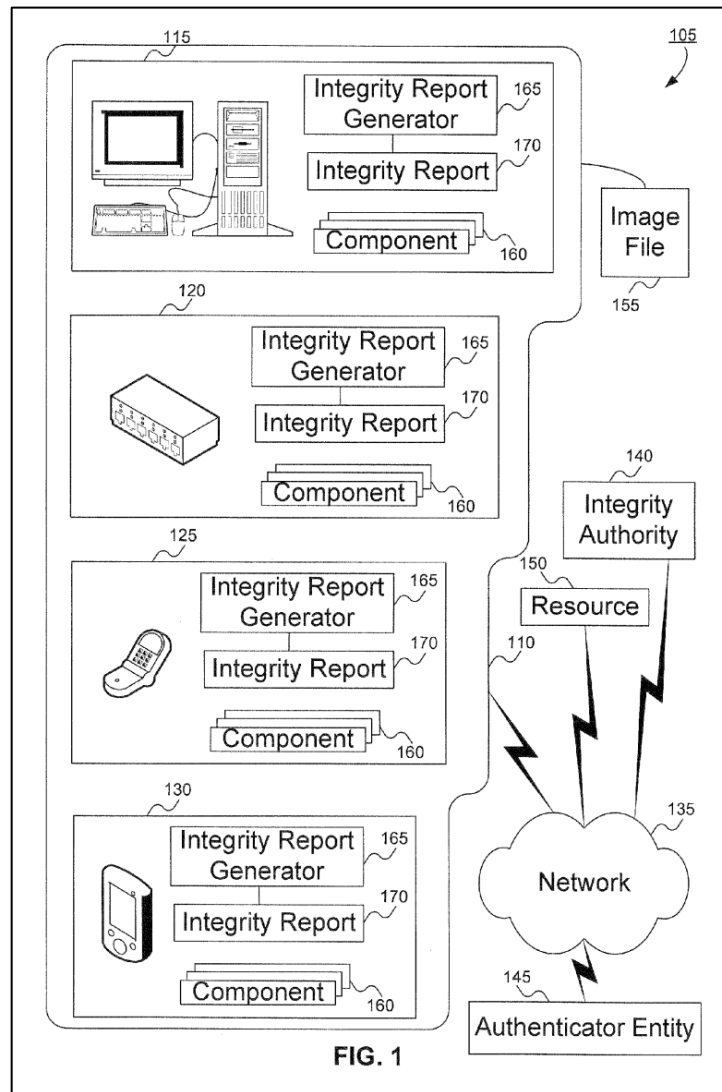
C. Hardjono

75. Hardjono teaches verifying a client prior to providing the client access to a resource or service on a network by validating hardware and software components of the client. EX1006 (*Hardjono*), Abstract, [0009] (disclosing a method and apparatus for “authenticating a client platform”). Hardjono’s verification and authentication provides a “way to identify and authenticate

components of a client platform—both hardware and software—that are in a potentially improper state before the client platform is given access to network services or resources.” EX1006, [0008]. This solves the well-known problem of devices being “tainted” with malware or viruses that may corrupt software on the device, such as when a device connects to a less secure network (i.e., public Internet when away from, e.g., a private corporate network). EX1006, [0006] (describing the problem of malware being introduced into corporate networks through “tainted” machines that were connected to “a network, such as the Internet,” when on a “business trip”). Hardjono also understands that the problem of malware and tainted clients is particularly important in electronic bank transactions, given that such transfers utilize sensitive information. EX1006, [0007]. Thus, the goal of Hardjono’s authentication method is to verify a software component or hardware component on a client is authentic and has not been compromised into an improper state by, e.g., malware.

76. Hardjono teaches the system 105 for performing the authentication process including various clients 110, such as a personal computer 115, network element 120, wireless phone 125, and PDA 130. EX1006, [0030]. The system 105 is connected together via network 135, which may be wireless and any type of network (e.g., LAN, WAN, Internet). *Id.* The system 105 also includes integrity authority 140 and authenticator entity 145 on network 135 and may be “combined

and operated” by a single physical machine. EX1006, [0031]. The authenticator entity 145 includes an authentication server 210, verification server 205, and policy server 215, which may be formed on a single server. EX1006, [0031], [0035]. The clients 110, 115, 120, 125, 130 may have an image file 155 installed. EX1006, [0033]. For example, an IT administrator may install “at least one” software component 160 from image file 155, or the administrator may install a software component 160 using tools provided by the component manufacturer. EX1006, [0033], [0095]. FIG. 1, below, shows the system 105 and its various elements:



EX1006, FIG. 1.

77. Integrity authority 140 and authentication entity 145 work together to perform client “validation and to determine whether to grant or deny access” to a network resource or service. EX1006, [0031]. A resource 150 is, for example, a service including a web service or application, a server, an internal network (such as internal to the organization), a printer, a flash memory disk, storage device, among other possibilities. EX1006, [0031]. Hardjono teaches a procedure verifying the

integrity of software components 160, which starts by the administrator installing “at least one” software component 160 on client 110. EX1006, [0095]. Once the software component 160 is installed, client 110 generates an integrity report 170. *Id.* To generate the integrity report, client 110 collects digests of its components 160, including collecting digests of “application software.” EX1006, [0036]-[0038]. The software applications include, e.g., word processor, spreadsheet application, and email. EX1006, [0037]. The digests include “hashes” of the software components 160. EX1006, [0038]. The integrity report 170 is signed to “verify the identity of the client,” such as by signing with a trusted platform module (TPM) certificate, or another certificate verifying the client 110. EX1006, [0038], [0076].

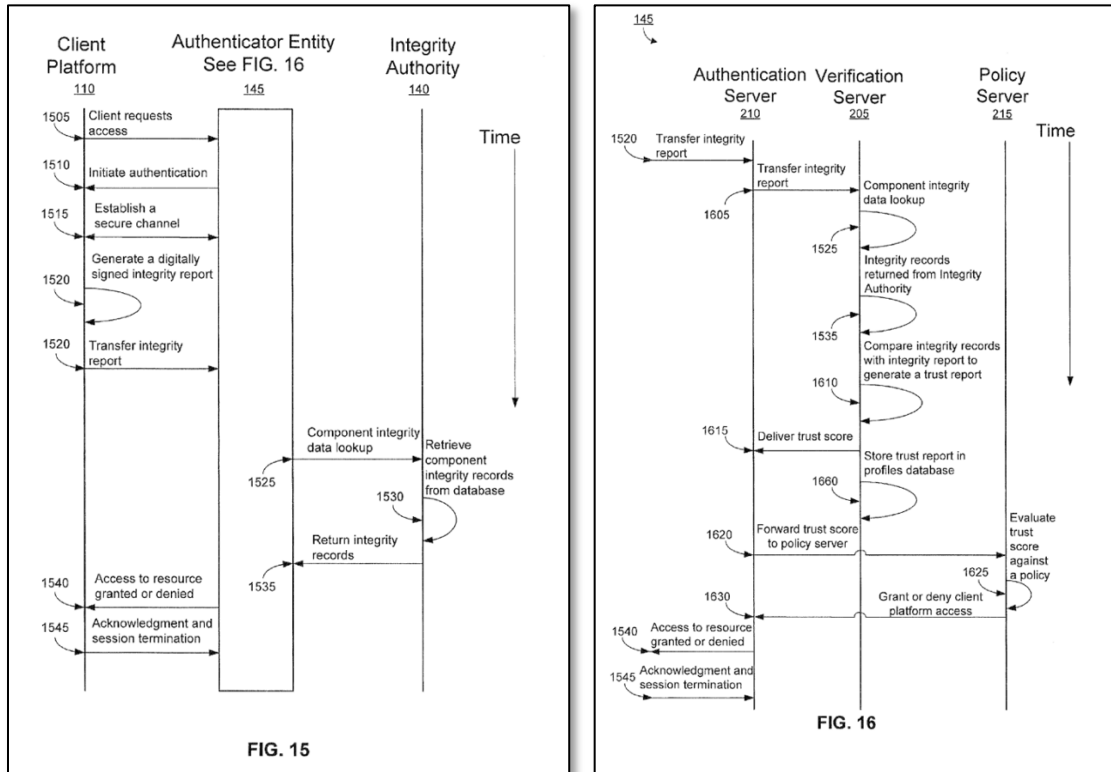
78. The integrity report 170 is used by authentication entity 145 to generate a trust report 220. EX1006, [0095]. For example, verification server 205 of authentication entity 145 generates the trust report 220. The trust report 220 is generated by comparing the component 160 digests in integrity report 170 to digests harvested by integrity authority 140 from the software component 160 source (e.g., from manufacturers, software vendors, etc.). EX1006, [0045], [0048], [0027], [0079], [0077], [0104]. The harvested digests are placed in integrity/trust records 320 that are stored at integrity authority 140. EX1006, [0048].

79. If the integrity report 170 digest matches one in the integrity authority 140, then a high trust score 225 is calculated in the trust report 220. EX1006, [0085],

[0068]. The trust report 220 includes various information, including the client platform identity and a reference to the TPM certificate of the verification server 205. EX1006, [0068]. The trust report 220 also includes “other information 520,” such as a reference to the TPM certificates of client 110, a digital signature of verification server 205, and a list of the one or more components 160, “each with a reference to corresponding integrity/trust record 320.” EX1006, [0068]. Once generated, the authentication entity 145 then sends the trust report 220 back to the client 110, so client 110 uses “trust report 220 as a sort of trust ticket” to access other network resources or services. EX1006, [0041]. The authenticator entity 145 may then check the trust report 220 to determine whether to provide a user access to another resource. *Id.*

80. FIGS. 15-16 show an operation flow of the verification procedure, which includes, among other steps: a client requesting access to a resource 150 (step 1505) and sending an integrity report 170 to the authentication entity 145 (including verification server) (step 1605), verification server 205 comparing the digests in integrity report 170 to digests in integrity records 320 from integrity authority 140 (step 1610), verification server 205 generating the trust report and sending the trust report 220 to policy server 215 that compares the trust score 225 in the report to thresholds to determine whether to provide access (steps 1620, 1625), and

authentication entity 145 forwarding the access decision to client 110 (step 1630, 1540):



EX1006, FIGS. 15-16, [0101]-[0104]. Though not illustrated, the trust report 220 itself may be sent to client 110 for future use. EX1006, [0041].

81. As discussed above, the field of endeavor of the claimed invention of the '055 Patent includes security of devices and/or software communicating data via a network. *See* ¶¶ 31-32. As demonstrated above, Hardjono teaches authenticating a client and its software components to access a network resource or service (e.g., communicate data via a network). Thus, Hardjono is directed to security of devices

and/or software communicating data via a network and accordingly is in the same field of endeavor as the claimed invention of the '055 Patent.

82. Hardjono is also reasonably pertinent to the problems solved by the inventors of the '055 Patent. As discussed above, the '055 Patent is directed to solving the problem of decreasing fraudulence at a device, application, and/or on a network. *See* ¶¶ 33-34. As demonstrated above, Hardjono is directed to ensuring clients that request network resource access are not in an “improper state” due to being tainted by malware or viruses. Thus, a POSITA would have consulted Hardjono and applied its teachings when faced with the problem of decreasing fraudulence at a device and/or on a network.

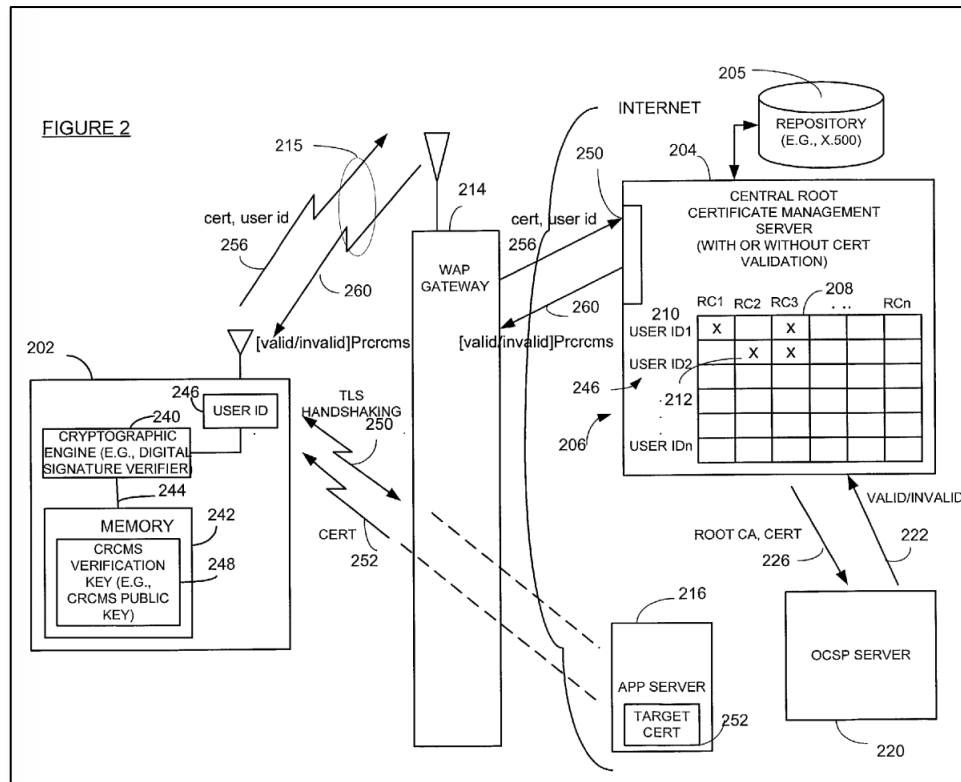
D. Zuccherato

83. Zuccherato teaches a method and system for validating a certificate sent by a wireless mobile unit to a certificate authority. EX1007 (*Zuccherato*), Abstract. The method places certificate validation on the certificate server, rather than the device, so that there is enough memory to accurately authenticate certificates by verifying their chain. EX1007, [0005], [0010]. Additionally, managing certificate validation at one entity allows for more efficient control of root certification authority certificates, to keep track of revoked certificates. EX1007, [0009]-[0010]. Zuccherato also understands the importance of protecting legitimate root certificates “for integrity and authenticity,” as the compromise of a certification authority in a

wireless network, e.g., if a virus is present, allows “unscrupulous party to be allowed to digitally sign information.” EX1007, [0003], [0009]. Thus, Zuccherato’s remote certificate validation and management of certificates helps mitigate the risk of compromising root certificates or allowing nefarious parties to commit fraud.

84. The system for validating certificates includes a wireless mobile unit 202 and a central root certificate management server 204. EX1007, [0023]. The management server 204 stores multi-subscriber root certificates in a store 208, which includes subscriber identifying data and data representing root certificates 212. EX1007, [0023]. The system 200 also includes an application server 216, such as web server or another server that wireless unit wants to communicate with. EX1007, [0024].

85. FIG. 2 illustrates an exemplary operation of system 200 for validating a certificate:



EX1007, FIG. 2, [0029]. A wireless device attempts to set up a transport layer security (TLS) session with the application server 216, via communication 250.

EX1007, [0031]. The application server 216, in response to wireless device 202 attempting to establish secure communication with server 216, sends the target certificate 252 to be validated to the wireless device 202 (via communication 252).

EX1007, [0032]. The wireless device 202 then forwards the target certificate 252 to the central root certificate management server 204 (via communication 256).

EX1007, [0032]. The management server 204 then performs validation of certificate 252 by constructing a certificate chain between the target certificate and a root certificate stored in the multi-subscriber root certificate store. EX1007, [0033]. If the

target certificate 252 is found valid, the management server 204 digitally signs validation data that forms the trusted validation response 260 sent back to the wireless device 202. EX1007, [0033]. If response 260 indicates the certificate 252 is valid, wireless device continues to set up TLS session with the application server 216. EX1007, [0033].

86. As discussed above, the field of endeavor of the claimed invention of the '055 Patent includes security of devices and/or software communicating data via a network. *See* ¶¶ 31-32. As demonstrated above, Zuccherato teaches validating a certificate from an application server to ensure the application server is trusted prior to a wireless device accessing the application server and communicating data with the application server. Thus, Zuccherato is directed to security of devices and/or software communicating data via a network and accordingly is in the same field of endeavor as the claimed invention of the '055 Patent.

87. Zuccherato is also reasonably pertinent to the problems solved by the inventors of the '055 Patent. As discussed above, the '055 Patent is directed to solving the problem of decreasing fraudulence at a device, application, and/or on a network. *See* ¶¶ 33-34. As demonstrated above, Zuccherato is directed to managing certificate validation at a central management server to ensure the source of the certificate is to be trusted, and to reduce the ability of unscrupulous people to create fraudulent digital signatures. Thus, a POSITA would have consulted Zuccherato and

applied its teachings when faced with the problem of decreasing fraudulence at a device and/or on a network.

E. Chen

88. Chen teaches a method of establishing secure communications by authenticating a communication session for a device via a credentialing server. EX1008 (*Chen*), Abstract, 8:16-38, 1:57-58. Chen's method of establishing secure communications addresses the problem of "privacy issues" related to communications between computers on a network, especially when performing communications over the Internet. EX1008, 1:12-42. "Hackers and other ne'er-do-wells may seek to tap into communications on a network," such as intercepting communications to identify confidential information, or obtain usernames and passwords to "later access resources on a network for malicious purposes. EX1008, 1:43-53.

89. Chen teaches the method of assisting in establishing secure communication being carried out by system 200, which includes a credential server 206 and an authentication server 216 communicating with client 202 over network 204. EX1008, FIG. 2, 6:34-37, 6:43-55, 8:16-19. The method starts with the key which is used to encrypt the data being shared between the credential server and the authentication server. EX1008, 8:27-30. Then, after the keys have been shared, a client requests a credential from the credential server. EX1008, 8:35-38. After

receiving the request, the credential server begins a credentialing process that “involves generating data for a password, and encrypting the data.” EX1008, 8:38-41. The password is randomly generated and may contain 16 bytes. EX1008, 8:45-51. The password is then encrypted “using, for example, a 128-bit key shared with an authentication server.” EX1008, 8:52-57. The password and a username are then sent to the client, which provides the “encrypted data” (i.e., password) to an authentication server. EX1008, 8:58-64. The authentication server derives the password from the message to create a 22-character password that is then used to “complete an authentication protocol with a client device.” EX1008, 8:64-66, 9:20-24.

90. As discussed above, the field of endeavor of the claimed invention of the '055 Patent includes security of devices and/or software communicating data via a network. *See* ¶¶ 31-32. As demonstrated above, Chen teaches establishing secure communication between devices and servers across a network by using a credential server and an authentication server. Thus, Chen is directed to security of devices and/or software communicating data via a network and accordingly is in the same field of endeavor as the claimed invention of the '055 Patent.

91. Chen is also reasonably pertinent to the problems solved by the inventors of the '055 Patent. As discussed above, the '055 Patent is directed to solving the problem of decreasing fraudulence at a device, application, and/or on a

network. *See* ¶¶ 33-34. As demonstrated above, Chen is directed to helping establish secure communication by providing an encrypted credential to a client from a credential server that is authenticated at the authentication server, to keep hackers from intercepting communications and fraudulently gaining access to network resources. Thus, a POSITA would have consulted Chen and applied its teachings when faced with the problem of decreasing fraudulence at a device and/or on a network.

F. Lundblade '330

92. Lundblade '330 teaches methods and an apparatus for providing an application credential for an application running on a device and using the application credential to authenticate to a data server. EX1009 (*Lundblade '330*), Abstract, [0010]. Lundblade '330 uses application credentials to “provide strong authentication” when an application running on a device requests data from a proprietary data server. EX1009, [0009]. Lundblade '330 understands the importance of verifying the “application is authorized to receive the data” before transmitting requested data from a data server to an application during execution of the application. EX1009, [0006]. “It is therefore necessary for the application to authenticate to the data server.” *Id.*

93. Lundblade '330 teaches a data network 100 that includes a wireless device 102 that communicates with data network 104 via a wireless communication

channel 106. EX1009, [0031]. The network 100 also includes a credential server 108 that provides services to wireless device 102 and other entities on network 104. *Id.* Network 104 also includes a data server 110 that provides data to devices on network 104, such as wireless device 102. *Id.* The wireless device 102 may be a PDA, phone, tablet computer, or any other type of portable device, and the credential server 108 may be part of a telecommunications network that provides telecommunication services to device 102. EX1009, [0028], [0031].

94. The device 102 includes a master credential 114 installed on the device and allows device 102 to authenticate to other entities, such as credential server 108. EX1009, [0032]. The credential system of Lundblade '330 provides an application credential to an application running on device 102 by, initially, an application program 118 requesting an application credential. EX1009, [0034]. The credential request includes an application ID. *Id.* The credential system uses the application ID and master credential of the device to generate the application credential, which is returned to application 118. EX1009, [0035].

95. Prior to generating the application credential, the credential system checks the application ID 212 and a signature 214 via an application/ID signature checker 202 to detect if the application has been modified or tampered with. EX1009, [0042]-[0043], [0047]-[0048]. This check ensures the application “is truly the one assigned the particular application ID.” EX1009, [0043]. The signature 214

is created by “some application signing service or authority and binds the application 210 with the application ID 212.” EX1009, [0042]. The application ID is checked before being included in the application credential request. EX1009, [0048].

96. The application credential is generated from the master credential and the application ID 212, such as by using a “hash function,” so the credential 208 cannot be “decoded or reversed to discover the master credential 208.” EX1009, [0049]. The application then uses the generated application credential to authenticate to a data server to retrieve data for use on the device. EX1009, [0050]. For example, the application 210 on device 102 sends the application credential 208 and application ID 212 in a request for data to data server 110. EX1009, [0056]. The data server, after receiving the request, then sends a request to the credential server, with the request including the application credential 208 and an authentication token 408. EX1009, [0058]. The credential server generates a server credential from the application ID 212 and a copy of the master credential 214 (stored at the credential server) and sends the server credential to the data server. EX1009, [0059]. The data server then compares the server credential 414 to the application credential 208 and, if they match, the application is authenticated as the data requester, and the data server then provides the requested data to the device 102. EX1009, [0060].

97. As discussed above, the field of endeavor of the claimed invention of the '055 Patent includes security of devices and/or software communicating data via

a network. *See* ¶¶ 31-32. As demonstrated above, Lundblade '330 teaches authenticating an application on a device when the application requests data from, e.g., a proprietary data server on a network. Thus, Lundblade '330 is directed to security of devices and/or software communicating data via a network and accordingly is in the same field of endeavor as the claimed invention of the '055 Patent.

98. Lundblade '330 is also reasonably pertinent to the problems solved by the inventors of the '055 Patent. As discussed above, the '055 Patent is directed to solving the problem of decreasing fraudulence at a device, application, and/or on a network. *See* ¶¶ 33-34. As demonstrated above, Lundblade '330 is directed to providing strong authentication of an application before allowing the application to access data on a proprietary server, thereby ensuring the application is not fraudulent or has not been tampered with by a nefarious individual. Thus, a POSITA would have consulted Lundblade '330 and applied its teachings when faced with the problem of decreasing fraudulence at a device and/or on a network.

G. Fosback

99. Fosback teaches an improved system and method of submitting, distributing, and/or managing digital products (e.g., applications) via a product distribution site. EX1010 (*Fosback*), Abstract, [0040] (disclosing applications as an example of “digital products”). The submission and management system processes

application submissions from a developer and then accordingly “authorize[s] distribution of approved digital products.” EX1010, [0040]. A client 110 downloads the applications from the distribution site 102 via data network 108. EX1010, [0043]. The connections over data network 108 between distribution site 102 and clients are “through secure connections, such as Secure Sockets Layer (SSL),” given that financial transactions occur over these connections. EX1010, [0045]. Thus, Fosback’s system provides secure and simplified distribution and management of applications over network 108.

100. A product submission and management system 104 is also connected to network 108 to process the application submissions and accordingly authorize approved applications. EX1010, [0040]. The approved applications are then stored in a products store 106 and are available for purchase through the product distribution site 102. EX1010, [0043], [0046] (disclosing management 104, store 106, and site 102 may be combined). The product submission and management system 104 requires a developer (i.e., “digital product submitter”) to provide login information. EX1010, [0053]. The management system 104 includes a royalty module 310 that allows a developer to monitor “distribution of all digital products,” including monitoring royalties, as well as an account management module 312 that allows a developer to centrally manage accounts for digital products, such as access permissions to the application, among other things. EX1010, [0054].

101. During product submission by a developer, the system checks the digitally uploaded product “for a required certification (or authentication)” to ensure the files “are appropriate” and trusted. EX1010, [0067], [0069]. Thus, the product distribution system ensure submitted products, such as applications, are trusted and secure via a certificate.

102. As discussed above, the field of endeavor of the claimed invention of the '055 Patent includes security of devices and/or software communicating data via a network. *See* ¶¶ 31-32. As demonstrated above, Fosback teaches authenticating an application submitted over a network to ensure the application is trusted by checking a certificate and securely communicating an application for download from the production site to the user’s device. Thus, Fosback is directed to security of devices and/or software communicating data via a network and accordingly is in the same field of endeavor as the claimed invention of the '055 Patent.

