

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. TBD
Patent No. 11,178,116

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 '116 PATENT	13
	A. Priority Claim	13
	B. Purported Innovation	13
	C. Specification	17
	D. Person of Ordinary Skill in the Art (“POSA”).....	21
	E. Prosecution History	24
	F. Challenged Claims.....	25
	1. Independent Claims 1 and 9	25
VI.	CLAIM INTERPRETATION	28
VII.	GROUND 1: Tajima (Claims 1, 5-8)	30
	A. Tajima (EX1005).....	30
	B. Claim 1:	38
	1. 1[PRE] A method for securing a data set, the method comprising:	41
	2. [1A] distributing the data set into a plurality of data chunks, wherein none of the data chunks are, by themselves, sufficient to reconstruct the data set	42
	3. [1B] encrypting each of the data chunks with a respective one of a plurality of encryption keys	45
	4. [1C] obfuscating each of the plurality of different encryption keys.....	48

5. [1D] separately storing each data chunk of the plurality of data chunks together with one of the plurality of obfuscated different encryption keys on a plurality of different storage devices	55
C. Claim 5: The method of claim 1, wherein storing the plurality of data chunks together with the plurality of obfuscated different encryption keys comprises storing each of the plurality of data chunks together with a respective one of the plurality of obfuscated different encryption keys	67
D. Claim 6: The method of claim 1, wherein storing the plurality of data chunks together with the plurality of obfuscated different encryption keys comprises including, with a data chunk of the plurality of data chunks, data indicative of an encryption key that was used to encrypt a different data chunk of the plurality of data chunks	69
E. Claim 7: The method of claim 1, wherein distributing the data set into a plurality of data chunks comprises using a random technique or pseudorandom technique.....	74
F. Claim 8: The method of claim 1, further comprising generating the data set by encrypting primary data	76
VIII.GROUND 2: Tajima-Foster (Claims 1-2, 9, 11-14).....	77
A. Storing plaintext keys with encrypted data was a known security risk78	
B. Reasons to encrypt Tajima’s keys	80
C. Foster’s key-encryption technique	85
D. Tajima-Foster Combination.....	87
E. Reasonable Expectation of Success	90
F. Claim 1	92
1. 1[PRE]-[1D]	92
2. [1C] obfuscating each of the plurality of different encryption keys.....	92
G. Claim 2: The method of claim 1, wherein obfuscating each of the plurality of different encryption keys comprises performing a transform on data indicative of at least one of the plurality of different encryption keys.....	94

H. Claim 9	95
1. 9[PRE]-[9B]	96
2. [9C] transforming each of the plurality of different encryption keys.....	96
3. [9D] separately storing each data chunk of the plurality of data chunks together with one of the plurality of transformed different encryption keys on a plurality of different storage devices	97
I. Claim 11: The method of claim 9, wherein storing the plurality of data chunks with the plurality of transformed different encryption keys comprises storing each of the plurality of data chunks with a respective one of the plurality of transformed different encryption keys	98
J. Claim 12: The method of claim 9, wherein storing the plurality of data chunks with the plurality of transformed different encryption keys comprises including, with a data chunk of the plurality of data chunks, data indicative of an encryption key of the plurality of encryption keys that was used to encrypt a different data chunk of the plurality of data chunks	99
K. Claim 13: The method of claim 9, wherein distributing the data set into a plurality of data chunks comprises using a random technique or pseudorandom technique.....	100
L. Claim 14: The method of claim 9, further comprising generating the data set by encrypting primary data	100
IX. GROUND 3: Tajima-Rivest (Claims 9-14).....	100
A. Public Accessibility of Rivest (EX1034)	101
B. Reasons to transform Tajima's key information	103
C. Tajima-Rivest Combination	104
D. Claim 9	107
E. Claim 10: The method of claim 9, wherein transforming each of the plurality of different encryption keys comprises performing an all- or-nothing transform on data indicative of at least one of the plurality of different encryption keys.....	108
F. Claim 11: The method of claim 9, wherein storing the plurality of data chunks with the plurality of transformed different encryption	

	keys comprises storing each of the plurality of data chunks with a respective one of the plurality of transformed different encryption keys	108
G.	Claim 12: The method of claim 9, wherein storing the plurality of data chunks with the plurality of transformed different encryption keys comprises including, with a data chunk of the plurality of data chunks, data indicative of an encryption key of the plurality of encryption keys that was used to encrypt a different data chunk of the plurality of data chunks	109
H.	Claim 13: The method of claim 9, wherein distributing the data set into a plurality of data chunks comprises using a random technique or pseudorandom technique.....	109
I.	Claim 14: The method of claim 9, further comprising generating the data set by encrypting primary data	110
X.	GROUND 4: Tajima-Rivest-Shyu (Claims 1-3, 6)	110
A.	Reasons to Vary Tajima’s Encryption Algorithms	110
B.	Tajima-Rivest-Shyu Combination.....	113
C.	Claim 1	114
D.	Claim 2: The method of claim 1, wherein obfuscating each of the plurality of different encryption keys comprises performing a transform on data indicative of at least one of the plurality of different encryption keys.....	115
E.	Claim 3: The method of claim 2, wherein performing a transformation on data indicative of at least one of the plurality of different encryption keys comprises performing an all-or-nothing transform on the data indicative of at least one of the plurality of different encryption keys.....	116
F.	Claim 6: The method of claim 1, wherein storing the plurality of data chunks together with the plurality of obfuscated different encryption keys comprises including, with a data chunk of the plurality of data chunks, data indicative of an encryption key that was used to encrypt a different data chunk of the plurality of data chunks	117
XI.	GROUND 5: Foster-Hayhurst (Claims 1-2, 5, 9, 11).....	117
A.	Foster (EX1006)	117

B. Hayhurst (EX1007).....	123
1. Summary of Hayhurst	124
2. Hayhurst Provisional	127
C. Foster-Hayhurst	130
1. Reasons to Combine.....	130
2. The Foster-Hayhurst Combination.....	137
3. Reasonable Expectation of Success	141
D. Claim 1	142
1. 1[PRE] A method for securing a data set, the method comprising:	142
2. [1A] distributing the data set into a plurality of data chunks, wherein none of the data chunks are, by themselves, sufficient to reconstruct the data set	143
3. [1B] encrypting each of the data chunks with a respective one of a plurality of encryption keys	145
4. [1C] obfuscating each of the plurality of different encryption keys.....	146
5. [1D] separately storing each data chunk of the plurality of data chunks together with one of the plurality of obfuscated different encryption keys on a plurality of different storage devices	148
E. Claim 2: The method of claim 1, wherein obfuscating each of the plurality of different encryption keys comprises performing a transform on data indicative of at least one of the plurality of different encryption keys.....	150
F. Claim 5: The method of claim 1, wherein storing the plurality of data chunks together with the plurality of obfuscated different encryption keys comprises storing each of the plurality of data chunks together with a respective one of the plurality of obfuscated different encryption keys.....	151
G. Claim 9	151
1. 9[PRE]-[9B]	151
2. [9C] transforming each of the plurality of different encryption keys.....	152

3. [9D] separately storing each data chunk of the plurality of data chunks together with one of the plurality of transformed different encryption keys on a plurality of different storage devices.	153
H. Claim 11: The method of claim 9, wherein storing the plurality of data chunks with the plurality of transformed different encryption keys comprises storing each of the plurality of data chunks with a respective one of the plurality of transformed different encryption keys	154
XII. APPENDIX: CLAIM LISTING	156

APPENDIX LISTING OF EXHIBITS

Exhibit	Description
1001	U.S. Patent No. 11,178,116 (“116 Patent”)
1002	Prosecution History of U.S. Patent No. 11,178,116
1003	Declaration of Samrat Bhattacharjee, Ph.D.
1004	Curriculum Vitae of Samrat Bhattacharjee, Ph.D.
1005	U.S. Patent Application Publication 2003/0028493 (“Tajima”)
1006	U.S. Patent Application Publication 2003/0200176A1 (“Foster”)
1007	International Application Publication WO 2004/008289A2 (“Hayhurst”)
1008	U.S. Provisional Application 60/396357 (“Hayhurst-Provisional”)
1009	Coversheet of U.S. Provisional Application 60/396357
1010	Application Data for Provisional Application 60/396357
1011	Security First Innovation’s 2023-08-14 Infringement Contentions, Chart A in <i>Security First Innovations, LLC v. Google LLC</i> , 2-23-cv-00097 (E.D. Va.)
1012	Security First Innovation’s 2023-08-14 Infringement Contentions, Chart C in <i>Security First Innovations, LLC v. Google LLC</i> , 2-23-cv-00097 (E.D. Va.) (“Contentions”)
1013	U.S. Patent Application Publication 2004/0049687 (“Orsini”)
1014	International Application Publication WO 2001/22322 (“Dickinson”)
1015	U.S. Patent No. 10,452,854 (“854 Patent”)
1016	U.S. Patent No. 6,021,391 (“Shyu”)
1017	U.S. Patent Application Publication 2004/0049700 (“Yoshida”)
1018	U.S. Patent Application Publication 2004/0003272A1 (“Bantz”)
1019	International Application Publication WO 2001/015162A2 (“Benaloh”)
1020	U.S. Publication No. 2002/0194209 (“Bolosky”)
1021	U.S. Patent No. 6,819,766 (“Weidong”)
1022	U.S. Patent No. 4,238,853 (“Ehrsam”)
1023	U.S. Patent No. 7,277,941 (“Ignatius”)
1024	U.S. Patent Application Publication 2005/0216754 (“Ehud”)
1025	International Application Publication WO 2004/057461A2 (“Ogihara”)
1026	U.S. Patent No. 7,178,033 (“Garcia”)
1027	U.S. Patent No. 6,490,353 (“Tan”)
1028	U.S. Patent Application Publication No. 2003/0167314 (“Marking”)
1029	U.S. Patent No. 7,529,834 (“Birrell”)
1030	U.S. Patent Application Publication 2001/0037474A1 (“Steegmans”)
1031	U.S. Patent Application Publication 2004/0172476A1 (“Chapweske”)

1032	U.S. Patent No. 7,099,994 (“Thayer”)
1033	U.S. Patent No. 6,151,659 (“Solomon”)
1034	R. L. Rivest, “ <i>All-or-nothing encryption and the package transform</i> ,” In Fast Software Encryption 1997, (E. Biham, ed.), Lecture Notes in Computer Science, 1267 (1997) pp. 210–218. (“Rivest”)
1035	U.S. Patent No. 6,308,277 (“Vaeth”)
1036	U.S. Patent Publication 2001/0053221A1 (“Takeda”)
1037	Security First Innovation’s 2023-08-14 Infringement Contentions, Chart D in Security First Innovations, LLC v. Google LLC, 2-23-cv-00097 (E.D. Va.)
1039	U.S. Patent Application Publication 2018/0367509A1 (“116 Publication”)
1040	ABS: The Apportioned Backup System (“Cooley”)
1041	U.S. Patent Publication 2003/0081790A1 (“Kallahalla”)
1042	Google LLC’s Proposed Claim Terms for Construction in Security First Innovations, LLC v. Google, LLC, 2:23-cv-00097-JWK-RJK (Nov. 11, 2023)
1043	Letter from A. Mayron to D. Perlson re Preliminary Proposed Constructions in Security First Innovations, LLC v. Google, LLC, 2:23-cv-00097-JWK-RJK (Nov. 11, 2023)
1044	U.S. Patent No. 9,338,140 (“140 patent”)

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, 2, 5-9, 11-14 (the “challenged claims”) of U.S. Patent No. 11,178,116 (“the ’116 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 ’116 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 Electrical and Computer Engineering at the University of Maryland. I have been an appointed faculty member at the University of Maryland since 1999.

4. 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.

5. 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.

6. 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 US patent (US 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.

7. 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.

8. 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.

9. 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.

10. 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.

11. 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 Section V.A below. In forming my opinions, I have studied and considered the materials identified in the list below.

Exhibit	Description
1001	U.S. Patent No. 11,178,116 (“116 Patent”)
1002	Prosecution History of U.S. Patent No. 11,178,116
1005	U.S. Patent Application Publication 2003/0028493 (“Tajima”)
1006	U.S. Patent Application Publication 2003/0200176A1 (“Foster”)
1007	International Application Publication WO 2004/008289A2 (“Hayhurst”)
1008	U.S. Provisional Application 60/396357 (“Hayhurst-Provisional”)
1011	Security First Innovation’s 2023-08-14 Infringement Contentions, Chart A in <i>Security First Innovations, LLC v. Google LLC</i> , 2-23-cv-00097 (E.D. Va.)
1012	Security First Innovation’s 2023-08-14 Infringement Contentions, Chart C in <i>Security First Innovations, LLC v. Google LLC</i> , 2-23-cv-00097 (E.D. Va.) (“Contentions”)
1013	U.S. Patent Application Publication 2004/0049687 (“Orsini”)
1014	International Application Publication WO 2001/22322 (“Dickinson”)
1015	U.S. Patent No. 10,452,854 (“854 Patent”)
1016	U.S. Patent No. 6,021,391 (“Shyu”)
1017	U.S. Patent Application Publication 2004/0049700 (“Yoshida”)
1018	U.S. Patent Application Publication 2004/0003272A1 (“Bantz”)
1019	International Application Publication WO 2001/015162A2 (“Benaloh”)
1020	U.S. Publication No. 2002/0194209 (“Bolosky”)
1021	U.S. Patent No. 6,819,766 (“Weidong”)
1022	U.S. Patent No. 4,238,853 (“Ehrsam”)
1023	U.S. Patent No. 7,277,941 (“Ignatius”)
1024	U.S. Patent Application Publication 2005/0216754 (“Ehud”)
1025	International Application Publication WO 2004/057461A2 (“Ogihara”)
1026	U.S. Patent No. 7,178,033 (“Garcia”)
1027	U.S. Patent No. 6,490,353 (“Tan”)
1028	U.S. Patent Application Publication No. 2003/0167314 (“Marking”)

1029	U.S. Patent No. 7,529,834 (“Birrell”)
1030	U.S. Patent Application Publication 2001/0037474A1 (“Steegmans”)
1031	U.S. Patent Application Publication 2004/0172476A1 (“Chapweske”)
1032	U.S. Patent No. 7,099,994 (“Thayer”)
1033	U.S. Patent No. 6,151,659 (“Solomon”)
1034	R. L. Rivest, “ <i>All-or-nothing encryption and the package transform</i> ,” In Fast Software Encryption 1997, (E. Biham, ed.), Lecture Notes in Computer Science, 1267 (1997) pp. 210–218. (“Rivest”)
1035	U.S. Patent No. 6,308,277 (“Vaeth”)
1036	U.S. Patent Publication 2001/0053221A1 (“Takeda”)
1037	Security First Innovation’s 2023-08-14 Infringement Contentions, Chart D in Security First Innovations, LLC v. Google LLC, 2-23-cv-00097 (E.D. Va.)
1039	U.S. Patent Application Publication 2018/0367509A1 (“116 Publication”)
1040	ABS: The Apportioned Backup System (“Cooley”)
1041	U.S. Patent Publication 2003/0081790A1 (“Kallahalla”)
1042	Google LLC’s Proposed Claim Terms for Construction in Security First Innovations, LLC v. Google, LLC, 2:23-cv-00097-JWK-RJK (Nov. 11, 2023)
1043	Letter from A. Mayron to D. Perlson re Preliminary Proposed Constructions in Security First Innovations, LLC v. Google, LLC, 2:23-cv-00097-JWK-RJK (Nov. 11, 2023)
1044	U.S. Patent No. 9,338,140 (“140 patent”)

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 ’116 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 ’116 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;
- use 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, suggestion, 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 obvious 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	Tajima (EX1005)	1, 5-8	Anticipation and/or Obviousness
2	Tajima and Foster (EX1006)	1-2, 9, 11-14	Obviousness
3	Tajima and Rivest (EX1034)	9-14	Obviousness
4	Tajima, Rivest and Shyu (EX1016)	1-3, 6	Obviousness
5	Foster and Hayhurst (EX1007)	1-2, 5, 9, 11	Obviousness

28. I understand that Tajima, Foster and Rivest are prior art (at least) based on the following publication dates, each of which is more than one year before the earliest-alleged priority date (2004-10-25):

Reference	Publication Date
Tajima	2003-02-06 (EX1005, code (43))
Foster	2003-10-23 (EX1006, code (43))
Shyu	2000-02-01 (EX1036, code (45))
Rivest	1997 (EX1034, 1-4)

29. I understand that Hayhurst is prior art based on any of the following publication, filing and priority dates, each of which is before the earliest-alleged priority date (2004-10-25):

Reference	Prior Art Dates
Hayhurst	Publication Date: 2004-01-22 (EX1007, code (43)) Filing Date: 2003-07-16 (EX1007, code (22)) Priority Date: 2002-07-17 (EX1007, code (30))

V. THE '116 PATENT

A. Priority Claim

30. I understand the '116 patent (EX1001) was filed on May 11, 2018 (EX1001, code (22)) and claimed priority through a series of continuation applications to a first non-provisional applications filed October 25, 2005 (EX1001, code (60)), which claimed priority to two provisional applications, the earliest of which was filed on October 25, 2004 (EX1001, code (60)). I understand that October 25, 2004 is the earliest priority date to which the '116 patent alleges priority (earliest-alleged priority date).

B. Purported Innovation

31. In the “Summary of the Invention,” the '116 patent alleges that “one aspect of the present invention” is a method for securing data from unauthorized access or use, where the method may involve three basic steps: (1) “parsing, splitting, and/or separating the data to be secured into two or more parts or portions,” (2) encrypting the data “prior to or after” the “parsing, splitting and/or

separating of the data,” and (3) “storing the parsed, split, and/or separated data that has been encrypted in one location or in multiple locations.” EX1001, 2:39-57.

32. Securing data by splitting the data into portions, encrypting those portions, and storing the encrypted data portions in different locations was well-known and routine, both as individual data security techniques and together as a combined data securing strategy. My opinion in this regard is corroborated by the prior art references that I briefly summarize below that each disclose the above conventional data security techniques.

- EX1005 (Tajima) discloses dividing a data set (e.g., a user’s personal information) into “a plurality of data portions” and distributing those data portions to multiple “data fragments.” Tajima (EX1005), [0111], [0113]. Each data fragment is then encrypted and stored on a database under the “separate control” of respective different service sites. EX1005, [0046]-[0048], [0111], [0113].
- EX1014 (Dickinson) discloses splitting sensitive data into a plurality of data portions using different keys and storing encrypted data portion/key pairs on different data storage facilities. EX1014, 20:20-22:7, FIGS. 7-8.
- EX1013 (Orsini) discloses dividing data to be secured into a plurality of shares, encrypting each of the shares, and storing each share with a key

that did not encrypt it on respective different storage depositories.

EX1013, [0293]-[0368].

- EX1007 (Hayhurst) discloses a “decentralized media delivery” system in which “a media file” is divided into segments that are “each encrypted” and “distributed among the network subscriber units.” EX1007, (54), [0006]-[0007]. EX1007 discloses that the “segmented and distributed storage of the media file provides a high level of security and deterrence to piracy.” EX1007, [0008], [0016]-[0031].
- EX1017 (Yoshida) discloses a “distributive storage controller for providing a data storage service to a plurality of computer devices connected to a network.” EX1017, [0007]. To securely store a file, the storage controller “divides and encrypts the data into a plurality of encrypted pieces” and “stores the encrypted pieces to selected corresponding computer devices” (*id.*) that are connected to a network, for example, a local area network (LAN) or the Internet. EX1017, [0007]-[0009], [0021]-[0031], [0047], [0057].
- EX1024 (Ehud) discloses “secure” and “trusted delivery or archive of digital items through the Internet” ([0001]) by “splitting” data into multiple “fragments” (based on “random parameters” such as “the number of fragments, fragment size and the method for selecting data

input for each fragment”) that are “distributed by the encryption system to designated servers at different network locations” ([0022]). EX1024, [0021]-[0025], FIG. 3 (illustrating that the data is split before encryption).

- EX1018 (Bantz) discloses a secure back-up system that includes splitting a file into M segments, encrypting each of the segments, encrypting the encryption key(s), storing the encrypted key(s) with the data segments and storing the M segments on different clients connected to the network. EX1018, [0029]-[0038].
- EX1040 (Cooley) discloses dividing a file to be backed-up into fragments (4.1.2 “Coding and Fragmenting”), encrypting each fragment with an encryption key that is added to the fragment’s metadata (4.1.4 “Encryption”), encrypting the fragment’s metadata to protect the encryption key (4.1.4 “Encryption”), and storing each fragment of the file on a different servers—i.e., no two fragments from the same file are “placed on the same physical machine” (4.1.2 “Coding and Fragmenting”). EX1040, 3-4.
- *See also* EX1025 (Ogihara), 7:2-8:1 (disclosing storing encrypted divided data on different server); EX1041 (Kallahalla), [0027]-[0051] (disclosing encrypting different file fragments with different keys);

EX1036 (Takeda), [0040]-[0054] (disclosing encrypting different file blocks with different encryption keys).

C. Specification

33. The description in the '116 patent's specification that most closely relates to the subject matter of the challenged claims is described in the embodiments illustrated in Figs. 21-24 and described at 57:10-61:13 and 77:13-43 of the '116 patent's specification. I have provided below a summary of this disclosure and identified how it relates to the subject matter of the independent claims.

34. Fig. 21 (which I have copied and annotated below) illustrates a process of securing a data set by encrypting the data set ("Encrypt" step) followed by: (1) dividing the data set into different "shares or portions of parsed data" (EX1001, 57:47-48) (the "Parse" step boxed in green); (2) "[e]ncrypting each share" using a different encryption key (EX1001, 57:59) (the "Encrypt Pieces" step boxed in blue); and (3) assigning each encryption key to a share it did not encrypt (the "Obfuscate" step boxed in red), which the specification describes as "store the encryption keys in different locations from the encrypted data portions or shares: Share 1 gets Key 4, Share 2 gets Key 1, Share 3 get Key 2, Share 4 gets Key 3" (EX1001, 57:59-62).

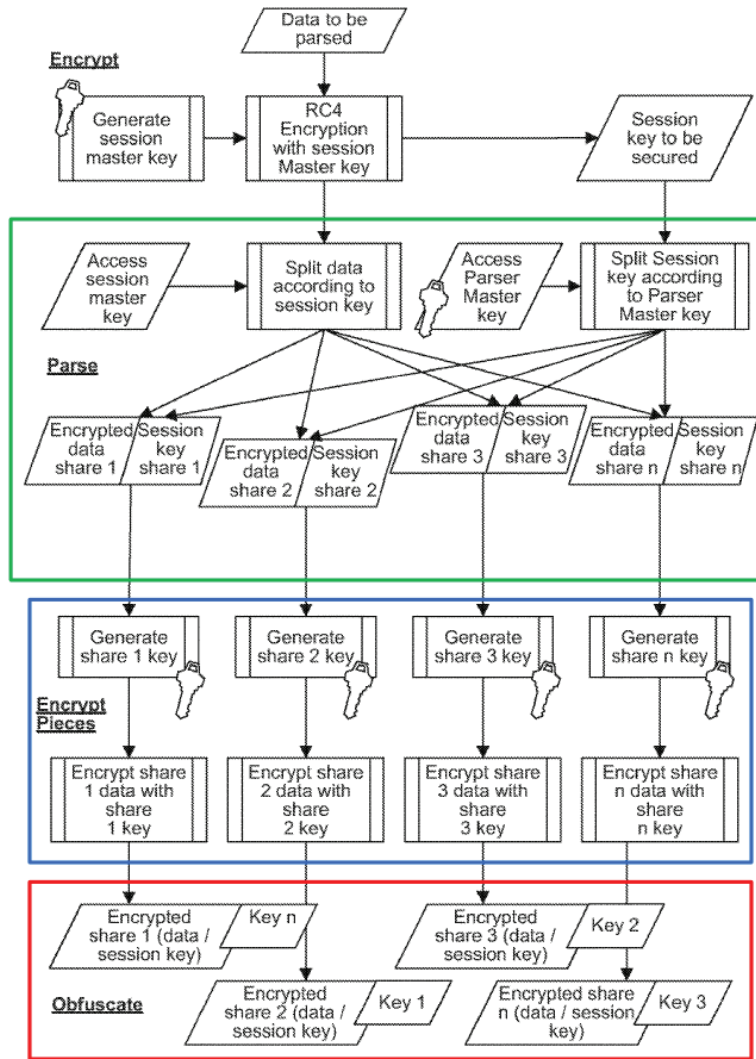


FIG. 21

35. I note that steps (1) and (2) (i.e., the “Parse” and “Encrypt Pieces” steps) correspond to the “distributing” and “encrypting” steps in each of independent claims 1 and 9 and step (3) (i.e., the “Obfuscate” step) corresponds to the “obfuscating” step in claim 1 as further discussed in §V.F below (“Challenged Claims”).

36. Similarly, Figs. 22-24 illustrate this same basic securing technique in which a data set is optionally first encrypted and divided into shares/portions that

are each encrypted with a respective encryption key that are each kept with a share/portion that the key did not encrypt. EX1001, 58:8-59:9. Each of the Fig. 22-24 embodiments describes the step labeled “**Obfuscate**” in the drawing in the same way: “store the encryption keys in different locations from the encrypted data portions or shares,” which is shown as “Share 1 gets Key 4, Share 2 gets Key 1, Share 3 gets Key 2, Share 4 gets Key 3” just like in the Fig. 21 embodiment. EX1001, 57:59-62, 58:32-36; 59:7-11, 59:54-57.

37. To “store the encryption keys in different locations from the encrypted data portions or shares” (*id.*), the ’116 patent discloses storing each key paired with a non-corresponding share (i.e., a share that was not encrypted by the key to which it is paired) resulting from the “**Obfuscate**” step on different depositories “so the key required to decrypt the share is stored in a different data depository.” EX1001, 60:49-52. One technique for achieving this is shown in the specification at 60:53-56 (boxed in **purple** below) where SD1-SD n are n shares that each include respective encrypted data portions of a data set S, and where K1-K n are the n respective encryption keys used to encrypt the same-numbered data portion so that each depository stores a share paired with a key that was used to encrypt a different share.

Depository 1: SD1, Kn
Depository 2: SD2, K1
Depository 3: SD3, K2
Depository n: SDn, K3.

38. This disclosure corresponds to the “separately storing” limitation in independent claims 1 and 9. This very same technique of dividing a data set into portions, encrypting each of the data portions with a different encryption key and storing the encrypted data portions with a key that encrypted a different data portion on different respective storage devices is disclosed in Tajima, as I discuss in detail in Ground 1 below (§VII).

39. The ’116 patent further explains that prior to parsing, the data set can be first encrypted (e.g., the “**Encrypt**” in step in annotated Fig. 21 above). EX1001, 57:45-46 (“Generating a session master key and encrypt the data using RS1 stream cipher”), 58:3-32 (same), 58:58-59 (“Optional encryption of the data using an existing or new encryption algorithm keyed with the Intermediary Key”), 59:34-35 (same). In addition to pre-parsing encryption, the ’116 patent explains that encrypting each of the “shares or portions of parsed data” (EX1001, 57:47-48, 58:16-17, 58:61-62, 59:37-38) is also optional (EX1001, 59:4-7, 59:52-54) and that it is “readily apparent to those of ordinary skill in the art that certain steps” of the securing methods illustrated in Figs. 21-24 “may be performed in different order, or repeated multiple times, as desired.” EX1001, 57:64-66; 58:38-41, 59:13-16,

59:59-62. Thus, not only does the '116 patent use well-known splitting and encryption techniques, it acknowledges that a POSA would have known that these conventional techniques can be applied in any order, can be repeated and are optional.

40. The '116 patent also discloses that “an All or Nothing Transform (AoNT)” may be used to transform “splitting keys, encryption keys, [or] any other suitable data elements” (EX1001, 77:13-17) and that “it will be understood that the AoNT may be applied to encryption keys used to encrypt the data portions following parsing and splitting” (EX1001, 77:37-39). This corresponds to the “transforming” limitation in claim 9. AoNT was a known technique (as the '116 patent acknowledges) developed by Rivest (EX1034) to minimize (or eliminate) the computational feasibility of brute-force attacks, as discussed in further detail in Ground 3 below (§IX). Rivest, 212-214.

D. Person of Ordinary Skill in the Art (“POSA”)

41. 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.

