

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner,

v.

SECURITY FIRST INNOVATIONS, LLC
Patent Owner.

Case No. IPR2024-00213
Patent No. 9,338,140

DECLARATION OF SAMRAT BHATTACHARJEE, PH.D.

TABLE OF CONTENTS

I.	Personal and Professional Background	1
II.	Materials Reviewed and Considered	4
III.	My Understanding of Patent Law	6
	A. Anticipation	8
	B. Obviousness	9
IV.	UNPATENTABILITY GROUNDS	12
V.	THE '140 PATENT	13
	A. Specification	13
	B. Person of Ordinary Skill in the Art (“POSA”)	16
	C. Prosecution History	19
	D. Challenged Claims	21
VI.	CLAIM INTERPRETATION	22
VII.	GROUND 1: OPHIR AND TAKEDA RENDER CLAIMS 1-6, 8, 10 OBVIOUS	25
	A. Ophir (EX1007)	25
	B. Takeda (EX1008)	32
	C. Ophir-Takeda Combination	43
	1. Reasons to Combine	43
	a. Increased Reliability	44
	b. Simple Substitution	45
	c. Use of Known Technique to Improve Similar Devices in the Same Way	50
	2. Reasonable Expectation of Success	51
	3. Resulting Combination	52
	D. Ophir-Takeda Meets All Limitations of Claims 1-6, 8, 10	54
	1. Claim 1	54
	a. 1[PRE]: A secure storage network comprising:	54
	b. [1A]:	56

i.	a plurality of physical storage devices.....	56
ii.	storing thereon a plurality of shares, the plurality of shares being associated with at least one session key used to secure a dataset.....	59
c.	[1B]: a secure storage system configured to:.....	63
i.	[1B-1]: present to a client device a virtual disk, the virtual disk comprising a directory mapped to the plurality of physical storage devices such that physical locations of the shares are hidden from the client device;	64
	(1) Block Services	67
	(2) File Services	74
ii.	[1B-2]: generate the plurality of shares for storage on the plurality of physical storage devices by performing a securing operation on the dataset received from the client device and distributing the dataset in the shares;	77
iii.	[1B-3]: include with each of the plurality of shares data indicative of the at least one session key used to secure the dataset; and	79
iv.	[1B-4]: reconstitute the dataset from at least a portion of the plurality of shares stored on the physical storage devices in response to a request from the client device for information in the dataset.	81
2.	Claim 2: The secure storage network of claim 1, wherein the secure storage system is further configured to generate the plurality of shares by performing a cryptographic operation on the dataset.	83
3.	Claim 3: The secure storage network of claim 1, further comprising a key management server configured to store workgroup keys.	83
4.	Claim 4: The secure storage network of claim 1, wherein the secure storage system is further configured to provide access to the reconstituted dataset.	85

5. Claim 5: The secure storage network of claim 1, wherein the secure storage system is further configured to establish a secure connection between the client device and the secure storage system.	87
6. Claim 6: The secure storage network of claim 1, wherein the shares contain a substantially random distribution of the received dataset.	88
7. Claim 8: The secure storage network of claim 1, wherein the secure storage system is configured to secure the dataset by encrypting the dataset using the at least one session key.....	90
8. Claim 10: The secure storage network of claim 1, wherein the data indicative of the at least one session key comprises: a portion of the at least one session key or data indicative of a portion of the at least one session key.....	90
VIII. GROUND II: OPHIR, TAKEDA, AND BIRRELL RENDER CLAIM 7 OBVIOUS.....	91
A. Birrell (EX1033).....	91
B. Ophir-Takeda-Birrell Combination.....	92
C. Claim 7: The secure storage network of claim 1, wherein the secure storage system is configured to reconstitute the received dataset from fewer than all of the shares	93
IX. GROUND III: OPHIR, TAKEDA, AND DICKINSON RENDER CLAIM 9 OBVIOUS.....	93
A. Dickinson (EX1005).....	93
B. Ophir-Takeda-Dickinson Combination.....	94
C. Claim 9: The secure storage network of claim 1, wherein the secure storage system is configured to secure the dataset by splitting the dataset using the at least one session key	95
X. GROUND IV: DICKINSON AND TAKEDA RENDER CLAIMS 1-10 OBVIOUS.....	96
A. Dickinson (EX1005).....	97
B. Takeda (EX1008)	102
C. Dickinson-Takeda Combination.....	102
1. Reasons to Modify Dickinson.....	104

2. Reasonable Expectation of Success	107
3. Resulting Combination.....	108
D. Dickinson-Takeda Meets All Limitations of Claims 1-10.....	112
1. Claim 1	112
a. 1[PRE]: A secure storage network comprising.....	112
b. [1A]: a plurality of physical storage devices storing thereon a plurality of shares, the plurality of shares being associated with at least one session key used to secure a dataset.....	113
c. [1B]: a secure storage system configured to:.....	116
i. [1B-1]: present to a client device a virtual disk, the virtual disk comprising a directory mapped to the plurality of physical storage devices such that physical locations of the shares are hidden from the client device;	117
ii. [1B-2]: generate the plurality of shares for storage on the plurality of physical storage devices by performing a securing operation on the dataset received from the client device and distributing the dataset in the shares;	120
iii. [1B-3]: include with each of the plurality of shares data indicative of the at least one session key used to secure the dataset; and	121
iv. [1B-4]: reconstitute the dataset from at least a portion of the plurality of shares stored on the physical storage devices in response to a request from the client device for information in the dataset.	123
2. Claim 2: The secure storage network of claim 1, wherein the secure storage system is further configured to generate the plurality of shares by performing a cryptographic operation on the dataset.	123
3. Claim 3: The secure storage network of claim 1, further comprising a key management server configured to store workgroup keys.	124

4. Claim 4: The secure storage network of claim 1, wherein the secure storage system is further configured to provide access to the reconstituted dataset.	125
5. Claim 5: The secure storage network of claim 1, wherein the secure storage system is further configured to establish a secure connection between the client device and the secure storage system.	125
6. Claim 6: The secure storage network of claim 1, wherein the shares contain a substantially random distribution of the received dataset.	126
7. Claim 7: The secure storage network of claim 1, wherein the secure storage system is configured to reconstitute the received dataset from fewer than all of the shares.....	126
8. Claim 8: The secure storage network of claim 1, wherein the secure storage system is configured to secure the dataset by encrypting the dataset using the at least one session key.....	127
9. Claim 9: The secure storage network of claim 1, wherein the secure storage system is configured to secure the dataset by splitting the dataset using the at least one session key.....	127
10. Claim 10: The secure storage network of claim 1, wherein the data indicative of the at least one session key comprises: a portion of the at least one session key or data indicative of a portion of the at least one session key.....	128
XI. APPENDIX: CLAIM LISTING	130

APPENDIX LISTING OF EXHIBITS

Exhibit	Description
1001	U.S. Patent No. 9,338,140 (“140 patent”)
1002	Prosecution History of U.S. Patent No. 9,338,140
1003	Declaration of Samrat Bhattacharjee, Ph.D. (“Bhattacharjee”)
1004	Curriculum Vitae of Samrat Bhattacharjee, Ph.D.
1005	International Application Publication WO 2001/022322 (“Dickinson”)
1006	RESERVED
1007	U.S. Patent Application Publication 2004/0143733 (“Ophir”)
1008	U.S. Patent Application Publication 2001/0053221 (“Takeda”)
1009	U.S. Patent No. 7,187,771
1010	U.S. Patent No. 7,571,320
1011	European Patent Application Publication EP 1 063 811
1012	U.S. Patent No. 6,151,659
1013	U.S. Patent No. 5,526,507
1014	U.S. Patent No. 6,065,096
1015	U.S. Patent No. 7,941,695
1016	U.S. Patent No. 8,352,518
1017	U.S. Patent Application Publication 2002/0174295
1018	U.S. Patent Application Publication 2002/0019908
1019	U.S. Patent No. 7,043,614
1020	U.S. Patent No. 6,973,455
1021	U.S. Patent No. 6,697,846
1022	U.S. Patent No. 6,324,648
1023	U.S. Patent Application Publication 2002/0019932
1024	U.S. Patent No. 5,825,890
1025	RESERVED
1026	RESERVED
1027	U.S. Patent Application Publication 2003/0028493
1028	U.S. Patent Application Publication 2002/0042859
1029	U.S. Patent Application Publication 2005/0216754
1030	U.S. Patent Application Publication 2003/0191938
1031	U.S. Patent Application Publication 2003/0167408
1032	RESERVED
1033	U.S. Patent No. 7,529,834 (“Birrell”)
1034	U.S. Patent No. 5,483,598
1035	RESERVED
1036	RESERVED

Exhibit	Description
1037	Merriam Webster's Collegiate Dictionary, 207, 381, 888 (10th ed. 1997)
1038	U.S. Patent No. 6,356,915
1039	U.S. Patent No. 4,067,058
1040	Redline Comparison of International Application Publication WO 2001/022322 to U.S. Patent No. 9,338,140
1041	Google LLC's Proposed Claim Terms for Construction in <i>Security First Innovations, LLC v. Google, LLC</i> , 2:23-cv-00097-JWK-RJK (Nov. 11, 2023)
1042	Letter from A. Mayron to D. Perlson re Preliminary Proposed Constructions in <i>Security First Innovations, LLC v. Google, LLC</i> , 2:23-cv-00097-JWK-RJK (Nov. 11, 2023)
1043	Microsoft Computer Dictionary, 99, 192, 362, 407 (5th ed. 2002)
1044	Security First Innovation's 2023-08-14 Infringement Contentions, Chart D in <i>Security First Innovations, LLC v. Google LLC</i> , 2-23-cv-00097 (E.D. Va.)
1045	Rule 16(b) Scheduling Order in <i>Security First Innovations, LLC v. Google, LLC</i> , 2:23-cv-00097-JWK-RJK, Dkt. 54 (July 20, 2023)

I, Samrat Bhattacharjee, Ph.D., declare:

1. I have been retained by Wolf Greenfield & Sacks, P.C., counsel for Petitioner Google LLC, to assess claims 1-10 (the “challenged claims”) of U.S. Patent No. 9,338,140 (“the ’140 patent”) (EX1001). I am being compensated for my time at my standard rate of \$700.00 per hour, plus actual expenses. My compensation is not dependent in any way upon the outcome of the *inter partes* review of the ’140 patent.

I. PERSONAL AND PROFESSIONAL BACKGROUND

2. My qualifications for forming the opinions set forth in this declaration are summarized below and explained in more detail in my current curriculum vitae, provided as EX1004. EX1004 also includes a list of my publications and patents.

3. I am a tenured professor in the Computer Science Department at the University of Maryland. I am also an Affiliate Professor with the Department of

4. Electrical and Computer Engineering at the University of Maryland. I have been an appointed faculty member at the University of Maryland since 1999.

5. I received a Ph.D. in computer science from the Georgia Institute of Technology in 1999. Before that I received a B.S. degree in Mathematics and Computer Science with Highest Distinction (Summa Cum Laude) and was recognized as the Outstanding Department Major at Georgia College and State

University, Milledgeville, Georgia in 1994.

6. I have been involved in the research and development of computing systems for over two decades. I perform research and I teach in the areas of computer networks, distributed systems, computer security, cryptography, and operating systems, among others. A major theme of my research has been the development of technology to improve the performance and security of data transfers and data storage on networks like the Internet. My research has examined problems ranging from network and operating system support audio and video streaming, large scale Internet video distribution, the design of novel network architectures for better and secure content distribution and storage, efficiency of content distribution over wireless networks, and so on.

7. For example, starting in the late 1990s, the focus of my research was the development of network and operating system technology to design a new network architecture that allowed for more efficient data transfer. Part of my Ph.D. research was to develop new and more efficient architectures for video delivery on the Internet, and I have published papers on this architecture during my graduate studies. I continued to work on video delivery and security and storage of digital content as a faculty member, and have published various papers in these areas. During 2007, I was a visiting researcher at AT&T Labs, and one of the projects I focused on was a video content delivery platform. This work resulted in both

publications and a granted U.S. patent (U.S. 8,752,100 B2). Among other research, I have authored papers and developed research software for public-key infrastructure KeyChains and a cryptographic library for “chit”-based security.

8. Along with working on research problems related to computer networking, computer security, cryptography, and video distribution, I routinely taught this material in both graduate and undergraduate courses. A number of my papers in these domains are with students at the University of Maryland, many of whom started in these areas after taking my classes.

9. For over 25 years, I have developed professional and academic experience in the field of cryptographic techniques for computer systems. For example, I have authored or co-authored over 100 articles in peer-reviewed journals, conference proceedings, and workshops relating to computer networking, computer system, and data security. Over the years I have received several fellowships, prizes, and awards for teaching excellence and technical papers, including an Alfred P. Sloan Jr. Fellowship in 2004 and the National Science Foundation CAREER award in 2001. I have also received an award from the SIGCOMM foundation for a “Test of Time” paper (that is given to an influential paper published ten years ago). My research work has been supported by multiple grants from the US National Science Foundation, and the Department of Defense. I have also started a Joint Ph.D. program with the University of Maryland and the

Max Planck Society in Germany, and co-founded the annual Cornell, Maryland, Max Planck Research School that provides research exposure to about 80 students from across the world during a week-long school.

10. I have served on numerous paper and proposal review panels and participated on program committees for ACM/IEEE groups, technical journals, and conferences in Networking, Security and Systems in the aforementioned areas of computer science. I have created research software for all of my projects, much of which is available online for others to build on.

11. I am a named inventor on four U.S. Patents. These patents are generally related to computer networking and the delivery of services over networks and video delivery systems. I have served as an expert witness and technical consultant in litigation and Inter Partes Review matters concerning videoconferencing, data conferencing and screen sharing, voice over IP (VoIP) telephony, multimedia networking, distributed systems, operating systems, computer networks, and datacenter networking, among others. I have testified in several trials and depositions as an expert witness.

II. MATERIALS REVIEWED AND CONSIDERED

12. My findings, as explained below, are based on my years of education, research, experience, and background in the field of cryptographic techniques for computer systems, as well as my investigation and study of relevant materials for

this declaration. When developing the opinions set forth in this declaration, I assumed the perspective of a person having ordinary skill in the art, as set forth in §V.B below. In forming my opinions, I have studied and considered the materials identified in the list below.

Exhibit	Description
1001	U.S. Patent No. 9,338,140 (“140 patent”)
1002	Prosecution History of U.S. Patent No. 9,338,140
1005	International Application Publication WO 2001/022322 (“Dickinson”)
1007	U.S. Patent Application Publication 2004/0143733 (“Ophir”)
1008	U.S. Patent Application Publication 2001/0053221 (“Takeda”)
1009	U.S. Patent No. 7,187,771
1010	U.S. Patent No. 7,571,320
1011	European Patent Application Publication EP 1 063 811
1012	U.S. Patent No. 6,151,659
1013	U.S. Patent No. 5,526,507
1014	U.S. Patent No. 6,065,096
1015	U.S. Patent No. 7,941,695
1016	U.S. Patent No. 8,352,518
1017	U.S. Patent Application Publication 2002/0174295
1018	U.S. Patent Application Publication 2002/0019908
1019	U.S. Patent No. 7,043,614
1020	U.S. Patent No. 6,973,455
1021	U.S. Patent No. 6,697,846
1022	U.S. Patent No. 6,324,648
1023	U.S. Patent Application Publication 2002/0019932
1024	U.S. Patent No. 5,825,890
1027	U.S. Patent Application Publication 2003/0028493
1028	U.S. Patent Application Publication 2002/0042859
1029	U.S. Patent Application Publication 2005/0216754
1030	U.S. Patent Application Publication 2003/0191938
1031	U.S. Patent Application Publication 2003/0167408
1033	U.S. Patent No. 7,529,834 (“Birrell”)
1034	U.S. Patent No. 5,483,598
1037	Merriam Webster’s Collegiate Dictionary, 207, 381, 888 (10th ed. 1997)
1038	U.S. Patent No. 6,356,915

Exhibit	Description
1039	U.S. Patent No. 4,067,058
1040	Redline Comparison of International Application Publication WO 2001/022322 to U.S. Patent No. 9,338,140
1041	Google LLC's Proposed Claim Terms for Construction in <i>Security First Innovations, LLC v. Google, LLC</i> , 2:23-cv-00097-JWK-RJK (Nov. 11, 2023)
1042	Letter from A. Mayron to D. Perlson re Preliminary Proposed Constructions in <i>Security First Innovations, LLC v. Google, LLC</i> , 2:23-cv-00097-JWK-RJK (Nov. 11, 2023)
1043	Microsoft Computer Dictionary, 99, 192, 362, 407 (5th ed. 2002)
1044	Security First Innovation's 2023-08-14 Infringement Contentions, Chart D in <i>Security First Innovations, LLC v. Google LLC</i> , 2-23-cv-00097 (E.D. Va.)

III. MY UNDERSTANDING OF PATENT LAW

13. In developing my opinions, I discussed various relevant legal principles with Petitioner's attorneys. I understood these principles when they were explained to me and have relied upon such legal principles, as explained to me, in the course of forming the opinions set forth in this declaration. My understanding in this respect is as follows:

14. I understand that "*inter partes* review" (IPR) is a proceeding before the United States Patent & Trademark Office for evaluating the patentability of an issued patent's claims based on prior-art patents and printed publications.

15. I understand that, in this proceeding, Petitioner has the burden of proving that the challenged claims of the '854 patent are unpatentable by a preponderance of the evidence. I understand that "preponderance of the evidence" means that a fact or conclusion is more likely true than not true.

16. I understand that, in IPR proceedings, claim terms in a patent are given their ordinary and customary meaning as understood by a person of ordinary skill in the art (“POSA”) in the context of the entire patent and the prosecution history pertaining to the patent. If the specification or prosecution history provides a special definition for a claim term that differs from the meaning the term would otherwise possess, that special definition controls. If a claim element is expressed as a “means” for performing a specified function, I understand that it covers the corresponding structure described in the specification and equivalents of the described structure. I have applied these standards in preparing the opinions in this declaration.

17. I understand that determining whether a particular patent or printed publication constitutes prior art to a challenged patent claim can require determining the effective filing date (also known as the priority date) to which the challenged claim is entitled. I understand that for a patent claim to be entitled to the benefit of the filing date of an earlier application to which the patent claims priority, the earlier application must have described the claimed invention in sufficient detail to convey with reasonable clarity to the POSA that the inventor had possession of the claimed invention as of the earlier application’s filing date. I understand that a disclosure that merely renders the claimed invention obvious is

not sufficient written description for the claim to be entitled to the benefit of the filing date of the application containing that disclosure.

18. I understand that for an invention claimed in a patent to be patentable, it must be, among other things, new (novel—*i.e.*, not anticipated) and not obvious from the prior art. My understanding of these two legal standards is set forth below.

A. Anticipation

19. I understand that, for a patent claim to be “anticipated” by the prior art (and therefore not novel), each and every limitation of the claim must be found, expressly or inherently, in a single prior-art reference. I understand that a claim limitation is disclosed for the purpose of anticipation if a POSA would have understood the reference to disclose the limitation based on inferences that a POSA would reasonably be expected to draw from the explicit teachings in the reference when read in light of the POSA’s knowledge and experience.

20. I understand that a claim limitation is inherent in a prior art reference if that limitation is necessarily present when practicing the teachings of the reference, regardless of whether a person of ordinary skill recognized the presence of that limitation in the prior art.

B. Obviousness

21. I understand that a patent claim may be unpatentable if it would have been obvious in view of a single prior-art reference or a combination of prior-art references.

22. I understand that a patent claim is obvious if the differences between the subject matter of the claim and the prior art are such that the subject matter as a whole would have been obvious to a person of ordinary skill in the relevant field at the time the invention was made. Specifically, I understand that the obviousness question involves a consideration of:

- the scope and content of the prior art;
- the differences between the prior art and the claims at issue;
- the knowledge of a person of ordinary skill in the pertinent art;
- and
- if present, objective factors indicative of non-obviousness, sometimes referred to as “secondary considerations.” To my knowledge, the Patent Owner has not asserted any such secondary considerations with respect to the ’140 patent.

23. I understand that in order for a claimed invention to be considered obvious, a POSA must have had a reason for combining teachings from multiple

prior-art references (or for altering a single prior-art reference, in the case of obviousness in view of a single reference) in the fashion proposed.

24. I further understand that in determining whether a prior-art reference would have been combined with other prior art or with other information within the knowledge of a POSA, the following are examples of approaches and rationales that may be considered:

- combining prior-art elements according to known methods to yield predictable results;
- simple substitution of one known element for another to obtain predictable results;
- user of a known technique to improve similar devices in the same way;
- applying a known technique to a known device ready for improvement to yield predictable results;
- applying a technique or approach that would have been “obvious to try,” *i.e.*, choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success;
- known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on

design incentives or other market forces if the variations would have been predictable to one of ordinary skill in the art;

- some teaching, suggestions, or motivation in the prior art that would have led one of ordinary skill to modify the prior-art reference or to combine prior-art reference teachings to arrive at the claimed invention. I understand that this teaching, suggestion or motivation may come from a prior-art reference or from the knowledge or common sense of one of ordinary skill in the art.

25. I understand that a universal motivation known in a particular field to improve technology can provide a motivation to combine prior art references even absent any hint or suggestion in the references themselves. I also understand that obviousness is determined in light of all the facts and that a given course of action often has simultaneous advantages and disadvantages, and that this does not necessarily obviate motivation to combine teachings from multiple references.

26. I understand that for a single reference or a combination of references to render the claimed invention obvious, a POSA must have been able to arrive at the claimed invention by modifying, implementing, or combining the teachings of the applied references.

IV. UNPATENTABILITY GROUNDS

27. The table below identifies the references, applicable claims, and basis for each ground of unpatentability.

Ground Number and Reference(s)		Claims	Basis
1	Ophir (EX1007), Takeda (EX1008)	1-6, 8, 10	Obviousness
2	Ophir, Takeda, Birrell (EX1033)	7	Obviousness
3	Ophir, Takeda, Dickinson (EX1005)	9	Obviousness
4	Dickinson, Takeda	1-10	Obviousness

28. I understand that Ophir is prior art to the '140 patent at least based on its January 16, 2003 filing date being before the earliest-alleged priority date of the '140 patent (2004-10-25).

29. I understand that Takeda is prior art to the '140 patent at least based on its December 20, 2001 publication date being more than one year before the earliest-alleged priority date of the '140 patent (2004-10-25).

30. I understand that Birrell is prior art to the '140 patent at least based on its September 22, 2000 filing date being before the earliest-alleged priority date of the '140 patent (2004-10-25).

31. I understand that Dickinson is prior art to the '140 patent at least based on its March 29, 2001 publication date being more than one year before the earliest-alleged priority date of the '140 patent (2004-10-25).

V. THE '140 PATENT

A. Specification

32. The '140 patent relates to a data storage system and method providing security for stored data to decrease the risk of unauthorized access. EX1001, Abstract. Much of the '140 patent's lengthy disclosure reproduces prior-art material in PCT Publication WO2001/022322 (EX1005 "Dickinson"), which is an application published years earlier by overlapping inventors. Specifically, FIGs. 1-20 and the corresponding written disclosure is reproduced substantially identically from Dickinson. *Compare* Dickinson, FIGs. 1-20, 7:34-54:2 *with* EX1001, FIGs. 1-20, 6:52-7:40, 9:16-52:54. This is shown in EX1040, which is a PDF I requested to be generated that compares in "redlines" the text of the written description of Dickinson to the '140 patent (*see* EX1040, 1-180) and shows side-by-side their respective Figures 1-20 (*see* EX1040, 181-201. The written description of Dickinson used to make the comparison in EX1040 was obtained from the European Patent Office's Espacenet Patent Search (<https://worldwide.espacenet.com/>), and the written description of the '140 patent used to make the comparison in EX1040 was obtained from the USPTO's Patent Public Search tool (<https://ppubs.uspto.gov/pubwebapp/>). I understand that Dickinson is prior art to the '140 patent because it published years earlier. Dickinson, Cover Page.

33. The '140 patent discusses a “data parser” that splits data into “shares,” encrypts the data using a key, and distributes the encrypted data among different storage locations. *Id.* This technique was already known to POSAs by the time the '140 patent application was filed. For example, the concept was already published in Dickinson. Dickinson, 2:25-29. POSAs recognized that this technique of splitting data and distributing it among different storage locations was less risky than storing all the data in a single storage location. Dickinson, 2:25-29, 14:30-15:3, 17:4-5, 19:27-20:14. POSAs knew that storing all data in a single location is riskier because if that storage location is compromised, for instance by being hacked into, then all the stored data would be exposed to attackers. Dickinson, *Id.*

34. The '140 patent expands on Dickinson by ***broadening*** the use cases for the security module and adding specifics on how data splitting can be performed, but those details don't appear in the claims. EX1001, 52:55-53:18.

35. Dickinson gives examples of its system operating on authentication data in electronic commerce (“e-commerce”) transactions. Dickinson, Abstract, 1:6-8. However, the '140 patent says that the data splitting technology, which it renames as a “secure data parser,” can be used for other purposes too. EX1001, 52:55-53:18. For example, the '140 patent states: “[t]he data splitting module, or secure data parser, comprises a cross-platform software module suite which integrates within an

electronic infrastructure, or as an *add-on to any application*¹ which requires the ultimate security of its data elements. This parsing process operates on *any type of data set*, and on *any and all file types, or in a database* on any row, column or cell of data in that database.” EX1001, 52:66-53:5. The ’140 patent says that the parsing process may be modular and “*any encryption process is suitable for use in the process of the present invention.*” EX1001, 53:6-9. The ’140 patent also says that:

The modular tiers of the parsing and splitting process of the present invention may include, but are not limited to, 1) cryptographic split, dispersed and securely stored in multiple locations; 2) encrypt, cryptographically split, dispersed and securely stored in multiple locations; 3) encrypt, cryptographically split, encrypt each share, then dispersed and securely stored in multiple locations; and 4) encrypt, cryptographically split, encrypt each share with a different type of encryption than was used in the first step, then dispersed and securely stored in multiple locations.

EX1001, 53:9-18. The above disclosures indicate that the ’140 patent’s various processes may be ordered in different ways and the ’140 patent considers those all to be acceptable without ascribing particular significance to any given ordering.

¹ I have added all emphasis throughout this declaration unless I have indicated otherwise.

36. In other words, the '140 patent says (which POSAs would have independently understood) that Dickinson's prior-art data parsing/splitting technology is applicable beyond e-commerce. *Id.* The '140 patent's figures and disclosure beyond that disclosed in Dickinson largely just describe either additional contexts (use cases) for the technology or specific encryption techniques, ***none*** of which appear in the challenged claims. EX1001, FIGs. 21-24, 46-47 (illustrating methods for parsing, splitting, and/or separating data and recovering data), 7:41-52, 8:39-44, FIGs. 30-32, 43, 53-54 (illustrative systems integrating the secure data parser), 8:1-3, 8:30-32, 8:59-64, FIGs. 33-39 (illustrating processes for splitting, restoring, and scattering data), 8:4-22, FIGs. 48-51 (illustrating parameters, data structures, and data formats for the secure data parser), 8:45-55, FIGs. 56-58 (use of workgroup keys), 9:1-8. Indeed, the '140 Patent's alleged invention applies to "***any*** application" for "***any*** type of data set" and can use "***any*** encryption process." EX1001, 52:66-53:9.