103. Fosback is also reasonably pertinent to the problems solved by the inventors of the '055 Patent. As discussed above, the '055 Patent is directed to solving the problem of decreasing fraudulence at a device, application, and/or on a network. *See* ¶¶ 33-34. As demonstrated above, Fosback is directed to providing a software distribution system that verifies submitted applications are authorized for distribution, such as by checking the authenticity of the application (via certificate) to ensure the application is not nefarious or untrustworthy. Additionally, Fosback’s

distribution system distributes applications over secure SSL links to ensure commercial transactions cannot easily be intercepted by someone attempting fraud. Thus, a POSITA would have consulted Fosback and applied its teachings when faced with the problem of decreasing fraudulence at a device and/or on a network.

H. Scarlata

104. Scarlata teaches a method for providing software-based security coprocessors to emulate a predetermined cryptographic processor. EX1011 (*Scarlata*), Abstract, (54). The virtual security coprocessor uses a physical trusted platform module (TPM) to support virtual TPMs in the processing system. EX1011, Abstract. TPMs were known to be included in various processing systems to provide a “security coprocessor” that “enhance[es] the security of the processing system.” EX1011, 1:31-39. The security services include attesting to the identity and/or integrity of the platform. EX1011, 1:53-63. Thus, when a server handles requests from a client, the server may “deny requests from any client system unless those requests are accompanied by valid, TPM-based platform attestation from that client system.” EX1011, 1:64-2:3. Thus, Scarlata’s TPM-based system provides authentication of a client when requesting something from a server (e.g., data, access), by attesting to the integrity of the client.

105. To provide such attestation proof, TPMs have a set of credentials. EX1011, 7:24-31. These credentials include ones generated at manufacture, but also

include credentials issued from a third party “to provide security,” such as a “certificate authority” that provides an identity credential for each attestation identity key of the TPM. EX1011, 7:31-46 (disclosing the “CA creates identity credentials for AIKs that TPM claims it owns”). The TPM then signs quotes using the identity credential to show “cryptographic proof of the current state of the platform,” in order to attest to the integrity of the client. EX1011, 7:46-51. For example, a conventional attestation approach includes a CA providing an identity credential for an AIK (Attestation Identity Key) of a TPM, and processing system 20 then using that identity credential to make a trustworthiness assertion to a remote processing system (e.g., server). EX1011, 7:51-57. Scarlata teaches TPMs typically included a processor to perform the TPM’s security-related logic. EX1011, 8:40-52.

106. As discussed above, the field of endeavor of the claimed invention of the ’055 Patent includes security of devices and/or software communicating data via a network. *See* ¶¶ 31-32. As demonstrated above, Scarlata teaches a TPM having an identity credential issued by a certificate authority that the TPM uses to attest to the integrity of the client when a client makes a request (e.g., requesting data communication) to a server across a network. Thus, Scarlata is directed to security of devices and/or software communicating data via a network and accordingly is in the same field of endeavor as the claimed invention of the ’055 Patent.

107. Scarlata is also reasonably pertinent to the problems solved by the inventors of the '055 Patent. As discussed above, the '055 Patent is directed to solving the problem of decreasing fraudulence at a device, application, and/or on a network. *See* ¶¶ 33-34. As demonstrated above, Scarlata is directed to providing TPMs that attest to the trustworthiness of a device, so a server receiving a request from a client can determine whether the device is legitimate (i.e., the TPM attests to integrity of the client) or is potentially fraudulent. Thus, a POSITA would have consulted Scarlata and applied its teachings when faced with the problem of decreasing fraudulence at a device and/or on a network.

VII. SUMMARY OF UNPATENTABILITY

108. I have reproduced the Proposed Grounds of Unpatentability from the Petition for ease of reference:

Claim(s)	Basis	Reference(s)
1-2, 6-11, 13-15, 18-19	§ 103	Lundblade (EX1004), Jobst (EX1005), Hardjono (EX1006)
3	§ 103	Lundblade (EX1004), Jobst (EX1005), Hardjono (EX1006), Zuccherato (EX1007)

4	§ 103	Lundblade (EX1004), Jobst (EX1005), Hardjono (EX1006), Chen (EX1008)
5, 16	§ 103	Lundblade (EX1004), Jobst (EX1005), Hardjono (EX1006), Lundblade '330 (EX1009)
12	§ 103	Lundblade (EX1004), Jobst (EX1005), Hardjono (EX1006), Fosback (EX1010)
17	§ 103	Lundblade (EX1004), Jobst (EX1005), Hardjono (EX1006), Scarlata (EX1011)

VIII. GROUND 1: CLAIMS 1-2, 6-11, 13-15, AND 18-19 ARE OBVIOUS OVER LUNDBLADE, JOBST, AND HARDJONO

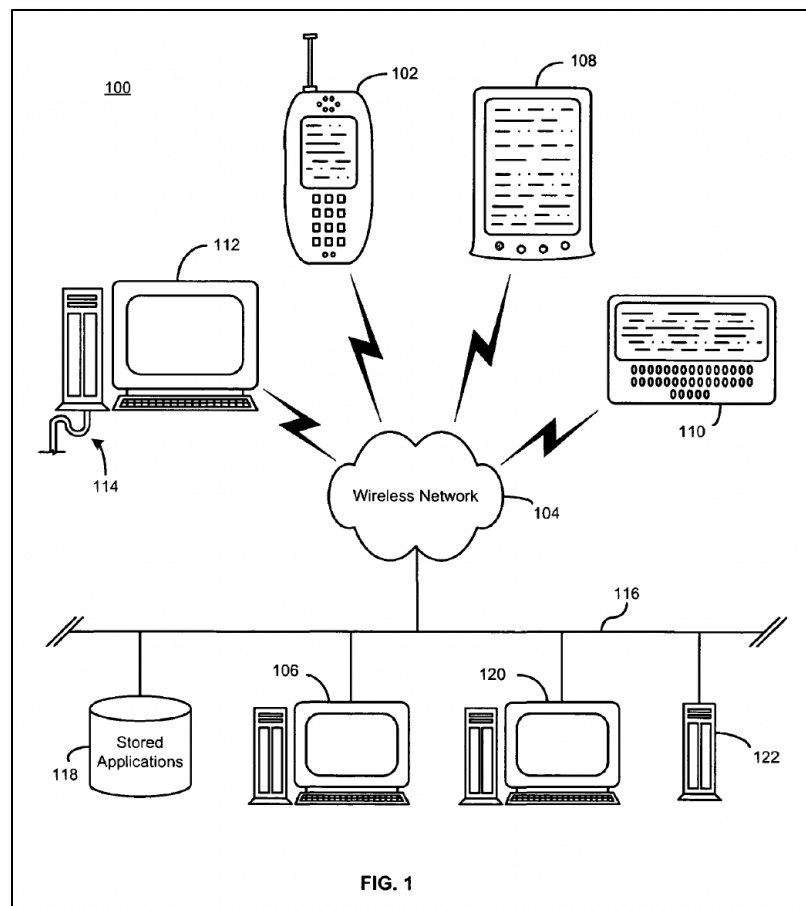
A. Claim 1

1. *1[Pre]: “A method of operating a network system, the method comprising:”*

109. Lundblade teaches a *method of operating a network system*¹; namely, a method for providing authentication via credentials on a network system (i.e., servers 106 and 120, wireless network 104 and carrier network 200) to provide secure communications between remote computing devices and servers. EX1004

¹ All emphasis added unless otherwise noted.

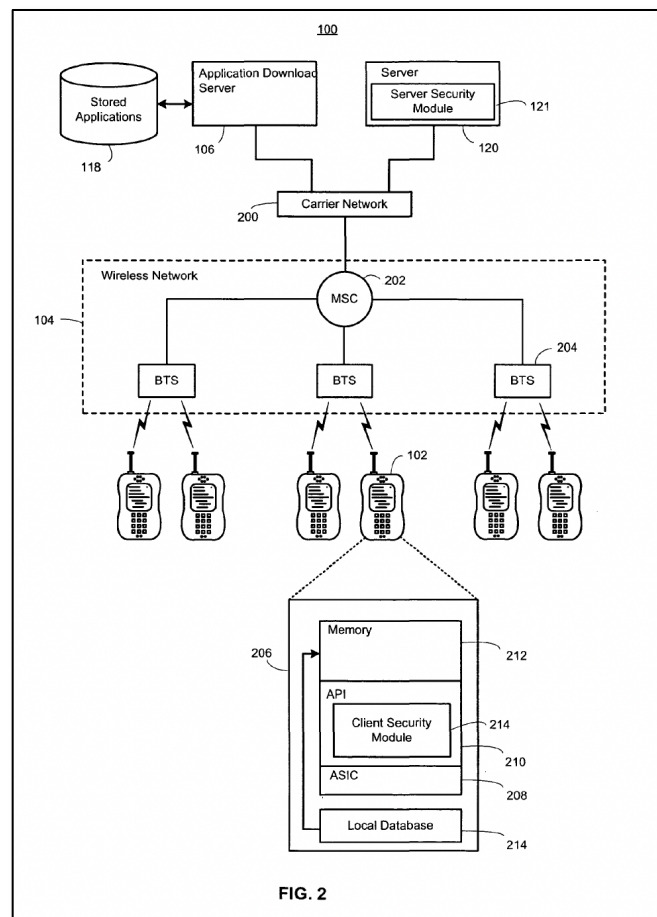
(Lundblade), Abstract (disclosing a “method for providing secure communications between remote computing devices and servers,” with such a method including sending information between the remote device and server “over the network”). FIG. 1 shows a system 100 including wireless devices (102, 108, 110, 112) in communication with servers (106, 120, 122) “across a wireless network 104[.]”



EX1004, FIG. 1, 10:19-23, 10:38-43. Any of the servers 106, 120, 122 may include “logic to perform secure communications” with wireless devices 102, 108, 110, 112. EX1004, 10:46-50. For example, application download server 106 selectively

transmits software applications across wireless network 104 to wireless devices 102, 108, 110, 112. EX1004, 10:19-26.

110. FIG. 2 also illustrates system 100, but in further detail, “showing the interrelation of the elements of the exemplary” system, including wireless network 104, a carrier network 200, and servers 106 and 120:

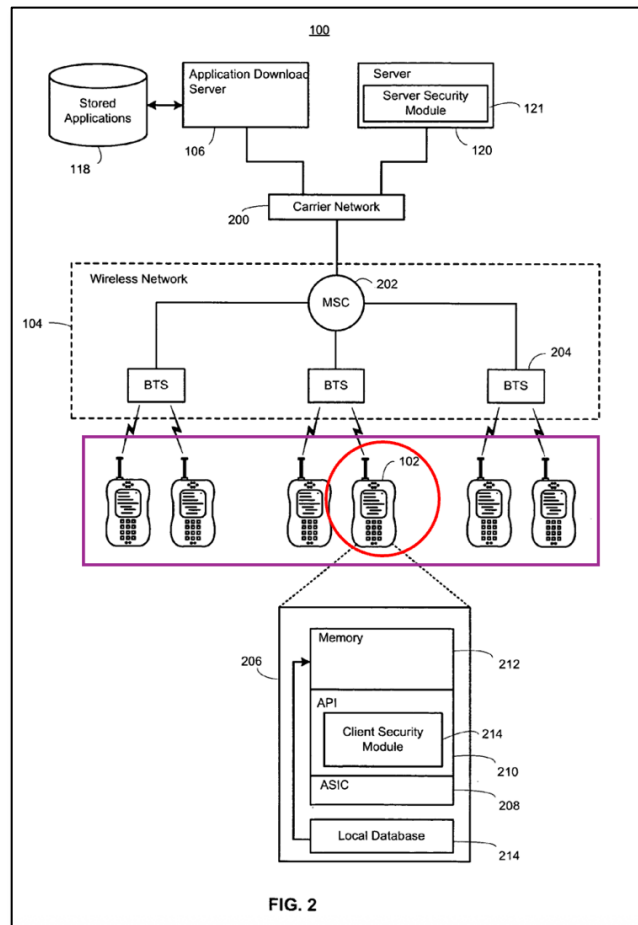


EX1004, FIG. 2. Servers 106, 120 “communicate with a carrier network 200, through a data link, such as the Internet, a secure LAN, WAN, or other network.” EX1004, 10:60-65. The *network system* being operated in Lundblade thus includes the carrier network 200, wireless network 104, and servers 106, 120.

2. ***1[a]: “receiving a first request from any first device in a plurality of similar wireless end-user devices, the first request for a secure device credential”***

111. Lundblade teaches *receiving a first request from any first device in a plurality of similar wireless end-user devices, the first request for a secure device credential*; namely, by the network server 400 (e.g., server 120) receiving a registration command signal from a client computing device 300 (e.g., mobile phone 102 – shown in red in annotated FIG. 1 below) for the purpose of getting credentials 312, as discussed further below.

112. The client device 300 (e.g., mobile phone 102) is a *first device in a plurality of similar wireless end-user devices*, shown in purple in annotated FIG. 2 below:

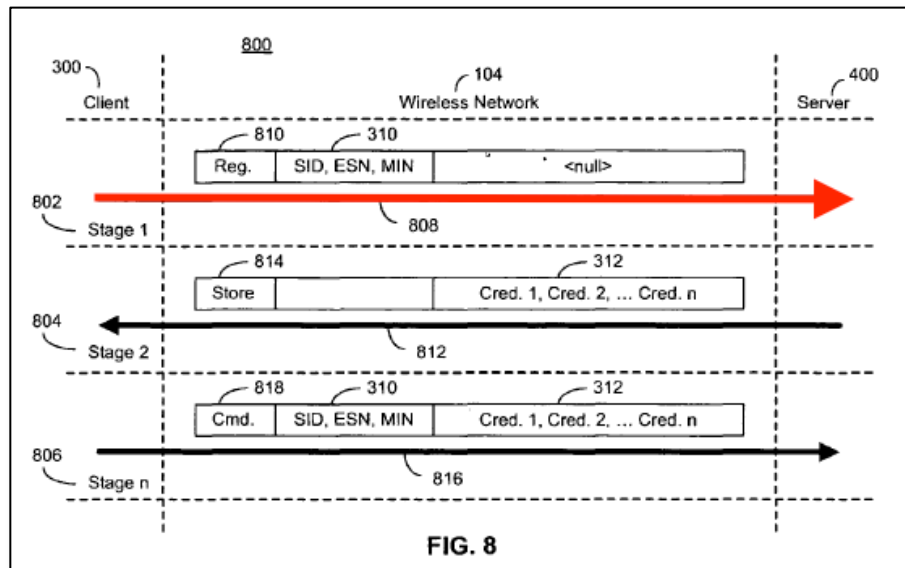


EX1004, FIG. 2. As FIG. 2 shows multiple similar wireless devices 102 (e.g., mobile phones 102), Lundblade teaches a *first device in a plurality of similar wireless end-user devices*.

113. Additionally, Lundblade teaches a “client computing device 300” that is “operable to provide secure communications with a remote server,” which may be “any of the wireless devices 102, 108, 110, 112 described in and shown in FIG. 1.” EX1004, 11:59-61, 15:63-65. Wireless devices 102, 108, 110, 112 include a mobile phone 102, PDA 108, pager 110, and computer platform 112 “that has a wireless communication portal.” EX1004, 10:26-32. Lundblade also teaches the secure

communication method being performed on any “form of remote module including a wireless communication portal,” including, e.g., personal computers, telephones without a display or keypad, etc. EX1004, 10:32-37, 11:61-12:3. Thus, Lundblade teaches client 300 being other wireless devices (e.g., laptop, PDA), and because all these wireless devices similarly are a “remote module including a wireless communication portal,” the *plurality* of devices are *similar*.

114. Lundblade FIG. 5 illustrates a method for obtaining credentials 312 for secure communications between client device 300 and a server 400. EX1004, FIG. 5, 16:49-59. As part of method 500, the client 300 is activated and, in response to being activated, retrieves and sends characteristics 310 to the server 400 (steps 506-510). EX1004, 16:49-59. Specifically, FIG. 8 illustrates an exemplary process 800 of communicating signals to carry out the secure communications method 500, which begins when “the client computing device 300 transmits a signal 808 containing a registration command 810 and characteristics 310 of the client computing device 300.” EX1004, 18:14-17 (step 802).



EX1004, FIG. 8 (annotated to show registration request signal in red). Signal 808 includes characteristics 310 and is sent from client 300 to 400 in the same manner step 508 of method 500 sends characteristics 310 to server 400. Thus, a POSITA would have understood process 800 implements method 500, and shows the actual signals transmitted/received to carry out method 500 of establishing secure communications. See ¶ 63.

115. A POSITA would have understood signal 808 is a *request* because client 300 is originating a communication for registration to server 400, thereby *requesting* the registration. As shown above at FIG. 8, signal 808 originates from client 300 and is sent to server 400 as the first signal in the secure communications process 800. EX1004, FIG. 8, 18:14-17. Thus, because the registration command originates from the client 300, client 300 is *requesting* the registration. Registering devices to server-side elements commonly involved a registration request from the

device in order to initiate communication. *See, e.g.,* EX1034 (3GPP TS 24.229, 9 (describing a user equipment (i.e., device) sending a “REGISTER request” as an initial registration and the server-side sending back a final “response”); EX1035 (Arnone), [0071] (generally describing a client sending a “registration request” to a “secure web application server”), [0073] (describing the web application server generating a “registration response for the requesting user”). Additionally, Lundblade teaches the server 400 responding to the registration signal 808, further indicating the signal 808 is a *request* allowing the server 400 to respond appropriately. EX1004, 18:17-21.

116. Additionally, the registration request signal 808 of Lundblade is a *request for a secure device credential*, because the signal 808 is a request for the server 400 to perform the action of providing credentials 312 for secure communication. Indeed, Lundblade teaches server 400 generating credentials 312 “**in response**” to the registration request, and thus the registration request signal 808 is requesting credentials 312 to be generated for secure communication. EX1004, 18:17-21, 14:59-63. Registration requests (e.g., registering to a network, activating a device) were known to serve as requests for credentials, particularly where credentials are sent to the device in response to the request. *See* ¶ 45. For example, Arnone’s secure communication method includes a registration request in response to which a server provides credentials. EX1035, [0073] (describing the web

application server response to the registration request including a “unique login value and a password value combination”). Additionally, Camenisch’s computing device sends a registration request to a registration computer, which then generates a “device related credential” in response to the registration request. EX1028 (*Camenisch*), [0007], [0016]. Additionally, the ’055 Patent describes a “credential request from end-use device 100” as including information identifying the device (e.g., device identifier). ’055 Patent, 38:24-49. This is similar to Lundblade’s registration request including Reg 800 and characteristics 310 (e.g., SID, ESN) identifying the device. EX1004, 18:14-17 (step 802), 12:29-42 (describing characteristics 310 as identifying the device). Thus, given registration requests were known to serve as a request for a device credential, Lundblade expressly teaches server 400 generating a credential in response to receiving the registration signal 808, and Lundblade’s registration request is much like the ’055 Patent’s described credential request, a POSITA would have understood the signal 808 is a *request for a secured device credential*.

117. Characteristics 310 are included in signal 808. EX1004, 18:14-17, FIG. 8. The server 400 generates multiple credentials 312 based on the characteristics 310--a unique client key 409 (to form credential 320) and a generic key 410 (to form credential 322). EX1004, 12:23-29 (referring to credential 320 and 322 collectively as “credentials 312”), 16:59-17:3. The server uses any of a variety of cryptographic

algorithms to produce each credential. EX1004, 12:51-58. For example, Lundblade teaches credential 320 being formulated “by performing a one-way cryptographic operation on characteristics 310 of the client computing device 300 and the client key,” and teaches other credentials (e.g., credential 322) being generated “using the same or different cryptographic algorithms.” EX1004, 12:51-58. Cryptographic algorithms secure underlying data. *See, e.g.,* EX1033 (*Kurn*), [0041] (describing “cryptographic techniques” to provide security of transacted data to ensure the data is kept “private, unmodified, and provable”); Ex1036 (*An Introduction to Computer Security – The NIST Handbook*), 1 (describing cryptographic algorithms as an “important tool for protecting information and is used in many aspects of computer security”). One-way cryptographic algorithms, such as, e.g., hashing, were implemented to secure data. EX1037 (*Secure Programming*), 151 (describing “cryptographic algorithms” being “necessary to keep a system secure”), 155 (describing one-way cryptographic “secure hash algorithms” that provide data security). Thus, Lundblade teaches each credential (i.e., credential 320 and credential 322, either collectively (referred to as credentials 312) or individually) is a *secure device credential* because they are generated through cryptographic algorithms, such as a one-way hash function. Once credentials 312 are generated, the server sends credentials 312 to client device 300 where they are stored in system memory 302. EX1004, 17:4-10.

3. ***1[b]: “performing a verification step of one or more device identifiers supplied by the first device, and, upon a successful verification in the verification step,”***

118. Lundblade in view of Jobst teaches performing a verification step of the characteristics 310 provided by the client device 300 prior to generating and credentials 312, as discussed below.

119. Lundblade teaches that the client computing device 300 transmits device characteristics 310 to the server 400 alongside the registration command 810 within signal 808. *See* Claim 1[a]. The characteristics 310 include one or more device identifiers, such as a hardware ID (e.g., electronic serial number (ESN), system identification number (SID), etc.) EX1004, 12:29-42. Indeed, Lundblade teaches the characteristics 310 are “useful *in uniquely identifying the particular client computing device 300.*” EX1004, 12:43-51. Hardware identifiers such as an ESN and a system identification number (SID) were known to be device identifiers that uniquely identify a particular device. *See*, ¶ 46; *see also*, EX1038 (*Roka*), 1:29-31 (describing device IDs for a mobile device that “unique identifies the mobile device, such as an Electronic Serial Number (ESN)). Lundblade’s characteristics that identify the client 300 (e.g., ESN, SID) are similar to the ’055 Patent’s “device identifier (e.g., any identifier associated with the end-user device).” EX1001 (’055 *Patent*), 38:28-31. Thus, a POSITA would have understood that characteristics 310

such as the SID and ESN are unique to the client device 300 and constitute *one or more device identifiers supplied by the first device*.

a. Jobst's Teachings

120. Jobst teaches verifying a device identifier (i.e., IMEI of a phone) at a server by comparing a received IMEI from the phone to a database table of IMEIs stored at a certification server to determine if the requesting device has an open account. EX1005, 7:15-22. For example, Jobst teaches, after receiving a request message from a phone, a certification center 35 checks the “validity” of the request by checking in a database table if the phone has “an open account.” EX1005, 7:15-20. The database table contains IMEI codes that are compared with the received IMEI by the server to determine whether the phone should have access to software. EX1005, 7:20-22. This verification procedure is much like the '055 Patent, describing verifying the device identifier from a table of allowed device IDs on the server. EX1001, 46:62-65 (describing the “device identifier” being verified from a “device table” on the server that “contains a list of valid devices” that are allowed to connect to server). Thus, Jobst's IMEI verification is a *verification step*. When the IMEI is verified, Jobst's certificate server calculates and outputs a phone password (i.e., device credential) based on the IMEI code and a master password. EX1005, 7:23-32 (disclosing “if the certification center 35 deems the request message 36 to be valid the certification center 35 then calculates a phone password 40”).

121. Jobst teaches a mobile phone that downloads an application or program segment for an application from a software provider. EX1005, 6:57-7:4. Jobst also teaches that the mobile phone may receive an instruction that “activates an application that has been present in the phone as software code.” EX1005, 8:45-49. The IMEI checked for request validity “is a universal code unambiguously identifying the phone.” EX1005, 7:34-35. If the request is deemed valid, the certification center produces the phone password 40 that can be used to download code from the software provider. EX1005, 7:23-8:27.

122. As discussed below, Lundblade’s process of providing secure communication would be modified to include the verification step, per Jobst. Jobst teaches *performing a verification step of one or more device identifiers supplied by the first device* by verifying the IMEI supplied by the phone. In the combination, any of the device identifying characteristics 310 contemplated by Lundblade, for example, a SID or ESN, is compared to database entries on the server 400, as described by Jobst, to determine if the device has permission to use the network. If the characteristic matches a characteristic stored in the server 400 database (e.g., a device ESN matches an ESN in the database), the process of Lundblade continues, and the device is provided with credentials to perform secure communications using the network. In the modified system, server 400 only generates credentials 312 for client 300 when a successful verification occurs (i.e., when the received

characteristic matches one in the authorized device list on server), as Jobst only calculates the phone password (i.e., a device credential) when the device ID is validated. EX1005, 7:23-25. Thus, a POSITA would have understood credentials 312 for the client 300 are only generated *upon a successful verification in the verification step*, in the same manner that Jobst only calculates a phone password if the IMEI is valid.

b. Motivation to Combine Lundblade and Jobst

123. A POSITA would have been motivated and found it obvious to modify Lundblade's secure communication method to verify a characteristic 310 of the device at server 400 by determining if the characteristic 310 matches a value stored in a database, as taught by Jobst. EX1005, 7:15-22. The modified system only generates credentials 312 after successful verification. Such a modification is nothing more than use of a known technique (i.e., verifying that a device has permission to access a network and/or server by verifying the device identifier against stored identifiers at a server) to improve a similar method (providing credentials to wireless devices) in the same way; i.e., by ensuring only authorized individuals are able to set up secure communications with the server 400.

