United States Patent and Trademark Office

Before the Patent Trial and Appeal Board

Google LLC

(Petitioner)

v.

Security First Innovations, LLC

(Patent Owner)

IPR2024-00212 | **U.S. Patent No. 11,178,116**

IPR2024-00214 | **U.S. Patent No. 11,068,609**

IPR2024-00215 | **U.S. Patent No. 10,452,854**

Petitioner's Demonstrative Exhibits

February 27, 2025

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Google Exhibit 1118
Google v. SFI
IPR2024-00212

Summary of All Grounds

’116 Patent (IPR2024-00212)

Ground Number and Reference(s)		Claims	Basis
1	Tajima (EX1005)	1, 5-8	§§102/103
2	Tajima, Foster (EX1006)	1-2, 9, 11-14	§103
3	Tajima, Rivest (EX1034)	9-14	§103
4	Tajima, Rivest, Shyu (EX1016)	1-3, 6	§103
5	Foster, Hayhurst (EX1007)	1-2, 5, 9, 11	§103

’116 Paper 2 at 1

’854 Patent (IPR2024-00215

Ground Number and Reference(s)		Claims	Basis
1	Orsini (EX1005)	1-10	§§102/103
2	Foster (EX1006)	1-10	§103

’854 Paper 2 at 3

’609 Patent (IPR2024-00214)

Ground Number and Reference		Claims	Basis
1	Orsini (EX1005)	1-10	§§102/103
2	Foster (EX1006)	1-3, 7-8 4-6, 9-10	§§102/103 §103

’609 Paper 2 at 3

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Agenda

1. **Tajima-based Grounds: ’116 Grounds 1-2** (independent claim 1 + claims 5-8)

2. **Foster-Hayhurst Grounds: ’116 Ground 5** (independent claims 1 & 9 + claims 2, 5, 11)
   ’854 Ground 2 (all challenged claims 1-10)

3. **Foster Ground: ’609 Ground 2** (all challenged claims 1-10)

4. **Orsini Grounds: ’854 Ground 1** (all challenged claims 1-10)
   ’609 Ground 1 (all challenged claims 1-10)

5. **Tajima-Rivest-based Grounds: ’116 Ground 3** (independent claim 9 + claims 10-14)
   ’116 Ground 4 (independent claim 1 + claims 2-3, 6)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Agenda

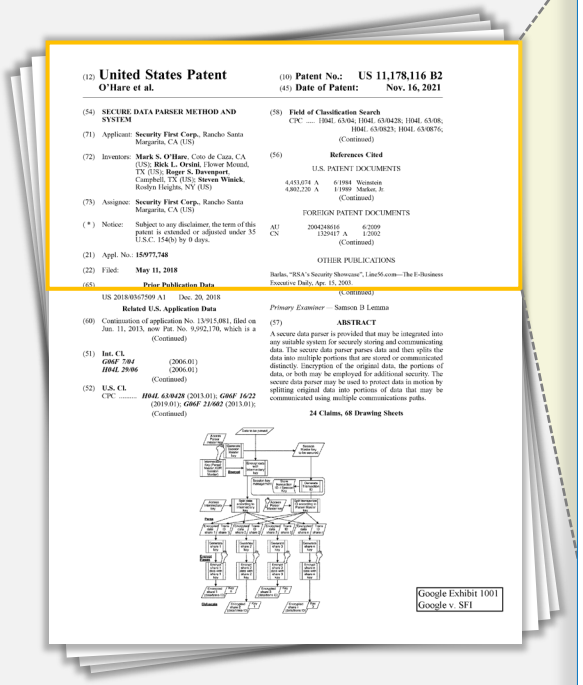
1. **Tajima-based Grounds: ’116 Grounds 1-2** (independent claim 1 + claims 5-8)
2. Foster-Hayhurst Grounds: ’116 Ground 5 (independent claims 1 & 9 + claims 2, 5, 11)
   ’854 Ground 2 (all challenged claims 1-10)
3. Foster Ground: ’609 Ground 2 (all challenged claims 1-10)
4. Orsini Grounds: ’854 Ground 1 (all challenged claims 1-10)
   ’609 Ground 1 (all challenged claims 1-10)
5. Tajima-Rivest-based Grounds: ’116 Ground 3 (independent claim 9 + claims 10-14)
   ’116 Ground 4 (independent claim 1 + claims 2-3, 6)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Patent

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Challenged U.S. Patent No. 11,178,116



(12) **United States Patent**
O'Hare et al.

(10) **Patent No.: US 11,178,116 B2**
(45) **Date of Patent: Nov. 16, 2021**

(54) **SECURE DATA PARSER METHOD AND SYSTEM**

(71) Applicant: **Security First Corp.**, Rancho Santa Margarita, CA (US)

(72) Inventors: **Mark S. O'Hare**, Coto de Caza, CA (US); **Rick L. Orsini**, Flower Mound, TX (US); **Roger S. Davenport**, Campbell, TX (US); **Steven Winick**, Roslyn Heights, NY (US)

(73) Assignee: **Security First Corp.**, Rancho Santa Margarita, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **15/977,748**

(22) Filed: **May 11, 2018**

(65) **Prior Publication Data**

(58) **Field of Classification Search**
CPC H04L 63/04; H04L 63/0428; H04L 63/08; H04L 63/0823; H04L 63/0876;
(Continued)

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,453,074 A 6/1984 Weinstein
4,802,220 A 1/1989 Marker, Jr.
(Continued)

FOREIGN PATENT DOCUMENTS

AU 2004248616 6/2009
CN 1329417 A 1/2002
(Continued)

OTHER PUBLICATIONS

Barlas, "RSA's Security Showcase", Line56.com—The E-Business Executive Daily, Apr. 15, 2003.

'116 Patent (EX1001)

’116 Patent – Claim 1

’116 Patent, Claim 1

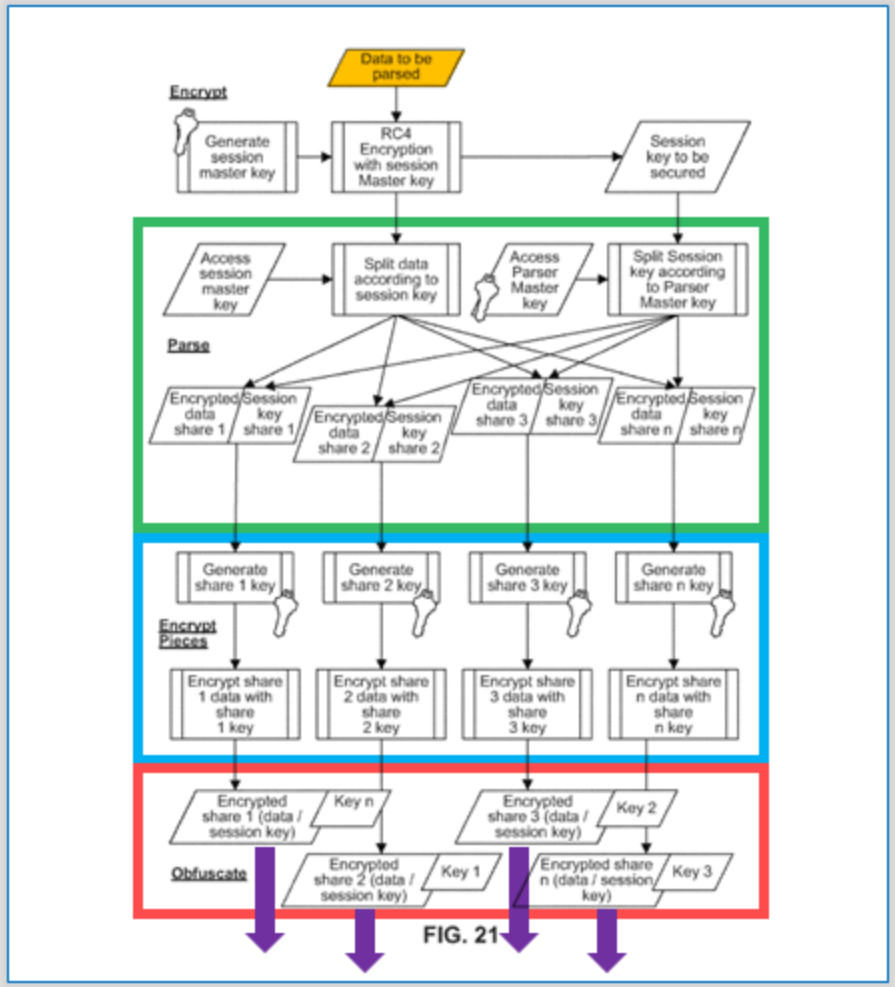
[1PRE] 1. A method for securing a data set, the method comprising:

[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.



’116 EX1001, FIG. 21 (annotated)
(cited ’116 Pet., 2)

’116 Patent – Claim 9

’116 Patent, Claim 9

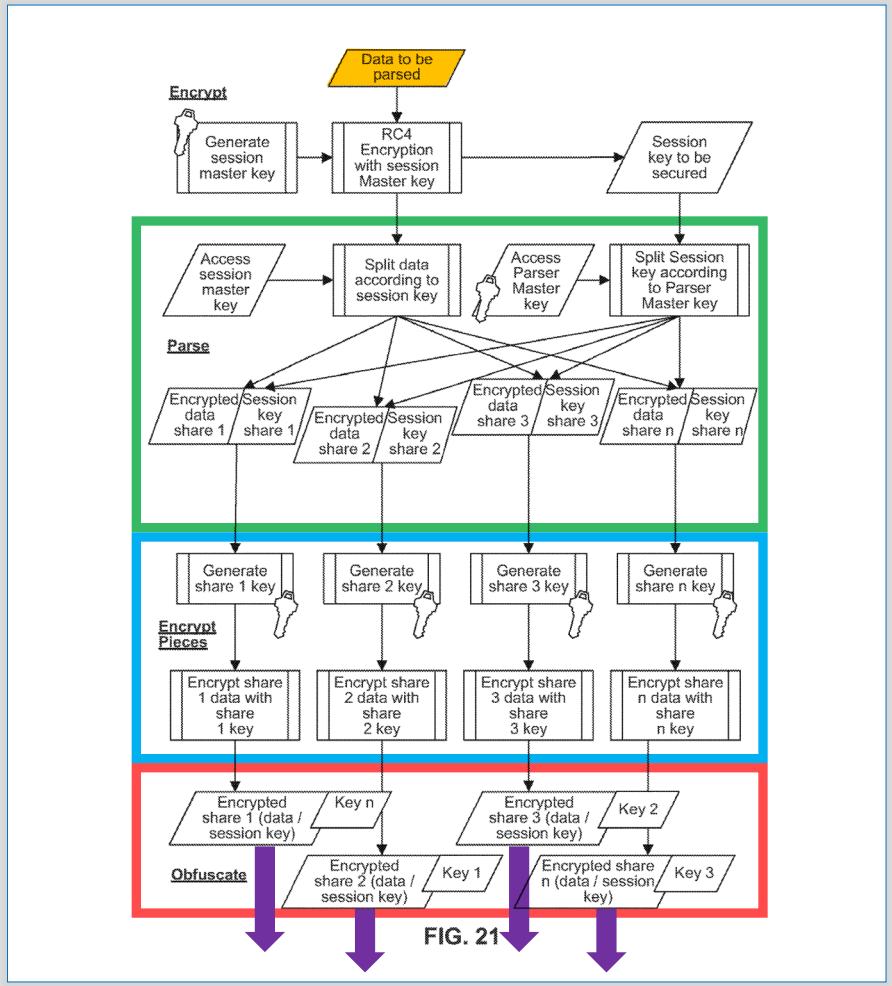
[9PRE] 9. A method for securing a data set, the method comprising:

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

[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.



’116 EX1001, FIG. 21 (annotated)
(cited ’116 Pet., 2-6, 83)

Overview of the ’116 Petition’s Grounds

GROUND	CLAIMS	BASIS
1. Tajima (EX1105)	**1, 5-8**	**Obviousness/ Anticipation**
2. Tajima, Foster (EX1106)	1-2, 9, 11-14	Obviousness
3. Tajima, Rivest (EX1034)	9-14	Obviousness
4. Tajima, Rivest, Shyu (EX1016)	1-3, 6	Obviousness
5. Foster, Hayhurst (EX1007)	1-2, 5, 9, 11	Obviousness

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 1: The Parties’ disputes are narrow

- No dispute that Tajima discloses limitation [**1PRE**]-[**1C**], and limitations added by claims 1, 5-8

- **Only dispute concerns limitation [1D]:** if Tajima’s “key information” is not the encryption key (anticipation) would a POSA have motivation to store Tajima’s encryption keys in the “key information” (obviousness)

’116 POR, 1-8
’116 Sur-Reply, 13-14

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

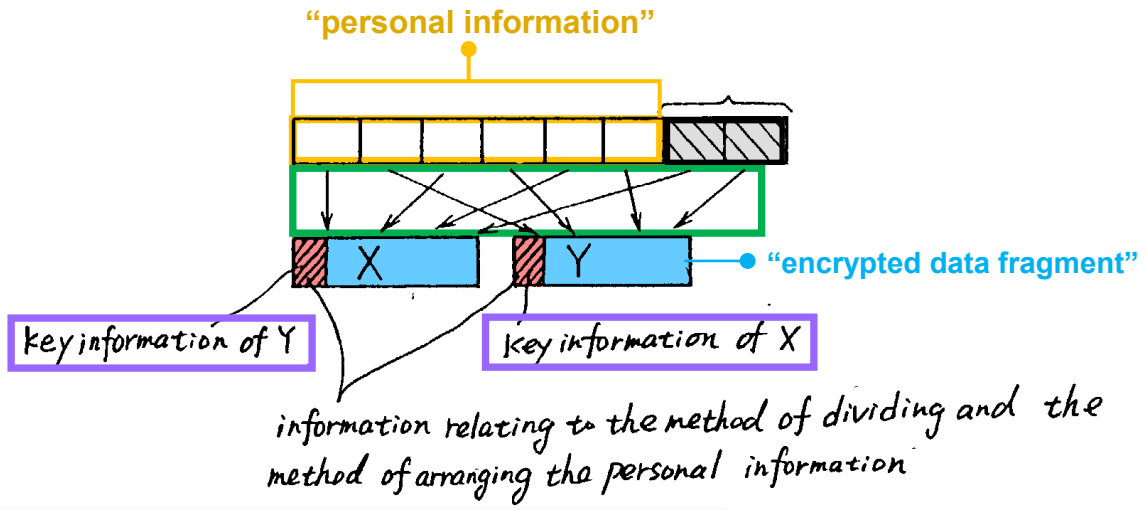
’116 Ground 1: Claim 1 is obvious over and anticipated by Tajima

	’116 Patent, Claim 1	Tajima
[1PRE]	A method for securing a data set, the method comprising:	**[0113]** As yet another method of dividing personal information, as shown in FIG. 13, 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, each of fixed length as shown in FIG. 8 or of arbitrary length as shown in FIG. 9 or FIG. 10. 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 fragment X and data fragment 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.
[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.	

’116 EX1005, [0113] (cited ’116 Pet., 14-25)

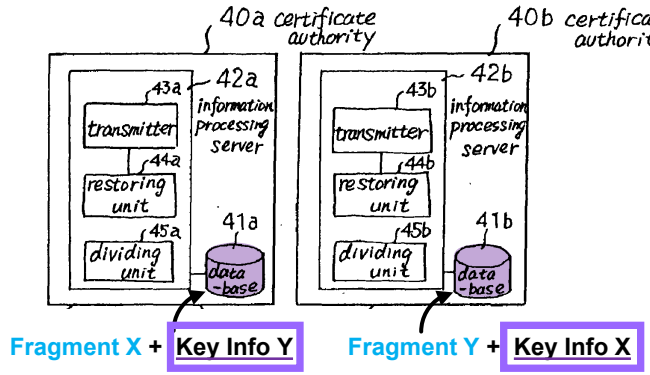
Tajima

Fig. 13



’116 EX1005, FIG. 13 (annotated) (cited ’116 Pet., 16-22)

Fig. 5



’116 EX1005, FIG. 5 (excerpt) (annotated) (cited ’116 Pet., 23)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 1: Multiple pieces of corroborating evidence demonstrate that storing an encryption key with an encrypted data fragment was well-known

Foster (EX1006)

[0064] Referring now to **FIG. 8**, the encryption and attachment of a title key to encrypted content under the present invention is shown. As depicted, an elementary content stream **100** comprising of compressed content is parceled into content units **102**. Each content unit includes a header **104** and a content packet **106**. Header **104** includes information helpful in identifying the content such as a time stamp, type of content: video, audio or other. Header **104** is especially useful, for example, in synchronizing sound data with video data. Once parceled, header **104** of each content unit **102** is parsed from content packet **106** and content packet **106** is encrypted with title key **108** to yield encrypted content packet **114**. Then, key encrypting key **110** is used to encrypt title key **108**, which is then attached to content packet **106** along with header **104**. The resulting processed content unit **116** can then be transmitted to a recipient. That

’116 EX1006, [0064] (cited ’116 Pet., 19)

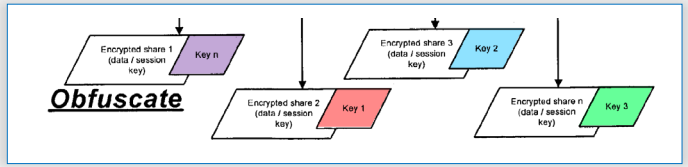
Shyu (EX1016)

sponding one of the data segments. Each of the segment headers is placed before the corresponding one of the data segments, and each of the source data segments is scrambled according to the scrambling type and the key word identified in the corresponding one of the segment headers to obtain an encrypted data stream.

’116 EX1016, 1:53-58 (cited ’116 Pet.,19)

Orsini (EX1013)

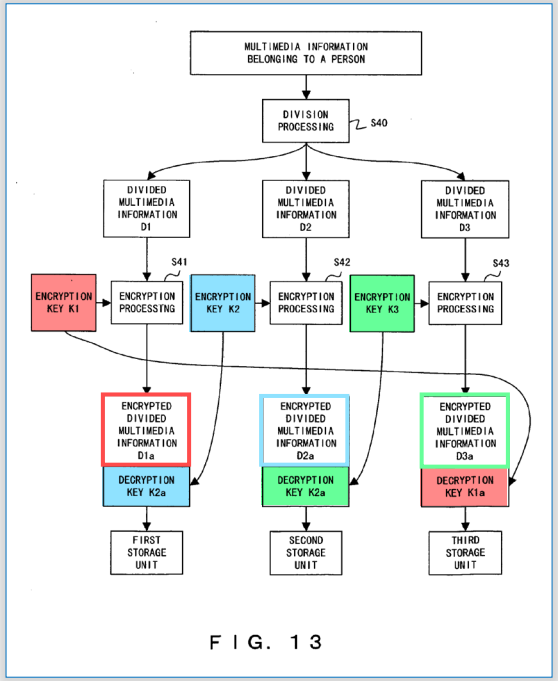
[0299] 5. Encrypting each share, then store the encryption keys in different locations from the encrypted data portions or shares: Share 1 gets Key 4, Share 2 gets Key 1, Share 3 gets Key 2, Share 4 gets Key 3.



’116 EX1013, [0297], [0299], FIG. 21 (partial)
(cited ’116 Pet., 6; ’116 Reply, 9)

EX1113

[0092] Then the encrypted divided multimedia information D1a is stored in the first storage unit 53 along with the decryption key K2a (i.e., the encryption key K2); the encrypted divided multimedia information D2a is stored in the second storage unit 31 along with the decryption key K3a (i.e., the encryption key K3); and the encrypted divided multimedia information D3a is stored in the third storage unit 41 along with the decryption key K1a (i.e., the encryption key K1).



F I G. 1 3

’116 EX1113, [0092] FIG. 13
(cited ’116 Reply, 9)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 1: Foster’s “key information” includes the encryption key and is stored with its encrypted piece of data

Foster

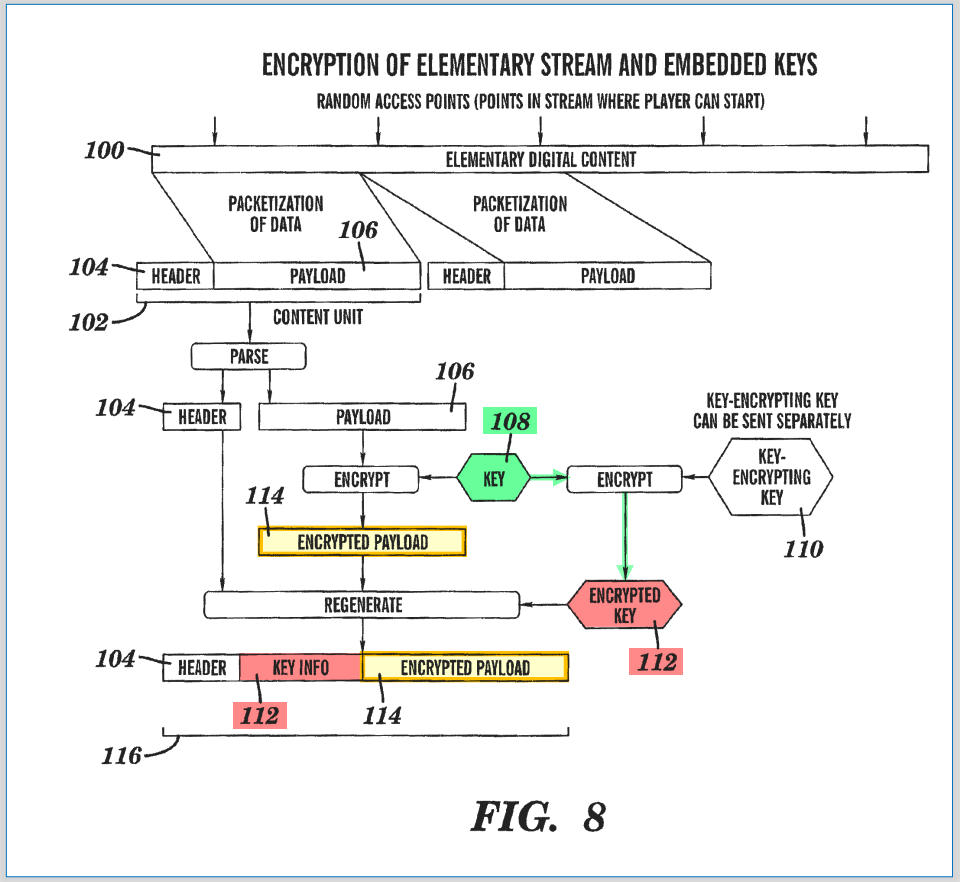


FIG. 8

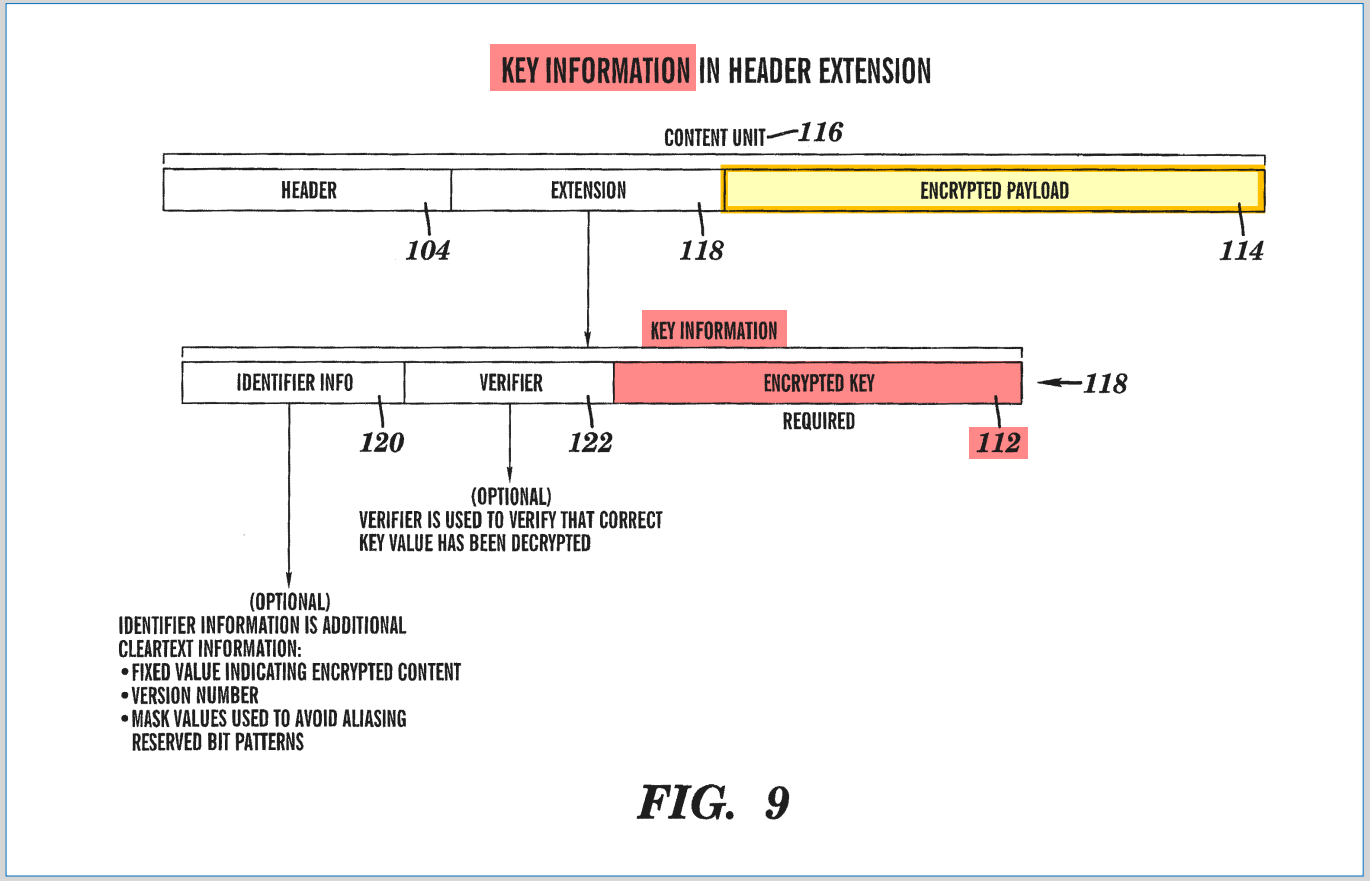


FIG. 9

’116 EX1006, FIG. 8-9 (cited ’116 Reply, 4)

’116 Ground 1: Tajima itself stores an “encryption key” with encrypted data

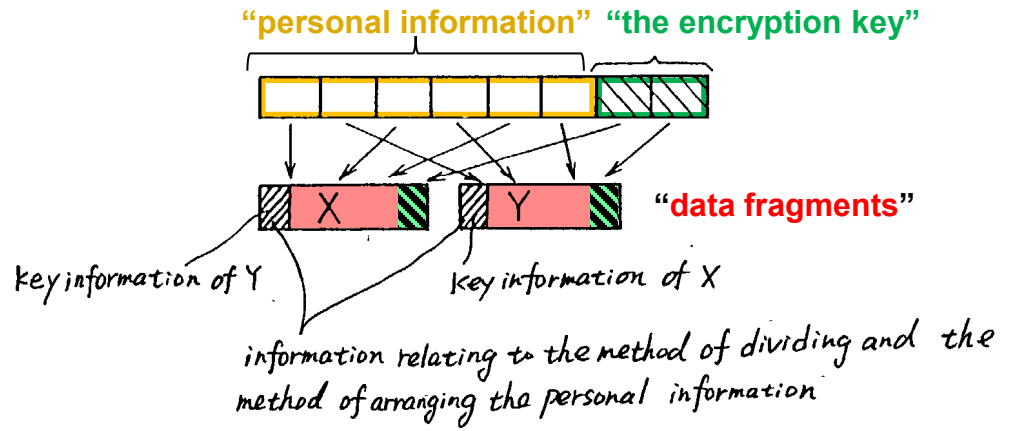
Tajima

4. A personal information management system according to claim 1, wherein said authentication means: divides said personal information into a plurality of data portions each of a predetermined fixed length, arranges the plurality of divided data portions in at least two data fragments according to a set method of arranging, and registers each data fragment with a database provided in said authentication means or in said other areas.

8. A personal information management system according to claim 4, wherein said authentication means encrypts said personal information and divides the encrypted personal information and the encryption key used in the encryption into a plurality of data portions.

’116 EX1005, Claims 4, 8
(cited ’116 Pet., 21; ’116 Reply, 6-8)

Fig. 13



’116 EX1005, FIG. 13 (cited ’116 Pet., 21-22)

Reply

which is false. POR, 5-6. The Petition’s obvious implementation relies on the same evidence in Tajima’s disclosure that (1) nothing more than the key information attached to the data fragment is used to decrypt them, (2) all the information needed to recover the personal information is stored in the data fragments, and (3) Tajima’s claims demonstrate the “key information” includes the key. Petition, 21-22 (citing Bhattacharjee, ¶¶96-106); Bhattacharjee-Reply, ¶¶24-25.

’609 Paper 21 at 8

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 1: POSAs had motivation to include the corresponding key in Tajima’s “key information”

Dr. Bhattacharjee

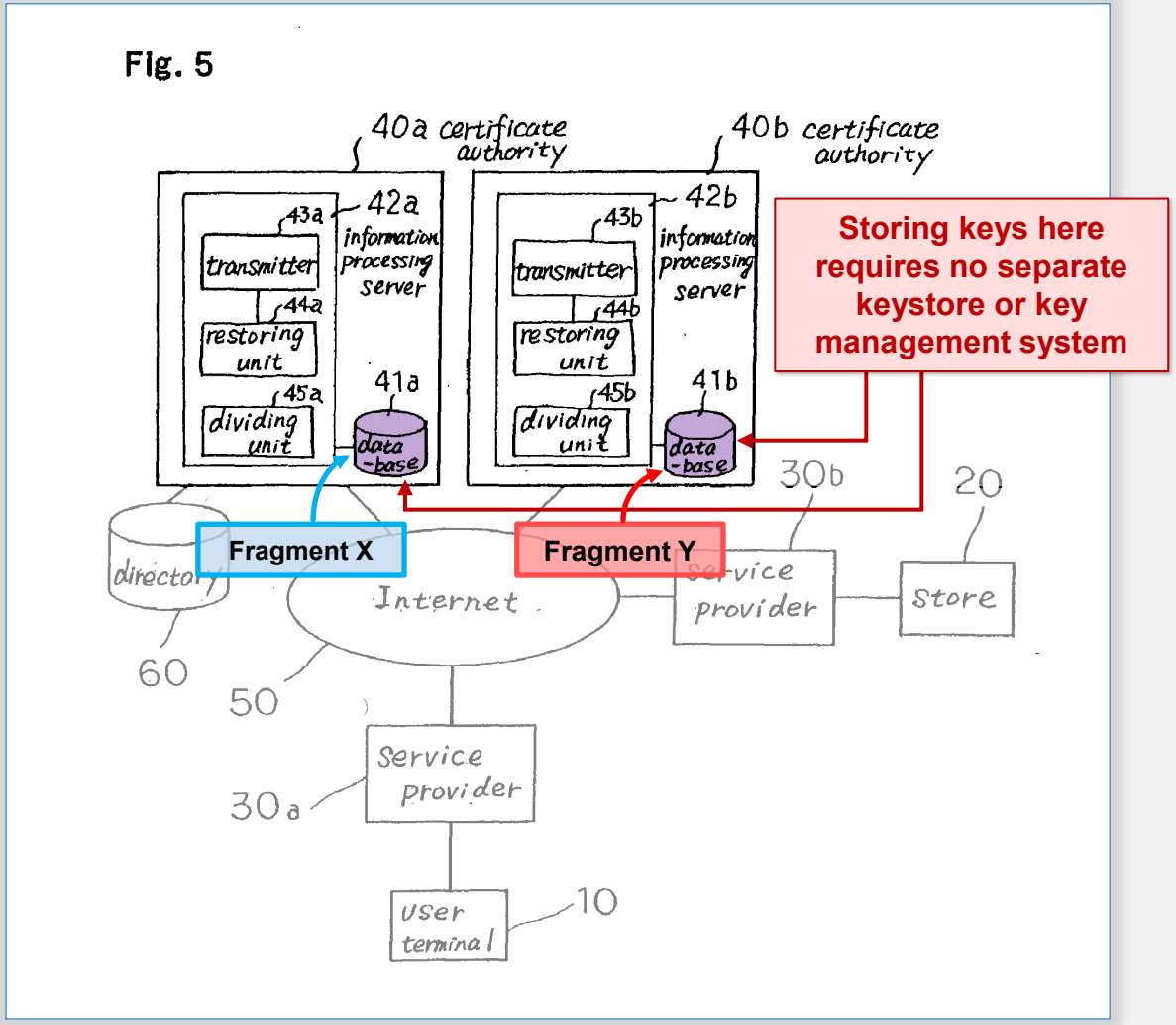
104. Even if Tajima had not disclosed that its key information in the Fig. 12 and 13 embodiments (Tajima, [0111]-[0114]) includes the corresponding encryption key, that would have been an obvious and conventional way to implement what Tajima discloses so that the service site authorized to recover personal 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.

’116 EX1003, ¶104 (cited ’116 Pet., 22)

SFI’s POR

DOES NOT DISPUTE

Tajima



’116 EX1005, FIG. 5 (cited ’116 Pet., 23)

’116 Ground 1: Including Tajima’s encryption key in the key information was a suitable design choice

SFI’s Dr. Rubin

Q. There is an encryption key, correct?

A. There is.

Q. And it has to be stored somewhere, right?

A. It does.

Q. So if Tajima is not storing its encryption keys with the data, the encryption keys are being stored somewhere else?

A. I think that would make sense.

Q. And so -- but Tajima doesn't tell you about where the keys are stored; is that correct?

A. It doesn't go into that detail.

Q. So it's your opinion, though, that a person skilled in the art would know that there must be some storage system that's storing Tajima keys?

A. I think that that is a natural conclusion short of there being like smart cards with keys on them or keys derived from passwords, but, in general, I think what it's saying, that there's some storage somewhere of the keys.

follow. All Tajima discloses is that “information regarding the key” is stored with the data fragment. It is silent about what key information is and where the key itself is stored.

’116 EX1099, 178 (cited ’116 POR, 8-9)’116 EX2033, ¶45 (cited ’116 POR, 4)

Dr. Bhattacharjee

28. POSAs knew that to decrypt the data fragments using the key information as Tajima discloses, either the keys are included in the key information, or the keys are stored and maintained elsewhere. Both techniques were known, “conventional” and obvious. If Tajima were in fact “silent about what key information is and where the key itself is stored,” as Dr. Rubin alleges (EX2033, ¶45), any implementation would require modifying Tajima in some way that “involve[s] where and how to store encryption keys.” EX2033, ¶49. But according to Dr. Rubin, any such technique cannot be “merely ‘conventional’”—i.e., there would be no conventional way to implement Tajima. I disagree.

’116 EX1111, ¶28 (cited ’116 Reply, 10)

“[O]ur case law does not require that a particular combination must be the preferred, or the most desirable, combination described in the prior art in order to provide motivation for the current invention.”

In re Fulton, 391 F.3d 1195, 1200 (Fed. Cir. 2004) (italics original) (cleaned up) (cited ’116 Reply at 11)

“It’s enough for Intel to show that there was a known problem of cache coherency in the art, that Bauman's secondary cache helped address that issue, and that combining the teachings of Kabemoto and Bauman wasn't beyond the skill of an ordinary artisan. Nothing more is required to show a motivation to combine under *KSR*…”

Intel Corp. v. PACT XPP Schweiz AG, 61 F.4th 1373, 1379-81 (Fed. Cir. 2023) (cleaned up) (cited ’116 Reply at 12)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 1: Tajima itself suggests including the key in the “key information” because Tajima says only “key information” is needed to decrypt a fragment

Petition

First, Tajima discloses that to restore secured personal information, “key information that has been attached to the encrypted data fragment X is used to decrypt data fragment Y, and key information that has been attached to encrypted data fragment Y is used to decrypt data fragment X.” Tajima, [0112], [0114]. This teaches that the key information includes the key, because Tajima never says that anything else beyond the key information is needed for decrypting the corresponding data fragment and does not disclose any additional steps nor any separate key management infrastructure. Bhattacharjee, ¶¶97-98, 103. Tajima also teaches storing with the data fragments how personal information was split and arranged, reinforcing that Tajima’s data fragments include sufficient information for the authorized service site to recover the personal information. Tajima, [0104]-[0105], [0107]-[0108], [0112], [0114]; Bhattacharjee, ¶98.

* * *

Even if Tajima had not disclosed that its key information includes the corresponding key, that would have been an obvious and conventional way to implement what Tajima discloses—i.e., a system that needs only the key information to decrypt data fragments and not additional key management infrastructure. Bhattacharjee, ¶104.

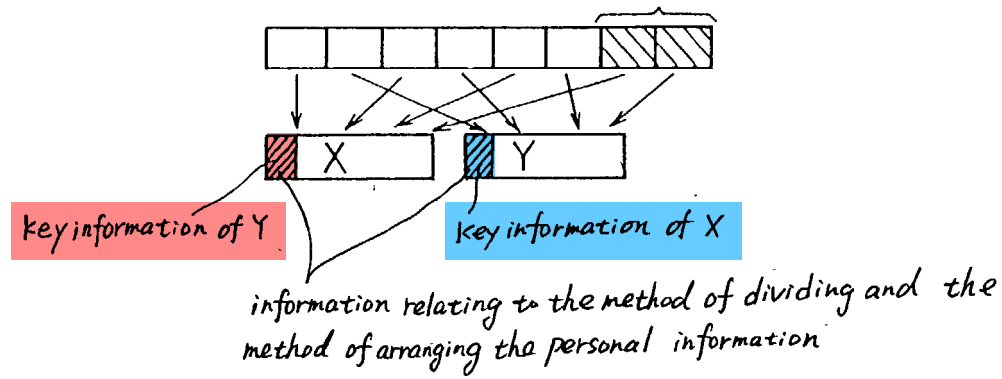
’116 Paper 2 at 20-22

Tajima

[0114] When combining data fragment X and data fragment Y that have been divided in this way, 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. Then, if information relating to the

’116 EX1005, [0010], [0013]-[0014]
(cited ’116 Pet., 21-22)

Fig. 13



’116 EX1005, FIG. 13 (cited ’116 Pet., 19-21)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 1: SFI’s Arguments

SFI argues storing an obfuscated encryption key with Tajima’s encrypted data fragments was not obvious because:

1. The limitation of storing an encrypted data chunks with a key ([1D]) is missing in the prior art

---

2. Storing keys with encrypted data chunks would have been complex and entailed risks

’116 POR, 5-8
’116 Sur-Reply, 13-14

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 1: Including the corresponding encryption key in Tajima’s “key information” was obvious

1. Storing an encryption key with encrypted data was not a missing limitation; it was well known and conventional, as demonstrated by multiple corroborating references

---

2. A POSA had motivation to include Tajima’s encryption key in Tajima’s “key information” because it would have beneficially not required any additional key-storage or key-management infrastructure in Tajima, consistent with Tajima’s disclosure

’116 Pet., 8-33
’116 Reply, 1-12

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 1 (obviousness): SFI’s Argument #1 (“missing limitation”) fails because it is undisputed that storing encryption keys with encrypted data chunks was known and conventional

SFI’s POR

Alternatively, Petitioner argues that it would have been obvious to include the key itself in “information about the key.” Pet., 22 (citing EX1003 ¶ 104). But beyond Dr. Bhattacharjee’s conclusory testimony, which has no support in Tajima, Petitioner offers no evidence or explanation as to why this “would have been an obvious and conventional way” to modify Tajima. To the extent Petitioner relies on the purported “common sense” of a POSITA, that be used to supply a missing limitation “***only*** when ‘the limitation in question [is] unusually simple and the technology particularly straightforward.’” *DSS Tech. Mgm’t* v. *Apple Inc.*, 885 F.3d 1367, 1374 (Fed. Cir. 2018) (citations omitted). The complexity and risks associated with modifying cryptographic systems bar such an argument. (*See* EX2033 ¶ 48).

’116 Paper 17 at 5-6

Reply

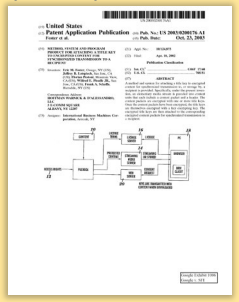
Thus, even if that evidence were found not to demonstrate that Tajima’s key information includes the key, straightforwardly implementing Tajima that way plainly would have been obvious and desirable so that a site authorized to recover Tajima’s secured data “need only retrieve the stored encrypted data fragments to acquire the keys needed to decrypt them,” rather than requiring “a separate key-management system” to store those keys. Bhattacharjee, ¶104; EX1099 (“Rubin-Depo.”), 177-178 (admitting keys “ha[ve] to be stored somewhere”). SFI never disputes that implementing Tajima in this way would have been obvious and beneficial. Bhattacharjee-Reply, ¶¶26-27.

SFI ***also*** does not dispute this implementation was conventional. Storing an encryption key with the encrypted data was known. Like Tajima, EX1013 and EX1113 both store encryption keys with encrypted data it cannot decrypt. EX1013, [0294]-[0299], [0354]-[0357] (reproduced below); EX1113, [0090]-[0098], FIG. 13 (annotated below); Bhattacharjee, ¶¶31-32, 54 (cited Petition, 6); Bhattacharjee-Reply, ¶¶28-32.

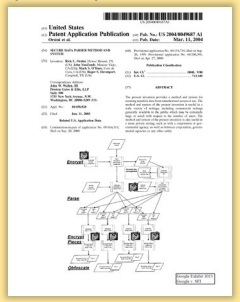
’116 Paper 21 at 8-9

Corroborating Evidence

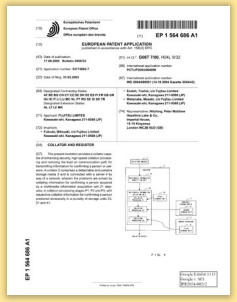
Foster (EX1006)



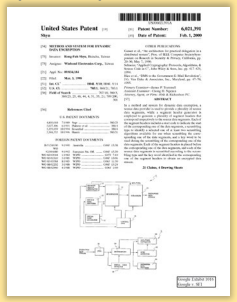
Orsini (EX1013)



EX1113



Shyu (EX1016)



’116 Pet., 6, 19;
’116 Reply, 8-10

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 1 (obviousness): SFI’s Argument #1 (“missing limitation”) fails because evidence of a POSA’s background knowledge does not need to be named in the ground to be considered

SFI’s Sur-Reply

Petitioner also argues that modifying Tajima’s key information to include the key itself would have been “obvious” and “conventional.” Reply, 8-9. As to conventionality, Petitioner relies on prior work by three of the ’116 Patent’s inventors that it did not include as a ground for invalidity in the Petition. *See* Reply, 9-10. As to obviousness, the POR showed that the Petition could not rely on “common sense” to supply a limitation that Tajima did not itself disclose. POR, 5-6. Petitioner’s Reply now disclaims reliance on common sense, and contends it relied on “reasoned analysis,” but they amount to the same thing: its expert’s same unfounded inferences and speculations already addressed in the POR. *See* Reply, 8-11; POR, 4-8.

’116 Paper 29 at 14

First, Philips argues…§ 311(b) prohibits use of general knowledge to supply a missing claim limitation in an inter partes review. We disagree.…
Here, Google properly alleged that although SMIL 1.0 did not disclose each and every element of the claimed invention, the differences between the claimed invention and SMIL 1.0 are such that the claimed invention would have been obvious to a person having ordinary skill in the art when considering SMIL 1.0. In particular, Google properly alleged that a skilled artisan would have known about pipelining and been motivated to combine pipelining with SMIL 1.0.

Koninklijke Philips N.V. v. Google LLC, 948 F.3d 1330, 1337-38 (Fed. Cir. 2020) (cited ’609 Reply, 24; ’854 Reply, 22)

’116 Ground 1: SFI’s Argument #1 (“missing limitation”) also fails because the Petition’s obviousness argument is supported by evidence and analysis, not bare common sense

SFI’s POR

Petitioner offers no evidence or explanation as to why this “would have been an obvious and conventional way” to modify Tajima. To the extent Petitioner relies on the purported “common sense” of a POSITA, that be used to supply a missing limitation “***only*** when ‘the limitation in question [is] unusually simple and the technology particularly straightforward.’” *DSS Tech. Mgm’t* v. *Apple Inc.*, 885 F.3d 1367, 1374 (Fed. Cir. 2018) (citations omitted). The complexity and risks associated with modifying cryptographic systems bar such an argument. (*See* EX2033 ¶ 48).

’116 Paper 17 at 6

SFI’s Sur-Reply

DOES NOT DISPUTE

See ’116 Paper 28 at 14

Reply

POSAs knew that to implement Tajima, either keys are included in the key information stored with the data, or keys are stored and maintained elsewhere. Both techniques were known, “conventional,” and obvious. Bhattacharjee-Reply, ¶¶28-32; Rubin-Depo., 177-178. Unable to dispute this, SFI alleges that “to the extent” Petitioner relies on “common sense,” that reliance would be improper. POR, 6 (citing *DSS Tech.*). Unlike in *DSS Tech*, where the Federal Circuit faulted the Board’s invocation of “ordinary creativity” as a “wholesale substitute for reasoned analysis and evidentiary support” (*DSS Tech.* at 1376-77), Dr. Bhattacharjee and the Petition straightforwardly reasoned that implementing Tajima so that nothing beyond the key information is needed to decrypt the data fragments (as Tajima discloses), rather than implementing key-management technology (about which Tajima says nothing), would have been conventional and obvious. