

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner,

v.

HEADWATER RESEARCH LLC,
Patent Owner.

IPR2026-00203
U.S. Patent No. 9,232,403 B2

**DECLARATION OF ERIK DE LA IGLESIA IN SUPPORT OF PATENT
OWNER'S RESPONSE**

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

Executed on October 8, 2026, at Mountain View, California.



Erik de la Iglesia

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	BACKGROUND AND QUALIFICATIONS	2
III.	LEGAL PRINCIPLES	6
A.	Claim Construction	6
B.	Burden of Proof	7
C.	Obviousness	7
IV.	OVERVIEW OF THE '403 PATENT	9
A.	The Claimed Device Messaging Agent and Secure Interprocess Communication Service.....	9
B.	The Challenged Claims.....	15
V.	PERSON OF ORDINARY SKILL IN THE ART	16
VI.	CLAIM CONSTRUCTION.....	16
VII.	THE ASSERTED REFERENCES AND DR. TRAYNOR'S PROPOSED COMBINATION	17
A.	TS-23.140	17
B.	Ogawa	18
C.	Dr. Traynor's Proposed Combination of TS-23.140 with Ogawa.....	23
VIII.	DR. TRAYNOR'S RELIANCE ON IPR2024-00341 DOES NOT ADDRESS THE TECHNICAL ISSUE PRESENTED HERE	24
IX.	DR. TRAYNOR HAS NOT SHOWN THAT GROUND 1 RENDERS ANY CHALLENGED CLAIM OBVIOUS.....	29
A.	Dr. Traynor Does Not Explain How His Proposed Combination Would Allow Recipient Applications to Read the Securely Transmitted Data.....	29

1. The intended purpose of Dr. Traynor’s combination is secure delivery of data that recipient applications can read.....	29
2. In Dr. Traynor’s combination, only the MMS User Agent can decrypt the data it sends.	33
3. Neither TS-23.140 nor Ogawa teaches how a recipient program could decrypt inherent-key data.	37
4. Dr. Traynor’s combination is inoperable for its intended purpose.....	41
B. Dr. Traynor’s Combination Conflicts with the Purpose of Ogawa’s Inherent Key and Undermines His Own Rationale	43
C. A POSITA Would Not Have Had a Reasonable Expectation of Success ...	48
D. Dr. Traynor also does not show the obviousness of a “secure interprocess communication service” in his proposed combination.....	50
E. I also disagree with Dr. Traynor’s new theories articulated for the first time at deposition.....	51
X. DR. TRAYNOR’S ANALYSIS OF GROUNDS 2A–4D SUFFERS FROM THE SAME DEFICIENCIES	54
XI. DR. TRAYNOR HAS NOT SHOWN THE DEPENDENT CLAIMS TO BE OBVIOUS FOR ADDITIONAL REASONS	55
A. Claim 11.....	56
B. Claim 15.....	58
C. Claim 16.....	59
D. Claim 21.....	63
XII. CONCLUSION.....	67

I, Erik de la Iglesia, hereby declare as follows:

I. Introduction

1. I am over the age of eighteen (18) years and otherwise competent to make this declaration.

2. I have been retained as an expert witness on behalf of Patent Owner for the above-captioned *inter partes* review (“IPR”). I understand that the petition for *inter partes* review involves U.S. Patent No. 9,232,403 (“the ’403 patent”).

3. I make this declaration based on my personal knowledge, educational background and training, consideration of the materials I discuss herein, and my expert opinions.

4. My work on this matter is being billed at my normal hourly consulting rate of \$750, with reimbursement for actual expenses. My compensation is not related to the outcome of any proceeding involving the ’403 patent. I have no personal interest in the outcome of the case.

5. In preparing this Declaration, I have reviewed and considered the ’403 patent (Ex. 1001), the ’403 patent’s prosecution history (Ex. 1002), the Petition (Paper 2) and the prior art references and other exhibits submitted with the Petition, including TS-23.140 (Ex. 1004) and Ogawa (Ex. 1005), the declaration submitted by Dr. Patrick Traynor in this proceeding (Ex. 1003), the transcript of Dr. Traynor’s September 21, 2026 deposition (Ex. 2014), the petition and Final

Written Decision in IPR2024-00341 (Exs. 1038, 1022), and each document cited in my declaration.

II. Background and Qualifications

6. My qualifications for forming the opinions given in this declaration are summarized here and are addressed more fully in my curriculum vitae, which is submitted as Exhibit 2013.

7. I hold a Bachelor of Science degree in Electrical Engineering from the University of Florida, and a Master of Science degree in Electrical Engineering from Stanford University. While at Stanford, I was an NSF Graduate Research Fellow. My graduate focus included VLSI CMOS architecture, high-speed circuit design, computer networks and protocols, and artificial intelligence. I have worked in several computer and electrical engineering related fields for over 25 years resulting in 68 issued U.S. patents that have been cited over 2850 times in USPTO patent applications. Since 2003, I have served in a number of chief architect and founder roles of companies that have been acquired or gone public. My experiences include the design or implementation of networks and network-based computation and protocol systems relevant to the technology of the '403 patent, including network security, messaging security, and systems that interface between applications and the operating system.

8. Between December 1997 and November 1999, I was a circuit design engineer at Intel Corporation within the Mobile and Handheld Products Group. My responsibilities included working on several aspects of mobile processors and chipsets including clock and power distribution, dynamic clock frequency selection, and the circuit design, device simulation, and validation of such features. Until March of 2000, I worked on similar functionality within the Microprocessor Products Group for the Itanium 3 and 4 products.

9. My work in power distribution included thermal modeling and predictive control of silicon devices and their packaging such as multi-tiered cooling systems for embedded, mobile, and desktop environments. Such systems included multiple tiers of thermal sensors, active and passive thermal control systems, and the feedback and analytic firmware and software control. My models were part of the development of the Thermal Design Power (TDP) metrics used by OEM and system designers to plan thermal control solutions for consumer and industrial products.

10. Between March 2000 and September 2003, I worked as a logic designer and architect for WebStacks and, through acquisition, Extreme Networks. WebStacks built a fully hardware-based TCP and HTTP processing stack for network processing functions including load balancing, proxy processing, and

content rewriting. I was responsible for the architecture and design of the HTTP processing engine including header extraction and processing.

11. Between August 2003 and August 2007, I worked as Founder, Chief Architect and Director of Engineering for Reconnex, later acquired by McAfee. Reconnex built a gigabit line-rate network security analyzer capable of classification, heuristic analysis, and policy enforcement on arbitrary protocols and content types. The Reconnex product captured and analyzed content transmitted over the internet using HTTP and other protocols, including storage protocols. Design of the Reconnex system was based on a modified and customized version of the Linux kernel which included specifically designed driver source code and interfacing between the application and kernel domains. Reconnex technology was employed in the audit and compliance verticals based on its ability to quickly analyze and classify the content of transmissions and interact with protocols and communications crossing corporate and departmental boundaries.

12. Between August 2007 and September 2008, I founded Strangways, a company focused on addressing messaging security for webmail and internet messaging protocols. Products and technology developed by Strangways were sold to SendMail Inc., Iron Port (Cisco), and other companies. The Strangways product utilized a Squid caching server and ICAP content rewrite server to translate HTTP to SMTP.

13. Between September 2008 and April 2014, I worked as Chief Architect for Gridiron Systems and, through acquisition, Senior Director of Technology for Violin Memory. I held the Senior Director position with Violin Memory both before and after the company's IPO. My responsibilities included architecture and design of flash media controllers, adaptive caching design, and machine learning for application storage behavior analytics.

14. The Gridiron product was a storage accelerator used in many analytic environments such as big data architectures with constant ingest and processing of financial, consumer and IoT data. In my role as Chief Architect, I often worked with customers to implement and accelerate their analytics environments and spoke on related topics at trade shows and industry user groups. During product development, I worked with the data science team at Shopzilla to improve database and storage performance for real-time shopping recommendation and user tracking for advertising.

15. My work at Reconnex and Strangways relates directly to the technology at issue in this proceeding, including network and messaging security, the protection of content transmitted between systems, and the interfacing between the application and kernel domains of an operating system. My work at Webstacks and Reconnex also related directly to packet processing, and approximately two dozen of my issued patents relate directly to these concepts.

16. Additional information regarding my employment history and prior expert testimony is included in Exhibit 2013. As reflected in Exhibit 2013, I have consulted on several matters involving intellectual property disputes including instances of alleged patent infringement requiring analysis of source code, reviewing and analyzing patents, writing expert reports, and testifying in court. I have testified at deposition and at trial in a number of these cases.

III. Legal Principles

A. Claim Construction

17. I understand that the first step in performing a validity analysis of the patent claims is to interpret the meaning and scope of the claims by construing the terms and phrases found in those claims. I understand that the appropriate construction of a claim term is its ordinary and accustomed meaning as understood by one of ordinary skill in the art at the time of the invention in the context of the entire patent and the prosecution history. I discuss the plain meaning of “interprocess communication service” in Section IV.A below.

18. I understand that the standard for claim construction in an *inter partes* review is the same standard as is applied in district court proceedings.

19. I understand that a determination of the meaning and scope of the claims is a matter of law. I have been informed that to determine the meaning of

the claims, one should consider the intrinsic evidence, which includes the patent's claims, written description, and prosecution history.

B. Burden of Proof

20. I understand that in an *inter partes* review, the petitioner has the burden of proving unpatentability by a preponderance of the evidence.

C. Obviousness

21. I have been instructed by counsel and understand that a combination of prior art references may render a claim obvious if, at the time of the invention, a person of ordinary skill in the art would have selected and combined those prior-art elements in the normal course of research and development to yield the claimed invention.

22. I understand that in an obviousness analysis, one should consider the *Graham* factors, including: the scope and content of the prior art; the differences between the claimed inventions and the prior art; and the level of ordinary skill in the art. I further understand the obviousness analysis is to be performed on a claim-by-claim basis. I understand that a person of ordinary skill in the art is a person of ordinary creativity, not an automaton.

23. I have been instructed by counsel and understand that obviousness requires more than a mere showing that the prior art includes separate references covering each separate limitation in a claim under examination. I understand

obviousness requires the additional showing that a person of ordinary skill at the time of the invention would have been motivated to combine those references in a manner that would include all limitations of the challenged claim, and, in making that combination, a person of ordinary skill in the art would have had a reasonable expectation of success.

24. I also understand that an obviousness analysis must be conducted with awareness of the distortion caused by hindsight bias and with caution of arguments reliant upon *ex post* reasoning. For instance, I understand that when considering obviousness, I should put myself in the position of a person of ordinary skill in the field at the time of the invention, rather than considering new information that is known today, but was not known before the priority date of the challenged patent.

25. I understand that a petitioner must provide a clear, evidence-supported account of how its proposed combination would actually work, and that conclusory statements are insufficient. I further understand that expert testimony that does not disclose the underlying facts or data on which the opinion is based is entitled to little or no weight.

26. I understand that a proposed modification that would render the prior art inoperable for its intended purpose, or that would require a change in the basic principle under which a prior art reference was designed to operate, weighs against a conclusion that the modification would have been obvious. I also understand that

where the prior art's teachings undermine the very reason being offered for why a person of ordinary skill would have combined the references, that weighs strongly against obviousness.

27. I understand that a reasonable expectation of success refers to the likelihood of success in combining references to meet the limitations of the claimed invention.

28. I understand that the Petition defines the theories of unpatentability in this proceeding, and that the petitioner is expected to identify with particularity the evidence that supports each ground. Accordingly, in this declaration I respond to the opinions Dr. Traynor set forth in his declaration (Ex. 1003), and I have noted where Dr. Traynor's deposition testimony departs from those opinions.

IV. Overview of the '403 Patent

A. The Claimed Device Messaging Agent and Secure Interprocess Communication Service

29. The '403 patent describes a device-side service processor whose agents communicate with a network-side service controller over a secure service control link. Ex. 1001, 37:37–43, 68:9–48. On the device, a service control device link 1691 receives messages from the network and delivers their contents to the appropriate device agents over an agent communication bus 1630. *Id.* 41:42–43:4, 44:62–45:5, FIG. 16.

30. The specification describes securing that bus in many different ways. It explains that “[i]n some embodiments, the agent communication bus is secured, signed, encrypted, hidden, partitioned and/or otherwise protected from unauthorized monitoring or usage,” and that “there are multiple layers of security applied to the agent communication bus 1630 communication protocols,” including “point to point message exchange encryption using one or more keys that are partially shared or shared within the service processor 115 agent group,” “point to point message exchange that using one or more keys that are private to the two endpoints of the communication,” and “a set of secure messages that can only be decoded or observed by the agents they are intended for.” *Id.* 41:55–42:6.

31. Where encryption is the means by which security is provided, the specification describes keys being provided to the communicating agents. For example, the specification describes “agent bus encryption using shared key for all agents provided and updated by the secure server,” “agent bus encryption using point to point keys in which the secure server informs the bus and agents of keys and updates as appropriate,” and “agent level encryption using agent to agent shared keys in which the secure server informs agents of the key and updates the key as appropriate.” *Id.* 42:19–28. The specification also notes that “the system has the capability to replace keys or signatures on occasion or on a regular basis.” *Id.* 42:15–18. The bus may be “a variant of D-bus” or “another inter-process

communication protocol or system, running a session bus in which all communications over the session bus can be secured, signed, encrypted or otherwise protected.” *Id.* 42:48–58; *see also id.* 88:9–22.

32. By way of technical background, an interprocess communication (“IPC”) service is a facility—typically provided by the operating system or by middleware such as a message bus—that allows separately executing processes to exchange data. On a modern operating system, each process executes in its own protected address space, so one process generally cannot read data held in another process’s memory. Data that one process wants another process to use must therefore be delivered through an IPC mechanism (for example, a message bus such as D-Bus) in a form that the receiving process can use, rather than the processes simply communicating in some fashion with no service being provided. *See* Ex. 2015 at 51 (*Distributed IPC using Virtual Device Driver in Monolithic Kernel*, noting that “large scale distributed computing applications require the efficient and transparent distributed interprocess communication (IPC) as a service from the underlying operating systems”); *id.* at 52 (“The kernel-level distributed IPC architecture creates a system-wide virtual layer of service to user applications....”); *id.* at 53 (“The registration command is used by the application processes to register to VxD in order to use distributed IPC service of the kernel.”). In my opinion, these teachings show that an IPC service is provided to the

applications that utilize it. Another reference describes “a real-time interprocess communication service (RT-IPC) suitable for distributed systems built over ATM networks” which is “implemented in the Chorus distributed operating system, as servers running in the micro-kernel address space.” Ex. 2016 (*A Real-Time IPC Service Over ATM Networks for the Chorus Distributed System*) at 1013. The article shows that the service is provided to the entities that use the service, not by those entities themselves. *See id.* at 1014 (*noting* that the “IPC user will not overtake the traffic contract admitted by the service provider”); *id.* at 1015 (introducing “a new IPC service for the Chorus system” which “support[s] the level of service expected by distributed hard real-time applications”); *id.* at 1019 (discussing “the time the message is submitted to the RT-IPC service, until its reception by the destination RT-port,” making clear that the IPC service is limited to transferring data between endpoints, rather than specifying how applications manage and verify information after receiving it).

33. In my opinion, a POSITA reading the ’403 patent would have understood the claimed “interprocess communication service” consistently with this background. An “interprocess communication service” is not merely a communication path between two processes that the processes themselves implement; it is a service that is provided to the communicating processes by some component (for example, the operating system, a message bus, or a

communications hub), and on which the messages are delivered. The '403 patent's description of its agent communication bus is consistent with that understanding. The specification describes agent communication bus 1630 as “a functional description for providing communication for the various service processor 115 agents and functions”; describes implementing it, where the agents are software, as “an inter-process software communication bus,” such as “a variant of D-bus (e.g., a message bus system for inter-process software communication that, for example, helps applications/agents to talk to one another),” or “another inter-process communication protocol or system, running a session bus in which all communications over the session bus can be secured, signed, encrypted or otherwise protected”; and describes a component (the access control integrity agent 1694 or the service control device link 1691) acting as a “central secure communications hub” for agent communications. Ex. 1001, 41:42–45, 42:48–58, 44:33–36, 44:62–67. In each case, a component other than the communicating agents provides the communication—and, where applicable, its security—to those agents.

34. The '403 patent's embodiments are illustrative, and they reflect a principle that I rely on throughout this declaration: for an encryption scheme for interprocess communications to be workable, the process receiving the encrypted data must be given what it needs (e.g., a decryption key) to read that data. In each

of the '403 patent's encryption embodiments described above, the specification identifies both the source of the keys (e.g., the secure server) and the recipients that are provided with the keys (the agents). Ex. 1001, 42:19–31.

35. This principle is not unique to the '403 patent; it is fundamental to cryptography and was well understood long before 2009. With a symmetric (or “conventional”) encryption algorithm, “the same key is used for both encryption and decryption,” and authentication “depends upon the two participants in a conversation being the only two principals (apart possibly from trusted servers) who know the key that is being used to encrypt the transmitted material.” Ex. 1030 (Needham), 994. A cryptographic design therefore must address two distinct problems: (1) how data is encrypted, and (2) how the key needed to decrypt the data is made available to the intended recipient(s) (and only the intended recipient(s)). The second problem is commonly referred to as key distribution or key management. Dr. Traynor agreed that “[k]ey generation and key distribution are two very distinct operations.” Ex. 2014, 26:10–13. A design that specifies how data is encrypted but not how the intended recipient obtains the key to decrypt it has addressed only the most surface-level aspect of an encryption system, because the primary complexity of encryption almost always depends on how the system is designed to allow for secure decryption of data.

B. The Challenged Claims

36. I understand that claims 1–21 of the '403 patent are challenged. Claim 1 is the only independent claim, and it recites:

Claim 1: [1PRE] A mobile end-user-area device comprising:

[1A] a wireless wide-area network (WWAN) modem to exchange Internet data via a connection to a first WWAN, when configured for and connected to the first WWAN;

[1B1] a device messaging agent to receive secure Internet data messages, on behalf of a plurality of software applications capable of execution on the device, and over a secure connection to a network message server reachable via the WWAN,

[1B2] wherein at least a subset of the secure Internet data messages contain an identifier for a corresponding one of the software applications and application data from a respective network application server corresponding to that application; and

[1C1] a secure interprocess communication service,

[1C2] wherein the device messaging agent, for each message in the subset of the secure Internet data messages, maps the identifier to the corresponding one of the software applications in order to forward the application data on the secure interprocess communication service to a software

process corresponding to the identified software application.

Ex. 1001, claim 1 (bracketed labels from Dr. Traynor's Claim Listing Appendix, Ex. 1003, Section XI).

V. Person of Ordinary Skill in the Art

37. Dr. Traynor opines that a person of ordinary skill in the art ("POSITA") in the early 2009 timeframe would have had at least a bachelor's degree in computer science, electrical engineering, or a related field, and 3–5 years of experience in services and application implementation in communication networks, and that more education could substitute for experience, and vice versa. Ex. 1003 ¶33. In my opinion, that formulation is correct so long as a time period of three years is applied. Stated differently, if a person has the level of ordinary skill in the art at 3 years of experience (or equivalent additional education), then a person having 5 years of experience (or equivalent) would have *more* than ordinary skill in the art. My opinions do not depend on the precise level of skill, and they apply equally under Dr. Traynor's formulation. I meet at least these requirements, and have met them since before January 2009.

VI. Claim Construction

38. I understand that Dr. Traynor does not propose express constructions for any claim term and instead applies the plain meaning of the claims. Ex. 1003

¶43. In forming my opinions, I have applied the plain and ordinary meaning of the claim terms as a POSITA would have understood them in the context of the '403 patent. In my opinion, no term needs to be construed to resolve the technical issues I address in this declaration, but I note that “interprocess communication service” should be interpreted consistently with my discussion in Section IV.A above.

VII. The Asserted References and Dr. Traynor’s Proposed Combination

A. TS-23.140

39. TS-23.140 is a 3GPP technical specification describing the Multimedia Messaging Service (“MMS”). Ex. 1004, 10; *see* Ex. 1003 ¶44. An MMS User Agent on user equipment exchanges multimedia messages (“MMs”) with an MMS Relay/Server over the MM1 interface. Ex. 1004, 14, 17–19, 23–24.