42. The '116 patent involves technology in the field of securing data from unauthorized access or use. *See* EX1001, 1:21-22 (“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. I use this timeframe because the face of the '116 patent (EX1001, codes (22) and (60) indicates an earliest claimed priority date of October 25, 2004 as I discussed above at the beginning of §V. 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 '116 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 timeframe, even if I do not say so specifically in each case. Demonstrating that the '116 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. 8,266,438, which the '116 patent claims priority to), because the POSA's knowledge only increases over time.

43. 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 for any claims in the '116 patent.

44. In my opinion, the person of ordinary skill in the art (POSA) at the time of the '116 patent's October 24, 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 access or use. *See, e.g.*, EX1001, 1:21-22. A higher level of education may substitute for less experience. This person would have been capable of understanding and applying the teachings of the '116 patent and the prior-art references discussed in this declaration.

45. By 2004, I held a Ph.D. in computer science from the Georgia Institute of Technology, and I had over 10 years of experience with cryptographic techniques for computer systems to secure data from unauthorized access or use. 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.

E. Prosecution History

46. The '116 patent's specification was filed with one claim. EX1002, 453. This single claim (claim 1) was canceled and new claims 2-25 were added in a "First Preliminary Amendment" (EX1002, 246-252). The added claims 2-25 were amended once in a "Reply to Final Office Action" ("Amendment") responding to a final Office Action ("FOA") in which the Patent Office rejected all of the claims as obvious over Dickinson (EX1014) and Bolosky (EX1020). EX1002, 133-155 (FOA), 117-125 (Amendment).

47. Specifically, claims 2 and 10 (which issued as claims 1 and 9) were amended as follows:

"separately storing each data chunk of the plurality of data chunks together with one of the plurality of obfuscated [claim 2]/transformed [claim 10] different encryption keys on a plurality of different storage devices." (EX1002, 118-119).

48. Claims 16 and 22 (which issued as claims 15 and 21) were similarly amended (EX1002, 120-122).

"separately store each data chunk of the plurality of data chunks together with one of the plurality of obfuscated [claim 16]/transformed [claim 21] different encryption keys, wherein the plurality of data chunks is stored on a plurality of different storage devices." (EX1002, 120-122).

49. The Patent Owner alleged that “[c]laims 2, 10, 16, 22 are amended to clarify the claimed subject matter” (EX1002, 123) and argued patentability by alleging the amended claims distinguished Bolosky. EX1002, 124 (“there is no disclosure in Bolosky that recites separately storing each data chunk with one of the plurality of obfuscated different encryption keys...”) (Emphasis original).

50. The claims were allowed without further amendment and issued as claims 1-24 of the ’116 patent. EX1002, 89-99.

F. Challenged Claims

51. The Challenged Claims of the ’116 patent (claims 1-3 and 5-14) are reproduced in the Claim Listing Appendix (§XII). Independent claims 1 and 9 are reproduced below with the limitations labeled 1[PRE]-[1D] in the manner they are labeled in the Claim Listing Appendix.

1. Independent Claims 1 and 9

52. The challenged independent claims recite similar subject matter, but differ in whether encryption keys are “obfuscated” (claim 1) or “transformed” (claim 9), as illustrated below with the different “obfuscating” and “transforming” language emphasized in italic and bold.

1[PRE]. A method for securing a data set, the method comprising:

[A] distributing the data set into a plurality of data chunks, wherein none of the data chunks are, by themselves, sufficient to reconstruct the data set;

[B] encrypting each of the data chunks with a respective one of a plurality of encryption keys;

[C] *obfuscating* each of the plurality of different encryption keys;
and

[D] separately storing each data chunk of the plurality of data chunks together with one of the plurality of *obfuscated* different encryption keys on a plurality of different storage devices.

9[PRE] A method for securing a data set, the method comprising:

[A] distributing the data set into a plurality of data chunks, wherein none of the data chunks are, by themselves, sufficient to reconstruct the data set;

[B] encrypting each of the data chunks with a respective one of a plurality of encryption keys;

[C] *transforming* each of the plurality of different encryption keys;
and

[D] separately storing each data chunk of the plurality of data chunks together with one of the plurality of *transformed* different encryption keys on a plurality of different storage devices.

53. Thus, the common limitations shared by claims 1 and 9 (i.e., limitations [A]-[B] and [D]) recite a data securing process comprising (1) dividing data into “data chunks,” (2) encrypting the data chunks with respective encryption keys, and (3) storing the data chunks with the encryption keys on different storage devices, all of which was routine and conventional as corroborated by numerous

prior art references that I briefly summarized above in §V.B (see, e.g., EX1005, [0046]-[0048], [0111], [0113]; EX1014, 20:20-22:7; EX1013, [0293]-[0368], EX1018, [0029]-[0038]; EX1040, 3-4).

54. “Obfuscating” encryption keys (claim 1) and “transforming” encryption keys (claim 9) were also well-known techniques utilized to protect the keys that protect the data being secured. For example, Tajima (EX1005), in addition to the above-enumerated limitations, also discloses storing each different encryption key used in encrypting respective data fragments with a data fragment that it did not encrypt—i.e., Tajima obfuscates encryption keys in the same way the ’116 patent obfuscates them as I discussed in §V.C. above. Tajima thus anticipates claim 1 as demonstrated in Ground 1. *See also*, EX1014, 20:20-22:7 and EX1013, [0293]-[0368] disclosing “obfuscating” encryption keys in this same way; EX1036, [0053]-[0054] (attaching ciphering keys and attributes to different data blocks at random).

55. Similarly, “transforming” (claim 9) was also a known technique used to secure data. Indeed, as I discussed in §V.C above in the discussion of the ’116 patent’s specification, the ’116 patent acknowledges that AoNT techniques were known and that the applicability of AoNT to transforming keys was understood—i.e., obvious. As Ground 3 demonstrates, claim 9 would have been obvious in

view of Tajima and Rivest (EX1034), which is the seminal reference describing the author's development of AoNT.

56. Additionally, I have reviewed Patent Owner's complaint (EX1011) and infringement contentions (EX1012) and understand from my review that Patent Owner argues in the district court litigation that both "obfuscating" and "transforming" encryption keys are met by encrypting the encryption keys (see e.g., EX1012, 5, 8, 12). Encrypting keys was also a routine and conventional technique for protecting encryption keys, for example, as taught by Foster (EX1006) and other. See EX1006, [0064]-[0068]; EX1016, 4:21-24; EX1021, 1:66-2:4; EX1023, 18:28-31; EX1040, 4; EX1018, [0032], each disclosing the routine and conventional use of encrypting encryption keys to protect the keys that protect the data. As Grounds 2 and 5 demonstrate, claims 1 and 9 would have been obvious in view of Tajima and Foster (Ground 2) and Foster and Hayhurst (Ground 5) at least under Patent Owner's claim interpretations.

57. In my opinion, the challenged dependent claims recite conventional features that were well-known and disclosed in the grounds' references.

VI. CLAIM INTERPRETATION

58. 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 (EX1042, EX1043) and/or under the claim scope that Patent Owner ascribes to the claims in its infringement contentions (EX1012). In my opinion, Petitioner’s proposed constructions (EX1064) 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).

VII. GROUND 1: TAJIMA (CLAIMS 1, 5-8)

59. I have been asked to provide my opinion concerning whether claims 1 and 5-8 of the '116 patent are anticipated by Tajima or would have been obvious to a POSA in light of Tajima and the knowledge of a POSA. For the reasons explained below (§§VII.B-VII.F), it is my opinion that each of claims 1 and 5-8 is anticipated by Tajima and/or would have been obvious to a POSA over Tajima.

A. Tajima (EX1005)

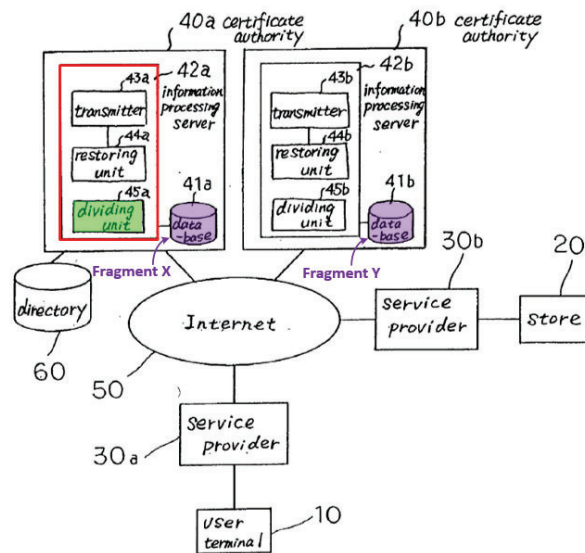
60. Tajima published on February 6, 2003 (code (43)), which I understand means it is prior art to the '116 patent even if the '116 patent were entitled to claim priority to its earliest-alleged priority date of October 24, 2004. Tajima was not cited during prosecution of the '116 patent (code (56)).

61. Tajima discloses techniques for securing personal information submitted by a user to facilitate secure online purchases of goods or services over the Internet. Tajima, [0067]-[0074], [0100]-[0114]. Tajima noted that previous techniques for securely storing a user's personal information "with one area that is connectible to the Internet" had a single point-of-failure vulnerability to "unauthorized access from the outside through a security hole or by unauthorized access by someone inside the system." Tajima, [0042]. That is, breach of the single location at which a user's personal information was secured would have procured the entirety of the user's personal information.

62. To address this vulnerability, Tajima’s system (1) divides personal information into data fragments “such that each data fragment is completely unintelligible when taken independently” (Tajima, [0072], [0122]) and (2) stores (i.e., “registers”) each data fragment in “areas that are each under separate control” (Tajima, [0048])—i.e., different databases belonging to separate service sites connected to the Internet—so a security breach at any one area is insufficient to recover a user’s personal information. That is, an attacker that successfully gained unauthorized access to a service site could procure only a single “unintelligible” data fragment and the attacker would not be able to recover the user’s personal information. Tajima, [0042], [0046]-[0048], [0067], [0118], [0152].

63. Tajima’s Fig. 5 (copied below with colored annotations added) illustrates an example of a “personal information management system” (PIMS) in which personal information submitted by a user via terminal 10 (e.g., to conduct an online transaction with online store 20 via Internet 50) is divided into multiple data fragments that are stored separately on different databases “under separate control” of respective service sites, which are shown as certificate authorities in Tajima’s Fig. 5 system. Tajima, [0048], [0054], [0067]-[0068], [0073]-[0074], [0079].

Fig. 5

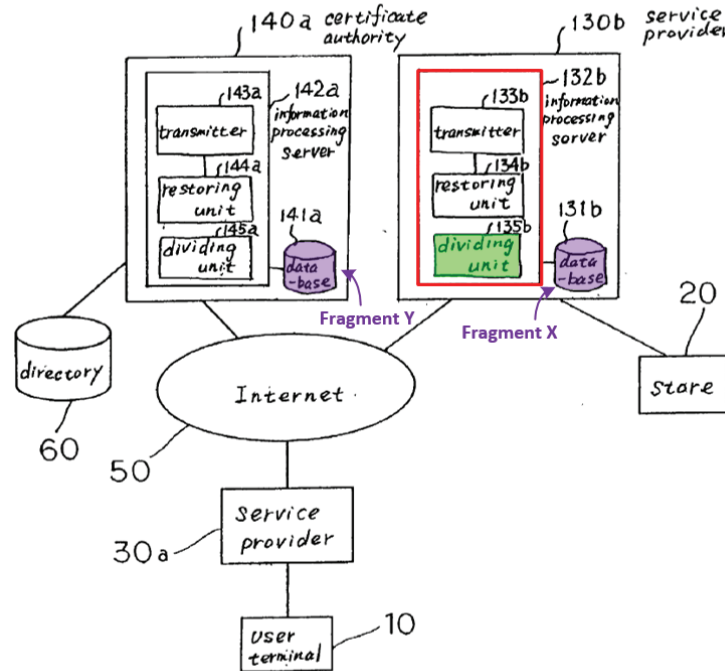


64. Specifically, certificate authority 40a comprises information processing server 42a (boxed red) to secure the personal information via a dividing unit 45a (highlighted green) configured to: (1) divide a user's personal information into data fragments X and Y; (2) store data fragment X on database 41a (highlighted purple); and (3) send data fragment Y via Internet 50 to certificate authority 40b for storage on database 41b (also highlighted purple). Tajima, [0067], [0073]-[0074].

65. Tajima's Fig. 14 (copied below with annotations) illustrates a similar system but where service provider 130b associated with store 20 is the service site that secures the user's personal information. That is, dividing unit 135b implemented on service provider 130b's information processing server 132b divides the user's personal information into data fragments that are then securely

stored on separate databases 131b and 141a of the respective service sites (here, service provider 130b and certificate authority, respectively). Tajima, [0118].

Fig. 14



66. Each of Tajima's service sites (whether certificate authority or service provider) has its own information processing server comprising a dividing unit configured to divide personal information into data fragments that are then distributed to one or more other service sites for storage. Tajima, [0048], [0067]-[0068], [0073]-[0074], [0079], [0093], [0118], [0122]-[0123], [0151]-[0153], [0156]. In this way, each service site can divide personal information and distributively store the resulting data fragments (via its dividing unit), and can receive and store data fragments on its database that were divided by a different service site. *Id.* Thus, the difference between the example systems illustrated in

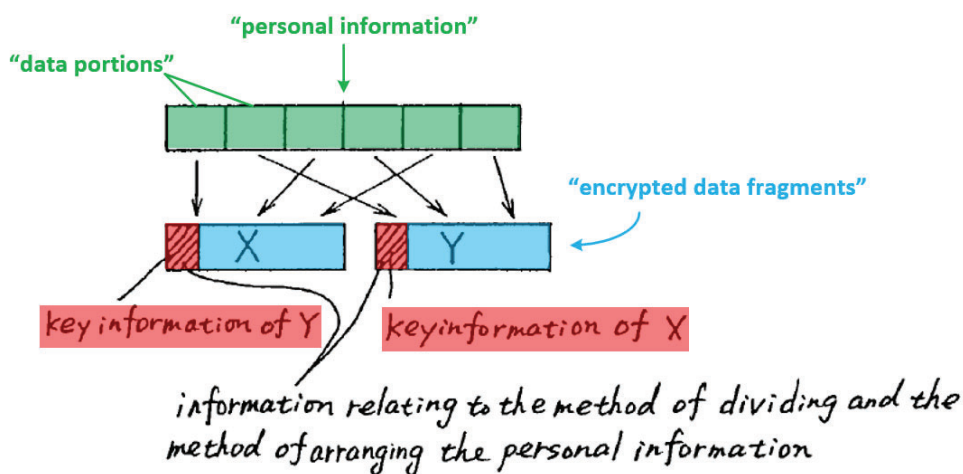
Fig. 5 and 14 is the type of service site that is tasked with dividing the user's personal information and distributively storing the resulting data fragments at different respective service sites.

67. To recover the securely stored data fragments, the service site that divided the personal information stores "link information" that identifies where each data fragment is stored so that the service site can retrieve and recover the personal information when requested. Tajima, [0046]-[0047], [0073], [0093], [0111]-[0114], [0116], [0123], [0144], [0153]. Tajima further discloses two different strategies for distributively storing encrypted data fragments that differ in whether the service site that divided a user's personal information stores one of the resulting data fragments or whether all the resulting data fragments are stored at other service sites. *Id.* Specifically, in a first distributive storage strategy, the service site that divides the user's personal information stores one of the resulting data fragments and link information to the other service sites at which data fragment are stored. Tajima, [0073], [0093], [0123], [0144]. In a second distributive storage strategy, the service site that divides the user's personal information stores "only link information" to other service sites that store all of the resulting data fragments. Tajima, [0116], [0153]. Additionally, Tajima further discloses that link information held by the service site that divided the personal information may be encrypted. Tajima, [0093].

68. Tajima describes different “method[s] of dividing personal information” (Tajima, [0100]) illustrated in Figs. 8-13 that may be performed by the service site, using its dividing unit, dividing the user’s personal information and securely storing the resulting data fragments on databases of different service sites, either in the system configuration of Fig. 5 or Fig. 14. Tajima, [0100]-[0116], [0118], [0122], [0156]. That is, any service site (e.g., whether it be a certificate authority, service provider, etc.) can be configured to perform any of the Fig. 8-13 embodiments using either distributive storage strategy discussed above using any number of data fragments. *Id.* I discuss the Fig. 12 and 13 embodiments in further detail below.

69. In the specific dividing method of Tajima’s Fig. 12 embodiment (copied and annotated below), a user’s personal information is “first divided into a plurality of data portions” (annotated **green**) that are then “arranged into two data fragments, data fragment X and data fragment Y, according to a set method of arrangement.” Tajima, [0111].

Fig. 12



70. Next, “[d]ata fragments X and Y are then each encrypted” with a respective key (encrypted data fragments highlighted **blue**), key information of fragment Y is attached to encrypted data fragment X, and key information for fragment X is attached to encrypted data fragment Y (as highlighted **red**). *Id.*

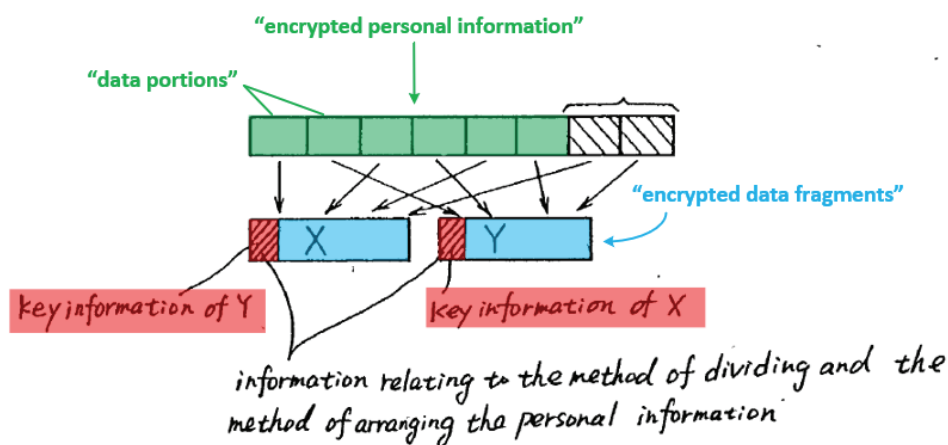
71. Finally, encrypted data fragments X and Y (each with key information for the other data fragment) are separately “registered” with database 41a and database 41b “under separate control” of the different certificate authorities 40a and 40b, respectively (highlighted in **purple** in Fig. 5 and 14). Tajima, [0048], [0111]. A POSA understood that when a data fragment is “registered” with a data base the data fragment is stored on the corresponding storage media. Registering may also include additional processes that databases conventionally use to record information about the stored data, record information about the location of the data

in storage, assign access privileges, etc., but the principal act of registering data with a database is storing the data so that it can be accessed as authorized.

72. Thus, the above-colored steps in Tajima's Fig. 12 embodiment correspond to the same-colored steps of the '116 patent's Fig. 21 embodiment that I discussed above in §V.C ("Specification") and map to the same limitations in claim 1 that I color-coded in the same way in the table in §VII.B below.

73. Tajima's Fig. 13 embodiment (copied below with the same color-coded annotations) is identical to the Fig. 12 embodiment but adds an additional step that mirrors the first "Encrypt" step in FIGS. 21-24 of the '116 patent where the data set can be optionally encrypted before being divided into data portions or shares. EX1001, 57:45-46, 58:58-59:10, 59:34-57.

Fig. 13



74. Specifically, in Tajima's Fig. 13 embodiment, the personal information is "first encrypted, and the encrypted personal information and

information on the key that was used in encrypting the personal information are then divided into a plurality of data portions” (Tajima, [0113]). The same sequence of steps performed on the “data portions” in the Fig. 12 embodiment are then performed on the “data portions” (which are encrypted) in the Fig. 13 embodiment. Tajima, [0111], [0113].

75. Tajima’s Fig. 12 and 13 embodiments mirror the ’116 patent’s disclosure that a data set can be divided first (as in the Fig. 12 embodiment) or an initial encryption step (as in the Fig. 13 embodiment) can optionally be performed prior to dividing the data set into data portions. EX1001, 57:45-46, 58:58-59:10, 59:34-57.

B. Claim 1:

76. In my opinion, Tajima’s Fig. 12 embodiment and Tajima’s Fig. 13 embodiment (which performs the same steps securing steps as the Fig. 12 embodiment but additionally encrypts the personal information before dividing it) each anticipate claim 1. Tajima, [0111], [0113]. For example, the table below illustrates with color coding the correspondence between Tajima’s Fig. 12 embodiment described in [0111] and claim 1. Each of claim 1’s limitations is disclosed and Tajima anticipates claim 1 for at least this reason.

Claim 1	Tajima
A method for securing a data set, the method comprising:	“[0111] As yet another method of dividing personal information, as shown in FIG. 12, personal information that has been

distributing the data set into a plurality of data chunks, wherein none of the data chunks are, by themselves, sufficient to reconstruct the data set; encrypting each of the data chunks with a respective one of a plurality of encryption keys; obfuscating each of the plurality of different encryption keys; and separately storing each data chunk of the plurality of data chunks together with one of the plurality of obfuscated different encryption keys on a plurality of different storage devices.	submitted to certificate authority 40a by the user of user terminal 10 is first divided into a plurality of data portions...and this plurality of data portions is then arranged into two data fragments, data fragment X and data fragment Y, according to a set method of arrangement. Data fragments X and Y are then each encrypted, and encrypted data fragment X is then registered with database 41a of certificate authority 40a and encrypted data fragment Y is registered with database 41b of certificate authority 40b. Information regarding the key that was used for the encryption of data fragment Y is attached to encrypted data fragment X, and information regarding the key that was used in the encryption of data fragment X is attached to encrypted data fragment Y.”
---	---

Claim 1 and Fig. 12 Embodiment

77. The Fig. 13 embodiment described in [0113] first encrypts the personal information and then performs all the same steps on the encrypted personal information that the Fig. 12 embodiment performs on the unencrypted personal information and therefore meets claim 1 in the same way as the Fig. 12 embodiment, as shown in the table below showing the correspondence between Tajima’s Fig. 13 embodiment described in [0113] and claim 1. Tajima anticipates claim 1 for this additional reason.

Claim 1	Tajima
A method for securing a data set, the method comprising:	“[0113] As yet another method of dividing personal information, as shown in FIG. 13,

distributing the data set into a plurality of data chunks, wherein none of the data chunks are, by themselves, sufficient to reconstruct the data set;

encrypting each of the data chunks with a respective one of a plurality of encryption keys;

obfuscating each of the plurality of different encryption keys; and

separately storing each data chunk of the plurality of data chunks together with one of the plurality of obfuscated different encryption keys on a plurality of different storage devices.

personal information that the user of user terminal 10 has submitted to certificate authority 40a is first encrypted, and the encrypted personal information and information on the key that was used in encrypting the personal information are then divided into a plurality of data portions... This plurality of data portions is then arranged into two data fragments, data fragment X and data fragment Y, according to a set method of arrangement, and further, data fragments X and Y are each encrypted. The encrypted data fragment X is then registered with database 41a of certificate authority 40a and the encrypted data fragment Y is registered with database 41b of certificate authority 40b. Information regarding the key that was used to encrypt data fragment Y is attached to encrypted data fragment X, and information regarding the key that was used to encrypt data fragment X is attached to encrypted data fragment Y.”

78. Thus, in my opinion, Tajima anticipates claim 1 for multiple reasons (including the above correspondence between claim 1 and the Fig. 12 and 13 embodiments and/or additional disclosure in Tajima), as demonstrated in the limitation-by-limitation analysis below. While it is my opinion that a POSA would have understood each of claim 1’s limitations to be expressly disclosed in Tajima, claim 1 is alternatively obvious over Tajima in view of the knowledge of a POSA.

1. 1[PRE] A method for securing a data set, the method comprising:

79. To the extent the preamble of claim 1 is limiting, Tajima discloses a “*a method for securing a data set,*” as claimed.

80. For example, Tajima discloses that “an object of the present invention” is to provide a “personal information management *method*” that can “*improve the security of personal information* that is registered with areas that are connectible to the Internet.” Tajima, [0044]. Such methods for securing personal information that meet 1[PRE] are described in Tajima’s Fig. 12 and 13 embodiments, which are two of the different methods Tajima discloses for dividing, encrypting and securely storing personal information as distributed encrypted data segments on different service site. Tajima, [0111], [0113].

81. In Tajima’s Fig. 12 embodiment, the personal information whose “security” is improved meets the claimed “*data set*” secured by the method because the personal information is a collection of separate elements of information that are related by their connection to the user (e.g., name, address, telephone number, e-mail address, etc.) that are treated as a unit in handling the data in preparing it for storage at the service sites (see discussion of Tajima in §VII.A above) entered into a “personal computer” as electronic data transmitted over the Internet (Tajima, [0006], [0067]), and divided into parts referred to as “*data portions*” (Tajima, [0111]-[0114]). Tajima’s personal information meets the

claimed “*data set*” under both parties’ proposed constructions. EX1042, 9 (Petitioner construing “data set” as “collection of related information made up of separate elements that can be treated as a unit in data handling”); EX1043, 6 (Patent Owner construing “data set” as a “collection of information for storage”).

82. In Tajima’s Fig. 13 embodiment, the personal information that meets the claimed “*data set*” is encrypted before dividing it. This encrypted personal information meets the claimed “*data set*” because it’s still a data set after encryption. The ’116 patent’s dependent claim 8 recites that claim 1’s “*data set*” is generated by “*encrypting primary data*” confirming that the claimed data set can be either encrypted or unencrypted data—i.e., “*encrypt[ed] primary data*” is still a “*data set.*” The encrypted personal information in the Fig. 13 embodiment is secured by the same steps that are applied directly to the personal information in Tajima’s Fig. 12 embodiment. Thus, Tajima’s FIG. 12 and FIG. 13 techniques for securing personal information each meet 1[PRE].

2. [1A] distributing the data set into a plurality of data chunks, wherein none of the data chunks are, by themselves, sufficient to reconstruct the data set

83. In my opinion, Tajima’s method of dividing the user’s personal information into multiple “data fragments” where “each data fragment is completely unintelligible when taken independently” (Tajima, [0072]) meets [1A].

84. For example, in the Fig. 12 and Fig. 13 embodiments, Tajima’s data fragments X and Y ([1A]’s “*plurality of data chunks*”) are formed by dividing personal information (the claimed “*data set*” in the Fig. 12 embodiment) or by dividing the encrypted personal information (the claimed “*data set*” in the Fig. 13 embodiment) into “a plurality of data portions” of “fixed” or “arbitrary length” that are “then arranged into two data fragments, data fragment X and data fragment Y, according to a set method of arrangement.” Tajima, [0111], [0113].

85. This meets “*distributing the data set into a plurality of data chunks,*” as recited in [1A] under both Petitioner’s construction (EX1042, 2: “splitting the data set into a plurality of data chunks”) and Patent Owner’s construction (EX1043, 2: “dividing and placing the data set into a plurality of data chunks”. As shown in annotated Figs. 12 and 13 below, Tajima’s dividing method splits the data set into data portions and places the resulting data portions into a plurality of data chunks.

Fig. 12

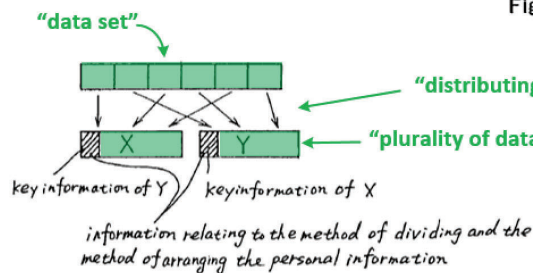
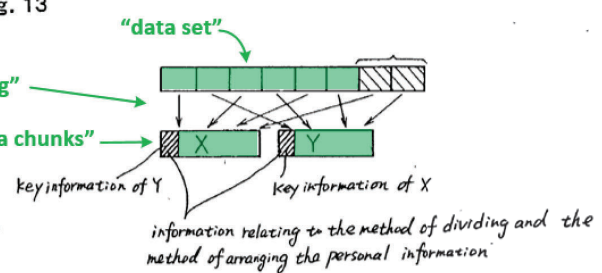


Fig. 13



86. Tajima’s data fragments X and Y each include a non-contiguous subset of the divided data portions of the data set (the personal information in Fig.