124. A POSITA would appreciate that it is desirable to verify whether the client 300 is permitted to gain access to the network system prior to providing credentials, in order to prevent providing credentials to unauthorized devices. It is

desirable to ensure client 300 is permitted to gain access to the network system (e.g., communicate with server 400 via network 104 and carrier network 200), to keep fraudulent devices and unauthorized users from gaining access to the network. EX1016 (*Giles*), 4:53-63 (describing checking received parameters, such as IMEI, against parameters stored in the database to “limit access to the network 114, for example, to a computing device paired to a user by requiring that a given user only accesses the network when using a given device”). Indeed, keeping unauthorized users from gaining access to a network was a known problem. *See* ¶ 41. For example, given Lundblade’s exemplary network system contains a carrier network 200, a POSITA would appreciate ensuring the device requesting registration is actually permitted to use the network to communicate with server 400. *See, e.g.*, EX1038 (*Roka*) (generally describing checking the device ID (i.e., IMEI) of a mobile phone on the server side); EX1005 (*Jobst*), 2:27-33 (generally describing carrier networks, such as GSM, were known to check the IMEI code of a phone “in order to identify the phone”). If such a verification step is not used, then any device (including one unauthorized) would be provided credentials 312, which would compromise the network. *See* ¶¶ 41-42 (explaining the importance of ensuring only authorized people gain access to a network). Thus, a POSITA would appreciate including a verification step to ensure unauthorized devices do not receive credentials to perform secure communications over the network. Additionally, a POSITA would appreciate

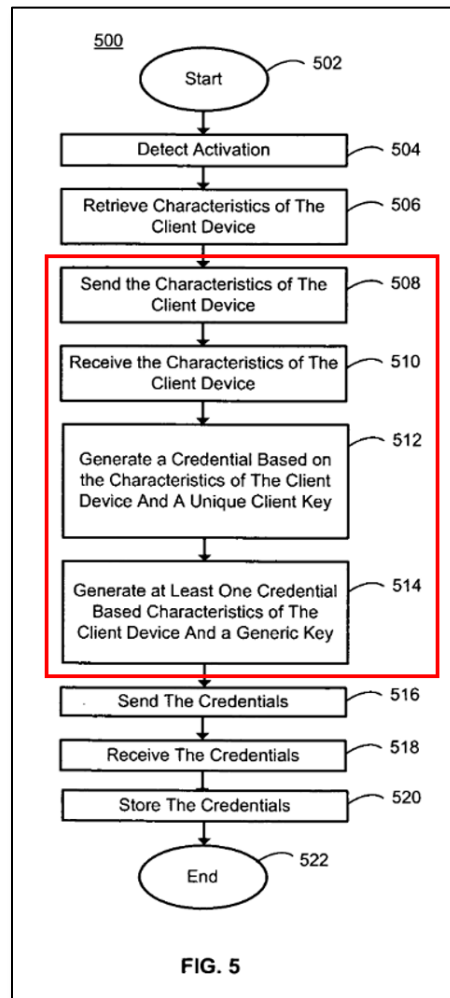
requiring verification so as to prevent wasting network resources by stopping the server 400 from generating credentials 312 for an unauthorized device.

125. A POSITA would have had a reasonable expectation of success in making the modification, given that Lundblade teaches server 400 having memory 402 that stores device characteristics 310 and that the client device 300 already sends characteristics 310 to the server 400 in the registration request 808. EX1004 (*Lundblade*), 15:11-15, 16:57-59, 18:14-17. Thus, the modification requires only programming the server 400 to contain additional logic to check the received characteristics (e.g., ESN, SID, etc.) against characteristics stored in the database to verify the identity of the client device. For example, server 400 may be modified to contain a list of characteristics of the clients 300 permitted to communicate over network 104 and carrier network 200. Then, once a client 300 requests credentials 312 at server 400 during registration, server 400 verifies the received characteristics 310 with the stored permitted device characteristics to ensure the client 300 is authorized to communicate over network 104 and 200. This modification would have been simple to a POSITA, requiring no undue experimentation, as verifying the identity of wireless devices trying to communicate over a network was a conventional server/database function. *See, e.g.*, ¶ 47. For example, Gile's network connection process includes matching a received "parameter," such as IMEI, with an IMEI stored in a database. EX1016, 4:53-63. Additionally, Roka's verification

system for verifying network credentials compares a received device ID (e.g., ESN, IMEI) with stored device IDs to ensure the device is authorized to access the network. EX1038, Abstract, 4:23-35.

4. *1[c]: “generating a secure device credential for the first device,”*

126. Lundblade teaches that the server generates credentials for the client (i.e., *for the first device*). After the client device 300 sends the characteristics 310 to the server 400 via the network, “the server 400 generates a credential 320 based on the characteristics 310 of the client computing device 300 and a unique client key 409[,]” and “at least one credential 322 based on the characteristics 310 of the client computing device 300 and a generic key 410.” EX1004, 12:66-13:5. FIG. 5 illustrates this process.



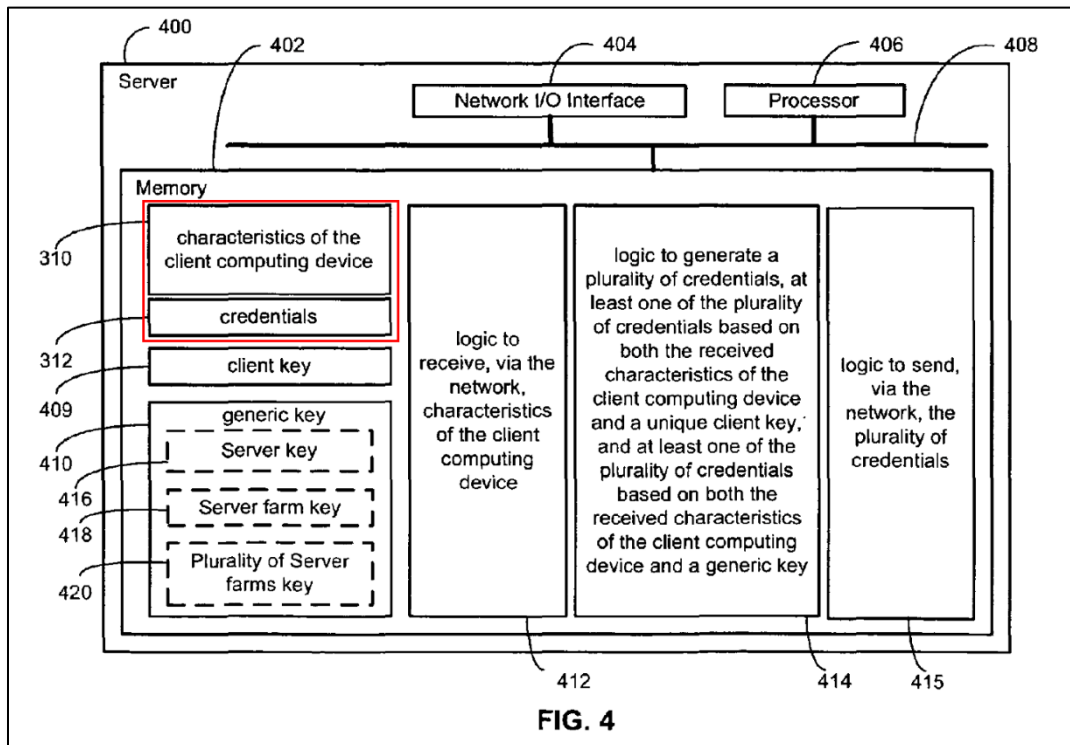
EX1004, FIG. 5 (annotated). As discussed above, credentials 320 and 322 are referred to together in Lundblade by credentials 312. *See* Claim 1(a).

127. Additionally, as discussed above, Lundblade’s credentials 312 are generated by using cryptographic algorithms. *See* Claim 1(a) (*citing* EX1004, 12:51-56). Lundblade's use of a one-way cryptographic algorithm would generate a hash, given one-way cryptographic algorithms were known to be hashing algorithms. *See* ¶ 117. This is consistent with the '055 Patent describing a “device credential” as being a “hash.” EX1001 ('055 Patent), 45:58-62, 38:33-36. Given cryptography was

one known way of securing data and/or information, Lundblade teaches the credentials 312 are *secure device credential[s]*. See ¶ 117. Additionally, because the credentials 312 are generated using characteristics 310 of client 300, the credentials 312 each are a *device credential*. EX1004, 16:61-17:3. Thus, Lundblade teaches *generating a secure device credential for the first device*.

5. *1[d]:“associating the secure device credential with one or more device identifiers of the first device, and”*

128. Lundblade in view of Jobst teaches associating the secure device credential with one or more device identifiers of the first device. Lundblade teaches that the generated credentials 312 for client device 300 and the characteristics 310 are stored in the server memory 402. EX1004, 15:11-22 (“located in memory 402 is characteristics 310 of the client computing device 300, credentials 312,...”); EX1001 (’055 Patent), 47:5-8 (“**service controller 122 stores the device credential** in database cluster 116 **as associated with** the subscriber identifier and **the device identifier** associated with end-user device 100.”). FIG. 4 shows the characteristics 310 and credentials 312 stored in the same memory 402:



EX1004, FIG. 4 (annotated to show credentials 312 and characteristics 310 stored in memory 402).

a. Jobst's Teachings

129. Jobst teaches storing a password unique to a device in a server record and addressing the record with the device's IMEI code so that a new password does not need to be calculated every time the device requests software from the server:

Alternatively the certification center may calculate the phone password when the account is established and *store the password in the record that may be addressed by the IMEI code*. Then there will be no need for calculating the phone password every time new software is requested.

EX1005 (*Jobst*), 9:11-15. Jobst's storage of the phone password (i.e., device credential) and addressing the stored password "by the IMEI code" indicates the phone password is stored in association with the IMEI. Additionally, given that the phone password is stored "in the record" and is addressed by the IMEI, the "record" includes the IMEI and phone password, thereby associating them with one another. This is similar to the '055 Patent, which stores the device credential at the server "as associated with" the device identifier "associated with the end-user device 100." EX1001, 47:5-8. Thus, a POSITA would have understood in addressing the phone password via the IMEI code when stored, the phone password is *associate[ed]* with the IMEI.

130. In the modification, and as discussed below, the Lundblade-Jobst network system associates the credentials 312 with the characteristics 310 (e.g., ESN, SID) in server memory 402 (i.e., *associating the secure device credential with one or more device identifiers of the first device*) in the manner described by Jobst, as addressed by the characteristics. This provides the benefit of Jobst, in that the server does not need to generate and provide new credentials 312 every time the client device 300 asks to establish secure communications with the server; rather, the credentials can be addressed knowing only the device characteristics 310.

b. Motivation to Combine Lundblade and Jobst

131. A POSITA would have been motivated and found it obvious to modify Lundblade to store the credentials 312 in a record addressed by the characteristics 310 in the manner discussed by Jobst, such that credentials 312 are stored in association with the characteristics 310. EX1005, 9:11-15. Such a modification is nothing more than the use of a known technique (i.e., storing device credentials in association with device identifier on server side) to improve similar devices (network systems) in the same way by creating an organized, streamlined storage system and reducing data required to be communicated from the client. Lundblade teaches various clients 300 (e.g., mobile phone 102) trying to communicate via network 104 and carrier network 200 with server 400, indicating server 400 would have multiple credentials 312 stored for each of the various devices. EX1004 (*Lundblade*), FIG. 2, 11:59-12:5. Additionally, given each client 300 is provisioned multiple credentials 312 (credential 320 and 322), this further increases the load of credentials 312 stored at server 400. EX1004, 16:61-17:3. Given these numerous credentials, a POSITA would have appreciated the benefits of indexing the credentials in some way within memory 402 to organize the various credentials 312. Indeed, storing credentials 312 with their associated characteristics 310 would have made it easier to look through and identify which credentials 312 are for which clients 300. A POSITA would appreciate that this would make it easier for an administrator of server 400 to view the stored data, such as when dealing with a

potential security breach. Thus, a POSITA would have been motivated to store credentials 312 in association with the characteristics 310 to make the various data more streamlined and easily identifiable for each client 300.

132. A POSITA also would have appreciated this association provides a simplified method to identify compromised devices. Lundblade teaches that the secure communication system is able to determine when device credentials 312 have been compromised and to generate new credentials 312 in response. EX1004, 16:29-33. Lundblade also contemplates that devices themselves can become compromised. EX1004, 3:30-33. Additionally, devices becoming compromised was a known problem, such as through hacking, duplication by unauthorized individuals, or infection with viruses. *See* ¶ 41. This is particularly an issue with remote devices that perform wireless communication. *Id.* Given that devices becoming compromised was a known issue and Lundblade understands the issue of compromised credentials, a POSITA would have appreciated that associating credentials 312 with the corresponding characteristics 310 in the server memory 402 allows a network operator to identify the device associated with compromised credentials at server 400. The network operator would not have to guess at what device is attached to which credentials in memory 402, allowing network operators to more quickly identify the compromised device using the credentials 312 and deactivate/fix the compromised client 300.

133. Additionally, a POSITA would appreciate storing the credential 312 with the associated characteristics 310 removes the need to send characteristics with each signal. Lundblade teaches that the secure communications sent between the server 400 and client 300 after generating credentials 312 always include the characteristics 310 in addition to the credentials 312. EX1004, 18:30-32. However, as was well understood, communications could be hacked or snooped by a nefarious individual, who could then gain access to both the characteristics 310 and credentials 312 to spoof the client 300 and gain unauthorized access to the network. *See, e.g.,* EX1017 (*Brown*), [0004]; EX1039 (*Sato*), [0007] (describing if the “device ID” of a phone has become compromised (i.e., is known to a nefarious individual), the “possibility” of the device being “hacked” is “large”). Thus, given such a problem, a POSITA would appreciate that storing the credential 312 in association with the characteristics 310 at server 400 keeps the client 300 from having to send characteristics 310 every time it communicates with server 400, thereby reducing the risk that the characteristics 310 could be intercepted by a bad actor and be replicated by an unauthorized user (e.g., a hacker trying to pose as the authorized client 300). Rather, the client 300 would send only the credentials 312, without needing to send characteristics 310 along with credentials 312 in subsequent communications. Additionally, storing the credentials 312 in association with the characteristics 310 in server 400 also provides the benefit of reducing the message

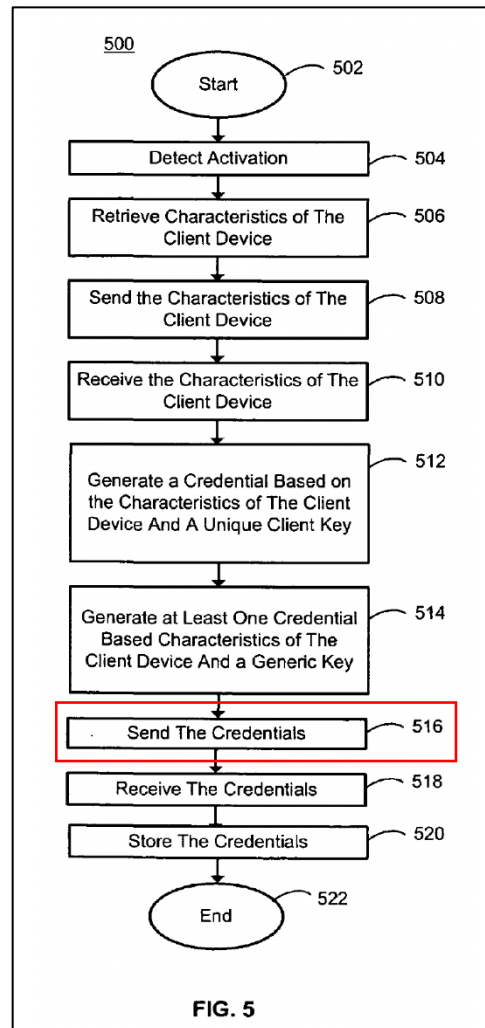
length of the subsequent signal (e.g., signal 816), because client 300 does not have to send characteristics 310 every time it communicates with server 400. EX1004, 18:27-30. This reduced message length reduces overall power consumption of client 300 when sending the signals.

134. A POSITA would have had a reasonable expectation of success in making this modification, given Lundblade already teaches that server 400 stores device characteristics 310 and credentials 312 in the same memory 402. EX1004, 15:11-22. Thus, all that is required for the modification is a simple change in the programming of the server 400 to contain additional logic to associate the credential 312 with the corresponding characteristics 310 in memory 402. For example, the server 400 would be programmed to store credential 312 directly with characteristics 310 or would be programmed to use a pointer or index (i.e., some way of “addressing”) to link credentials 312 with characteristics 310 in memory 402. This modification would have been within the skill of a POSITA and done without undue experimentation, as networks were known to store device credentials with device identifiers. *See* ¶ 48. Additionally, associating data with each other within a memory was well-known. *See, e.g.,* EX1040 (*Introduction to Programming*), 1-3 (stating pointers as being well-known for indicating the “address of some location in memory”).

6. 1[e]: “securely sending the secure device credential to the wireless end-user device;”

135. Lundblade teaches *securely sending the secure device credential to the wireless end-user device*. Namely, Lundblade teaches that in step 516 of FIG. 5 the server 400 sends the credentials 312 to the client computing device 300 “under a secure mode” to “prevent unwanted snooping of data by unauthorized programs and unauthorized entities.” EX1004, 17:4-20. For example, in discussing the background of the technology, Lundblade discusses how vulnerabilities can arise in systems “when both the cryptographic algorithm and the data used by such algorithms are freely available[.]” EX1004, 2:34-36. Indeed, prior systems that contained the credential with the cryptographic algorithm on the device to generate the secure device credential were known to be vulnerable to credential forgery, given someone could then hack a single device and have all necessary information to forge the secure device credential. EX1004, 2:30-41; *see also*, EX1041 (*Cellphone Security Far From Airtight*) (generally describing that when a hacker can obtain the secret cipher and SIM data, the phone can be impersonated by reproducing the credentials). Thus, to protect credential provisioning from such vulnerabilities, Lundblade employs the “secure mode” when sending credentials 312 from server 400 (step 516), as well as during other steps (e.g., steps 506, 508, 510, 518, 520). EX1004, 17:4-15. The secure mode is accomplished in “any number of currently known

techniques known in the art,” including the use of a “secure file system.” EX1004, 17:13-20.



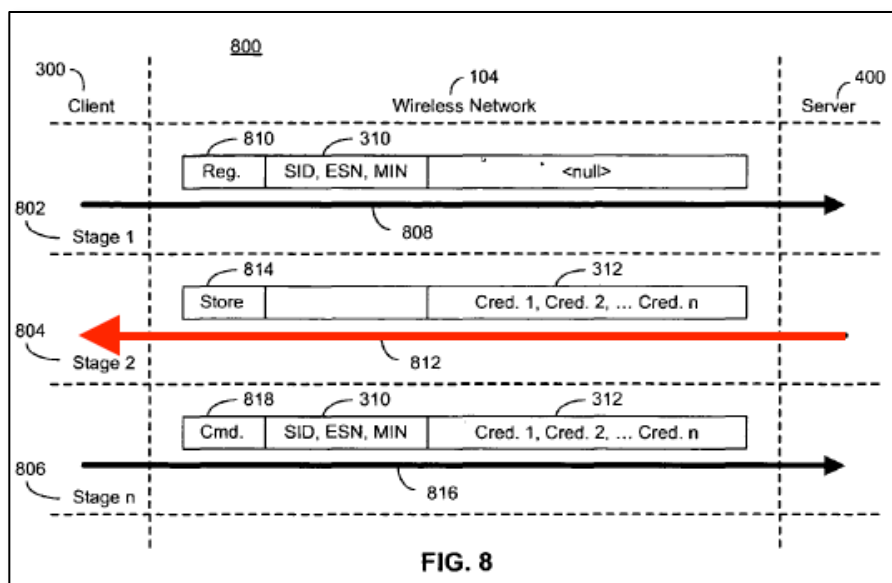
EX1004, FIG. 5 (annotated). A POSITA would have understood that this “secure mode” thus includes performing the cryptographic algorithm on data protected by the secure file system to prevent the vulnerabilities Lundblade is concerned with. Indeed, the secure file system protects the credentials 312 when stored at client 300, and performing the cryptographic algorithms on the protected credentials 312 when

sent secures the credentials 312 in-flight, to reduce the risk of a nefarious individual accessing the credential in order to spoof it. EX1004, 2:34-41. Otherwise, when the data is transmitted (and thus unprotected by the secure file system), a malicious actor could intercept the signal and spoof the credentials 312. This combination of encryption at rest and encryption in flight was well known and commonly employed. EX1033 (*Kurn*), [0041]; Ex1036 (*An Introduction to Computer Security – The NIST Handbook*), 1 (describing cryptographic algorithms as an “important tool for protecting information and is used in many aspects of computer security”).

136. Furthermore, since step 516 (“send the credentials”) is included in the list of steps performed in the secure mode (*id.*, 17:13-15), a POSITA would have understood that the sending would be performed via the secure mode, such as by “secure communications” enabled by the “cryptographic (encryption) algorithms” Lundblade discusses in this context. EX1004, 2:13-17, 2:30-41. Indeed, Lundblade himself indicates the “strong” security of providing “secure communications” by implementing cryptographic (encryption) algorithms. EX1004, 2:13-17. In the background of the technology, Lundblade discusses providing “secure communications” between clients and servers “using secure cryptographic key information,” further indicating the use of encryption (via cryptographic algorithms) when sending information. EX1004, 1:52-56. Additionally, cryptographic algorithms were known to secure data to provide secure transmissions and are often

referred to as a “secure mode” of communication. *See* ¶ 140. Thus, a POSITA would have understood Lundblade’s disclosure of sending credentials 312 from server 400 under a secure mode to teach transmitting the data in encrypted form.

137. Process 800 (implementing method 500) sends a signal 812 including credentials 312 to client 300 from server 400, as shown below in FIG. 8:



EX1004, FIG. 8 (annotated to show response containing credentials 312 sent from server 400 to client 300).

138. As discussed above, a POSITA would have understood the process of FIG. 8 carries out method 500. *See* ¶ 114. Accordingly, FIGS. 5 and 8 illustrate the same embodiment, with FIG. 8 showing signals sent back and forth to perform method 500. *Id.* Additionally, a POSITA would have understood FIG. 7 also illustrates the same embodiment as FIGS. 5 and 8 of providing secure communication, but illustrates a specific portion of the process; namely, what occurs

after the credentials are provided to client 300 and used by client 300 in creating secure communications with server 400. EX1004, 17:46-48 (method 700 is directed to the portion of the method for providing secure communications of “sending of a message containing the stored credentials 312”). Indeed, FIG. 8 shows signals sent from client 300 to perform method 700 (e.g., signal 816). EX1004, 18:23-35. Thus, because process 800 of FIG. 8 illustrates signals sent to perform methods 500 and 700 of FIGS. 5 and 7, a POSITA would have understood that FIG. 5, FIG. 7, and FIG. 8 depict the same embodiment at different levels of granularity of various steps in the secure communication process.

139. A POSITA would have understood that transmitting the credentials 312 to the client device 300 using a “secure mode” constitutes *securely sending the secure device credential to a wireless end-user device*. Indeed, Lundblade expressly teaches using a “secure mode” for step 516 to “prevent” programs “and unauthorized entities” from snooping on the secure communication establishment. EX1004, 17:13-17. Furthermore, Lundblade understands the importance of keeping credentials 312 secure. For example, cryptographic algorithms are used to generate credentials 312. EX1004, 12:51-58. Thus, because sending the credentials 312 is performed in a “secure mode,” Lundblade teaches *securely sending* the credentials 312.

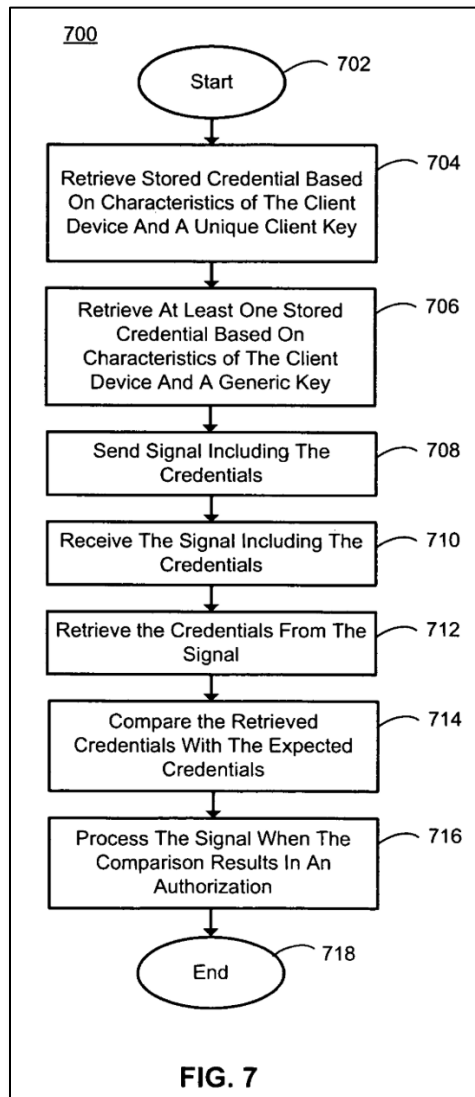
140. Further, a POSITA would have understood there were many commonly used secure modes of transmission known in the art (e.g., encryption). *See* ¶ 49. Encryption was understood to be a “secure mode” of transmitting data. *See, e.g.,* EX1042 (*Weinans*), 2:18-26 (describing the well-known Bluetooth standard for wireless communication as invoking “encryption” if “secure mode is used,” including an access point where “secure mode communications” are offered); EX1043 (*Waldron*), 1:15-23 (describing “secure communication” using “terminals” (e.g., phone) as “usually operat[ing] in both clear and secure modes”), 1:37-40 (describing terminals operating “in their secure modes, and communications are encrypted for transmission and decrypted upon receipt”). Indeed, encryption prevents eavesdroppers or interceptors of a communication, including the government, from reading that communication.” EX1044 (*Telecommunications*), 327. Thus, a POSITA would have understood Lundblade’s “secure mode” would be operated through any of the various well-known security techniques, including, e.g., encryption. EX1004, 17:18-20.

7. 1[f]: “based at least in part on the secure device credential, negotiating a secure message link between a message link server in the network system and a device link agent in the first device;”

141. Lundblade teaches using the credentials to negotiate a secure message link between the server 400 and the server security module of the client device.

142. Lundblade teaches that after receiving the credential 312, subsequent signals sent between the client device 300 and the server 400 include the credentials for secure communications. EX1004, 18:23-35 (disclosing, after receiving credentials 312 from server 400, client 300 then includes credentials 312 in subsequent messages to create a “standard secure signal” that “are each authorized at server using” credentials 312), 14:42-45 (disclosing the “received plurality of credentials 312 are stored in memory 302” of client 300 for “subsequent inclusion in signals sent from the computing device”).

143. FIG. 7 illustrates a method for using the stored credentials to secure communications from the client device to a server:



EX1004, FIG. 7. As discussed above, FIG. 5, FIG. 7, and FIG. 8 all illustrate the same embodiment at different steps in the secure communication process and with varying granularity. *See* ¶ 138. In FIG. 7, the credential stored on the client device 300 is retrieved and included in the signal sent to the server 400. EX1004, 17:56-61. Once the signal is received by the server 400, the server compares the retrieved credentials with the expected credential, and if they match, the signal is considered authorized. EX1004, 17:61-18:5 (disclosing server 400 “then operates to compare,

in step 714, the retrieved credentials 312 with the expected credentials” stored on server 400 and, if there is a match, “the server 400, proceeds, in step 716, to process the associated signal as an authorized signal[.]”).

144. Furthermore, Lundblade describes FIG. 8 as “using signals to initiate secure communications” and “transmissions ... that establish and begin secure communications.” EX1004, 9:17; 18:12-14. A POSITA would have understood that *negotiating* a secure message link includes “initiat[ing]” or “establish[ing] and “begin[ning]” communications in the manner taught by Lundblade would be an example of *negotiating a secure message link*. Negotiation was commonly used in establishing secure communication. For example, TLS/SSL was a known secure protocol for providing secure communication via negotiating a secure link. *See, e.g.,* EX1045 (*TLS Protocol Version 1.0*), 4, 29 (generally describing TLS being used to “establish a secure connection” for secure communication between a client and server, with the process also “negotiating the strongest possible connection”).

145. Similarly, a POSITA would have understood that server 400 receiving credentials and determining if they are authorized, per FIG. 7, to be in line with the above example of *negotiating* a message link between the client and server *based...on the secured device credential*. EX1004, 17:56-18:5. This may be another step in the exemplary negotiating process of FIG. 8, or a different process. Either way, both FIG. 7 and FIG. 8 show exemplary *negotiating* of a link. The client 300

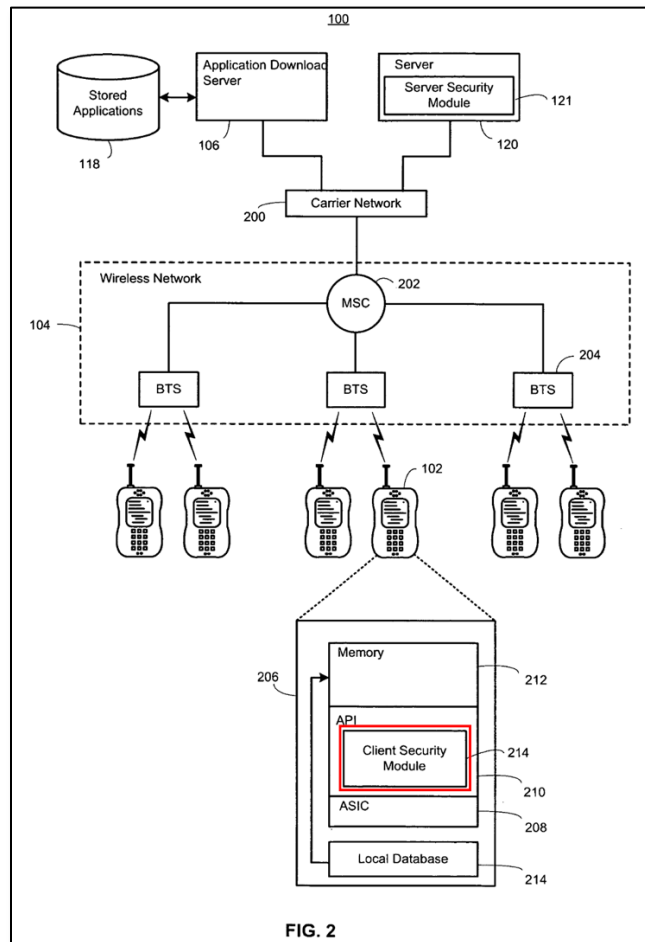
provides information (e.g., credentials) to server 400 so that server 400 can either authorize or deny communication between the two devices. EX1004, 18:7-13, 18:35. Because server 400 authorizes the communications based on the credentials 312, the server 400 and client 300 *negotiat[e] a secure message link that is based at least in part on the secure device credential*. By requiring a match between the received and stored credentials, the server can either authorize the signal or not. This ability of server 400 to deny signals that do not have credentials 312 matching the expected credentials indicates, when the credentials 312 actually do match, that a secure link is established. For example, the process of establishing a secure message link, such as via transport layer security (TLS), included exchanging a device credential (e.g., certificate) to authenticate the device in order to negotiate the secure connection. EX1045, 29; EX1013 (*Hind*), 10:39-65 (describing a device exchanging certificates to provide authentication under SSL/TLS protocol and, upon authentication, being passed a “link key” to establish a secure SSL/TLS link); EX1006 (*Hardjono*), [0101] (using certificates either exchanging by both client and server or just “one of the machines” having a certificate and using the certificate (or certificates) to “establish a secure channel”). If the received client credential cannot authenticate the device, then no secure link is created. Thus, credential exchange between a server and device is one way of “enabl[ing] the creation of a secure link.” EX1032 (*Matta*), 16:1-6. Lundblade expressly teaches server 400 verifying credentials 312 sent from client

300 to “authenticate” the device, to provide “strong authentication” between client 300 and server 400. EX1004, 1:10-12 (Describing the “secure client credentials” being used in “providing strong authentication between remote computing devices and servers”). Thus, by requiring a match between the received and stored credentials, the client and server of Lundblade exemplify *negotiating a secure message link*.

146. Lundblade’s FIG. 7 illustrates a method “for providing secure communications.” EX1004, 17:45-46. The method includes client 300 transmitting messages containing credentials 312 that are then verified by server 400 to authorize the message as authentic, if the credentials 312 match expected credentials stored at server 400. EX1004, 17:56-18:5. As discussed above, sending device credentials to authenticate a device creates a secure message link. *See* ¶ 145. Thus, given that Lundblade’s matching ensures the transmitted signal is authorized, a POSITA would have understood this transmission of secure messages between client 300 and server 400 to form a secure link so as to provide “secure communications.”

147. Lundblade also teaches that the client device 300 (cellular phone 102 of FIG. 2) includes a client security module 214 that contains “logic configured to provide for secure communications over the carrier network 200. Such client security module 214 operates in conjunction with a [server] security module 121 to provide secure communications.” EX1004, 11:34-39. Though Lundblade’s quote

states client security module 214 operating in conjunction with a “client security module 121,” a POSITA would have understood that 121 refers to the server security module as shown in FIG. 2 and is only mistakenly called the “client security module 121,” in the passage quoted. Lundblade is concerned with providing secure communication between client 300 and server 400, so the client security module 214 operates in conjunction with server security module 121 to provide such secure communications. “Further, the communications over the wireless network 104 are performed in a secure manner, at least in part, because of the interaction and operation between the client security module 214 and the server security module 121.” EX1004, 11:52-55.



EX1004, FIG. 2 (annotated).

148. A POSITA would have understood that the secured communications discussed in FIG. 7 are established between the client security module 214 and server security module 121 shown in FIG. 2. Indeed, Lundblade repeatedly teaches client security module 214 and server security module 121 as cooperating together to provide the secure communications. EX1004, 11:34-39, 11:52-55. Thus, when viewing FIG. 7's secured communication establishment (via verifying credentials 312 match), the messages are being sent between client security module 214 and

server security module 121. Thus, the *secure message link* is negotiated between the server security module 121 (part of a *message link server in the network system*) and the client security module 214, *a device link agent in the first device*. This is much like the '055 Patent describing an agent (i.e., “control device link 1691”) providing a secure communication link. EX1001 ('055 Patent), 15:1-2, 44:38-40.

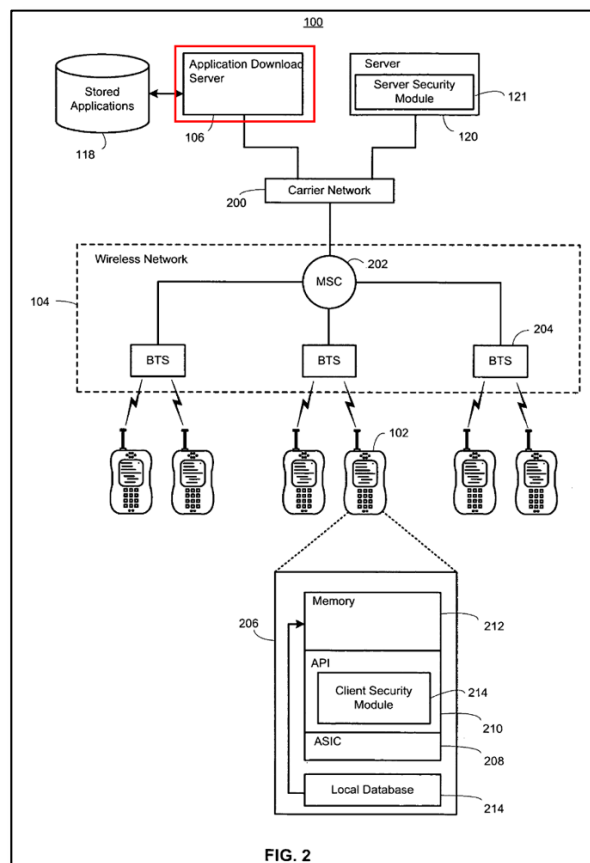
8. ***1[g]: “receiving, over the secure message link, a second request from the first device, the second request for a secured application credential, for any first application identified from a plurality of device applications registered to use wireless network communications, the second request comprising a general application credential for the first application; and”***

149. Lundblade in view of Hardjono teaches transmitting a request for access to a network resource that includes a digest (*general application credential*) to obtain a trust report (*secured application credential*) that allows access to network resources.

150. Lundblade teaches that the client device “can receive and execute software applications transmitted from the application download server 106[.]” and that “[i]n at least one embodiment the server 400 can be any of the servers 106, 120, 122 shown and described in relation to FIG. 1.” EX1004, 11:20-23, 14:63-65. Given server 400 may be any of servers 106, 120, 122, which include, e.g., an application download server 106, a POSITA would have understood that application download servers are one type of server that can perform secure communications with client

300 over the network. EX1004, 10:22-26 (disclosing “application download server 106 that selectively transmits software applications and components to wireless devices” across network 104). Indeed, Lundblade teaches “logic to perform secure communications (including server security module 121)” is located at “any one or more of the application download server 106, second server 120 and stand-alone server 122.” EX1004, 10:46-50.

151. FIG. 2 shows the various exemplary servers 106, 120 that may provide secure communications with client 300:



EX1004, FIG. 2 (annotated to show application download server 106).

152. Further, Lundblade recognizes its teachings are applicable to any communications between a wireless device and any other component connected via a wireless network. EX1004, 10:54-60 (disclosing system 100 is “merely exemplary” and “can include any system whereby remote modules, such as wireless client computing devices 102, 108, 110, 112 communicate over-the-air between and among each other and/or between and among components connected via a wireless network 104,” including wireless network carriers and/or servers). Thus, a POSITA would have understood Lundblade is not limited to the server functionality described and illustrated. FIGS. 1-2 merely show an example network system configuration, and a POSITA would have understood that other server functionalities could be included within the network system for providing secure communications.

153. Additionally, a POSITA would have understood that signals containing various types of content can be sent through the secure communications between the client device 300 and server 400 once the credentials are generated and stored on the client device. Lundblade broadly teaches “system 100” provides secure communication “for the deletion and reloading of software application components on a wireless device” in communication across network 104 with application download server 106. EX1004, 10:19-24. The application download server selectively transmits software applications and “components” to the wireless device 102 across wireless network 104. EX1004, 10:25-24-26, 10:45-55 (disclosing client

300 (e.g., mobile phone 102) downloading applications “such as games, news, stock monitors, and the like, from” the download server 106 via secure communications). However, Lundblade also teaches servers 106, 120, 122 provide “separate services and processes,” indicating communications for other “services” other than software download are contemplated as being performed via the secure communication process. EX1004, 10:38-43. Lundblade also teaches client 300 sends signals containing “some type of command 818,” the credentials 312 and “delivery data (not shown)” to server 400. EX1004, 18:26-35. Thus, given various commands are contemplated by Lundblade for sending via secure communications, and such signals also include various “delivery data,” a POSITA would have understood the secure communication process applies to any communications over network 104, such as, e.g., accessing a resource via applications.

a. Hardjono’s Teachings

154. Hardjono refers to “another embodiment” in various locations throughout the specification. There is no indication that the embodiments in Hardjono would not be implemented in at least one single embodiment.

155. Hardjono teaches a method to ensure a client device and its components are in a proper state (i.e. trustworthy and not tainted with malware) prior to providing access to network services and resources. EX1006 (*Hardjono*), [0008], [0006]. The components on the client device to be validated include hardware and “software

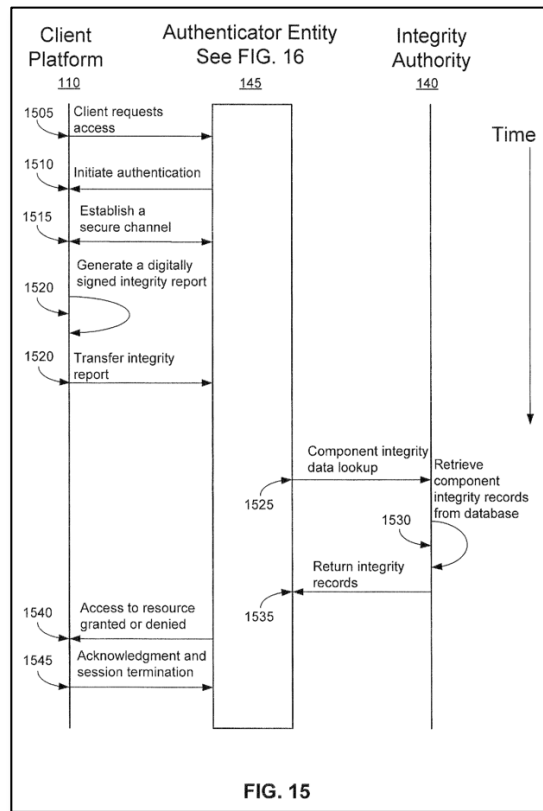
components.” EX1006, Abstract, [0036]-[0037] (disclosing component 160 on client 110 that include, e.g., “application software” such as word processor or e-mail). A client platform communicates with a server to obtain access to a network resource by transmitting an integrity report that must be verified by the network prior to the device gaining access to the resource. EX1006, [0009], [0030]. A client platform may be a personal computer 115, a wireless phone 125, a personal digital assistant (PDA) 130, etc. EX1006, [0030].

156. Hardjono teaches an authentication entity 145 and an integrity authority 140 that work in cooperation to validate the client 110 components and, according to the validation, either grant or deny access to the requested network resource or service. EX1006, [0031]. The resource 150 requested by client 100 “can be a service including a web service or application, a server, a network (such as a network internal to an organization, to which controlled access is provided), a printer, a flash memory disk or other solid state memory device, one or more storage devices,” etc. *Id.* Hardjono teaches the authentication entity 145 and integrity authority 140 “functions” being “combined and operated by the same physical machine.” EX1006, [0031]. Additionally, the authentication entity 145 comprises verification server 205, authentication server 210, and policy server 215, which all are “in a single place” and in a “single server with the functions of authentication server 210 and

policy sever 205² [sic] performed by verification server 205 alone.” EX1006, [0035]. Thus, authentication entity 145 (and all its verification, policy, and authentication components) and integrity authority 140 are on a single server.

157. Thus, Hardjono teaches a multi-step process whereby a device requests access to a resource from a server. *See also*, ¶¶ 75-80. The process starts with a device sending a request for a network resource or service to the authentication entity 145. EX1006, [0101], [0039] (disclosing platform verification “beginning with an initial request by client platform 110 for access to resource 150”). FIG. 15, below, shows the process being initiated by a client requesting access at step 1505:

² A POSITA would have understood “policy sever 205” to be referring to policy server 215, and is merely a typographical error.



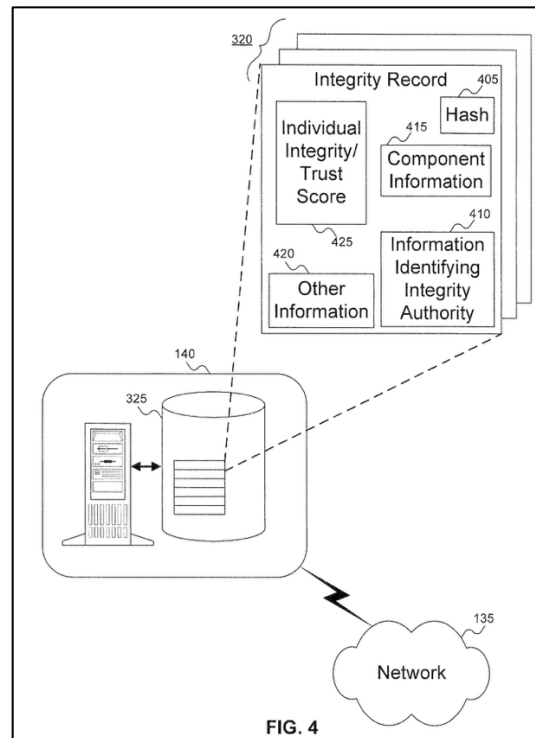
EX1006, FIG. 15. The “authentication server 210” of entity 145 responds to the request by “initiating the platform integrity verification.” EX1006, [0101]. To verify the integrity of client 110, client 110 sends an integrity report 170 to authentication entity 145 (step 1520 above). *Id.*

158. The transmitted integrity report 170 includes “digests” of software components 160. EX1006, Abstract, [0026], [0036]-[0038]. A digest is generated for an application on the device, as “digests can be generated for each executable application.” EX1006, [0037]. The digests in integrity report 170 also “include hashes of components 160,” using one-way cryptographic hash functions. EX1006, [0038], [0077] (disclosing the digests being “calculated as hashes 615 of components

160”). A POSITA would have understood that a digest teaches a *general application credential* because it uniquely identifies the application and includes a hash. Indeed, a digest is made for each executable application on client 110, meaning each digest is unique to each application. Additionally, Hardjono’s hash within the application digest in integrity report 170 is similar to the application credential described by the ’055 Patent as “a hash” and being “associated with the installed application.” EX1001 (’055 Patent), 57:44-48. Thus, because the digest uniquely identifies the installed application and contains a hash, the digest in integrity report 170 teaches a *general application credential*.

159. The authentication entity receives the integrity report 170 and compares the digests in the integrity report with integrity/trust records to determine a level of trustworthiness for the device. EX1006, [0104], [0027]. The integrity/trust records 320 are held by integrity authority 140 and contain digests (including a hash) of various software components that are harvested from the software source (e.g., manufacturer, software vendor). EX1006, [0045] (disclosing integrity authority 140 harvesting raw information into digests from software sources and storing each digest in an “integrity/trust record 320”), [0048] (disclosing the integrity/trust record 320 including a “hash 405” for the software). The digests in integrity report 170 are compared with the digests in integrity/trust records 320. EX1006, [0077]. The integrity report 170 digests are compared with digests in records 320 to “assess a

level of trustworthiness for client platform 110.” EX1006, [0079], [0104]. The information that can be included in the integrity record 320 is shown in FIG. 4.



EX1006, FIG. 4.

160. The authentication entity 145 then generates a trust report 220 and grants access to the requested resource. EX1006, [0104]. Specifically, the authentication entity 145 grants access to the requested resource if the trust report 220 contains a high enough trust score, which is based on the digest comparison. EX1006, [0104]. For example, if a digest in the integrity report 170 matches one in the records 320, a higher trust score 225 is calculated in trust report 220, indicating the software component is trusted (and has not been tampered with or otherwise compromised). EX1006, [0085] (disclosing differing hashes as denying access, and

thus hashes matching causes authentication entity 145 to grant access). Thus, if the digest matches, access is granted because the component is trustworthy. If the software component 160 is trusted, authentication entity 145 then grants access to the resource or service. Hardjono teaches providing the trust report to client, which allows the device to access other network resources or services. EX1006, [0041].

161. A POSITA would have understood multi-step requests (such as Hardjono's multi-step request (i.e., requesting the resource, server sending an authentication initiation message, and client sending a response—integrity report 170—for verification)) for resources are common, particularly where the resource being requested is access-controlled, such as by a password. Indeed, Hardjono's multi-step resource access is analogous to the well-known HTTP authorization. HTTP authorization is initiated when a client requests a protected resource from a server, such as Hardjono's client requesting access to a resource on a server. EX1046 (*Hypertext Transfer Protocol*), 36-37 (describing the resource request "GET" message sent from client); EX1006, [0101] (step 1505). The server then sends a client a challenge to provide its credentials for authentication. EX1046, 66 (describing server sending a "401 Unauthorized" challenge when the request "requires user authentication"); EX1006, [0101] (step 1510 where the server sends client a response to initiate authentication after receiving resource request). The client then responds to the challenge by sending its credentials (e.g., username and

password) to the server, much like Hardjono's client sending the integrity report 170 (which is signed with the device TPM certificate). EX1046, 107 (describing client sending an "authorization" response to the server challenge that contains "credentials," such as a password); EX1006, [0076], [0101] (step 1520 where client sends the integrity report 170 that is signed with the client's credential (i.e., certificate)). The server then verifies the sent credentials during HTTP authorization, and if the credentials are valid, grants the client access to the resource, as in Hardjono's comparison of the digests in integrity report 170 with the stored records 320, granting access to the resource if the digest in integrity report 170 matches. EX1046, 58 (describing a 200 OK message sent from server to client after the server verifies the credentials, that indicates the "request has succeeded," and the response may include the resource); EX1006, [0104] (describing sending back a "grant" result when the digests match).

162. The trust report may include information relating to the components 160, a signature, and a reference to the certificates of the device. EX1006, [0068]. For example, Hardjono teaches that the trust report 220 includes a reference "to TPM certificates of the client platform 110" and "a digital signature of verification server 205." *Id.* The trust report 220 also includes "a list of **one or more components 160, each with a reference to corresponding integrity/trust record 320**" that contains the digest (including the hash) of the software component harvested from the source.

Id. The authentication entity 145 (via verification server 205) signing the trust report 220 that contains a device credential (i.e., TPM certificate of client 110) and general application credential (i.e., matching hash of the software component 160) is much like the '055 Patent describing signing the application credential using the application credential and device credential. EX1001 ('055 Patent), 57:50-53.

163. A POSITA would have understood the multi-step request process of steps 1505 through 1520 (excluding step 1515) form a *second request from the first device for a secured application credential*, because the authentication entity generates a trust report 220 (i.e., *secured application credential*) in response to receiving a resource request and integrity report. As discussed above, to initiate the component authentication process, client 110 requests access to a network resource or service and sends the generated integrity report 170. *See* ¶ 158. Authentication entity 145 then responds by generating a trust report 220 based on the contents of the integrity report 170. Furthermore, multi-step requests for accessing a resource, much like Hardjono, were common, such as the HTTP authorization framework. *See* ¶ 161. Thus, because the trust report 220 is generated in response to receiving the resource request and the integrity report 170, a POSITA would have understood the combination of the steps 1505-1520 form a *second request*. Additionally, provisioning a secured application credential in response to a request for access to network resources was known in the art. *See* ¶ 51. For example, Lundblade '330's

request for accessing data on a network serves as a request for a secure application credential from a device. EX1009 (*Lundblade* '330), [0045]-[0046] (generally describing in order to access data, the application 210 must obtain an application credential), [0049] (describing generating an application credential that is secure because it hashes the device credential (i.e., master credential 204) with the general application credential (i.e., verified application ID 212)). Thus, the resource request and integrity report 170 serve as a *second request from the first device for a secured application credential*.

164. In the modified system, the step 1515 would be excluded because, as noted above at Claim 1[f], the secure channel would have already been established (as part of the “first request” from the device) as taught by Lundblade. *See* Claim 1[g] Motivation to Combine. Namely, because the Lundblade’s credential 312 provisioning and sending already creates secure communication via negotiating a secure link prior to requesting a network resource from server 400, the combination would not need to include Hardjono’s step 1515, because the resource request would be sent over the link using the established secure communication.

165. The second request (i.e., resource request and sent integrity report 170) comprises a *general application credential for the first application*, because the integrity report 170 includes a digest containing a hash of “at least one software component.” EX1006, [0095]. As discussed above, the hash of a digest in integrity

report 170 is unique to a certain application on client 110, which is much like the '055 Patent describing an application credential being a hash. *See* ¶ 158. Thus, the hash is a *general application credential for the first application*. Additionally, Hardjono teaches the verification process being performed for “at least one” software component installed on the client 110, indicating the verification process via authentication entity 145 is performed for a single application. EX1006, [0095]. Furthermore, Hardjono teaches trust report 220 that is generated from integrity report 170 including “a list of **one** or more components 160,” further indicating the integrity report 170 contains a digest having a hash for a *first application*, such as when a user is executing a specific application to request a network resource.

b. Combination of Lundblade and Hardjono

166. The Lundblade-Hardjono combination takes the secure communication between client device 300 and server 400 taught by Lundblade and incorporates the process for validating a software component of the client device, as taught by Hardjono. Specifically, as taught by Hardjono, the client device has multiple applications capable of requesting resources from the network (*a plurality of device applications registered to use wireless network communications*). *See* EX1006, [0037] (disclosing client 100 having email, word processor, among other applications installed thereon), [0029]. For example, Hardjono teaches client 110 having a word processor application, an “email package,” and a “spreadsheet”

application.” EX1006, [0037]. Additionally, Hardjono teaches records 320 harvested may be for “game software,” indicating client 110 including a game software for verification with record 320 to generate a trust report. EX1006, [0070]. Thus, Hardjono’s client has multiple applications installed (i.e., *a plurality of device applications registered to use wireless network communications*). A POSITA would have understood, when accessing a network resource via one of the applications (i.e., accessing email or accessing memory for certain network data), Hardjono would verify the application which is requesting the network resource. For example, Hardjono teaches the network resource 150 to be accessed may include, a memory, a network-protected application, such as email, or a printer. EX1006, [0031]. Given some of the examples of network resources 150, such as accessing email, a printer, or memory (e.g., for retrieving certain data), are accessed on client 110 via an application (e.g., printing doc from word processor, accessing email via “email package,” or retrieving data when completing a “spreadsheet”), a POSITA would have understood the single requesting application would be checked via authentication entity 145 to determine access to such a network resource. EX1006, [0037]. Additionally, accessing a network resource through an application was known to require authentication of the application. *See* ¶¶ 43, 52. Thus, when accessing a network resource via one of the applications (i.e., accessing email or accessing memory for certain network data), Hardjono would verify the application

which is requesting the network resource, and thus the *second request* is for *any first application identified from a plurality of device applications registered to use wireless network communications* (i.e., from a plurality of applications on client 110).

167. Server 400 of Lundblade incorporates all the functionalities of the authentication entity of Hardjono, allowing the server to grant or deny access to a resource. EX1006, Abstract. This includes the functionality of the authentication server 210, policy server 215, and verification server 205 of Hardjono.

168. As explained in Claim 1[Pre]-[f], the client device and server communicate to establish credentials for use in forming a secure communications channel (i.e., generate a secure link) for future transmissions between the client device and the server. *See* Claim 1[pre]-1[f]. In the combination, when one of the applications requires access to a resource (i.e., *second request*), for example, “an email package,” the device sends a message to the server including credentials 312 and the request for access to resources for the application (*receiving, over the secure message link, a second request from the device, the second request for a secured application credential*), comprising steps 1505-1520 (excluding 1515) as illustrated in Hardjono FIG. 15. The request sent to the server specifies the desired resource and includes an integrity report, and it is sent along the same link already established via secure communication. EX1006, [0101]. The integrity report includes the digest

containing a hash (*the second request comprising a general application credential*) for the specific application (e.g. an email package, or banking application) (*a first application*) that requires the resource. EX1006, [0037], [0044].

169. In the modification, the server 400 performs the integrity verification of the selected application by comparing the digest received with the digests stored on the server. EX1006, [0102]-[0104]. If the server determines the request is trustworthy (i.e., digests match with digest stored on server), the server makes a decision to grant the request and provide a trust report (*a secured application credential*) to the client device than can be used to access other related resources or services. EX1006, [0104], [0041]. The trust report includes information such as the digest, the relevant application, a digital signature of the verification server 205 and a TPM certificate associated with the device. EX1006, [0068], [0048]. In the Lundblade-Hardjono combination, the credentials produced in the process taught by Lundblade are used in concert with the TPM certificate. A POSITA would have appreciated using both the credentials 312 and TPM certificate in generating the trust report from the integrity report, to both attest to the application digest and to tie the digest in the integrity report to the client 300. The TPM certificate would attest to the accuracy of the application digest to ensure the digest reflects the application running on client 300, while credentials 312 tie the application digest (and integrity report) to client 300. Additionally, the trust report includes the signature of the server

400, which has the capabilities of verification server 205. Because the trust report includes the digital signature of the server, the trust report is *secured*. Indeed, digital signatures were known to allow a receiving party to ensure the message/document come from a secure and trusted source. EX1047 (*Digital Signatures*), 1 (describing a “popular form of digital cryptography is digital signature” because it “authenticat[es]” the message and/or document is from a trusted source); EX1044 (*Telecommunications*), 347. Thus, given that trust report 220 is signed by the verification server 205 (i.e., part of authentication entity 145), the trust report 220 is *secured*. Furthermore, as discussed above, the signing of the trust report 220 is much like the signing of the application credential described in the ’055 Patent. *See* ¶ 162.

170. Therefore, the combination of Lundblade and Hardjono teaches using secured communications to send a request to a network system for a trust report to access a resource (*a second request from the first device*) which includes a digest of a specific application (*a general application credential of [a] first application*). The application of the digest is one application of a plurality that is *registered to use wireless network communications*, as discussed above. *See* ¶ 166.

c. Motivation to Combine Lundblade and Hardjono

171. A POSITA would have been motivated and found it obvious to modify Lundblade’s network system to perform Hardjono’s software component verification. EX1006, [0101]-[0102]. A POSITA would have understood that this

modification is nothing more than combining prior art elements (i.e., Lundblade's secure network communication and Hardjono's software verification) according to known methods to yield predictable results of increasing the security of the network by verifying the integrity of software applications loaded onto client 300 prior to granting said applications access to resources. As discussed above, the authentication of applications and/or software components on wireless devices prior to providing network resources was known to be important, as software applications on devices could become hacked or otherwise compromised, and accordingly breach the security of a network. *See* ¶ 43. Given Lundblade is concerned with secure communications across a network, and further concerned with identifying compromises within the network, a POSITA would have been motivated to modify Lundblade to not only authenticate the client 300 itself (via credentials 312), but also to authenticate an application client 300 is attempting to access a resource from to keep the network secure. EX1004 (*Lundblade*), 16:29-33.

172. Hardjono provides an express teaching, suggestion, or motivation for implementing the resource verification process: to “identify and authenticate components of a client platform—both hardware and software—that are in a potentially improper state before the client platform is given access to network services or resources.” EX1006, [0008]. Both devices and software were known to be vulnerable to compromise as a result of downloading counterfeit software

components and/or viruses. EX1006, [0004]-[0005]. Furthermore, allowing these compromised devices and applications to then access a network resource could breach network security and corrupt the network. *See* ¶ 41-43. Given Lundblade's client 300 contains downloaded applications, a POSITA would appreciate performing Hardjono's resource verification procedure to ensure a downloaded application is not corrupted, malicious, or otherwise non-secure before providing the client 300 access to a specific network resource requested by the application. EX1004, 11:20-23, 10:43-46. Indeed, authenticating applications on a device prior to providing network resource access was known to provide the benefit of keeping hacked, compromised, or otherwise fraudulent applications from affecting the integrity of a network. *See* ¶ 43, 52. Further, using the credentials 312 taught by Lundblade to create the trust report is nothing more than merely combining prior art elements to yield a predictable result of the trust report containing a device credential. This would ensure that the trust report is established for an authenticated device in the Lundblade-Jobst-Hardjono system, while additionally using the TPM certificates would ensure that the digests being compared in making the trust report are legitimate and have not been falsified.

173. Additionally, a POSITA would have been motivated to use the credential based secure communication link taught by Lundblade to secure all future communications between the server and client device, including for the signals used

to request data to a network resource, and for the trust report sent back to the device. Indeed, Lundblade even expressly teaches using credentials 312 in subsequent commands from client 300, indicating the network resource request would similarly include credentials 312 and accordingly be communicated on the same secure link. Additionally, Lundblade's credentials 312 authenticate the device, which would be advantageous and supplement the trust report 220 that verifies the specific software component. *See* Claim 1[f]. Thus, a POSITA would appreciate performing the software verification procedure over Lundblade's already-established secure link, so the device is authenticated (via credentials 312), and the software application is then authenticated based on the network access request. Additionally, using the same link reduces resource requirements when a client registers to the network 104 and then wants to access a network resource once registered. Indeed, a POSITA would appreciate using the same link would decrease communication "congestion" on network 104 and save device resources (e.g., save CPU time by opening and closing fewer connections). EX1046 (*Hypertext Transfer Protocol*), 44 (describing various "advantages" to persistent connections over "separate" connections for each client/server communication). For example, a POSITA would appreciate using the same secure link that was already established by credentials 312 when client 300 requests registration (*see* Claim 1[a], 1[f]) with server 400 to also perform network resource request verification. Indeed, once a user is registered to the server 400, they

would desire that the downloaded application is fully functional, including being able to access the network resources required to execute the application. Thus, a POSITA would have been motivated to use the same Lundblade secure link for the software verification process to reduce resource requirements and keep both client 300 and server 400 from having to establish a new link.

174. Further, the modification would have had a reasonable expectation of success, given Hardjono teaches the authentication entity 145 components and integrity authority 140 being implemented on the same physical server. EX1006, [0031]. Additionally, much like Lundblade, Hardjono teaches performing such software component verification for wireless devices on a wireless network 135, including mobile phones, laptop, PDAs, etc. EX1006, [0031]-[0032]; EX1004, 10:19-37. Additionally, Lundblade teaches client 300 having various applications that are executable. EX1004, 10:13-18, 11:45-52. Furthermore, as discussed at Claim 1[f], Lundblade teaches a method of authenticating device signals for secure communication to establish a secure message link. Thus, all that is required in the modification is implementing the authentication entity 145 and integrity authority 140 on server 400 of Lundblade, which would include programming client device 300 to send and generate an integrity report 170 (including a digest for at least one software application) signed with credentials 312 to server 400 over the secure message link when requesting access to a network resource. For example, when

client 300 requests a network resource when executing an application (e.g., checking company email, checking personal stocks via a “stock monitor[]” application), the client would be programmed to send an integrity report that is then checked by server 400 to verify the trustworthiness of the application on client 300. EX1004, 11:45-52 (describing various device applications including, e.g., news, stock monitors, “and the like”); EX1006, [0037] (describing word processor, email, spreadsheet being exemplary installed applications), [0102]-[0104]. Given Lundblade teaches any of servers 106, 120, 122 having the functionality for performing secure communications with client 300 (e.g., mobile phone 102), a POSITA would have understood the software verification functionality of Hardjono would be placed in whichever server on network 104 and carrier network 200 is performing secure communications with client 300. EX1004, 14:63-65 (disclosing server 400 “can be any of the servers 106, 120, 122”), 10:46-50 (disclosing the secure communications “logic” may be located at any “one or more of the application download server 106, second server 120 and stand-alone server 122”).

175. With the authentication entity 145 and integrity authority 140 being implemented on server 400 of Lundblade, the modified server 400 would be programmed to compare the digest (which includes a hash) of the at least one software application in the integrity report 170 to digests in the integrity records 320 of the integrity authority 140, generate a trust report 220 based on the digest and the

device credential 312, and then send the trust report 220 to the client 300 over the secure message link for subsequent use in accessing network resources. Additionally, using the credentials produced in the process of Lundblade to generate the trust report would have been a basic software modification to a POSITA, especially considering Hardjono contemplates the use of “other certificates” when sending the integrity report 170 used to generate trust report 220. EX1006, [0076], [0068]. It would have been well within the skill of a POSITA to implement this functionality onto the server 400 of Lundblade, given that methods of authenticating applications prior to accessing network resources were known. *See* ¶¶ 52-55 (*citing, e.g.,* EX1009 (*Lundblade* ’330), [0034]-[0035]; EX1033 (*Kurn*), [0118]). Additionally, generating hashes and digests for applications and comparing them to determine the software’s authenticity was well-known. *See, e.g.,* EX1048 (*TCG Specification Architecture Overview*), 8-11 (describing generating “digests” of “program code” that are a hash and performing attestation by a “Challenger” comparing a computed digest with the “digest” in the device), 12 (describing reference digests that are generated by the software manufacturer and then “compared to actual (runtime) measurements enabling detection of changes”). Thus, a POSITA would not have found the modifications challenging.

176. Lastly, a POSITA would have understood that implementing the software verification teachings of Hardjono would not interfere with implementing

the device identifier verification or server memory organization taught by Jobst, at least because the modifications are independent and occur at different parts of Lundblade's process. For example, the verification occurs prior to providing credentials 312 to client 300 and association occurs after generating the credentials 312 at server 400. The software verification process of Hardjono occurs after credentials 312 have already been provisioned and used in establishing the secure link because a user is now requesting resource access via a specific application. A POSITA would have the knowledge required to implement the teachings into a single system without undue experimentation. Indeed, both device authentication via device credentials and authenticating applications requesting a network resource were important processes in data communication over a network, and thus combining the two into a single system would not have required undue experimentation. *See* ¶¶ 41-55.

9. ***1[h]: “in response to the second request, generating, based on the identified first application and the secure device credential, a first secured application credential unique to the first device, and”***

177. Lundblade in view of Hardjono discloses this element. *See* Claim 1[g]. Hardjono discloses generating a trust report based on the component (the application), the digest, and the TPM certificate. Specifically, Hardjono teaches the generated trust report 220 containing “one or more references to TPM certificates of

the client platform” and the name of the software component “with a reference to corresponding record 320.” EX1006, [0068]. As discussed above, record 320 contains the digest (in the form of a hash) of the software. *See* Claim 1(g) (citing EX1006, [0045], [0048]). Thus, given that the trust report 220 contains references to the software component hash and the TPM certificate, the trust report 220 is *generat[ed], based on the identified first application and the secure device credential*. The trust report 220 is also generated *in response to* the client 110’s resource request that includes the integrity report (i.e., *the second request*). EX1006, [0101]-[0104], [0095]; *see also*, Claim 1[g].

178. Additionally, as discussed above, trust report 220 includes the authentication entity’s 145 signature. *See* Claim 1[g]; EX1006, [0068] (disclosing trust report 220 including “a digital signature of verification server 205”). Hardjono’s trust report 220 is based on the TPM certificate and hash (contained within record 320 digest), much like the ’055 Patent describing generating a signed application credential using the application credential and device credential. EX1001 (*’055 Patent*), 57:43-56. For example, the ’055 Patent describes the device credential being a “certificate,” much like the TPM certificate of client 110 that the trust report 220 is based on. EX1001, 57:54-56. Additionally, the ’055 Patent describes the application credential coming from a “trusted source,” being “associated with the installed application,” and being, e.g., “a hash,” much like the hash in record 320

being harvested from trusted sources and corresponding to the hash in integrity report 170 for the installed software component 160. EX1001, 57:43-48; EX1006, [0038], [0045], [0068]; *see also*, ¶ 158. The '055 Patent also describes the application credential being “signed” and generated “using the application credential and a device credential,” much like Hardjono’s trust report 220 being signed by verification server 205 and being generated using the software component hash and the TPM certificate. EX1001, 57:50-52; EX1006, [0068]; *see also* ¶ 162. As discussed in Claim 1[g], in the combination, Lundblade’s credentials 312 are used in generating the trust report, in addition to the TPM certificate. *See* Claim 1[g] (providing the Motivation to Combine for this modification).

179. As taught by Hardjono, it would have been known to a POSITA that other credentials, like the credentials of Lundblade, could be utilized and placed in the integrity report 170 to then be referenced in trust report 220. EX1006, [0076] (disclosing “integrity report 170” including a digital signature from “a certificate derived from a TPM-bound certificate or signed by other certificates”). Accordingly, Lundblade-Hardjono teaches generating a trust report (*generating...a first secured application credential unique to the first device*) including the credentials 312 of the client device in response to the request. The trust report 200 is *unique to the first device* because the trust report 220 is generated using credentials 312 unique to the client 110. Hardjono teaches that the trust report 220 is generated for a specific client

110 that requested network resource access. EX1006, [0029], [0031]. In the modified system, the trust report is generated using credentials 312 that are unique to client 300 of the network system. *See* ¶ 169. Furthermore, the credentials 312 are generated using the device identifier unique to client 300. *See* Claim 1[c]. Thus, given trust report 220 is based on the client credentials 312, trust report 220 is *unique to the first device*. As discussed above, trust report 220 is a *first secured application credential* because it is signed by verification server 205 and is for the requesting software component. *See* Claim 1[g]. Furthermore, the trust report is *based on* (1) the hash (contained within digest) of the application and (2) credentials 312 provided in the request (the secured application credential *based on the identified first application and the secure device credential*). *See* ¶ 175.

10. 1[i]: “sending the first secured application credential to the first device over the secure message link.”

180. Lundblade in view of Hardjono discloses this element. *See* Claim 1[g]. As discussed in Claim 1[g], the Lundblade-Hardjono combination discloses sending the trust report 220 to the client device 300 to be used to access other resources or services. EX1006, [0041]. In the combination, the trust report 220 is transmitted over the link established via Lundblade’s secure communications used for all communications between the client device 300 and server 400. *See* Claim 1[g] (providing the Motivation to Combine). Indeed, Lundblade’s credentials 312 are

“always present” in signals sent between client 300 and server 400 when providing secure communication, and thus the same link is used for subsequent transfers between the two entities. EX1004 (*Lundblade*), 18:30-35, 15:55-61; *see also*, ¶ 173 (providing Motivation for all communications over the same link). Therefore, the Lundblade-Hardjono combination teaches sending the trust report (*sending the first secured application credential*) to the client device (*first device*) over the secure link formed via secure communications (*the secure message link*).

B. Claim 2: “The method of claim 1, wherein generating the first secured application credential comprises creating a combination of the general application credential and the secure device credential.”

181. As discussed above in Claim 1[g], Lundblade’s server 400 is modified by Hardjono to implement the authentication entity 145 and integrity authority 140 processes, which includes generating the trust report 220 (*secured application credential*). In the modified system, the trust report 220 includes a reference to credentials 312 (*secure device credential*) and a reference to the software component digest (containing the hash) (i.e., *general application credential*). *See* Claims 1[g]-1[h]. The server 400 references the credentials 312 and the digest within the trust report 220, combines the hash and credential 312 information together into trust report 220, thereby *creating a combination*. Accordingly, because the trust report 220 combines the digest (hash) and credentials 312 together into trust report 220, the

process of generating the trust report 220 in the modified system comprises creating a combination of the digest and the credentials.

C. Claim 6: “The method of claim 1, wherein the first secured application credential is a run-time application credential.”

182. Lundblade-Jobst-Hardjono teaches generating a trust report (*secured application credential*). See Claim 1[g]-[h]. The trust report 220 is generated in response to client device 300 requesting access to a network resource for a software application on the client device 300. See Claims 1[g]-1[h]. The software application (i.e., software component 160) is already installed on the client platform 110 and executable. EX1006 (*Hardjono*), [0037], [0095]. Hardjono specifically discusses that the resource 150 can be a web service or application. EX1006, [0031]. For example, tiered access to bank account information can be granted to the client device depending on the trust score 225 included in the trust report 220. EX1006, [0044]. Hardjono discloses allowing customers of a bank to “access their bank accounts through client platform 110,” providing “bank account information,” including, e.g., account balances, account statements, etc. *Id.* A POSITA would have understood the client 110 contains a banking application to allow a user to access their bank account. Indeed, mobile banking applications were well-known and common prior to 2009. See generally, e.g., EX1049 (*Banking on Mobiles*), 3 (describing “mobile banking application[s]” being wirelessly sent to a user’s phone for use in accessing their bank account), 15-16 (describing various known mobile

applications where the “interactive session can be managed by an application on the user’s handset that drives a menu or by a remote application controlled in the network”); EX1050 (*SMS Banking Services*), 227 (describing mobile banking applications that utilize SMS to allow a user to access their bank account or perform certain banking operations). Thus, a POSITA would have understood that the resource is access to bank account information, and the request comes from a user executing banking software on the device.

183. Additionally, Hardjono discloses trust report 220 as being a “trust ticket” that provides access to “network resources or services.” EX1006, [0041]. A “ticket” is an item that provides a user access to a good or service (e.g., a movie or a train). Given Hardjono teaches using the trust report 220 when accessing “any number of resources or services,” a user executing a software application component 160 to access a network resource uses trust report 220 to access such a resource and gain access. For example, if I am using Microsoft Word on client 110 and want to print a document using the printer, Microsoft Word uses its trust report 220 to get access to the printer. EX1006, [0031]. When a user of the banking application wants to access account information, the banking application uses its trust report 220 to get access to the account information. EX1006, [0044]. Thus, a POSITA would have understood in using the “trust ticket” (i.e., trust report 220) to access network resources with such resources including, e.g., accessing a printer to print something

or accessing account information in a banking application, that the trust report is a *run-time application credential* because it is generated for an executable application attempting to obtain a resource when it is being run. The '055 Patent describes a “run-time application credential” being “for an application that intends to initiate execution and obtain access network service.” EX1001 (*'055 Patent*), 54:43-47. This is similar to the trust report 220, which is for an application component 160 that, during execution, intends to obtain a network resource or service. Thus, a POSITA would have understood that trust report 220 is a *run-time application credential* because it is generated for an executable application attempting to obtain a resource.

D. Claim 7: “The method of claim 6, further comprising receiving, from the device link agent over the secure message link, the run-time application credential, and evaluating the run-time application credential at an element of the network system.”

184. The Lundblade-Jobst-Hardjono combination teaches obtaining a trust report at a client device. *See* Claim 1[h]. The trust report 220 is a run-time application credential. *See* Claim 6. Hardjono contemplates that the trust report is a “sort of ‘trust ticket’ to access other network resources or services.” EX1006 (*Hardjono*), [0041]. The Lundblade-Jobst-Hardjono combination also teaches that every communication made by the client device over the network is a secure communication via a secure link (*secure message link*) between the client security module (*device link agent*) of the client device and a network server. *See* Claim 1[f], [g], [i]. A POSITA would have understood that to access a network resource or service with the trust report

stored on the client device, the trust report is sent to the server 400 of the combination using secure communications (i.e., *receiving, from the device link agent over the secure message link, the run-time application credential*). Indeed, Hardjono teaches “[t]he other network resources or services can also have an authenticator entity to check the validity of trust report 220” when the “trust ticket” is used to “access other network resources or services,” indicating the trust report 220 is sent to a server to access network resources for the software application. EX1006, [0041]. Thus, in the combination, after client 300 receives trust report 220, trust report 220 is sent to server 400 to access a network resource when executing the software application on client 300. As discussed above, Lundblade’s secure communication logic may be stored in any server (server 106, 120, 122) on network 104. *See* Claim 1[g]. Additionally, Lundblade’s credentials 312 are included in all communications once secure communication is established. *See* ¶ 168. Thus, in the combination, trust report 220 is sent to server 400 from client 300 over the same secure link initially established via credentials 312, given all communications over the network system are secured with credentials 312.

185. As discussed above, Hardjono teaches trust report 220 being authenticated by authentication entity 145 when sent from the client for accessing network resources. *See* ¶ 183. In the combination, the authenticator entity is incorporated into the server 400 of Lundblade. *See* Claim 1[g]. Because the

authenticator entity “check[s] the validity of trust report 220,” a POSITA would have understood that the authenticator entity of server 400 receives the trust report and determines if the trust report is valid (*evaluating the run-time application credential at an element of the network system*) prior to granting access to the requested network resources. EX1006, [0041]. Authenticator entity of server 400 evaluates the validity of the trust report in order to grant access to the network resource, such as, accessing certain required network elements. For example, when an application on client 300, having a trust report 220, requests access to network resources (e.g., banking application requesting account information, Word application requesting printer access, email application requesting server access), the authentication entity of server 400 receives the trust report and determines if the report is valid to grant access to the requested network resources, such as granting access to network elements required in accessing the resource (e.g., back-end banking server that electronically holds account information, printer on network, email server on network).

E. Claim 8: “The method of claim 7, further comprising, in response to evaluating the run-time application credential, returning a network policy setting over the secure message link to the device link agent, the network policy setting applicable to the first application.”

186. Hardjono teaches that, upon evaluation of Hardjono’s trust report 220 (i.e., *in response to evaluating the run-time application credential*), the

authentication entity 145 determines, for example, what banking privileges are allowed for a banking application requesting network resource access. Namely, the trust score 225 contained within the trust report 220 is evaluated against thresholds by a policy server 215, which forms part of the authentication entity 145. EX1006, [0044], [0035]. Each threshold allows the device access to different resources or information for the application. EX1006, [0044], [0100]. Though Hardjono provides the specific example of gaining access to bank account information, because Hardjono also discloses policy server 215 implementing customized policies for “a given institution,” Hardjono also contemplates using such tiered access via threshold comparison for other applications that access a different network resource or service and accordingly have their own trust report 220. For example, given an exemplary application on client 110 is a “spreadsheet” application and an exemplary resource 150 is a storage device, tiered access would be implemented to gain access to certain confidential or highly important information within the storage device. EX1006, [0031], [0037]. In the banking application scenario, when a lower threshold is met, the client device is only enabled to view account balances. EX1006, [0044], [0100]. Conversely, when a higher threshold is met, the client device is enabled to view account balances and perform money transfers. EX1006, [0044], [0100]. The thresholds are driven by the policy server 215 and can be specific to a given institution. EX1006, [0044].

187. Hardjono teaches that the policy server 215 (incorporated into server 400 in the modified system) decides whether to grant or deny access to a resource based on the predetermined policy. EX1006, [104] (disclosing policy server 215 comparing trust score 225 “with a predetermined policy or threshold” to determine whether to grant or deny access”). The grant/deny result is forwarded to the client device. *Id.* A grant or deny result is, itself, a *network policy setting* because it “identif[ies] conditions in which a network access policy should be implemented.” EX1001 (’055 *Patent*), 13:41-45. Namely, for Hardjono’s grant response, the network access policy would never be implemented, as the access is fully granted. The policy enforcement agent would never be invoked to restrict access to the resource, i.e., authentication server 210 as the “Policy Enforcement Point (PEP)” would never be invoked by policy server 215 to restrict access. *Compare* EX1006 (*Hardjono*), [0035], [0104] *with* EX1001, 13:41-45. For Hardjono’s “deny” response, the network access policy would always be implemented (i.e., the “PEP” (much like the policy enforcement agent described in the ’055 Patent) would always be invoked to restrict access to the resource). *Id.*

188. Furthermore, Hardjono additionally discloses the forwarded result may inform client 110 that a “required remediation” be performed, such as re-installing the applications, or may provide “reduced” access to the network resource or service. EX1006, [0099]. For example, in the banking scenario, a lower trust score 225 may

warrant a lower tier (i.e., reduced) access be provided. EX1006, [0044]. For Hardjono's "required remediation" response, the network policy setting specifies the conditions under which the network access policy would be implemented. Namely, when providing the "required remediation" response, the network policy setting (as analyzed by the policy server 215) invokes the PEP to restrict access to the resource until the required remediation has taken place. This is much like the '055 Patent describing identifying conditions where the network access policy should be implemented and then causing a policy enforcement agent to apply the policy. EX1001, 13:41-45; EX1006, [0035], [0104] (policy server 215 determining whether to require remediation based on the policy and then causing PEP to apply the policy (i.e., require client to remediate)).

189. Because the grant/deny result comes from a policy server 215 that uses "predetermined polic[ies]" to determine the level of access, and the forwarded grant/deny result specifies the level of access to a network resource or service (e.g., level of access to banking information), the grant/deny result is a *network policy setting*. EX1006, [0104]. Indeed, the grant/deny result informs the software application what level of access it has to the network resource (i.e., all access granted, partial access, no access). Thus, a POSITA would have understood the grant/deny result forwarded to the client to be a *network policy setting*.

190. In the modified system, server 400 contains the authentication entity 145 functionality, which includes policy server 215. *See* Claim 1[g] (providing Motivation to Combine). Thus, the server 400 of Lundblade-Jobst-Hardjono *return[s] a network policy setting over the secure link established via Lundblade’s secure communication (i.e., over the secure message link to the device link agent) in response to evaluating the trust score 225 in the trust report 220 (i.e., in response to evaluating the run-time credential).*

191. A POSITA would have understood that the grant/deny result is *applicable to the first application* (e.g. the banking application) because the trust report (i.e., *secure application credential/run-time application credential*) and by extension, the trust score is created from the digest of that application. *See* Claim 1[g]-[h]. As discussed above at Claim 1[g], the trust report 220 is generated for “at least one software component,” such as a software application requesting access to a network resource. The trust report 220 includes the trust score 225, which is determined based on the comparison of the integrity report 170 digest to the record 320 digest. EX1006, [0104] (describing comparing digest in integrity report 170 to digest in record 320 to determine the “level of trustworthiness” that is then “embodied in trust report 220 including overall integrity/trust score 225”). Thus, because the trust score 225 is based on the software application digest, the grant/deny

result that comes from comparing the score 225 to thresholds via policy server 215 is *applicable to the first application*.

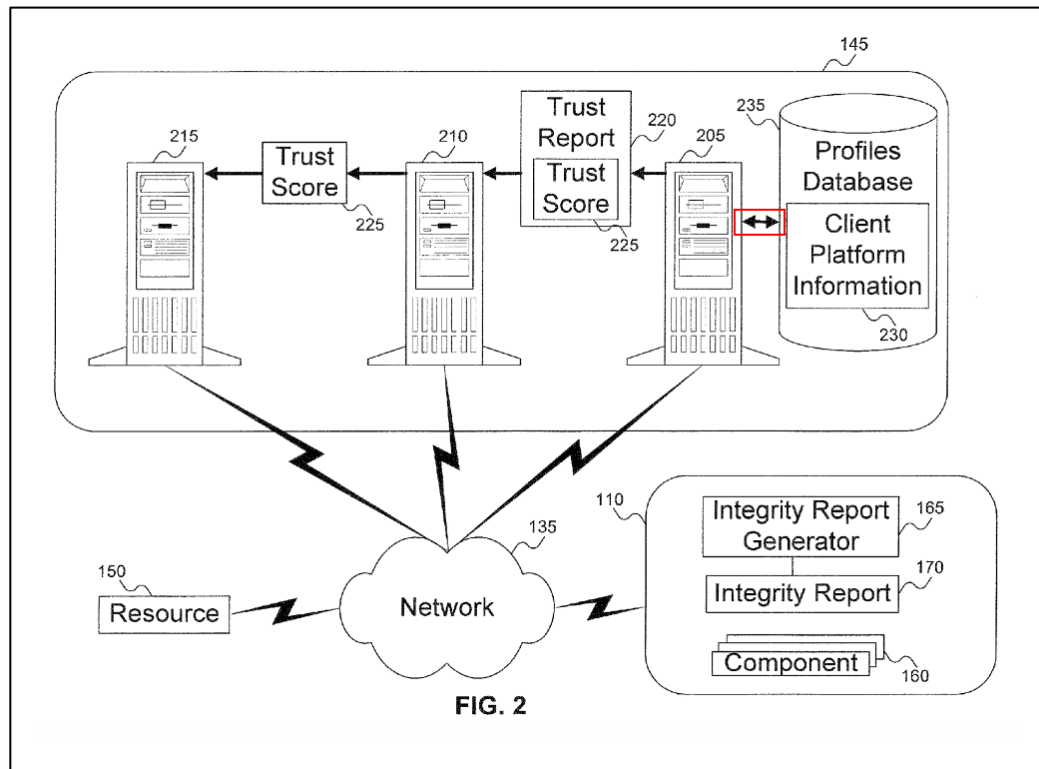
192. A POSITA would have further understood that the grant/deny result (*network policy settings*) would only be applicable to the specific application requesting the resource. As discussed above, Hardjono provides various resource examples that may be requested for access, including storage, web services, applications, etc. EX1006, [0031]. Hardjono's client 110 has various installed applications whose digest is collected for verification, including a word processor application, spreadsheet application, an "email package," and a banking application EX1006, [0037], [0044]; *see also*, ¶ 182 (explaining banking example is for a banking application on client 110 trying to access account information). Given banking resources have a higher privacy requirement than, for example, email or a word processor, a POSITA would have understood that the banking thresholds would not apply to other applications, such as an email package. EX1006, [0007] (disclosing banks were known to be an "institution that is especially vulnerable to" malware risks and "other illegitimate component" issues). Indeed, Hardjono even indicates different institutions having differing thresholds and, accordingly, different grant/deny results. EX1006, [0044]. Thus, given this, a POSITA would have understood the grant/deny result is only applicable to the *first application* that is requesting access to the particular network resource.

F. Claim 9: “The method of claim 7, wherein the element of the network system is an authentication server, the network system further comprising an application credential database coupled to the authentication server and storing secured application credentials, including the first secured application credential.”

193. As discussed above, Lundblade-Jobst-Hardjono system receives and evaluates a run-time application at the network system. *See* Claim 7. The evaluation is done by server 400 of Lundblade, which incorporates the functionality of the authenticator entity 145 and integrity authority 140 of FIG. 15 of Hardjono, including profiles database 235, verification server 205, authentication server 210, and policy server 215, as illustrated in FIG. 16. EX1006, [0035]; *See* Claim 1[g]. Thus, server 400 of the Lundblade-Jobst-Hardjono combination is an authentication server (*wherein the element of the network system is an authentication server*). Namely, server 400 includes the components to perform the functionality of the authentication entity 145, including the functionality of the authentication server. Additionally, server 400 also authenticates the communications coming from client 300 by checking credentials 312, further indicating server 400 is an authentication server. *See* Claim 1[f]; EX1004 (*Lundblade*), 12:61-63 Thus, server 400 is *an authentication server*.

194. Hardjono teaches that the authentication entity 145 is coupled to the profiles database 235, which stores trust reports 220 (*storing secured application credentials, including the first secured application credential*) along with other

information. EX1006, [0043], FIG. 2. As part of the software component verification process, trust report 220 is stored in profiles database 235 by verification server 205, which forms part of authentication entity 145. EX1006, [0035], [0104]. Since server 400 in the Lundblade-Jobst-Hardjono system contains the functionality of authentication entity 145 which itself includes the functionality of verification server 205, it (server 400) must also be coupled to the profiles database (an *application credential database coupled to the authentication server*). The profiles database is an *application credential database* because it stores all the trust reports 220 (i.e., *secured application credentials*). FIG. 2 shows the profiles database 235 being electrically coupled to verification server 205 of authentication entity 145 so that verification server 205 can store trust report 220 therein:



EX1006, FIG. 2 (annotated to show the coupling of verification server 205 to profiles database 235).

195. The profile database 235 being communicatively coupled to verification server 205 of authentication entity 145 is much like the '055 Patent describing an “application credential database 221” interfacing with an authentication server. '055 Patent, 23:24-25. In the modified system, because server 400 accesses database 235 to store trust reports 220 therein, the server 400 is likewise *coupled to* the profiles database.

- G. Claim 10: “The method of claim 1, further comprising based on a first device event, refreshing the first secured application credential to create a different secured application credential unique to the first device, and sending the different secured application credential to the first device, the different credential superseding the first secured application credential.”**

196. The Lundblade-Jobst-Hardjono client 300 receives a trust report 220 from server 400 that can be used to access network resources for an application. *See* Claim 1[g]-[i]; EX1006, [0041]. Hardjono discloses that the trust report does not provide access to resources for an unlimited amount of time. Specifically, periodic reevaluation can be performed “according to a corporate policy.” EX1006, [0079] (disclosing entity 145 “periodically reevaluat[ing] client platform” according to corporate policy). A POSITA would have understood that a company policy outlines events that require the system to perform reevaluation before providing access to resources, so that the periodic reevaluation can be performed “according to” the company policy. This allows the system to know when reevaluation is required, based on the company policy. For example, when a laptop computer connects to a network external to the corporate network during a business trip, it can become tainted by malware, which is subsequently introduced to the network. EX1006, [0006]. Given malware would compromise network security if introduced into the network, a POSITA would have understood it to be beneficial for a corporate policy to require reevaluation after a device connects to an external network. In this way, the company can be ensured the device is not tainted with malware prior to being

provided access to a network resource when back at the company. A POSITA would have understood that obtaining a new trust report 220 at the occurrence of certain events in a company policy (*device event[s]*) would better protect the system from malware. Requiring new trust reports according to certain events occurring that are outlined in a company policy, such as after connecting a work laptop to an unsecure external network, reduces the risk that malware introduced to the device after the initial trust report is obtained can be used to access network resources. Managing access to resources in this manner better secures the system.

197. Further, a POSITA would have understood that a company policy could include requiring reevaluation after a set period of time, device connection to an external network, or the application updates. The trust scores of Hardjono includes date/time of verification and Hardjono contemplates versions of software resulting in different integrity scores. EX1006, [0068], [0073] (disclosing if the software version is “outdated” or older, the “score can be lower” than a computer with “a more current version”). Basic company policies would have included requiring new trust reports after a period of time, when the application is updated, or after the device is connected to an unknown network, in line with the disclosure of Hardjono. For example, certain institutions, such as a banking company that requires a high degree of security to ensure sensitive information is not tainted by malware, would periodically reevaluate the hardware and software components of

the device after such events. EX1006, [0007], [0044]. Such events being device-dependent (e.g., updating application on device, connecting device to external network) and requiring reevaluation according to company policy indicates the company policy defines which *device event[s]* require recertification.

198. Hardjono, discloses that reevaluation is performed in the same manner as the initial evaluation (device requested due to requesting a resource), or the authenticator entity itself can request an integrity report for the client device. EX1006, [0079]. For the same reasons discussed above (i.e., increasing network system security by verifying components to keep malware from affecting network), the Lundblade-Jobst-Hardjono system would reevaluate based on events occurring as outlined in a company policy (i.e., *based on a first device event*) and accordingly generate a new trust report. *See* Claim 1[g] (providing Motivation to Combine). A POSITA would have understood that a new trust report is returned to the device after the revaluation (*refreshing the first secured application credential to create a different secured application credential unique to the first device, and sending the different secured application credential to the first device*). In the same manner as the original trust report, the new trust report would be sent to the client device to be used to access company resources until the company policy requires yet another updated trust report. This new trust report 220 would be the refreshed version of *the first secured application credential* for the same application, namely a *different*

secured application credential, given, for example, a new trust report 220 is generated due to reevaluating after updating a software component. See ¶ 197. The new trust report 220, much like the original trust report 220, is still *unique to the first device*, because it is generated for that specific client 110. See Claim 1[h].

199. Further it would have been understood by a POSITA that the old trust report is no longer valid, and the new trust report is used to access resources (*the different credential superseding the first secured application credential*). Security is only increased if the old trust report is no longer valid and the new trust report is used instead. If both reports remain valid, no security benefit would be achieved by requiring a new trust report to access resources for the application. Additionally, given the new trust report more accurately depicts the current state of the device (including the state of the application requesting a resource), the new trust report *supersed[es]* the old trust report.

H. Claim 11: “The method of claim 1, further comprising, in response to the second request, verifying that the general application credential matches a known-application credential for the first application.”

200. As discussed above at Claim 1[g]-1[h], Lundblade is modified by Hardjono to implement the resource verification procedure via authentication entity 145 and integrity authority 140 functionality on server 400 to request a network resource (*second request*) and generate a trust report (*secured application credential*). As part of generating the trust report 220 for the software component,

the digest (*general application credential*) of the component (application) in the integrity report is compared to a digest of that application in the integrity records 320 in a global signature database 325 (a *known-application credential for the first application*). EX1006, [0045], [0104] (describing comparing digests in integrity report 170 to digests in record 320). The digest in record 320 is harvested from the software source, such as from software purchased from the manufacturer, software vendors, etc. EX1006, [0045]. The record 320 digest also includes a hash. EX1006, [0048]. This is much like the '055 Patent describing a “known-application credential” that is a “hash for the known application program.” EX1001 ('055 Patent), 58:11-14. Because digest in record 320 contains a hash and is harvested from the software source of the application program, the digest is a *known-application credential for the first application*.

201. Hardjono also teaches comparing integrity report digests to digests in records 720 stored in a “local” database during the comparison, and, if no match is found, the global database is then searched. EX1006, [0085], [0038], [0077]. Records 720 in local database are “periodically updated and synchronized with integrity/trust records 320” in the global database 325. EX1006, [0081]. Thus, records 720 are records 320 that have been synced and stored locally. If a hash of the digest in the integrity report 170 does not sufficiently match a hash of the digest in records 320/720, access may be denied or partially denied; conversely, if the

hashes of the digests match, then the software application is verified as being trusted. EX1006, [0085], [0104] (disclosing using the comparison to determine the level of trustworthiness). Thus, a POSITA would have understood that the comparison resulting in a match verifies that the digest included in the integrity report 170 matches the known digest (*verifying that the general application credential matches a known-application credential for the first application*). Therefore, the comparison of the digest in integrity report 170 to known digests in records 320/720 is *verifying that the general application credential matches a known-application credential for the first application*. Accordingly, the Lundblade-Jobst-Hardjono system verifies the integrity report 170 digest against a known digest for the first application, *in response to the second request*.

I. Claim 13: “The method of claim 11, further comprising obtaining the known-application credential from an app store.”

202. The combination of Lundblade-Jobst-Hardjono performs the resource verification procedure by comparing a digest included within integrity report 170 to integrity records 320 in the process of generating the trust report 220. *See* Claim 11. Hardjono teaches that the digests stored in the integrity records 320 are formed from raw information 305 of software components that comes “from their sources (e.g., manufacturers, software vendors, and hardware vendors, among other possibilities).” EX1006, [0045]. A POSITA would have understood that a “software vendor[]” is an app store, because a software vendor sells various software

applications (i.e., “apps”) in a marketplace (the *app store*). Additionally, a “software vendor” is an inclusive term that includes direct sales from the software producer and indirect sales from a distributor, such as an app store. *See, e.g.,* Ex1051 (*Netscout Systems Inc.*), 5 (software vendor Netscout describing distributing their products through direct sales and distributors, resellers, service providers, etc.). Thus, the software vendor is an app store. Hardjono discloses harvesting the record 320 digests from “their sources” because “[i]t is desirable that the technique that results in the highest overall level of trust” be used, such as “the chain of custody” being as secure as possible. EX1006, [0045]. Given Hardjono’s desire to have the “chain of custody” be as secure as possible, a POSITA would have understood the digest being harvested directly from the app store for those apps purchased from the app store. Thus, Lundblade-Jobst-Hardjono’s system *obtain[s] the known-application credential* (i.e., hash of record 320 generated from digests of harvest information 305) *from an app store*.

J. Claim 14: “The method of claim 1, the verification step further comprising evaluating a subscriber identifier associated with one or more wireless end-user devices, including the first device.”

203. Lundblade-Jobst-Hardjono teaches performing verification of characteristics 310 against expected characteristics stored at the server. *See* Claim 1[b]. Lundblade’s characteristics 310 include both device identifiers as discussed, as well as subscriber identifiers (e.g., mobile directory number (MDN), IMSI, etc.).

EX1004, 12:29-42. The mobile directory number is the 10-digit phone number associated with a user of a mobile device. *See, e.g.,* Ex1052 (*Wollersheim*), [0009]. Subscriber identifiers were well-known to be included in access requests, and included phone numbers, PINs, IMSI, etc. *See* ¶ 46. Jobst teaches, when validating the device ID (i.e., IMEI), the software provider 33 “can find the phone number of the requesting phone 1 based on the IMEI code.” EX1005, 7:15-22. Given that the IMEI is validated and used to find the phone number, a POSITA would have understood that the validation of the Lundblade-Jobst-Hardjono combination verifies both a device ID (e.g., SID, ESN) and subscriber ID (i.e., MDN) (*the verification step further comprising evaluating a subscriber identifier associated with one or more wireless end-user devices, including the first device*). In the combination, the phone number (MDN) provided in the characteristics 310 is checked against the expected phone number for the device and thus verifies that the subscriber is valid, as well as the SID or ESN being checked against the expected device ID to verify the device itself is valid. A POSITA would have been motivated to apply Jobst’s verification step to Lundblade’s characteristics 310 for the same reasons provided at Claim 1[b], including verifying both the device identifier and a subscriber identifier.

K. Claim 15: “The method of claim 14, further comprising generating the secure device credential based at least in part on one or more of the device identifiers and also on the subscriber identifier.”

204. Lundblade teaches generating credentials 312 based on “any one or more of” a list of the characteristics 310, including both device identifiers (e.g., ESN, SID) and subscriber identifiers (e.g., removable user identity module (RUIM), mobile directory number (MDN), subscriber billing account, etc.). EX1004, 12:29-42, 15:53-60. Lundblade’s mobile directory number and subscriber billing account are exemplary *subscriber identifier[s]*, because they identify the subscriber of the device. See ¶ 46. Thus, Lundblade teaches a credential 312 that is generated at least in part from characteristics 310 that include both device identifiers and subscriber identifiers (*generating the secure device credential based...on or more of the device identifiers and also on the subscriber identifier*).

L. Claim 18: “The method of claim 1, wherein a first subscriber is associated with both the first device and with a second wireless end-user device, the method further comprising the network system generating, based on the identified first application and a second secure device credential associated with the second device, a second secured application credential unique to the second device.”

205. Lundblade teaches a plurality of wireless devices being connected to network 104/200, including mobile phones 102, personal computer, personal digital assistant 108, etc., and that any of these devices can be used in the system as client device 102. EX1004, 10:19-37, 15:63-65 FIG. 1. Lundblade also teaches

characteristics 310 that are used in generating the device credential 312 includes subscriber identifiers (e.g., IMSI, user ID, PIN, etc.). EX1004, 12:19-42.

206. Hardjono teaches performing the resource verification on various types of wireless devices connected to network 135, such as a personal computer 115, wireless phone 125, personal digital assistant 130, etc. EX1006, [0030], FIG. 1. Thus, a variety of devices such as a personal computer, wireless phone, or PDA can utilize the invention of Hardjono to verify their software components. EX1006, [0030]. A POSITA would have understood that in a situation where a user has both a company laptop and company phone, the same application is downloaded onto both, and a unique trust report would need to be generated for each device (*a first subscriber is associated with both the first device and with a second wireless end-user device*). EX1006, [0095], [0033]. It was common for a single user, such as an employee, to “utilize a variety of electronic terminals” over a private network, such as a company network. EX1032 (*Matta*), 11:14-22; EX1053 (*Internet and Email Use for Work*) (article generally describing employees commonly owning multiple devices, such as cell phone and laptop, and using these devices to access the same applications, such as email). Additionally, the trust report is unique to the device that requested the network resource, as the trust report is based on the device certificate. EX1006, [0068].

207. Thus, Lundblade-Jobst-Hardjono teaches that the device credentialling procedure can be applied to multiple different devices (mobile phone, personal computer, personal digital assistant) that can belong to a user, and the resource verification procedure can be performed for an application on any of those devices. Further, a POSITA would have understood that an individual user can own and use multiple devices with the same applications. Indeed, for many applications, such as the email package discussed by Hardjono, a user would commonly have the email package installed on multiple devices, such as a personal computer and mobile phone. *See, e.g.,* EX1053, 5-6 (describing working professionals often owning multiple devices (e.g., laptops, mobile phone, PDA)), 8-9 (describing these working professionals checking email on multiple devices); EX1054 (*Microsoft Reveals New Windows Mobile 6 Smartphone Software*) (generally describing the Windows mobile phone software introduced in early 2007 allowing users to view their emails on the phone from a “corporate server”). Additionally, given the proliferation of workers using multiple wireless devices, other work-related applications would also be installed on the multiple devices, such as Microsoft Word and Excel on the Windows mobile phone and a Windows laptop. EX1054, 1 (describing the Mobile 6 smartphone software allowing users to utilize Microsoft applications “previously available only on PCs”).

208. Given (1) Lundblade-Jobst-Hardjono's network contains multiple devices, (2) performs resource verification on multiple devices, and (3) associates devices with a subscriber (by including subscriber ID in characteristics 310), a POSITA would have understood the system generating a new device credential for a second wireless device associated with the same user and performing the resource verification process for the same application (e.g., the email package) downloaded onto the second device to generate a trust report (*second secured application credential*) unique to the second device. For example, a single user in the Lundblade-Jobst-Hardjono system accessing their company phone and laptop would have different device credentials for each device (given the differing device identifiers in characteristics 310), and would have a different trust report generated from the same application downloaded on both devices that is unique to the second device (given the trust report 220 is based on the unique device credential). *See* Claim 1[c], 1[g]-1[h]. Thus, in the Lundblade-Jobst-Hardjono system, a second unique trust report (*second secured application credential unique to the second device*) is generated for the same user (*first subscriber*) for a second device (*second wireless end-user device*), based on the same application (*first application*) requesting a resource and the credentials of the second device (*second secure device credential associated with the second device*).

M. Claim 19: “The method of claim 2, wherein the combination comprises a hash.”

209. Lundblade-Jobst-Hardjono generates trust report 220, which includes a combination of the credentials 312 and the digest of the component. *See* Claim 2. As discussed in Claim 1[g], the digest includes a hash of the component. *See* Claim 1[g]. Because the digest included in the trust report 220 contains a hash, *the combination*, which is the combination of the component digest and credentials 312 within trust report 220, *comprises a hash*.

IX. GROUND 2: CLAIM 3 IS OBVIOUS OVER LUNBLADE, JOBST, HARDJONO, AND ZUCCHERATO

A. Claim 3: “The method of claim 1, further comprising, in response to receiving the first request and prior to performing the verification step, returning to the first device a network system credential that is verifiable through a trusted certificate authority.”

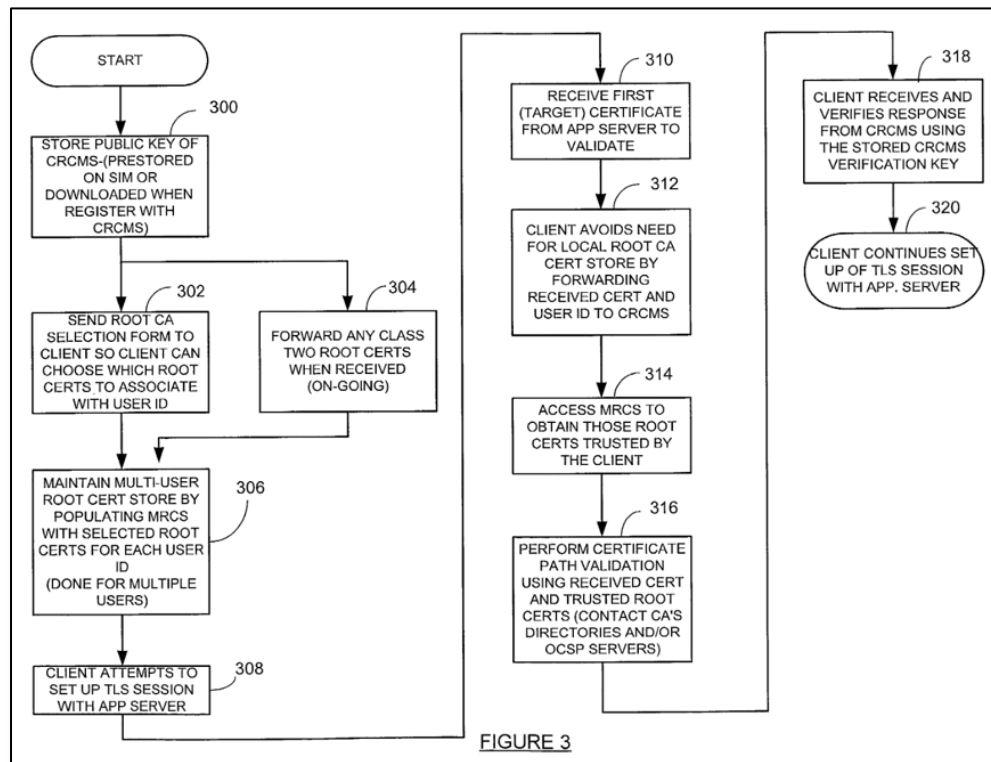
210. Lundblade-Jobst-Hardjono teaches a network server 400 receiving a request for secure communications from a client device 300, verifying the characteristics 310 within the request, and returning credentials 312 to the client device 300. *See* Claim 1[Pre]-[e], *supra*. While Lundblade-Jobst-Hardjono does not expressly discuss establishing the authenticity of the network system as a part of the communication between the first device and the network system, a POSITA would have understood that such connections are commonly performed over protocols such as SSL/TLS, and that (as a part of establishing a connection using such a protocol), a digital certificate is commonly used to establish the authenticity of the server side

of the connection. For example, TLS protocols were commonly implemented when establishing connections and were known to include the server side providing a digital certificate to establish the server authenticity, in addition to the device providing its digital certificate. *See, e.g.,* EX1032 (*Matta*), 5:8-10 (describing extensible application protocol transport layer security (EAP-TLS) being a “widely considered” civilian security technology), 16:26-17:6 (describing, when establishing a secure link between a laptop and access point, a server transmitting “his own digital certificate to the user” when securing a wireless session via EAP-TLS); EX1045 (*TLS Protocol Version 1.0*), 36 (describing a “server certificate” sent from server during TLS protocol). This server authentication was commonly implemented to prevent man-in-the-middle attacks. *See* ¶ 213. Indeed, during the verification process, Hardjono contemplates the server (as well as the client) both “exchanging” certificates over, e.g., SSL/TLS protocols to establish a secure channel connection. EX1006, [0101]. A POSITA would thus be motivated to seek out a reference like Zuccherato providing details on how to set up a secure link.

a. Zuccherato’s Teachings

211. Zuccherato teaches a method of setting up a transport layer security session (TLS session) between a mobile device 202 and a server 216. EX1007 (*Zuccherato*), [0031]. Zuccherato teaches that when a wireless mobile device 202 attempts to set up a TLS session with a server, the server 216 sends a certificate 252

to the device 202 (*returning to the first device a network system credential*). EX1007, [0032]. The device forwards this certificate to a certificate management server 204 that checks the validation of the certificate 252 by constructing a certificate chain (*verifiable through a trusted certificate authority*). EX1007, [0032]-[0033]. If the certificate 252 is valid, the server 204 sends the device 202 a validation response, and the mobile device 202 continues setting up the TLS session with the server 216. EX1007, [0033].



EX1007, FIG. 3.

212. The Lundblade-Jobst-Hardjono-Zuccherato combination modifies the secure communication process taught by Lundblade-Jobst to include an additional step of sending a certificate to the client device 300 for validation at a certificate

management server 252, which is also part of the network system. If the certificate is determined to be valid, the secure communication establishment process continues. Specifically, the registration request is sent by client 300, the server certificate is validated, and then the verification process taught by Jobst is performed, and if the characteristics 310 are verified, the credentials 312 are generated and sent back to the client device 300 to establish secure communications as taught by Lundblade. *See* Claim 1[a]-1[e]. Indeed, the server certificate would only be received after client 300 attempts to register (i.e., sends registration request). *See* Claim 1[a] (client 300 requesting credentials via registration request). Furthermore, the server certificate would be validated prior to server 400 verifying the characteristics, so client 300 knows whether the server is trusted. *See* Claim 1[b]. When the server is validated, the server would then perform verification of the characteristics 310 and, if valid, the credentials 312 are generated and sent back to the client device 300. *See* Claim 1[c]-1[e]. Thus, the validation process of Zuccherato is performed *in response to receiving the first request and prior to performing the verification step*.

b. Motivation to Combine with Zuccherato

213. A POSITA would have found it obvious and been motivated to modify Lundblade's network system and process to include the verifiable server certificate taught by Zuccherato. This would include either validating the server 400 certificate

at a known certificate management server in the network or sending a previously validated server 400 certificate to the device. The claim does not require the client to send the *network system credential* (i.e., server 400 credential) to a *trusted certificate authority*. Thus, either implementation above would be obvious to a POSITA, per Zuccherato. Indeed, certificate authorities (CA) were commonly used to verify certificates by signing such certificates, and thus, if client 300 receives a server 400 certificate signed by a CA, the client 300 knows the server 400 is verified. EX1007, [0002]-[0003] (disclosing CAs were widely known to issue certificates that were verified via signature key pairs). A POSITA would have understood that this modification is nothing more than combining prior art elements (i.e., Lundblade's network system and Zuccherato's verifiable server certificate) according to known methods to yield the predictable result of increasing the security of the network.

214. A POSITA would have been motivated to modify the network system of Lundblade-Jobst-Hardjono to send client 300 a certificate after receiving a registration request and before completing device verification to allow a client to confirm that the network server is legitimate (and not, for example, a man-in-the-middle). Indeed, certificates signed by a CA were known to indicate the "certificate can be trusted." EX1013, 8:4-8; EX1045 (*TLS Protocol Version 1.0*), 38-39 (describing the server certificate sent to the device requesting access to be an X.509v3 certificate that traces back to a root certificate having been signed by a

certificate authority). This is to differentiate a trusted server (via the trusted certificate) from “illegitimate server certificates” often created during a man-in-the-middle attack. EX1055 (*Wilson*), 3:8-16 (describing Man In the Middle Attacks as an attacker obtaining a certificate from a legitimate certificate issuer, using that certificate to “illicitly spawn a bogus certificate issuer, and use the bogus certificate issuer to create illegitimate server certificates”). Indeed, only providing one-way authentication (of the device) was known to be problematic, providing opportunities for man-in-the-middle attacks. EX1056 (*An Initial Security Analysis of the IEEE 802.1X Standard*), 7 (describing “the lack of mutual authentication can be exploited to mount man-in-the-middle attacks”). Thus, to combat Man-In-The-Middle attacks, a POSITA would have been motivated to include server verification in Lundblade’s secure communication method.

215. Indeed, a POSITA would appreciate ensuring that the server is a trusted server would be important for the security of the entire process. Otherwise, any credential generated and sent to the device may be compromised itself or cause the device to be compromised if used. This is because, if the server is not verified, an attacker may pose as the server and provide bogus credentials to the device to gain access to the device. Thus, a POSITA would understand the risk of a device connecting to a fraudulent server and accordingly compromising the device or allowing the fraudulent server to gain access to sensitive information on the device.

Based on such a risk, a POSITA would have been motivated to include server verification (via a trusted server certificate being sent to the client) in Lundblade's secure communication process.

216. Further, a POSITA would have had a reasonable expectation of success in modifying Lundblade's system, given that networks were known to send a credential to a device when a device attempts connection in protocols like TLS and given that certificate authorities were well-known. Indeed, the 1999 TLS protocol describes servers sending certificates to the device that are signed by a trusted certificate authority. EX1045 (*TLS Protocol Version 1.0*), 38-39; EX1013 (*Hind*), 8:4-8. Thus, all that is required is making simple programming changes to server 400 and client 300 to implement the validation process using the well-known certificate architecture of the Internet at the time, which would not require undue experimentation. For example, server 400 would be issued a signed certificate from a certificate authority (which was common and well-known), indicating the server 400 is trusted, and server 400 would be programmed to send client 300 the signed certificate after receiving the registration request. Servers sending their own certificate to client devices was common, so a POSITA would not have found such a modification challenging. EX1045, 38-39; EX1032 (*Matta*), 16:26-17:6.

X. GROUND 3: CLAIM 4 IS OBVIOUS OVER LUNDBLADE, JOBST, HARDJONO, AND CHEN

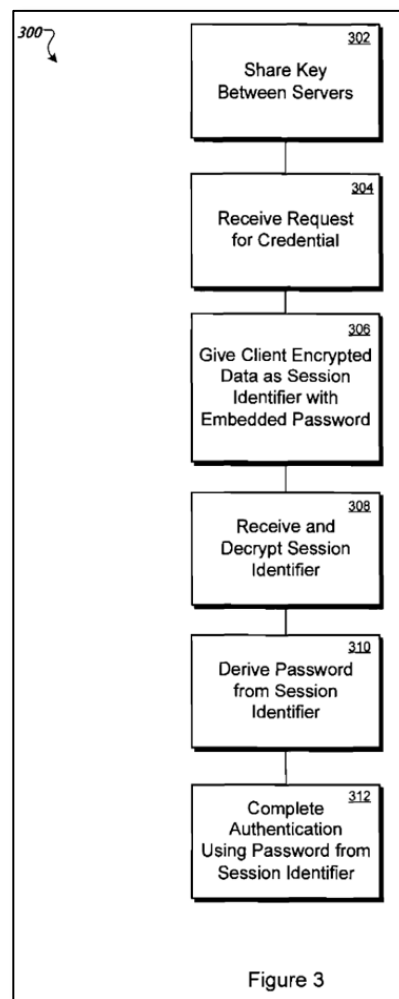
- A. Claim 4: “The method of claim 1, wherein generating a secure device credential for the first device comprises encrypting the credential with a key known by at least one element of the network system.”**

217. Lundblade teaches generating credential 312 for client 300 includes applying cryptographic algorithms, such as a one-way cryptographic algorithm, or “different cryptographic algorithms.” EX1004, 12:51-60. Further, Lundblade discloses that the credentials 312 are formulated through one-way cryptographic operation on characteristics 310 and a client key. EX1004, 12:51-60. Based on Lundblade’s disclosure of “different cryptographic algorithms,” a POSITA would be motivated to seek out a reference teaching ways of encrypting data, such as public-key algorithms, shared-secret-key algorithms, and keyed hashes. Indeed, various cryptographic algorithms for encrypting data were known, including those applying public key, shared secret, and keyed hashes. *See, e.g.,* EX1037 (*Secure Programming*). Chen is one such reference.

a. Chen’s Teachings

218. Chen teaches a credential server for generating a password and encrypting a password using a key for establishing secure communication (*generating a secure device credential including encrypting the credential with a key known by at least one element of the network system*). EX1008 (*Chen*), 8:16-59,

FIG. 3. The credential server receives a request for a credential from a client. EX1008, 8:35-36. In response, the credential server generates a password. EX1008, 8:38-51. The password is encrypted using a 128-bit key to generate a 16-byte binary ciphertext. EX1008, 8:52-58. This text is transmitted to the client for use in secure communications with servers that possess the key, allowing for secure communications. EX1008, 8:58-62.



EX1008, FIG. 3.

219. The combination of Lundblade and Chen uses the process of generating credentials 312 as taught by Lundblade, except that the generated credentials are encrypted with a known key by server 400 before being sent to the device, as taught by Chen. Specifically, the unique client key 409 and generic key 410 of Lundblade are modified to be 128-bit keys, and the server 400 of Lundblade uses these keys to encrypt the credentials 312 before they are sent to the client device. In this way, the “secure mode” implemented by Lundblade further includes encrypting credentials 312 using 128-bit keys before the credentials 312 are sent to client 300. EX1004, 17:11-20.

b. Motivation to Combine with Chen

220. A POSITA would have found it obvious and been motivated to modify Lundblade’s server 400 to encrypt credentials 312 using the unique client key 409 and generic key 410 prior to sending them to the client device as taught by Chen. EX1008, 8:35-57. Such a modification is nothing more than the use of a known technique (encryption) to improve methods (establishing secure communication) in the same way, by increasing the security of the credentials 312 sent to client 300 and saved in memory 402 of server 400. In encrypting the credentials 312, the credentials 312 would be harder for a bad actor to intercept, replicate, or spoof, when accessing server 400 or eavesdropping on communications between client 300 and server 400. EX1004, 15:11-13 (credentials 312 stored in memory 402 or server 400), 17:4-6

(sending credentials 312 to client 300). Lundblade already focuses on securing the credentials 312 by using a “secure mode” when sending credential 312 to the client 300 and by generating the credentials 312 using cryptographic algorithms. EX1004, 17:4-6, 12:51-60. Thus, a POSITA would have understood using 128-bit keys to additionally encrypt the credentials 312 when in storage at server 400 would provide the added benefit of protecting the confidential information if a bad actor accesses server 400.

221. Furthermore, though Lundblade teaches using one-way cryptographic algorithms for generating credentials 312, a POSITA would appreciate additionally encrypting the credentials 312 to increase the security of the credentials 312 so that only devices with access to the relevant key can decrypt them. Using shared encryption keys to encrypt data (either asymmetric or symmetric algorithms) requires the receiving device to have access to the relevant key to decrypt the said data. *See, e.g.,* EX1013 (*Hind*), 8:11-22 (describing using private/public key encryption to sign a “hash” as being well-known, and that the private/public key requires the encrypting entity to have “the private key corresponding to the public key”); EX1014 (*Baldwin*), [0221], [0226]. Encrypting the credentials using a key known only to the network (but not the client) would prevent a compromised client from leaking the individual components making up credentials 312, a problem Lundblade was concerned with. EX1004 (*Lundblade*), 2:13-41. Thus, a POSITA

would appreciate encrypting the credentials 312 with keys known only to network elements, such that the network elements (e.g., server 400) can decrypt and accordingly authenticate credentials 312 received from client 300, while keeping the credentials 312 at the client from being easily forged by a hacker who gains access to client 300. EX1004, 17:8-10 (disclosing client 300 storing credentials 312, thus leaving credentials 312 potentially vulnerable if client becomes compromised). Encryption was well-known to provide the advantage of securing data. *See, e.g.*, ¶ 49; EX1037 (*Secure Programming*), 151 (describing encrypting as keeping both the “system” and data on the system “secure”).

222. Further, a POSITA would have found a reasonable expectation of success in modifying Lundblade’s network system as server 400 already uses cryptographic algorithms and keys stored at the server to generate the credentials 312. EX1004, 12:51-60. Lundblade expressly teaches credentials being “formulated using the same or different cryptographic algorithms,” indicating Lundblade contemplated using a variety of encryption algorithms as would be known to a POSITA. A POSITA would have understood that similar cryptographic algorithms (such as the encryption taught by Chen) can be used to encrypt the credential prior to sending the credential to the client device to protect the credential 312 when stored on server 400 and when sent and stored on client 300 from attackers. Thus, all that is necessary to implement the teachings of Chen is to utilize those keys to encrypt

the credentials 312 before sending them to the client device. In the modifications, the credentials 312 would be encrypted via server 400 using a key only server 400 knows, to keep credentials 312 from being easily accessed by a hacker when a client 300 is compromised, and this would not require undue experimentation, given various key-based encryptions were known. *See* ¶ 217.

XI. GROUND 4: CLAIMS 5 AND 16 ARE OBVIOUS OVER LUNDBLADE, JOBST, HARDJONO, AND LUNDBLADE '330

- A. Claim 5: “The method of claim 1, further comprising routing a network message to the identified first application over the secure message link, using the first secured application credential to indicate that the identified first application is to receive the network message.”**

223. Lundblade teaches negotiating a secure message link using credentials 312 between server 400 and client 300. *See* Claim 1[f]. Lundblade-Jobst-Hardjono also teaches generating and sending a trust report 220 (i.e., secured application credential) to client 300 from server 400, with the trust report being for a first identified application. *See* Claim 1[g]-1[i]. Hardjono teaches using the trust report as a “trust ticket” when the application on client device 300 requests another network resource, and that the trust report may be verified by an authenticator when sent from client device 300 to provide access to the other network resource. EX1006, [0041]. The server stores the trust report. EX1006, [0043]. Hardjono also teaches that such resources are used by applications on the device and can be stored on a server. EX1006, [0031]. However, Lundblade-Jobst-Hardjono does not expressly teach how

this “trust ticket” is subsequently used to route messages. Accordingly, a POSITA would be motivated to seek out a reference teaching the use of application credentials to access resources, given that Hardjono is silent to certain details of the resource access process when using a previously provided trust report 220. Lundblade ’330 is one such reference.

a. Lundblade ’330’s Teachings

224. Lundblade ’330 teaches using an application credential to request resources, such as data, from a server. EX1009 (*Lundblade ’330*), [0048]-[0060]. The application credential is generated from an application’s ID 212. EX1009, [0048]. The application credential is used to request data for use by the application on the device. EX1009, [0050]-[0051]. Specifically, Lundblade ’330 teaches transmitting a secured application credential to a data server when requesting data, and if the application credential is authenticated, sending the data back to the application for which the application credential was generated. The secured application credential 208 is transmitted to the data server in the request for data. EX1009, [0056]. A credential server then checks the credential 208 against an expected application credential (i.e., “server credential”), and if they match (i.e., the application is authentic), the data is sent back to the application, thus providing the requested resource. EX1009, [0059]-[0060], [0051]. Lundblade ’330 teaches data server 110, which provides the data when the application is authenticated, being part

EX1009, FIG. 1 (annotated). The data 126 is a message that is communicated using network 104 between data server 110 and device 102. Furthermore, data server 110 is in network 104, thus the data sent from data server 110 is a *network message*.



IPR2025-01572
Apple EX1003 Page 178

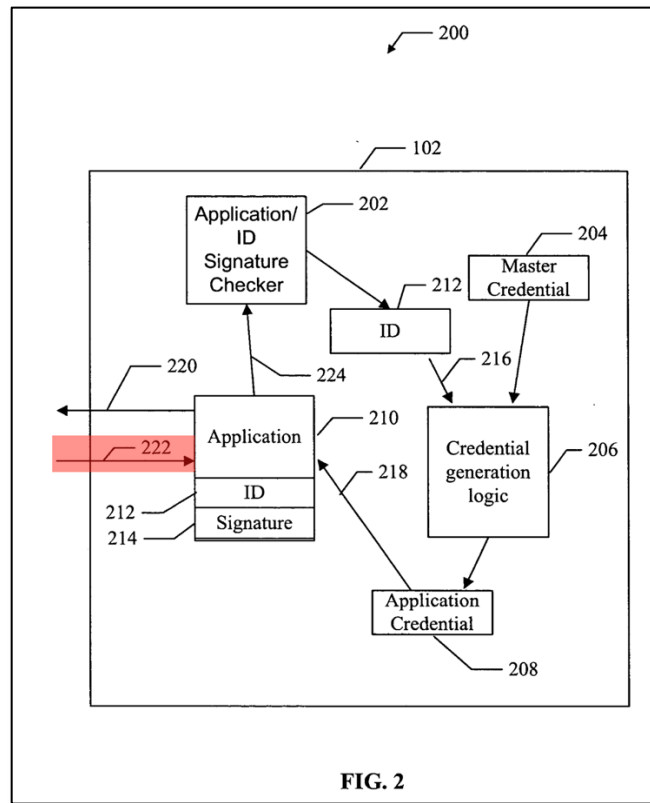


FIG. 2

EX1009, FIG. 2 (annotated to highlight the data path to the application in a mobile device).

225. The combination uses the trust report of Hardjono in the manner specified by Lundblade '330 to obtain a resource from the server. Specifically, after the trust ticket is obtained from server 400 as discussed in claim 1[Pre]-[i], the trust ticket is sent to server 400 to obtain an additional resource. Upon receiving the trust ticket, the server 400 compares it to the trust ticket stored on the server (as taught by both Hardjono and Lundblade '330). If the trust ticket is verified, the resource (data) is sent to the requesting application associated with the trust ticket in the mobile device in the manner taught by Lundblade '330. As taught by Lundblade, every

communication between the client 300 and server 400 is done using secure communications. *See Claim 1[f].* Thus, the combination teaches routing a resource to the application of the trust report over secure communications (*routing a network message to the identified first application over the secure message link*). Furthermore, the trust report is generated for the application requesting the resource (the digest of the application used to generate that specific trust report, *see Claim 1[g]-1[h]*). Thus, a POSITA would have understood that the trust report indicates the application to receive the resource (*using the first secured application credential to indicate that the identified first application is to receive the network message*). Namely, a trust report for a single application would not provide network resources to other, unverified applications on the device. Additionally, the trust report is generated when a specific application requests a resource. *See Claim 1[g].* Thus, when the same specific application requests another resource using the trust report, a POSITA would have understood the trust report 220 to indicate the specific application as receiving the resource, given the trust report contains a reference to the digest of that software application. *See Claim 1[h].* As a result, the trust report identifies the application that both requested the resource and, accordingly, should receive the resource.

b. Motivation to combine with Lundblade '330

226. A POSITA would have been motivated and found it obvious to modify Lundblade-Jobst-Hardjono's network system to route a network message (such as data requested through the network) to the identified software component in client 300 that made the request, per Lundblade '330. For example, in the modified system, when a software application on client 300 requests data stored on the network (i.e., a network resource), the request would include the trust report, which would be verified by server 400. Server 400, upon verifying the software application, would then route a message containing the data to client 300. Such a modification is nothing more than the use of a known technique (routing network messages to an application based on the application credential) to improve similar methods (providing software components (applications) access to network resources) in the same way by ensuring the application requesting the network resource is trusted and authentic prior to providing the requested information. Using an application credential to authenticate the application and, accordingly, provide the application network access, such as access to a resource, was a well-known benefit in the art to protect network resources. *See* ¶¶ 43, 53-54.

227. A POSITA would have been motivated to route a network message to the device using the trust report 220 to ensure the correct (and authenticated) requesting software receives the network resource access. Applications being tampered with or infected due to viruses or malware was a well-known problem to

network security. *See* ¶ 43; EX1006, [0006]-[0008]. To combat such issues, applications were authenticated to ensure the application is trusted and authorized to access the network. *See* ¶ 43. Thus, a POSITA would appreciate routing the requested network resource back to the actual authenticated application (i.e., the application of trust report 220) to ensure unauthorized applications (that may be compromised) do not gain access to sensitive or confidential information on network.