40. MMS may also be used to transport “Application Data,” which TS-23.140 defines as “[i]nformation / data specific to an application other than the MMS User Agent / VASP which is intended to be transported without alteration by using MMS.” *Id.* 14. When an abstract message carries a destination application identifier, “the MMS User Agent shall immediately route the received MMS information on to the destination application that is referred to from the destination application identifier (based on the negotiated details upon application registration process) without presentation to the user.” *Id.* 56; *see* Ex. 1003 ¶49.

41. TS-23.140 expressly does *not* describe how messages are routed from the MMS User Agent to any destination application:¹ “[d]etails of these applications or how an MMS User Agent or an MMS VAS Application would interface with them are outside the scope of this specification,” and “[t]he application registration process is outside the scope of this specification.” Ex. 1004, 54. Likewise, “[t]he further handling and processing of the information by the destination application is outside the scope of this specification.” *Id.* 56. Dr. Traynor acknowledges that TS-23.140 “leaves to a POSA the ‘[d]etails of ... how an MMS User Agent ... would interface with’ the user device’s destination applications.” Ex. 1003 ¶87.

B. Ogawa

42. Ogawa is titled “Removable Storage Device and Associated Methodology of Data Encryption.” Ex. 1005, cover. It describes a client site that receives data encrypted with a “shared encryption key” from a database server, decrypts it, expands it, and then re-encrypts it with an “inherent encryption key 53” before saving it to a local storage device 30. *Id.* 5:59–6:9.

43. Importantly, Ogawa uses its two keys in fundamentally different ways. The “shared” key is, as its name suggests, distributed between multiple

¹ All emphases added unless otherwise noted.

entities. Ogawa describes “processing operations that distribute the shared encryption keys” through a Diffie-Hellman key exchange, after which “the distribution of keys is carried out between the data input site 1 and the client site 3.” *Id.* 6:42–43, 7:1–24. Dr. Traynor agrees that “[t]he shared encryption key is distributed to both data input site 1 and client site 3.” Ex. 1003 ¶56. The inherent key is not distributed. It “is generated from an inherent identification code that was assigned during manufacturing and stored on the internal memory device of the removable storage 5.” Ex. 1005, 6:3–7. The removable storage may be “a USB memory device, flash drive, zip disk, floppy disk, compact disc, miniSD, or even DVD RAM.” *Id.* 5:49–51.

44. A POSITA would understand that, while the removable storage is connected, the program that accesses it may hold the inherent key (or the identification code from which the key is generated) temporarily in the client computer’s working memory. Ogawa describes the client site requesting the identification code 300 from the removable storage, which transmits the code back for use in encryption or decryption. Ex. 1005, 8:11–15, 8:26–31, FIG. 4 (steps S44–S46, S51–S53). Ogawa also claims an arrangement in which “the shared encryption key and the local encryption key” (i.e., the key generated from the removable storage’s identification code) “are unloaded from the client site when the removable storage is disconnected.” *Id.*, claim 15. Dr. Traynor similarly

testified that copies of keys stored on removable media might temporarily reside in a device's main memory. Ex. 2014, 30:1–16, 31:15–32:4. But in each instance, the key is used only by the program that has access to the removable storage device, and only while that device is connected. Nothing in Ogawa describes distributing the inherent key beyond the single program loaded from the removable storage device itself, or providing any other program with access to the inherent key.

45. In Ogawa's exemplary embodiment, all of the client-side units responsible for encryption and decryption using the inherent key—including encryption unit 35 and decryption unit 38—“are implemented as a single computer executable instruction set 51,” which Ogawa describes as a single “[l]oaded program (51)” that “is loaded to the client site 3 from a memory of the removable storage.” Ex. 1005, 5:24–32, 5:44–47, FIG. 1. When a user later wants to view the stored encrypted data, the same program's decryption unit 38 “will decrypt the inherent key encrypted data stored on the storage device 30, using the inherent encryption key 53 stored on the removable storage 5,” and the decrypted data are displayed. *Id.* 6:10–18; *see id.* 8:24–34. Dr. Traynor's own description of Ogawa likewise describes the inherent-key-encrypted data being “accessed again within the client device” by the same decryption unit 38. Ex. 1003 ¶94.

46. Ogawa explains that inherent-key-encrypted data cannot be decrypted without the removable storage that holds the inherent key, and that this is for

security purposes: “because the downloaded data is saved after being encrypted with an inherent encryption key, unless there is a corresponding removable storage 5 which holds the inherent encryption key, there is no way to decrypt the saved data. Therefore, if a laptop with downloaded data is stolen, and if the removable storage 5 is secured separately the data of the laptop remains secure.” Ex. 1005, 8:44–50; *see also id.* 10:58–63 (data “may only be decrypted only with the same PC and same removable storage”). When Ogawa contemplates others reading inherent-key data, it does not create an electronic copy of the key untethered from its removable storage device. Nor does it provide another program with access to the inherent key. Instead, it relies on the intended reader having physical access to the specific removable hardware that contains both the inherent key and the decryption program: a group member who needs to decrypt “must borrow a removable storage device.” *Id.* 10:41–49. Even the decryption program itself is loaded from the removable storage device. *Id.* 5:44–47. This ties the security of the scheme—including the decryption program—to physical possession of the removable storage device, such that only users who have access to that device can read the inherent-key-encrypted data. Dr. Traynor’s description of Ogawa does not address any of these teachings. *See* Ex. 1003 ¶¶51–57, 93–94.

47. Ogawa’s other variations follow the same principle. Ogawa describes requiring a pair of removable storage devices to be present for decryption (Ex.

1005, 10:64–11:5), and a removable storage device that holds both a “common key, common to a group,” and “an individual key, unique to the removable storage 5” (*id.* 11:24–31). In each variation, the ability to decrypt depends on physical possession of the appropriate removable storage device. None involves handing a key electronically to another program.

48. Ogawa also shows that when it passes data between separate programs or sites, it does not use its inherent key. Between sites, Ogawa uses its shared key, which is distributed to both ends through the key-exchange protocol described above. *Id.* 6:42–7:24. The only embodiment in which Ogawa expressly describes passing data from one program to another on the same client is its FIG. 9 embodiment, which adds a second layer of transport encryption (“TCP2”) on top of the FIG. 1 system. *Id.* 10:4–30. There, the database server’s TCP2 program encrypts the data “during the sending process of the sending unit 23” (*id.* 10:17–22)—the same sending unit that transmits the shared-key-encrypted data in FIG. 1 (*id.* 5:18–23). On the client, a TCP2 program 60 removes the TCP2 encryption and passes the data to program 51: “the received data are decrypted with the decryption unit 63 in TCP2 and are supplied in the sending unit 64 and are sent to the receiving unit 32 of the program 51.” *Id.* 10:23–28. Both programs are loaded from the same removable storage 5. *Id.* 10:9–13. Because “the parts other than TCP2 are the same as the configuration shown in FIG. 1” and “[t]he other processes are

completed similarly as shown in the system of FIG. 1” (*id.* 10:4–8, 10:28–30), a POSITA would understand that the data program 51 receives from program 60 is the same shared-key-encrypted data that, in FIG. 1, program 51’s receiving unit 32 receives and its decryption unit 33 decrypts with the shared encryption key 52. *Id.* 5:59–65. In other words, in the one instance where Ogawa hands data from one program to another on the same client, the receiving program already holds the key needed to read the data. Ogawa never applies its inherent key to data passed from one program to another.

C. Dr. Traynor’s Proposed Combination of TS-23.140 with Ogawa

49. Dr. Traynor’s Ground 1 combination implements Ogawa’s decryption and encryption units “as part of the MMS User Agent” in TS-23.140. Ex. 1003 ¶¶83–84, 101. His combination also adds a software bus between the User Agent and other applications on the device. *Id.* ¶¶87–92, 102.

50. For the “secure” requirement, Dr. Traynor proposes “to include the re-encryption functionality as part of MMS-Ogawa’s MMS User Agent, e.g., to secure decrypted received application data that was received over MM1 before transmitting it to any other component on the user device.” *Id.* ¶95. Thus, according to Dr. Traynor, “MMS-Ogawa’s interprocess communications are secured using Ogawa’s inherent key when data is routed by the MMS User Agent over TS-23.140’s software bus for interprocess communications ... to a destination

application.” *Id.* ¶150; *see id.* ¶¶102, 151. In other words, rather than Ogawa’s program 51 (which is loaded from the removable storage device) storing the encrypted data for later retrieval by the same program that has access to the inherent key needed to decrypt it, in Dr. Traynor’s combination that program would encrypt the data and send the encrypted data to a *different* program.

51. At deposition, Dr. Traynor summarized the operation of his proposed combination as follows: the MMS User Agent “would decrypt it first to determine the application ID and then re-encrypt it with the inherent key.” Ex. 2014, 42:11–14.

52. That inherent-key mechanism is Dr. Traynor’s only source of “secure” in every ground. For Grounds 2A–2C, he adds Cole’s modem and/or Sathish’s D-Bus but relies on “the bus secured as I discussed above in Section VII.C.6.” Ex. 1003 ¶¶216, 234–236. For Grounds 3A–4D, he adds Papineau and Ellison to MMS-Ogawa without changing that mechanism. *Id.* ¶¶237–282.

VIII. Dr. Traynor’s Reliance on IPR2024-00341 Does Not Address the Technical Issue Presented Here

53. Dr. Traynor repeatedly relies on the Final Written Decision in IPR2024-00341. *E.g.*, Ex. 1003 ¶¶58, 86, 92, 97, 122, 147–148, 152, 211. He states that “the Board considered a combination with these same modifications in a previous IPR, IPR2024-00341, for claims 1 and 27” of the ’733 patent (*id.* ¶58),

and that the Board “‘determine[d]’ that the arguments discussing Ogawa’s inherent keys and securing communications within MMS-Ogawa’s device ... were ‘fully supported by the record’” (*id.* ¶97). I am not offering any opinion on the legal effect of that decision. From a technical standpoint, however, it is my opinion that the technical issue I address in this declaration—whether a recipient application could read data that the MMS User Agent has encrypted with Ogawa’s inherent key—was not presented in that proceeding.

54. The ’733 claims addressed in IPR2024-00341 recited “a plurality of device agents communicatively coupled to the service control device link agent through an agent communication bus.” Ex. 1022, 4. The independent claims did not require that the bus be secure, and none of the claims recited a secure “interprocess communication service.” The issues addressed there concerned whether TS-23.140’s destination applications were separate from the MMS User Agent and whether a POSITA would have implemented a bus at all. *Id.* 18–32, 45–49.

55. The inherent-key theory on which Dr. Traynor now relies played no part in IPR2024-00341’s challenge to the independent claims. The petition there expressly invoked Ogawa’s inherent key only for dependent claim 27 of the ’733 patent. And, contrary to Dr. Traynor’s characterization of the prior combination as involving “these same modifications” (Ex. 1003 ¶58), it was a technically different

theory regarding how the inherent key would be used and where the inherent-key-encrypted data would be sent: under the IPR2024-00341 theory, Ogawa's own display units 37–39 would decrypt the inherent-key data. Ex. 1038, 69–74. Specifically, that petition argued that “[b]ecause both the MMS User Agent and MMS-Ogawa's display units 37-39 use the same key for encryption and decryption,” the inherent key was “shared by encryption unit 35 in the MMS User Agent (the service control device link agent, as claimed) and the destination display units 37-39 (the particular agent, as claimed).” *Id.* 74. Those display units are part of the same Ogawa program that encrypts the data in the first place. *See, e.g.,* Ex. 1005, 5:24–32, FIG. 1.

56. That earlier theory had a technical coherence that Dr. Traynor's present theory lacks. Because display units 37–39 are part of the same program 51 that holds the inherent key, decryption by the display units is exactly what Ogawa describes. Ex. 1005, 6:10–18. Under that theory, the inherent key never needed to cross a program boundary.

57. Dr. Traynor's present theory is entirely different: third-party downloaded applications (e.g., chess and weather applications), rather than display units 37–39 within the same program that encrypts the data, are the recipients of the encrypted data. Ex. 1003 ¶¶121–124, 145, 150–151, 158; Ex. 2014, 41:19–22. Dr. Traynor identifies no recipient even allegedly capable of decrypting data using

the inherent key, and his analysis of Ground 1 does not identify Ogawa’s display units 37–39—the receiving and decrypting entities that the IPR2024-00341 theory relied on—as recipients of the encrypted data. *See* Ex. 1003 ¶¶93–97, 146–159. And when asked at his deposition “which entity is doing the decryption in your proposed combination,” Dr. Traynor did not identify Ogawa’s display units. *See* Ex. 2014, 53:12–54:10. I did not see any mention of Ogawa’s display units anywhere in his deposition.

58. In my opinion, there is a sound technical reason why Dr. Traynor does not adopt the display-unit decryption theory. Claim 1 of the ’403 patent recites a “secure *interprocess* communication service,” whereas Ogawa implements its display units within the same program 51 that performs the original encryption. Ex. 1005, 5:24–32. A POSITA would understand that passing data among functional units of a single program—for example, through function calls within the same address space—is not interprocess communication. Moreover, the IPR2024-00341 petition framed its display-unit theory in terms of “data received by the MMS User Agent ... that was intended for *immediate display to the user*” (i.e., data rendered on the device’s display), and did not identify any separate downloaded application, such as a chess or weather application, as a recipient able to decrypt inherent-key data. Ex. 1038, 74.

59. In addition, the IPR2024-00341 claim 27 theory identified only a single agent capable of receiving and decrypting the data, consistent with claim 27's recitation of a single "particular agent" that must share the claimed "second encryption key." Claim 1 of the '403 patent, by contrast, requires the device messaging agent, "for *each message* in the subset of the secure Internet data messages," to forward application data "on the secure interprocess communication service," which Dr. Traynor maps to the subset of "abstract messages" intended for a plurality of destination applications. Ex. 1003 ¶¶135–137, 154–157. From a technical standpoint, securing deliveries to many different recipient applications raises key-distribution questions (for example, whether every application receives the same key or a different key, and how each application obtains its key) that a single-recipient, within-program theory never had to confront. Dr. Traynor did not say how his proposed combination would answer those questions, offering only that "I would imagine that different keys are used for different applications" and that the answer "all depends on what your security model is." Ex. 2014, 57:9–59:15. Accordingly, from a technical standpoint, the IPR2024-00341 theory, whatever its merits as to claim 27 of the '733 patent, says nothing about whether Dr. Traynor's different theory would work.

IX. Dr. Traynor Has Not Shown That Ground 1 Renders Any Challenged Claim Obvious

A. Dr. Traynor Does Not Explain How His Proposed Combination Would Allow Recipient Applications to Read the Securely Transmitted Data

1. The intended purpose of Dr. Traynor's combination is secure delivery of data that recipient applications can read.

60. In Dr. Traynor's combination, the transmission of encrypted data to destination applications has only one possible purpose: secure transmission of data that allows the recipient destination applications to actually *read* and *use* that data. *See* Ex. 1003 ¶¶95–96, 150–151, 157.

61. That intended purpose—secure transmission of data to entities that will be able to decipher it—follows from the basic function of encryption. Encryption transforms readable data (“plaintext”) into unintelligible data (“ciphertext”) that can be restored to plaintext only by someone holding the corresponding decryption key. Its purpose is to protect data on its way to a recipient who is meant to use it, while keeping it unintelligible to anyone else.

62. Data that no one can ever decrypt is meaningless. With the strong symmetric ciphers in common use by 2009 (for example, AES or Triple-DES), ciphertext is, for practical purposes, indistinguishable from random bits to anyone who does not hold the key. To a recipient without the key, encrypted application data is no more useful than if it had been deleted.

63. Accordingly, a transmission scheme in which the sender encrypts the data and nothing downstream can decrypt it is not useful, and in my opinion a POSITA would not have regarded such a scheme as an obvious way to deliver data to applications that need to use it.

64. Dr. Traynor acknowledges that his combination cannot serve its intended purpose if the destination application cannot ultimately obtain access to decrypted data: “At the end of the day, the application has to have access to decrypted data.” Ex. 2014, 42:20–25; *see also id.* 53:22–24 (“The destination application has to be able to get access to that data somehow, which implies decryption.”).

65. The destination applications on which Dr. Traynor relies confirm that those applications must be able to *read* the data transmitted to them. Dr. Traynor relies on a “chess application” that “initially need[s] to register with the appropriate MMS User Agent” in order to send and receive data (Ex. 1003 ¶121 (quoting Ex. 1004, 54–55)), on application data forwarded to “a specific process for moving the opponent’s chess pieces” (*id.* ¶158), and on MMS VAS Applications that “provid[e] Value Added Services (e.g. news service or weather forecasts) to MMS users” (*id.* ¶46 (quoting Ex. 1004, 14); *see also id.* ¶¶145, 156). Chess moves and weather forecasts are useful to these applications only if the applications can actually read them. Dr. Traynor agreed: his analysis rests on “my

assumption that the application at the end will get access to the data so that applications like chess or weather are useful,” with “the assumption being that that application can use it.” Ex. 2014, 53:5–8, 55:9–13; *see also id.* 43:4–11 (“I’m assuming for applications like chess, the destination application would need to see the contents of the message.”).

66. TS-23.140 and Dr. Traynor’s own descriptions of his combination point the same way. TS-23.140 defines “Application Data” as data “intended to be transported without alteration,” and it requires the MMS User Agent to route the received information on to the destination application. Ex. 1004, 14, 56. Dr. Traynor describes the User Agent as performing the steps needed “to ensure that the destination (on the user device) received data with correct format and content.” Ex. 1003 ¶96. And Dr. Traynor asserts that the inherent-key-encrypted application data is “transported without alteration” of the underlying data. *Id.* ¶157. From a technical standpoint, encrypted data is “without alteration” of the underlying data only in the sense that the original data can be recovered by decryption. To a recipient that lacks the key, the bytes it receives have been altered beyond use, and the data has not been received with “correct format and content” in any meaningful sense.

67. The intended purpose of Dr. Traynor's combination, in short, is to deliver application data securely to destination applications *that can read and use that data*.

68. Accordingly, any sensible implementation of Dr. Traynor's proposal requires that the data the User Agent encrypts before sending it over the bus can later be decrypted, so that the destination application can use it. That requirement comes from Dr. Traynor's reliance on encryption for security (and from the purpose of that encryption, which is allowing the recipient applications to securely make use of the transmitted data), not from the claims themselves.²

69. Ogawa itself reflects the understanding that any sensible encryption scheme must include a mechanism by which the intended recipient can decrypt the encrypted data. Every time Ogawa encrypts data with its inherent key, it also provides the program that will later decrypt that data. Ex. 1005, 6:7–18, 8:24–34.

² Claim 1 of the '403 patent does not itself require that any recipient decrypt data, or even that encryption be used. My opinions concern the specific mechanism Dr. Traynor chose. Dr. Traynor chose transmission of encrypted data as his sole source of security for the claimed interprocess communication service, and that choice makes technical sense only if the transmitted data can later be decrypted by or for the recipient. In my opinion, Dr. Traynor has not shown that any combination of TS-23.140 and Ogawa would have been obvious in which both (1) the data is encrypted with Ogawa's inherent key, which allows only a program tied to Ogawa's removable storage to encrypt or decrypt data, and (2) a separate recipient program (i.e., a program other than the encrypting program) has the capability to decrypt it.

The inherent key's design goal is to *preserve* decryptability for the holder of the removable storage while denying it to everyone else. *Id.* 8:44–50.

70. In my experience, a POSITA would have recognized that encryption by itself is not a difficult problem: standard, well-documented algorithms were readily available. The more complex problem is encrypting data in a way that allows the intended recipient to *decrypt* the data while simultaneously *preventing* unintended recipients from doing so. That is the key-management problem discussed in Section IV.A above. As one widely used cryptography text put it, “key management is the hardest part of cryptography.” Ex. 2017, ch. 8. Ogawa solves that problem for its own purposes by tying decryption to a physical removable storage device. But, as I explain below, Dr. Traynor offers no mechanism for ensuring secure decryptability in his proposed combination of TS-23.140 with Ogawa.

2. In Dr. Traynor's combination, only the MMS User Agent can decrypt the data it sends.

71. Dr. Traynor is explicit that Ogawa's encryption and decryption units are placed inside the MMS User Agent. Ex. 1003 ¶¶83–84, 95, 101–102. The User Agent decrypts the incoming message, reads the destination application identifier, re-encrypts the data with the inherent key, and sends it over the bus. Ex. 2014, 42:6–14. No other entity is given the inherent key. The network does not have it.