12 and the encrypted personal information in Fig. 13) “such that each data fragment is completely unintelligible when taken independently.” Tajima, [0072]); *see also* Tajima, [0111], [0113], [0122]. Thus, in my opinion none of the data fragments X and Y (again, the claimed “*data chunks*”) in Tajima “*are, by themselves, sufficient to reconstruct the data set,*” as recited in [1A] for at least two independent reasons. First, each data fragment is only a subset of the data set and not its entirety. Thus, possession of any one data fragment cannot be used to reconstruct the data stored in the other data fragments even if the content of that fragment could be ascertained. Second, each subset is also unintelligible by itself due to Tajima’s technique of dividing the data set in data portions and distributing them into data fragments. Tajima, [0072], [0111], [0113], [0122], [0156].

87. In my opinion, nothing in the ’116 patent or the prosecution history supports an interpretation of a “data chunk” as requiring anything more than what limitation [1A] recites—i.e., that “*none of the data chunks are, by themselves, sufficient to reconstruct the data set.*” The ’116 patent primarily uses the terms “portions” or “shares” when describing the data subsets into which a data set being secured is split, which “portions” or “shares” may include a subset of the bits of the data set that are contiguous or non-contiguous, encrypted or plaintext. EX1001, 21:55-22:4, 58:58-62; 77:51-82:2.

88. The '116 patent's specification uses the term "chunk" in a single paragraph (81:61-82:2) to refer to an intermediary collection of bits (one bit from each of a plurality of "segments" into which data set was split) that are then distributed into a plurality of shares, demonstrating that a "chunk" can be a subset of non-contiguous bits of the data being secured. EX1001, 81:38-82:2. I found no evidence that the claimed "*data chunk*" cannot encompass a portion including some but not all of the data set. Thus, in my opinion, Tajima's method of dividing personal information (e.g., Fig. 12 embodiment) or encrypted personal information (e.g., Fig. 13 embodiment) into data fragments meets [1A] for the same reasons the '116 patent's embodiments do.

3. [1B] encrypting each of the data chunks with a respective one of a plurality of encryption keys

89. In my opinion, Tajima meets [1B] because in both the Fig. 12 and Fig. 13 embodiments "data fragment X and data fragment Y are each encrypted" using different keys, each of which is then stored with the data fragment other than the fragment that key encrypted, as illustrated in Figs. 12 and 13 (shown below with encrypted data fragment X and the key information used to encrypt it highlighted **red** and encrypted data fragment Y and the key information used to encrypt it highlighted in **blue**). Tajima, [0111], [0013].

Fig. 12

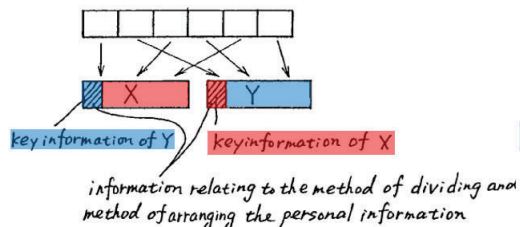
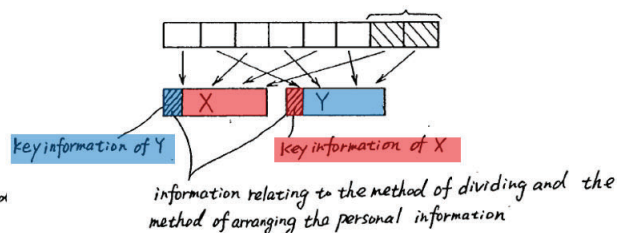


Fig. 13



90. POSAs understood from Tajima’s discussion of “the key that was used in the encryption of data fragment X” and “the key that was used for the encryption of data fragment Y” (Tajima, [0111], [0113]) that the keys are different. If this were not the case, there would be no reason for Tajima to store each fragment’s key with the other fragment or for the key information to be “used to decrypt” a different data fragment from the fragment with which the key information was stored (Tajima, [0112], [0114]). Figs 12 and 13 specifically differentiate “key information of X” and “key information of Y” that a POSA understood reinforces that the keys are different. Thus, in my opinion, Tajima encrypts data fragment X and data fragment Y “*with a respective one of a plurality of encryption keys*” even if [1B] requires distinct keys.

91. Even if Tajima had not disclosed different keys for data fragments X and Y, it is my opinion that that would have been an obvious and conventional way to implement what Tajima describes. POSAs would have been motivated to use different keys to achieve Tajima’s goal of increasing security (see e.g., Tajima, [0044]) and would have reasonably expected success because it was well-known to

use different keys for different fragments of data to increase security, as corroborated by the following prior art references:

- EX1006 discloses that it was known that the “use of multiple keys provides enhanced security by preventing an entire content stream from being accessed with a single key” ([0004]) and teaches encrypting each content packet of digital content with a different title key ([0064]-[0065]) so that “a party who illicitly obtains a title key cannot access the entire stream” and “the pirated key is only good for decrypting the content packet to which is was attached” ([0071]).
- EX1016 discloses fixed encryption schemes were “weak” in protecting against piracy (1:24-29) and thus teaches an encryption scheme in which the “scrambling algorithm and the key word for the source data are dynamically changed to effectively guard against piracy” (1:35-42). *See also* 3:56-60 (“Therefore, the *source data segments* are scrambled *using different scrambling algorithms and different key words* that are recorded in the respective segment header to effectively prevent piracy of the original source data.”).
- *See also* EX1036, [0040]-[0048]; EX1020, [0039]-[0040]; EX1025, 7:2-10:15 also disclosing the routine and conventional technique of using different encryption keys to increase security.

92. Thus, in my opinion, this obvious implementation of using different keys to encrypt Tajima’s data segments (which in my opinion is plainly disclosed in Tajima for the reasons discussed above) meets [1B].

4. [1C] obfuscating each of the plurality of different encryption keys

93. Tajima meets [1C] because in the Fig. 12 and Fig. 13 embodiments, “key information of Y” is attached to encrypted data fragment X and “key information of X” is attached to encrypted data fragment Y, as illustrated in Figs. 12 and 13 below. Tajima, [0111], [0113].

Fig. 12

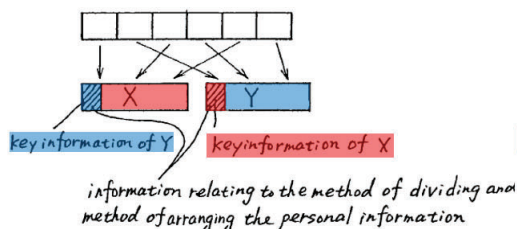
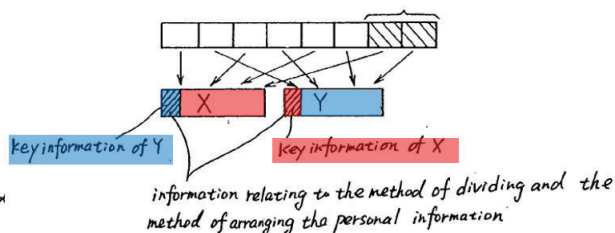


Fig. 13



94. As discussed above in the discussion of the ’116 patent (see §V.C above), the ’116 patent discloses obfuscating in this same way: describing the “**Obfuscate**” step in Figs. 21-24 (boxed in **red** in the portion of Fig. 21 below) as “Share 1 gets Key 4, Share 2 gets Key 1, Share 3 gets Key 2, Share 4 gets Key 3”— i.e., each encryption key is stored with encrypted data that the key did not encrypt. EX1001, 57:59-62, 58:32-36; 59:7-11, 59:54-57.

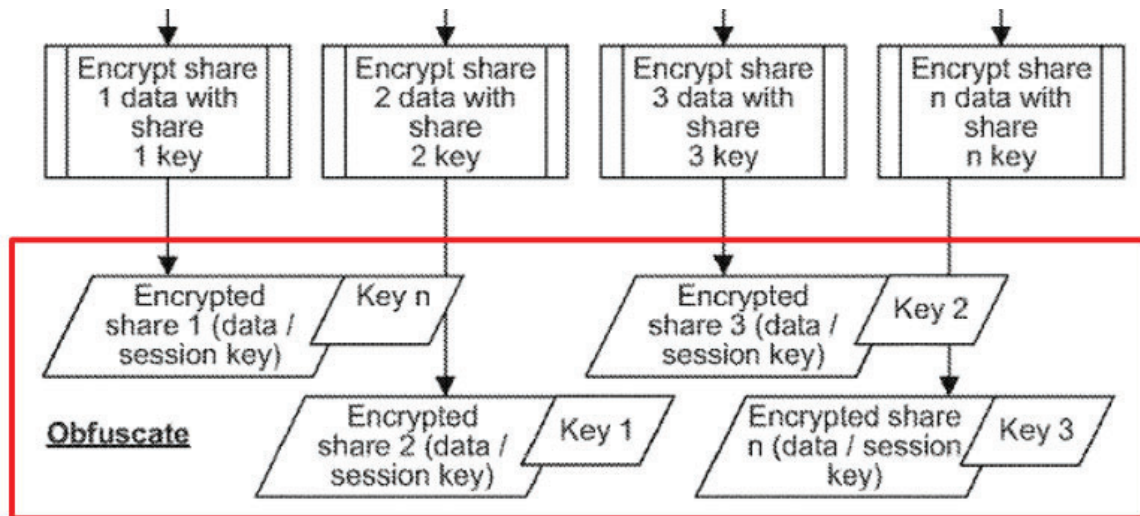


FIG. 21

95. By attaching key information to a data fragment that it did not encrypt, Tajima obfuscates the same way the '116 patent obfuscates, and in my opinion therefore meets [1C]. While the parties did not propose constructions for [1C], both parties proposed a construction for the “*obfuscated different encryption keys*” recited in limitation [1D] discussed below. Petitioner construes storing each data chunk with one of the plurality of “obfuscated different encryption keys” as storing each data chunk “with a key other than the key that encrypts that chunk” and cites as evidence the '116 patent's Figs. 21-24 and the accompanying description of the illustrated “Obfuscate” step discussed above and in §V.C's discussion of the '116 patent. EX1042, 7. Patent Owner construes an “obfuscated different encryption keys” as a key that has been “rendered unclear or obscure” and also points to the '116 patent's Fig. 21-24 embodiment's “Obfuscate” step as evidence. EX1043, 5. In my opinion, Tajima's technique of attaching key

information to a data fragment that it did not encrypt meets [1C] under both instructions because pairing key information with a non-corresponding data fragment obfuscates the key by making it unclear or obscure to an unauthorized user which data fragment the key can decrypt.

96. Tajima’s “key information” is “information regarding the key that was used for the encryption of [the respective] data fragment” and that is “used to decrypt” retrieved data fragment when recovering a user’s personal information. Tajima, [0111]-[0114]. A POSA understood that Tajima’s key information includes at least the key for several reasons based on Tajima’s disclosure.

97. First, Tajima discloses that to restore secured personal information, “the key information that was attached to the encrypted data fragment X is used to decrypt data fragment Y, and the key information that was attached to encrypted data fragment Y is used to decrypt data fragment X” (Tajima, [0112], [0114]). Tajima does not disclose that the key information is insufficient to decrypt the corresponding data fragment, does not disclose that any additional steps are needed to obtain the key, and does not describe a key management technique or infrastructure (e.g., a separate keystore) to implement a system where the key information does not include the encryption key. Thus, a POSA would have understood this to mean the key information itself is all that is needed to decrypt the corresponding data fragment—i.e., the key information includes the key.

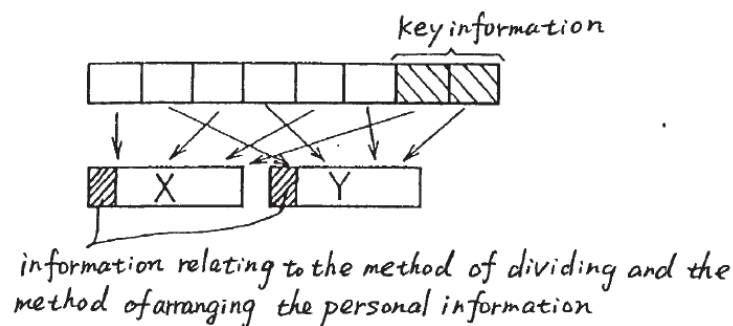
98. Tajima’s disclosure that the method of dividing the personal information, including the different arbitrary lengths of the data portions, and the method of arranging the arbitrary-size data portions in the data fragments must be attached to each of the data fragment reinforces a POSA’s understanding that Tajima’s data fragments include sufficient information for the authorized service site to recover the personal information by the authorized service site—i.e., Tajima’s key information includes at least the encryption key so that the authorized service site can decrypt the data fragments by retrieving all of the data fragments. Tajima, [0104]-[0105], [0107]-[0108], [0112], [0114].

99. Second, Tajima’s claims reinforce that “key information” includes the corresponding encryption key. For example, Tajma’s claim 8 recites a system that “encrypts said personal information and divides the encrypted personal information and the *encryption key* used in the encryption into a plurality of data portions” that are arranged into “at least two data fragments” (see claim 4 from which claim 8 depends and see also claims 19, 29, 38 and 48), which is disclosed in the Fig. 11 and 13 embodiments where “encrypted personal information and *information regarding the key* that [was] used in the encryption are then divided into a plurality of data portions...[and] arranged in two data fragments” (Tajima, [0109], [0114]).

100. Tajima uses the term “information regarding the key” and “key information” interchangeably regarding both the key used to encrypt the personal

information and keys used to encrypt individual data fragments. Tajima, [0109]-[0114], Figs. 11-13. For example, in the FIG. 11 embodiment, the “information regarding the key that [was] used in encrypting the personal information” (Tajima, [0109]) is referred to as “key information” in Fig. 11 (below) and as “the key information that was attached to the encrypted personal information” (Tajima, [0110]) when describing the recovery of the personal information.

Fig. 11



101. Likewise for the FIG. 13 embodiment, the “information on the key that was used in encrypting the personal information” (Tajima, [0113]) is referred to as “the key information that is attached to the combined personal information” (Tajima, [0114]) when describing the recovery process (Fig. 13 shows this key information as in Fig. 11 but the label is missing). Similarly, for the Fig. 12 and 13 embodiments, Tajima uses the terms “information regarding the key” and “key information” interchangeably regarding the keys used to encrypt the individual

data fragments. See Tajima, [0111]-[0112], Fig. 12 for the Fig. 12 embodiment and Tajima, [0113]-[0114], Fig. 13 for the Fig. 13 embodiment.

102. Thus, the subject matter Tajima describes interchangeably as “information regarding the key” and “key information” in the Fig. 11 and 13 embodiments is claimed by Tajima as including “the encryption key” itself, reinforcing a POSA’s understanding that Tajima is likewise disclosing that the “information regarding the key” and “key information” regarding the keys used to encrypt data fragments also include the corresponding encryption key. Tajima, [0111]-[0114], Figs. 12 and 13. That is, because what Tajima calls “key information” is claimed to include “the encryption key” itself, a POSA understood that the same term “key information” when applied to encrypt individual data fragments (Tajima, [0111]-[0114]) would likewise include the corresponding encryption key.

103. Finally, as discussed above and in §VII.D below, a POSA would have understood that Tajima’s key information may (and likely would) include *additional* metadata about the key to facilitate recovering the user’s personal information by the authorized service site, a POSA would have understood from Tajima’s disclosure that the key information includes the corresponding key for the reasons discussed above and from the absence in Tajima of disclosure of information that could be used to replace the key (e.g., a pointer to the key that

would require storage of the key somewhere else), or any technique of recovering the personal information with such information, that Tajima's key information includes the corresponding key.

104. Even if Tajima had not disclosed that its key information in the Fig. 12 and 13 embodiments (Tajima, [0111]-[0114]) includes the corresponding encryption key, that would have been an obvious and conventional way to implement what Tajima discloses so that the service site authorized to recover personal information need only retrieve the stored encrypted data fragments to acquire the keys needed to decrypt them (which in my opinion is what Tajima discloses), rather than implementing a separate keystore or key management system that would be needed if the key information did not include the key itself.

105. Additionally, even if the key information did not include the key, Tajima would still meet [1C] by attaching the key's information to a data fragment it did not encrypt so the key's information obtained by acquiring a data fragment cannot be used to decrypt that data fragment. That is, the key itself would still be obfuscated because pairing a key's information with non-corresponding data fragments obfuscates the key by making it unclear or obscure to an unauthorized user as to which data fragment the key can decrypt. EX1043, 5 (Patent Owner interpreting an "obfuscated" key as a key that has been "rendered unclear or obscure").

106. Thus, Tajima, or alternatively an obvious implementation thereof, meets [1C] at least because Tajima obfuscates encryption keys in the same way the '116 patent does.

5. [1D] separately storing each data chunk of the plurality of data chunks together with one of the plurality of obfuscated different encryption keys on a plurality of different storage devices

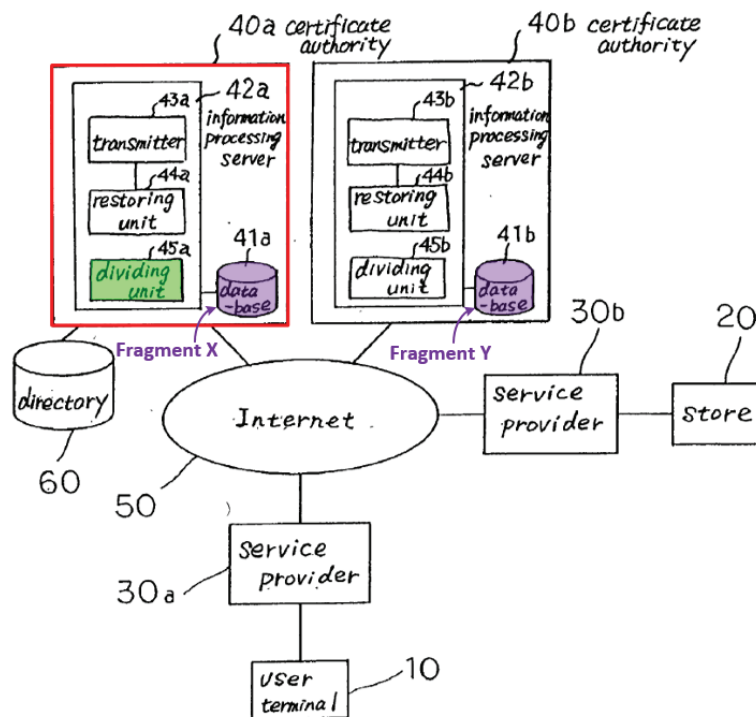
a. Tajima separately stores data chunk/key pairs in the same way as the '116 patent's embodiments

107. In my opinion, Tajima's attaching key information to encrypted data that it did not encrypt—e.g., “[i]nformation regarding the key that was used for the encryption of data fragment Y is attached to encrypted data fragment X, and information regarding the key that was used in the encryption of data fragment X is attached to encrypted data fragment Y” (Tajima, [0111], [0113], Figs. 12 and 13)—and registering the resulting encrypted data fragments with databases belonging to and under the control of respective service sites (e.g., on certificate authority 40a's database 41a and certificate authority 40b's database 41b in Fig. 5) meets [1D]. Tajima, [0111], [0113], [0122], [0156].

108. As I discussed above in §VII.A, a POSA understood that registering a data fragment with a data base includes storing the data fragment on corresponding storage media. Additionally, POSAs understood that Tajima's separate databases that each “belong to” the corresponding service site (Tajima, [0129], [0152]) to

which respective data fragments are registered (e.g., certificate authority 40a's database 41a and certificate authority 40b's database 41b in Fig. 5 below) are part of “different storage devices” as recited in [1D].

Fig. 5



109. Indeed, a security improvement Tajima emphasizes is storing different data fragments in “areas that are connectible to the Internet” and “each under separate control” (Tajima, [0048]) and that are “closed to the outside by means of the access control function of the server or firewall” (Tajima, [0079]) so that the entirety of the user’s personal information cannot be recovered by a security breach of a single service site. Tajima, [0042], [0048], [0073], [0079], [0111], [0113]. POSAs would have understood this to disclose that each service site’s database is implemented on different storage devices from other service sites’

databases, at least because providing databases on the same storage device to store different data fragments of a user's personal information would defeat a primary purpose of Tajima's improved security system (i.e., distributively storing data fragments so that a breach of any given service site does not comprise the data fragment stored at other service sites, thus preserving the security of the user's personal information).

110. Thus, as explained above, Tajima stores each encrypted data fragment together with an "*obfuscated different encryption key*" (i.e., the key information that encrypted the other data fragment) on "*a plurality of different storage devices*" (i.e., a plurality of different databases, each under separate control of a respective different service site) in the same way the '116 patent discloses storing encrypted data portions, each with the encryption key of a different data portion, on different depositories to avoid having "any one depository contain enough information to decrypt the information held there, the key required to decrypt the share is stored in a different depository" (EX1001, 60:49-52), as shown in the excerpt from 60:53-56 of the '116 patent copied below. *See also* EX1001, 60:57-61:13.

Depository 1: SD1, Kn
Depository 2: SD2, K1
Depository 3: SD3, K2
Depository n: SDn, K3.

111. That is, the '116 patent discloses that each encrypted share (SD1-SDn) is paired with one of the encryption keys (K1-Kn) used to encrypt a different share and each SD/K pair is stored on a different depository (EX1001, 60:30-56), which is the same technique Tajima discloses in the Fig. 12 and 13 embodiments—i.e., pairing key information with a non-corresponding data fragment and storing each pair on a different database. Tajima, [0111], [0113], [0116], [0153], [0156]. For example, applying the '116 patent's notation to Tajima's Fig. 5 system, Tajima discloses:

Database 41a: encrypted data fragment X, key information of Y

Database 41b: encrypted data fragment Y, key information of X

112. Thus, in my opinion, Tajima meets [1D] at least because Tajima separately stores data fragment/key information pairs on different storage devices in the same way the '116 patent describes, which also meets Petitioner's proposed construction (EX1042, 7). Patent proposes that [1D] requires "separately storing each data chunk...together with one of the...different encryption keys that has been rendered unclear or obscure." EX1043, 5. Tajima meets [1D] under that construction because separately storing each data fragment with a key that did not encrypt that fragment renders the key unclear or obscure to an attacker as to which fragment the key can decrypt.

b. A POSA would have understood the separately storing step of [1D] to only require storing each data chunk/key pair once

113. In my opinion, a POSA would not have interpreted limitation [1D] as requiring that each data chunk/encryption key pair be stored on multiple different storage devices. First, if limitation [1D] were interpreted as requiring redundant storage of each data chunk/encryption key pair on multiple different storage devices, claim 1 would fail to cover the very above-described embodiments the '116 patent describes for storing data portion/encryption key pairs on different storage devices.

114. Second, in my opinion, the prosecution history also does not support interpreting [1D] to require storing each chunk/key pair on multiple different storage devices and instead reinforces that [1D] should be interpreted to cover storing each chunk/key pair on a respective storage device that the '116 patent describes. As originally presented, limitation [1D] recited “*storing the plurality of data chunks with the plurality of obfuscated different encryption keys on a plurality of different storage devices*” (EX1002, 247), which broadly covers at least the '116 patent's above-discussed embodiments in which each share/key pair is stored on a respective storage device (depository).

115. As I discussed in §V.E above summarizing the prosecution history, Patent Owner amended claim 1 (then claim 2) to read: “separately storing each

data chunk of the plurality of data chunks together with one of the plurality of obfuscated different encryption keys on a plurality of different storage devices” (added language underlined) and cited “paragraphs [0431] and [0431]-[0438]” of the published application (EX1039, which corresponds to EX1001, 63:11-67) as providing support for the amended subject matter. EX1002, 118, 123. None of the cited paragraphs [0431]-[0438] disclose redundantly storing each share/encryption key pair on multiple different depositories:

- EX1039, [0431] (corresponding to EX1001, 63:11-16) reiterates the ’116 patent’s above-described storing technique (EX1001, 60:49-61:13), which reinforces that this amendment did not exclude those embodiments (i.e., embodiments in which each chunk/key pair is stored on respective storage devices, but each chunk/key pair is not redundantly stored on multiple storage devices).
- EX1039, [0432] (corresponding to EX1001, 63:17-40) says that it “is readily apparent to those of ordinary skill in the art that the data securing methods and computer system of the present invention are widely applicable to any type of data in any setting or environment,” and provides exemplary environments such as “any non-commercial or commercial environment or setting” including “any organization, government agency or corporation.”

- EX1039, [0433]-[0438] (corresponding to EX1001, 63:41-67) disclose examples of different keys that can be employed in a data securing method ("Parser Master Key," "Session Master Key," "Intermediary Key").

116. Thus, none of the cited paragraphs disclose or support redundantly storing each share/encryption key pair on multiple different depositories, demonstrating to a POSA that this is not what the amendment requires. Indeed, Patent Owner asserted the amendment was made "to *clarify* the claimed subject matter," not to change the claim scope to newly require redundantly storing each chunk/key pair on multiple different storage devices. EX1002, 123. Patent Owner's prosecution arguments likewise reinforce the aspect of the storing technique being clarified was that *each* data chunk is paired with *one* of the obfuscated encryption keys and stored separately from the other data chunk/obfuscated encryption key pairs rather than all the data chunk/key pairs being stored together. EX1002, 123-125. Specifically, in distinguishing Bolosky, the Patent Owner alleged:

"there is no disclosure in Bolosky that recites separately storing each data chunk with one of the plurality of obfuscated different encryption keys...Consequently, Bolosky merely describes a single data stream that stores all of the data and keys for decrypting all of the data, which is not the same as separate storing as require [sic] by amended claim 2." EX1002, 124 (emphasis original).

117. Patent Owner’s emphasis reinforces that the amended claim requires that *each* data chunk is stored with *one* of the obfuscated encryption keys and is stored separately from other data chunk/encryption key pairs—i.e., chunks and keys are paired and stored separately like the ’116 patent’s above-described embodiments). Patent Owner never argued that the amended claim requires that *each* data chunk/encryption key pair be stored on *multiple* different storage devices—the Patent Owner did not mention the storage devices at all. EX1002, 124.

118. Moreover, construing [1D] to require redundant storage of data chunk/encryption key pairs on multiple devices would require determining that the same amendment made to each of the independent claims resulted in claims 1 and 9 newly requiring redundant storage, but not the parallel system claims 15 and 21. Specifically, as discussed in the summary of the prosecution history in §V.E above, parallel system claims 16 and 22 (which issued as claims 15 and 21) were similarly amended (EX1002, 120-122).

“separately store each data chunk of the plurality of data chunks together with one of the plurality of obfuscated [claim 16]/transformed [claim 21] different encryption keys, wherein the plurality of data chunks is stored on a plurality of different storage devices.” (EX1002, 120-122).

119. Because the system claims recite storing the plurality of data chunks on a plurality of different storage device in a wherein clause that is not affected by

the amendment, there is no reading of the system claims as requiring redundantly storing *each* data chunk/key pair on *multiple* storage devices, only that “*the plurality* of data chunks is stored on a plurality of different storage device,” just as each of the independent claims recited as first presented before amendment.

During prosecution, Patent Owner argued that the system claims distinguished over the prior art “for the similar reasons as recited...with respect to claim 2” (issued claim 1) and did not separately or provide any other reasons (EX1002, 124), reinforcing that the “*each* data chunk” in the amended claims modifies separately storing with “*one* of the plurality of obfuscated different encryption keys” and does not require storing *each* data chunk (paired with one of the keys) on *multiple* storage devices.

c. It would have been obvious to implement redundant storage in Tajima

120. A POSA would have been motivated to redundantly store Tajima’s fragments (which each included a data fragment/key pair) to provide backup copies of information stored at service sites implementing Tajima’s data security scheme (e.g., certificate authorities, service providers, etc.) for fault tolerance reasons as was routinely and conventionally done to securely store data, for example, by configuring Tajima service sites to: (1) replicate its own stored fragments on multiple storage devices, e.g., via a redundant array of inexpensive disks (RAID); and/or (2) make copies of each fragment resulting from a service site dividing a

user's personal information (e.g., Tajima's Fig. 12 or Fig. 13 dividing techniques) that are redundantly stored at different service sites. Both techniques were routine and conventional ways to implement redundancy to ensure the availability of protected data as discussed below in connection with exemplary prior art systems implementing these technique that corroborate my opinion that implementing redundancy in Tajima would have been obvious, routine and conventional. Providing data redundancy was (and still is) standard practice in data storage and it would be surprising to implement a secure data storage system *without* data redundancy. Indeed, a POSA implementing Tajima's system would have known to provide redundancy and it would have been at the very least an obvious way to implement Tajima.