228. Further, A POSITA would have had a reasonable expectation of success when making the modification, given the Lundblade-Jobst-Hardjono system already verifies an application before granting access to a network resource (e.g., before transmitting data for that application to a device). *See* Claim 1[g]. Additionally, both Lundblade '330 and Hardjono disclose systems for authenticating an application credential (Hardjono's "trust ticket" and Lundblade '330's application credential) prior to sending a resource (e.g., data stored on network) to the requesting application on the device. EX1006, [0041]; EX1009, [0051], [0059]-[0060]. Thus, all that is required to make the combination is simple programming changes to the software of server 400 and client device 300 to route a requested network resource to the application specified in the trust report 220. Hardjono already teaches a client using the trust report after being issued it for other resource requests. EX1006, [0041]. Thus, for example, server 400 would be programmed to verify trust report

220 received from client 300 with a stored trust report 220 at server 400 to authenticate the application, and then, upon verification, provide access to the network resource, such as sending the requested data to client 300. This would not have been challenging and would not have required undue experimentation, given the Lundblade-Jobst-Hardjono system already has a server storing the generated trust report 220 and contemplates verifying the authenticity of a received trust report 220 from a client. EX1006, [0041], [01014].

B. Claim 16: “The method of claim 1, further comprising forwarding network messages to the first application on the first device over the secure message link, based on the first secured application credential.”

229. *See* Claim 5.

XII. GROUND 5: CLAIM 12 IS OBVIOUS OVER LUNDBLADE, JOBST, HARDJONO, AND FOSBACK

A. Claim 12: “The method of claim 11, wherein the known-application credential is uploaded to the network system via an application developer service design center.”

230. The Lundblade-Jobst-Hardjono combination teaches performing a verification procedure that compares a digest within an integrity report 170 sent to server 400 with a digest in integrity records 320 stored on the server 400 as part of generating a trust report 220. *See* Claim 11. Hardjono teaches that the digests are formed from raw information 305 of software components that is harvested “from their sources (e.g., manufacturers, software vendors, and hardware vendors, among

other possibilities).” EX1006, [0045]. Additionally, Hardjono teaches such raw information (e.g., digests) can be obtained “directly from the manufacturer” by uploading from the manufacturer or “otherwise obtaining the information through a direct means or using other techniques.” EX1006, [0045]. As discussed above with respect to claim 13, a POSITA would understand Hardjono to teach that the credential (i.e., harvested digest) should come from the source of the application (such as an application developer) for chain-of-custody purposes. *See* Claim 13. Ensuring that credentials are distributed by the original application developer would provide the highest assurance of authenticity, aligning with Hardjono’s emphasis on securing the “chain of custody.” *See* Claim 13. A POSITA would have understood that relying on the developer’s own credentials minimizes the risk of tampering or substitution from intermediate sources. Accordingly, based on Hardjono’s teachings, a POSITA would be motivated to seek out a reference that teaches how an app developer can make *known-application credentials* available.

a. Fosback’s Teachings

231. Fosback teaches uploading applications from a product submission and management system 104. EX1010, [0040]. The product submission and management system 104 allows an application developer (i.e., “digital product submitter”) to log in and submit new products (i.e., upload new software), edit previous submissions, review and submit contracts, monitor distribution of their

digital products, and manage access permissions, email settings, and account information to be associated with the digital products. EX1010, [0054]. A user may then download software from the product distribution site 102 containing the product submission and management system 104. EX1010, [0043]. A POSITA would have understood the product submission and distribution site 102 is an *application developer service design center*, because the product submission and distribution site allows a developer to track the usage of the application and view access permissions. EX1001 (*'055 Patent*), 23:39-43. For example, Fosback teaches the product submission and management system 104 as allowing developers to “access monthly royalty reports,” as well as “provide centralized account management for the digital products, including, for example, access permissions” to the applications (i.e., users permitted access to the product). This is much like the *'055 Patent* describing an *application developer service design center* as allowing developers to track usage of the application and verify access policies for the application.

232. The combination modifies the Lundblade-Jobst-Hardjono system to incorporate the product submission and distribution site 102 of Fosback to process the submission of digital products and authorize distribution. The Lundblade-Jobst-Hardjono system harvests digests of the applications from the product submission and distribution site 102 (*the known-application credential is uploaded...via an application developer service design center*), in the same manner that the digests

would be harvested directly from a manufacturer. In order to verify such applications, a POSITA would be motivated to harvest the record 320 digests from the source of download, i.e., from the product distribution site the developer used to submit the software. In particular, a POSITA would have recognized that obtaining digests directly from the developer's distribution channel reduces the possibility of modification and/or tampering, thereby ensuring a higher level of trust in Hardjono's verification process. In this way, the record 320 comes from as close to the source (i.e., the developer) as possible and thus makes the comparison process more accurate. EX1006, [0045].

b. Motivation to combine with Fosback

233. A POSITA would have been motivated and found it obvious to incorporate the production submission and distribution site 102 into the system of the Lundblade-Jobst-Hardjono combination and to receive digests from site 102. Doing so ensures that the digests originate from the same trusted source the application was originally submitted to by developer and that manages application submissions, thereby strengthening the "chain of custody" and increasing confidence in the accuracy of the verification process. *See* Claim 13. A POSITA would appreciate that a product submission and management site provides a convenient location to access information related to applications produced from a multitude of independent software developers (so that it can be, for example, provided to users

via an app store such as product distribution site 102). Centralizing this information at a trusted site would facilitate consistent access to accurate digests, reducing the risk of discrepancies across different distribution channels. Additionally, because the product submission and management site has to approve the developer software prior to accepting the submission, a stronger trust in the legitimacy of the software is provided when harvesting. EX1010 (*Fosback*), [0069]. Thus, a POSITA would have been motivated to use such a site to access a large database of digests without reaching out to individual developers. This streamlined approach conserves computational resources and saves time for the network system harvesting the digests, while still ensuring that the digests come from a trusted source (i.e., developer submissions). Such a modification is nothing more than applying a known technique (i.e., uploading software from a product distribution site) to a known method (i.e., resource verification via authenticator 145 and integrity authority 140) ready for improvement to yield predictable results of making software applications more easily and readily available for authentication via the resource verification procedure. Indeed, utilizing a product submission and management system within Lundblade-Jobst-Hardjono would make harvesting digests from trusted sources much more efficient.

234. A POSITA would similarly recognize that the user of client 300 would appreciate access to the variety of applications located on the product distribution

site 102 while still implementing the authentication entity 145/integrity authority 140 to ensure the application has not been compromised when attempting to access network resources. EX1006, [0006], [0008]. Indeed, much like the harvesting being more efficient for a multitude of software applications from various developers, a user would similarly appreciate being able to easily locate and download such multitude of applications from the product submission and management site. EX1010, [0043].

235. Furthermore, a POSITA would have had a reasonable probability of success in making such a modification, given that the modified system already harvests digests from software vendors and directly from the manufacturer and that Hardjono expressly states “other possibilities” of software sources. EX1006, [0045]. For example, Hardjono teaches the harvested digests coming from a variety of sources, including harvesting directly from “a purchased version, directly from the manufacturer by uploading or otherwise obtaining the information, or using other techniques.” *Id.* Additionally, the network system contains application servers 106 from which applications can be downloaded. EX1004 (*Lundblade*), 10:19-26. Given these teachings, a POSITA would have found it simple to modify the system to upload software from a product distribution site 102. All that is required is connecting a product distribution site to the network system and programming the integrity authority 140 to harvest digests from the applications available on the

distribution site 102. For example, the product distribution site would become part of the network system, corresponding to the application servers 106, and would be communicatively coupled to server 400 (which contains integrity authority 140 functionality). The modification would not require undue experimentation because, in the same way that server 400 would harvest digest from other sources, server 400 would also be programmed to harvest digests from the product distribution site.

XIII. GROUND 6: CLAIM 17 IS OBVIOUS OVER LUNDBLADE, JOBST, HARDJONO, AND SCARLATA

- A. Claim 17: “The method of claim 1, wherein the secure device credential is a credential associated with a service processor on the first device, the method further comprising confirming the identity of the service processor based at least in part on the secure device credential.”**

236. Lundblade teaches generating credentials 312 specific to client device 300 for secure communication using characteristics of the device and/subscriber. *See* Claim 1[c]. Lundblade also teaches providing secure communication between a client security module 214 of the client device 300 and a server 400. *See* Claim 1[f]; EX1004, FIG. 2, 11:1-4, 11:34-39. Finally, Lundblade teaches a “secure mode,” but does not provide details about the hardware used to implement the secure mode using the client security module 214. EX1004, 17:11-20. For example, the “secure mode” includes “the use of a secure file system” and uses encryption, but various details of hardware used in implementing these security measures are not present in Lundblade. EX1004, 17:11-20. Accordingly, a POSITA would be motivated to look

to existing hardware to support Lundblade's secure file system and secure communications. Indeed, a POSITA viewing Lundblade would have been motivated to seek out references explaining hardware for accomplishing the secure mode, to prevent the "unwanted snooping" Lundblade refers to. EX1004, 17:11-20. Scarlata is one such reference.

a. Scarlata's Teachings

237. Scarlata provides further detail regarding a secured device credential. Namely, Scarlata teaches a processing system (such as a computer, tablet, telephone, or PDA) with a Trusted Platform Module (TPM) 30. EX1011 (*Scarlata*), 3:20-32, 4:12-22. The TPM 30 includes a program code segment, a processor, non-volatile memory, and an active memory. EX1011, 8:48-59. The TPM 30 "provides security services such as attesting to the identity and/or integrity of the platform, based on characteristics of the platform." EX1011, 1:53-55. The TPM 30 receives a certificate from a privacy certification authority (CA) 76 to make a trustworthiness assertion to a remote processing system. EX1011, 7:51-57. Thus, the TPM contains a processor and helps provide communication across a network. This is much like the '055 Patent describing a "service processor" on the device that governs communication activity over a network. EX1001 (*'055 Patent*), 8:10-20. Thus, Scarlata's TPM is a *service processor*.

238. Furthermore, the identity credential sent back from the certification authority is used to confirm the TPM is trustworthy. As an exemplary scenario, Scarlata discloses, in a conventional attestation approach, that a “privacy CA 76” provides an identity credential to the TPM in a processing system 20, and the processor system 20 “could use that identity credential to make a trustworthiness assertion to a remote processing system such as a challenger 78.” EX1011, FIG. 1, 7:51-57. The processing system 20 thus uses the TPM identity credential to assert proof of trustworthiness to the remote processing system, indicating to a POSITA that the remote processing system in a network, in response to the assertion, confirms the identity of the TPM (i.e., *service processor*) based at least in part on the received identity credential. Indeed, attestation commonly involved a remote recipient of the TPM credential (e.g., AIK certificate) validating the certificate to ensure it came from a “genuine TPM.” EX1057 (*Attestation Explained*), 2 (describing a TPM forwarding an AIK certificate to a verifier (i.e., recipient) to “authenticate itself”). Thus, the remote processing system (e.g., server) in network verifies the TPM is trusted using the identity credential. A third-party can rely on the identity credential “as cryptographic proof of the current state of the platform.” EX1011, 7:47-51.

239. In the combination, the process of establishing secure combinations as taught by Lundblade-Jobst (*see* Claim 1[Pre]-[f]) is modified to utilize the TPM of Scarlata. Specifically, the client security module is implemented using Scarlata’s

TPM, which is responsible for secure communications “under the secure mode” taught to be desirable by Lundblade. EX1004, 17:15-17. A TPM is used “to protect data and to attest to” the trustworthiness of the device to enhance the security of the device. EX1011, 1:36-39. TPMs aid in protecting data by storing “sensitive information, such as encryption keys” used by the device to secure data, as well as by verifying “file integrity.” EX1011, 1:47-52, 1:59-63. Given this, the client security module (now implemented using Scarlata’s TPM) that provides secure communications “under the secure mode,” would provide enhanced security, such as to Lundblade’s “secure file system.” EX1058 (*Putting the TPM to Work*), 1 (describing a TPM facilitating key management and encryption/decryption of data, such as emails and “digital documents”).

240. A POSITA would have understood that a TPM is a type of hardware security module that was commonly used for secure processing and communications. Indeed, TPMs were known to provide security to enterprise networks by increasing the security of the devices themselves and “implementing network access controls to improve overall endpoint security. EX1058, 1. (Hardware) TPMs were often used to implement (software) security modules, as TPMs secure data and help aid in securing communications. *See, e.g.*, EX1059 (*Kotani*), Abstract (describing providing a “high level of security” during data migration via a “security module such as supported by a trusted platform module

(TPM)"); EX1028 (*Camenisch*), [0016] (describing a TPM “which may also be called a security module”). Prior to the process taught by the Lundblade-Jobst-Hardjono combination, the TPM receives an identity credential from a privacy certification authority. EX1011 (*Scarlata*), 7:40-44.

241. As discussed above with respect to claim 1, the Lundblade-Jobst-Hardjono system generates credentials 312 (i.e., *secure device credential*) via server 400 from the characteristics 310 in the request, per Lundblade. *See* Claim 1[a]-1[c]. The secure communications channel between the client 300 and server 400 is established when the server 400 verifies that the received credentials 312 match the expected credentials stored at server 400. *See* Claim 1[f]. Then, when client 300 requests a network resource through a software application, server 400 reviews the integrity report 170 sent from client 300 that contains the software digest (i.e., *general application credential*) and compares the digests to expected digests in records 320 to generate the trust report (i.e., *secured application credential*). *See* Claims 1[g]-1[h]. In the Lundblade-Jobst-Hardjono system, as modified by *Scarlata*, the identity credential of the TPM is provided to the server 400 in the request for credentials, along with the other characteristics 310, and together they are used to generate credentials 312 (*wherein the secure device credential is a credential associated with a service processor on the first device*). All secure communications between the client device 300 and server 400 compare received credentials 312

(generated from the identity credential of the TPM) with stored credentials (*comprising confirming the identity of the service processor based at least in part on the secure device credential*). See Claim 1[f]. Subsequently, the TPM may be used to secure the *general application credential* (i.e., the digest (hash) within integrity report 170), with the integrity report 170 also containing credentials 312. See Claim 1[g]. Thus, in the modified system, client 300 uses the TPM identity credential to sign the generated software application digest. Server 400 verifies the TPM signature comes from the TPM identity credential of the client to confirm that the integrity report digest comes from the trusted device, then uses the TPM signature to verify that the digest is authentic, and generates the trust report by comparing the digest to the record 320 digests. Such verification of the TPM identity credential prevents compromised devices from sending falsified general application credentials in requests for a secured application credential. For example (discussed below), verifying a TPM identity credential for a TPM signature in the digest attests that the device is trusted, ensuring a bad actor is not providing a digest spoofing the device to gain network resource access.

b. Motivation to Combine with Scarlata

242. A POSITA would have found it obvious and been motivated to incorporate the TPM and its functionality as taught by Scarlata into the Lundblade-Jobst-Hardjono combination. As discussed above, a POSITA would have modified

the Lundblade-Jobst-Hardjono system to provide an identity credential of the TPM in the characteristics 310 (along with the other characteristics 310) in the request for credentials, to use the characteristics 310 (including the TPM credential) to generate the credentials 312, to sign the integrity report digests with the TPM credential when requesting resource access, and finally to validate the TPM signature. Such a modification is nothing more than using a known technique (implementing a TPM and using an identity credential generated by the TPM to confirm the trustworthiness of the client) to improve similar systems (i.e., network systems providing device authentication for security) in the same way by allowing the server 400 to confirm the client 300 is trustworthy based on the identity of the TPM. Indeed, both the hardware and software of the device are vulnerable to malware attacks or other incidents that would otherwise compromise the integrity of these components. *See* ¶¶ 41, 43; *see also*, EX1006 (*Hardjono*), [0005]-[0006]. For example, impostors “attempt to impersonate” devices by “spoofing” identifying information, such as credentials or the unique identifier. EX1013 (*Hind*), 12:57-60; EX1004, 2:66-3:3 (describing the issue of a bad actor gaining access to credentials “that might allow unauthorized entities to spoof the identity of an otherwise authorized device”). Using TPM identifiers prevents such attacks, given that the TPM has a private key used to generate identity credentials, making these credentials more difficult to spoof compared to a device identifier. EX1060 (*Security Assessment*), 8; EX1061 (*The*

Role of TPM in Enterprise Security), 2 (stating the TPM coprocessor’s “state cannot be compromised by potentially malicious host system software,” and thus TPM-based attestation is a “powerful tool for establishing trust” of a device).

243. Accordingly, a POSITA would have understood that utilizing a TPM identifier provides added security over only verifying a device identifier by more rigorously ensuring that the client-provided general application credentials are authentic (i.e., come from the actual device, and are not spoofed by a hacker). Indeed, TPM-signed data provides additional security that the data is not “tampered with” and that the “TPM is actually the one on the attested system,” thereby ensuring any data, such as client-provided digests, comes from the actual device and is not spoofed. EX1061, 4. Namely, an application credential, such as the digests, supplied by the client is generally simple and can be spoofed by malicious actors, while the credential supplied by a TPM is difficult to spoof, providing extra insurance that the device generating the application credential (i.e., digest) is trustworthy. EX1060, 8 (a TPM credential is harder to spoof because it is generated from a private key securely stored and only known by the TPM). Indeed, TPMs were used for exactly this purpose: attesting to the legitimate identity of software and the device holding the software. EX1011 (*Scarlata*), 1:53-59, 7:47-57 (disclosing “a quote signed by an AIK that is accompanied by an identity credential is cryptographic proof of the

current state of the platform,” including any software digests signed using the AIK and identity credential).

244. Additionally, Lundblade teaches using a security module 214 in providing secure communications, indicating Lundblade is concerned with keeping communications between client 300 and server 400 secure from attacks, such as spoofing. EX1004, 11:1-4, 11:37-41. Accordingly, a POSITA would have been motivated to include a TPM for performing the logic of security module 214 to provide further enhanced security for the client 300 and communications therefrom, given TPMs were known to provide devices with “various facilities and services for enhancing the security of the processing system.” EX1011 (*Scarlata*), 1:31-39. In addition, a POSITA would also have been motivated to use a TPM in the Lundblade-Jobst-Hardjono system, given Hardjono teaches using TPM certificates within integrity report 170, which are generated by a TPM, like that of *Scarlata*. EX1006 (*Hardjono*), [0076].

245. Furthermore, a POSITA would have had a reasonable expectation of success in making the required modifications. A POSITA would have found the modification simple. Lundblade already teaches credentials 312 being associated with other device characteristics 310, and these characteristics 310 may include “other information” that is “useful in identifying the client computing device 300.” EX1004, 12:29-46. Additionally, Lundblade teaches using a security module 214 on

client 300 that has logic for providing secure communications between client 300 and server 400. EX1004, 11:34-37. The security module 214 provides such secure communications by using credentials 312 to authenticate the device. EX1004, 17:61-18:6. Thus, all that is required to modify the system is the inclusion of the identity credential of the TPM as a characteristic 310 when generating the credentials 312 and signing the generated software digest within integrity report 170 with the TPM identity credential. Accordingly, when the server 400 compares the credentials 312 to stored credentials to authenticate signals from client 300, the server 400 would confirm the identity of the TPM based on credentials 312 (i.e., *confirming the identity of the service processor based at least in part on the secure device credential*). EX1004, 17:61-18:6. Additionally, when server 400 receives integrity report 170, server 400 would verify the TPM signature to ensure the generated digests actually come from the client 300. These modifications would not have been challenging nor would they have required undue experimentation, as processing systems, such as PCs, mobile phones, etc. were known to include TPMs, TPMs were often used to implement security modules, and generating TPM signed data to provide validation was known. *See* ¶ 240 (explaining TPMs supporting security modules); EX1059 (*Kotani*), 2:5-14 (describing TPMs as commonly being implemented in PCs, cellphones, etc.); *see* ¶ 242 (explaining TPM signing of data being known).

XIV. CONCLUSION

246. I declare that all statements made herein of my knowledge are true, and that all statements made on information and belief are believed to be true, and that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Dated: October 1, 2025

By:


Dr. Charles D. Creusere

All cases since 2014:

- Technical expert for defense (Evaluated expired PD patent and wrote Dec. demonstrating Dell could not have infringed, case was dropped); Case: Princeton Digital v. Dell, 2014-2015, case dismissed June 2015, **no deposition**.
-> **A company named Princeton Digital was threatening to sue Dell over possible infringement on an expired patent related to JPEG image compression. I did some analysis to show that PD COULD have easily determined prior to patent expiration whether or not Dell actually infringed.**
- Technical expert for defense; Noninfringement & IPR (6,597,812), Real-time v. SAP/HP Enterprises, 1/2016-6/2016, **2 depositions (IPR)**. -> **the patents in suit were related to lossless compression technology. Patent in IPR suit dealt with combined run-length coding and dictionary-based coding.**
- Technical expert for defense; IPRs (7,378,992 & 7,415,513), Real-time v. Riverbed (and others), 2/2016-present, **1 deposition**. -> **The patents in suit were related to lossless compression technology with the specific being to provide content dependent and independent compression**
- Technical expert for defense; IPRs (8,643,530, 7,378,992, 7,161,506, & 9,054,728), Real-time v.
- Dell (and others), 2/2016-7/2018, **2 depositions**. -> **All of the patents in suit were related to lossless data compression. One set dealt with "accelerated data storage and retrieval" while the other was related to adaptive content dependent/independent compression**
- Technical expert for defense; Noninfringement, Real-time v. HP Enterprises, 4/2016-10/2018, **not deposited, settled before deposition**. -> **Dealt with noninfringement for patents listed for the IPRs above.**
- Technical expert for defense; IPR (7,358,867, 7,161,506, & 9,054,728), Real-time v. Teradata
- 11/2016-8/2018, **1 deposition**. -> **Patents dealt with lossless compression. Specifically, patents claimed that different encoding schemes would be selected for each of a plurality of data blocks based on some criterion. Specific to one of these patents, the encoder could also choose NOT to encode a block if that was desirable.**
- Technical expert for defense; IPRs (8,643,513 & 7,378,992), Real-time v. Veritas 12/2016-2018, **no additional deposition -> Duplicate of Dell IPR above**
- Technical expert for defense; IPRs (7,151,916; 7,075,917; 6,304,612) , Uniloc LLC v. Apple, 2018-2019, **not deposited**. -> **Patents dealt with aspects of communications systems, including tail-biting convolutional coding and ARQ systems for error correction**
- Technical expert for defense; Noninfringement, Realtime Adaptive Systems v. YouTube/Google, 2018-2019, **1 deposition, settled before trial**.
- Technical expert for defense; IPR (7,558,730), AVRS v. Apple, 2019-2020, **2 depositions**. -> **dealt with an internet-based voice transcription service.**
- Technical expert for defense; IPR (10,129,590), Maxell v. Apple, 2019-2020, Oct. 2020-2021. -> **Dealt with video compression. Specifically, a methodology for capturing video from a device and transmitting it wirelessly to at least two other devices (one through the internet) where one device has a higher priority than the other. Settled before deposition.**
- Technical expert for defense; U.S. ITC case; Nokia v. Lenovo, Oct. 2020-March 2021 -> **Patents related to H.264 video compression**. Hired as a testifying expert witness but ultimately have provided primarily technical analysis of opposing expert witness report. **No deposition.**

- Technical expert for defense; IPR (10,176,848) , Maxell v. Apple, 2019-2020, Dec. 2020-2021. -> **Deals with video compression. Specifically, a methodology for identifying a person of interest in recorded digital video. Settled before deposition.**
- Technical expert for defense; Claims construction, Gesture Tech v. Apple/Lenovo/Motorola, 11/2021-12/2021, 8/2023. **No depositions.**
- Technical expert for plaintiff; USAA v. PNC Bank; October 2021-January 2023; Expert Arguing against invalidity in two U.S. District Court cases (2:20-cv-00319-JRG and 2:21-cv-00246-JRG) and 4 IPRs (IPR2022-00075, IPR2022-00076, IPR2021-01073, and IPR2021-01070). **All patents dealt with mobile remote check deposit. Deposed 4 times and attended both trials but did not testify.**
- Technical expert for the defense; US ITC case; VideoLabs, Inc. v. HP Inc; August 2023 scheduled; **Patents invalidated by IPRs prior to start of ITC case.**
- Technical expert for defense; Federal District Court; Gesture Partners v. Lenovo/Motorola, December 2023-present. Wrote invalidity and noninfringement reports (source code and device analysis), Deposed April 2024. USP 8,878,949 dealt with optical-based gesture recognition. **Summary judgement of noninfringement, September 2024.**
- Technical expert for defense; IPRs; Nokia v. Amazon; USP 10,536,714, USP 8,077,991; 2024-2025; **'714 Deposed; Case settled May 2025**
- Technical expert for defense; IPRs; Maxell v. Samsung; USP 11,445,241; 2024-present; **Deposed.**
- Technical expert for defense; Ex Parte Reexam; USP 6,389,473; BSD Crown LTD. v. Amazon; July-August 2024.
- Technical expert for defense; IPRs; Nokia v. Amazon; USP 9,571,833; 2024-May 2025; **Case settled May 2025.**
- Technical expert for defense; IPR; Erikson v. Lenovo; USP 10,708,618; 2024-May 2025; **Case settled May 2025.**
- Technical expert for defense; IPR; Nokia v. HiSense & Snap; USP 10,536,714; 2025-present; **Awaiting institution decision**