B. Person of Ordinary Skill in the Art ("POSA")

37. I have been informed and understand that for purposes of assessing whether prior-art references disclose every element of a patent claim (thus "anticipating" the claim) and/or would have rendered the claim obvious, the patent and the prior-art references must be assessed from the perspective of a person having ordinary skill in the art ("POSA") to which the patent is related, based on the

understanding of that person at the time of the patent claim's priority date. I have been informed and understand that a POSA is presumed to be aware of all pertinent prior art and the conventional wisdom in the art, and is a person of ordinary creativity. I have applied this standard throughout my declaration.

38. The '140 patent involves technology in the field of securing data from unauthorized access or use. *See* EX1001, 1:17-18 ("The present invention relates in general to a system for securing data from unauthorized access or use."). I have been asked to provide my opinions as to the state of the art in this field by 2004/2005. I use this timeframe because the face of the '140 patent indicates an earliest claimed priority date of October 25, 2004. Whenever I offer an opinion in this declaration about the knowledge of a POSA, the manner in which a POSA would have understood the claims of the '140 patent or its description, the manner in which a POSA would have understood the prior art, or what a POSA would have been led to do based on the prior art, I am referencing the 2004/2005 timeframe, even if I do not say so specifically in each case. Demonstrating that the '140 patent's claims were anticipated or would have been obvious by October 2004 demonstrates that they remained so any time after (including as of the October 25, 2005, filing date of the non-provisional application corresponding to U.S. Patent No. 8,226,438, which the '140 patent claims priority to as a continuation), because the POSA's knowledge only increases over time.

39. I understand that the Patent Owner may attempt to prove that the alleged invention recited in the challenged claims was conceived at some time prior to the earliest claimed priority date on the face of the patent. At the time of this declaration, I am unaware of the Patent Owner having alleged any earlier conception date or produced any evidence to establish any earlier conception date.

40. In my opinion, the person of ordinary skill in the art (POSA) at the time of the '140 patent's October 25, 2004 earliest claimed priority date would have at least a bachelor's degree in computer science, computer engineering, or a related field, with three (3) years of experience in the area of securing data from unauthorized use or access. A higher level of education may substitute for less experience. This person would have been capable of understanding and applying the teachings of the '140 patent and the prior-art references discussed in this declaration.

41. By 2005, I held a Ph.D. in computer science from the Georgia Institute of Technology, and I had over 10 years of experience with securing data from unauthorized use or access. *See* §I above. Therefore, I was a person of more than ordinary skill in the art during the relevant timeframe. However, I worked with many people who fit the characteristics of the POSA, and I am familiar with their level of skill. When developing the opinions set forth in this declaration, I assumed the perspective of a person having ordinary skill in the art, as set forth above.

C. Prosecution History

42. I am aware that the claims of the '140 patent changed during examination by the U.S. Patent Office. I understand that the challenged claims were numbered as claims 14-30 during examination before the patent office. I also understand that claims 14-25 were included when the application was filed and that the other claims were added during the examination process. EX1002, 275-79.

43. I am also aware that the Examiner rejected claims 14-25² as obvious over a combination including U.S. Patent No. 7,187,771 (EX1009, "'771 Patent"), which is assigned on its face to Security First Corporation (*i.e.*, Patent Owner), listing Alexander Dickinson as the lead inventor, filed the same day as Dickinson (EX1005), and including substantially identical FIGs. 1-20 as Dickinson. EX1002, 325-32; *compare* Dickinson, FIGs, 1-20 *with* EX1009, FIGs. 1-20. The Examiner found the '771 Patent disclosed splitting data and encryption keys into shares and

² I understand that additional rejections were made during the examination process for lack of written description, double-patenting, anticipation, and obviousness. EX1002, 314-35, 738-48, 800-08, 936-40, 1233-42. The other obviousness rejections were overcome by arguments that one or more of the references was not prior art.

storing them across storage devices. EX1002, 327-28. I'm aware the Applicant did not disagree. EX1002, 669-71.

44. I understand that in the rejection preceding allowance, the Examiner applied Dickinson as the primary reference in an obviousness rejection, finding that Dickinson disclosed nearly all limitations of the claim set. EX1002, 1235-42.

45. I understand that the Examiner found Dickinson lacked only that the recited "virtual disk" comprises a directory mapped to the plurality of storage devices. EX1002, 1239. For that, I understand the Examiner relied on a secondary reference named Bendix. *Id.*

46. Applicant did not dispute that Dickinson met most claim limitations. EX1002, 1245-46. I understand that, instead, Applicant asserted that the rejections inconsistently identified which secondary reference was combined with Dickinson, and the Applicant stated it was "unable to address with specificity the assertions made with the combination of Dickinson and Bendix." *Id.* I understand that the alleged inconsistency, though, would not have precluded Applicant from responding to the Examiner's finding that Dickinson disclosed all but one claimed feature. *Id.*

47. Rather than respond to the Examiner's finding, Applicant amended each independent claim by adding a limitation reciting "include with each of the plurality of shares data indicative of the at least one session key used to secure the dataset." *Id.*, 1247-50. Applicant argued that neither Dickinson nor the secondary

reference “teach or suggest storing any additional information with each of the data shares, much less information used to secure a dataset.” *Id.*, 1245-46. Thus, the only feature in the claim set that Applicant argued was **not** disclosed by Dickinson was this added limitation. As I discuss in §§ VII.B and VII.C below, that added limitation is taught in the prior art by (at least) Takeda, which I understand was not a reference of record during prosecution of the ’140 patent. I understand the Examiner allowed the application in response to the amendment. EX1002, 1360-61.

48. For the reasons I discuss below in §§VII-X, it is my opinion that Grounds I-IV demonstrate unpatentability of all challenged claims.

D. Challenged Claims

49. Claims 1-10 are reproduced in the Claim Listing (§XI below), and independent claim 1 is reproduced below with limitations labeled 1[PRE]-[1B-4]:

1[PRE]. A secure storage network comprising:

[1A] a plurality of physical storage devices storing thereon a plurality of shares, the plurality of shares being associated with at least one session key used to secure a dataset; and

[1B] a secure storage system configured to:

[1B-1] present to a client device a virtual disk, the virtual disk comprising a directory mapped to the plurality of physical storage devices such that physical locations of the shares are hidden from the client device;

[1B-2] generate the plurality of shares for storage on the plurality of physical storage devices by performing a securing operation on the dataset received from the client device and distributing the dataset in the shares;

[1B-3] include with each of the plurality of shares data indicative of the at least one session key used to secure the dataset; and

[1B-4] reconstitute the dataset from at least a portion of the plurality of shares stored on the physical storage devices in response to a request from the client device for information in the dataset.

50. As Ground 1 demonstrates (§VII below), in my opinion, each limitation recited in claim 1 (and claims 2-6, 8, 10) would have also been obvious to a POSA over the combination of Ophir and Takeda. As Ground 2 demonstrates (§VIII below), claim 7 would have been obvious over the combination of Ophir, Takeda, and Birrell. As Ground 3 demonstrates (§IX below), claim 9 would have been obvious over the combination of Ophir, Takeda, and Dickinson. As Ground 4 demonstrates (§X below), claims 1-10 would have been obvious over the combination of Dickinson and Takeda.

VI. CLAIM INTERPRETATION

51. I understand that determining whether a claimed invention is novel and/or non-obvious requires comparing the prior art to the claim. I understand that in some scenarios, an express construction defining the outer bounds of one or more claim terms may be appropriate during an IPR proceeding—for example, where the prior art would meet a challenged claim under one proposed construction but not

another construction. In this IPR proceeding, I have been asked to provide my opinion concerning whether certain limitations are met under the parties' preliminary claim constructions in the related district court litigation (EX1041, EX1042, which are summarized in the table below) and/or under the claim scope that Patent Owner appears to ascribe to the claims in its infringement contentions (EX1044). In my opinion, Petitioner's proposed constructions (EX1041) are reasonable interpretations of the claim terms to a POSA at the time of the alleged invention. For my opinions regarding anticipation and obviousness, it is unnecessary to further analyze which party's proposed construction most correctly sets forth the exact bounds of the ordinary meaning of any claim term for which either party has proposed a construction, because I have applied each party's construction to every construed term (and I have applied the ordinary and customary meaning to any term for which neither party has proposed a construction in the district court) and, for the reasons discussed in the Grounds below, it is my opinion that the challenged claims are met by at least one combination of the prior art either way (*i.e.*, under either party's proposed constructions).

Claim term	Petitioner's construction (EX1041)	Patent Owner's construction (EX1042)
<i>"distributing the data set in the shares"</i>	"splitting the data set into shares"	No construction necessary. In the alternative: "dividing and placing the

Claim term	Petitioner's construction (EX1041)	Patent Owner's construction (EX1042)
		data set into a plurality of the shares"
"data indicative of ... key(s)"	"metadata about ... key(s); not an encryption key"	No construction necessary. In the alternative: "information that points out, points to, makes known, shows, directs to, hints at, or suggests the at least one session key"
"dataset"	"collection of related information made up of separate elements that can be treated as a unit in data handling"	"a collection of information for storage"
"virtual disk"; "the virtual disk comprising a directory mapped to the plurality of physical storage devices"	"a distributed file system shared among file servers that includes a directory with a clients' view of all file assets mapped to the physical storage devices on which they are stored"	No construction necessary. In the alternative: "a system that simulates or emulates a physical storage device, comprising a directory mapped to a plurality of physical storage devices such that physical locations of the shares are hidden from the client device"
"session key"	"Key generated each time data is secured"	"a key created to both secure and restore a specific collection of information"
"a plurality of data shares / the plurality of shares / the shares"	"A plurality of data shares, the plurality of shares and the shares are	[N/A: did not separately construe]

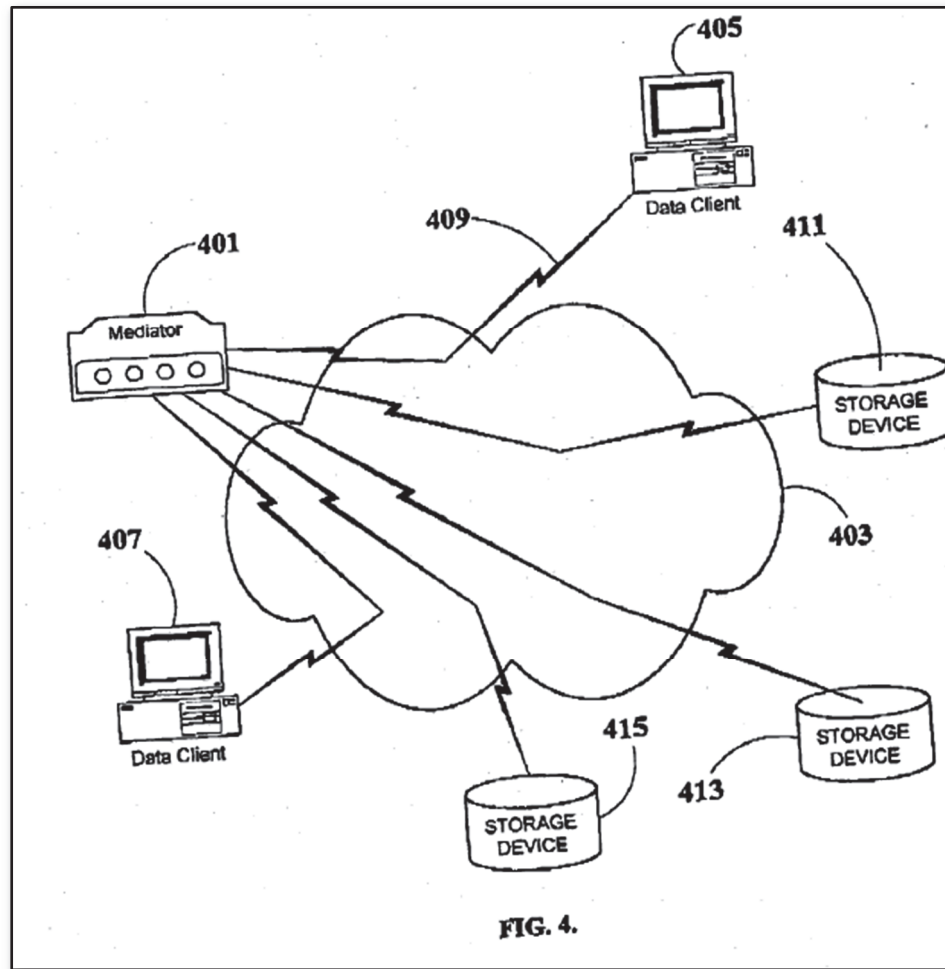
Claim term	Petitioner's construction (EX1041)	Patent Owner's construction (EX1042)
	all the same set of shares”	
“at least one session key / the at least one session key”	“at least one session key and the at least one session key are the same key”	[N/A: did not separately construe]
“virtual disk / the virtual disk”	“virtual disk and the virtual disk are the same”	[N/A: did not separately construe]
“a plurality of data storage devices / the plurality of data storage devices”	“a plurality of data storage devices and the plurality of data storage devices are the same set of storage devices”	[N/A: did not separately construe]
“a secure storage system / the secure storage system”	“a secure storage system and the secure storage system are the same system”	[N/A: did not separately construe]
“a secure storage network / the secure storage network”	“a secure storage network and the secure storage network are the same network”	[N/A: did not separately construe]

VII. GROUND 1: OPHIR AND TAKEDA RENDER CLAIMS 1-6, 8, 10 OBVIOUS

A. Ophir (EX1007)

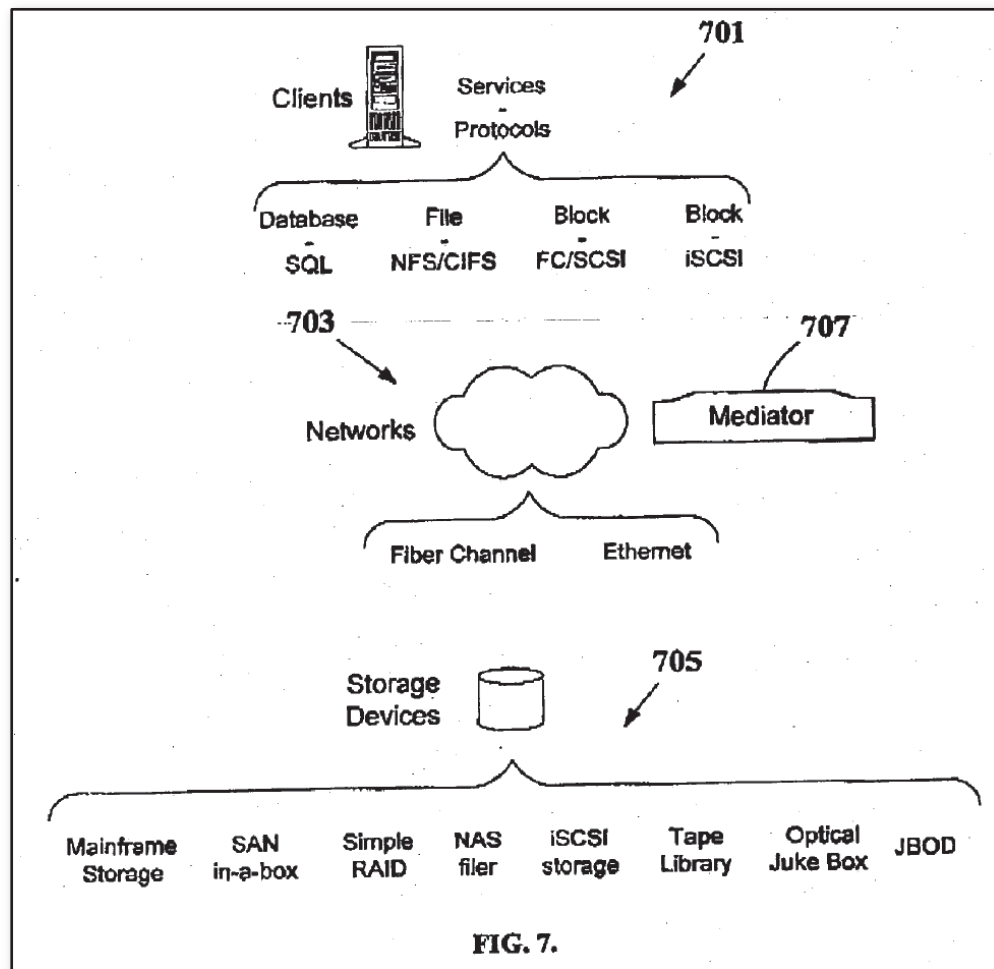
52. Ophir provides a secure network data storage mediator. Ophir, Title, Abstract. Ophir states that “the mediator connects over the network to one or more data clients and to one or more data storage devices, and provides secure storage of data for the data clients on the data storage devices.” Ophir, Abstract, *see also* [0015]. I have reproduced Ophir’s FIG. 4 below. As Ophir states, FIG. 4 shows that “mediator [401] operates as a central point for handling secure storage over a

network both from the standpoint of the data clients as well as from the standpoint of the data storage devices, while not requiring the data clients to be involved with the protection of the data while in storage.” Ophir, [0027].



53. Ophir’s Figure 4 above also shows multiple data clients 405 and 407. Ophir’s data clients, such as 405 and 407, are “any client [that] wishes to place data in storage or retrieve data from storage.” Ophir, [0004]. Ophir’s mediator provides a number of different services to the data clients including file services, block services, and database services. Ophir, [0031], [0034]. This is illustrated in Ophir’s

FIG. 7, which I have reproduced below. As shown in FIG. 7, the “Clients” provide the three types of services that I listed and operate to provide those services using the protocols illustrated in FIG. 7:



54. Data client communication protocols include “network file system” (“NFS”) or “common internet file system” (“CIFS”) protocols for file services, and “fiber channel” (“FC”), “small computer system interface” (“SCSI”) or “internet small computer system interface” (“iSCSI”) protocols for block services. Ophir, FIG. 7, [0025], [0034].

55. I have also reproduced Ophir's FIG. 5 below. Figure 5 illustrates an example of the mediator of Ophir's FIG. 4. Ophir, [0030]. The mediator 501 has an interface 503 for communicating with data clients and an interface 527 for communicating with data storage devices. Ophir, [0030]. The mediator 501 also includes an encryption/decryption unit 517 within a data storage processor 519. Ophir, [0032]. The encryption/decryption unit 517 optionally includes an encryption unit 513 that operates to "encrypt[] data from the data clients into encrypted data for safe storage in data storage devices." Ophir, [0032]. The encryption/decryption unit 517 also optionally includes a decryption unit 515 that operates to "decrypt[] data retrieved from data storage devices into decrypted data for sending to data clients." Ophir, [0032].

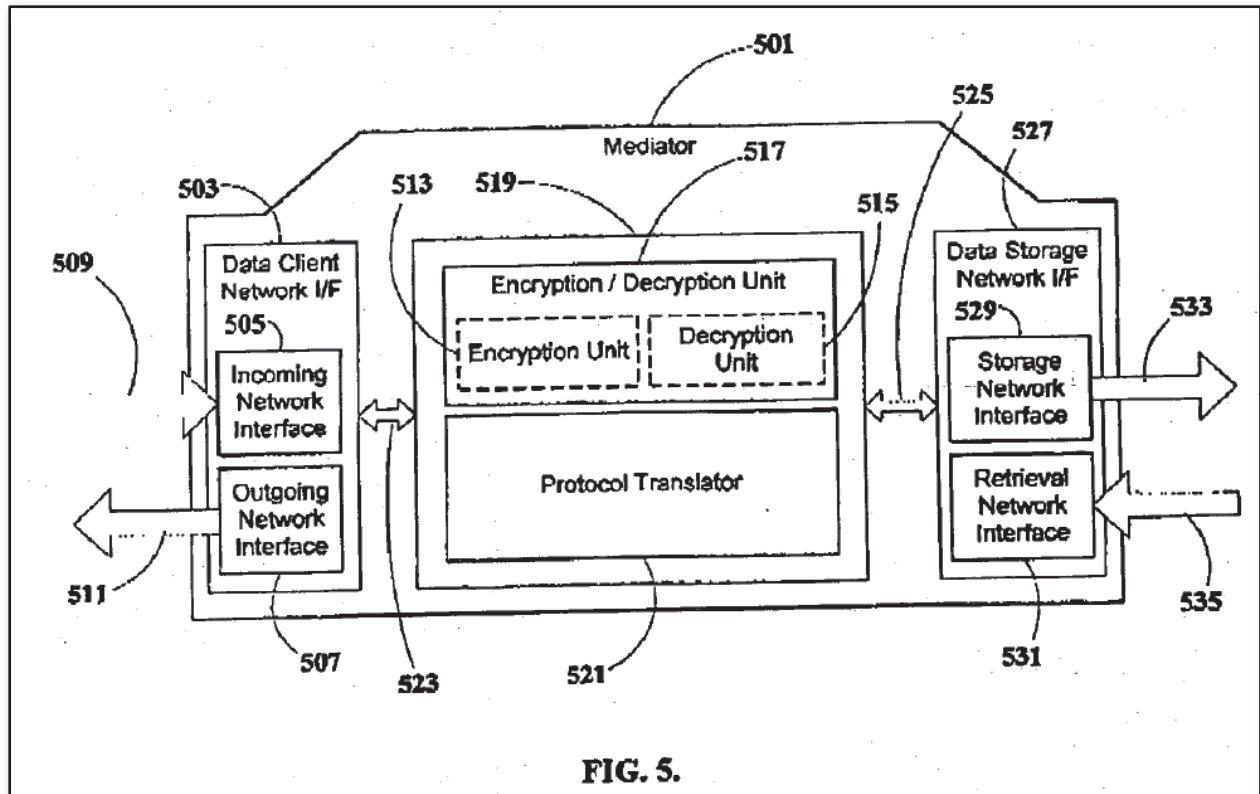


FIG. 5.

56. Ophir's mediator, labeled as item 401 in FIG. 4 and item 501 in FIG. 5, encrypts the data that is to be stored on the storage devices and it also decrypts data that has been stored on the storage devices to return it to the clients. Ophir, FIG. 5, [0010]-[0011], [0013], [0032].

57. In a section titled "Encryption Scenarios," Ophir provides examples of a "typical mediator data encryption scenario for writing data to storage." [0035]-[0037]. Paragraphs [0038]-[0043] provide a first scenario for writing data to storage, and paragraph [0044] provides a variation.

58. Ophir also provides a "typical mediator data decryption scenario for reading data from storage" in paragraphs [0045]-[0050].

59. Ophir says that “[a]dditional variations on the above scenarios involve using a key server to generate, store and retrieve encryption keys according to a unique ID which the mediator stores for each storage object (e.g. logical units, files, directories).” Ophir, [0051].

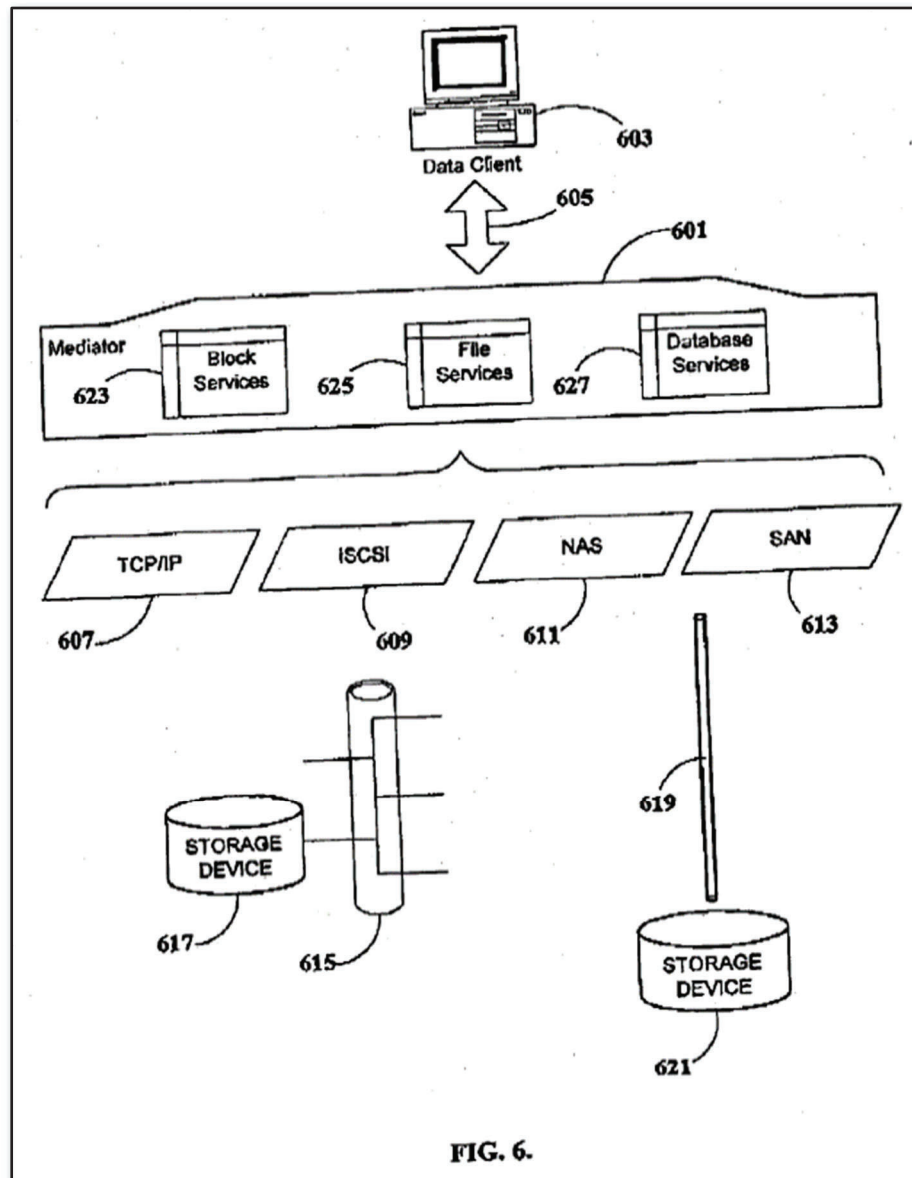
60. Ophir makes clear that the scenarios listed above are just examples, and that other scenarios are possible. For example, Ophir says the examples “represent *possible* encryption scenarios in embodiments of the present invention. It is noted that these are all *non-limiting examples* provided for illustration, and that other scenarios are also possible within the framework of the invention.” Ophir, [0036].

61. As shown in the bullet points below, Ophir emphasizes that the mediator provides virtual storage:

- “At the same time, however, the *mediator provides secure virtual storage* to data clients without requiring them to be involved in any of the encryption or decryption operations.” Abstract.
- “In is manner, the mediator functions as a central coordinator for data storage between one or more clients requesting data storage and one or more storage devices providing data storage. In this central point, the *mediator serves as a virtual secure storage device.*” [0013].
- “FIG. 6 illustrates the capacity of a *mediator 601 to effort secure virtual data storage* for a data client 603 over a network connection

605. The storage is considered “virtual” because the data from data client 603 can be stored on a variety of storage devices using a variety of protocols, technologies, and services, as managed by mediator 601.” [0033].

62. Below I have reproduced Ophir’s FIG. 6:



63. As I explain below in §VII.D.1.c.i, Ophir's mediator operating to provide secure virtual storage satisfies the limitations required by the challenged claims relating to a "virtual disk." Abstract.

B. Takeda (EX1008)

64. In this section, before discussing Takeda's disclosure about its encryption methods and systems I begin by discussing various terms used in Takeda and how they relate to terms used in the '140 patent and challenged claims.

65. Takeda uses the term "ciphering." People skilled in the art relevant to the '140 patent knew that the terms "ciphering" and "encryption" are synonyms. For example, one dictionary definition lists the definition of "encrypt" as "encipher." EX1037, 381. Another dictionary defines "encipher" by directing the reader to "encrypt" and for the definition of "encrypt" that dictionary notes that the term is also called "encipher" or "encode." EX1043, 192.

66. "Ciphertext" is another term that appears in Takeda. Takeda discloses that "[c]iphertext" is "the enciphered form of a text or of its elements." EX1037, 207. Another dictionary defines "ciphertext" as "the scrambled or otherwise encoded text of an encrypted message." EX1043, 99.

67. "Plaintext" is another term that appears in Takeda. POSAs understood that "plaintext" is nonencrypted (or "unciphered") information. For instance, one dictionary definition of "plaintext" is "the intelligible form of an encrypted text or

of its elements,” as distinguished from “ciphertext”. EX1037, 888. Another dictionary definition of “plaintext” is “nonencrypted or decrypted text.” EX1043, 407. I further note that Takeda says “plaintext” is not limited to text data but can also be image data, which is consistent with a POSA understanding that unencrypted / unciphered image data could be referred to as “plaintext” data even though it’s not “text.”

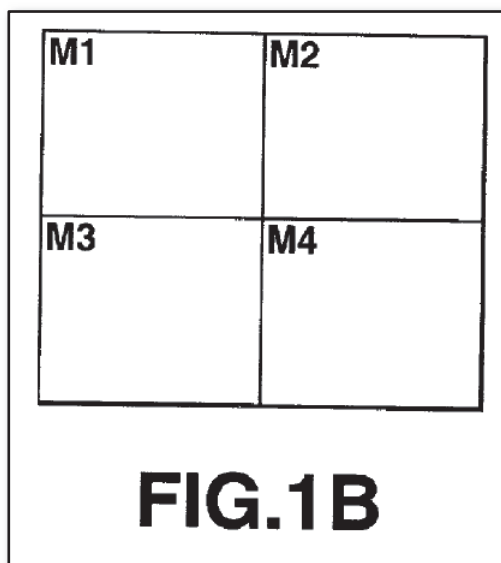
68. As I discuss below in this section, Takeda provides both a ciphering apparatus and method. Takeda, Title, Abstract, [0001]. According to Takeda, conventional ciphering techniques divided data into blocks and then applied the same ciphering to each block. [0003]-[0004]. Takeda says that that approach of applying the same ciphering to the different blocks is risky because if any part of the ciphered data is compromised, then there is the possibility that all the ciphered data will be compromised (because knowledge of the ciphering technique for one block provides knowledge of the ciphering technique for all blocks). Takeda, [0003]-[0004]. Takeda’s ciphering technique improves on the conventional approach because it does not use the same ciphering for each block of data, and therefore “even if a part of the data is deciphered, the rest of the data is not easily deciphered.” Takeda, [0005]-[0006]. This approach therefore provides increased security for the data by using different ciphering techniques for each block of data.

69. Takeda's ciphering technique divides plaintext into blocks, ciphers the different blocks using different "ciphering attributes" (e.g., ciphering keys and/or ciphering algorithms), and outputs the ciphered blocks with the ciphering attributes. Takeda, [0007]-[0009], [0039], [0042]. Takeda explains that its methods can be performed both on two-dimensional (2D) data, and on stream (linear) data. The first embodiment relates to operation on 2D data, is discussed from paragraph [0035] to paragraph [0064], and relates to FIGs. 1-10. The second embodiment relates to operation on stream data, is discussed beginning at paragraph [0065], and relates to FIGs. 11-18. Takeda, [0065]-[0086]. The two embodiments are substantially the same in their features relevant to this declaration. For simplicity, the following discussion refers primarily to Takeda's operation in the context of the first embodiment relating to 2D data. However, my opinions with respect to Takeda and with respect to combining Takeda with other references as discussed below apply equally to Takeda's operation on stream data. Therefore, to the extent I refer to specific portions of Takeda relating to the first embodiment when explaining my opinions, it should be understood that it is my opinion that the corresponding portions of Takeda relating to the second embodiment apply equally well.

70. The ciphering attributes can be stored with the ciphered blocks. Takeda, [0015], [0048] ("in FIG. 3A and FIG. 3B, the ciphertext {Cn} and attribute {An}

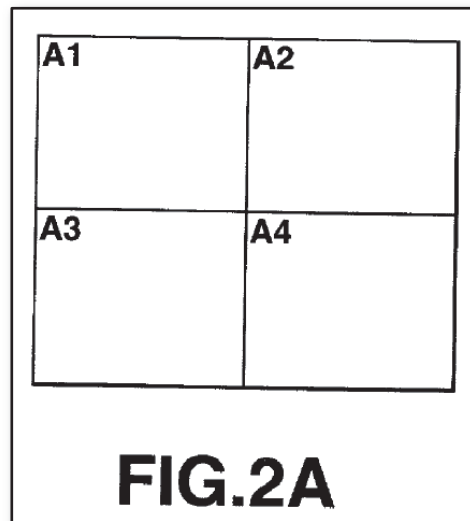
are stored as one file in such a manner that the correspondence relation between the data can be recognized.”), FIG. 3A; [0072], FIG. 11E.

71. In Takeda’s ciphering process, the plaintext data is first divided into blocks, which is illustrated in FIG. 1B (copied below). Takeda, [0007]-[0008]; *see also* [0068], FIG. 11B (blocks M1-M5). Figure 1B illustrates a division of plaintext M into four blocks Mn. Takeda, [0038]. In this context, “n” is an index that represents an integer. In the example of FIG. 1B, since there are four blocks, n can be 1, 2, 3, or 4 and therefore the blocks are labeled as M1-M4, as shown in FIG. 1B below. Takeda, [0038].



72. Takeda’s ciphering process also includes setting the ciphering attributes that are used to cipher the plaintext blocks. Takeda, [0007]-[0008], [0068]. This is illustrated in Takeda’s FIG. 2A, which I have reproduced below. Figure 2A illustrates a table of attributes An defining the ciphering that will be performed on

the respective blocks M_n of the plaintext. Takeda, [0040]; *see also* FIG. 11C. So again here “ n ” represents the index that I explained in connection with FIG. 1B. And so the ciphering attribute A1 is for plaintext block M1, ciphering attribute A2 is for plaintext block M2, and so on. Takeda, [0040], [0068].



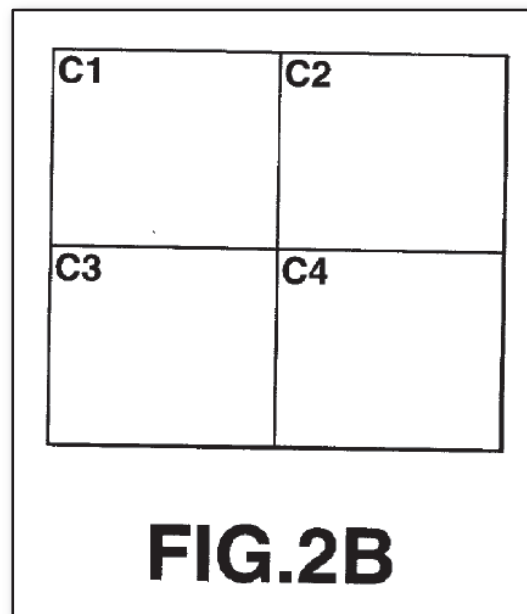
73. Takeda explains that the ciphering attributes A1-A4 (*e.g.*, FIG. 2A) can include various attributes, including ciphering keys and/or ciphering algorithms. Takeda, [0039], [0042]. I have reproduced Takeda’s FIG. 4 below, which shows an example of the constituents of an attribute table storing attribute data. Takeda, [0016], [0053]-[0054]. As shown in FIG. 4 below, each attribute table record can include a ciphering key and additional information including the key length, bit rectangular start point, bit rectangular size, and access privilege identifying who is

permitted access to decipher the data. Takeda, [0054]. FIG. 12 shows a comparable table for the second embodiment relating to ciphering stream data. Takeda, [0076].

RECORD1	BIT RECTANGULAR START POINT	BIT RECTANGULAR SIZE	ACCESS PRIVILEGE	KEY LENGTH	CIPHERING KEY
RECORD2	BIT RECTANGULAR START POINT	BIT RECTANGULAR SIZE	ACCESS PRIVILEGE	KEY LENGTH	CIPHERING KEY
RECORD3	BIT RECTANGULAR START POINT	BIT RECTANGULAR SIZE	ACCESS PRIVILEGE	KEY LENGTH	CIPHERING KEY
RECORD4	BIT RECTANGULAR START POINT	BIT RECTANGULAR SIZE	ACCESS PRIVILEGE	KEY LENGTH	CIPHERING KEY
...					
FIG.4					

74. Each of the plaintext blocks (e.g., M1-M4) that has been generated by dividing the plaintext M is ciphered using its corresponding ciphering attribute (e.g., A1-A4). Takeda, [0007]-[0008], [0041], [0068], [0071]. For instance, the block M1 is ciphered using the ciphering attributes A1, the block M2 is ciphered using the attributes A2, and so on. The result of ciphering the plaintext blocks Mn with the ciphering attributes An is ciphertext blocks C1-C4, as shown below in FIG. 2B. Takeda, [0007]-[0008]; *see also* FIG. 11D. The result of applying each ciphering attribute to a respective plaintext block is ciphertext blocks Cn, where again n represents the same index that I explained in connection with FIG. 1B and FIG. 2A.

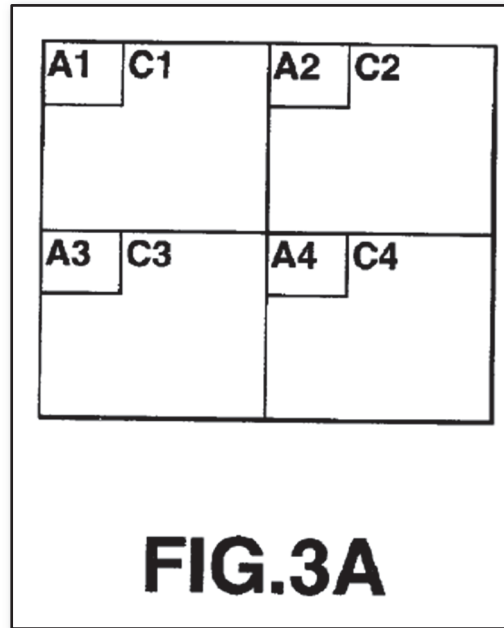
For example, as shown in FIG. 2B (which I have reproduced below), there are four ciphertext blocks C1-C4. Takeda, [0007]-[0008], [0041]. Takeda further discloses that the ciphering attributes (the An's) can differ for the different blocks being ciphered. Takeda, [0009], [0054] (disclosing that the records of FIG. 4's attribute table can differ for each plaintext block).



75. Takeda explains that the attributes (An) can be stored with their respective ciphertext blocks (Cn). Takeda, [0048]; *see also* [0072]. I've reproduced FIG. 3A below, which illustrates the result of ciphering the blocks M1-M4 to generate ciphertext blocks C1-C4 and storing the ciphering attributes A1-A4 with the respective ciphertext blocks. Takeda, [0048]; *see also* [0072]. As I noted above, Takeda explains that the ciphering attributes can include the ciphering keys. Takeda, [0042]; *see also* FIGs. 4 and 12. That means that as shown in FIG. 3A (which I have

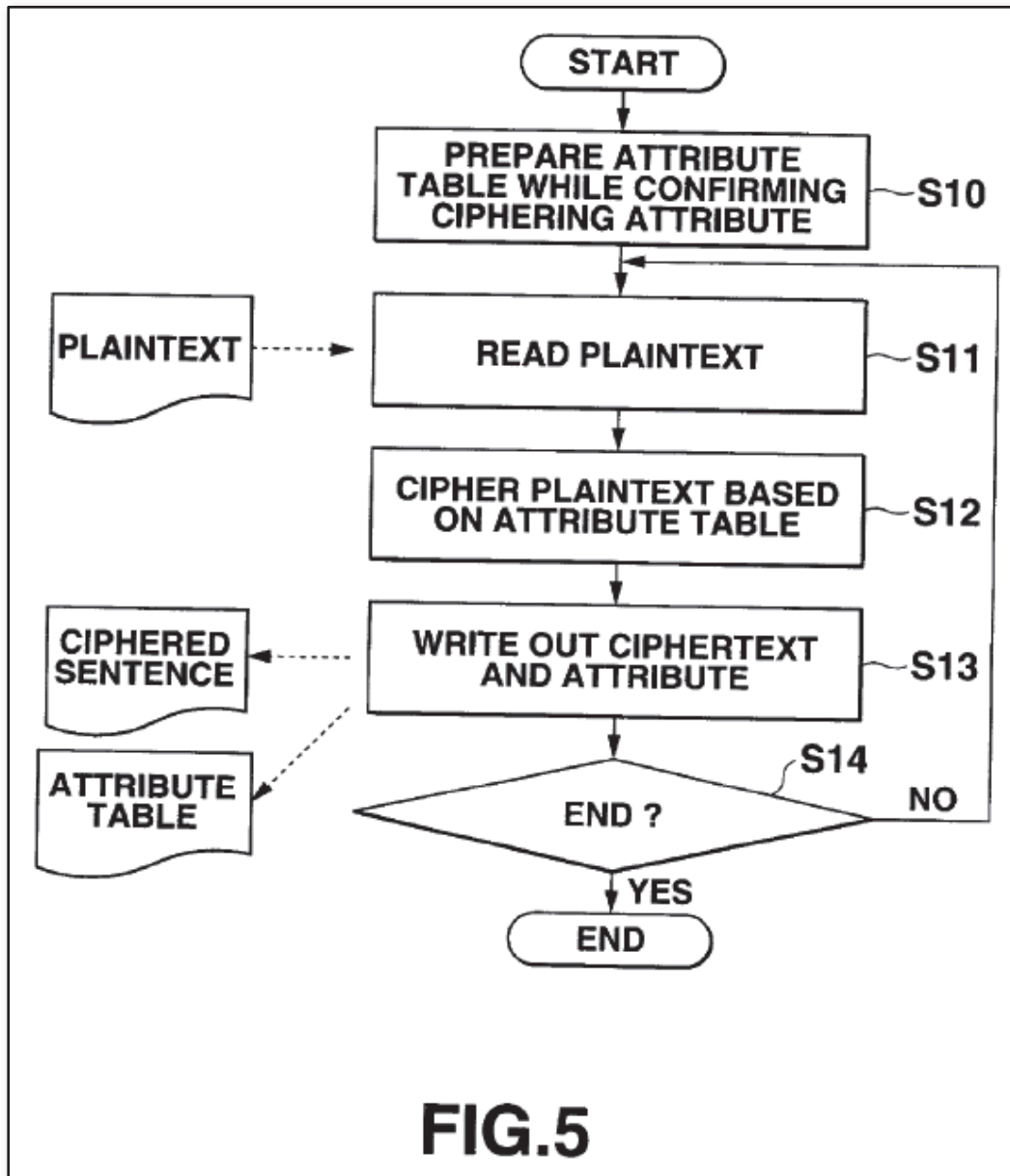
reproduced below), the ciphering keys can be stored with the ciphertext blocks.

Takeda, [0042], [0048], [0072]; *see also* FIGs. 11E-11F.

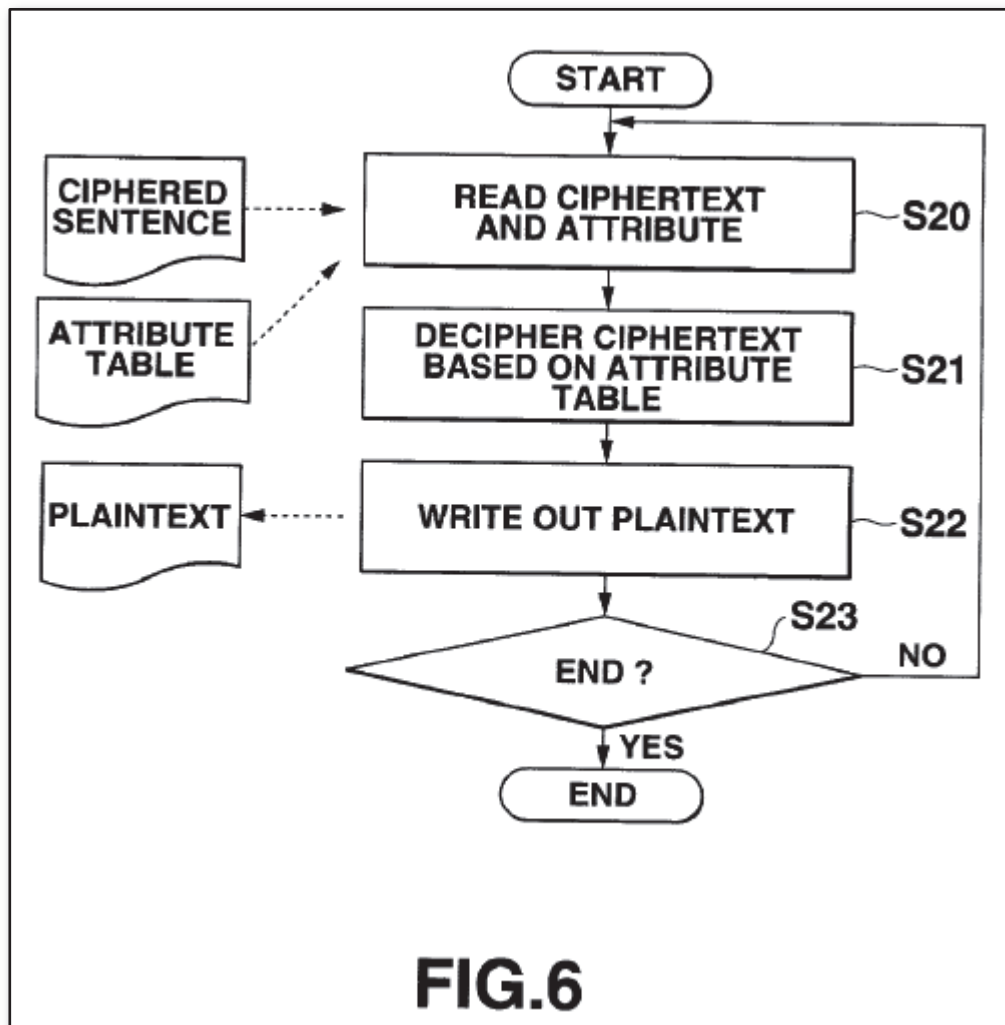


76. Takeda provides an alternative to storing the ciphering attributes with the ciphertext. Takeda, [0040]. In the alternative, a ciphering attribute can be stored separately from its respective ciphered block, and a pointer can be included with the ciphered block pointing to where the attribute for that block is stored. Takeda [0040].

77. I've reproduced FIG. 5 of Takeda below. That figure shows one embodiment of Takeda's above-discussed ciphering process. Takeda, [0017], [0055]-[0056].



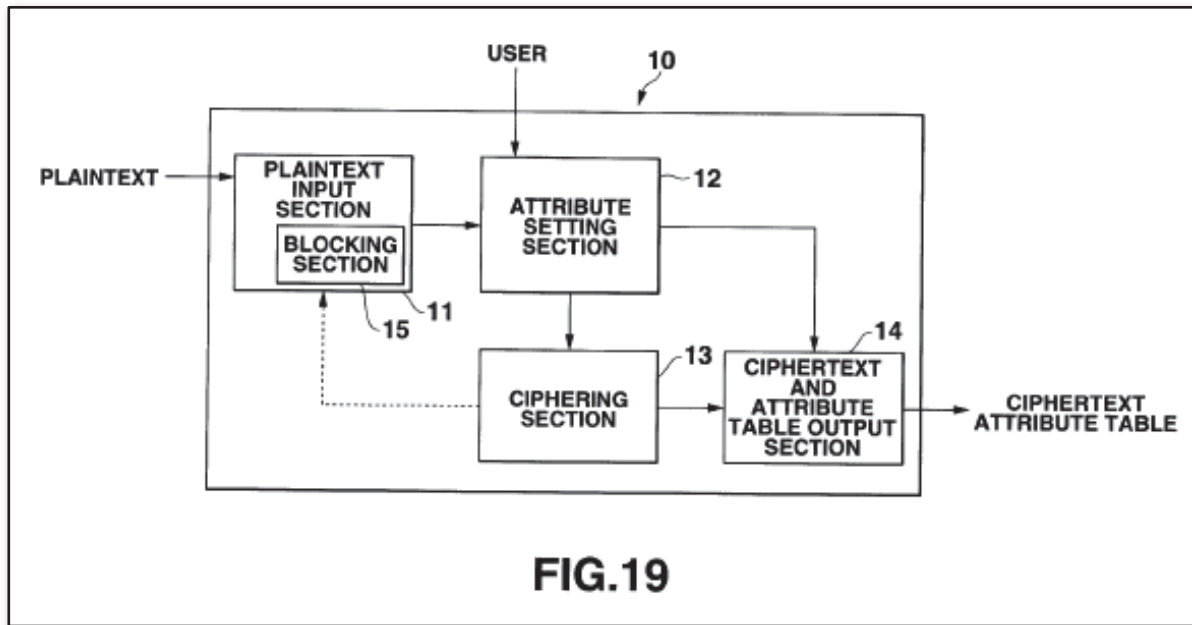
78. I've reproduced FIG. 6 of Takeda below. That figure shows a deciphering process, in which stored ciphertext is retrieved and deciphered using the corresponding attribute. Takeda, [0018], [0057]-[0058].



79. Takeda's FIG. 19, which I have reproduced below, illustrates a ciphering apparatus 10 that can perform Takeda's ciphering method. Takeda, [0031], [0087]. As Takeda explains, plaintext that is input to plaintext input section 11 is formed into input blocks by blocking section 15. Takeda, [0088]. The user can

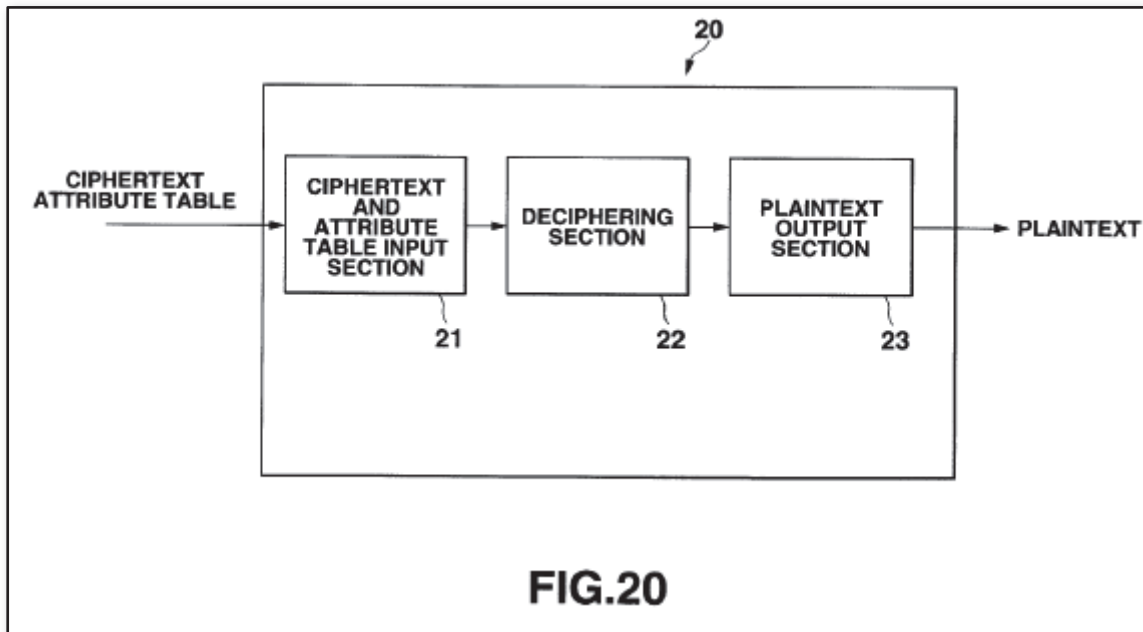
then set the attribute for each block using attribute setting section 12, and then ciphering section 13 ciphers each plaintext block using the corresponding attribute.

Id. The resulting ciphertext and attribute table are output by section 14. *Id.*



80. Takeda's FIG. 20 (which I have reproduced below) illustrates a deciphering apparatus 20. Takeda, [0032]. As illustrated in FIG. 20, the stored ciphertext and attribute table are retrieved and input through input section 21 to deciphering section 22, which consults the attribute table and decipheres the

ciphertext to recover the plaintext, which is output through section 23. Takeda, [0090].



C. Ophir-Takeda Combination

1. Reasons to Combine

81. In my opinion, a POSA would have modified Ophir to implement Takeda's ciphering/de-ciphering method in Ophir's mediator for at least three independent reasons: (1) POSAs would have expected the modification to increase Ophir's encryption/decryption reliability; (2) the modification would have represented a simple substitution of a known method and element for another method and element and would have yielded predictable results; and (3) the modification would have represented the use of a known technique to improve a similar device in the same way.

a. Increased Reliability

82. In my opinion, a POSA would have been motivated by Takeda to implement Takeda's ciphering/deciphering as the encryption/decryption performed by Ophir's encryption/decryption unit, for example the encryption/decryption unit 517, to achieve "a more reliable ciphering technique" as taught by Takeda. Takeda, [0006], [0009]. They would have expected that the implementation would have successfully achieved a more reliable ciphering technique because by using Takeda's technique "even if a part of the data is deciphered, the rest of the data is not easily deciphered." Takeda, [0006], [0009].

83. As I explained above in §VII.A, Ophir's mediator, such as mediator 401 in FIG. 4 and mediator 501 in FIG. 5, encrypts data to be stored and decrypts stored data that is to be sent to a data client. Ophir, FIG. 5, [0010]-[0011], [0013], [0032]. Ophir provides the example "Encryption Scenarios" I noted above, but as I also noted above Ophir states that the encryption and decryption performed by the mediator are not limited to the examples Ophir provides. Ophir, [0036]. Those examples are "non-limiting examples provided for illustration" because Ophir explains that "other scenarios are also possible within the framework of the invention." Ophir, [0036].

84. In my opinion, a POSA would have recognized, based on at least what Takeda explains, that not using different encryption attributes for different parts of

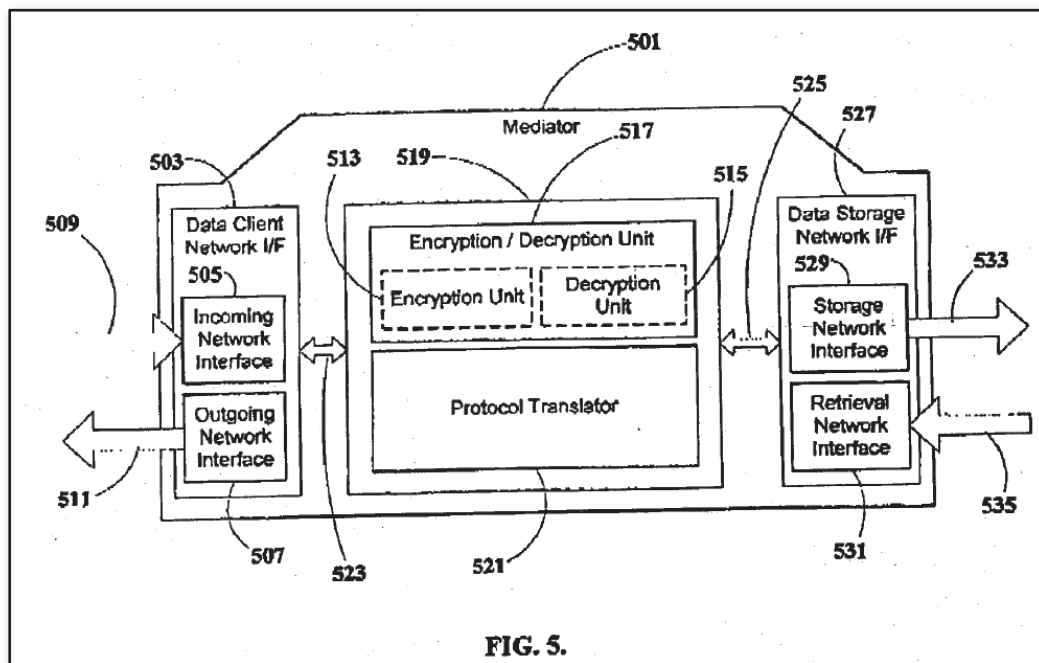
a data set in Ophir is risky. Takeda, [0004]. Takeda explains in connection with using the same ciphering for all the data blocks that “when a part of the ciphered data is deciphered, there is a risk or chance that all data will be deciphered.”

Takeda, [0004]. As I explained above, the risk when using the same ciphering for all the data blocks is that compromise of a part of the data may lead to compromise of all data. For example, an attacker need only determine one decryption technique to gain access to Ophir’s full data set because the data is not divided and different encryption techniques used on different portions. Therefore, if any part of Ophir’s encrypted data is undesirably decrypted, then all the encrypted data can be decrypted. Takeda, [0004]. Given that, it is my opinion that a POSA would have been motivated by Takeda to implement Takeda’s ciphering/deciphering technique (splitting the data into blocks and encrypting different blocks with different ciphering attributes) in Ophir to achieve a more reliable encrypting/decrypting technique that is less susceptible to unauthorized access than what Ophir alone describes. Takeda, [0006], [0009].

b. Simple Substitution

85. In my opinion, as I explain further below, implementing Takeda’s ciphering/deciphering as the encryption/decryption performed by Ophir’s encryption/decryption unit is merely a simple substitution of one known method and element for another, and it would have produced predictable results.

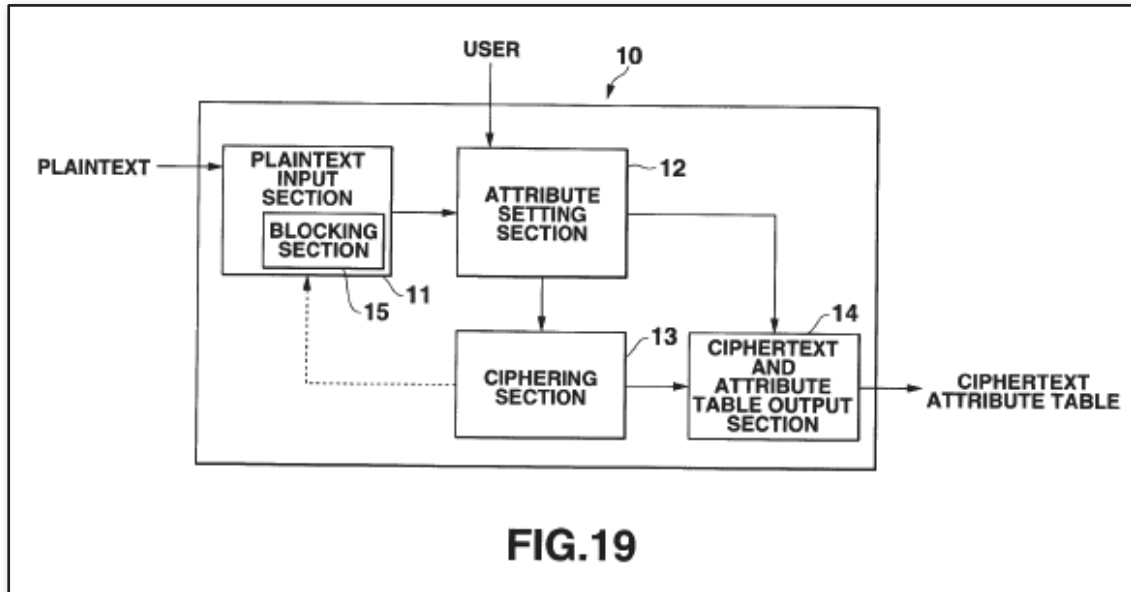
86. As I noted above in §VII.A, Ophir's mediator, for example mediator 501 which I have reproduced below (FIG. 5), includes data storage processor 519 that has encryption/decryption unit 517, and that encryption/decryption unit 517 can be optionally implemented with an encryption unit 513 and decryption unit 515. Ophir, [0030], [0032]. Ophir says "Encryption/decryption unit 517 encrypts data from the data clients into encrypted data for safe storage in data storage devices, and decrypts data retrieved from data storage devices into decrypted data for sending to data clients." Ophir, [0032]. Thus, Ophir's mediator, such as mediator 401 in FIG. 4 and mediator 501 in FIG. 5, encrypts the data before it is stored and then decrypts it to send to a data client after it is retrieved from one of the data storage devices. Ophir, FIG. 5, [0010]-[0011], [0013], [0032].



87. The encryption/decryption unit 517 can have an encryption unit 513 that operates to “encrypt[] data from the data clients into encrypted data for safe storage in data storage devices.” Ophir, [0032]. The encryption/decryption unit 517 can also have a decryption unit 515 that operates to “decrypt[] data retrieved from data storage devices into decrypted data for sending to data clients.” Ophir, [0032].

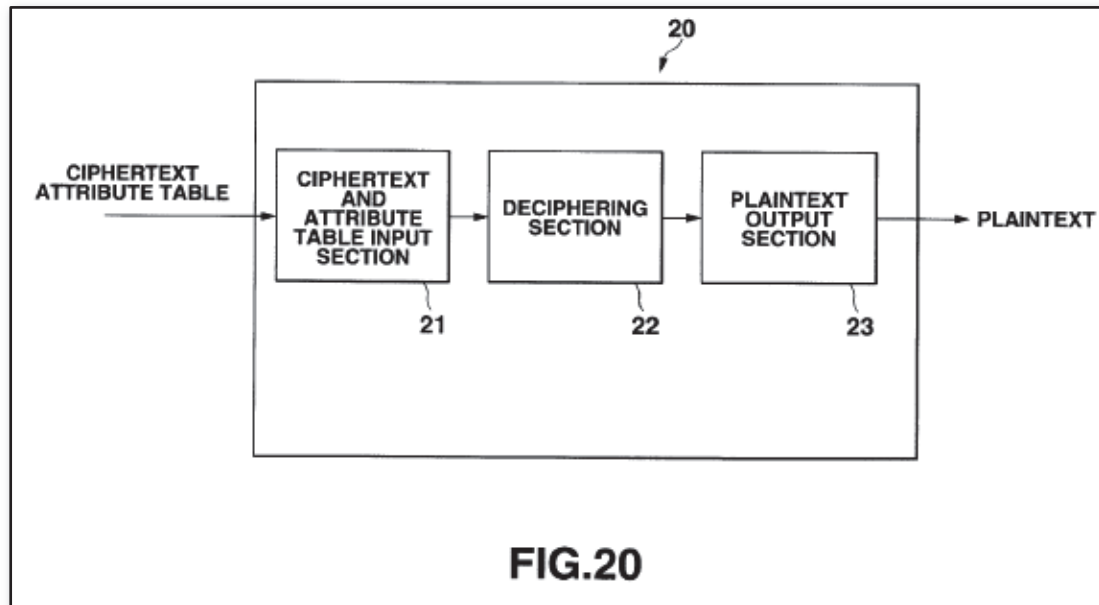
88. Takeda’s FIG. 19, which I reproduce below, illustrates a ciphering apparatus 10. The ciphering apparatus 10 includes the plaintext input section 11, blocking section 15, attribute setting section 12, ciphering section 13, and “ciphertext and attribute table output section” 14. Takeda, [0031], [0087]-[0088]. The ciphering apparatus 10 receives the “PLAINTEXT” as input to the plaintext input section. [0088]. The user can set the attributes for ciphering using the attribute setting section 12, but a POSA would also understand that the attributes could be set in another way not by user since Takeda says that user setting of the

attributes is an example. [0056], [0088]. The ciphering apparatus 10 outputs the “CIPHERTEXT” and “ATTRIBUTE TABLE.” [0088].



89. FIG. 20 illustrates a deciphering apparatus 20, which includes “ciphertext and attribute table input section” 21 that receives the ciphertext and attribute table, deciphering section 22 that decipheres the ciphertext, and plaintext

output section 23 that outputs plaintext resulting from deciphering of the ciphertext. Takeda, [0032], [0090].



90. In my opinion, implementing Takeda's ciphering/deciphering technique in Ophir's encryption/decryption unit 517 is a simple substitution of Takeda's known encryption/decryption technique for Ophir's encryption/decryption technique. This can be done in various ways. For instance, it could be done by substituting the ciphering apparatus 10 of Takeda's FIG. 19 for Ophir's encryption unit 513 and substituting the deciphering apparatus 20 of Takeda's FIG. 20 for Ophir's decryption unit 515. Alternatively, it could be done by adapting/programming Ophir's encryption/decryption units to perform Takeda's techniques. For example, Takeda's ciphering apparatus 10 and Ophir's encryption unit 513 both take in data to be ciphered (i.e., encrypted) and output ciphered data.

Takeda, [0088]; Ophir, [0032]. Devices and methods doing that were known in the art. For example, U.S. Pat. 7,571,320 (EX1010) provides a ciphering engine that produces encrypted data as an output. EX1010, 1:35-41. Another document, EP 1 063 811 (EX1011), explains how cryptographic algorithms are used to encrypt data. EX1011, [0002]. That document also refers to “cryptographic algorithms” as ciphers.”. Similarly, Takeda’s deciphering apparatus 20 and Ophir’s decryption unit 515 both take in ciphered data and output deciphered (i.e., decrypted) data. Takeda, [0090] (Takeda’s “ciphertext and attribute table input section” 21 receives “ciphertext” which is the ciphered data along with the attribute table and outputs “plaintext” which is deciphered data).

91. In my opinion, substituting Takeda’s ciphering/deciphering techniques and/or components for Ophir’s encryption/decryption techniques and/or components would have predictably resulted in Ophir’s data ciphering/deciphering (i.e., encrypting/decrypting) being performed as described by Takeda.

c. Use of Known Technique to Improve Similar Devices in the Same Way

92. In my opinion, a POSA would also have recognized that Ophir’s mediator was a candidate for improvement using Takeda’s known ciphering/deciphering technique. Both Ophir and Takeda relate to methods and systems for securely storing data by using ciphering/deciphering (i.e., encryption/decryption) techniques. Takeda, Abstract; Ophir, Abstract. Ophir’s

mediator effects secure storage of clients' data on storage devices by encrypting data to be stored on the storage devices and decrypting data retrieved from the storage devices before providing the data to the clients. Ophir, Abstract, [0010]-[0011], [0013], [0015], [0032], FIGs. 4-5. Takeda's improved ciphering/deciphering techniques were a known technique to improve security and they were directly applicable to Ophir's system. Takeda, [0005]-[0009]. A POSA would have recognized that applying Takeda's ciphering/deciphering in the Encryption/Decryption Unit (FIG. 5) of Ophir's mediator would have improved Ophir's secure storage system in the same way those techniques improved Takeda's system, by making the protected data more secure since "even if a part of the data is deciphered, the rest of the data is not easily deciphered." Takeda, [0006].

2. Reasonable Expectation of Success

93. A POSA would have had a reasonable expectation of success in modifying the encryption/decryption unit of Ophir's mediator to implement Takeda's ciphering/de-ciphering techniques and/or components, for the reasons I explained above in §§VI.C.1. Takeda provides sufficient detail about its ciphering/deciphering techniques to allow a POSA to implement them in Ophir. Also, a POSA would have understood that Takeda's ciphering/deciphering techniques were compatible with Ophir's system. For example, techniques

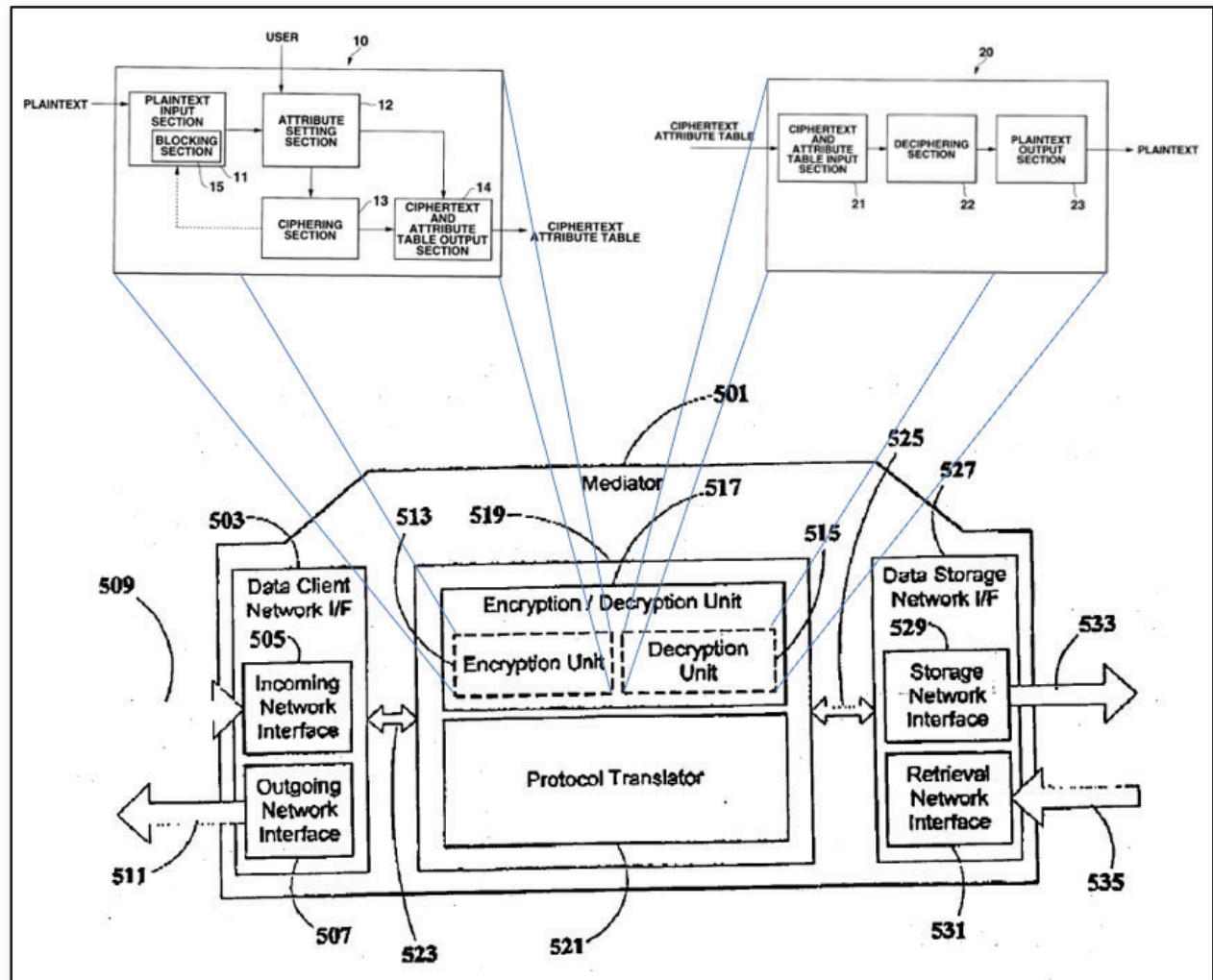
performed by Takeda's ciphering apparatus 10 can be implemented in Ophir's encryption unit 513 as both intake data to be ciphered (i.e., encrypted) and output ciphered data, while techniques performed by Takeda's deciphering apparatus 20 can be implemented in Ophir's decryption unit 515 as both intake ciphered data and output deciphered (i.e., decrypted) data. Takeda, [0088], [0090].

94. Therefore, it is my opinion that a POSA would have reasonably expected that implementing Ophir's encryption unit 513 and decryption unit 515 to use Takeda's ciphering apparatus 10 and deciphering apparatus 20, respectively, (or their techniques) would have successfully resulted in Ophir's mediator performing Takeda's ciphering/deciphering techniques and otherwise operating as Ophir describes.

3. Resulting Combination

95. A POSA would have implemented Takeda's ciphering/deciphering method in Ophir's mediator (mediator 401, 501, 601, 707) to replace Ophir's encryption/decryption performed in the "Encryption/Decryption Unit." For instance, I've included a figure below illustrating what would result from implementing the functionality of Takeda's ciphering apparatus 10 and deciphering apparatus 20 in Ophir's encryption unit 513 and decryption unit 515, respectively. Ophir's encryption unit 513 will perform Takeda's ciphering method and Ophir's

decryption unit 515 will perform Takeda's deciphering method. I call the resulting system and method the Ophir-Takeda Combination or simply "Ophir-Takeda."



Result of Modifying Ophir Based on Takeda

96. The Ophir-Takeda Combination modifies Ophir's encryption/decryption techniques to use those of Takeda. Otherwise, Ophir-Takeda does not need to change the other features of Ophir. In operation, a data block input to Ophir's block services 623, or a file input as plaintext to Ophir's file services 625, is divided (further divided in the case of blocks input to block

services 623) into blocks to be ciphered using Takeda's ciphering method (e.g., Takeda FIG. 1B, blocks M1-M4), ciphered (i.e., encrypted), and output as ciphered blocks for storage on any of Ophir's storage devices (see Ophir, FIGs. 4, 6-7). Each ciphered block contains its respective ciphering attributes, such as the attributes shown in FIG. 4 of Takeda (or FIG. 12 of Takeda), or a pointer to the ciphering attributes as explained in Takeda. Takeda, [0040], [0048].

D. Ophir-Takeda Meets All Limitations of Claims 1-6, 8, 10

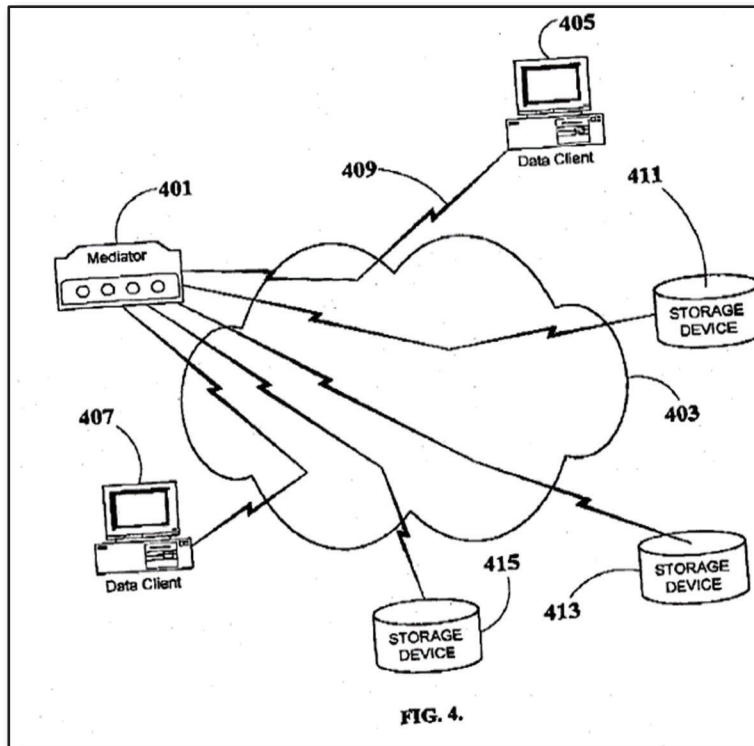
1. Claim 1

a. 1[PRE]: A secure storage network comprising:

97. As I explained above in §VII.C.3, Ophir-Takeda includes Ophir's data clients, mediator, and storage devices. Those components together represent a secure storage network. Ophir, FIGs. 4, 6-7, Abstract, [0001]. A POSA understood that the claim term "network" appearing in the '140 patent encompasses a set of devices that are connected by a communication network. For example, Ophir's devices including data clients, mediator, and physical storage devices, and communication network is Ophir's network 403 which connects those devices. This is consistent with the definition of network as "a group of computer and associated devices that are connected by communications facilities." EX1043, 362. This is also consistent with the recitation in challenged claim 1 of a "network comprising: a plurality of physical storage devices...and a secure storage system" (EX1001, 98:2-7) and the '140 patent's description that those devices

communicate with each other “through any [communication] network” (EX1001, 97:41-51).

98. As an example, I reproduce Ophir’s FIG. 4 below, which illustrates data clients 405 and 407, mediator 401, physical storage devices 411, 413, 415, and network 403 that together form a secure storage network. Ophir, Abstract, [0011], [0015], FIGs. 6-7. As Ophir notes in its Abstract, “The mediator connects over the *network* to one or more data clients and to one or more data storage devices, and provides *secure storage* of data for the data clients on the data storage devices.” Moreover, in connection with FIG. 6, Ophir states, “**FIG. 6** illustrates the capacity of a mediator 601 to effort [*sic*] *secure virtual data storage* for a data client 603 *over a network* connection 605.” [0033]. And Ophir provides networking technologies and protocols, such as “Network File System” or “NFS” for communication between the devices. Ophir, [0034]; *see also* FIG. 7.



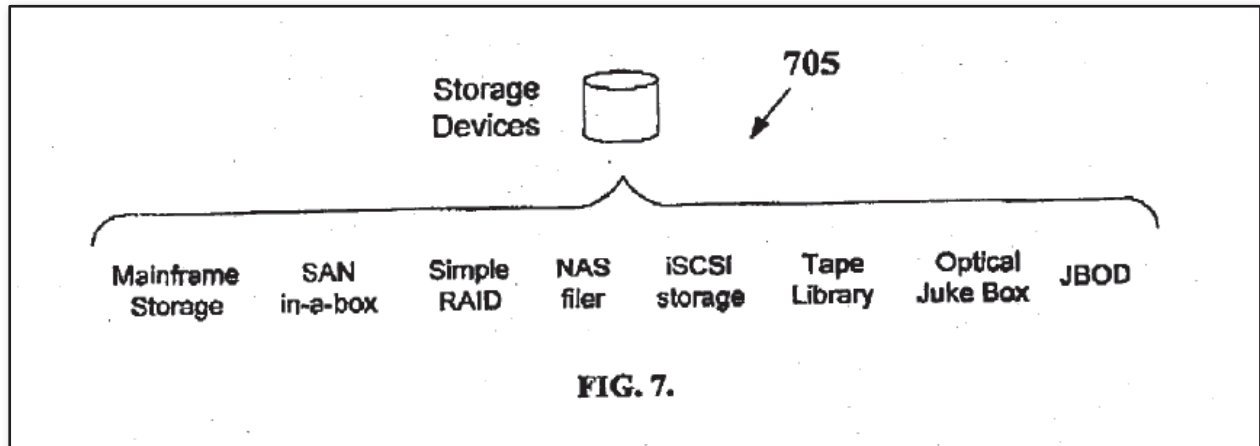
b. [1A]:

99. I break limitation [1A] into two pieces and address them in the following two subsections.

i. a plurality of physical storage devices

100. I've reproduced FIG. 7 of Ophir below. That figure "illustrates some representative and non-limiting client services and protocols, networks, and storage device technologies supported by a configuration according to the present invention." Ophir, [0025]. Ophir says that the "Storage devices 705 encompass various devices known in the art, including, but not limited to: mainframe storage; SAN-in-a-box; simple RAID; NAS filer; iSCSI storage; tape library; optical juke box; and JBOD ('Just a Bunch Of Disks'), which herein denotes any collection of

one or more disk drives which does not necessarily include any special coordinating controller or data processing.” Ophir, [0034]. “SAN” in “SAN-in-a-box” stands for “storage area network.” “NAS” in “NAS filer” stands for “network attached storage.”



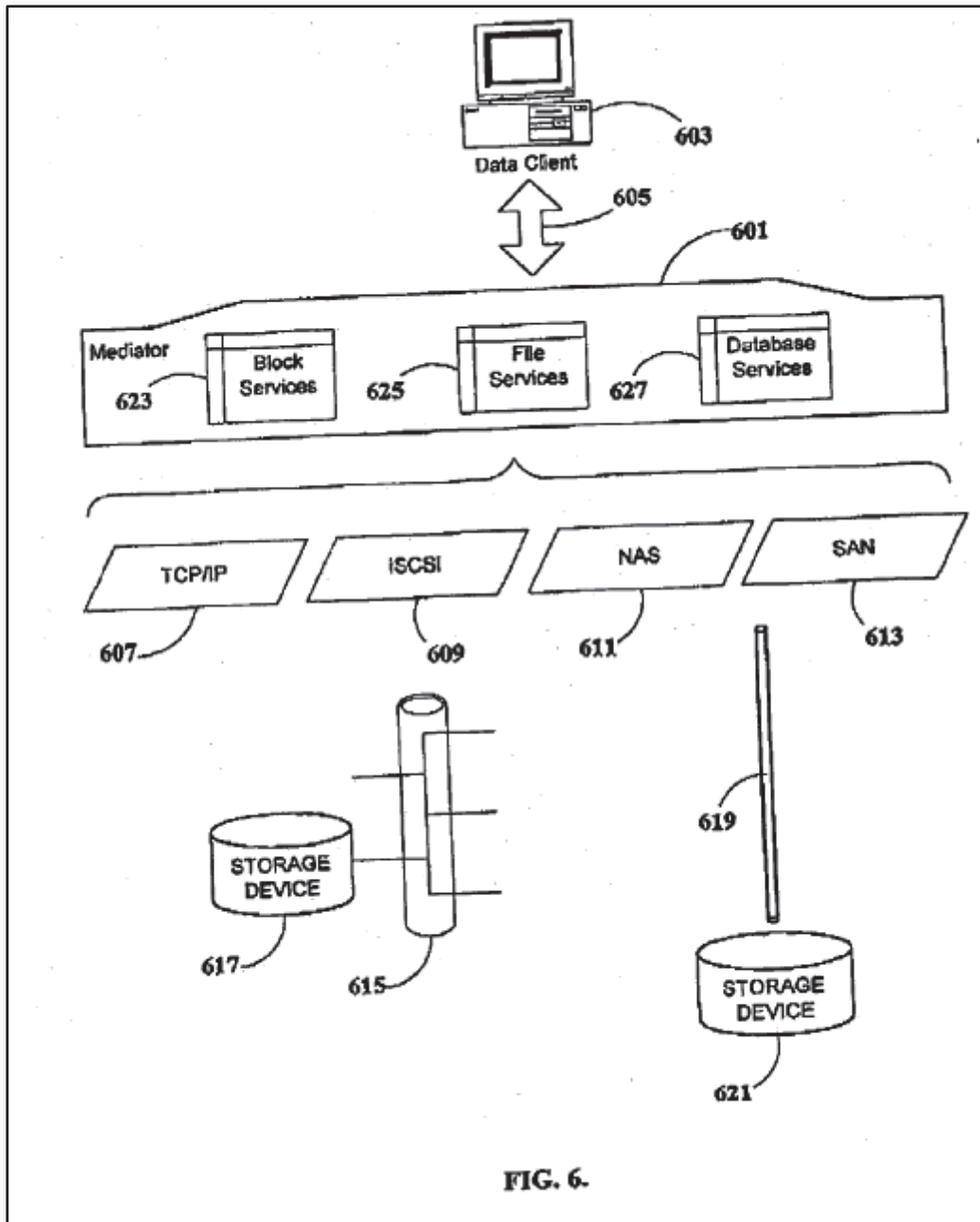
101. POSAs understood these are all examples of physical storage devices. Also, POSAs would understand that at least some of these examples of physical storage devices by their nature include a plurality of devices, such as JBOD (“Just a Bunch Of Disks”) and RAID. “RAID” is a known term in the field which is written out as “redundant array of independent disks” or sometimes as “redundant array of inexpensive disks.” Irrespective of that minor variation in wording that sometimes appears in the art, POSAs understand that RAID by its nature operates by having multiple disks with data stored on the multiple disks, as indicated by the reference to “redundant” and “disks” in the name.