Id. 27:3–6. No destination application, operating-system service, or proxy is given the key anywhere in Dr. Traynor’s declaration. Ex. 1003 ¶¶93–97, 146–151; Ex. 2014, 27:7–29:11, 57:9–59:15. I have reviewed Dr. Traynor’s declaration and found no such disclosure.

72. Accordingly, the data the User Agent encrypts and sends can never be decrypted by or for its recipient. The encryption step that Dr. Traynor adds in an attempt to satisfy the “secure interprocess communication service” limitation thus defeats the very purpose for which he added it. In my opinion, Dr. Traynor’s combination is inoperable: a chess application will receive moves it cannot read, and a weather application will receive a forecast it cannot analyze. In practice, an application receiving ciphertext it cannot decrypt would fail to parse it and would, at best, discard it or report an error.

73. Dr. Traynor’s treatment of dependent claim 12 confirms that he understood that a recipient must be provided with the key in order to decrypt. For claim 12, which concerns messages sent from the User Agent to a network-side MMS VAS Application, Dr. Traynor explained that “the recipient MMS VAS Application ... needs to receive the shared encryption key that is used to encrypt the message to enable decryption of the message received from the MMS User Agent,” and he identified Ogawa’s Diffie-Hellman key exchange as the mechanism for getting the key to that recipient. Ex. 1003 ¶¶182–183. Thus, when Dr. Traynor

relied on Ogawa's *shared* key, he recognized that the recipient needs the key and identified how the recipient would obtain it. He offers no comparable mechanism for the *inherent* key he uses to secure the bus. Furthermore, Ogawa's key-exchange procedure is described only for the shared key, not for the inherent key, which Ogawa instead generates from the removable storage device's manufacturing code. Ex. 1005, 6:3–7, 6:42–7:24.

74. Dr. Traynor did not dispute that his declaration leaves this problem unaddressed. Asked to identify “any discussion in your declaration of another program having access to the inherent encryption key or decrypting the data,” he answered: “I haven’t talked about how precisely the decryption is performed, but I look to the ’403 patent and its spec for guidance.” Ex. 2014, 22:13–24. He could not identify any paragraph of his declaration describing any entity other than the User Agent having access to the inherent key. *Id.* 27:7–29:11. He could not say whether each destination application would receive the same inherent key or a different one, offering only that “I would imagine that different keys are used for different applications”—even though Ogawa ties each inherent key to a removable storage device and nowhere describes different inherent keys for different programs on the same device. *Id.* 57:9–58:1; Ex. 1005, 6:3–7. Asked for “one example as set forth in your declaration as to how the inherent keys in Ogawa are distributed to the destination application,” he identified none in his declaration,

offering only hypotheticals. Ex. 2014, 58:14–59:15. And when asked for a single paragraph number describing how the recipient application would be able to read the inherent-key-encrypted data, he answered: “Again, I’m not sure how to answer this further than I have.” *Id.* 61:2–11.

75. Instead, Dr. Traynor repeatedly characterized these questions as “implementation details” comparable to the device’s battery, screen size, and case color. *Id.* 23:14–24:1, 28:1–29:2, 47:20–48:2, 50:18–51:1, 59:9–15. I disagree. Battery size has nothing to do with whether Dr. Traynor’s proposed encryption scheme can deliver readable data to its recipients, which is the whole point of the “secure interprocess communication service” he alleges is present in his combination. In my experience designing secure network and messaging systems, the questions of which entity holds a decryption key, how that key gets there, and how the key is protected from entities that should not have it are core design decisions that determine both whether an encryption scheme works at all and whether it is secure. They are the substance of a security design, not physical dimensions or cosmetic details. Dr. Traynor himself acknowledged that how an encryption key is distributed could impact whether a system is secure, “[c]ertainly if it’s done poorly.” Ex. 2014, 59:16–24.

76. I also note that Dr. Traynor was not asked on redirect to explain how his combination would work. Ex. 2014, 61:22–23. In my opinion, Dr. Traynor’s

declaration does not set forth any coherent way in which his proposed combination could operate in the manner he alleges.

3. Neither TS-23.140 nor Ogawa teaches how a recipient program could decrypt inherent-key data.

77. Neither TS-23.140 nor Ogawa discloses how any program other than the program holding Ogawa's inherent key could decrypt data encrypted with that key, and Dr. Traynor does not identify any such disclosure. *See* Ex. 1003 ¶¶93–97, 146–151.

78. Dr. Traynor relies on TS-23.140's disclosure of encryption by the MMS User Agent. Ex. 1003 ¶¶50, 83–84. But TS-23.140 mentions encryption by the MMS User Agent only “on an end-user to end-user basis.” Ex. 1004, 19. A POSITA would understand “end-user to end-user” encryption in the MMS context to refer to encryption applied at the originating user's side and removed at the recipient user's side—that is, protection of an MM between two users across the network. TS-23.140 says nothing about encrypting or decrypting data passed from the User Agent to a destination application on the same device. It instead leaves the interface between the User Agent and those applications “outside the scope of this specification,” as Dr. Traynor acknowledges. *Id.* 54; Ex. 1003 ¶87; *see also* Ex. 1004, 56 (“further handling and processing of the information by the destination application is outside the scope of this specification”).

79. Ogawa likewise does not teach how inherent-key-encrypted data could be passed between two different programs on a device in a secure manner that would allow the receiving program to access the data.

80. As discussed above, Ogawa deliberately uses two different kinds of keys. Its shared key is distributed between sites through a Diffie-Hellman exchange. Ex. 1005, 6:42–43, 7:1–24. Dr. Traynor’s own declaration says so: “[t]he shared encryption key is distributed to both data input site 1 and client site 3.” Ex. 1003 ¶56.

81. Ogawa’s inherent key is fundamentally different. It “is generated from an inherent identification code that was assigned during manufacturing and stored on the internal memory device of the removable storage 5.” Ex. 1005, 6:3–7. The fact that the key is derived from an identification code inherent to the removable storage device is what makes the key “inherent.” Ogawa’s decryption process can perform decryption only by using the *same* inherent code. *Id.* 8:65–9:14 (decrypted plain text 900 is identical to the original plain text 100 only “if an inherent identification code 300 and 700 of the removal storage ... is the same”). The code passes only from the removable storage to the client program that uses it, and that client program is itself loaded from the removable storage. *Id.* 5:44–47, 8:10–19, 8:24–34. Dr. Traynor could identify no Ogawa embodiment in which the inherent

key is not stored on or derived from the removable storage. Ex. 2014, 29:18–32:16, 35:14–19 (“I don’t recall reading that”).

82. Dr. Traynor’s declaration describes the inherent key only as “a second encryption key that is different from the first ‘shared’ encryption key.” Ex. 1003 ¶¶57, 94. At deposition, he went further, suggesting that “[s]hared is another word for symmetric key,” and that Ogawa’s shared and inherent keys are simply “key A and key B.” Ex. 2014, 24:21–25:17. That is not how Ogawa uses those terms. Both keys are symmetric, but Ogawa distinguishes them by who has access to them: the shared key is exchanged between sites, while the inherent key is generated locally from the removable storage device’s identification code. Ex. 1005, 6:3–7, 6:42–7:24. Ogawa’s claims, which refer to the key generated from the identification code as the “local encryption key,” expressly require that it be “different from the shared encryption key.” *Id.*, claim 1. The distinction Ogawa draws is precisely about distribution.

83. Ogawa explicitly ties its inherent key to the removable storage device that is the focus of its invention. *Id.* 8:44–50 (“[U]nless there is a corresponding removable storage 5 which holds the inherent encryption key, there is no way to decrypt the saved data.”); *id.* 10:58–63 (data “may only be decrypted only with the same PC and same removable storage”). When another person wants to read inherent-key data, the key is not electronically transmitted to that person over a

bus; instead, that person “must borrow a removable storage device” that contains the key. *Id.* 10:41–49.

84. Even Ogawa’s group-sharing embodiment, in which multiple removable storage devices each contain “the same program (or encryption key),” shares the key only by duplicating the physical removable storage among the members of a group—“allow[ing] for encrypted data to be shared by members of a common group”—not by handing the key to other programs on the same device. *Id.* 5:54–58. A POSITA would understand that embodiment as a way to extend physical-possession-based access control to multiple authorized people, each of whom holds a removable storage device; it does not suggest that the key would be detached from the removable storage and provided to other software.

85. And in Ogawa’s exemplary embodiment, only one program—a program loaded from the removable storage device itself—uses the inherent key. Ogawa’s encryption unit 35 and decryption unit 38 are both part of “a single computer executable instruction set 51,” which is itself loaded from the removable storage device. *Id.* 5:24–32, 5:44–47, 6:1–18. Although Ogawa notes that these functions may alternatively be expressed as “object oriented modules, cooperating and/or independent daemons, separate utility applications,” it never describes providing the inherent key to any program other than one loaded from the removable storage device. *Id.* 5:32–37.

86. At deposition, Dr. Traynor also suggested that Ogawa's storage device 30 "could be just memory," such that another program might receive the data from it. Ex. 2014, 18:3–21:23. I disagree that this cures the deficiency. Ogawa describes storage device 30 as the place where data are "stored" by a "saving unit 36" and later retrieved for display (Ex. 1005, 5:59–6:18, 8:16–19, 8:24–34), and Ogawa's security rationale of keeping data secure on a stolen laptop presupposes persistent storage. *Id.* 8:44–56. In any event, *where* the encrypted data is placed does not address *who* can decrypt it. Whether the ciphertext sits on a disk or in memory, a program that does not hold the inherent key cannot read it.

4. Dr. Traynor's combination is inoperable for its intended purpose.

87. As discussed above, neither Ogawa nor TS-23.140 discloses or suggests any mechanism by which a destination application in Dr. Traynor's combination would be able to read the inherent-key-encrypted data transmitted from the MMS User Agent. And Dr. Traynor's declaration provides no explanation of how this capability would be present in his combination.

88. TS-23.140's transport of application data has the purpose of delivering data to the destination application "without alteration." Ex. 1004, 14, 56. Dr. Traynor modifies that transport so that the destination application receives data encrypted with a key that only the User Agent holds, with no articulated way to

restore the encrypted data to its unencrypted form. Ex. 1003 ¶¶95, 150–151. In my opinion, that modification renders TS-23.140’s application-data transport inoperable for its intended purpose, and the resulting combination would be, at a minimum, a seemingly inoperative device for delivering usable application data.

89. The ’403 patent itself does not present this problem, because when it discusses encrypting communications between agents it describes keys that are shared among, and provided to, the communicating agents—for example, “encryption using one or more keys that are partially shared or shared within the service processor 115 agent group,” and keys “provided and updated by the secure server.” Ex. 1001, 41:59–42:4, 42:19–28. That approach is incompatible with Dr. Traynor’s reliance on Ogawa’s *inherent* key, which is not distributed electronically to other programs and is designed to remain bound to the removable storage device. Ex. 1005, 6:3–7, 8:44–50, 10:41–49, 10:58–63.

90. My opinion on this point does not depend on any claim requirement. It depends only on the mechanism Dr. Traynor chose. Dr. Traynor chose the inherent key as his sole source of security for the alleged “secure interprocess communication service,” assigned that mechanism the purpose of protecting data delivered to destination applications, and never explained how the destination applications could read the data. Ex. 1003 ¶¶95, 150–151.

B. Dr. Traynor's Combination Conflicts with the Purpose of Ogawa's Inherent Key and Undermines His Own Rationale

91. Even setting aside the inoperability of the combination, it is my opinion that a POSITA would not have been motivated to make it, because Ogawa's teachings undermine the very reason Dr. Traynor offers for the modification.

92. Dr. Traynor's stated reason for applying the inherent key to bus communications is that "an implementation where internal communications were protected using an inherent key would have improved security of data in MMS-Ogawa's user device when stored or internally transmitted and would have helped, e.g., prevent unauthorized access to the data by rogue/unauthorized software on the device." Ex. 1003 ¶95. Ogawa provides no support for that rationale. Ogawa uses its inherent key for a single purpose: protecting files saved to a local disk so that they are useless if the computer is stolen without the separately secured removable storage. Ex. 1005, 8:44–56. Ogawa's own example is a salesperson who carries only the removable storage and downloads data to a customer's PC, where "there is no way of decrypting the data unless there is a corresponding removable storage 5." *Id.* 8:51–56.

93. Ogawa's rationale concerns data at rest on a machine that may be lost or stolen. It says nothing about messages passed between processes on a running

device. These are different threat models. Ogawa's threat is an attacker who obtains the computer but not the removable storage device; the inherent-key scheme defeats that attacker precisely because the key is *absent* from the computer. In interprocess communication on a running device, by contrast, the threat Dr. Traynor identifies is other software ("rogue/unauthorized software on the device," Ex. 1003 ¶¶95) running on the same device at the same time—the same device on which the key would have to be present for any recipient to decrypt. Once the removable storage is connected and the key is loaded, Ogawa's scheme by itself provides no separation among programs running on that device (other than the program loaded from the removable storage device itself). Ogawa does not describe any operating-system access control or other mechanism that would determine which other programs on a running device could use the key.

94. Dr. Traynor characterizes Ogawa's encryption unit 35 as re-encrypting data "before being transmitted elsewhere within the client device, e.g., for storage." Ex. 1003 ¶¶93; *see also id.* ¶¶57, 150. But the only example he identifies, and the only destination Ogawa describes for inherent-key-encrypted data, is storage device 30, from which the same program 51 later retrieves and decrypts the data. Ex. 1005, 6:7–18, 8:16–34. When Ogawa *does* pass data from one program to another on the same client, it does not use the inherent key. In FIG. 9, the TCP2 program 60 removes the TCP2 transport encryption and passes the

data to program 51, which already holds the shared key needed to decrypt the data as in FIG. 1. *Id.* 5:59–65, 10:23–30; *see* Section VII.B above. Ogawa thus teaches that its inherent key is for storage, not for inter-program transfer. Dr. Traynor identifies no disclosure in Ogawa, TS-23.140, or any other reference that applies a storage-bound key like Ogawa’s to communications between applications. *See* Ex. 1003 ¶¶93–97.

95. Dr. Traynor’s citation to Yami (Ex. 1039), which he describes as disclosing “secure communications between applications within a device using symmetric encryption keys” (Ex. 1003 ¶¶93, 95), does not cure this deficiency. I understand that Yami is not part of any ground, and Dr. Traynor does not rely on Yami for how a recipient application would obtain an inherent key like Ogawa’s in a way that would not undermine the security motivations behind the inherent key. *Id.* If anything, Yami confirms the deficiency in Dr. Traynor’s analysis. Yami’s inter-process scheme includes its own mechanism by which the *receiving* process obtains the key it needs: the generated symmetric key is itself encrypted with a static key and stored in association with a token, and “[w]hen a process receives the encrypted job data, the process retrieves the token,” retrieves the static key, “decrypt[s] the encrypted encryption key,” and then uses the decrypted key “to decrypt the job data.” Ex. 1039, Abstract; *see also id.* ¶[0023] (security library “suitably adapted to generate encryption keys and manage access thereto”). Dr.

Traynor does not propose to incorporate Yami's key-management mechanism into his combination, and doing so would replace, rather than use, Ogawa's removable-storage-bound inherent key.

96. To make his proposal work, Dr. Traynor would need to get the inherent key (or the ability to decrypt with it) to each destination application, and to do so in a way that would still satisfy each of the claim requirements, including that the application data be forwarded “on the secure interprocess communication service.” He never proposes that modification in his declaration, and in my opinion for good reason. Dr. Traynor's destination applications are third-party applications downloaded to the device. Ex. 1003 ¶¶121–123, 275–279; Ex. 2014, 41:19–22; Ex. 1004, 54. Dr. Traynor himself describes those applications as ones “that should not be trusted with access to sensitive device data/functions.” Ex. 1003 ¶272.

97. Distributing a single device-bound decryption key to every such application would place that key in the hands of the very “rogue/unauthorized software” that Dr. Traynor says the encryption guards against. Ex. 1003 ¶95. With a single symmetric key shared among all destination applications, any application holding the key could decrypt the messages intended for every other application—so a malicious game could read the data delivered to any other application on the device. That arrangement is contrary to the well-known principle of least privilege, under which “[e]very program and every user of the system should operate using

the least set of privileges necessary to complete the job.” Ex. 1032 (Saltzer), 1282. Dr. Traynor himself cites Saltzer as describing “process separation and controlled communication between processes.” Ex. 1003 ¶272. The security benefit that Dr. Traynor invokes as his motivation would disappear.

98. Nor would generating a different key for each application, as Dr. Traynor speculated at deposition (Ex. 2014, 57:21–58:1), be a use of Ogawa’s inherent key. Ogawa’s inherent key is a single key derived from a single manufacturing code of a single removable storage device. Ex. 1005, 6:3–7. Ogawa provides no mechanism for deriving multiple application-specific keys from that code or for distributing such keys to applications, and Dr. Traynor’s declaration proposes none.

99. Distributing the inherent key to applications would also abandon what makes Ogawa’s inherent key work. Ogawa ties decryption to possession of the corresponding removable storage device. Ex. 1005, 8:44–50, 10:58–63. A key handed out to every downloaded application on a handset is no longer bound to any physical guarantee of security. In my opinion, a POSITA reading Ogawa would not have been led to discard the defining feature of the very key Dr. Traynor selected; doing so would change the basic principle under which Ogawa’s inherent-key scheme was designed to operate.

C. A POSITA Would Not Have Had a Reasonable Expectation of Success

100. For the same reasons, it is my opinion that a POSITA would not have had a reasonable expectation of success in arriving at the claimed invention through Dr. Traynor's proposed combination. Dr. Traynor's opinion on this point as to the inherent-key bus consists of a single clause asserting that a POSITA "would have reasonably expected success doing so," followed by the same security rationale discussed above. Ex. 1003 ¶¶95. His only other expectation-of-success opinion concerning the inherent-key modification addresses the sending side alone: he states that "the MMS User Agent would continue to receive, decrypt, and send data within TS-23.140's user device," and that Ogawa's units "would continue to convert decrypted data into its original format before encrypting it for transmission within the user device." *Id.* ¶¶96. Neither opinion addresses how the destination application would read the data, which entity would hold the inherent key, or how the key would reach it. *Id.* ¶¶95–96; Ex. 2014, 22:21–24, 27:7–29:11, 58:14–61:11. Likewise, his expectation-of-success opinion for incorporating Ogawa's encryption and decryption units relies on Ogawa's *shared* key being used "to encrypt/decrypt data received from the MMS Relay/Server" (Ex. 1003 ¶85), which says nothing about inherent-key encryption on the bus.

101. Dr. Traynor's deposition testimony confirms that these omissions matter. He acknowledged that Ogawa provides only "some high level mechanisms for key generation, but there are lots of details that aren't included here," and that he would look to outside standards such as FIPS 140-2 to fill them in. Ex. 2014, 33:23–34:9, 35:4–8. He agreed that how an encryption key is distributed could impact whether a system is secure, "[c]ertainly if it's done poorly." *Id.* 59:16–24. And he conceded that "[t]here are many different ways to deliver data securely, and there are ways you could get it wrong too." *Id.* 57:1–8.

102. In my opinion, a POSITA could not reasonably have expected success from a scheme whose essential details are supplied by neither Dr. Traynor's declaration nor any asserted reference, particularly where Dr. Traynor acknowledges that those details matter.

103. Making a destination application able to decrypt inherent-key data on a 2009-era mobile device would have required a series of additional design decisions that appear nowhere in the record: where on the device the key would be stored; how it would be protected from other applications; how the User Agent would know which key to use for which application; how keys would be provisioned when an application is downloaded or registered; and how keys would be revoked when an application is removed. Each of these is a nontrivial engineering problem with security consequences. Dr. Traynor's suggestion that

key distribution “could be an independent key generation process as part of the registration process” (Ex. 2014, 58:20–25) itself confirms that a new mechanism would be needed beyond anything in Ogawa or TS-23.140 because Ogawa discloses no such process and TS-23.140 leaves the registration process “outside the scope of this specification.” Ex. 1004, 54.

D. Dr. Traynor also does not show the obviousness of a “secure interprocess communication service” in his proposed combination.