121. Regarding redundancy technique (1) above, RAID was one known and wide-spread technology used to redundantly store data (see e.g., EX1035, 8:13-23, 10:36-43; EX1032, 1:5-13; EX1033, 1:9-12; EX1013, [00338]), including to provide redundant storage of a certificate authority database in an online merchant context like Tajima's (EX1035, 8:13-23, 10:36-43), providing both motivation and reasonable expectation of success in implementing Tajima's system to use RAID technology. Thus, it would have been routine and conventional to configure Tajima's service sites to replicate their respective stored fragments (i.e., any fragment received from the service site that divided the user's personal

information that each include a data fragment/key pair) on multiple RAID disks so that failure of any one disk would not prevent that service site from providing a requested fragment provided at least one disk on which a requested fragment has been redundantly stored remains available. Such a RAID implementation would have been nothing more than the use of a known technique (RAID for redundant storage) for its intended purpose to yield the predictable and intended result of providing fault tolerance in Tajima's secure storage system.

122. Regarding (2), EX1007, EX1017 and EX1030 (discussed in further detail below) all describe systems that, like Tajima, segment a data set and store different segments on different storage devices to protect the data set, and each discloses redundantly storing the segmented data on multiple different storage devices for fault tolerance in recovering the data set, providing both motivation and reasonable expectation of success in implementing Tajima's system by configuring the service site that divides a user's personal information (e.g., Tajima, [0111], [0113]) to make copies of the resulting fragments (e.g., a copy of each fragment including an encrypted data fragment/key pair) to redundantly store the fragments at different service sites (i.e., a service site will store only one fragment of a given user's personal information but each fragment will be stored at multiple service sites) so that the personal information can be recovered even if a service site becomes unavailable or fails. Tajima, [0115], [156] (disclosing that any number of

fragment and service cites may be utilized and on different networks). Redundantly storing Tajima's encrypted data fragment/key pairs on multiple service sites would have been nothing more than the use of a known technique (redundant storage) for its intended purpose to yield the predictable and intended result of providing fault tolerance in Tajima's secure storage system. My opinion that it would have been obvious to do so is corroborated by the following:

- EX1007 discloses at [0007]-[0008] a content delivery system in which a media file is segmented and encrypted and each encrypted segment is stored on a different subscriber to "provide[] a high level of security and a deterrence to piracy" and EX1007 teaches at [0009] to copy each encrypted segment that "are also distributed to the network subscriber in a manner that provides redundancy to the system so that in the event that a given subscriber unit is unable to provide the requesting subscriber with the segment stored on that given unit, there will be available on other units copies of that segment to complete the media file."
- EX1017 discloses at [0025]-[0031] a secure storage technique that segments a file, encrypts each segment, and stores each encrypted segment on different computer devices, and EX1017 teaches at [0058] that the encrypted segments are "redundantly distributed" on different devices in "geographically different locations to redundantly store the

[segments],” thus “enabl[ing] restoration of a file even when [the device] in one location becomes unusable because of, for example, an accident in that location.”

- EX1030 discloses at [0005]-[0010] that by “storing data to be backed-up...on different back-up servers which are spatially remote from one another” “data loss becomes very unlikely” and that additional protection against data loss occurs by distributing encrypted segmented data to multiple back-up servers in such a way that less than all segments are required to recover the original data but more than one segment is required, which gives the benefit that “[t]he breakdown of one back-up server is [] non-problematic.”

C. Claim 5: The method of claim 1, wherein storing the plurality of data chunks together with the plurality of obfuscated different encryption keys comprises storing each of the plurality of data chunks together with a respective one of the plurality of obfuscated different encryption keys

123. In my opinion, Tajima’s disclosure of attaching the key information (i.e., the “*obfuscated different encryption keys*”) used to encrypt one data fragment to a different data fragment meets claim 5.

124. A POSA would have understood that claim 5’s recitation that each data chunk is stored “*together with a respective one of the plurality of obfuscated different encryption keys*” does not require a data chunk to be stored with the

obfuscated encryption key used to encrypt that data chunk, only that each data chunk be stored with an obfuscated encryption key that is different than that stored with other data chunks—i.e., “*a respective one*” of the obfuscated keys. Specifically, claim 5 recites “*a* respective one” rather than “*the* respective one,” and thus the stored encryption key in claim 5 need not be the same encryption key recited in [1B] (“*encrypting each of the data chunks with a respective one of a plurality of encryption keys*”) for that data chunk. Instead, claim 5 introduces a new “*a respective one of the plurality of obfuscated different encryption keys*” that the data chunk is stored together with.

125. Thus, the obfuscated key with which a data chunk is stored in claim 5 may be different from the obfuscated key used in [1B] to encrypt that data chunk, which (1) narrows claim 1’s limitation [1D] that does not require each chunk to be stored with a “respective” encryption key, only that it be stored with “one” of them, and (2) is consistent with and does not exclude the ’116 patent’s embodiments describing storing data chunks together with those keys, e.g., the embodiments described at EX1001, 57:25-61:9, Figs. 21-24 that I discussed above in §§V.C and VII.B.5.

D. Claim 6: The method of claim 1, wherein storing the plurality of data chunks together with the plurality of obfuscated different encryption keys comprises including, with a data chunk of the plurality of data chunks, data indicative of an encryption key that was used to encrypt a different data chunk of the plurality of data chunks

126. To the best of my knowledge after reviewing the '116 patent, the “*data indicative of*” claim language does not appear in the '116 patent’s specification—i.e., the '116 patent does not use the phrase “*data indicative of*” outside of the claims. I note that in the district court litigation, Patent Owner has taken the position that “*data indicative of*” an encryption key is met by the key itself. EX1012 (Patent Owner’s infringement contentions), 8. Specifically, Patent Owner alleged that claim 2’s “performing a transform on *data indicative of at least one of the plurality of different encryption keys*” is met because “Google performs a transform on each of the different encryption keys *when it encrypts the DEKs using the KEKs.*” EX1012 (infringement contentions), 8 (“Within Google’s ‘Encryption at Rest,’ obfuscating each of the plurality of different encryption keys comprises performing a transform on data indicative of at least one of the plurality

of different encryption keys. For example, Google *performs a transform on each of the different encryption keys when it encrypts the DEKs using the KEKs.*)¹

127. Thus, in my opinion, Tajima’s technique of storing key information with an encrypted data fragment that the key information did not encrypt—i.e., storing key information with a non-corresponding data fragment—meets “including, with a data chunk of the plurality of data chunks, data indicative of an encryption key that was used to encrypt a different data chunk of the plurality of data chunks,” as recited in claim 6 at least under Patent Owner’s interpretation of “data indicative of” a key because Tajima’s key information includes the corresponding key. EX1012, 8.

128. Additionally, in its preliminary claim constructions, Patent Owner interprets “data indicative of” a key as “information that points out, points to, makes known, shows, directs to, hints at, or suggests” the key. EX1043, 3-4. Tajima’s storing key information with a data fragment that it did not encrypt meets this construction at least because for the key information to be “used to decrypt”

¹ Patent Owner also alleged that storing “data indicative of” a key is met by storing the key itself for claim 1 of co-asserted US Patent No. 9,338,140 (EX1044, sharing the identical specification) and claim 1 of US Patent No. 10,452,854 (EX1015, sharing the same relevant disclosure). EX1037, 34; EX1011, 9.

(Tajima, [0112], [0114]) the corresponding data fragment (i.e., a data fragment that the key information encrypted but to which it was not attached), the key information would need to show or make the key known. Tajima, [0111]-[0114]. That is, when the authorized service site acquires the encrypted data fragments of a user's personal information to recover it, "the key information that was attached to encrypted data fragment X is used to decrypt data fragment Y, and the key information that was attached to encrypted data fragment Y is used to decrypt data fragment X" (Tajima, [0112], [0114]), demonstrating that the key information at least "makes known" or "shows" the key so that the corresponding data fragment can be decrypted. EX1043, 3-4.

129. Moreover, as I discussed in §VII.B.4 above, in addition to the key used to encrypt a different data fragment, Tajima also attaches to each data fragment: (1) information indicating the different arbitrary lengths of the data portions into which the personal information was divided, (2) the method of arranging the arbitrary-sized data portions in the data fragments, and (3) the key information regarding the key used to encrypt the personal information before dividing it in the Fig. 11 and 13 embodiments (Tajima, [0109]-[0110], [0113]-[0114]).

130. As a result, Tajima's authorized service site must parse acquired encrypted data fragments to separate all of the information attached to the

encrypted data from the encrypted data and to separate this information into its components parts—i.e., (1) key information used on a different data fragment, (2) information indicating the data-portion size, (3) the method of distributing the data portions, and (3) key information that encrypted the personal information (in the Fig. 11 and Fig. 13 embodiments) must all be separated or differentiated before the data fragments can be decrypted and the personal information reassembled.

Tajima, [0104]-[0105], [0107]-[0114].

131. It was conventional and routine to additionally include metadata to facilitate this parsing process, for example, metadata indicating the total length of the dividing/encryption information and the length of its component parts and/or other metadata to facilitate parsing. See e.g., EX1016, 3:12-26 (including metadata in the segment header to indicate the beginning and end of the header to distinguish the key and the scrambling type from the data itself); EX1036, [0053]-[0054], [0076]-[0077], Figs. 3A, 4, 12 (including key length metadata along with the key in attributes attached to data blocks); EX1006, [0064]-[0067] (disclosing storing identifier information and/or verification information along with the key), FIG. 8 (also illustrating that “key info 112” includes the “encrypted key 112” reinforcing that a POSA would understand Tajima’s key information to include at least the key). Given the amount and variety of information Tajima attaches to the encrypted data, a POSA would have understood that Tajima would have included

such metadata with the encrypted data fragment or, at the very least, a POSA would have recognized the benefit of doing so and would have been an obvious way to implement Tajima to facilitate the parsing of retrieved encrypted data fragments.

132. In Tajima, since encryption keys would need to be differentiated from information indicating data-portion size and the method of arranging (and different encryption keys from each other in the case of the Fig. 11 and 13 embodiments), it would have been obvious to include the key length of the encryption keys in the key information of the corresponding key. *See e.g.*, EX1036, [0053]-[0054], [0076]-[0077], Figs. 3A, 4, 12. Including the key length in Tajima's key information would have the further benefit of allowing for variable-length keys both between individual data fragment encryption keys and encryption keys used to encrypt the personal information (Fig. 11 and 13 embodiments) to provide additional protection and/or flexibility by (1) allowing a service site to use different size keys for different instances of securing different user's personal information, (2) allowing a service site to change key size as appropriate or desired, and/or (3) allowing variable-length keys to be used to encrypt the entire personal information and to encrypt the data fragments, without requiring service sites to internally keep track of different key lengths within each instance of

secured personal information and across all instances of secured personal information.

133. Additionally, as discussed in §X.A below (Ground 4), it would have been obvious and beneficial to also include the encryption algorithm used with each key in Tajima’s key information to allow the encryption algorithm to be varied to increase protection by avoiding known risks of using just one type of encryption. *See e.g.*, EX1016, 1:36-42, 3:33-38, 3:56-60; EX1036, [0004]-[0009]). The obvious implementation of Tajima to include metadata of the key such as key length or the encryption type (i.e., an indication of the encryption algorithm used with the corresponding key) in Tajima’s key information meets claim 6 under Petitioner’s proposed construction. EX1042, 5 (construing “data indicative of” a key as “metadata about” the key).

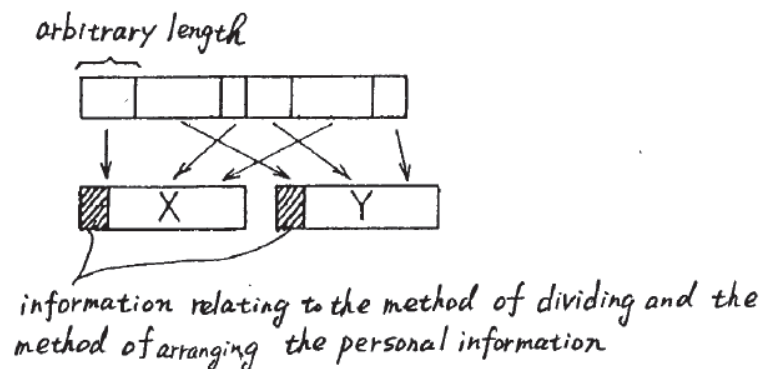
E. Claim 7: The method of claim 1, wherein distributing the data set into a plurality of data chunks comprises using a random technique or pseudorandom technique

134. Tajima discloses that the personal information (e.g., in the Fig. 12 embodiments) or the encrypted personal information (e.g., in the Fig. 13 embodiment) may be divided into data portions of “arbitrary length” (Tajima, [0111], [0113]) determined “according to a function of, for example, *random numbers*, time, or file capacity, [that are] then arranged into two data fragments,

data fragment X and data fragment Y, according to a set method of arrangement” (Tajima, [0103], [0106], [0111], [0113]).

135. For example, in Tajima’s Fig. 10 embodiment, the personal information is divided into “a plurality of data portions each of different *arbitrary length*” (see Fig. 10 copied below) that can be determined based on “random numbers”—i.e., a random number determines the length of each data portion that subsequently gets assigned to one of data fragment X and data fragment Y. Tajima, [0106]; See also [0103] (disclosing that a “random number” may be used to arbitrarily choose a fixed length for the data portions in the Fig. 9 embodiment).

Fig. 10



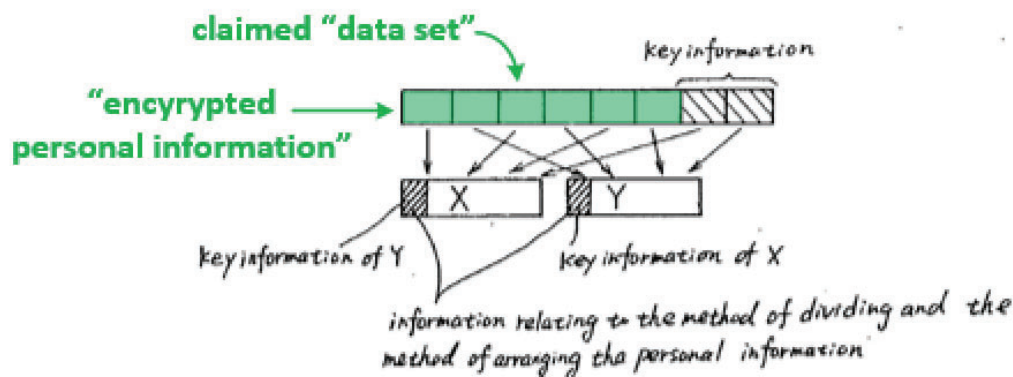
136. Tajima discloses that the Fig. 12 and Fig. 13 embodiments can use any of the methods of dividing personal information into data portions, including dividing data portions each of “arbitrary length as shown in FIG. 9 or FIG. 10.” Tajima, [0111], [0113]. Tajima’s techniques of using “random numbers” to determine the length of each data portion into which the data set is divided and

distributed into the data fragments is a “*random technique or pseudorandom technique*” and therefore meets claim 7.

F. Claim 8: The method of claim 1, further comprising generating the data set by encrypting primary data

137. In Tajima’s Fig. 13 embodiment, the user’s personal information “is first encrypted, and the encrypted personal information and information on the key that was used in encrypting the personal information are then divided into a plurality of data portions.” Tajima, [0113]. As discussed above in §VII.B.1 in connection with claim 1’s preamble 1[PRE], the “encrypted personal information” meets the claimed “*data set*” as illustrated in Fig. 13 that I annotated below.

Fig. 13



138. Thus, in the Fig. 13 embodiment, Tajima’s “personal information” meets the claimed “*primary data*” which is then encrypted to provide “encrypted personal information” that meets the claimed “*data set*” and therefore in my opinion Tajima’s Fig. 13 embodiment meets claim 8. As I demonstrated in

§§VII.B.2-VII.B.5 above, the Fig. 13 embodiment meets each and every limitation of claim 1 from which claim 8 depends at least because it performs all of the same steps on “encrypted personal information” that the Fig. 12 embodiment performs directly on the personal information (i.e., without first encrypting prior to dividing the personal information). As I discussed in §VII.A, Tajima’s Fig. 12 and 13 embodiments mirror the ’116 patent’s disclosure that a data set can be divided first (as in the Fig. 12 embodiment) or an initial encryption step (as in the Fig. 13 embodiment) can optionally be performed prior to dividing the data set into data portions. EX1001, 57:45-46, 58:58-59:10, 59:34-57.

VIII. GROUND 2: TAJIMA-FOSTER (CLAIMS 1-2, 9, 11-14)

139. I have been asked to provide my opinion concerning whether claims 1-2, 9, 11-14 of the ’116 patent would have been obvious to a POSA in light of the disclosures of Tajima, Foster, and the knowledge of a POSA. For the reasons explained below (§§VIII.F-VIII.L), it is my opinion that each of claims 1-2, 9, 11-14 would have been obvious to a POSA over the combination of Tajima and Foster.

A. Storing plaintext keys with encrypted data was a known security risk

140. It was well known to POSAs that storing plaintext keys with the encrypted data provides a known vulnerability, as corroborated by the exemplary references discussed below:

- EX1021 describes that some problems associated with key management of large numbers of keys (e.g., “[m]isplacing or losing encryption keys, or mismatching the data encryption keys and encrypted data,” 1:45-53) could potentially be addressed by storing/transmitting keys and the encrypted data together, but explains that such simple-to-use methods are likewise potentially simple to defeat. 1:52-65 (“However, a method of associating the encryption keys with the encrypted data which is simple and easy to use for the party intended to retrieve the encrypted data will also be potentially simple and easy for unauthorized parties to use, to the detriment of the security of the encrypted data... This potential difficulty is present where the data encryption keys are stored or transmitted together with the encrypted data. Although such a simple system ensures that the encryption keys are available to the recipient or person intended to retrieve the data, others may also find the encryption keys and decrypt the encrypted message.”).
- EX1023 also recognizes the “tradeoff here between the overall security of the storage management system and the user’s convenience” (18:3-8) in

connection with whether to store keys with the data on backup media and generally cautions against it. 17:44-47 (“Generally, encryption keys are not stored [] with the information that they protect since doing so is somewhat akin to locking a house and then putting the keys under the doormat”).

- EX1022 describes techniques for “generation, distribution, installation and management of cryptographic keys” (9:23-26) and notes that “the degree of protection that can be derived from a cryptographic system ultimately depends upon the security of the cryptographic keys” and therefore “cryptographic keys should never occur in clear form outside the cryptographic device” (10:42-48).

141. While Tajima achieves some level of protection by not storing the key used to encrypt a data fragment with that same data fragment, a POSA would have understood that by storing plaintext keys with the encrypted data, even though Tajima obfuscates those keys by storing them with a non-corresponding data fragment, Tajima’s approach subjects the protected data to risk because if an attacker is able to recover all of Tajima’s data fragments (e.g., both data fragments X and Y), the attacker will be in possession of the plaintext keys used to encrypt the entire data set. Decrypting the individual data fragments would become merely a matter of trying different key/data fragment combinations, which was within the capabilities of hackers in the relevant timeframe. Indeed, storing keys with non-

corresponding fragments would provide little deterrence to an attacker that has successfully acquired all fragments.

142. A POSA would have known of multiple techniques to address this vulnerability that a POSA would have recognized as obvious and beneficial to apply to increase the security of Tajima, including by making the keys themselves undecipherable such as by the routine and conventional technique of encrypting encryption keys with a key-encrypting key that is securely maintained separate from the encrypted data (e.g., as disclosed in EX1006 discussed further in §§VIII.B-VIII.C below) or by applying the known “all-or-nothing” transform to the keys (e.g., as disclosed in EX1034 discussed in further detail in §IX below).

143. Further, a POSA would have also been aware of known techniques that made it harder for an attacker having acquired plaintext keys to decrypt the data, including using different types of encryption for different portions of the data being secured, such as techniques disclosed in EX1016 and EX1036 discussed in further detail in §X below.

B. Reasons to encrypt Tajima’s keys

144. It was known that in systems that store encryption keys in association with encrypted data, encrypting the keys increases security. For example, EX1021 that I discussed in §VIII.A above regarding the known risk of keeping encryption keys with the encrypted data proposed that “[s]ecurity for such a system, where the

encryption keys are included with the encrypted message, is not to store or transmit the encryption keys in an easily readable format but rather to encrypt the encryption keys themselves” (1:66-2:4). Similarly, EX1023, that I also discussed in §VIII.A above regarding this known risk (and that noted that generally encryption keys are not kept on the same media as the encrypted data) teaches that “[when] data encryption keys are stored on the media, [] we additionally encrypt them” (18:28-31).

145. EX1006 (Foster) describes a similar problem to EX1021 (see 1:45-53) in connection with possible key mismatch problems that can result when multiple different keys used to encrypted digital content are kept separate from the encrypted data, which Foster addresses by encrypting the data encrypting keys so that the encrypted keys can be securely integrated with the encrypted data ([0004], [0007]-[0008], [0064]-[0071]).

146. There are numerous reasons why a POSA would have encrypted Tajima’s encryption keys, including for the reasons this technique was used generally—i.e., to protect the keys that protect the data—and particularly when keys are kept in association with the data. A POSA would have recognized the benefit of encrypting Tajima’s keys as a further security measure to storing keys with data the keys did not respectively encrypt to guard against the possibility of an attacker acquiring all of the data fragments.

147. More specifically, a POSA would have been motivated to encrypt Tajima's encryption keys so that an attacker that successfully acquired all of the data fragments would not possess all the keys in plaintext form. POSAs had good reason to add this further level of security in Tajima's system. In particular, Tajima's stated security improvement was to divide personal information into data fragments that are "completely unintelligible when taken independently" (Tajima, [0072], [0122]) and store those fragments in systems having separate access control to prevent breach of any one system from allowing access to all of the user's personal information. Tajima, [0042], [0046], [0048], [0079]. Tajima's Fig. 8-10 embodiments rely solely on this technique. Tajima, [0101]-[0108].

148. But Tajima also discloses further levels of security by first encrypting the personal information before dividing it (Figs. 11 and 13) and/or encrypting the data fragments after division (Figs. 12 and 13). Tajima, [0109]-[0113]. A POSA would have understood that this demonstrates that Tajima recognized the possibility of failure of Tajima's security technique of storing data fragments on different databases under separate control, such that additional techniques would be desirable to increase security. That is, if dividing a user's personal information into data portions, distributing those data portions into different data fragments that are distributively stored on different service sites without more (as in the Fig. 8-10 embodiments) provided sufficient security in all circumstances, the further levels

of security provided by the Fig. 12 and 13 embodiments would have been unnecessary.

149. As I discussed above in §VIII.A above, a POSA would have recognized that storing Tajima’s key information with encrypted data (albeit obfuscated) would still have been vulnerable to an attacker acquiring each of the data fragments. Thus, a POSA would have been motivated to apply the known technique of encrypting encryption keys to Tajima so that even if an attacker successfully obtained all data fragments, the information obtained would still be insufficient to decrypt the individual data fragments. Indeed, for the same reasons Tajima uses multiple layers of security in protecting user’s personal information, POSAs would have recognized the benefit of layering multiple techniques to increase security in protecting Tajima’s encryption keys themselves—i.e., to protect the keys that protect the data. Such “belt-and-suspenders” security techniques were well-known and desirable to increase security. For example, EX1016 at 3:12-60 discloses varying the encryption algorithm/keys used to encrypt different respective data segments to which they are attached to avoid single-encryption-algorithm vulnerabilities, and further teaches at 4:21-27 to optionally also encrypt the encryption keys to provide even “stronger protection” against piracy. As another example, EX1036 at [0003]-[0009] also discloses increasing security by using different keys and encryption algorithms for different

data blocks, and teaches at [0054] to also randomly attach encryption keys to different data blocks (much like Tajima attaches keys non-corresponding data fragments). As another example, EX1026 teaches at 10:7-28 to encrypt a data encryption key with *multiple* different key encryption keys to provide a higher level of security. Thus, the desirability of stacking multiple security techniques to improve the overall security was conventional and using Foster's key-encryption technique to encrypt Tajima's key would have been a simple and straightforward way to implement this known and desirable approach to layering security—i.e., by adding to the security provided by Tajima's technique (storing each encryption key with a data fragment it did not encrypt) the further security of encrypting those encryption keys as taught by Tajima.

150. Additionally, encrypting Tajima's keys would have been nothing more than applying a known technique (encrypting encryption keys) for its intended purpose (to increase security) to yield the predictable result of increasing security against unauthorized access to a user's personal information. That is, encrypting Tajima's encryption keys is merely the use of a known technique to address a known vulnerability in Tajima. POSAs would have reasonably expected success in doing so as encrypting keys in systems like Tajima's had been done widely in the art and required nothing more than the application of routine and

conventional cryptography. EX1006, [0064]-[0068]; EX1016, 4:21-24; EX1021, 1:66-2:4; EX1023, 18:28-31; EX1040, 4; EX1018, [0032].

C. Foster’s key-encryption technique

151. A simple and straightforward technique for addressing the above-described known security risks associated with keeping plaintext keys with the encrypted data would have been to encrypt Tajima’s key information using a key-encrypting key (“KEK”) as described in Foster (EX1006). Foster, [0064]-[0068]. POSAs would have understood the desirability of Foster’s KEK technique in providing a simple way to encrypt Tajima’s keys while requiring little modification to Tajima’s system.

152. As shown by the steps that I have highlighted with blue shading in Foster’s Fig. 8 copied below, Foster divides digital content 100 into content units 102, each having a “content packet” or “payload 106” that is encrypted with a different encryption key 108, just like Tajima divides personal information into multiple data fragments that are each encrypted by a respective different encryption key. Foster, [0064]-[0068], [0071] (“each content packet can be encrypted with a different title key”); Tajima, [0111], [0113].

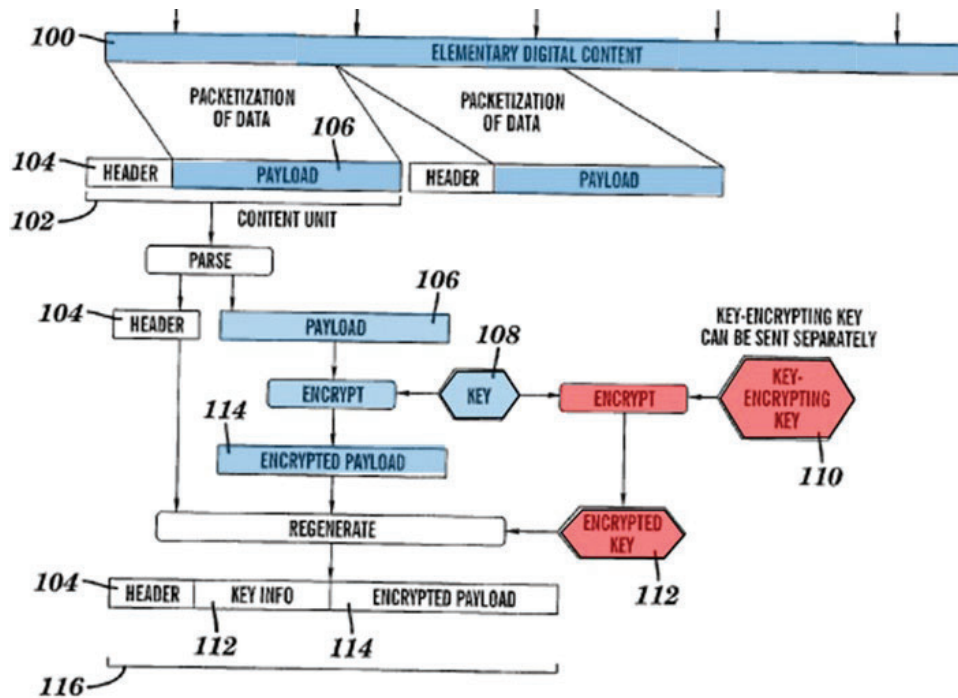


FIG. 8

153. As illustrated by the steps I shaded in **red** in Fig. 8 above, Foster teaches encrypting each key 108 with a key-encrypting key 110 so that encrypted key 112 can be securely stored/transmitted with the encrypted payload (see key info 112). Foster, [0064]-[0071]. That is, to safely include encryption keys with encrypted data, Foster teaches encrypting the encryption keys.

154. POSAs would have recognized the desirability of using Foster's key-encryption technique (i.e., the steps shaded in **red** above) to encrypt Tajima's keys before attaching them to a different encrypted fragment to benefit from the increased level of security that would prevent an attacker who successfully acquired all of Tajima's data fragment from possessing the plaintext key information needed to decrypt the data fragments as I discussed above in §§VIII.A-

VIII.B. A POSA also would also have recognized the desirability using Foster's key-encryption technique because little modification to Tajima's system would have been required to gain the above-described security benefit of both obfuscating and encrypting Tajima's keys to increase security of a user's personal information. For example, the similarity between Tajima's securing technique and Foster's steps highlighted in **blue** in Foster's Fig. 8 above demonstrate the simplicity of implementing Foster's KEK technique—i.e., an additional encryption step and storing of the KEK as discussed in further detail below in the discussion of the Tajima-Foster combination.

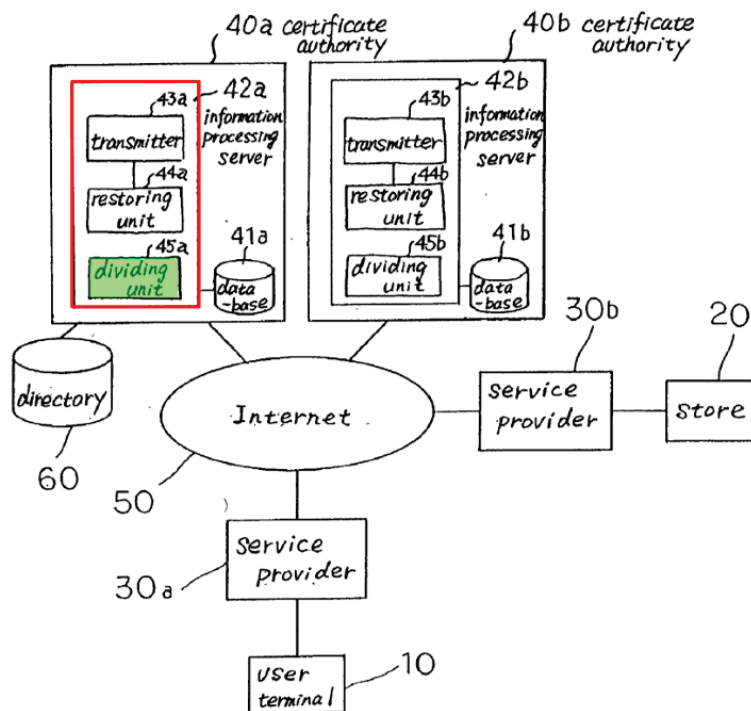
D. Tajima-Foster Combination

155. The obvious and desirable combination of Tajima and Foster ("Tajima-Foster") that a POSA would have had the above-described reasons to implement would have used (1) Tajima's technique of dividing personal information and securely storing data fragments on databases under control of different respective service sites; and (2) Foster's key-encryption technique (i.e., steps highlighted in **red** in Foster's Fig. 8 above)

156. As I discussed in the summary of Tajima in §VII.A above, Tajima's dividing unit implemented by each service site's information processing server (e.g., dividing unit 45a implemented by certificate authority 40a's server 42a in Fig. 5 below) performs Tajima's dividing techniques illustrated in Figs. 8-13,

including the Fig. 12 and 13 embodiments discussed in detail above. Tajima, [0067], [0100], [0111]-[0113], [0118], [0122]-[0123].

Fig. 5



157. Thus, to implement Foster's key encryption key (KEK) technique to prevent an attacker having acquired all data fragments from also possessing the complete set of keys in plaintext form that are needed to decrypt the data fragments as discussed in §§VIII.A-VIII.C, a POSA would have programmed Tajima's dividing units to also encrypt each data fragment's key information with a KEK as described in Foster and to maintain the KEK at the service site that divided and encrypted the personal information—i.e., at the service site where the personal information will be restored. Tajima, [0111]-[0114]. That is, a POSA would have understood that the KEK needs to be accessible by the authorized party recovering

the user's personal information, which in Tajima is the service site that initially secured the personal information. *Id.*; *see also* Foster, [0055]-[0056] (describing the plain fact that the KEK needs to be accessible by authorized recipients).

Because the service site tasked with dividing and distributively storing a user's personal information is the same service site that retrieves and restores the user's personal information, securely maintaining the KEK in Tajima is simplified—i.e., the service site need only securely store the KEK for the corresponding user's personal information.

158. As I discussed in the summary of Tajima in §VII.A above, Tajima discloses distributive storage techniques that differ primarily in whether or not the service site that divided the user's personal information also stores one of the resulting data fragments. Tajima-Foster would have improved security using either technique—i.e., both in the implementation in which a service site that divides personal information (via its dividing unit) stores one of the data fragments in its own database and distributes other data fragment to databases of other service sites (Tajima, [0073]-[0074]), and the implementation in which that service site distributes all resulting data fragments for storage on databases of other service sites and itself stores only link information to those destinations (Tajima, [0116], [0153], [0156]).

159. When the latter distributive storage technique is implemented, a POSA would have recognized an even further level of security because each of the data fragments *and* the KEK needed to decrypt them would all be stored on different databases under separate control of respective service sites. That is, the KEK stored at the service site that performed the dividing and encrypting of the data fragments would be secured separately from each of the encrypted data fragments stored at other service sites. Thus, an attacker would need to not only acquire each data fragment stored on multiple different databases under separate control, but to also gain unauthorized access to another system under separate control to acquire the KEK. This further Tajima's objective of increasing security by distributing the information needed to restore the user's personal information across multiple service sites that are independently controlled.

E. Reasonable Expectation of Success

160. A POSA would have had a reasonable expectation of success of implementing Tajima-Foster because the modification to Tajima is minimal and involved techniques that were already performed by Tajima and that were routine and conventional. Specifically, Tajima's dividing units were already configured to encrypt data with encryption keys (Tajima, [0067]-[0068], [0109], [0111], [0113], [0118], [0122]) such that configuring them to perform an additional encryption step (i.e., encrypting Tajima's keys using Foster's key-encryption technique)

would have been a straightforward modification requiring only routine skills that a POSA would have had a reasonable expectation of successfully implementing.

161. Additionally, as I discussed in the summary of Tajima in §VII.A above, the service site tasked with securing a user’s personal information already maintains “link information” identifying the registration destinations at which the data fragments are stored so that the service site can retrieve them when restoring a user’s personal information (Tajima, [0046], [0073], [0093], [0116], [0123], [0144], [0156]). Thus, each service site maintains link information for each instance of personal information that the service site secured for respective users—so also maintaining a KEK for each instance would impose little additional information management burden on the service site.

162. Indeed, Tajima discloses that the “link information may also be encrypted and held” (Tajima, [0093], [0144]) and so Tajima’s service sites already maintain encryption information as well. Thus, POSA’s would have recognized that the additional level of security imparted by encrypting Tajima’s encryption keys using Foster’s key-encryption technique would not have imposed any significant key management burden on the components of Tajima’s system.

163. Thus, for the reasons I discussed above, POSAs had good reason to employ Foster’s known technique of encrypting encryption keys in Tajima’s system and would have had a reasonable expectation of success in doing so.

F. Claim 1

1. 1[PRE]-[1D]

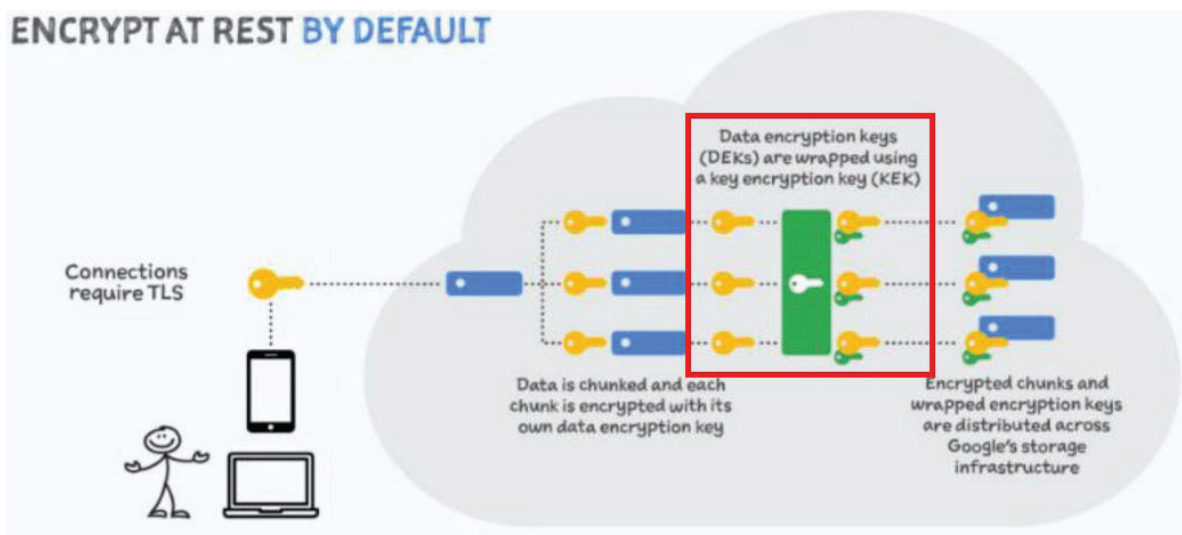
164. As I discussed above in §VIII.D describing Tajima-Foster, Tajima's system is modified to add the further encryption step of encrypting Tajima's key information with a KEK as taught by Foster. Accordingly, Tajima-Foster uses Tajima's dividing, encrypting, obfuscating and storing techniques, which meet 1[PRE]-[1D] for the same reasons I discussed in §§VII.B.1-VII.B.5 above. Tajima-Foster's further encrypting of Tajima's key information does not change how 1[PRE]-[1D] are met and therefore Tajima-Foster meets 1[PRE]-[1D] for the same reasons Tajima alone (or the obvious use of redundant storage in Tajima) does, and alternatively meets [1C] under Patent Owner's interpretation of the term "obfuscation" as discussed below.

2. [1C] obfuscating each of the plurality of different encryption keys

165. In addition to obfuscating by attaching key information to a different data fragment that that key information did not encrypt for the same reasons I discussed for [1C] in §VII.B.4 above in view of Tajima alone, Tajima-Foster additionally obfuscates Tajima's keys according to Patent Owner's interpretation advanced in the district litigation that "obfuscating" a key is met by encrypting it (EX1012, 4-5) and under Patent Owner's preliminary construction that an

“obfuscated key” is a key that has been rendered “unclear or obscure” (EX1043, 4-5).

166. Specifically, Patent Owner alleged in its infringement contentions that “*obfuscating each of the plurality of different encryption keys*” reads on encrypting data encryption keys (DEKs) with a key encryption key (KEK). EX1012, 4 (“Google *obfuscates* each of the different encryption keys *when it encrypts* the DEKs using the KEKs.”), 5 (“exemplary annotated image show[s] Google’s obfuscation of the DEKs by the KEKs.”); *see also* EX1011, ¶117. That is, Patent Owner alleges that an encryption key is obfuscated by encrypting it.



167. Additionally, encrypting Tajima’s keys with a KEK renders the keys “unclear or obscure” (EX1043, 5) as that is principally what encrypting does and the reason encrypting is performed. Thus, in my opinion, Tajima-Foster (§VIII.D) meets [1C] by storing key information with a data fragment other than the

fragment that key information was used to encrypt as discussed in §VII.B.4 above (i.e., in the way the '116 patent obfuscates and as Petitioner construes it), and additionally meets [1C] under SFI's interpretation that encrypting keys obfuscates them (EX1012, 4-5) renders them "unclear or obscure" (EX1034, 5). .

G. Claim 2: The method of claim 1, wherein obfuscating each of the plurality of different encryption keys comprises performing a transform on data indicative of at least one of the plurality of different encryption keys

168. In my opinion, Tajima-Foster meets claim 2 based on Patent Owner's interpretation of "*data indicative of*" a key as being met by the key itself. As discussed in §VII.D above in Ground 1 for claim 6, Patent Owner alleged that claim 2 is met by encrypting an encryption key—i.e., that "*performing a transform on data indicative of*" an encryption key is met by encrypting the key itself. EX1012, 8 ("[w]ithin Google's 'Encryption at Rest,' obfuscating each of the plurality of different encryption keys comprises performing a transform on data indicative of at least one of the plurality of different encryption keys. For example, Google *performs a transform on each of the different encryption keys when it encrypts the DEKs using the KEKs.*"). Patent Owner also proposed in its preliminary claim constructions that "transforming" means "changing the form" (EX1043, 5-6), which construction would be met by encrypting with a KEK because encryption changes the form of data from plaintext to ciphertext—i.e., from unencrypted to encrypted form.

169. Thus, in my opinion, Tajima-Foster (§VIII.D) meets claim 2 under Patent Owner’s infringement interpretation because Tajima-Foster encrypts Tajima’s key information (which includes at least the key used to encrypt a data fragment) with a KEK under Patent Owner’s preliminary construction because that encryption “chang[es] the form” of the key information from plaintext (unencrypted form) to ciphertext (encrypted form), thereby changing the form of each key in the respective key information from a plaintext key to an encrypted key. EX1042, 5-6; EX1012, 8. In my opinion, Tajima-Foster (§VIII.D) also meets claim 2 under Patent Owner’s construction of “*data indicative of*” because Tajima’s key information “makes known” or “shows” the key (EX1043, 3-4) as discussed above in §VII.D in connection with claim 6 in Ground 1.

H. Claim 9

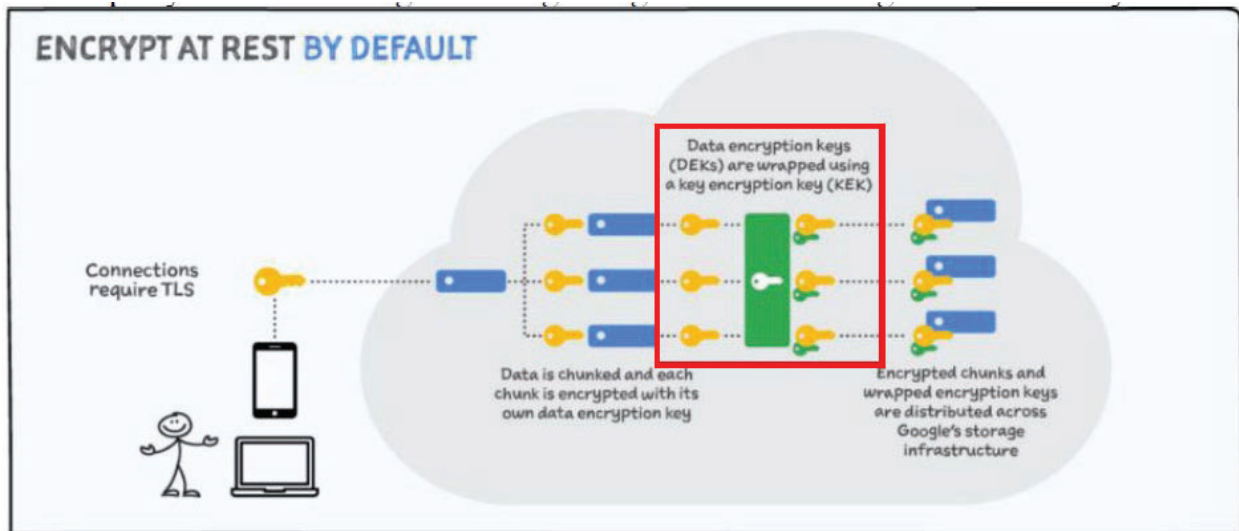
170. As I discussed in §V.F.1 (“Challenged Claims”), claim 9 differs from claim 1 in that claim 9 recites “transforming” encryption keys [9C] whereas claim 1 recites “obfuscating” them [1C]. Thus, as discussed in more detail below, Tajima-Foster meets 9[PRE]-[9B] and [9D] for the same reasons Tajima-Foster and Tajima alone meet 1[PRE]-[1B] and [1D] (see §§VII.B.1-VII.B.5 and VIII.F.1 above), and meets [9C] under Patent Owner’s interpretation because Tajima-Foster encrypts Tajima’s key information (see §VIII.H.2 below).

1. 9[PRE]-[9B]

171. 9[PRE]-[9B] recite the identical limitations recited in 1[PRE]-[1B], and therefore are met by Tajima-Foster for the same reasons Tajima alone meets 1[PRE]-[1B] as discussed in §§VII.B.1-VII.B.3 above, which are the same reasons Tajima-Foster also meets 1[PRE]-[1B] as discussed in §VIII.F.1 above.

2. [9C] transforming each of the plurality of different encryption keys

172. As I discussed above in §VIII.D describing the Tajima-Foster system, Tajima-Foster encrypts the key information used to encrypt individual data fragments with a KEK (Foster’s key-encrypting technique), thereby encrypting each of the keys contained therein. As discussed in §VIII.G above in connection with claim 2, Patent Owner alleged that encrypting an encryption key transforms it. In alleging infringement of claim 9, Patent Owner alleged that the “transforming” limitation of [9C] is met because “Google transforms each of the different encryption keys when it encrypts the DEKs using the KEKs.” EX1012, 12 (also referencing the drawing below). As I discussed in §VIII.G above in connection with claim 2’s “transform” limitation, Patent Owner proposed a preliminary construction for “transforming” as “changing the form” of the data. EX1043, 5-6.



173. Thus, in my opinion, Tajima-Foster meets [9C] under Patent Owner’s infringement interpretation and preliminary claim construction because Tajima-Foster encrypts Tajima’s keys (i.e., by encrypting Tajima’s key information using Foster’s key-encryption technique as discussed in §VIII.D above) which “changes the form” of each of Tajima’s keys from unencrypted to encrypted form. EX1043, 5-6; EX1012, 12.

3. [9D] separately storing each data chunk of the plurality of data chunks together with one of the plurality of transformed different encryption keys on a plurality of different storage devices

174. As I discussed in §VIII.D above, Tajima-Foster uses Tajima’s technique of storing each encrypted data fragment attached to key information that encrypted a different data fragment on separate databases “under the control” of different certificate authorities. That (or the obvious use of redundant storage in

Tajima) meets the “*separately storing*” limitation of [1D] as discussed in §VII.B.5. Tajima, [0048], [0111], [0113], [0116], [0153], [0156].

175. Limitation [9D] differs from [1D] in that the stored encryption keys are “*transformed*” encryption keys. In Tajima-Foster (§VIII.D), each encrypted encryption key, which is a “*transformed*” encryption key under Patent Owner’s interpretation that encrypting keys transforms them (see [9C]’s “*transforming*” step in §VIII.H.2 above), is attached to a data fragment it did not encrypt and stored on a respective different database. Therefore, in my opinion, Tajima-Foster meets [9D]’s “*separately storing*” of “*transformed*” keys for the same reasons Tajima alone (or the obvious use of redundant storage in Tajima) (§VII.B.5) and Tajima-Foster (§VIII.F.1) meet [1D]’s “*separately storing*” of “*obfuscated*” key.

I. Claim 11: The method of claim 9, wherein storing the plurality of data chunks with the plurality of transformed different encryption keys comprises storing each of the plurality of data chunks with a respective one of the plurality of transformed different encryption keys

176. In my opinion, Tajima-Foster (see §VIII.D above describing Tajima-Foster) meets claim 11 because each of Tajima’s data fragments is stored with a different (i.e., “*respective*”) one of Tajima-Foster’s encrypted (i.e., “*transformed*”) encryption keys, which is all that claim 11 requires. In this respect, claim 11’s storing “*with a respective one of*” limitation mirrors claim 5’s storing “*with a*

respective one of” and requires nothing more for the same reasons discussed in §VII.C above in connection with claim 5.

J. Claim 12: The method of claim 9, wherein storing the plurality of data chunks with the plurality of transformed different encryption keys comprises including, with a data chunk of the plurality of data chunks, data indicative of an encryption key of the plurality of encryption keys that was used to encrypt a different data chunk of the plurality of data chunks

177. As discussed in §VIII.D above, Tajima-Foster stores each data fragment with encrypted key information that includes the encryption key used to encrypt a different data fragment—i.e., Tajima’s storing technique described the Fig. 12 and 13 embodiments (Tajima, [0111], [0113]). In my opinion, Tajima-Foster therefore meets claim 12 at least under Patent Owner’s interpretation of “*data indicative of an encryption key*” as I discussed in connection with the same limitation in claim 6 in §VII.D above—i.e., because Tajima-Foster’s encrypted key information includes at least the corresponding encrypted key (EX1012, 8) that is attached to and stored with a non-corresponding encrypted data fragment, and Tajima-Foster’s key information “shows” or “makes known” the corresponding key (EX1043, 3-4).

K. Claim 13: The method of claim 9, wherein distributing the data set into a plurality of data chunks comprises using a random technique or pseudorandom technique

178. Tajima-Foster does not change the way in which Tajima distributes divided portions of personal information into the data fragments (see my discussion of Tajima-Foster in §VIII.D above)—i.e., Tajima-Foster also uses random numbers to divide personal information into data portions that are distributed into different data fragments (Tajima, [0106], [0111], [0113]). Therefore, it is my opinion that a POSA would have understood Tajima-Foster to have met claim 13 for the same reasons Tajima meets the identical limitation in claim 7 (see my discussion of claim 7 in §VII.E above).

L. Claim 14: The method of claim 9, further comprising generating the data set by encrypting primary data

179. As I discussed in §VIII.D above, Tajima-Foster also uses Tajima's technique of encrypting the personal information before dividing it (Tajima, [0113]) and thus, in my opinion, a POSA would have understood Tajima-Foster to have met claim 14 for the same reasons a POSA understood Tajima meets claim 8 discussed in §VII.F above.

IX. GROUND 3: TAJIMA-RIVEST (CLAIMS 9-14)

180. I have been asked to provide my opinion concerning whether claims 9-14 of the '116 patent would have been obvious to a POSA in light of the disclosures of Tajima, Rivest (EX1034), and the knowledge of a POSA. For the

reasons explained below (§§IX.D-IX.I), it is my opinion that each of claims 9-14 would have been obvious to a POSA over the combination of Tajima and Rivest.

A. Public Accessibility of Rivest (EX1034)

181. Rivest is titled “All-or-Nothing Encryption and the Package Transform.” Based on the evidence I discuss below, it is my opinion that Rivest is prior art to the challenged claims because it was publicly accessible before the earliest-alleged priority date of the ’116 patent (October 25, 2004) by being disseminated or otherwise made available to POSAs exercising reasonable diligence.

182. According to Rivest’s cover information, Rivest was published in the Lecture Notes in Computer Science (LNCS) 1267, a renowned series published by well-known and reputable publisher Springer-Verlag, that publishes proceedings papers—here the Proceedings of the 4th Intl. Workshop on Fast Software Encryption of 1997 (FSE’97). Rivest, 1. Rivest bears the text “© Springer-Verlag Berlin Heidelberg 1997” indicating that Rivest was copyrighted and thus publicly accessible by the year 1997 and includes corroborating serial (ISSN 0302-9743) and book numbers (ISBN 3-540-63247-6). Rivest, 3. I also note that Rivest is a stamped copy of the Wendt Library’s (University of Wisconsin – Madison) bound copy of LNCS 1267 (Rivest, 3-4) indicating this in an authentic copy that was

received by the library by September 1997 (Rivest, 4) and would have been publicly accessible by or around that date.

183. Additionally, Rivest is a well-known and oft-cited² paper describing the “all-or-nothing” (AoNT) and package transforms developed and named by its author, renowned cryptography pioneer and MIT Professor, Ronald Rivest, whose techniques were referenced in the ’116 patent itself (see e.g., EX1001, 77:13-17). I also note that Patent Owner cited Rivest as “Rivest, “All-Or-Nothing Encryption and the Package Transform,” Proc. Of the 4th Intl. Workshop on Fast Software Encryption (1997)” (EX1002, 286), corroborating Rivest as prior art published in 1997. In my opinion, a POSA would have been well-familiar with Ronald Rivest and AoNT and if not aware of the EX1034 publication would have been able to locate it prior to the ’116 patent’s earliest-alleged priority date, as would have any member of the public interested in the subject matter. I was personally familiar with AoNT and the Rivest paper.

² SpringerLink lists 197 cites to Rivest from a wide-variety of well-known publications, 26 of those cites were from publications before the ’116 patent’s earliest-alleged priority date of October 25, 2004.

<https://citations.springernature.com/item?doi=10.1007/BFb0052348>

B. Reasons to transform Tajima’s key information

184. As discussed above in §§VIII.A-VIII.B, keeping plaintext keys with encrypted data presented a known security risk that a POSA would have been motivated to address in Tajima to guard against an attacker that successively acquires all of Tajima’s encrypted data segments. As further discussed in §VIII.A, POSA’s were aware of multiple techniques suitable for addressing this vulnerability, including by applying known “all-or-nothing” transforms (AoNT) developed by Rivest (EX1034). Indeed, the ’116 patent acknowledges that AoNT was “well-known” and that “*it will be understood* that the AoNT may be applied to encryption keys used to encrypt the data portions following parsing and splitting.” EX1001, 77:13-43.

185. A POSA would have been motivated to apply AoNT (Rivest, 212-214) to transform Tajima’s key information (which includes Tajima’s encryption keys) to address security risks of keeping plaintext keys with encrypted data (e.g., for the reasons a POSA would similarly have been motivated to encrypt Tajima’s key information as discussed in §VIII.B above)—i.e., to prevent an attacker who has obtained all of Tajima’s data fragments from possessing all of Tajima’s keys in clear form. A POSA would have recognized that using AoNT beneficially provides protection for Tajima’s keys without needing to maintain a KEK and keep it secret. Rivest, Rivest, 212 (“all-or-nothing transform...makes no use of any secret key

information”)-214. Additionally, applying AoNT to Tajima would have merely been the obvious use of a known technique (AoNT) according to its intended purpose (transforming plaintext to a different and undecipherable form) to yield the predictable result of providing transformed keys that can be attached to encrypted data in a secure manner to achieve the objectives of Tajima in securing a user’s personal information.

C. Tajima-Rivest Combination

186. Rivest discloses a two-step transform that transforms a message m (i.e., the data to be transformed) composed of blocks $m_1, m_2, \dots m_s$ into a transformed message m' with the properties that the transformation is reversible, efficiently computable and it is computationally infeasible to recover m if any of the blocks are unknown. Rivest, 212-213. In a first step, an output sequence $m'_1, m'_2, \dots m'_s$ is computed as follows:

$$\text{Let } m'_i = m_i \oplus E(K', i) \text{ for } i = 1, 2, 3, \dots, s.$$

187. Here, K' is a random key chosen from a large space. In a second step an additional block $m'_{s'}$ is computed as the exclusive-or of the key K' and a hash of all previous computed m' blocks as follows:

Let

$$m'_{s'} = K' \oplus h_1 \oplus h_2 \oplus \dots \oplus h_s ,$$

where

$$h_i = E(K_0, m'_i \oplus i) \text{ for } i = 1, 2, \dots, s ,$$

188. Here, K_0 is a publicly known encryption key and K' “is not a secret share key” because “it is disclosed in the pseudo-message”. Rivest, 214. The original message m is recoverable by inverting the transform as follows:

$$K' = m'_s \oplus h_1 \oplus h_2 \oplus \dots \oplus h_s ,$$

$$m_i = m'_i \oplus E(K', i) \text{ for } i = 1, 2, \dots, s .$$

189. Thus, if any block of the transformed message m' is unknown, K' cannot be computed so it is infeasible to compute any message block and m cannot be recovered. A POSA would have understood multiple suitable implementations for applying Rivest’s transform to Tajima’s to transform Tajima’s key information. As the transform of the first step demonstrates, both the message m and the key K' are transformed in the same way. Thus, Tajima’s key information may be transformed either by using Tajima’s key information as the input message m or K' . In a first implementation, message m is Tajima’s key information and a K' is generated as disclosed (i.e., a random number). It would have been obvious and beneficial to use the user’s public key as the public key K_0 , which the service site receives from the user submitting personal information as so already has this key in its possession when securing the user’s personal information. Tajima, [0070]-[0071], Fig. 6 (see steps S2 and S3).

190. In a second implementation, message m is the encrypted data fragment and K' is Tajima’s key information that includes Tajima’s encryption

key, and the user's public key is used as K_0 as in the first implementation, obviating the need to generate a K' and providing the additional benefit of transforming the encrypted data in Tajima's data fragments. In both implementations, Tajima's key information is transformed using Rivest's AoNT technique (EX1034, 212-214) to provide a further layer of protection to Tajima by protecting Tajima's key—i.e., Tajima's key information is transformed according to K' in the first implementation and according to the encrypted data in the second implementation.

191. As I discussed above in IX.B, a POSA would have recognized that using AoNT beneficially provides protection for Tajima's keys without needing to maintaining separate secret keys. Rivest, 212-214. That is, the user's public key provided to the service site (Tajima, [0070]-[0071], Fig. 6 (see steps S2 and S3) can be applied as K_0 and either Tajima's keys themselves or random numbers can be applied as K' in Rivest's transform, the former of which would not thus be a secret shared key for the transform because it is disclosed in the pseudo-message in the same way the latter is. Rivest, 214. That is, Tajima's keys and random numbers are both secrets because both are unknown but are not "a secret shared key" in the transform for the same reason—i.e., by being "disclosed in the pseudo-message"—and both serve the same purpose. Rivest, 212-214.

192. A POSA would have had a reasonable expectation of success because implementing Tajima-Rivest requires nothing more than applying a known technique involving routine and conventional cryptography techniques. Indeed, implementing Tajima-Rivest as described above merely involves performing a sequence of exclusive-or operations in a prescribed manner, which was well within the capabilities of a POSA.

D. Claim 9

193. In my opinion, Tajima-Rivest (§IX.C) renders obvious claim 9 because transforming Tajima’s key information, which includes Tajima’s keys (see §VII.B.4 above), using Rivest’s AoNT techniques meets “*transforming each of the plurality of different encryption keys,*” as recited in [9C]. Specifically, Tajima-Rivest’s use of AoNT to transform Tajima’s key information changes the bits (and likewise changes the form) of Tajima’s keys without using a secret key (Rivest, 212-214), meeting [9C]’s “transforming” limitation under both Petitioner’s construction (EX1042, 8: “Changing without the use of a secret key the bits of a key”) and Patent Owner’s construction (EX1043, 5-6: “changing the form”).

194. Each of claim 9’s other limitations are met for the same reasons discussed in §§VIII.H.1 and VIII.H.3 above because the manner in which Tajima’s keys are transformed does not change the way 9[PRE]-[9B] and [9D] are met—i.e., because Tajima-Rivest uses Tajima’s dividing, encrypting and distributive storage

techniques that meet these limitations (see §§VII.B.1-VII.B.3, VII.B.5 above), where [9D]’s “*transformed encryption keys*” are met by Tajima’s keys transformed by Rivest’s AoNT techniques. Rivest, 212-214.

E. Claim 10: The method of claim 9, wherein transforming each of the plurality of different encryption keys comprises performing an all-or-nothing transform on data indicative of at least one of the plurality of different encryption keys

195. Tajima-Rivest (§IX.C) transforms Tajima’s key information using AoNT and therefore it is my opinion that Tajima-Rivest meet claim 10 because Tajima-Rivest’s transforming of keys “*comprises performing an all-or-nothing transform*” (i.e., by applying Rivest’s AoNT techniques) “*on data indicative of at least one of the plurality of different encryption keys*”—i.e., Tajima’s keys under Patent Owner’s interpretation that “*data indicative of*” a key is met by the key itself. EX1012, 8 (showing Patent Owner’s mapping of data indicative of a key to the DEK itself); *see also* §§VII.D, VIII.G above.

F. Claim 11: The method of claim 9, wherein storing the plurality of data chunks with the plurality of transformed different encryption keys comprises storing each of the plurality of data chunks with a respective one of the plurality of transformed different encryption keys

196. In my opinion, Tajima-Rivest (§IX.C) meets claim 11 because in Tajima-Rivest each of Tajima’s data fragments is stored with a different (i.e., “*respective*”) one of Tajima-Rivest’s “*transformed different encryption keys*”—i.e.,

Tajima's encryption keys transformed using Rivest's AoNT techniques, which is all that claim 11 requires as discussed in §VIII.I above.

G. Claim 12: The method of claim 9, wherein storing the plurality of data chunks with the plurality of transformed different encryption keys comprises including, with a data chunk of the plurality of data chunks, data indicative of an encryption key of the plurality of encryption keys that was used to encrypt a different data chunk of the plurality of data chunks

197. In my opinion, Tajima-Rivest (§IX.C) meets claims 12 because Tajima-Rivest stores each data fragment with transformed key information that includes an encryption key used to encrypt a different data fragment (i.e., Tajima's Fig. 12 and 13 storing technique) and thus each data fragment includes "*data indicative of an encryption key of the plurality of encryption keys that was used to encrypt a different data chunk of the plurality of data chunks,*" as recited in claim 12 at least under Patent Owner's interpretation that "*data indicative of an encryption key*" is met by the key itself as discussed in §VIII.J above. EX1012, 8 (showing Patent Owner's mapping of data indicative of a key to the DEK itself); *see also* §§VII.D, VIII.G above.

H. Claim 13: The method of claim 9, wherein distributing the data set into a plurality of data chunks comprises using a random technique or pseudorandom technique

198. In my opinion, Tajima-Rivest (§IX.C) meets claim 13 for the same reasons discussed in §§VII.E and VIII.K above because Tajima-Rivest employs

Tajima's technique of using random numbers to divide personal information into data portions that are distributed into different data fragments (Tajima, [0106], [0111], [0113]).

I. Claim 14: The method of claim 9, further comprising generating the data set by encrypting primary data

199. In my opinion, Tajima-Rivest (§IX.C) meets claim 14 for the same reasons discussed in §§VII.F and VIII.L above because Tajima-Rivest employs Tajima's technique of optionally encrypting the personal information before dividing it (Tajima, [0113]).

X. GROUND 4: TAJIMA-RIVEST-SHYU (CLAIMS 1-3, 6)

200. I have been asked to provide my opinion concerning whether claims 1-3 and of the '116 patent would have been obvious to a POSA in light of the disclosures of Tajima, Rivest and Shyu (EX1016) and the knowledge of a POSA. For the reasons explained below (§§X.C-X.F), it is my opinion that each of claims 1-3 and 6 would have been obvious to a POSA over the combination of Tajima, Rivest and Shyu.

A. Reasons to Vary Tajima's Encryption Algorithms

201. As I discussed in §VIII.A above, multiple techniques would have been known to a POSA and recognized as beneficial in mitigating risks associated with storing encryption keys together with encrypted data, including varying the type of

encryption applied with respective encryption keys on different portions of a data set being secured.

202. Shyu (EX1016) recognized that conventional techniques that used the same encryption algorithm to all data of a data set being secured puts the data being secured at risk should the encryption scheme be broken by hackers (Shyu (EX1016), 1:24-42) and so teaches dynamically changing the key and the scrambling algorithm for different data segments to enhance security and storing the key and an indication of the scrambling algorithm (i.e., a code indicating the scrambling type) in a header attached to the data segment. Shyu (EX1016), 3:12-38, 3:56-60 (“Therefore, the source data segments are scrambled using different scrambling algorithms and different key words that are recorded in the respective segment header to effectively prevent piracy of the original source data”).

203. Takeda (EX1036) recognized this same risk—i.e., that conventional techniques that applied the same encryption algorithm to all blocks of data being secured risked of the data being deciphered if part of the data is successfully deciphered by an unauthorized party (Takeda (EX1036), [0003]-[0004]) and likewise teaches using different ciphering algorithms for different blocks of a data set being secured so that “even if part of the ciphertext can be deciphered, the rest cannot be deciphered” (Takeda (EX1036), [0009]).

204. A POSA would have recognized the benefit of applying different encryption algorithms to Tajima's different data fragments as taught by Shyu (and as also taught by Takeda) to avoid known security risks of using a fixed encryption algorithm. Shyu (EX1016), 1:24-42, 3:12-60; Takeda (EX1036), [0003]-[0009]. A POSA would have recognized the benefit of both varying the encryption algorithm used to encrypt Tajima's different data fragments as taught by Shyu and transforming the encryption keys using Rivest's AoNT techniques as in Tajima-Rivest discussed above in §IX.C to provide multiple layers of protection for Tajima's keys like Tajima teaches providing multiple layers of security for the data so that failure of any one layer will not result in comprising a user's personal information. Indeed, Shyu teaches stacking variable encryption (Shyu (EX1016), 3:12-60) with scrambling the header that includes the keys to provide multi-layer security against unauthorized access to data (Shyu (EX1016), 4:21-28), demonstrating that using both variable encryption techniques and rendering keys undecipherable together was both known and desirable. Thus, a POSA would have recognized the benefit of applying Shyu's variable encryption in Tajima-Rivest (see §IX.C above).

205. Moreover, it would have obvious to vary the encryption algorithm in Tajima-Rivest's system as doing so is merely the use of a known technique of using different encryption algorithms on different portions of the data (e.g., as

disclosed in Shyu and Takeda) according to its intended purpose (to improve security) to yield the predictable result of making it more difficult for an attacker that has acquired multiple data fragments to successfully decrypt them to achieve Tajima's objective of securing a user's personal information.

B. Tajima-Rivest-Shyu Combination

206. Tajima-Rivest-Shyu uses Tajima's dividing, encrypting and distributive storage techniques of the Fig. 12 and 13 embodiments (Tajima, [0111]-[0114]) and further (1) applies Shyu's variable-encryption technique when performing Tajima's step of encrypting each data fragment with a different key—i.e., encrypts Tajima's data fragments using Shyu's variable-encryption technique with Tajima's different keys—and stores metadata identifying what encryption algorithm the corresponding key is used with as taught by Shyu (Shyu, 3:12-26, 3:56-60) (and optionally additional metadata) in Tajima's key information along with the corresponding key; and (2) transforms the key information using AoNT as in Tajima-Rivest (see §IX.C above). The resulting transformed key information is attached to a different data fragment than the data fragment to which it was applied—i.e., Tajima-Rivest-Shyu uses Tajima's obfuscating technique of the Fig. 12 and Fig. 13 embodiments (see, e.g., §VII.B.4 above)—and distributively stores the encrypted data fragment/transformed key information pairs on databases of different service sites as taught by Tajima (see, e.g., §VII.B.5 above).

207. As discussed in §VII.B.4, a POSA would have understood Tajima's key information to include at least the key and may include additional information. Per the above, Tajima-Rivest-Shyu's key information includes the encryption key and additionally includes at least an indication of the encryption algorithm used with the corresponding key as taught by Shyu (storing scrambling type and the corresponding key in the header) to implement Shyu's variable-encryption technique. Shyu (EX1016), 3:12-60, FIG. 3. It would have been obvious to include additional metadata, including any of Shyu's metadata stored in the segment header to assist in parsing and/or metadata of the key that may be used to increase security, facilitate parsing of the key information by the authorized service site, etc. As discussed in §VII.D above (Ground 1, claim 6), a POSA would have recognized the benefit of including a key length (see, e.g., Takeda, [0053]-[0054], [0077]) in Tajima's key information to allow for the use of variable-length keys to provide additional protection and/or flexibility (e.g., allowing a service site to use different size keys for different instances of securing different user's personal information, allowing a service site to change key size as appropriate or desired without changing the decryption process, etc.).

C. Claim 1

208. Tajima-Rivest-Shyu (§X.B) uses Tajima's dividing, encryption and distributive storing of key information with non-corresponding encrypted data

fragments on separate service sites as disclosed in the Fig. 12 and 13 embodiments and therefore, in my opinion, renders obvious claim 1 for the same reasons Tajima alone meets claim 1 as demonstrated in §VII.B above because additionally using (1) Shyu’s variable-encryption technique when performing Tajima’s data fragment encryption process and (2) Rivest’s AoNT to transform Tajima’s resulting key information does not change the way claim 1’s limitations are met.

D. Claim 2: The method of claim 1, wherein obfuscating each of the plurality of different encryption keys comprises performing a transform on data indicative of at least one of the plurality of different encryption keys

209. As discussed in §X.B above, Tajima-Rivest-Shyu’s key information includes the encryption key and metadata of the key including at least the encryption type used with the corresponding key to implement Shyu’s variable-encryption technique and optionally the key length of the corresponding key to facilitate use of variable-length key (see, e.g., Takeda (EX1036)), either of which alone or together is “*data indicative of at least one of the plurality of different encryption keys.*” Tajima-Rivest-Shyu (§X.B above) then transforms that key information using AoNT and therefore “*perform[s] a transform*” (i.e., AoNT) on “*data indicative of at least one of the plurality of different encryption keys*” (i.e., Tajima’s key information including metadata about the key) and therefore meets claim 2 under both Petitioner’s and Patent Owner’s constructions because (1) AoNT is a “*transform*” under both parties’ constructions as discussed in §IX.D

above in connection with claim 9's transforming step, and (2) Tajima's key information includes the corresponding key and metadata about the key (e.g., the encryption type used with the key, key length, etc.) and therefore is "*data indicative of*" the corresponding key under Patent Owner's interpretation (EX1012, 8; EX1043, 3-4; §VII.D) and Petitioner's construction requiring "metadata about the key" (EX1042, 5-6; §VII.D).

E. Claim 3: The method of claim 2, wherein performing a transformation on data indicative of at least one of the plurality of different encryption keys comprises performing an all-or-nothing transform on the data indicative of at least one of the plurality of different encryption keys

210. As I discussed above for claim 2, Tajima-Rivest-Shyu (§X.B) applies AoNT to Tajima-Rivest-Shyu's key information that includes Tajima's key and corresponding metadata of the key (e.g., scrambling type and/or key length) and therefore, in my opinion, Takeda-Rivest-Shyu's AoNT technique "*comprises performing an all-or-nothing transform on the data indicative of at least one of the plurality of different encryption keys,*" as recited in claim 3 under both parties' constructions as discussed above in §§VII.D, IX.C-IX.D and X.D. EX1042, 5-6, 8; EX1043, 3-6; EX1012, 8.

F. Claim 6: The method of claim 1, wherein storing the plurality of data chunks together with the plurality of obfuscated different encryption keys comprises including, with a data chunk of the plurality of data chunks, data indicative of an encryption key that was used to encrypt a different data chunk of the plurality of data chunks

211. Tajima-Rivest-Shyu (§X.B) stores Tajima-Rivest-Shyu’s key information (i.e., the key, corresponding scrambling type and optionally key length) used to encrypt a data fragment with a different fragment that it did not encrypt and, in my opinion, therefore meets “*including, with a data chunk...data indicative of an encryption key that was used to encrypt a different data chunk of the plurality of data chunks,*” as recited in claim 6 under both parties constructions as discussed in §§VII.D and X.D above. EX1042, 5-6, 8; EX1043, 3-6; EX1012, 8.

XI. GROUND 5: FOSTER-HAYHURST (CLAIMS 1-2, 5, 9, 11)

212. I have been asked to provide my opinion concerning whether claims 1-2, 5, 9 and 11 of the ’116 patent would have been obvious to a POSA in light of the disclosures of Foster, Hayhurst, and the knowledge of a POSA. For the reasons explained below (§§XI.D-XI.H), it is my opinion that each of claims 1-2, 5, 9 and 11 would have been obvious to a POSA over the combination of Foster and Hayhurst.

A. Foster (EX1006)

213. Foster discloses methods for securing digital content, like a data set for a digital movie, using encryption techniques that “attach[] a title [encryption]

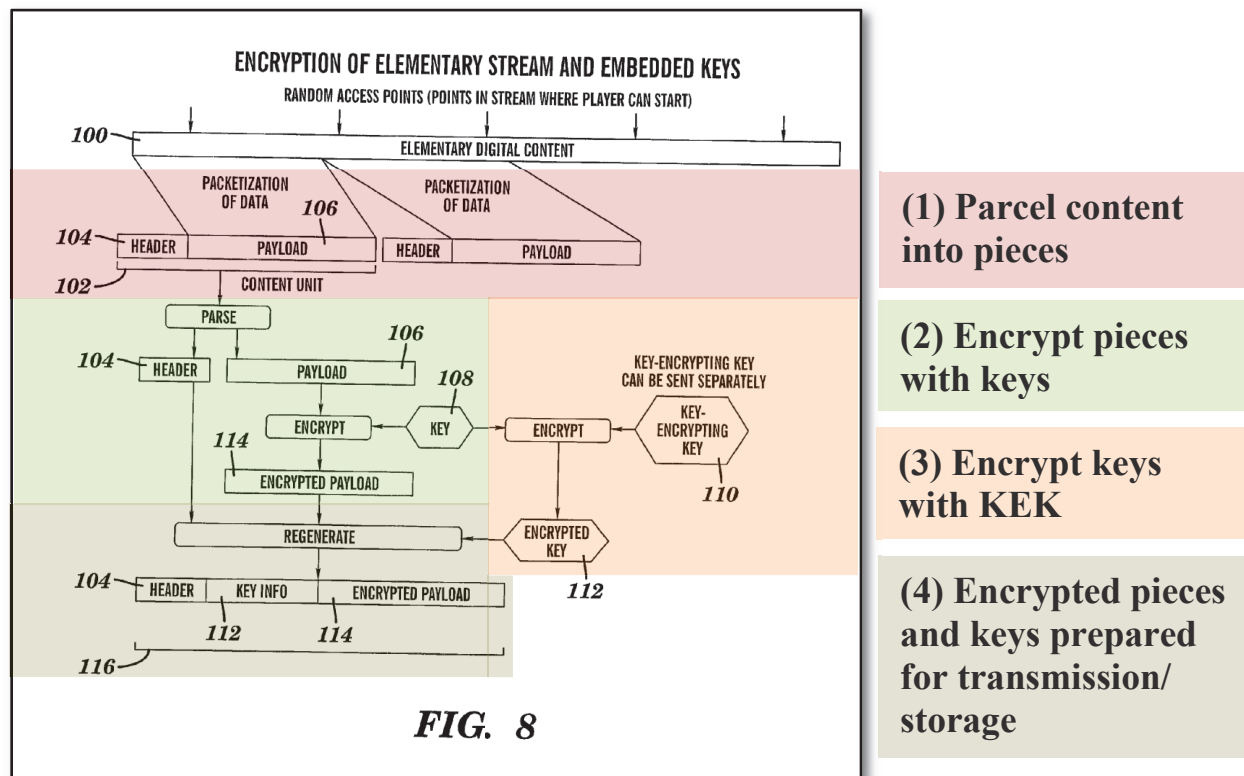
key to encrypted content for synchronized transmission to, or storage by, a recipient.” Foster (EX1006), Abstract, [0002]. Foster’s content includes “*any* data such as digital image, sound or binary data.” Foster, [0037].

214. Foster teaches that prior-art encryption methods used “multiple keys” to “provide[] enhanced security by preventing an entire content stream from being accessed with a single key.” Foster, [0004]. Those techniques, however, typically stored encryption keys separately from the encrypted content portions, which required subsequent “synchronization” of each encrypted portions with its corresponding encryption keys—i.e., a receiver such as a set-top box needed to pair each encryption key with the correct encrypted data portion stored/received separately from the keys (e.g., using bit flags or other synchronizing information). Foster, [0004]-[0006]. Separate encryption keys and encrypted data also created challenges in “providing random access” to portions of content. *Id.*

215. To address these and other deficiencies in the prior art, Foster’s encryption methods involve (1) dividing content up into multiple content packets, (2) encrypting each content packet with its own distinct “title” (encryption) key, (3) encrypting those distinct title keys with a common key-encryption-key (“KEK”), (4) attaching the now-encrypted distinct title keys to their respective content packets, and (5) transmitting the content packets to receivers (e.g., storage device in a consumer’s home). Foster, [0064], [0040]-[0041] (defining a

“Recipient” as including “a receiver” and a “Receiver” as including “a consumer device, such as a set-top box, a DVD player, etc.”); FIG. 8.

216. An embodiment of Foster’s method is illustrated in Fig. 8 (which I have annotated below):



217. First, “digital content 100” is “parceled into content units 102.” Foster (EX1006), [0064]. Each content unit comprises a “payload” portion of the original data set (also referred to as “content packet 106”) and a “header 104,” which “includes information helpful in identifying the content” (e.g., “type of content”). Foster (EX1006), [0064]; FIG. 8.

218. Second, payload portions are “encrypted with title key 108” to “yield encrypted content packet 114.” Foster, [0064]. Rather than using a single common title key for each portion, Foster teaches that “each content packet could be encrypted with a *different* title key” (Foster, [0065]), which “provides enhanced security by preventing an entire content stream from being accessed with a single key.” Foster, [0004], [0071] (“In addition, because each content packet can be encrypted with a different title key, a party who illicitly obtains a title key cannot access the entire stream. In contrast, the pirated key is only good for decrypting the content packet to which it was attached.”)

219. Third, each title key is encrypted with a common “key encrypting key 110” (*i.e.*, a KEK, which were (and are) well-known encryption keys that I discussed above in §VIII for Ground 2) to generate encrypted title key 112, which can be “attached to the encrypted content as a header extension” that includes additional information such as optional identifier information 120 and optional verifier 122. Foster (EX1006), [0048], [0064], [0066], FIGS. 8-10.

220. Fourth (and finally), the encrypted title key, encrypted content packet, and header (with header extension) are combined as a “processed content unit

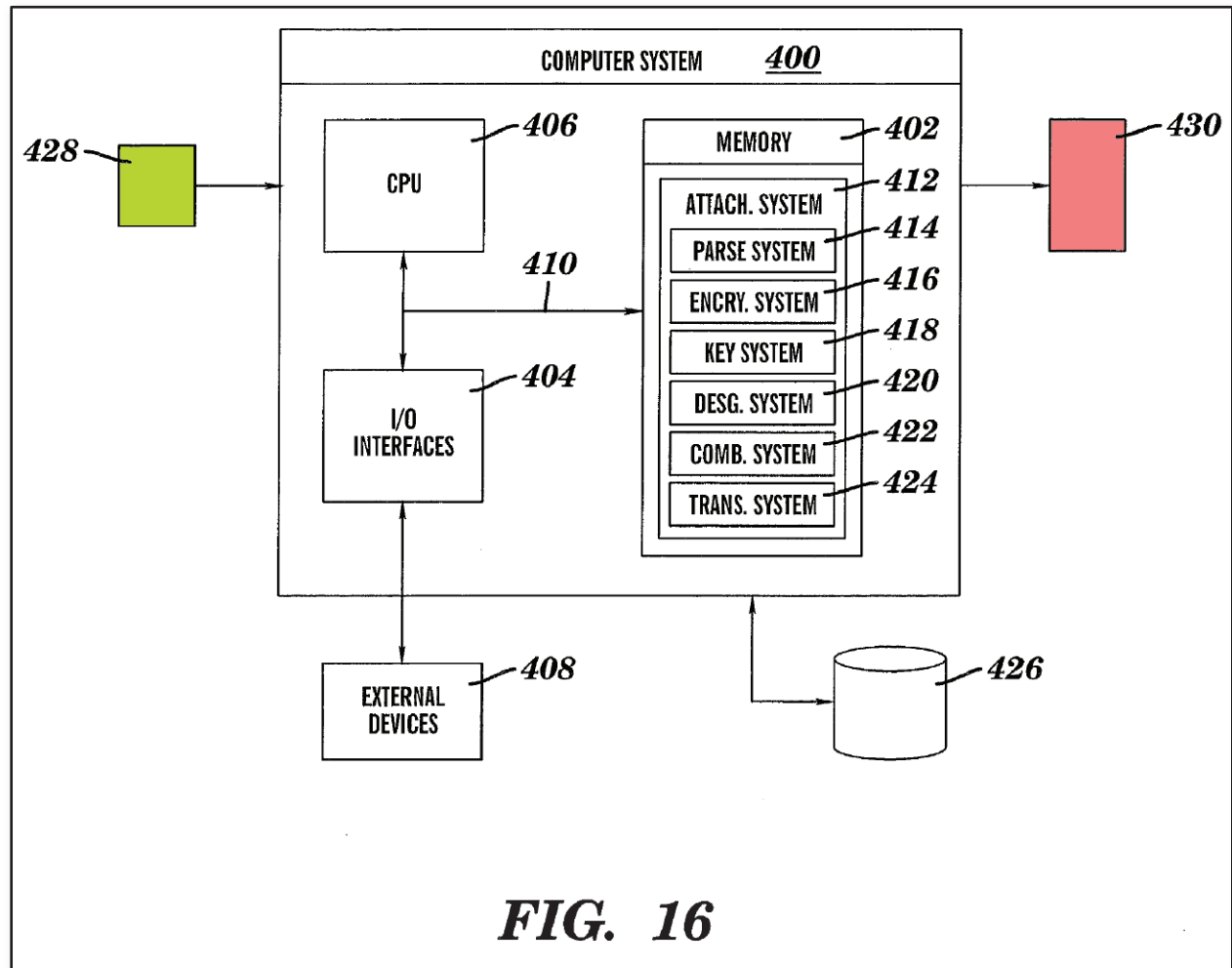
116,” which can be “stored by[] a recipient” in a receiver (*e.g.*, stored in a consumer’s set-top box). Foster (EX1006), [0064], [0040]-[0041].³

221. Foster’s use of different encryption keys benefits from the known “enhanced security” of “preventing an entire content stream from being accessed with a single key” (Foster, [0004]) and Foster’s KEK technique allowed encryption keys to be securely stored/transmitted with the encrypted data, addressing problems with conventional content delivery systems that stored/transmitted keys separately from the encrypted data. Foster, [0004], [0007]-[0008], [0064]-[0068], [0071].

222. Foster’s Fig. 16 (which I have annotated below) depicts an exemplary system in which its encryption technique can be employed. Foster, [0030], [0075]. The system includes a “source 428” (which I annotated in **green**), which can be a “content owner...such as a movie studio.” Foster (EX1006), [0038], [0042], [0078]. It also includes “recipient 430” (which I annotated in **red**), which can be a “receiver” such as a “set-top box” or other “consumer device” that receives content

³ As part of transmitting encrypted content to a storage device, Foster may further divide data based on the communication protocol (*e.g.*, TCP/IP protocol), but that division of data (*e.g.*, into IP packets) for purposes of transmission protocol compliance is outside the scope of the asserted claims.

from a Content Service Provider, which is “an entity, such as a cable service provider, that provides the ‘pipeline’ through which content is delivered from a content owner to a consumer.” Foster, [0039]-[0041], [0078].



223. Fig. 16’s system includes “computer system 400” between source 428 and recipient 430 and which is “any system capable of processing content units of an elementary content stream received from source 428.” Foster, [0078]. Thus, POSAs would have understood that in at least one embodiment computer system 400 would be (or would at least be part of) the “Content Service Provider” that acts

as a “pipeline” to deliver data (*e.g.*, a digital movie) from content owner 428 to recipients 430. Foster, [0039], [0075]-[0078].

224. As shown in Fig. 16, computer system 400 implements Foster’s encryption techniques—i.e., computer system 400 includes a computer comprising a CPU 406 coupled via bus 410 to a memory 402 that stores the computer programs constituting “parsing system 414, encryption system 416, key system 418,” etc. of Foster’s encryption system (Foster, [0074]-[0075], [0078])—demonstrating that Foster’s encryption system can be implemented by the Content Service Provider. Foster, [0039], [0064], [0075]-[0078].

B. Hayhurst (EX1007)

225. I understand that Hayhurst is prior art based on any one of Hayhurst’s (1) publication date of January 22, 2004 (EX1007, code (43)); (2) filing date of July 16, 2003 (EX1007, code (22)); and (3) priority date of July 17, 2002 (EX1007, code (30)), each of which is before the ’116 patent’s earliest-alleged priority date of October 25, 2004 (EX1001, code (60), 1:7-17). I understand that the priority date of July 17, 2002 is based on the filing date of US Provisional Application No. 60/396357 (EX1008, “Hayhurst-Provisional”).

226. I have been asked to provide my opinion as to whether the Hayhurst-Provisional (1) supports at least one of Hayhurst’s claims; and (2) supports the subject matter in Hayhurst relied upon to demonstrate unpatentability of

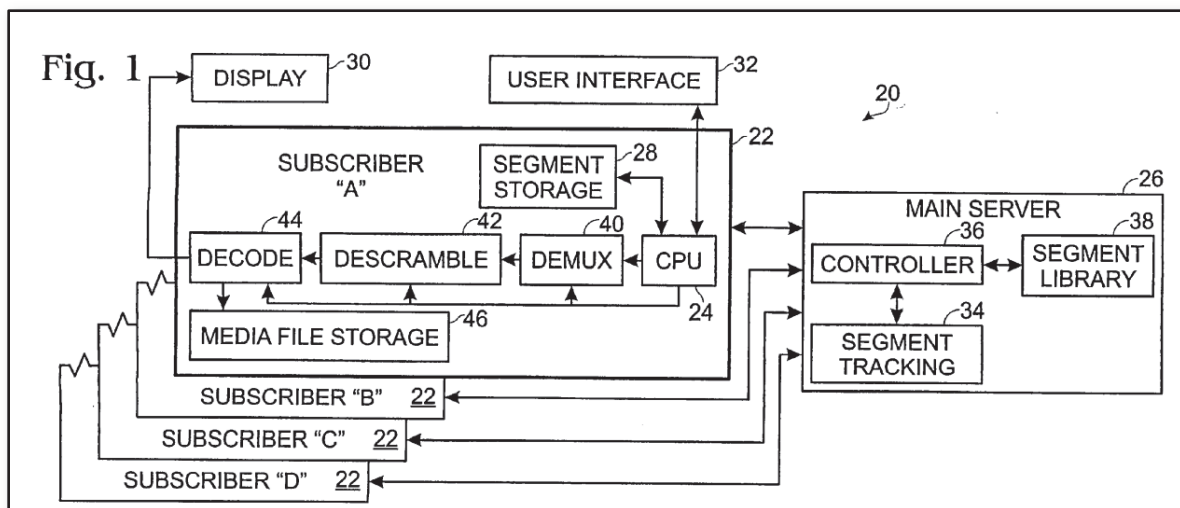
challenged claims, which I understand are requirements for Hayhurst to be entitled to the July 17, 2002 filing date of the Hayhurst-Provisional. In my opinion, the Hayhurst-Provisional satisfies both requirements and, on the basis of those requirements, it is also my opinion that Hayhurst is entitled to the claimed priority date of July 17, 2002, as discussed in further detail in the following sections.

1. Summary of Hayhurst

227. In the following Summary of Hayhurst (EX1007), citations to Hayhurst's disclosure include in parentheses citations to the Hayhurst-Provisional (EX1008) where Hayhurst's disclosure is supported in the Hayhurst-Provisional.

228. Like Foster, Hayhurst concerns "electronic media delivery, such as video-on-demand" and how the data is "secure[ly] handl[ed] and stor[ed]." Hayhurst, [0002] (Hayhurst-Provisional, 6). Hayhurst says then-existing systems for "delivery-on-demand" of media content "require[d] an extensive amount of bandwidth" and, as a result, video on-demand for a large selection of movies "was usually not economically feasible." Hayhurst, [0004] (Hayhurst-Provisional, 6). To address that problem and provide other advantages, Hayhurst's media files are segmented and "distributed among the network of subscriber units," which store the segments so individual subscribers can retrieve an entire media file (e.g., to view a movie on-demand) using segments retrieved from the various storage devices maintained at subscribers. Hayhurst, [0006] (Hayhurst-Provisional, 6).

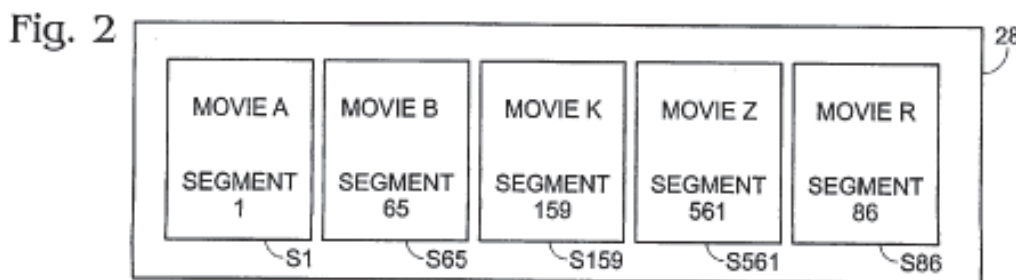
229. Hayhurst's Fig. 1 (copied below) illustrates a system for "decentralized media delivery" ([0012]), comprising subscriber units (subscribers A-D are representative of "thousands of subscribers," [0016]) "in communication with a main server 26," where each unit comprises "a storage device, such as a conventional hard disk" (called "segment storage" devices) for storing "segments of one or media files" ([0017]-[0018]). Hayhurst, [0016]-[0018] (Hayhurst-Provisional, 6-7, FIG. 4).



230. Main server 26 segments a media file (e.g., "digitized data representing a movie" or "any type of media filed that can be stored and displayed," [0018]) and distributes the segments to different subscriber units that each store only one segment of the media file. Hayhurst, [0006], [0008], [0018] (Hayhurst-Provisional, 6-7). Each segment "includes data representing a small portion of the movie" and "a header that includes a packet identifier, descrambling

control, and other information,” which is used to confirm all “packets are collected.” Hayhurst, [0020] (Hayhurst-Provisional, 7).

231. Hayhurst’s Fig. 2 (Hayhurst-Provisional, Fig. 5) reproduced below illustrates the segment storage 28 of a subscriber unit 22 on which segments from different movies (e.g., segment 1 of movie A, segment 65 of movie B, segment 159 of movie K, etc.) are stored; one segment from each of a plurality of movies. Hayhurst, [0008], [0020] (Hayhurst-Provisional, 7).



232. Hayhurst’s “segments are each encrypted” before being distributed “to prevent unauthorized viewing of individual segments.” Hayhurst, [0007] (Hayhurst-Provisional, 6, describing that the requesting subscriber unit requests from the main server “the encryption key to unlock the movie”). Hayhurst’s technique advantageously (for content providers) “shifts the bandwidth requirements away from the content provider and to the network of subscriber units” ([0005]) and “provides a high level of security and a deterrence to piracy” because “[n]o single subscriber has more than a segment of a media file” ([0008]). Hayhurst, [0005]-[0008] (Hayhurst-Provisional, 6-7).

233. Hayhurst further discloses distributing copies of segments to multiple subscriber units so that each segment is available even if “a given subscriber unit is unable to provide the requesting subscriber with the segment stored on that given unit.” Hayhurst, [0009], [0025] (Hayhurst-Provisional, 7). That is, while each subscriber unit stores a single segment of a given movie, each segment of the movie is redundantly stored on multiple subscriber units so that retrieval of the entire movie can be achieved even in the event a given copy of a segment cannot be provided by one of the subscriber units.

234. In response to a request from one of the subscribers to access a media file (e.g., a movie), main server 26 instructs each subscriber unit storing a segment of the requested media file to send it to the requesting subscriber. Hayhurst, [0007], [0028] (Hayhurst-Provisional, 6). After receiving all the media-file segments (e.g., after the requesting subscriber has received the entire movie), the requesting subscriber obtains from main server 26 “the encryption key to unlock the movie.” Hayhurst, [0007], [0028] (Hayhurst-Provisional, 6).

2. Hayhurst Provisional

a. Hayhurst-Provisional Supports Claim 1 of Hayhurst

235. The following table reproduces claim 1 of Hayhurst (EX1007, 8) in the first column with the subject matter disclosed in the Hayhurst-Provisional (EX1008) that meets the corresponding limitation in the second column.

Hayhurst's Claim 1	Hayhurst-Provisional
<i>"A method of delivering a media file that is requested by a requesting subscriber of a network of subscribers, comprising the steps of:"</i>	Title: "Decentralized Media Delivery" Abstract: "A media file is partitioned into small segments, and then distributed in pieces to subscribers' hard drives. A request is sent to the main server for a specific media offering by a subscriber. The main server sends out a message to all subscribers to send any part of the media that is stored on their system directly to the subscriber who is requesting the media."
<i>"segmenting the media file into a plurality of segments,"</i>	Page 7: "A movie in its entirety is converted into a digital format. <i>The movie is split into segments FIG. 2</i>" Figure 2
<i>"distributing the segments of the media file among the network of subscribers for storage on respective units of the subscribers; and"</i>	Page 7: "A server attached through a high-speed connection to the Internet will <i>distribute a segment to hard drives located in the set-top box, or computer or other storage medium of various subscribers a, b, c, d FIG. 4</i>. Each subscriber a, b, c, d will receive one segment of the movie." Figure 4

“responding to a request from the requesting subscriber by directing to that requesting subscriber all of the plurality of segments of the media file.”

Page 7-8: *“When a subscriber, such as subscriber a desires to view a movie, a request is sent either over the network directly to other subscribers or to the main server which then sends a message to the subscribers b, c, d who are storing the segments of the movie on their hard drives. Copies of the movie segments are then sent directly to the requesting subscriber a from the subscribers b, c, d who have stored segments on their hard drives FIG. 6.”*

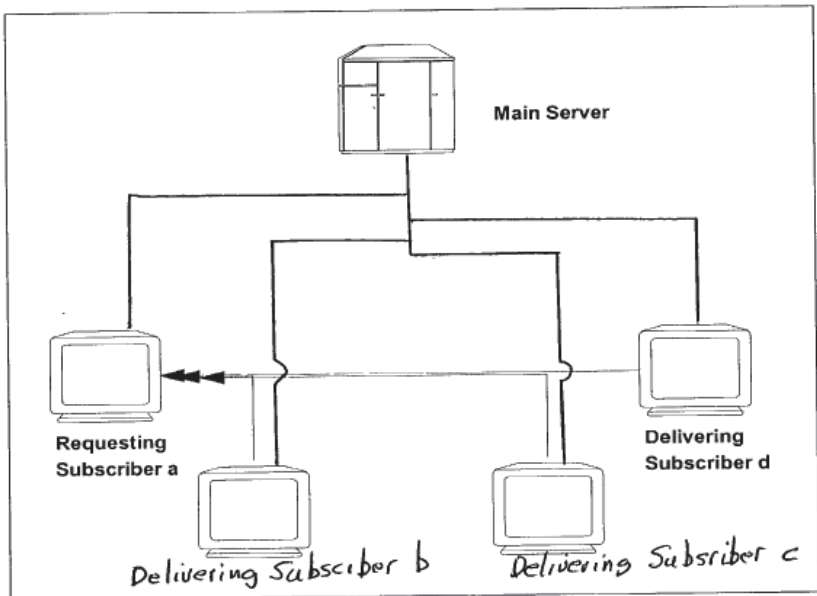


Figure 6

236. Thus, in my opinion, the Hayhurst-Provisional supports claim 1 of Hayhurst.

b. Hayhurst-Provisional Supports Hayhurst’s Relied-Upon Disclosure

237. As the table I provided above (§XI.B.2.a), the Hayhurst-Provisional discloses the same system and method for decentralized media delivery disclosed in Hayhurst, including: (1) segmenting a media file; (2) storing the media-file

segments on “hard drives” or other “storage medium” of different subscribers; (3) and directing all of the media-file segments to a subscriber requesting the entire media file (Hayhurst-Provisional, 6-8)—i.e., the same distributive storage technique disclosed in Hayhurst as also demonstrated by the citations to the Hayhurst-Provisional’s disclosure corresponding to Hayhurst’s disclosure that I provided above in the summary of Hayhurst (see §XI.B.1 above).

238. Citations to the Hayhurst-Provisional supporting Hayhurst’s relied-upon disclosure that I provide in the discussion of the Foster-Hayhurst combination in §XI.C below and in the mapping of Foster-Hayhurst to the challenged claims in §§XI.D-XI.H below support my opinion that the Hayhurst-Provisional supports all the relied-upon disclosure in Hayhurst.

C. Foster-Hayhurst

239. In my opinion, in light of Foster, Hayhurst, and the background knowledge of POSAs, POSAs had reason to store Foster’s encrypted content units on multiple storage devices (e.g., consumer set-top boxes like Hayhurst’s subscriber units) based on Hayhurst’s techniques for storing and distributing encrypted data segments to multiple storage devices.

1. Reasons to Combine

240. In my opinion, POSAs would have recognized several benefits in storing Foster’s individual (encrypted) content units on distributed storage devices

as Hayhurst taught. As discussed in §XI.A above, Foster discloses a system in which a content provider delivers digital content to consumers (e.g., by transmitting content to consumers' set-top boxes) and where Foster's encryption techniques can be implemented in the content provider computer system. Foster, [0039]-[0043], [0064]-[0071], [0075]-[0080]. POSAs would have had numerous reasons to adopt Hayhurst's distributive storage technique in Foster's system so that Foster's encrypted content units for a media file would be distributively stored on different consumer devices (e.g., consumer set-top boxes) as Hayhurst teaches.

241. As I discussed above in §XI.A, Foster's content provider is responsible for delivering content directly to consumer devices as shown, e.g., in Foster's Fig. 16 reproduced. As I discussed in §XI.B.1 above, Hayhurst teaches that content delivery systems where a content service provider delivers content to multiple subscribers can be improved by storing media-file segments (substantively similar to Foster's content units) on multiple subscriber units (substantively similar to Foster's receivers) because Hayhurst's technique: (1) solves problems with on-demand delivery of a large selection of content (e.g., a large movie library) by shifting the bandwidth requirement in delivering digital content away from the content provider (Hayhurst, [0005]; Hayhurst-Provisional, 6); (2) "provides a high level of security and a deterrence to piracy" (Hayhurst, [0008], Hayhurst-Provisional, 6); and (3) supports redundancy in storage

(Hayhurst, [0009], Hayhurst-Provisional, 7). Thus, Hayhurst would have explicitly motivated a POSA to apply Hayhurst's distributed-storage technique to, for example, Foster's embodiments in which the content service provider implements Foster's encryption system (i.e., Foster's system sends encrypted content directly to consumers as shown, e.g., in Foster's Fig. 16) to improve content delivery in multiple ways.

242. Indeed, a POSA would have recognized multiple benefits and advantages in implementing Foster's system with Hayhurst's distributive storage techniques to provide economic, faster and more reliable on-demand delivery of large amounts of digital content such as "delivery-on-demand [of] a large number of movie selections" (Hayhurst, [0004]-[0005] (Hayhurst-Provisional, 6-7)) that would have been beneficial to Foster's consumers. For example, in Foster's embodiments in which the content service provider implements Foster's encryption system (as I discussed in §XI.A above describing Foster's system), a POSA would have been motivated to configure Foster's content service provider to distributively store Foster's segmented content units on multiple devices as taught by Hayhurst to achieve Hayhurst's explicitly taught benefits (enumerated above) and any one or more well-known benefits of distributive-storing techniques like Hayhurst's, including improvements to: (1) speed of accessing content by a subscriber because segments would be retrieved from multiple sources, thus avoiding bottlenecks of a

single source servicing numerous subscribers; (2) efficiency and intelligence (e.g., by being able select storage device(s) that would provide the best delivery, based on availability, location, etc.); (3) security in using geographically distant storage device; and (4) fault tolerance.

243. Specifically, in addition to Hayhurst’s explicit teachings about the benefits in bandwidth and cost, security, and redundancy (also referred to as “fault tolerance”), POSAs were generally aware of those benefits and would have recognized that storing Foster’s content units on multiple storage devices in a distributed fashion would have provided advantages in terms of one or more of the following:

- aggregate speed – for example, by allowing segments to be retrieved from multiple sources as opposed to one storage device the speed of retrieving the stored content would have, in general, been improved. My opinion in this regard is corroborated by EX1028 at [0022]-[0023], which discloses a system similar to Hayhurst that provides a “large advantage” over storing media files in one storage device (EX1028, [0022]) because “the bandwidth needed to download content ... is vastly reduced” by distributing portions to multiple storage devices. Because a single source does not have to provide all content to numerous subscribers, bottlenecks at the single source are avoided.

- efficiency / “intelligence” – for example, by being able select storage device(s) that would provide the best delivery, based on availability, location, etc., would have improved the reliability and speed of retrieval. My opinion in this regard is corroborated by EX1031, which discloses at [0012] that by splitting a file “into multiple pieces across the N servers, e.g., four servers” the file “may be transferred from all of them in parallel,” thereby multiplying the potential total transfer rate. *See also* EX1031, [0003], [0013], [0020].
- security – for example, by using geographically separated storage devices it would make it more challenging for an unauthorized user to retrieve and/or intentionally corrupt all of the segments associated with the same data set. My opinion in this regard is corroborated by EX1030, which discloses at [0005]-[0010] that by “storing data to be backed-up...on different back-up servers which are spatially remote from another” it “minimis[es] the risk of unauthorised access by third parties to the backed-up data.”
- fault tolerance – for example, by storing duplicates of segments on multiple storage device the system would have been more tolerant of issues with individual storage devices (e.g., storage devices that were unavailable, suddenly left the network, had poor network conditions, etc.). My opinion in this regard is corroborated by EX1030, which discloses at [0005]-[0010]

that by “storing data to be backed-up...on different back-up servers which are spatially remote from one another” “data loss becomes very unlikely” and that additional protection against data loss occurs by distributing encrypted segmented data to multiple back-up servers in such a way that less than all segments are required to recover the original data but more than one segment is required, which gives the benefit that “[t]he breakdown of one back-up server is [] non-problematic” but because “it is not possible to recover the original data on the basis of only one data stream.” This is further corroborated by EX1029, which teaches at 1:19-22, 2:31-38, 11:11-26, and 16:6-20 that “when storage is distributed among n storage device, fault-tolerance maintains data reliability in the event of failure of some (but no more than k) of the n storage devices,” where failure includes “storage device is down, network delay is high...the file to be stored (backed up) is corrupted or destroyed,” etc. EX1029 at 15:39-16:20 similarly teaches that “[r]edundancy can, in fact, be added by replication or encoding,” where encoding is done so that data can be restored from fewer than all data blocks. See also EX1031, which discloses at [0003], [0012]-[0013], [0015], [0018]-[0024] that “downloading data from multiple sources or channel in parallel” helps “provide[] reliable downloads even when the sources have poor connectivity, and is thus ideal for distributing data over unreliable networks”

and provides advantages in terms of “speed,” “intelligence,” “security,” “fault tolerance,” among other advantages.

244. Thus, a POSA would have recognized that Hayhurst’s distributed storage technique provides improvements to content delivery that would have benefited Foster’s content delivery system and thus had multiple independent reasons discussed above to configure Foster’s system to distributively store encrypted content units on different storage devices (e.g., on consumer devices such as a set-top box). Hayhurst’s distributed storage technique enables economical delivery of video-on-demand of large selections of content (see e.g., Hayhurst, [0004]-[0010] that a POSA would have recognized would likewise benefit Foster’s system in enabling Foster’s content service provider to be able to deliver economical and efficient on-demand delivery of large selections of content.

245. The Foster-Hayhurst combination that I discussed above in this section would have also been to POSAs an obvious combination of prior-art elements—i.e., Foster’s encryption system to generate encrypted data segments, and Hayhurst’s method for storing encrypted data segments on multiple distributed storage devices—according to known encryption, storage and associated computer science methods, to yield the predictable result of Foster’s encrypted content units being stored on and retrievable from distributed storage devices as taught in Hayhurst.

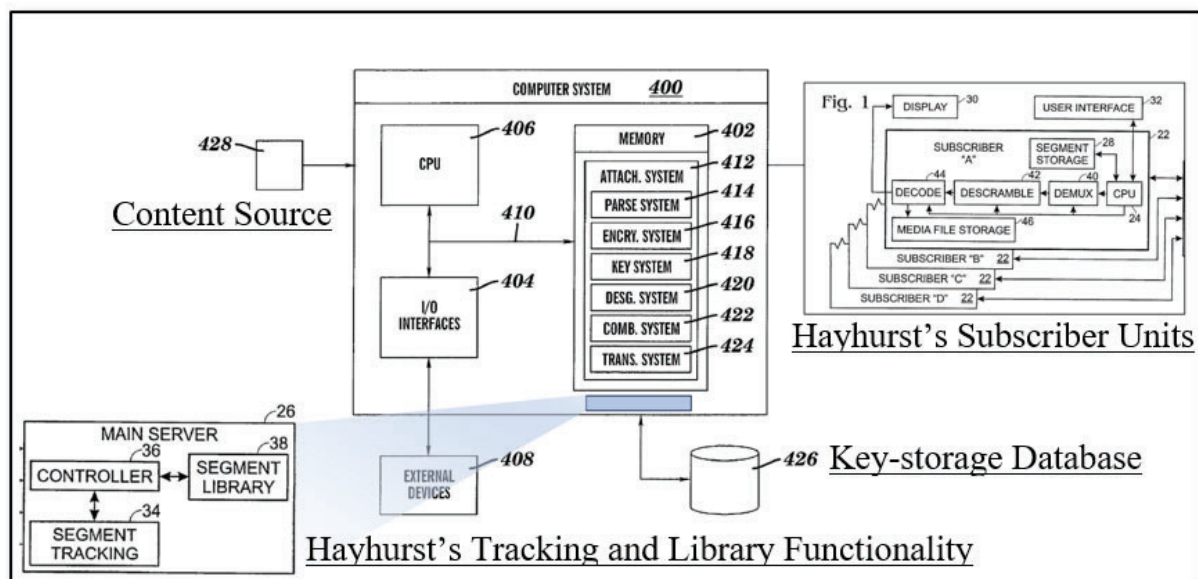
246. The Foster-Hayhurst combination that I discussed above in this section would have also been nothing more than the obvious application of Hayhurst's known distributed storage technique to Foster's system, which was ready for the improvement of storing segmented digital content on distributed storage devices to address bandwidth issues in providing content to subscriber, increase security, increase efficiency, and/or add redundancy in the manner those improvements are achieved in Hayhurst.

2. The Foster-Hayhurst Combination

247. The Foster-Hayhurst combination that I discussed above would have used at least (1) Foster's technique of securing digital content by encrypting parceled segments of the content with respective different encryption keys that are themselves encrypted with a KEK (Foster, [0064]-[0071]); (2) Hayhurst's technique of storing encrypted-media segments in a distributed manner across a network of subscriber devices (Hayhurst, [0006]-[0008], [0016]-[0023]; Hayhurst-Provisional, 6-7); and (3) Hayhurst's technique of redundantly storing media segments on the subscriber units (Hayhurst, [0025]-[0028]); Hayhurst-Provisional, 7).

248. I have illustrated below one exemplary implementation of Foster-Hayhurst that is based on Foster's Fig. 16 and Hayhurst's Fig. 1, in which Foster's "recipients 430" are implemented as Hayhurst's "subscriber units" and Foster's

content service provider implemented by “computer system 400” is augmented to include functionality of Hayhurst’s main server 26 so it can control distribution of, and access to, Foster’s encrypted content portions stored on the subscriber units. Hayhurst, [0017], [0023]-[0024], [0026], FIGS. 1, 4 (Hayhurst-Provisional, 6-8, FIGS. 4, 6); Foster, [0040]-[0041], [0075]-[0080], FIG. 16.



Foster-Hayhurst (exemplary implementation)

249. In particular, in Foster-Hayhurst’s content provider (Foster’s content provider computer system 400 implementing the Hayhurst main server’s distributive and redundant storage functionality), parse system 414 and encryption system 416 would have performed Foster’s Fig. 8 (annotated below) technique of: (1) parceling digital content into individual content units 104, (2) encrypting each content packet (“payload”) 106 with a different title key 108, (3) encrypting each title key 108 with key-encrypting key 110, and (4) attaching the resulting

encrypted key 112 (with or without header extension 118) to encrypted-content packet 114 with header 104 to generate encrypted content unit 116. Foster, [0064]-[0071], [0074]-[0079].

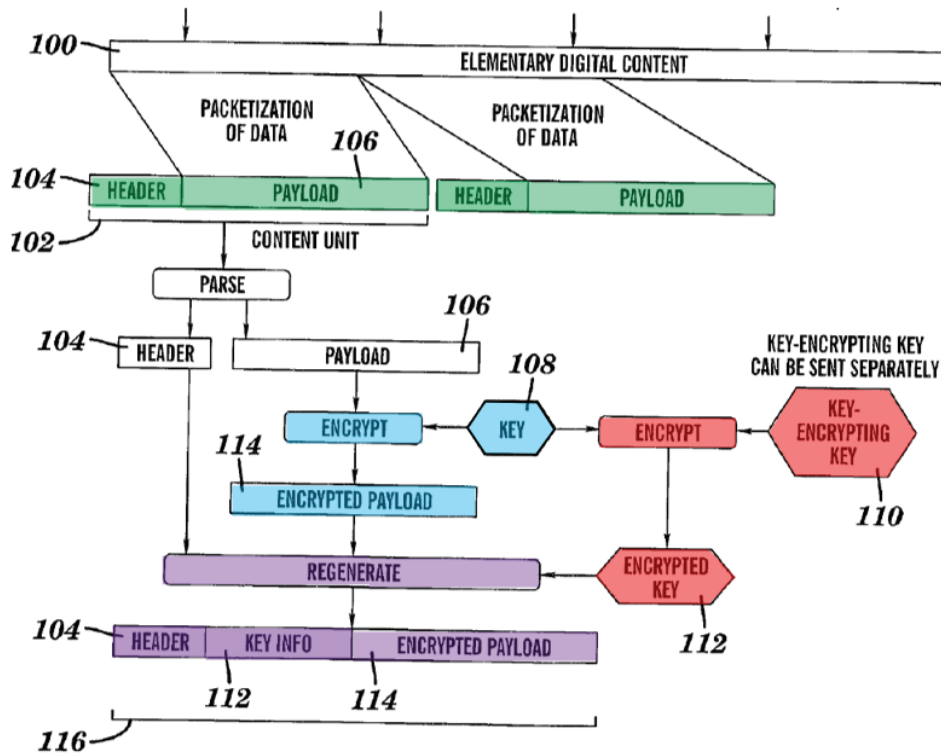


FIG. 8

250. The Foster-Hayhurst content provider would have then distributed the processed content units 116 to different respective Foster-Hayhurst subscriber units (Foster's recipients 430 adapted to perform the functionality of sending stored content segments to a requesting subscriber unit of Hayhurst's subscriber units), including making one or more copies of each processed content unit 116 so that each processed content unit 116 is redundantly stored—i.e., each Foster-Hayhurst

subscriber would have stored one content unit 116 of the digital content 100, but multiple subscribers would have stored each content unit.

251. I note that Foster at [0055] discloses the KEK is “typically delivered to a recipient concurrently with or prior to the delivery of encrypted content 26.” Hayhurst discloses that the “encryption key to unlock the movie” stored on the main server until a subscriber’s unit sends a “request” to “unlock the movie,” which the “main server provides...and billing information is then recorded in the main server.” Hayhurst, [0007]. POSAs understood that Foster’s disclosure assumes that recipients are already authorized to decrypt the encrypted content they receive (which is why they receive the KEK before or along with the encrypted content), whereas Hayhurst’s disclosure assumes that recipients are not yet authorized to decrypt the content they receiver (which is why they have to make a request and payment to the main server to receive the encryption key). Both Foster and Hayhurst thus provide the KEK only when a recipient is deemed authorized. In the Foster-Hayhurst combination, the KEK would be maintained in a storage device (e.g., Foster’s database 426) until requested by a subscriber, consistent with Hayhurst’s disclosure that the subscriber units are storing portions of a data set before the subscriber has been authorized to decrypt the entire data set and consistent with Foster’s providing the KEK to authorized recipients.

3. Reasonable Expectation of Success

252. POSAs would have reasonably expected in the Foster-Hayhurst combination that I described above because Foster and Hayhurst references both teach segmenting content for content providers and encrypting each segment. POSAs also understood that Foster’s content units are individually (randomly) accessible and agnostic to where and how they are stored compared to other content units and thus POSAs understood Foster’s content units were compatible with being stored on multiple, separate storage devices (like Hayhurst’s subscriber storage devices). For example, Foster (EX1006) discloses that a “recipient” that stores encrypted content includes “a **receiver**, capable of receiving transmissions” ([0041]) and discloses that a “receiver” includes “a consumer device, such as a set-top box, a DVD player, etc.” ([0040]). *See also* Foster, [0061] (“receipt by a recipient (e.g., content service provider or **consumer**)”).

253. Thus, POSAs would have expected success in the Foster-Hayhurst combination because having Foster’s encrypted content units sent to and stored by multiple consumer storage devices, as taught by Hayhurst, would have been similar to Foster’s disclosure of storing its encrypted content units in a “receiver” – e.g., a “consumer device...that receives content ... **from a content owner or from another receiver within a consumer home network.**” Foster, [0040]; Hayhurst (EX1007), [0023]-[0029], [0031], claim 1 (method for “delivering a media file”),

claim 12 (claim 1's method for delivering further includes "encrypting each of the segments") (Hayhurst-Provisional, 6-8). In other words, Foster's content units are generally agnostic to where and how they were stored and Hayhurst's distributed storage devices are generally agnostic to how portions of content and/or encryption keys were encrypted. Thus, POSAs would have also reasonably expected success in using ordinary computer programming skill to distribute Foster's encrypted content units to multiple storage devices based on Hayhurst's teachings by incorporating into Foster's system functionality to distribute data to multiple storage devices based on Hayhurst's teachings.

D. Claim 1

1. 1[PRE] A method for securing a data set, the method comprising:

254. In my opinion, Foster-Hayhurst (§XI.C.2 above) performs "*a method for securing a data set*" by encrypting digital content (i.e., a "*data set*" such as "video data," Foster, [0064]) using Foster's encryption technique that "provides enhanced security" by using multiple encryption keys (Foster, [0004], [0071]) and provides a "high level of security and a deterrence to piracy" (Hayhurst, [0008], Hayhurst-Provisional, 6) by applying Hayhurst's distributed storage technique to Foster's "encrypted content." Foster, [0064]-[0068], [0071]; Hayhurst, [0006]-[0008], [0020] (Hayhurst-Provisional, 6-7). Foster's digital content is a "data set" under both parties' constructions because POSAs understood that, e.g., a video file

is a “collection of related information made up of separate elements [e.g., video frames] that can be treated as a unit [e.g., a file] in data handling” (Petitioner: EX1042, 9-10) and “a collection of information for storage” (Patent Owner: EX1043, 6-7).

2. [1A] distributing the data set into a plurality of data chunks, wherein none of the data chunks are, by themselves, sufficient to reconstruct the data set

255. As I discussed above in §XI.C.2, Foster-Hayhurst performs Foster’s encryption method. As illustrated in Foster’s Fig. 8 that I copied and annotated below, Foster distributes digital content 100 into a plurality of “payloads” by “parcel[ing]” digital content 100 (i.e., the claimed “*data set*” as I discussed in §XI.D.1 above in connection with 1[PRE] is distributed). Foster, [0064], FIG. 8. Specifically, digital content 110 (blue) is “parceled into content units 102” that each includes a “content packet” or “payload” 106 that is a portion of the digital content 100 and a header 104 that includes information about the payload (e.g., a timestamp) that can be used to reconstruct the digital content 100 from the content units 102 into which it is parceled. *Id.*

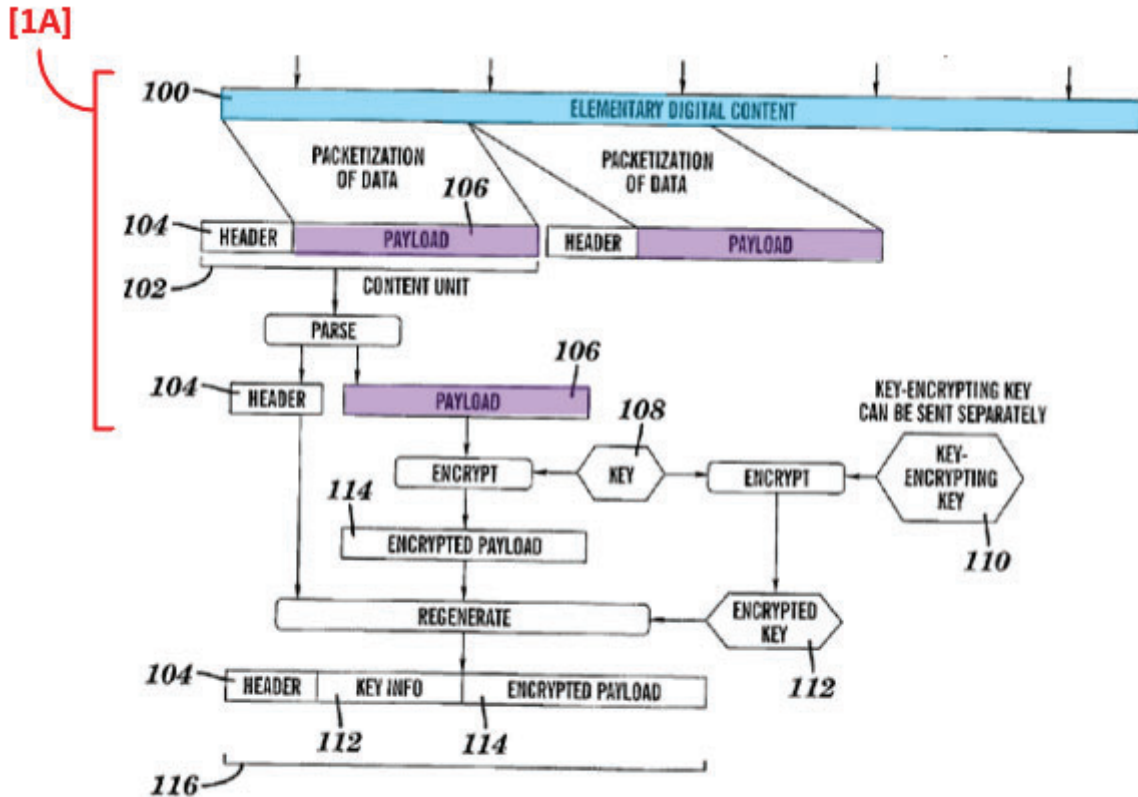


FIG. 8

256. Each content unit 102 is then parsed to separate the header 104 and content packet 106 (illustrated in Fig. 8 as payload 106 highlighted in **purple**). These “content packets” or “payloads” 106 are “a plurality of data chunks” as claimed because each includes only a portion of digital content 100 (i.e., a subset of the full data set), which is all that is required of a data chunk as I discussed in §§VII.B.1-VII.B.2 above. Foster-Hayhurst thus meets both parties’ proposed constructions of “*distributing the data set...*” because Foster is “splitting” (Google) and “dividing and placing” (SFI) “*the data set into a plurality of chunks*” (EX1042,

2-5; EX1043, 2) that are each distributed to different subscribers after being encrypted (§XI.C.2).

257. Thus, in my opinion, Foster-Hayhurst’s use of Foster’s steps bracketed in **red** in annotated Fig. 8 above thus performs “*distributing the data set into a plurality of data chunks*,” as claimed because digital content 100 is distributed into a plurality of content packets (“payloads”) and none of the payloads 106 “*are, by themselves, sufficient to reconstruct the data set*,” as recited in [1B], because each payload is only a subset of the data set and not the entirety of digital content 100.

3. [1B] encrypting each of the data chunks with a respective one of a plurality of encryption keys

258. As I discussed in §XI.C.2 above, Foster-Hayhurst encrypts each payload 106 with a different title key 108, as taught by Foster ([0064]-[0065]). As illustrated in steps boxed-**red** in Fig. 8 below, Foster discloses that after parceling, “content packet 106 is *encrypted* with title key 108 to yield encrypted content packet 114.” Foster, [0064]. “[E]ach content packet can be encrypted with a *different* title key.” Foster, [0071], [0065].

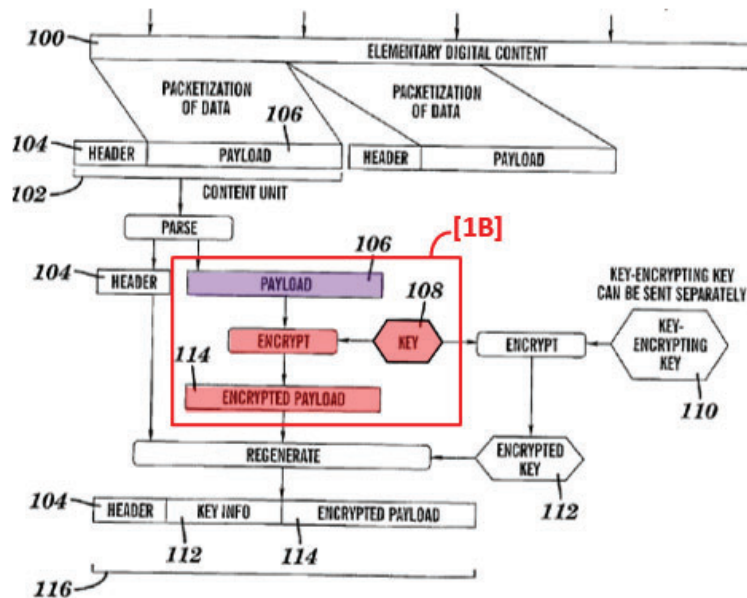


FIG. 8

259. Thus, in my opinion, Foster-Hayhurst performs “*encrypting each of the data chunks with a respective one of a plurality of encryption keys,*” as recited in [1B] because, as I discussed above in §XI.C.2, Foster-Hayhurst uses Foster’s technique of encrypting each content packet (payload) with a different title key. Foster, [0064]-[0068], [0071].

4. [1C] obfuscating each of the plurality of different encryption keys

260. In my opinion, Foster-Hayhurst (§XI.C.2) meets [1C] under Patent Owner’s interpretation that encrypting keys obfuscates them because Foster-Hayhurst uses Foster’s technique of encrypting each title key 108 with a key encrypting key (KEK) 110 to yield encrypted key 112, as highlighted in **red** in Foster’s Fig. 8 below. Foster, [0064]-[0065].

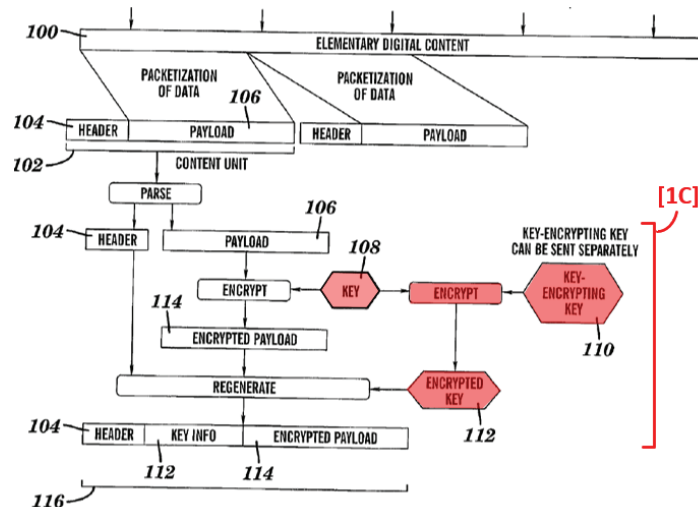
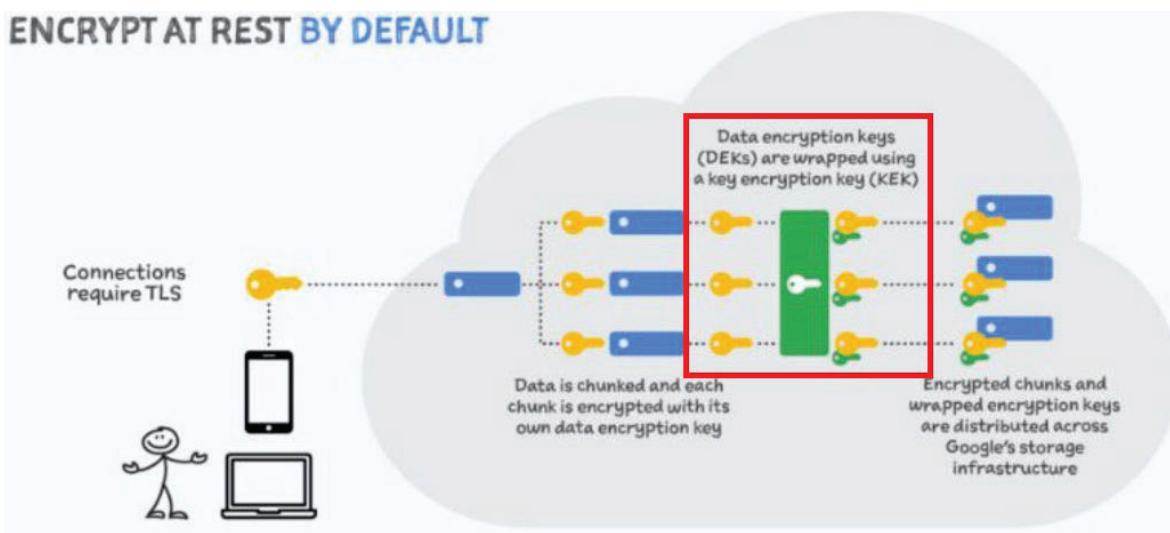


FIG. 8

261. As I discussed above in §VIII.F.2 above in connection with [1C]’s obfuscating step in Ground 2, Patent Owner has taken the position that “obfuscating each of the plurality of different encryption keys” reads on encrypting data encryption keys (DEKs) with a key encryption key (KEK), alleging that the below “exemplary annotated image show[s] Google’s obfuscation of the DEKs by the KEKs.” EX1012, 4-5; EX1011, ¶117.

ENCRYPT AT REST BY DEFAULT



262. Thus, in my opinion, Foster-Hayhurst meets [1C] under the interpretation of “obfuscating” advanced by Patent Owner in litigation—i.e., Foster-Hayhurst (§XI.C.2) obfuscates by encrypting each of the different title keys 108 with KEK 110 (Foster, [0064]-[0071]), thereby also rendering the keys “unclear or obscure” as Patent Owner construes an “obfuscated” encryption key (EX1043, 5).

5. [1D] separately storing each data chunk of the plurality of data chunks together with one of the plurality of obfuscated different encryption keys on a plurality of different storage devices

263. In my opinion, Foster-Hayhurst (§XI.C.2 above) meets [1D] because it applies Hayhurst’s distributive storing techniques to the encrypted content units 116 of Foster’s digital content. Hayhurst, [0006]-[0007], [0016]-[0020], [0025] (Hayhurst-Provisional, 6-8); Foster, [0043], [0064]-[0071].

a. Foster-Hayhurst meets [1D] the same way the ’116 patent’s embodiments do

264. As illustrated in Foster’s Fig. 8 (which I annotated below), after encrypting each payload with a different title key 108 and then encrypting the title key with a KEK 110, Foster “regenerates” the parcel by attaching the encrypted key 112 (highlighted **red**) and the header (highlighted **green**) to the corresponding encrypted content package (highlighted **purple**) to produce the “processed content

unit 116” (boxed **red**) that Foster discloses is stored by the content provider or transmitted to a receiver (e.g., set-top box). Foster, [0039]-[0040], [0064]-[0068].

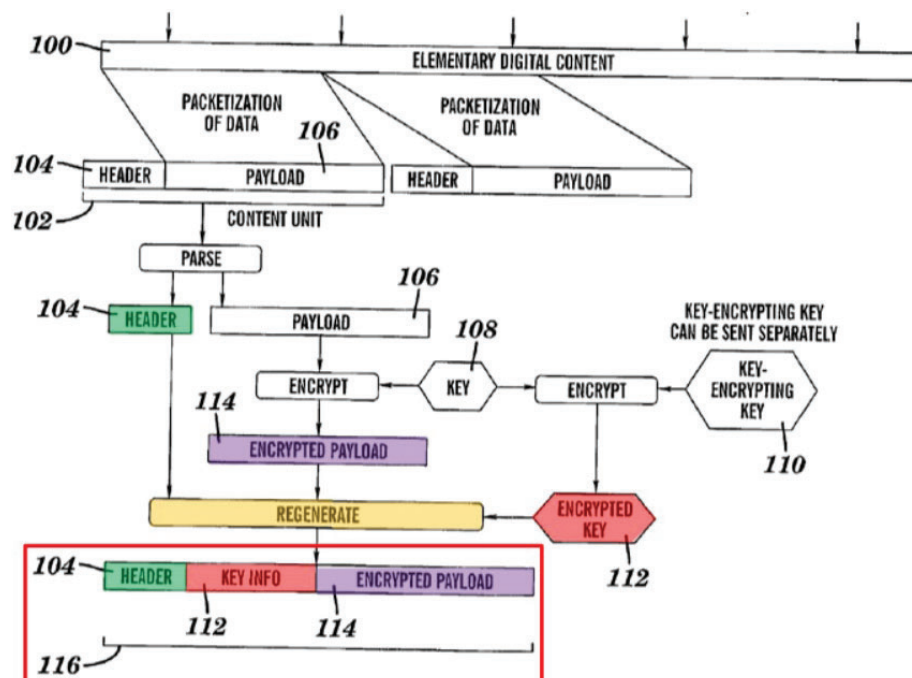


FIG. 8

265. In Foster-Hayhurst (§XI.C.2), each encrypted content unit (i.e., “processed content unit 116”) is stored on a different Foster-Hayhurst subscriber unit, as taught by Hayhurst, which “can be adapted from a conventional set-top box,” [0016]). Hayhurst, [0006]-[0007], [0016]-[0020], (Hayhurst-Provisional, 6-7). In my opinion, by storing each processed content unit 116, which includes the respective different encrypted key 112 (i.e., the respective “*obfuscated different encryption key*”) attached to its corresponding encrypted content packet 114 (i.e., a “*data chunk*”) on a different subscriber unit, Foster-Hayhurst meets [1D] because [1D]’s “*separately storing*” does not require redundant storing chunk/key pairs as I

discussed in §§VII.B.5.a-VII.B.5.b above where I also note that neither parties construe [1D] to require redundant storage (EX1042, 7; EX1043, 5).

b. Foster-Hayhurst meets [1D] if construed to require redundantly storing each data chunk/key pair on multiple different storage devices

266. As I discussed in §XI.C.2 above, Foster-Hayhurst uses Hayhurst’s redundant storage of encrypted segments on different subscriber units. That is, each of Foster’s encrypted content units is “copied and distributed to a plurality of subscribers to provide for redundant availability” in case a subscriber unit “becomes inaccessible.” Hayhurst, [0025] (Hayhurst-Provisional, 7). In my opinion, by applying Hayhurst’s redundant data storage to Foster’s encrypted content units, Foster-Hayhurst meets [1D] if construed to require redundantly storing each processed content unit 116.

E. Claim 2: The method of claim 1, wherein obfuscating each of the plurality of different encryption keys comprises performing a transform on data indicative of at least one of the plurality of different encryption keys

267. In my opinion, by encrypting each encryption key with a KEK as taught by Foster (Foster, [0064]-[0065]), Foster-Hayhurst (discussed in §XI.C.2 above) meets claim 2 under (1) Patent Owner’s infringement contentions that data indicative of a key may be the key itself and that encrypting an encryption key transforms it (EX1012, 8: alleging that “Google performs a transform on each of the different encryption keys when it encrypts the DEKs using the KEKs”), and (2)

Patent Owner constructions that a “changing the form” of a key is a “*transform*” (EX1042, 5-6) and that “*data indicative of*” a key “makes known” or “shows” the key (EX1043, 3-4), as I discussed in §VIII.G above. That is, in my opinion, Foster-Hayhurst meets claim 2 under Patent Owner’s interpretation because Foster-Hayhurst encrypts Foster’s encryption keys 108 with Foster’s KEK 110 to produce encrypted key 112. Foster, [0064]-[0067].

F. Claim 5: The method of claim 1, wherein storing the plurality of data chunks together with the plurality of obfuscated different encryption keys comprises storing each of the plurality of data chunks together with a respective one of the plurality of obfuscated different encryption keys

268. As I discussed in §XI.C.2, Foster-Hayhurst attaches each encrypted key 112 (each “*obfuscated different encryption key*” to its corresponding encrypted content packet 114 (i.e., “*the plurality of data chunks*”) as taught by Foster (e.g., Foster, [0064]-[0071]) and stores each resulting processed content unit 116 on a different subscriber unit as taught by Hayhurst (e.g., Hayhurst, [0006]-[0007], [0020], Hayhurst-Provisional, 6-7). Therefore, in my opinion, Foster-Hayhurst meets claim 5.

G. Claim 9

1. 9[PRE]-[9B]

269. 9[PRE]-[9B] are identical to 1[PRE]-[1B] and, in my opinion, are therefore met by Foster-Hayhurst for the same reasons I discussed in §§XI.D.1-XI.D.3 above.

2. [9C] transforming each of the plurality of different encryption keys

270. In my opinion, Foster-Hayhurst meets [9C] in two ways. First, Foster-Hayhurst (§XI.C.2) encrypts each different encryption key 108 with KEK 110, which meets [9C] under the interpretation advanced by Patent Owner that encrypting an encryption key transforms it by changing its form. EX1012, 12 (alleging [9C] is met by “encrypt[ing] the DEKs using the KEKs”); EX1043, 5-6 (“changing the form” of a key is “*transforming*”). *See also* §VIII.H.2 (discussing [9C] in connection with Tajima-Foster’s encrypting of encryption keys).

271. Second, Foster-Hayhurst (§XI.C.2) transforms the encryption keys by using Foster’s identifier info 120 to indicate that the key information has been masked by modifying “part or all of the key information to distinguish it from a known code that may have a specialized purpose.” Foster, [0066]. As illustrated in Foster’s Fig. 9 annotated below, the header extension 118 of content unit 116 includes information 120 (shaded **red**) that may be used to indicate that the key information has been modified (e.g., inverted) to avoid aliasing so that the recipient knows to reverse the modification before decrypting. Foster, [0066].

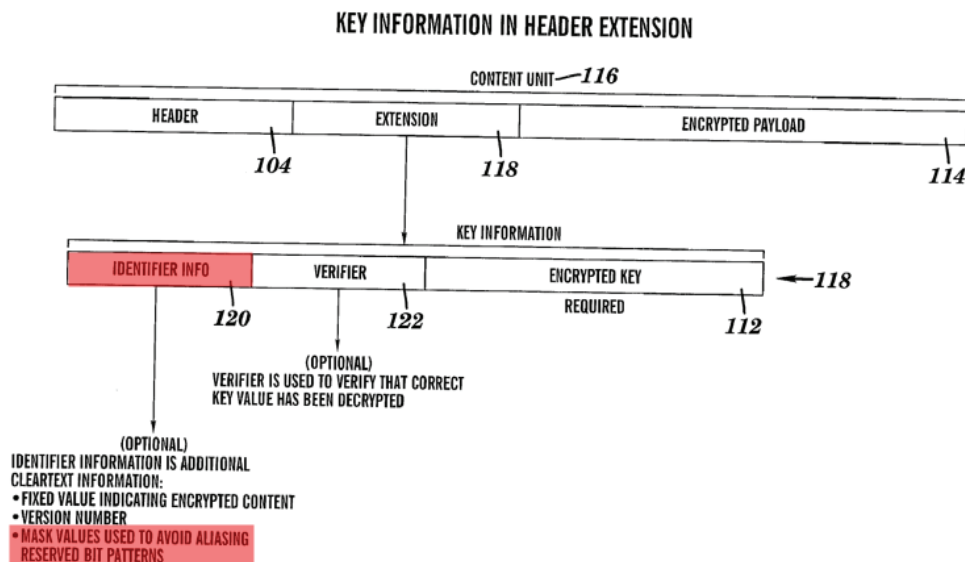


FIG. 9

272. Inverting the bits of the key information transforms Foster’s encryption key stored therein by “changing the form” of the encryption under Patent Owner’s construction (EX1043, 5-6) and by “changing without the use of a secret key the bits of the key” under Google’s construction (EX1042, 8). Thus, in my opinion, Foster-Hayhurst transforms encryption keys (1) under Patent Owner’s interpretation by encrypting Foster’s encryption keys; and/or (2) by inverting the key information that includes Foster’s encrypted keys under both parties’ constructions.

3. **[9D] separately storing each data chunk of the plurality of data chunks together with one of the plurality of transformed different encryption keys on a plurality of different storage devices.**

273. In my opinion, Foster-Hayhurst (§XI.C.2) meets [9D] for the same reasons Foster-Hayhurst meets [1D] that I discussed in §XI.D.5 above. While [9D]

differs from [1D] in that the stored encryption keys are “*transformed*” encryption keys rather than “*obfuscated*,” Foster’s encrypted key 112 is both a “*transformed*” and “*obfuscated*” encryption key under Patent Owner’s interpretation that encrypting keys both transforms and obfuscates them as I discussed in §§XI.D.4 and XI.G.2 above (citing EX1012, 4-5, 12) and is a “*transformed*” encryption key by inverting the key information as I also discussed in §XI.G.2 above.

274. In my opinion, Foster-Hayhurst’s use of Hayhurst’s redundant data storage technique (see §XI.C.2 above) meets [9D] even if construed to require redundantly storing each processed content unit 116 as discussed above in connection with claim 1’s limitation [1D] in §XI.D.5.b above.

H. Claim 11: The method of claim 9, wherein storing the plurality of data chunks with the plurality of transformed different encryption keys comprises storing each of the plurality of data chunks with a respective one of the plurality of transformed different encryption keys

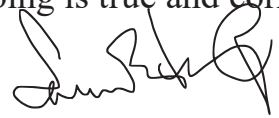
275. As I discussed in §XI.C.2, Foster-Hayhurst attaches each encrypted key 112 (i.e., each “*transformed different encryption key*”) to its corresponding encrypted content packet 114 (i.e., “*the plurality of data chunks*”) and stores each resulting processed content unit 116 on a different subscriber unit. *See, e.g.*, Hayhurst, [0008] (Hayhurst-Provisional, 6). Therefore, in my opinion, Foster-Hayhurst meets claim 11.

* * *

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 22nd, 2023

A handwritten signature in black ink, appearing to read 'Samrat Bhattacharjee', with a stylized, cursive script.

Samrat Bhattacharjee

XII. 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]. The method of claim 9, further comprising generating the data by encrypting primary data.
[1A] distributing the data set into a plurality of data chunks, wherein none of the data chunks are, by themselves, sufficient to reconstruct the data set;
[1B] encrypting each of the data chunks with a respective one of a plurality of encryption keys;
[1C] obfuscating each of the plurality of different encryption keys; and
[1D] separately storing each data chunk of the plurality of data chunks together with one of the plurality of obfuscated different encryption keys on a plurality of different storage devices.
Claim 2
2. The method of claim 1, wherein obfuscating each of the plurality of different encryption keys comprises performing a transform on data indicative of at least one of the plurality of different encryption keys.
Claim 3
3. The method of claim 2, wherein performing a transformation on data indicative of at least one of the plurality of different encryption keys comprises performing an all-or-nothing transform on the data indicative of at least one of the plurality of different encryption keys.
Claim 5
5. The method of claim 1, wherein storing the plurality of data chunks together with the plurality of obfuscated different encryption keys comprises storing each of the plurality of data chunks together with a respective one of the plurality of obfuscated different encryption keys.
Claim 6
6. The method of claim 1, wherein storing the plurality of data chunks together with the plurality of obfuscated different encryption keys comprises including, with a data chunk of the plurality of data chunks, data indicative of an encryption key that was used to encrypt a different data chunk of the plurality of data chunks.
Claim 7

7. The method of claim 1, wherein distributing the data set into a plurality of data chunks comprises using a random technique or pseudorandom technique.
Claim 8
8. The method of claim 1, further comprising generating the data set by encrypting primary data.
Claim 9
9[PRE]. A method for securing a data set, the method comprising:
[9A] distributing the data set into a plurality of data chunks, wherein none of the data chunks are, by themselves, sufficient to reconstruct the data set;
[9B] separately storing each data chunk of the plurality of data chunks together with one of the plurality of obfuscated different encryption keys on a plurality of different storage devices.
[9C] transforming each of the plurality of different encryption keys; and
[9D] separately storing each data chunk of the plurality of data chunks together with one of the plurality of transformed different encryption keys on a plurality of different storage devices.
Claim 10
10. The method of claim 9, wherein transforming each of the plurality of different encryption keys comprises performing an all-or-nothing transform on data indicative of at least one of the plurality of different encryption keys.
Claim 11
11. The method of claim 9, wherein storing the plurality of data chunks with the plurality of transformed different encryption keys comprises storing each of the plurality of data chunks with a respective one of the plurality of transformed different encryption keys.
Claim 12
12. The method of claim 9, wherein storing the plurality of data chunks with the plurality of transformed different encryption keys comprises including, with a data chunk of the plurality of data chunks, data indicative of an encryption key of the plurality of encryption keys that was used to encrypt a different data chunk of the plurality of data chunks.
Claim 13
13. The method of claim 9, wherein distributing the data set into a plurality of data chunks comprises using a random technique or pseudorandom technique.
Claim 14
14. The method of claim 9, further comprising generating the data by encrypting primary data.