102. For example, U.S. Patent 6,151,659 (EX1012) is titled “Distributed RAID Storage System” and “relates to a redundant array of independent disks (RAID) system in which multiple host operating systems share access to the disk drives.” EX1012, 1:4-6. The patent discloses a data storage system that has at least three data storage disks. EX1012, Abstract. The patent also notes that “A common RAID configuration is several hard disk drives, where data is written in a "stripe" across the drives, consisting of corresponding data sectors on each drive, in which one of the data sectors stores parity data for the other data sectors in the stripe.” EX1012, 1:13-17.

103. U.S. Patent 5,526,507 (EX1013) refers to RAID as “Redundant Arrays of Inexpensive Disks.” EX1013, 1:57-63. The patent explains how it was known to provide disk drives in an array. EX1013, 1:47-63.

104. U.S. Patent 6,065,096 (EX1014) refers to RAID as an array of inexpensive disk drives. EX1014, 1:13-18. The patent explains there are various forms of RAID, referred to as RAID “levels.” EX1014, 1:19-40. Simple “RAID 1” mirrors the contents from one disk drive onto another disk drive. EX1014, 1:25-27. The reference to mirroring means that the same information is stored on both disk drives. EX1014, 1:25-27.

105. FIG. 6 of Ophir also illustrates a plurality of physical storage devices, labeled as 617 and 621. Ophir, [0024], [0033]. I’ve reproduced that figure below.

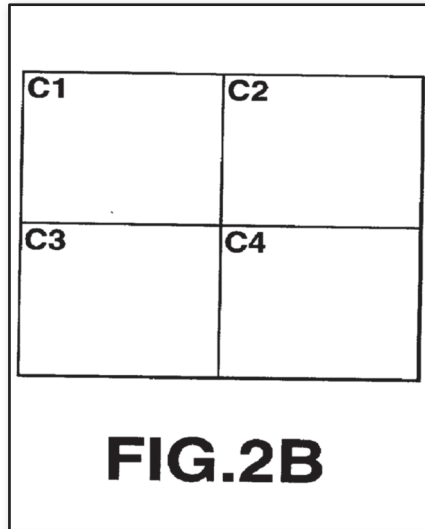
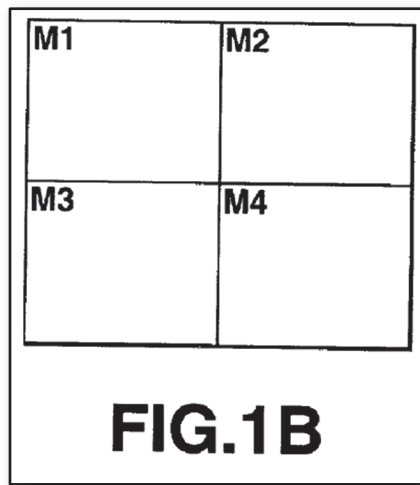


- ii. storing thereon a plurality of shares, the plurality of shares being associated with at least one session key used to secure a dataset

106. Ophir-Takeda implements Takeda's ciphering technique, which divides received data into a plurality of blocks M_n . See above in §§VII.B, VII.C.3;

Takeda, FIG. 1B (illustrating four blocks M1-M4), FIG. 11B, [0038]. The received data in Ophir-Takeda is the data from one of Ophir's data clients. Ophir explains that the data is a "storage object (e.g., logical unit[], file[], director[y])" for storage. Ophir, [0051], [0040]. I understand the parties have offered different constructions of the term "dataset" appearing in the claims of the '140 patent. POSAs understood that an object of the type given in Ophir (e.g., file, directory) is a "dataset" under both parties' constructions because it is a "collection of related information made up of separate elements [e.g., bits of a file, files in a directory] that can be treated as a unit [an "object (e.g., logical unit...)" (Ophir, [0051])] in data handling" (Google: EX1041, 9-10) and it is "a collection of information for storage" (SFI: EX1042, 6-7).

107. In Ophir-Takeda, each data block Mn is ciphered using a respective cipher attribute An (which may include a ciphering key—Takeda, [0042]) to form a ciphered block Cn. Takeda, FIGs. 2B, 11B-11D, [0040]-[0042], [0068]; §§VII.B, VII.C.3 *supra*.



108. Each ciphered block (Cn) is a “share” as claimed because it is a portion of the ciphered (i.e., encrypted) dataset, which the ’140 patent’s specification calls a “share.” EX1001, 53:19-28 (equating “share” and “portion”), 55:59-66 (discussing “storing the shares” after each “portion[]” is encrypted); 56:25-27 (“[s]eparating...encrypted data” results in “shares or portions of parsed data”), 56:60-62, 57:36-38, 58:11-13; *see also* EX1001, 59:15-18 (describing encrypted data “segments” as “shares”). Ciphertext block (Cn) together with its corresponding key (“ciphering attribute” An) (Takeda, FIG. 3A) meet a “share” as claimed if the share is construed to require storing information linking the share to the key that is used to secure the share. Each ciphertext block (Cn) together with a pointer to its corresponding key (“ciphering attribute” An) is a “share” as claimed if a “share” is construed to require storing information linking the share to the key used to secure it. (Takeda, [0040]).

109. The plurality of ciphered blocks (“shares”) are associated with at least one session key (attributes A1-An) used to secure the dataset, because attributes A1-An are used to cipher (i.e., encrypt) the dataset to create ciphertext blocks C1-Cn. Takeda, FIGs. 4 and 12, [0048]. The ’140 patent’s specification describes “encrypt[ing]” a dataset as “securing” the dataset. EX1001, 60:52-64, 61:28-30, 64:56-59. The specification of the ’140 patent is clear that a key used to encrypt a share is “associated with” that share as claimed. EX1001, 77:11-13, 77:31-33; 77:38-41. Similarly, Takeda’s attribute An is used to cipher (encrypt) a corresponding plaintext block Mn to create a share Cn, and therefore is associated with the share Cn.

110. I understand the parties have offered different constructions of the term “session key” appearing in the claims of the ’140 patent. The ciphering key of each attribute An is a “session key” under both parties’ constructions because it is a “[k]ey generated each time data is secured” (Google: EX1041, 13) and “a key created to both secure and restore a specific collection of information” (SFI: EX1042, 10). Takeda, [0042] (“attributes A1 to A4 are, for example, ciphering keys”), FIGs. 5, 7, 13, 15. Each ciphering key An is generated each time data is secured and is created to secure and restore a respective block of plaintext Mn. Takeda, FIGs. 2A, 4, 5 step S10, [0056] (attribute is prepared for ciphering specific plaintext), FIGs. 7, 13, 15.

111. Ophir-Takeda stores the plurality of shares (whether just the ciphered blocks (C_n) or the ciphered blocks together with information linking to the corresponding ciphering keys) on the system's plurality of storage devices. §§VII.B, VII.C.3, *supra*; Ophir, FIGs. 4, 6-7, [0030], [0032], [0034]. As I explained earlier *supra* §VII.C.3, in Ophir-Takeda, Takeda's ciphering is performed in Ophir's "Encryption/Decryption Unit" 517 to at least support Ophir's block services 623 and file services 625 (FIG. 6). Ophir, FIGs. 4, 6-7, [0030], [0032], [0034]; Takeda, [0040]. The encrypted shares are stored on the above-discussed plurality of storage devices such as Simple RAID and/or JBOD storage devices. Ophir, FIGs. 4, 6-7, [0030], [0032], [0034].

c. [1B]: a secure storage system configured to:

112. Ophir-Takeda includes Ophir's mediator, which is a secure storage system as claimed. Ophir, FIGs. 4-5, Abstract (the mediator provides "secure virtual storage" to data clients). Based on the way claim 1 is written, a POSA would understand that the claimed "secure storage system" does not need to include the physical storage devices on which data is stored, because claim 1 recites the secure storage system (limitation [1B]) as distinct from the "plurality of physical storage devices" (limitation [1A]). Also, the functions claim 1 gives to the "secure storage system" are not described in the '140 patent as being performed by the physical storage devices. Ophir's mediator in Ophir-Takeda, like the data

parser in the '140 patent, is configured to perform the functions that claim 1 recites for the secure storage system as I explain below.

- i. **[1B-1]: present to a client device a virtual disk, the virtual disk comprising a directory mapped to the plurality of physical storage devices such that physical locations of the shares are hidden from the client device;**

113. I note that the term “virtual disk” does not appear in the '140 patent's specification. I also note the parties give different constructions for the term “virtual disk.” See §VI (above); EX1041, 10-11; EX0142, 8. Ophir-Takeda uses Ophir's mediator, which meets the requirements of claim limitation [1B-1] at least under SFI's construction of “virtual disk,” which is “a system that simulates or emulates a physical storage device” (EX1042, 8).

114. Ophir states in paragraph [0033]:

FIG. 6 illustrates the capacity of a *mediator* 601 to effort [*sic*] *secure virtual data storage* for a data client 603 over a network connection 605. The *storage is considered “virtual”* because the data from data client 603 can be stored on a variety of *storage devices* using a variety of protocols, technologies, and services, as managed by mediator 601...

115. Ophir explains in paragraph [0013] that, because “the mediator is able to store data in, and retrieve data from, any suitable storage device having access to the network”:

[i]n [this] manner, the mediator functions as a central coordinator for data storage between...clients requesting data storage and...storage devices providing data storage. In this central point, the mediator serves as a *virtual secure storage device*. The data clients do not have to be involved in any storage or retrieval operation with any storage devices, and *need not know the locations where the data is stored*.

116. Ophir also states that “data sent to the mediator for storage by a particular data client can be retrieved by the mediator from storage and sent back to that same data client.” Ophir, [0012].

117. POSAs would have understood that the above disclosures teach that Ophir’s mediator system simulates or emulates a physical storage device, which meets SFI’s proposed construction of “virtual disk,” because the client can make storage requests to the mediator which handles storing and retrieving the data from the actual physical storage devices, such as the storage devices in Ophir’s FIG. 7. Because the client can retrieve stored data through the mediator without knowing the data’s stored location (Ophir, [0012]-[0013]) (i.e., “physical locations of the shares are hidden from the client device” as claimed (EX1001, 98:10-11)), POSAs would have understood that Ophir maintains a directory that maps where in the physical storage locations each piece of client data is stored, so that the mediator can retrieve the data when the client desires. Alternatively, maintaining a directory of the type just explained, that maps to the physical storage devices, would have

been a conventional and obvious way to implement what Ophir describes, because it would allow the mediator to retrieve client data for the data clients when the client does “not know the locations where the data is stored.” Ophir, [0013].

118. For example, U.S. Patent 8,352,518 (EX1016) is titled “Mechanism For Handling File Level and Block Level Remote File Accesses Using the Same Server.” That patent provides an example of a set of file systems, referred to as “volumes” which contains a directory of so-called “i-nodes” that represent data blocks and that point to the location of a data block in physical storage by providing the address. EX1016, 5:39-6:16. A mapping from a volume ID to the i-node file for that volume can be provided. EX1016, 5:59-62.

119. Also, U.S. Patent 6,065,096 (EX1014) provides an example of a RAID controller and notes that “In general, RAID subsystems comprise an array of inexpensive disk drives operated by a RAID controller which presents the array to attached host systems as though operating as a single large, fast, reliable single disk drive. The RAID controller maps requests from attached host systems into appropriate requests to the array of disk drives.” EX1014, 1:13-18. A POSA would understand that this means that because the multiple physical drives are presented to the host system as if they were a single disk drive, the controller will include a mapping to the physical disk drives so that, for example, when the host system requests data the data can be located on the physical disk drives.

120. Additionally, Ophir states at paragraph [0033]:

Moreover, *mediator 601 is also able to provide block services 623, file services 625*, and database services 627..., while providing protocol translation between application protocols used with clients and application protocols used for storage devices and encrypting and decrypting the data that is stored on the storage devices.

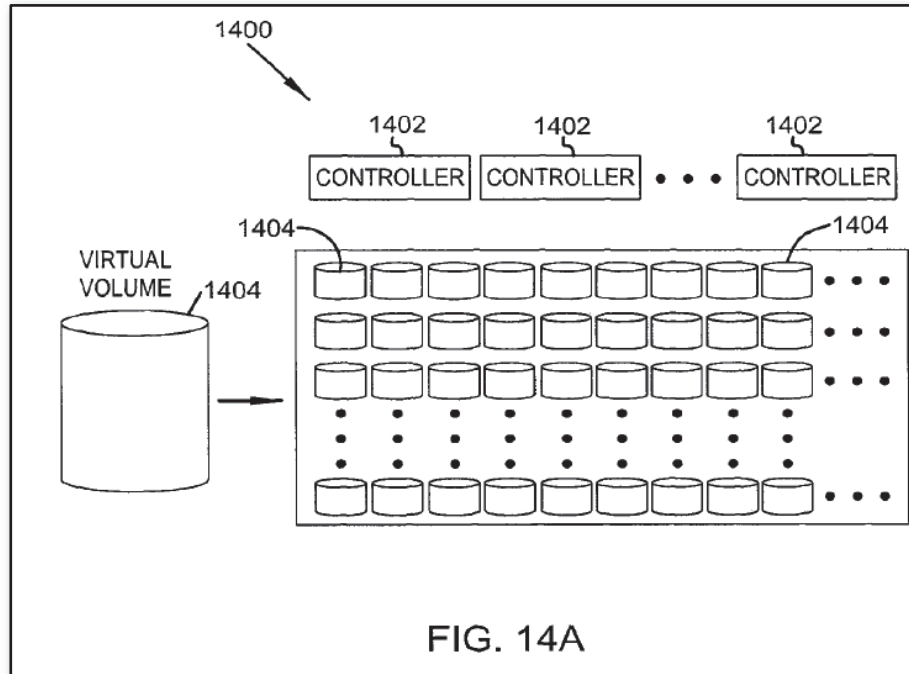
121. Ophir’s “block services” also meet claim limitation [1B-1] under SFI’s construction of “virtual disk.” And, Ophir’s “file services” meet claim limitation [1B-1] under both parties’ constructions of “virtual disk.” I explain why below.

(1) Block Services

122. Ophir’s mediator effects “secure virtual data storage,” and can do so by providing “block services” (Ophir, [0033]), which POSAs would have understood means that the mediator provides block-based virtual storage. POSAs knew that block-based storage addresses and operates on data in blocks, and that in a “virtual data storage” system like Ophir’s, the blocks present virtual disks. For example, see the following references.

123. U.S. Patent 7,941,695 (EX1015) is titled “Virtual Disk Drive System and Method.” I reproduce FIG. 14A below, which “illustrates one embodiment of a disk drive system having a virtual volume matrix of disk storage blocks for

dynamically allocating data in the system in accordance with the principles of the present invention.” EX1015, 4:42-45.



124. The figure shows a matrix of disk storage blocks or virtual volumes 1404. EX1015, 8:55-9:3. The data is stored by the controllers in the disk storage blocks and retrieved from the disk storage blocks. EX1015, 8:55-9:3. The virtual volumes are a virtual disk device that is presented to servers and that provides a storage abstraction of multiple physical storage devices (RAID devices). EX1015, 7:32-40.

125. U.S. Patent 8,352,518 (EX1016) is titled “Mechanism for Handling File Level and Block Level Remote File Accesses Using the Same Server.” That patent refers extensively to data blocks and managing them. EX1016, 1:16-21,

2:57-3:3. The patent also explains that the blocks of a file can represent a virtual disk, stating “A SAN volume contains a single file whose blocks represent a virtual disk.” EX1016, 5:44-45.

126. U.S. Patent Application Publication 2002/0174295 (EX1017) provides an enhanced file system that includes a plurality of disk drives that store storage blocks that include data blocks. EX1017, Abstract. The reference explains that “In the RAID architecture, multiple disks a[re] typically mapped to a single ‘virtual disk’”. EX1017, [0020]. A POSA would understand the “multiple disks” are physical disks per a general understanding of RAID. The reference further says that ([0020]):

“Consecutive blocks of the virtual disk are mapped by a strictly defined algorithm to a set of physical disks with no file level awareness. When the RAID system is used to host a conventional file system, it is the file system that maps files to the virtual disk blocks where they may be mapped in a sequential or non-sequential order in a RAID stripe.”

Moreover, the file system can include a directory structure. EX1017, [0033], [0044].

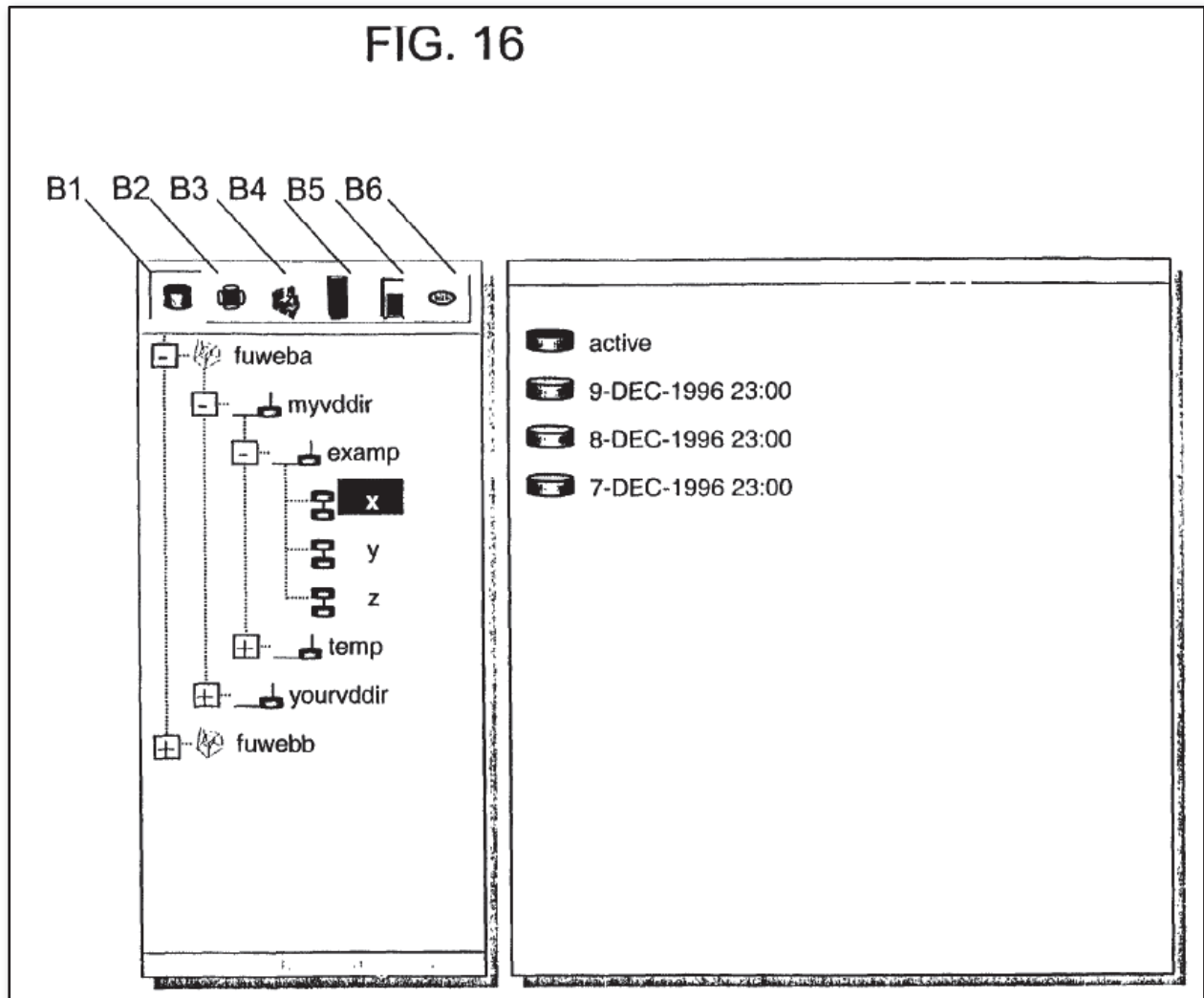
127. This means that “virtual disk blocks” are mapped to a file. EX1017, [0020].

128. U.S. Patent Application Publication 2002/0019908 (EX1018) is titled “System and Method for Managing Virtual Storage.” That reference explains that “The virtual disk object 150’ represents a block-store disk as seen by a host system. It is independent of physical storage and is a logical object that contains the data that the system stores on behalf of host systems.” EX1018, [0045]. The virtual disk can include data blocks to which data can be written and from which data can be read. EX1018, [0046]-[0048].

129. That same reference also explains how a mapping is provided from the virtual disk to the physical storage ([0029]):

In some non-limiting preferred embodiments, the present invention can be applied in a virtualized storage area network (SAN) system 100 using one or more distributed mapping tables, as needed to form one or more virtual disks for input/output (I/O) operations between hosts and storage containers 160, as illustrated in FIG. 1. In particular, the table contains a mapping that relates a position in a virtual disk 150 with an actual location on the storage containers 160.

130. FIG. 16 of the reference even shows how the virtual disk might be presented to a user. EX1018, [0112]. I have reproduced that figure below. It can be seen that the view presented is that of a directory. See EX1018, [0073].



131. U.S. Patent No. 7,043,614 (EX1019) is titled “Storage Services and Systems” and explains the use of virtual disks. EX1019, Abstract. That reference says that “Storage virtualization conceals the physical storage details from

user/client applications that access the physical storage.” EX1019, 1:39-50. It also says:

“Typically, when providing storage virtualization, storage management techniques will interject a middle layer of software that is interposed between user applications and the interfaces of the underlying physical storage devices. This middle layer of software is capable of centrally managing the physical storage devices and creating a virtual view of the storage data. In some cases the middle layer of software is embodied as a Volume Manager (VM) application.” EX1019, 2:13-24.

It would have been typical to implement a virtual view of this type as a directory.

132. Given this background knowledge and Ophir’s disclosure that the mediator provides “virtual” storage using “a variety of storage devices” and that data clients need not know the physical location of the stored data (Ophir, [0033], [0013]), POSAs would have understood that storage blocks presented to a data client are blocks of a virtual (rather than physical) disk, wherein the virtual disk comprises a directory mapped to the plurality of physical storage devices. EX1016, 5:39-58; EX1017, [0020]; EX1018, [0029], FIG. 16. In other words, the client data stored in a block of a virtual disk is physically stored in one or more physical “locations” on one or more of Ophir’s physical storage devices (Ophir, [0013], [0033]-[0034]), and the virtual disk maintains a directory mapping each virtual

block to the physical locations where its data is stored so that the data can be written to and read from the correct physical storage locations. *Id.* The mediator's block services thus present to a client device a "virtual disk" under SFI's construction as "a system that simulates or emulates a physical storage device." EX1042, 8.

133. To the extent that Ophir is considered to not expressly disclose the use of a virtual disk to support Ophir's block services, or implementing a virtual disk by using a mapping directory, that would have been the conventional and obvious way to implement what Ophir describes – i.e., that the mediator provides "virtual" storage, data is accessed by a client on a block level, and the client doesn't need to know where the data is physically stored. Ophir, [0013], [0033]; *see also* EX1019, 1:39-50 ("Storage virtualization conceals the physical storage details from user/client applications that access the physical storage."), 2:13-24 ("Typically, when providing storage virtualization," a "middle layer" will "creat[e] a virtual view of the storage data"); EX1015, FIG. 14A, 7:32-40, 8:55-9:3; EX1016, 2:57-3:3, 5:39-58; EX1017, [0020]; EX1018, [0029], [0045]-[0048]. Virtual disks, at least under SFI's proposed construction of that term, were a well-known technique for providing virtual block-level storage services to clients that need not know where the data is physically stored, and such virtual disks were known to use a

mapping directory to map virtual storage blocks to physical storage locations, as the above-cited references corroborate. Ophir, [0013], [0033].

134. As I explained above, as just one example, U.S. Patent 8,352,518 (EX1016) provides an example of a set of file systems, referred to as “volumes” which contains a directory of so-called “i-nodes” that represent data blocks and that point to the location of a data block in physical storage by providing the address. 5:39-6:16. A mapping from a volume ID to the i-node file for that volume can be provided. 5:59-62.

(2) File Services

135. I understand that in the co-pending litigation Google proposes construing “virtual disk comprising a directory mapped to the plurality of physical storage devices” as “a distributed file system shared among file servers that includes a directory with a clients’ view of all file assets mapped to the physical storage devices on which they are stored.” EX1041, 10.

136. POSAs would have understood that the file services 625 of Ophir’s mediator presents a file system to the data clients, because Ophir explains that (1) the storage objects can be files and (2) file services 625 communicates using file system application protocols, such as NFS (“network file system”) and CIFS (“common internet file system”). Ophir, [0025], [0033]-[0034], [0051], FIGs. 6-7; EX1020, 1:21-27 (“NFS” means “Network File System”), 3:15-28 (“CIFS” means

“Common Internet File System). The storage objects, including the files, could be any dimensionality, including two-dimensional or linear. Ophir, [0025], [0033]-[0034], [0051]. Like the ’140 patent’s parsed file system, Ophir’s file system provides virtual storage to the data client that hides from the client the physical location(s) where the file’s data is stored. Ophir, [0033], [0034]. POSAs would have understood that Ophir’s file system is a distributed file system shared among file servers because Ophir explains that the mediator’s data processing is “highly distributed” and can be performed by a number of “master devices” (Ophir, [0029], [0022], FIG. 4), which in the context of file services would be understood to be file servers.

137. A POSA would have understood that Ophir’s file system comprises a directory with a client’s view of all the client’s file assets, since the client device is able to retrieve the files which they have stored via the mediator. Ophir, [0046]-[0047].

138. A POSA would further have understood that the directory of Ophir’s file system maps each file to the physical storage locations where the file’s data is physically stored because that is how Ophir’s mediator can retrieve stored data from the physical storage devices and, moreover, that is how file systems were known to work. Ophir, [0046]-[0047].

139. For example, U.S. Patent 6,973,455 (EX1020) discloses both the use of a file system mapping table and a directory of files in the file system including access attributes and pointers to where the file data resides in a disk array.

EX1020, 19:61-20:4.

140. As another example, U.S. Patent 6,697,846 (EX1021) is titled “Shared file system.” In its Background section it explains some characteristics of file systems. EX1021, 1:14-2:21. The reference explains that “data on a storage device is generally block accessible, in that data is addressed with the smallest granularity of a block; with multiple blocks forming an extent.” EX1021, 1:16-21. Also, “[f]ile systems generally hide the details of how data is stored on a storage device from the application program.” EX1021, 1:16-18. File systems present data to application programs, and the file systems “are required to map this application level interface to the often non-contiguous data blocks stored on the storage device. Generally, information required to map a particular file or directory to the physical locations of the storage device is stored by the file system in an inode within a data block. Inodes contain information, called attributes, about a particular file, such as file type, ownership information, access permissions and times, and file size. Inodes also contain a list of pointers which address data blocks. These pointers may address single data blocks or address an extent of several consecutive blocks.” EX1021, 1:64-2:21.

141. Alternatively, if a POSA would not have understood that the directory of Ophir's file system maps each file to the physical storage locations where the file's data is physically stored, then it is my opinion, and a POSA would have understood, that maintaining a "directory" mapped to the physical storage locations would have been a conventional and obvious way to implement Ophir's file system because a file system conventionally maintains such directories. EX1020, 19:61-20:4. EX1021, 1:64-2:21.

142. A POSA would have understood that the presentation of a file system by Ophir's mediator to the client device as part of providing file services simulates or emulates the underlying physical storage devices, and therefore presents a "virtual disk" under SFI's proposed construction as well. EX1042, 8.

ii. [1B-2]: generate the plurality of shares for storage on the plurality of physical storage devices by performing a securing operation on the dataset received from the client device and distributing the dataset in the shares;

143. In Ophir-Takeda, either ciphertext blocks C_n , blocks C_n together with respective ciphering attributes A_n , or blocks C_n together with pointers to their respective ciphering attributes meet the claimed "shares," as I discussed earlier at §VII.D.1.b.ii. Ophir's mediator generates these shares using Takeda's ciphering techniques in Ophir-Takeda, by performing a securing operation (encryption/ciphering) on the plaintext M , which is a dataset, received from the

client device. *See* above §§VII.B, VII.C.3, VII.D.1.b.ii. As I explained earlier, per Takeda's ciphering techniques, the plaintext M is divided into blocks M_n which are ciphered (i.e., encrypted) to create ciphertext blocks (C_n). *See* above §VII.B; Takeda, [0041]. Ciphering/encrypting blocks is a securing operation as claimed. Takeda, [0041]. The '140 patent explains that processes for securing data include encrypting data shares, stating:

“The data is encrypted with a secure key and split into one or more shares, and in one embodiment, four shares, according to the secure key. The secure key is stored safely with a reference pointer which is secured into four shares according to a secure key. The data shares are then encrypted individually and the keys are stored safely with different encrypted shares. When combined, the entire process for securing data according to the methods disclosed herein becomes a comprehensive package for data security.” EX1001, 60:56-64.

144. The '140 patent also states “In one embodiment of the present invention, the data securing method uses three sets of keys for an encryption operation.” EX1001, 61:28-30. It further states “As such, the encryption scheme itself may become a component of the methods of the present invention for securing sensitive data from unauthorized use or access.” EX1001, 64:56-59. These passages indicate that encrypting data blocks is a securing operation.

145. I understand the parties proposed different constructions for “*distributing the dataset in the shares.*” See EX1041, 3 (Google construction: “splitting the data set into shares”); EX1042, 2 (SFI construction: “dividing and placing the data set into a plurality of [data chunks/the shares/a plurality of shares]”). Ophir-Takeda discloses that feature under either party’s proposed construction. Ophir-Takeda’s dataset (M) provided from the client is split/divided into blocks M_n which are encrypted and placed into the plurality of shares C_n (or C_n plus A_n , or C_n plus a pointer to A_n). See §§VII.C.3, VII.D.1.b.ii *supra*. Takeda FIG. 3A shows an example with four shares into which the dataset was distributed. Takeda, FIG. 3A.

146. The shares are generated for storage on the plurality of physical storage devices in Ophir-Takeda, as I explained earlier in §VII.C.3.

iii. [1B-3]: include with each of the plurality of shares data indicative of the at least one session key used to secure the dataset; and

147. As I explained earlier in §VII.D.1.b.ii, Ophir-Takeda’s mediator generates shares (ciphertext C_n) associated with at least one session key (respective ciphering key of ciphering attribute A_n) used to secure a dataset (an object for storage). Ophir-Takeda includes with each of the shares the respective ciphering attribute or a pointer to the respective ciphering key, as discussed earlier in §§VII.B, VII.C.3, VII.D.1.b.ii. This includes with each of the plurality of shares

data indicative of the at least one session key used to secure the dataset under either party's construction, as discussed below. I understand that in the co-pending litigation, the parties offer competing constructions for the meaning of "data indicative of the at least one session key," with Google proposing the term covers metadata about the key but not a key itself, and SFI proposing the term covers "information that points out, points to, makes known, shows, directs to, hints at, or suggests..." the at least one session key. EX1041, 5; EX1042, 3-4. Ophir-Takeda meets either construction.

148. As I explained above in §VII.B, Ophir-Takeda's attributes An can include the key length, which is metadata about the ciphering key but is not the ciphering key itself. Takeda, FIG. 4, FIG. 12. Thus, the metadata in attribute An, such as the key length, is "data indicative of the at least one session key" under Google's proposed construction of "metadata about [the] key[]; not an encryption key." EX1041, 5.

149. The attribute An can include the ciphering key itself, which meets SFI's construction of "data indicative of the at least one session key" because, e.g., it "makes known" or "shows" the ciphering key. EX1042, 3-4; Takeda, [0042], FIG. 4, FIG. 12; *see also* EX1044, 34-35 (alleging that storing "data indicative of" a key is met by storing the key itself).

150. Alternatively, Ophir-Takeda stores a pointer with the ciphered data, which also meets SFI's construction because it "points to" or "directs to" the ciphering key. EX1042, 3-4; Takeda, [0040].

151. Claim 1 does not limit the manner of including the data indicative of the at least one session key with each share.

152. If the "share" is just the encrypted data, then Ophir-Takeda storing the attribute An or a pointer to An in the same file as its encrypted data block Cn (per Takeda, [0040], [0048]) includes data (An or its pointer) indicative of the session key (An) with the share (Cn) as claimed. EX1001, FIG. 21 (key with encrypted data share).

153. Alternatively, if the share is the encrypted data plus the data indicative of the session key, then Ophir-Takeda storing the attribute An or a pointer to An in the same file as its encrypted data block Cn (per Takeda, [0040], [0048]) also includes data (An or its pointer) indicative of the session key (An) with, and in fact within, the share (Cn and An, or An and its pointer) as claimed. *Id.*

iv. [1B-4]: reconstitute the dataset from at least a portion of the plurality of shares stored on the physical storage devices in response to a request from the client device for information in the dataset.

154. The Encryption/Decryption Unit 517 of the mediator in Ophir-Takeda decrypts the stored data in response to a client device's request for the data. Ophir, FIG. 5. Ophir states that "Encryption/ decryption unit 517 encrypts data from the

data clients into encrypted data for safe storage in data storage devices, and decrypts data retrieved from data storage devices into decrypted data for sending to data clients.” Ophir, [0032]. Ophir states that a “typical data decryption scenario for reading data from storage” includes “extracting the storage properties of the requested data from the client protocol.” Ophir, [0046]. Ophir continues by saying the scenario includes “decrypting the data and encapsulating the data within the client protocol (e.g., block device protocols, file system protocols, database services protocols) as a response to the data client.” Ophir, [0050]. Those passages indicate that the data is requested by the client device and returned to the client in response to the request.

155. In Ophir-Takeda, Encryption/Decryption Unit 517 implements Takeda’s decryption techniques (see §VII.C.3 *supra*), which reconstitute the original dataset (plaintext M) from the plurality of shares stored on the physical storage devices. Takeda, FIGs. 6, 14, [0048] (“During deciphering, the attribute data {An} is set for the corresponding ciphertext {Cn} to decipher the data, and the original plaintext M is obtained.”).

156. In conclusion, it is my opinion that Ophir-Takeda meets every limitation of claim 1 and renders it obvious to a POSA under both parties’ constructions of the various claim terms for which the parties propose different constructions.

2. **Claim 2: The secure storage network of claim 1, wherein the secure storage system is further configured to generate the plurality of shares by performing a cryptographic operation on the dataset.**

157. The ciphering operation that Ophir-Takeda's mediator performs on the dataset (M) to generate the shares (see §VII.D.1.c.ii regarding claim limitation [1B-2]) is a cryptographic operation, as claimed. EX1037, 380 (encrypt means encipher).

158. Thus, in my opinion Ophir-Takeda meets every limitation of claim 2 and renders it obvious to a POSA under both parties' claim constructions.

3. **Claim 3: The secure storage network of claim 1, further comprising a key management server configured to store workgroup keys.**

159. The '140 patent describes "a workgroup key" as being used to "further encrypt[]" the "encryption keys, splitting keys, or both." EX1001, 75:63-67.

160. Ophir likewise teaches "to use '*master keys*' to encrypt encryption/decryption keys, thereby making it safe to store encryption/decryption keys on external storage instead of in limited internal memory." Ophir, [0032].

161. Ophir's "master keys" are "workgroup keys" as claimed. *Id.* The '140 patent uses "workgroup keys" to encrypt encryption keys. EX1001, 75:63-67. Likewise, Ophir's "master keys" encrypt Ophir's encryption/decryption keys. Ophir, [0032]. A POSA would have used Ophir's master keys to encrypt Ophir-

Takeda's ciphering keys to make it safe to store the ciphering keys externally in those implementations of Ophir-Takeda in which the ciphering keys are not stored with the ciphertext. Ophir, [0032].

162. To the extent claim 3's "workgroup keys" are interpreted to require providing different levels of access to different groups of users, it is my opinion that Ophir-Takeda still renders claim 3 obvious to a POSA. Ophir-Takeda designates access privileges for users as part of Takeda's ciphering attributes. *See* §§VII.B, VII.C.3; Takeda, FIGs. 4 and 12 ("access privilege"), [0054], [0060]. A POSA would have implemented access privileges via Ophir's "master keys" in Ophir-Takeda to provide enhanced security by restricting access to only designated users. Takeda, [0050], [0054], [0060].

163. I note that the term "key management server" appears only in claim 3 but does not appear in the specification of the '140 patent. The '140 patent illustrates a "key management system" 4214 which "may provide for external key management for keys utilized in the encryption and/or splitting operations of the secure parser." EX1001, FIG. 42, 82:31-33. Ophir likewise teaches that "encryption/decryption keys...are either generated locally (such as by encryption/decryption unit 517, or...are *stored on an external key server* and retrieved by encryption/decryption unit 517...)." Ophir, [0032]. Ophir's external key *server* thus performs the same key management function as the key

management system 4214 of the '140 patent, and therefore it is a “key management server” as claimed. POSAs would have known to similarly store Ophir’s “master keys” on such an external key server so that they would not take up “limited internal memory” (Ophir, [0032]), thus meeting the limitations of claim 3.

164. Thus, in my opinion Ophir-Takeda meets every limitation of claim 3 and renders it obvious to a POSA under both parties’ claim constructions.

4. Claim 4: The secure storage network of claim 1, wherein the secure storage system is further configured to provide access to the reconstituted dataset.

165. The mediator in Ophir-Takeda’s secure storage network is configured to provide access to the reconstituted dataset. Ophir, [0032].

166. I note that neither claim 4 nor the specification of the '140 patent explains what is meant by “the reconstituted dataset.” In light of the specification, a POSA would interpret claim 4 to encompass retrieving encrypted shares of the dataset from storage and decrypting them to recover the dataset. For instance, the '140 patent states that “This method also optionally comprises reconstituting or re-assembling the secured data into its original form for authorized access”. EX1001, 2:46-48.

167. Ophir-Takeda’s mediator “retrieve[s]” the stored data from the physical storage locations and “send[s] the data back to [the] data client” (Ophir,

[0012]) after the data is decrypted by encryption/decryption unit 517. Ophir, [0016], [0032]. The decryption in Ophir-Takeda is performed using Takeda's deciphering techniques, as I explained above in §VII.C.3. Takeda, [0048]. The decryption involves ordering the blocks appropriately to reconstitute the input data M in its original form. Takeda, [0048].

168. To the extent Takeda is not determined to expressly disclose ordering the retrieved data blocks in the dataset's original order before providing them back to the requesting client, doing so would have been the obvious, common-sense way to implement Ophir-Takeda. POSAs knew it was desirable to have the mediator properly order the dataset blocks before returning them to the client to make the dataset understandable, and because it was the mediator that broke the dataset into blocks, it would be difficult (if not impossible) for the client to determine the correct order of the returned data blocks if they were out of order. The decrypted and properly ordered blocks of the dataset is a reconstituted dataset as claimed.

169. Thus, in my opinion Ophir-Takeda meets every limitation of claim 4 and renders it obvious to a POSA under both parties' claim constructions.

5. Claim 5: The secure storage network of claim 1, wherein the secure storage system is further configured to establish a secure connection between the client device and the secure storage system.

170. I see that the term “secure connection” appears only in the claims, but that the specification of the ’140 patent does not provide any limiting description of this term.

171. Ophir teaches that data transmitted between the client device and the mediator (the claimed secure storage system – see §VII.D.1.c *supra*) is “protect[ed]” by further encrypting the data for transmission between the client and the mediator over the communication network. Ophir, [0014]-[0015]. A POSA would have understood that this establishes a secure connection between the client and mediator, as claimed. *Id.*.

172. If it is considered that Ophir does not explicitly disclose that its data clients and mediator are connected by a secure connection, that would have been the routine, conventional, and obvious way to implement what Ophir describes. Secure connections between network components were well known before the filing date of the ’140 patent. EX1022, FIG. 1 (secure connections 52, 54, 56); 4:19-22; EX1023, [0013], [0015]; EX1024, 5:15-28 (describing secure socket layers). A POSA would have been motivated to use such a secure connection in Ophir-Takeda because the very purpose of Ophir to ensure security in the storage of data would be enhanced by further securing the data for its transmission from

the client for storage in the secure storage system. Ophir, Title (“*Secure* Network Data Storage Mediator”), [0009].

173. Thus, in my opinion Ophir-Takeda meets every limitation of claim 5 and renders it obvious to a POSA under both parties’ claim constructions.

6. Claim 6: The secure storage network of claim 1, wherein the shares contain a substantially random distribution of the received dataset.

174. A POSA would have implemented Ophir-Takeda by randomizing the plaintext data M when creating the blocks Mn. Takeda does not specify the manner in which blocks Mn need be formed, other than that a single dataset should be split into multiple blocks. Takeda, [0038]-[0039]; [0068]. Randomization of data in a security context was a well-known technique to enhance data security, as evidenced by multiple prior-art references. EX1027 (“Tajima”), [0106] (dividing personal information into portions of arbitrary lengths as function of random numbers); EX1028, [0058] (randomly fragmenting data enhances privacy); EX1029, [0011] (securing digital data involves splitting data into fragments based on random factors); EX1030, [0016]-[0018] (randomizing segment size improves data obfuscation); EX1031, Title, Abstract, [0004] (random dispersal of data reduces likelihood of data being compromised); Dickinson, 15:26-27, 23:9-11. A POSA would have known to randomize the plaintext data within the plaintext blocks Mn during the process of creating the blocks, to provide enhanced security.

Id. Thus, each block M_n would contain randomly arranged plaintext data from the received dataset M .

175. The POSA would have had a reasonable expectation of success in implementing such randomization because techniques for randomizing data, including as part of splitting data into blocks, were well known. EX1027, [0106]; EX1028, [0058]; EX1029, [0011]; EX1030, [0016]-[0018]; EX1031, Title, Abstract, [0004]; Dickinson, 15:26-27, 23:9-11. Moreover, Takeda teaches that the attribute data can include the bit starting point and the ciphered size, which is information that would facilitate accurate performance of Takeda's ciphering/de-ciphering methods when the data is randomized. Takeda, FIGs. 4 and 12. ("bit rectangular start point," "bit start point," "bit rectangular size," "ciphered bit length."), [0054], [0077].

176. The result of randomizing the data would have been predictable. The plaintext blocks M_n would contain random distributions of the plaintext M , so that the resulting ciphertext blocks C_n , which I explained earlier are formed by encrypting the plaintext blocks M_n and which represent "shares" as claimed, would contain randomized distributions of the received dataset (M). That would meet the requirements of claim 6.

177. Thus, in my opinion Ophir-Takeda meets every limitation of claim 6 and renders it obvious to a POSA under both parties' claim constructions.

7. Claim 8: The secure storage network of claim 1, wherein the secure storage system is configured to secure the dataset by encrypting the dataset using the at least one session key.

178. The mediator of Ophir-Takeda implements Takeda's ciphering method using attributes A_n , which include session keys (e.g., encryption keys), to secure the dataset by securing respective portions of the dataset. *See* §§VII.D.1.b.ii, VII.D.1.c.iii *supra*. Since the ciphering attributes A_n perform an encryption on the respective portions M_n of plaintext, Ophir-Takeda secures the dataset by encrypting the dataset using the at least one session key (ciphering key). Takeda, FIG. 3A, [0042].

179. Thus, in my opinion Ophir-Takeda meets every limitation of claim 8 and renders it obvious to a POSA under both parties' claim constructions.

8. Claim 10: The secure storage network of claim 1, wherein the data indicative of the at least one session key comprises: a portion of the at least one session key or data indicative of a portion of the at least one session key.

180. Claim 10 requires that the “data indicative of the *at least one* session key” comprises a portion of the at least one session key or data indicative of a portion of “the *at least one* session key.” In Ophir-Takeda, each key A_n used to encrypt a block C_n is a session key as I explained above in §VII.D.1.c.iii. That means all the keys A_1 - A_n collectively meet the *at least one* session key.

181. The alternative implementations of Ophir-Takeda that store with the ciphertext block C_n either the key (A_n) and/or key length, or a pointer to the key

(An) that was used to cipher that block (see §VII.C.3 *supra*) meet claim 10. For each block (Cn), the respective key An stored with it is “a portion of the at least one session key” (A1-An) as claimed, because a single key An is a portion of the collective “at least one session key” (A1-An) used to encrypt the entire dataset. Similarly, for each block (Cn), the key length or pointer to respective key An is “data indicative of a portion of the least one session key” (A1-An) under both parties’ competing constructions as discussed *supra* §VII.D.1.c.iii. EX1041, 5-6; EX1042, 3-4.

182. Thus, in my opinion Ophir-Takeda meets every limitation of claim 10 and renders it obvious to a POSA under both parties’ claim constructions.

VIII. GROUND II: OPHIR, TAKEDA, AND BIRRELL RENDER CLAIM 7 OBVIOUS

A. Birrell (EX1033)

183. Birrell discloses backing up data on a plurality of network-connected computers. Birrell, Title, Abstract. Birrell also provides so-called “fault tolerance” when backing up data to multiple storage devices, which is done to protect against failure of some of the storage devices. Birrell, 15:39-42, 16:7-17:32. This fault tolerance is achieved by providing data redundancy by either replication or encoding. Birrell, 15:42-43. Encoding for redundancy can be used to make up for lost data without having to replicate the data. Birrell, 15:39-57. Whether using

replication or encoding, only a subset of the data need be retrieved to restore the data. Birrell, 18:14-18.

B. Ophir-Takeda-Birrell Combination

184. In my opinion, a POSA would have further modified Ophir-Takeda (§VII.C above) to enhance its ciphering algorithms by beneficially providing redundancy of the ciphered data via encoding or replication. Ophir-Takeda's ciphertext C_n is stored on Ophir's storage devices as I explained above in §VII.C.3. A POSA would have recognized based on Birrell's disclosure that the risk of one or more of the storage devices failing – thus hindering the mediator's ability to restore the original data – could have been reduced by providing redundancy of the ciphertext blocks C_n . Birrell, 15:39-42, 16:7-17:32. Takeda does not mandate any particular ciphering algorithm, and therefore a POSA implementing Takeda's method in Ophir-Takeda would have recognized the benefit and compatibility in implementing a Ophir-Takeda ciphering algorithm to provide redundancy by encoding and/or replication, as taught in Birrell. The resulting Ophir-Takeda-Birrell combination would have generated ciphertext blocks C_n with redundancy, such that reconstituting the data involves retrieving only a subset of the ciphertext blocks C_n (without need to retrieve redundant blocks).

185. A POSA would have had a reasonable expectation of success in implementing a Ophir-Takeda ciphering algorithm providing redundancy of the

ciphertext blocks C_n, as taught in Birrell. Birrell provides multiple techniques for providing such redundancy, including encoding and mere replication. Birrell, 15:39-18:29. Applying Birrell's encoding to the ciphertext blocks C_n ("shares"), or even just replicating them, would have been well within the POSA's level of skill because those techniques were well-known, conventional, and required only ordinary programming skill to implement.

C. Claim 7: The secure storage network of claim 1, wherein the secure storage system is configured to reconstitute the received dataset from fewer than all of the shares

186. Ophir-Takeda-Birrell's mediator generates ciphertext blocks C_n with redundancy. *See* §VIII.B above. Therefore, when the mediator reconstitutes the original plaintext data, it does so using fewer than all the ciphertext blocks, because it need not use the blocks that are redundant in order to reconstitute the original plaintext data.

187. Thus, in my opinion Ophir-Takeda-Birrell meets every limitation of claim 7 and renders it obvious to a POSA under both parties' claim constructions.

IX. GROUND III: OPHIR, TAKEDA, AND DICKINSON RENDER CLAIM 9 OBVIOUS

A. Dickinson (EX1005)

188. I discussed various aspects of Dickinson in §§V.A, V.C above (and §X below). In prior art disclosure in Dickinson that is identical to portions of the '140 patent itself (*see* EX1040), Dickinson discloses that a set of sensitive data (S) can be

secured by “splitting” the data into at least two independently undecipherable pieces of data by performing a conventional “XOR” operation (a well-known mathematical operation that changes the bits of binary data) on the sensitive data (S) with a random number. Dickinson, 23:9-12 (“the authentication engine 215 employs the data splitting module to *mathematically operate on the enrollment authentication data so as to split the data into at least two* independently undecipherable, randomized, numbers. As mentioned in the foregoing, the at least two numbers may comprise a statistically random number and a binary XORed number.”); *see also* EX1001, FIG. 8 (steps 805, 810, 820, 825, and 830 as part of “data splitting process”), 20:23-48 (same), 20:23-48 (same), 23:26-33. The result of the XOR operation is an encrypted version of the data (the XOR output) and the random number used in the XOR process. *Id.* Dickinson states that the sensitive data is therefore “split” because it cannot be recovered from the XOR result or the random number alone. Dickinson, 23:9-12.

B. Ophir-Takeda-Dickinson Combination

189. In my opinion, a POSA would have further modified Ophir-Takeda (§VII.C above), based on Dickinson (§IX.A), to implement Takeda’s ciphering of the plaintext blocks M_n by performing the mathematical operation $C_n = M_n \text{ XOR } A_n$, where M_n is the plaintext block and A_n is the ciphering attribute (key), and where A_n is implemented as a random number, as I discuss below in this section.

190. Takeda does not restrict what ciphering attributes A_n (including keys or algorithms) can be used in its ciphering process. Takeda, [0041]-[0042]. Implementing an XOR operation as the ciphering algorithm in Ophir-Takeda and implementing the ciphering key A_n as a random number would have represented the use of a known technique from Dickinson, which I refer to as the Ophir-Takeda-Dickinson system and method. The resulting system and method would have carried predictable results (the ciphertext C_n would have represented the predictable result of the XOR between M_n and A_n). EX1001, 20:48-50 ('140 patent conceding XOR operation was well known). A POSA would have had a reasonable expectation of success in Ophir-Takeda-Dickinson because Dickinson's XOR operation was a known, precise mathematical operation for changing the bits of binary data according to, *e.g.*, the values of a random number / encryption key. The '140 patent itself admits that the XOR operation was "well known." EX1001, 20:48-50.

C. Claim 9: The secure storage network of claim 1, wherein the secure storage system is configured to secure the dataset by splitting the dataset using the at least one session key

191. The resulting Ophir-Takeda-Dickinson Combination (discussed above in §IX.B), splits the dataset (M) (*see* §VII.D.1.b.ii above) using the at least one session key (ciphering attributes A_n including a ciphering key, *see* §VII.D.1.b.ii). This splitting process is within the scope of claim 9 (at least) because the '140 patent discloses that the purported invention is not limited to a particular splitting process.

See EX1001, 22:6-11. Performing an XOR operation of each plaintext data block M_n with a respective ciphering key of the ciphering attributes A_n (claimed “session key”) meets claim 9 by performing the same XOR splitting operation discussed in the ’140 patent. Dickinson, 23:9-12; EX1001, 23:26-33.

192. Thus, in my opinion Ophir-Takeda-Dickinson meets every limitation of claim 9 and renders it obvious to a POSA under both parties’ claim constructions.

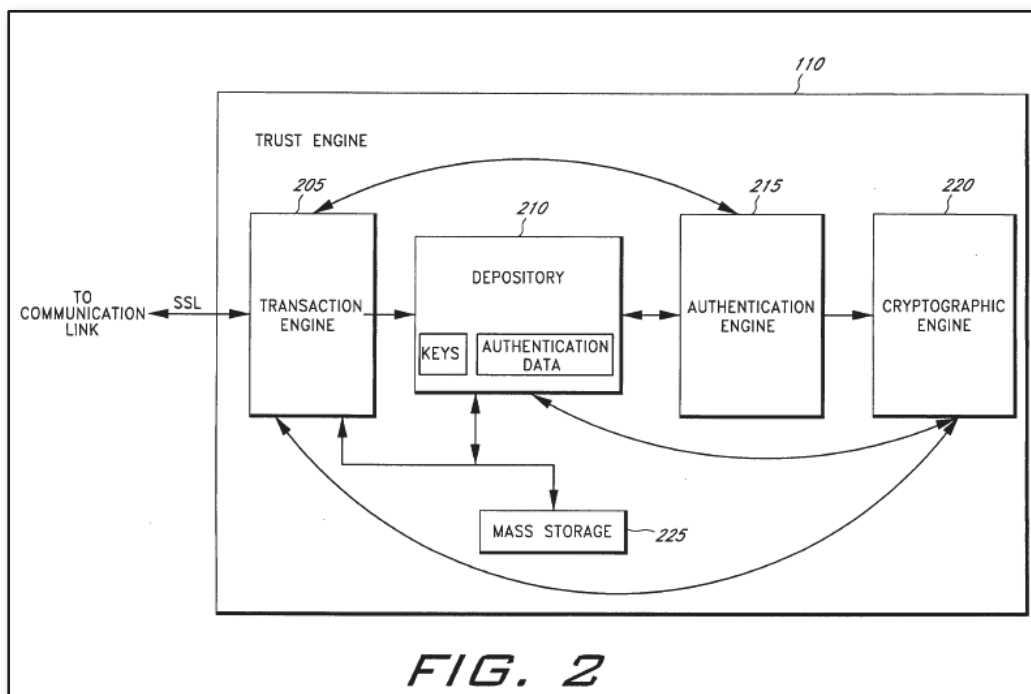
X. GROUND IV: DICKINSON AND TAKEDA RENDER CLAIMS 1-10 OBVIOUS

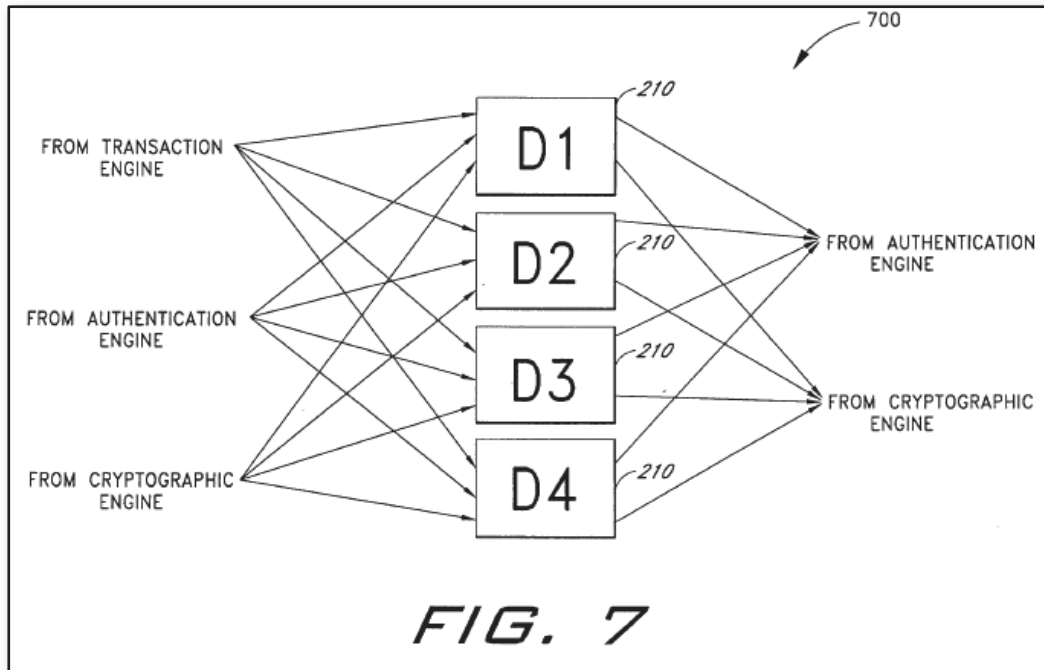
193. As I discussed above in §V.C, the USPTO Examiner during prosecution of the ’140 patent found that Dickinson disclosed substantially all of claim 1’s limitations. The Examiner did not find *explicit* mention in Dickinson that the virtual disk comprises a directory mapping to the plurality of physical storage devices ([1B-1]), and appears to have found that Dickinson did not disclose “includ[ing] with each of the plurality of shares data indicative of the at least one session key used to secure the dataset” ([1B-3]) EX1002, 1236-1242, 1247-1250 (original claim 14), 1360-1361. As I discuss further below in §X.D.1.c.i, Dickinson does in fact disclose (or alternatively render obvious in combination with Takeda) limitation [1B-1]; and as I discuss further below in §X.D.1.c.iii, Takeda discloses limitation [1B-3] but (as I discussed above in §V.C), Takeda was not before the Examiner. For the reasons I discuss below in §§X.D.1-.10, it is my opinion that the Dickinson-Takeda

combination (which I discuss in §X.C below), would have rendered obvious all challenged claims (1-10) of the '140 patent.

A. Dickinson (EX1005)

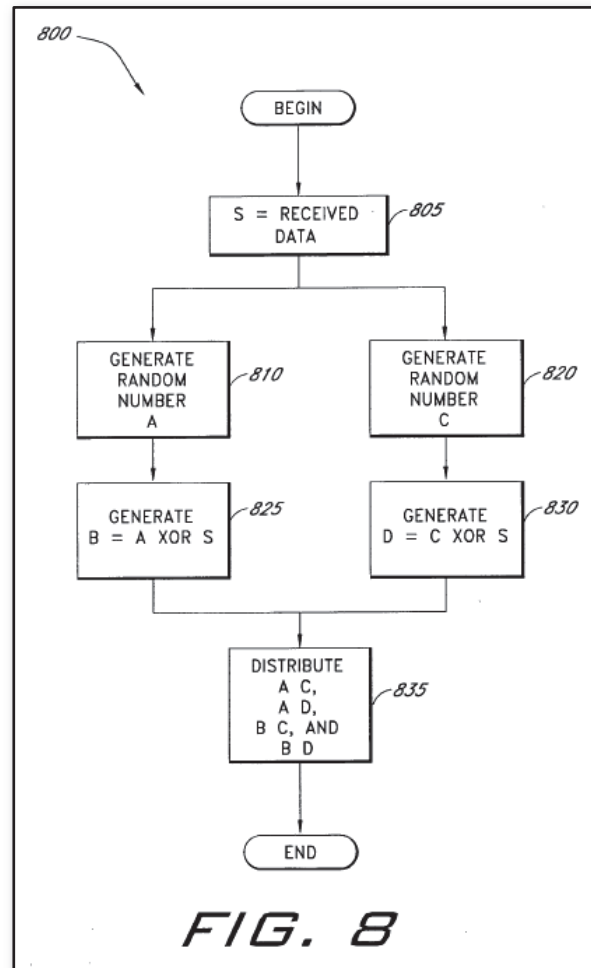
194. I discussed various aspects of Dickinson in §§V.A, V.C and §IX above. Dickinson discloses that a set of sensitive data “S” (e.g., a user’s “password, login, financial credentials, or the like”) that is associated with a user system 105 can be stored in a “sensitive data vault” that is in “the data storage facilities of the depository 210 [of trust engine 110, FIG. 2 (reproduced below)], *or split and stored in the depository system 700,*” which Dickinson discloses includes multiple data storage facilities D1-D4 (as shown in Dickinson’s FIG. 7, which I have reproduced below). Dickinson, 29:13-23.





195. Dickinson’s “[d]ata splitting process” 800 (which is illustrated in FIG. 8 reproduced below) “splits” sensitive data S into at least two independently undecipherable portions—the four portions identified as A-D in FIG. 8—and distributes those portions among the data storage facilities D1-D4. Dickinson, 8:11-14, 19:16-20:18, 21:10-14. Dickinson says that these portions are “independently

undecipherable” because no one portion is sufficient, by itself, to reconstruct the original data S. Dickinson, 20:30-21:14, 23:6-15.



196. The independently undecipherable portions in FIG. 8 include: (1) two “encrypted portions” (B and D) that are encrypted versions of the full data S and (2) the keys (A and C) that required to decrypt respective encrypted portions. Dickinson, FIG. 8 (steps 825 and 830), 20:25-21:14. Specifically, process 800 performs two separate encryption steps 825 and 830 in parallel on sensitive data S. Dickinson, 20:32-21:1, 21:10-14. Step 825 performs an exclusive OR (“XOR”) of

S with a random number A of the same bit-length, resulting in encrypted data B. Dickinson, FIG. 8, 20:33-21:1. Step 830 XORs S with a different random number C resulting in differently encrypted data D. *Id.* POSAs understood that these XOR operations with random numbers are encryption steps. My opinion in this regard is corroborated by EX1034 at 4:25-29, which discloses a system where “encryption and decryption [is] accomplished by means of [] XOR circuits.”

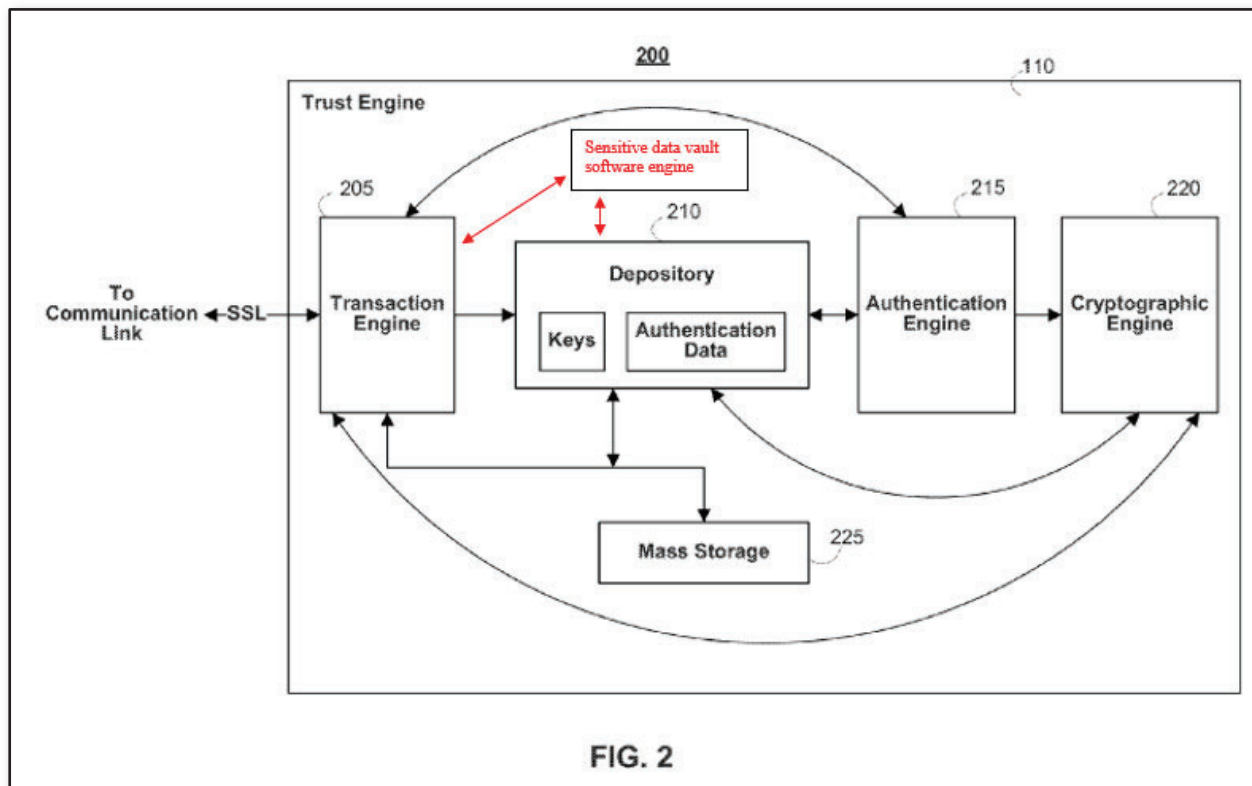
197. In step 835, numbers A, B, C, and D are paired and distributed among data storage facilities D1-D4, in such a way that “none of the pairings contain sufficient data, by themselves, to reorganize and decipher the original sensitive data S.” Dickinson, 21:1-14. For example, portions A and B are never stored on the same storage system, and neither are portions C and D. Dickinson, 21:3-14 (describing storing all other combinations of A-D). The reason for this separate storage is that, when encrypting data with an XOR operation, the encrypted data can be decrypted by XORing it again with the same number that was used to encrypt it—and so, as Dickinson states, “when the data assembling module receives one of A and B, or, C and D, [it] can advantageously reassemble the sensitive data S.” Dickinson, 21:28-30.

198. That reassembly of the sensitive data S is what occurs when the authorized user needs access to the user’s sensitive data S. Upon request from the trust engine’s transaction engine (FIGs. 1-2), stored (encrypted) data is forwarded

by the depository to a data assembling module to assemble it into a useable format (e.g., sensitive data S), which is then presented to the user system 105. Dickinson, 19:27-29. As Dickinson states, data assembling module 525 “uses data portions from at least two of the data storage facilities D1 through D4”—since “any two” of the pairings “AC, AD, BC, and BD” will provide either both “A and B” or both “C and D”—allowing module 525 to reassemble S as “ $S = A \text{ XOR } B$ or $S = C \text{ XOR } D$.” Dickinson, 21:21-32.

199. Dickinson’s trust engine also includes a “stand-alone software engine implementing the [above-described] data vault functionality” that includes splitting and storing the data (FIG. 8) for later reassembling and serving that reassembled data to the user system 105 or other requesting application. Dickinson, 29:26-30. The

data vault software engine is shown in Dickinson's FIG. 2, which I have annotated and reproduced below. *Id.*



Dickinson, FIG. 2 (annotated)

B. Takeda (EX1008)

200. As I discussed above in §VII.B, Takeda's teaching includes storing with a block of encrypted data attributes such as the encryption key and key length, or a pointer to a stored encryption key that was used to encrypt the block of data. Takeda [0040], [0054], FIGs. 4 and 12.

C. Dickinson-Takeda Combination

201. In my opinion, a POSA having the teachings of Dickinson and Takeda before them would have readily recognized at least three obvious alternatives for

storing Dickinson's encrypted data portions (B and D) and the keys that encrypted them (key A that encrypted B, and key C that encrypted D):

1. Store encrypted data B with key C, and store encrypted data D with key A, as Dickinson teaches. Dickinson, 21:3-5.
2. Apply to Dickinson Takeda's teaching to store with a data portion a pointer to a key and/or metadata about the key (*e.g.*, key length) rather than the key itself, resulting in storing encrypted data B with a pointer to and/or metadata about key C, and storing encrypted data D with a pointer to and/or metadata about key A. Takeda, [0040], [0054].
3. Further apply Takeda's teaching to store with each data portion a pointer to and/or metadata about the key that encrypted that portion, rather than a different key—resulting in storing encrypted data B with a pointer to and/or metadata about key A, and storing encrypted data D with a pointer to and/or metadata about key C. Takeda, [0048].

202. As I explain in §X.C.3 below, it is my opinion that a POSA had reason to implement Dickinson's data splitting process 800 in any of the three alternative ways discussed above, and a reasonable expectation of success in doing so. Any of these alternatives meets all limitations of claims 1-10, as I demonstrate in §§X.D.1-.10 below.

1. Reasons to Modify Dickinson

203. Alternative (1) that I discussed in the paragraphs directly above requires no modification to Dickinson.

204. As to alternatives (2) and (3), POSAs understood that Dickinson's system must ultimately retrieve both an encrypted data portion and the key that encrypted it (*e.g.*, either both B and A, or both D and C) to recover the original dataset S. Dickinson, 20:30-21:32. Dickinson does not specify how its system knows where to retrieve the correct key (*e.g.*, A) for decrypting the corresponding encrypted data portion (*e.g.*, B) to recover data S.

205. A POSA would have understood that Takeda's technique of storing, with encrypted data, a pointer to and/or metadata about a key rather than a key itself (alternative (2)) (Takeda, [0040]) was a way to facilitate retrieving a correct key and also would have beneficially allowed for separate management of the encrypted data and the keys for that data, and thus, for example, allow for deciding the best use of limited storage space. Takeda, [0048], [0056].

206. My opinion that separate management of data and keys would have been recognized by POSAs as a beneficial implementation, *e.g.*, for managing limited storage is corroborated by Ophir at [0032], which discloses "making it safe to store encryption/decryption keys on external storage instead of in limited internal memory." *See also* §VII.A above.

207. A POSA would have also understood that Takeda's further technique of storing, with each encrypted data portion, a pointer to and/or metadata about the key used to encrypt/decrypt that particular data portion (as in alternative (3)) provides a technique for providing the association between the encrypted data portion and its key "in such a manner that the correspondence relation between the data can be recognized" from the data/key pair being stored together. Takeda, [0048]. The stored metadata could also be used beneficially during decryption.

208. In my opinion, a POSA had numerous reasons to implement Dickinson's data splitting process 800 by storing with an encrypted data portion (B or D) a pointer to and/or metadata about the encryption key (A or C) used to encrypt that data portion or the other data portion.

209. **First**, Takeda provides an express teaching that an encryption key and encrypted data portions can be associated by storing a key pointer/metadata with the encrypted data, which would have motivated a POSA to use that technique to provide that association in Dickinson, which (as I discussed above) would need a way to associate data with corresponding keys in order to reconstruct the original data. Takeda, [0040].

210. **Second**, implementing Dickinson's data splitting process 800 to use Takeda's pointer/metadata technique would have been the application of Takeda's known pointer/metadata technique to Dickinson's device, which was ready for the

improvements of separate organization/storage of keys and providing an association between data portion pairs (e.g., A and B) that together provide the original data S. The result of this implementation would have been predictable because these were well-known, conventional encryption and storage techniques based on precise mathematics.

211. **Third**, the Dickinson-Takeda alternatives that I discussed above would have been the simple substitutions of one known technique (Takeda's pointer/metadata techniques) of associating encrypted data with locations of and metadata about encryption keys for another (e.g., Dickinson's undisclosed technique for forming the association). Takeda, [0040].

212. **Fourth**, the Dickinson-Takeda alternatives would have been the combination of prior art elements (Takeda's pointer/metadata techniques with Dickinson's data-splitting process) according to known techniques to yield predictable results. POSAs understood that storing a pointer/metadata with the encrypted data portion would have provided for straightforward (and conventional) retrieval of the encryption key (e.g., A or C) from its storage location when needed to decrypt the encrypted data portion (e.g., B or D, respectively). Takeda, [0040], [0072]. The pointer's/metadata's function in Dickinson-Takeda would have been the same as in Takeda's system – to identify the location of where the encryption key is stored so that the key can be retrieved to decrypt the encrypted portion. *Id.*

2. Reasonable Expectation of Success

213. In my opinion, a POSA would have had a reasonable expectation of success in modifying Dickinson's FIG. 8 data splitting process to implement Takeda's pointer/metadata techniques in the manners I discussed above in §X.C.1. POSAs were aware that storing data, including metadata, was a common technique, including storing data/metadata with encrypted data. For example, Takeda itself (in FIGs. 3A-3B, 4 at [0048]) disclosing storing encrypted data portions with ciphering attributes that include metadata about an encryption key, such as key length. Pointers were also a known software tool used to identify the location of stored data. My opinion in this regard is also corroborated by Takeda (at [0044], which discloses storing encrypted data portions with a ciphering attribute) and further corroborated by EX1039 at 3:35-38, which discloses using a "[w]orkspace pointer, WP, [which] contains the address of [a pointer to] the first of 16 consecutive memory locations that are actively used as workspace elements."

214. In my opinion, a POSA readily could have implemented a pointer and/or metadata in Dickinson's FIG. 8 data-splitting process to point to the storage location of the encryption keys used in steps 825 and 830 because pointers to data (whether an encryption key or otherwise) were common in the prior art and well-understood. My opinion that POSAs could have readily used a point/metadata to point to the storage location of encryption keys is corroborated by Takeda at

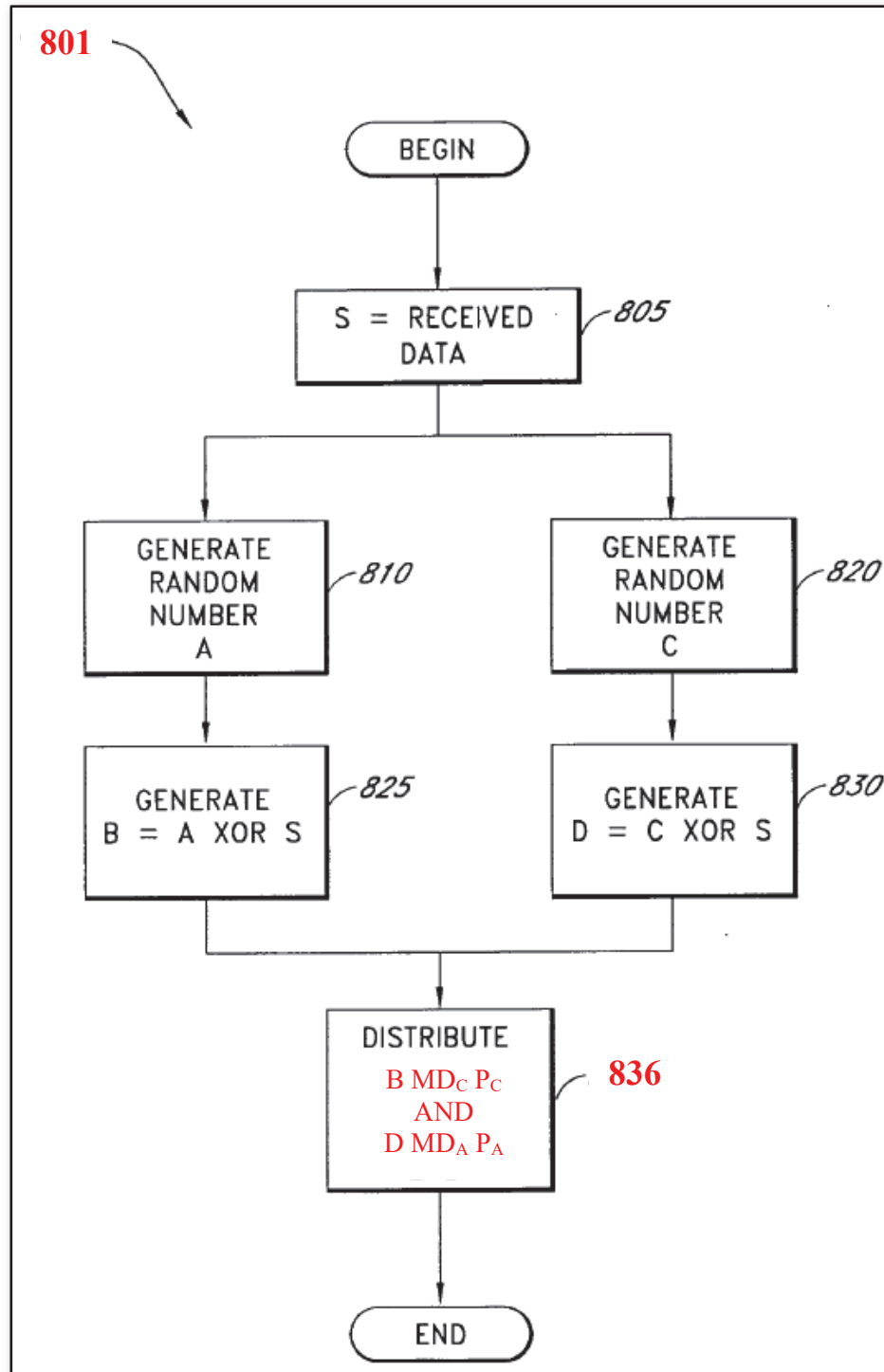
[0044] and EX1039 at 3:35-38, which I discussed in the paragraph above and demonstrate that pointers to/metadata regarding data (including encryption keys) were conventional and well-understood in the art.

215. In the above alternatives of Dickinson-Takeda, the Dickinson-Takeda combination would have retained the characteristic of Dickinson’s system wherein pairs of values stored on the same repository (*see* step 835 – AC, AD, BC, BD) do not include encrypted data with its key (that is, for example, A and B are stored on different repositories, as are C and D—because the key itself is not stored with any encrypted data portions in both alternatives (2) and (3)). Dickinson, 21:1-14. *See also* §X.C.3 below (where I discuss what I have labelled steps **836/837**, which are based on Dickinson’s step **835**).

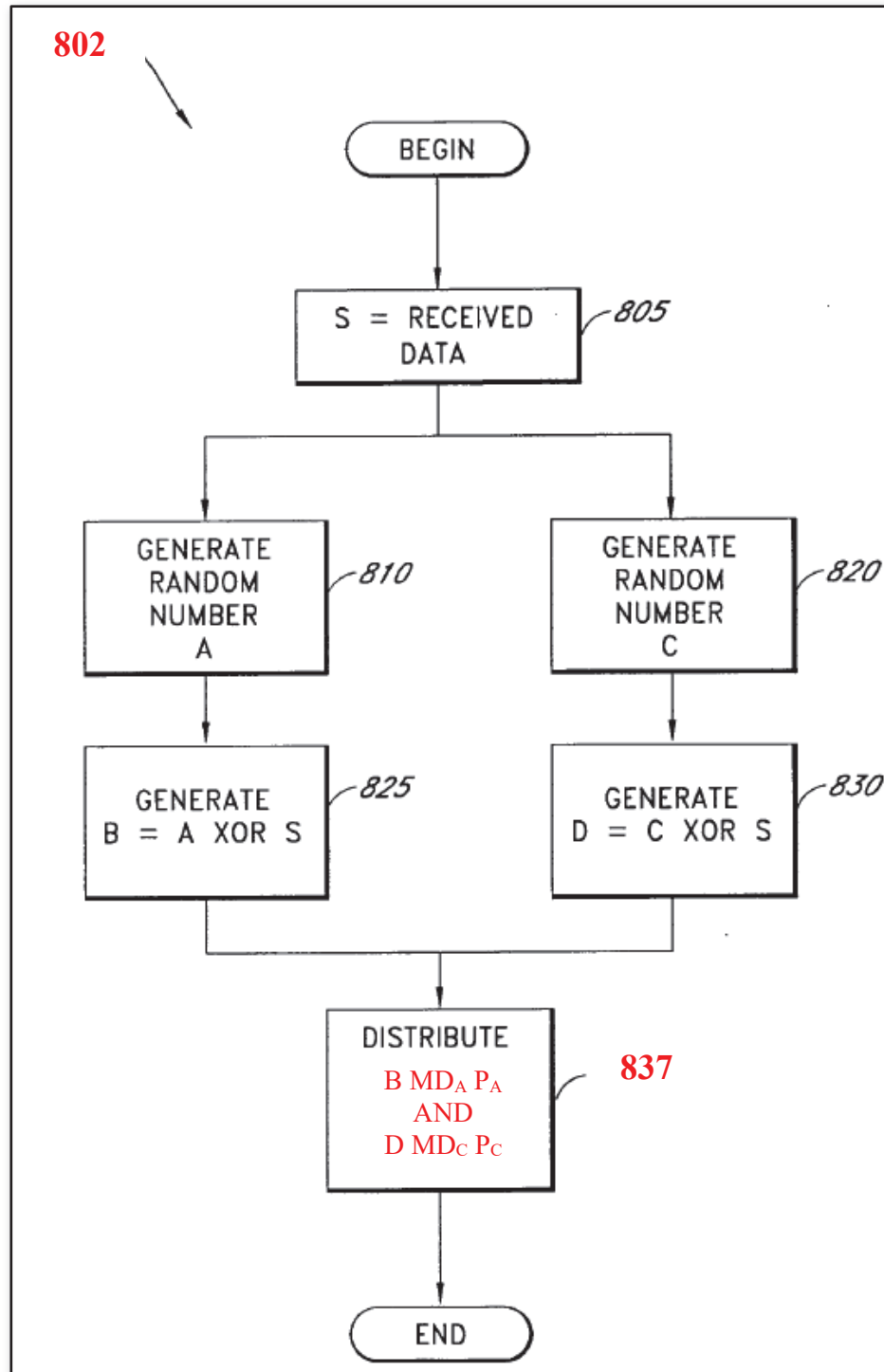
3. Resulting Combination

216. As I discussed above (§§X.C and X.C.1), Dickinson-Takeda alternative (1) does not modify Dickinson, and alternatives (2) and (3) differ from Dickinson only in implementation of a modified data-splitting process, performed by the sensitive data vault software engine, in which the encrypted data (e.g., B and D) is stored with either: (i) additional information (“MD”) about the data string used to encrypt/decrypt a different portion of encrypted data and/or a pointer (“P”) to the data string used to encrypt/decrypt the different portion of encrypted data, represented by what I label below as distribute step “836” in the data splitting

process I label as “801” in the first annotated version of Dickinson’s FIG. 8, below; or (ii) additional information (“MD”) about the data string used to encrypt/decrypt the encrypted portion with which the additional information is stored and/or a pointer (“P”) to the data string used to encrypt/decrypt the encrypted portion with which the additional information is stored (e.g., A and C, respectively), represented by what I label below as distribute step “837” in the data splitting process I label as “802” in the second annotated version of Dickinson’s FIG. 8 below.



First Annotated FIG. 8 of Dickinson



Second Annotated FIG. 8 of Dickinson

D. Dickinson-Takeda Meets All Limitations of Claims 1-10.

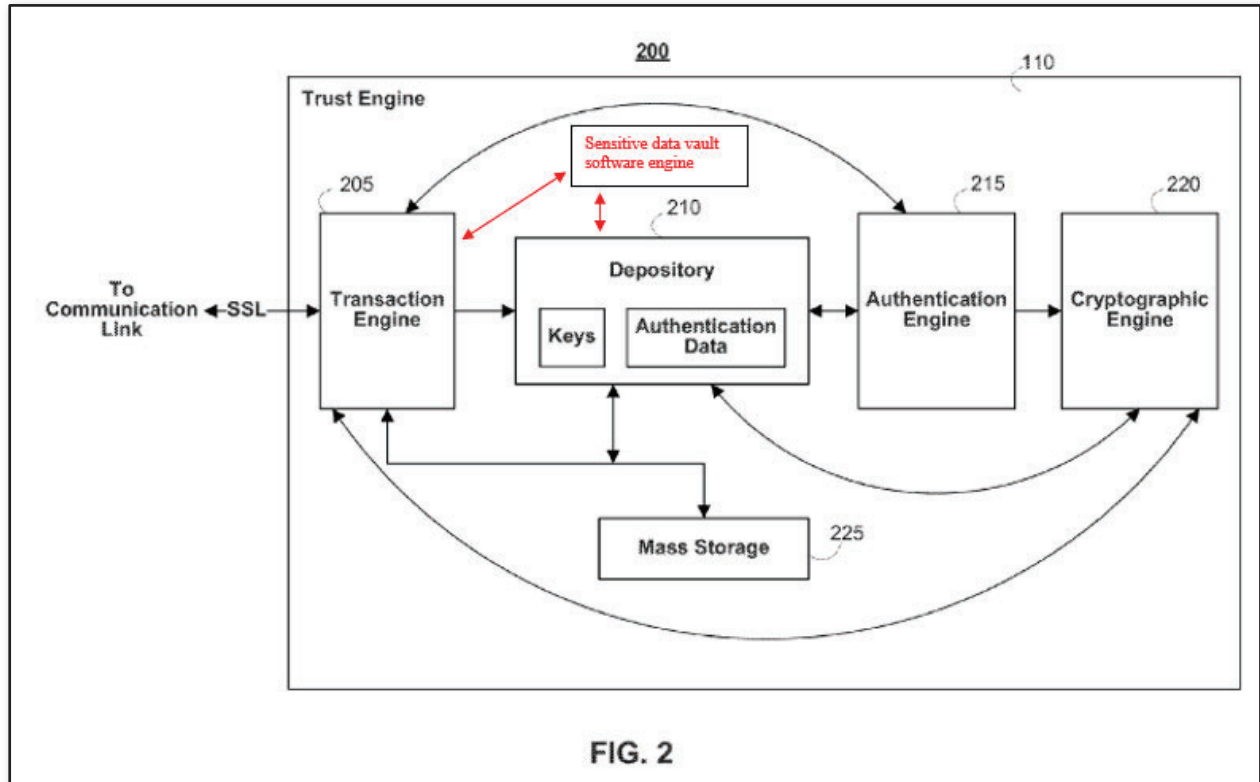
1. Claim 1

a. 1[PRE]: A secure storage network comprising

217. Dickinson-Takeda (which I described in §X.C above) includes Dickinson's cryptographic system, which includes user system 105, vendor system 120, trust engine 110 and certificate authority 115, all connected via communication links (e.g., networks) 125. Dickinson, 7:34-35, 10:20-22, 13:1-4, FIG. 1. This is the same type of cryptographic system disclosed in the '140 patent (EX1001, 6:52-53, FIG. 1; *see also* EX1040, 20, 181), which never uses the term "secure storage network" outside of the claims, and never explains what component(s) of the cryptographic system correspond to the claimed secure storage network.

218. Dickinson's trust engine 110 (which I have annotated in FIG. 2, below) includes multiple components forming a network, including a sensitive data vault software engine controlling storage and retrieval of sensitive data in the depository 210. Dickinson, 10:20-22, 14:31-33. Thus, POSAs understood that at least Dickinson's trust engine is a secure storage network as claimed. If Patent Owner argues that the '140 patent's secure storage network includes additional components of the cryptographic system, then Dickinson has the same components as disclosed in the '140 patent because of Dickinson's identically overlapping

disclosure, and discloses a secure storage network in the same way. *See* EX1040, 29, 36.



Dickinson, FIG. 2 (annotated)

219. For the reason above, Dickinson-Takeda meets 1[pre].

- b. [1A]: a plurality of physical storage devices storing thereon a plurality of shares, the plurality of shares being associated with at least one session key used to secure a dataset

220. Dickinson-Takeda (which I described in §X.C above) includes Dickinson's trust engine depository system 700 having "physically separated" data storage facilities D1-D4, each being a physical storage device. Dickinson, 8:11-12, 17:4-5, 19:16-20, 20:1-3. Though my opinion does not depend on the USPTO's

Examiner's finding, I further note that, as I discussed above in §V.C, the USPTO Examiner similarly cited Dickinson's D1-D4 as physical storage devices (EX1002, 1236), which, Applicant did not rebut (EX1002, 1245-46).

221. As in Dickinson alone, Dickinson-Takeda stores encryption results ("B" and "D") on at least two of storage devices D1-D4. Dickinson, 21:3-9. "B" and "D" each is a "portion" of sensitive data S, with the random number ("A" and "C," respectively) used to do the encryption (XOR) being another "portion," as Dickinson discloses. Dickinson, 21:10-14, 18:20-28, 19:31-34. The '140 patent equates "portions" with "shares" and describes the same XOR encryption process that Dickinson discloses as producing "shares." EX1001, 58:64-59:5 (for "b = a XOR s": "a" and "b" are "shares" or "portions" of "s"); 53:19-28 (splitting data produces portions or shares). Thus, Dickinson-Takeda's data splitting process **801/802** (see §X.C.3 above) produces shares ("B" and "D") stored on physical storage devices D1-D4. Dickinson, 21:3-9. *See also* EX1002, 1236-37 (Examiner citing "portions" resulting from Dickinson's data splitting process 800 as "shares"), 1245-46 (Applicant did not rebut that finding).

222. In Dickinson-Takeda's data splitting process **801/802**, shares "B" and "D" are associated with random numbers "A" and "C" used in the XOR processes, as disclosed in Dickinson. *See* Dickinson, FIG. 8, steps 825 and 830. Each of random numbers "A" and "C" is a key used to secure dataset S because no

encrypted share (B or D) is recoverable without its associated key (A or C, respectively). Dickinson, 21:27-32; EX1001, 58:45-59:14.

223. As in Dickinson alone, the sensitive data S in Dickinson-Takeda can be, for example, password login, financial credentials, or the like. Dickinson, 29:15-16. The sensitive data S is a “dataset” under both parties’ proposed constructions because, as I explain below. Any of a password login or financial credentials represents a collection of related information made up of separate elements (e.g., characters or other data elements) that can be handled as a data unit, and thus meets Google’s construction. EX1041, 9. Any of a password login or financial credentials is similarly a “collection of information for storage,” and thus meets Patent Owner’s construction. EX1042, 6.

224. Each of random numbers “A” and “C” in Dickinson-Takeda’s data splitting process **801/802** is a session key under both parties’ constructions because it is “a key generated each time data is secured” (Google: EX1041, 13) and a “[k]ey created to both secure and restore a specific collection of information” (Patent Owner: EX1042, 10). This is because each random number (A, C) is generated each time data S is secured and is created to secure and restore a respective dataset S, as disclosed in Dickinson. Dickinson, FIG. 8; 20:25-32; EX1001, 32:36-52, 58:45-59:14, 73:41-46. Though my opinion does not depend on the USPTO’s Examiner’s finding, I further note that, as I discussed above in

§V.C, the USPTO Examiner cited to Dickinson’s FIGs. 8 and 12 and page 19, lines 31-34 and page 21, lines 10-14 as disclosing session key (EX1002, 1236-37), which Applicant did not rebut (EX1002, 1245-46).

225. For the reasons above, Dickinson-Takeda meets [1A].

c. [1B]: a secure storage system configured to:

226. Dickinson-Takeda (which I described in §X.C above) includes a secure storage system comprising at least the transaction and data vault engines of Dickinson’s trust engine 110. *See also* §X.D.1.a (1[PRE]) above. The ’140 patent specification does not identify a “secure storage system” but says a data splitting module as used in the trust engine’s authentication and cryptographic engines can be integrated into a “storage system.” EX1001, 66:46-49 (data parser can be integrated into a storage system); 52:60-64 (equating “data splitting module” with “secure data parser”). Dickinson similarly discloses a trust engine with a data vault engine that performs the same data splitting. Dickinson, FIGs. 1, 2, 5-6, 29:22-30. As I discussed above in §VII.D.1.c, the claimed “secure storage system” need not include the physical storage devices because claim 1 recites the secure storage system as working with but being distinct from the “physical storage devices.” Dickinson’s transaction and data vault engines securely store data on physical storage devices D1-D4 by not storing the portions needed to recover

sensitive data S on the same storage device (*see* §X.A above), and are thus a “secure storage system.”

227. For the reasons above, Dickinson-Takeda meets [1B].

228. As I explain below, Dickinson’s transaction and data vault engines perform all the claimed functions of the “secure storage system,” which would have also been performed in Dickinson-Takeda.

- i. **[1B-1]: present to a client device a virtual disk, the virtual disk comprising a directory mapped to the plurality of physical storage devices such that physical locations of the shares are hidden from the client device;**

229. According to Patent Owner, a virtual disk is “a system that simulates or emulates a physical storage device.” EX1042, 8. Dickinson’s depository 210, which may comprise FIG. 7’s storage facilities D1-D4, comprises a single ***logical memory*** structure. Dickinson, 16:28-31, 17:4-5, 29:22-30. That logical memory structure is a virtual disk under Patent Owner’s construction because it is a virtual (“logical”) representation of storage that appears to the user to be a physical disk but where the actual physical data storage facilities (e.g., D1-D4) on which the data is stored are hidden from the user. Dickinson, 16:28-31, 29:12-30:2. Though my opinion is based on a POSA’s understanding of Dickinson’s and Takeda’s disclosures, I further note that, as I discussed above in §V.C, the USPTO Examiner cited depository 210’s logical memory structure as a virtual disk (EX1002, 1237-

39), which Applicant did not rebut (EX1002, 1245-46). Therefore, the logical memory structure simulates or emulates a physical storage device. *See* EX1042, 8 (Patent Owner's constructions).

230. The depository's logical memory structure in Dickinson-Takeda comprises an index indexing a user ID against the user's stored information. Dickinson, 16:28-31. Dickinson describes the index before describing the data vault engine. Dickinson, 16:28-31, 29:22-30. However, a POSA would have understood that in the embodiment that has a data vault engine, the data stored therein is included in the index of the user's data. Indeed, Dickinson at 54:3-4 itself notes that the combinations of disclosed features "will be apparent to the skilled artisan." Alternatively, having the data stored in the data vault engine include the index of the user's data would have been the obvious way to implement what Dickinson describes so that the user can see all the user's stored data in the index and retrieve the data the user desires to access.

231. Dickinson's index is a directory mapped to the data storage facilities D1-D4 so that for each piece of sensitive data S in the index there is a mapping to the particular locations among the data storage facilities D1-D4 where the data is stored. Dickinson, 16:28-31. To the extent Dickinson is read to not expressly disclose a directory with this mapping, that would have been the conventional and obvious way to implement Dickinson's logical memory structure so that each piece

of sensitive data S included in that logical memory structure can be retrieved from data storage facilities D1-D4 when desired.

232. The logical memory structure (virtual disk) with its index (directory) is presented to a client device by Dickinson's data vault engine and transaction engine ("secure storage system"). Dickinson, 29:12-30:2. When the user system 105 requests the sensitive data from the sensitive data vault, the logical memory structure ("virtual disk") is presented to the user system 105 ("client device") by the transaction engine so the user can select the desired sensitive data to access. Dickinson, 29:15-16 ("provide the user access to password login, financial credentials, or the like"), 16:28-31, *generally* 29:12-30:2. To the extent Dickinson is read to not expressly disclose that the logical memory structure is presented to the user system 105, that would have been an obvious way to implement what Dickinson describes because, for example, the data vault stores "passwords for an almost limitless number of systems and applications" (Dickinson, 29:14-16), and a POSA would have found it desirable to provide the user with an index of the user's stored passwords so the user can choose the password of interest to be retrieved from the data vault.

233. My opinion that it was well-known and conventional to present a logical memory structure of a virtual disk to a user is corroborated by EX1038, which illustrates in FIG. 9 a screen shot of "a virtual file drive" "as logical drive

H:” and also illustrates “icons 930 and names 940 for the virtual files and virtual folders in the virtual drive.” EX1038, 21:3-6.

234. For the reasons above, Dickinson-Takeda meets [1B-1] under both parties’ constructions.

ii. [1B-2]: generate the plurality of shares for storage on the plurality of physical storage devices by performing a securing operation on the dataset received from the client device and distributing the dataset in the shares;

235. Dickinson-Takeda’s data vault engine performs the data splitting processes **801/802** (described in §§X.A, X.C above), on the dataset S to generate shares B and D (*see* §X.A above) and distribute them for storage on data storage facilities D1-D4. Dickinson, 21:1-14. Data splitting process **801/802** is a securing operation on the sensitive data S (“dataset”) received from the user system 105, and distributes S in shares, because it splits S into individually undecipherable portions including shares B and D. *Id.* Though my opinion does not depend on the USPTO’s Examiner’s finding, I further note that, as I discussed above in §V.C, the USPTO Examiner cited Dickinson’s FIG. 8 data splitting and distribution as meeting this claim limitation (EX1002, 1238), which the Applicant did not rebut (EX1002, 1245-46).

236. For the reasons above, Dickinson-Takeda meets [1B-2].

iii. [1B-3]: include with each of the plurality of shares data indicative of the at least one session key used to secure the dataset; and

237. Each of alternatives (1)-(3) of Dickinson-Takeda that I discussed in §X.C above meets limitation [1B-3], for the reasons I discuss below.

238. Patent Owner has taken the position in the related district court litigation that “*data indicative of*” a key is met by the key itself. *See* EX1042, 3-4 (Patent Owner construing “data indicative of a key” as including “information that...makes known” or “shows” the key); EX1044, 34 (Patent Owner alleging that storing data indicative of a key is met by storing the key itself). If [1B-3] is construed this way, Dickinson’s technique in alternative (1) meets [1B-3] under that construction because Dickinson’s technique includes with each of the shares (B and D) a key (key C is included with B, and key A is included with D) that meets the claimed “session key used to secure the dataset,” as I discussed above in §VII.D.1.b.ii. Key A secures dataset S by encrypting A to form B, and key C secures dataset S by encrypting C to form D. Dickinson, 20:32-21:3.

239. To the extent Patent Owner argued in prosecution that Dickinson does not include these keys with the shares (EX1002, 1245-46), it is my opinion that the Examiner erred in agreeing with that mischaracterization for the reasons discussed above.

240. Alternatives (2) and (3) of Dickinson-Takeda (*see* §X.C above) meet both parties' proposed constructions, including Google's construing "*data indicative of*" a key as "metadata about" the key. EX1041, 5. In both alternatives (2) and (3), Dickinson-Takeda's data vault engine performs data-splitting process **801/802** that includes with each of the shares (B and D) metadata "MD" about and/or a pointer to a key (A or C) used to encrypt the dataset S. *See* §X.C.3 (illustrating Dickinson, Annotated FIGs. 8 (step **836/837**)). As I discussed above (§X.C), the metadata about the key may include, for example, key length, encryption type, etc. and therefore is "*data indicative of*" the corresponding key under both Patent Owner's interpretation (EX1042, 3-4) and Petitioner's (EX1041, 5-7). The pointer in step **836/837** (*see* §X.C.3) is also data indicative of at least one session key under Patent Owner's interpretation because the pointer "points to" the key (EX1042, 3). *See also* §VII.D.1.c.iii above (where I explain how Ophir-Takeda meets this limitation).

241. As discussed above, both A and C are "session key[s] used to secure the dataset" as claimed. In alternative (2), each of these sessions keys is stored with a data portion other than the one it encrypted. In alternative (3), each is stored with the data portion that it encrypted. *See* §X.C above (where I discuss the Dickinson-Takeda combination). Both of these alternatives meet [1B-3], for the reasons discussed above.

242. For the reasons above, Dickinson-Takeda meets [1B-3] under both parties' proposed constructions.

iv. [1B-4]: reconstitute the dataset from at least a portion of the plurality of shares stored on the physical storage devices in response to a request from the client device for information in the dataset.

243. Dickinson-Takeda's data vault engine reconstitutes the dataset S from at least a portion (e.g., "any two"—Dickinson, 21:23-32) of the shares stored on data storage facilities D1-D4 in response to a request from user system 105 for the sensitive data S (e.g., a password). Dickinson, 21:21-32, 29:22-30:9. *See also* §X.A above (where I discuss Dickinson's disclosure).

244. For the reasons above, Dickinson-Takeda meets [1B-4]. Accordingly, for the reasons I discussed above in this §X.D.1, Dickinson-Takeda meets every limitation of claim 1 and renders it obvious to a POSA.

2. Claim 2: The secure storage network of claim 1, wherein the secure storage system is further configured to generate the plurality of shares by performing a cryptographic operation on the dataset.

245. Dickinson-Takeda's (§X.C) data vault engine performs Dickinson's data splitting to generate encrypted data portions ("shares") B and D. *See* §X.A (discussing Dickinson's disclosure), §X.C (describing the Dickinson-Takeda combination). The XOR operations (steps 825, 830) disclosed in Dickinson that are performed in Dickinson-Takeda are cryptographic operations performed on dataset

S because they encrypt the dataset S into undecipherable portions, as I discussed above in §X.A (explaining that Dickinson’s XOR operation is an encryption operation; citing as corroboration EX1034, 4:25-29). Dickinson, FIG. 8, 20:32-21:1.

246. Thus, in my opinion Dickinson-Takeda meets every limitation of claim 2 and renders it obvious to a POSA under both parties’ claim constructions.

3. Claim 3: The secure storage network of claim 1, further comprising a key management server configured to store workgroup keys.

247. Based on Dickinson, Dickinson-Takeda’s trust engine includes a key management server storing keys. *See* §X.A (discussing Dickinson’s disclosure), §X.C (describing the Dickinson-Takeda combination); Dickinson, FIG. 2 (depository 210 storing keys), 10:20-22, 13:5-7, 14:17-19. Based on Takeda, a POSA would have associated with the keys different access privileges for different groups of people, making them workgroup keys meeting claim 3. *See* §X.B (discussing Takeda’s disclosure), §X.C (describing the Dickinson-Takeda combination). Takeda, FIG. 4 (“access privilege”).

248. Thus, in my opinion Dickinson-Takeda meets every limitation of claim 3 and renders it obvious to a POSA under both parties’ claim constructions.

4. Claim 4: The secure storage network of claim 1, wherein the secure storage system is further configured to provide access to the reconstituted dataset.

249. As in Dickinson, Dickinson-Takeda's data vault engine reassembles (reconstitutes) dataset S from the shares in data storage facilities D1-D4 and presents it to the user system 105 via the transaction engine, thus providing access to the reconstituted dataset. Dickinson, 21:21-32, 29:22-30:9.

250. Thus, in my opinion Dickinson-Takeda meets every limitation of claim 4 and renders it obvious to a POSA under both parties' claim constructions.

5. Claim 5: The secure storage network of claim 1, wherein the secure storage system is further configured to establish a secure connection between the client device and the secure storage system.

251. Dickinson-Takeda's secure storage system establishes via its transaction engine (§X.D.1.c above) a secure connection to user system 105 using *secure* socket layer (SSL) communication. Dickinson, FIGs. 1-2 (disclosing communication links 125 that use SSL), 14:25-27 (explaining that, as POSAs knew, SSL provides secure communication).

252. Thus, in my opinion Dickinson-Takeda meets every limitation of claim 5 and renders it obvious to a POSA under both parties' claim constructions.

6. Claim 6: The secure storage network of claim 1, wherein the shares contain a substantially random distribution of the received dataset.

253. In my opinion, a POSA would have implemented Dickinson-Takeda's data-splitting process to create the shares (B, D) such that they contain a substantially random distribution of dataset S, for the same reasons I explained in §VII.D.6 above (citing as corroboration EX1027, [0106]; EX1028, [0058]; EX1029, [0011]; EX1030, [0016]-[0018]; EX1031, Title, Abstract, [0004]) and because Dickinson teaches that randomization enhanced security. Dickinson, 18:20-28.

254. Thus, in my opinion Dickinson-Takeda meets every limitation of claim 6 and renders it obvious to a POSA under both parties' claim constructions.

7. Claim 7: The secure storage network of claim 1, wherein the secure storage system is configured to reconstitute the received dataset from fewer than all of the shares.

255. Dickinson-Takeda's secure storage system (via its data vault) reconstitutes dataset S from *either* encrypted share B *or* D, which is fewer than all the shares B and D. See §§X.A, X.C above; Dickinson, 21:25-32. Though my opinion does not depend on the USPTO's Examiner's finding, I further note that, as I discussed above in §V.C, the USPTO Examiner similarly cited Dickinson's ability to reconstitute sensitive data from a subset of portions created by

Dickinson's data splitting process (EX1002, 1242), which Applicant did not rebut (EX1245-46).

256. Thus, in my opinion Dickinson-Takeda meets every limitation of claim 7 and renders it obvious to a POSA under both parties' claim constructions.

8. Claim 8: The secure storage network of claim 1, wherein the secure storage system is configured to secure the dataset by encrypting the dataset using the at least one session key.

257. As in Dickinson alone, Dickinson-Takeda's data vault engine secures the dataset S by encrypting it using the at least one session key (A, C—see §§X.A, X.C above). Dickinson, FIG. 8; 29:22-30. Though my opinion does not depend on the USPTO's Examiner's finding, I further note that, as I discussed above in §V.C, the USPTO Examiner similarly found that Dickinson discloses claim 8's limitations (EX1002, 1239), which Applicant did not rebut (EX1002, 1245-46).

258. Thus, in my opinion Dickinson-Takeda meets every limitation of claim 8 and renders it obvious to a POSA under both parties' claim constructions.

9. Claim 9: The secure storage network of claim 1, wherein the secure storage system is configured to secure the dataset by splitting the dataset using the at least one session key.

259. As in Dickinson alone, Dickinson-Takeda's data vault engine secures the dataset S by "splitting" it into "individually unusable encrypted portions" (Dickinson, 21:10-20) using at least one session key (A, C—see *supra* §§X.A, X.C). Dickinson, FIG. 8, 8:13-14, 20:22-23 (FIG. 8 is a "data splitting process").

Though my opinion does not depend on the USPTO's Examiner's finding, I further note that, as I discussed above in §V.C, the USPTO Examiner similarly found that Dickinson discloses claim 9's limitations (EX1002, 1239), which Applicant did not rebut (EX1002, 1245-46).

260. Thus, in my opinion Dickinson-Takeda meets every limitation of claim 9 and renders it obvious to a POSA under both parties' claim constructions.

10. Claim 10: The secure storage network of claim 1, wherein the data indicative of the at least one session key comprises: a portion of the at least one session key or data indicative of a portion of the at least one session key.

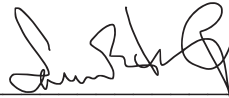
261. In Dickinson-Takeda, the claimed "*at least one session key*" comprises session keys A and C (§X.A above), and the data indicative of the at least one session key includes, e.g., A, pointer P_A, or metadata MD_A (step **835/836/837** described in §X.C.3 above), which is indicative of only session key A and thus indicative of a portion of the "at least one session key" (*A and C*). Takeda, [0040]; Dickinson, FIG. 8.

262. Thus, in my opinion Dickinson-Takeda meets every limitation of claim 10 and renders it obvious to a POSA under both parties' claim constructions.

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

I declare under penalty of perjury that the foregoing is true and correct.

Dated: November 27, 2023



Samrat Bhattacharjee

XI. APPENDIX: CLAIM LISTING

The following claim listing assigns element labels (*e.g.*, 1[PRE], [1A], etc.) to certain claims for convenience of reference.

Claim 1
1[PRE]. A secure storage network comprising
[1A] a plurality of physical storage devices storing thereon a plurality of shares, the plurality of shares being associated with at least one session key used to secure a dataset; and
[1B] a secure storage system configured to:
[1B-1] present to a client device a virtual disk, the virtual disk comprising a directory mapped to the plurality of physical storage devices such that physical locations of the shares are hidden from the client device;
[1B-2] generate the plurality of shares for storage on the plurality of physical storage devices by performing a securing operation on the dataset received from the client device and distributing the dataset in the shares;
[1B-3] include with each of the plurality of shares data indicative of the at least one session key used to secure the dataset; and
[1B-4] reconstitute the dataset from at least a portion of the plurality of shares stored on the physical storage devices in response to a request from the client device for information in the dataset.
Claim 2
2. The secure storage network of claim 1, wherein the secure storage system is further configured to generate the plurality of shares by performing a cryptographic operation on the dataset.
Claim 3
3. The secure storage network of claim 1, further comprising a key management server configured to store workgroup keys.
Claim 4
4. The secure storage network of claim 1, wherein the secure storage system is further configured to provide access to the reconstituted dataset.
Claim 5
5. The secure storage network of claim 1, wherein the secure storage system is further configured to establish a secure connection between the client device and the secure storage system.

Claim 6
6. The secure storage network of claim 1, wherein the shares contain a substantially random distribution of the received dataset.
Claim 7
7. The secure storage network of claim 1, wherein the secure storage system is configured to reconstitute the received dataset from fewer than all of the shares.
Claim 8
8. The secure storage network of claim 1, wherein the secure storage system is configured to secure the dataset by encrypting the dataset using the at least one session key.
Claim 9
9. The secure storage network of claim 1, wherein the secure storage system is configured to secure the dataset by splitting the dataset using the at least one session key.
Claim 10
10. The secure storage network of claim 1, wherein the data indicative of the at least one session key comprises: a portion of the at least one session key or data indicative of a portion of the at least one session key.