104. As I have explained above, an “interprocess communication service” is not simply communication functionality implemented by the communicating processes, but is instead a service that is provided to the processes by some other component (e.g., the operating system). Because Dr. Traynor does not specify how the encryption and decryption functionality between the MMS User Agent and destination application in his proposed combination would actually operate, he fails to show that any “secure” communication is provided as a service to the communicating processes as the claim requires. For example, he does not specify what entity is responsible for the decryption: if it is just the receiving destination application responsible for decrypting and the MMS User Agent responsible for encrypting, any security is provided only by the communicating processes themselves (rather than a “service” provided to those processes). The only security mechanism Dr. Traynor’s declaration actually describes is the MMS User Agent’s

own re-encryption of the data with Ogawa’s inherent key before sending it over the bus. Ex. 1003 ¶¶95, 150–151. That is security implemented by one of the communicating processes itself. The bus on which Dr. Traynor relies is described only as a channel “used to send information between applications” (Ex. 1031, 10:56–62; Ex. 1003 ¶¶89, 148), and Dr. Traynor does not contend that the bus, the operating system, or any other component provides encryption, decryption, or key management as a service to the MMS User Agent and the destination applications. *See* Ex. 1003 ¶¶146–151. And to the extent Dr. Traynor’s deposition theories contemplate the operating system or a “trusted proxy” performing the decryption, those theories appear nowhere in his declaration and, as discussed in Section IX.E below, he could not say whether they would satisfy the claim. Ex. 2014, 56:5–20.

E. I also disagree with Dr. Traynor’s new theories articulated for the first time at deposition

105. At deposition, Dr. Traynor offered alternatives: perhaps the destination application decrypts the data, or perhaps the operating system or some “trusted proxy” does. Ex. 2014, 19:10–14, 20:7–17, 22:5–12, 27:11–18, 43:16–22. I have reviewed Dr. Traynor’s declaration, and none of these theories appears in it. Ex. 1003 ¶¶93–97, 146–151. Nor could Dr. Traynor identify any disclosure in Ogawa of an operating system or proxy decrypting inherent-key data. Ex. 2014, 43:12–45:7. And he identified nothing in his declaration explaining how decrypted

data would travel from an operating system or proxy to the destination application.

Id. 45:8–47:15, 49:8–50:14, 56:21–57:8.

106. Dr. Traynor pointed to Ogawa’s FIG. 4 as showing that the key “is held in the client site in the operating system.” *Id.* 44:2–5; *but see id.* 44:11–18. I disagree with that reading. FIG. 4 shows the removable storage transmitting its identification value to the client site for use by program 51 in encrypting and decrypting data. Ex. 1005, 8:11–15, 8:26–31, FIG. 4. Nothing in FIG. 4 or its description refers to an operating system holding the key, much less using it on behalf of some other program.

107. Even if considered, the deposition theories do not set forth a coherent obviousness theory. Every one of them requires some entity other than the User Agent to hold Ogawa’s inherent key—precisely the modification Dr. Traynor never proposes in his declaration and that, as explained in Section IX.B above, would defeat Ogawa’s design and Dr. Traynor’s own rationale.

108. And Dr. Traynor himself would not say that these theories work. Asked whether a system in which the operating system decrypts the data and sends it on to the destination application would satisfy the secure interprocess communication service limitation, he answered: “It depends.” Ex. 2014, 56:5–11. He gave a similar answer for a trusted proxy: “it depends on the details of how that’s implemented. Potentially.” *Id.* 56:12–20. He rejected the proposition that any

operating-system delivery of decrypted data would qualify: “That’s not my testimony.” *Id.* 48:3–9. And when presented with an operating system that decrypts the data and then broadcasts it to every component on the device, he could not say whether the limitation would be met. *Id.* 54:15–55:16.

109. These theories also leave the final step—from the operating system or proxy to the application—unprotected. Once an intermediary decrypts the data and hands it to the application, the data travels between processes in plaintext. In that arrangement, the inherent-key encryption would protect only the hop from the User Agent to the operating system or proxy, while the hop to the destination application would be ordinary, unencrypted interprocess communication. Whether such an arrangement provides any meaningful protection against the “rogue/unauthorized software” Dr. Traynor identifies (Ex. 1003 ¶95) would depend entirely on details—how the intermediary authenticates the recipient, how it restricts delivery, and how the operating system isolates the plaintext—that Dr. Traynor’s declaration does not supply.

110. Finally, when asked where the missing details could be found, Dr. Traynor looked not to the prior art but to the patent under review: “I look to the ’403 patent and its spec for guidance. In column 42, there’s a lot of ways this could be implemented.” Ex. 2014, 22:21–23:7; *see also id.* 23:14–18 (“To interpret that, I look at column 41 and 42, which gives lots of examples of how that could be

implemented.”), 49:13–18, 50:10–14. Column 42 of the ’403 patent is where the specification describes providing bus encryption keys to the communicating agents. Ex. 1001, 42:19–31. In my opinion, the key-provisioning concepts Dr. Traynor invoked at deposition come from the ’403 patent’s disclosure, not from TS-23.140 or Ogawa. Neither reference describes a secure server (or any other entity) informing communicating processes of encryption keys, or any comparable mechanism for provisioning keys for interprocess communication.

X. Dr. Traynor’s Analysis of Grounds 2A–4D Suffers from the Same Deficiencies

111. Each of Dr. Traynor’s remaining grounds incorporates MMS-Ogawa’s inherent-key mechanism unchanged. For Grounds 2A–2C, he adds Cole’s WWAN and WLAN modems and Sathish’s D-Bus, but he relies on “the bus secured as I discussed above in Section VII.C.6”—that is, secured with Ogawa’s inherent key. Ex. 1003 ¶¶211–236; *see id.* ¶¶234–235. Sathish’s D-Bus is described only as “an example of a device inter-process communication channel used to send information between applications” (Ex. 1031, 10:56–62; Ex. 1003 ¶232); nothing in the cited portion of Sathish describes encrypting bus messages or distributing decryption keys to applications. For Grounds 3A–3D, he adds Papineau for claims 7–9, and for Grounds 4A–4D, he adds Ellison for claim 10 (and, in Grounds 4B–4D, claims 1–6 and 13–21), without altering how MMS-Ogawa secures

communications on the device. Ex. 1003 ¶¶237–282; *see id.* ¶281 (“Ellison’s teachings do not affect how and why a POSA would have combined those references for those claims”). Dr. Traynor does not cite any of Cole, Sathish, Papineau, or Ellison for a recipient application’s ability to decrypt inherent-key data. My opinions in Section IX therefore apply equally to each of these grounds.

112. The Ellison grounds make the problem more acute. Dr. Traynor’s rationale for adding Ellison is to place the MMS User Agent in an isolated, secure execution environment while “un-trusted third-party device application[s]” run outside that environment, with the User Agent having “authorized communications” with those applications over the “secure interprocess communication service.” Ex. 1003 ¶¶271–280. Providing Ogawa’s inherent key to those untrusted applications—which would be necessary for them to read the data—would move the key outside the very secure environment Ellison is invoked to provide.

XI. Dr. Traynor Has Not Shown the Dependent Claims to Be Obvious for Additional Reasons

113. Because, in my opinion, Dr. Traynor has not shown claim 1 to be obvious under any ground, the same is true of each dependent claim. In addition, Dr. Traynor’s analysis of the following dependent claims is deficient for further, independent technical reasons.

A. Claim 11

114. Claim 11 requires that the device messaging agent send secure upload Internet data messages, “at least a subset” of which “are sent responsive to a corresponding request received on the secure interprocess communication channel from a corresponding one of the software applications,” and that each such message contain “an identifier for a respective network application server corresponding to the requesting software application.” Ex. 1001, claim 11.

115. First, Dr. Traynor never shows that any request is received on a *secure* channel. His combination secures only one direction: the User Agent re-encrypts data before sending it *to* a destination application. Ex. 1003 ¶¶95, 150–151. For the reverse direction, Dr. Traynor says only that “communication between MMS-Ogawa’s MMS User Agent application and other applications on the user device occur over MMS-Ogawa’s bus (‘secure interprocess communication service’).” *Id.* ¶175. He never explains how a request *from* an application would be secured. Under Dr. Traynor’s inherent-key theory, the requesting application would itself need Ogawa’s inherent key and an encryption unit, neither of which he contends would be included in any application. *See* Sections IX.A.2–3 above. In Ogawa, only program 51 contains an encryption unit and has access to the inherent key (Ex. 1005, 5:24–32, 6:1–9), and Dr. Traynor places those components only in the User Agent (Ex. 1003 ¶¶83–84, 95, 101). An application that has neither the key

nor an encryption unit cannot produce inherent-key-encrypted requests. And providing each requesting application with the inherent key and an encryption unit would contradict Ogawa's design, which provides inherent-key encryption and decryption capability only to the single program loaded from the removable storage device. Ex. 1005, 5:24–32, 5:44–48, 8:44–50.

116. Second, Dr. Traynor does not show the claimed identifier. He relies on TS-23.140's "destination application identifier." Ex. 1003 ¶¶177–179. As its name indicates, that identifier identifies the destination *application* to which a VAS Application routes the data. Ex. 1004, 55–56. Dr. Traynor himself describes the identifier as being "for use by the MMS VAS Application to route the message to the correct destination application." Ex. 1003 ¶179. TS-23.140 does not describe the identifier being used by the MMS User Agent (or MMS Relay/Server) to route the message to a VAS Application, and Dr. Traynor identifies no such disclosure. In my opinion, an identifier of an application is not "an identifier for a respective network application server."

117. TS-23.140 itself distinguishes between these two kinds of identification. It explains that applications residing on an MMS VAS Application "have an additional level of addressing – in addition to the identification of the VASP and the MMS VAS Application." Ex. 1004, 54–55. In other words, TS-23.140 treats the application identifier as a separate addressing level from the

identification of the VASP and VAS Application (i.e., the network-side server). That application identifiers may include “the application developer’s URL” in order to “guarantee their global uniqueness” (*id.* 55) does not change this; that is a naming convention for making application identifiers unique, and Dr. Traynor does not rely on it. Dr. Traynor never explains how TS-23.140 would be modified to arrive at an identifier for a respective network application server. *See* Ex. 1003 ¶¶176–179.

118. Because claim 12 depends from claim 11, and Dr. Traynor’s analysis of claim 12 relies on his analysis of claim 11 (Ex. 1003 ¶¶181–184), my opinions regarding claim 11 apply equally to claim 12.

B. Claim 15

119. Claim 15 requires that “access by the software applications to the interprocess communication channel is subject to a security policy.” Ex. 1001, claim 15. Dr. Traynor contends that access by the applications is “subject to a security policy”—“one that requires a specific encryption key”—because the bus is encrypted. Ex. 1003 ¶¶190–191.

120. But in Dr. Traynor’s combination, no software application holds that key. *See* Section IX.A.2 above. A key that no application can obtain does not govern applications’ “access” to anything; it simply prevents every application from reading what it receives. From a technical standpoint, a security policy

governing access distinguishes between entities that are permitted access and entities that are not, as in the '403 patent's description of agents that "can only be accessed by one another as necessary or permitted by agent communication policies, or by the service controller or other authorized network function with proper security credentials," and of keys provided by a secure server to authorized agents. Ex. 1001, 42:19–35, 88:9–22. A uniform inability of every application to read the data is not such a policy.

121. I also note that Dr. Traynor's analysis of claim 15 describes MMS-Ogawa's interprocess communication service as "secured using shared key encryption." Ex. 1003 ¶191. That is inconsistent with the mechanism he actually proposes, which uses Ogawa's *inherent* key—a key that, as discussed in Section IX.A.2 above, Ogawa does not share with any other program. *Id.* ¶¶94–95, 150; Ex. 1005, 6:3–7.

C. Claim 16

122. Claim 16 requires that "at least one of the secure Internet data messages comprises multiple identifier/data pairs." Ex. 1001, claim 16. Dr. Traynor offers two theories. Ex. 1003 ¶¶192–198. In my opinion, neither succeeds.

123. Dr. Traynor first contends that "[w]hen an unavailable recipient MMS User Agent becomes available, e.g., by moving into coverage, the MMS Relay/Server delivers the group of collected messages to the MMS User Agent,"

and that this group is a single message comprising multiple identifier/data pairs.

Ex. 1003 ¶¶193–194. A group of the claimed messages, however, is not *a* claimed

message. TS-23.140 specifies how a User Agent retrieves an MM: “[t]he recipient

MMS User Agent shall issue an MM1_retrieve.REQ to the recipient MMS

Relay/Server to initiate the retrieval process,” providing “a reference, e.g., URI, for

the MM,” and “[t]he MMS Relay/Server shall respond with an

MM1_retrieve.RES, which contains MMs control information and the MM

content.” Ex. 1004, 69. Each request and response concerns a single MM.

Consistent with this, the information elements of the MM1_retrieve.RES include a

single “Message ID,” described as “[t]he message ID of the MM.” *Id.* 72 (Table

12).

124. The section Dr. Traynor cites for the “group” delivery, TS-23.140 § 7.1.3, likewise describes retrieval of “an MM” “based on the Message Reference received in a notification,” and states that upon a retrieval request the Relay/Server “shall deliver the MM to the recipient MMS User Agent.” Ex. 1004, 29; *see* Ex. 1003 ¶194. Although § 7.1.3 states that the Relay/Server stores messages until the recipient User Agent becomes reachable (Ex. 1004, 29), I see nothing in that section, or elsewhere in TS-23.140, describing multiple MMs being combined into a single message to the User Agent.

125. Dr. Traynor identified the MM1_retrieve.RES as his example of a secure Internet data message containing an identifier. Ex. 1003 ¶136; Ex. 2014, 8:16–9:7. Yet when asked whether any provision of § 8.1 of TS-23.140 describes an MM1_retrieve.RES carrying more than one MM, he answered, “I don’t have a direct reference,” offering only his understanding of what is done in practice, and he did not identify where his declaration addresses the question. Ex. 2014, 12:24–14:12.

126. Dr. Traynor alternatively contends that “multiple higher-layer data units, such as multiple HTTP messages each carrying a distinct MM, would be aggregated within a single lower-layer frame,” relying on the RLC and MAC layers of the 3G radio interface. Ex. 1003 ¶¶195–198. But the claimed “secure Internet data messages” are the messages the device messaging agent receives “over a secure connection to a network message server.” Ex. 1001, claim 1. Dr. Traynor’s device messaging agent is the MMS User Agent, which he says “sits at the application layer,” well above the radio link protocols. Ex. 1003 ¶208.

127. The MMS User Agent never receives an RLC protocol data unit or a MAC transport block. In the layered architecture Dr. Traynor describes, data received over the radio interface passes up through the MAC, RLC, and PDCP layers, then the IP and transport layers, before the reassembled application-layer message is delivered to the MMS User Agent. *See* Ex. 1003 ¶196; Ex. 1053, 8–9.

At the RLC layer, received protocol data units are “reassembled” into service data units and “delivered to upper layers,” with the RLC header removed. Ex. 1052, 15, 17. The lower-layer framing exists only within those lower layers and is invisible to the application-layer User Agent.

128. Nor does concatenating or multiplexing segments of lower-layer data units create “identifier/data pairs” within a secure Internet data message. The RLC “Length Indicators” that accompany concatenation mark the boundaries between RLC service data units (Ex. 1052, 14, 16, 33), and the MAC header’s C/T field “provides identification of the logical channel instance,” while its UE-Id field identifies the user equipment (Ex. 1051, 35). None of these fields identifies a software application or pairs an application identifier with application data. Moreover, in Dr. Traynor’s own combination the MM1 connection is “secured both with SSL/TLS and encrypted using MMS-Ogawa Message Encryption” (Ex. 1003 ¶100), so the HTTP messages carrying MMs are encrypted before they ever reach the radio layers. The RLC and MAC layers therefore operate on opaque ciphertext and have no visibility into any destination application identifier contained in an MM.

129. Dr. Traynor could not say at which layer the MM1_retrieve.RES would reside, and he did not recall offering any opinion on the subject. Ex. 2014, 9:8–10:7. In my opinion, Dr. Traynor’s lower-layer theory does not show that any

secure Internet data message received by the device messaging agent comprises multiple identifier/data pairs.

D. Claim 21

130. Claim 21 requires that “at least one of the applications and the network application server corresponding to that application authenticate with each other prior to passing application data via the device messaging agent and network message server.” Ex. 1001, claim 21. Dr. Traynor relies on TS-23.140’s application registration process, opining that registration constitutes authentication because, without it, the applications would not be “taking steps to establish that they are respectively permitted to use MMS to exchange data.” Ex. 1003 ¶¶209–210.

131. Registration, as TS-23.140 describes it, is not mutual authentication. TS-23.140 describes registration in a single note: the application “provisions an MMS User Agent or an MMS VAS Application with its application identification value and may negotiate with the MMS User Agent or MMS VAS Application the details (amount and format) of information to be exchanged between the two entities.” Ex. 1004, 54. “The application registration process is outside the scope of this specification,” and it “may be an inherent process e.g., in the application’s integration into a mobile phone.” *Id.* When asked whether the note describes anything beyond provisioning an application identification value and negotiating

the amount and format of information, Dr. Traynor pointed only to the note's final sentence and identified no credential, secret, or verification step. Ex. 2014, 36:11–37:22, 40:4–41:2.

132. Providing an identifier and agreeing on a data format do not verify anyone's identity. In the context of secure computer communications, "authentication means verifying the identity of the communicating principals to one another." Ex. 1030 (Needham), 993. Authentication typically requires a party to prove its identity, for example by demonstrating knowledge or possession of a secret (such as a password or cryptographic key) or by presenting a credential that the other party can verify. Absent implementation details (which TS-23.140 and Dr. Traynor do not provide), presentation of an identifier does not authenticate identity; rather, it merely provides a claim of identity that any application could duplicate. And mutual authentication (parties that "authenticate with each other") requires that each party verify the identity of the other. Nothing in TS-23.140's registration note describes a credential, a secret, a challenge, or any verification step by either party. Dr. Traynor's characterization of registration as establishing that applications are "permitted to use MMS" (Ex. 1003 ¶210) describes, at most, authorization (i.e., a determination of what an entity may do) rather than authentication, which verifies who the entity is or what authenticated credentials they have.

133. Dr. Traynor also does not show that a destination application residing on an end-user device would *ever* register with an MMS VAS Application. TS-23.140 states that an application registers “with the appropriate MMS User Agent or MMS VAS Application.” Ex. 1004, 54. A POSITA would have understood the “appropriate” MMS User Agent or MMS VAS Application to be the one on which the application resides (which for a device application is the MMS User Agent on the same device) because that is the entity with which the application interacts directly, and that entity in turn handles any coordination with remote MMS entities. TS-23.140 itself describes applications in exactly these terms: “[a]pplications that intend to transport application specific data using MMS may either reside on an MMS User Agent or on an MMS VAS Application,” and applications residing on an MMS VAS Application “may trigger the MMS VAS Application as a MMS front-end.” *Id.* 54–55. Likewise, “an application may trigger an MMS User Agent or an MMS VAS Application to submit certain abstract messages” “[b]ased on the negotiated details upon application registration process.” *Id.* 55. In each case, the application deals with its own local MMS entity, which acts as its front-end to the MMS network.

134. Dr. Traynor testified that the note contemplates “the possibility of both,” but that a device application “might do so locally with its MMS user agent.” Ex. 2014, 38:22–39:3. Although he speculated that the User Agent “may then need

to, as a proxy or on behalf of application, further register with the MMS VAS application” (*id.* 39:3–5), TS-23.140 describes no such step, and Dr. Traynor’s declaration does not propose one. *See* Ex. 1003 ¶¶209–210. In any event, registration by the User Agent on the application’s behalf would not be the application itself authenticating with its network application server. Registration with the device’s own User Agent is not the application and the network application server “authentica[t]ing with each other.”

135. Finally, the MM7 authentication Dr. Traynor cites occurs between the MMS Relay/Server and a VASP, not between a device application and its network application server. Ex. 1004, 41; Ex. 1003 ¶210. TS-23.140 explains that “MM7 should use transport layer security mechanisms to authenticate the VASP,” for example by having “the MMS Relay/Server or the VASP ... authenticate the VASP during each session established for message submission,” and that “[t]he VASP may authenticate the MMS Relay/Server using similar mechanisms.” Ex. 1004, 41. The device-side application is not a party to that exchange, and Dr. Traynor does not explain how authentication between a VASP and the MMS Relay/Server could constitute an application and its network application server authenticating with each other. In my opinion, Dr. Traynor has not shown that TS-23.140 discloses or suggests the mutual authentication claim 21 requires.

XII. Conclusion

136. For the reasons explained above, it is my opinion that Dr. Traynor has not shown that any of claims 1–21 of the '403 patent would have been obvious under any ground he addresses.

137. I reserve the right to supplement or amend my opinions in response to any additional information, arguments, or testimony that Petitioner or Dr. Traynor may present, including any reply declaration.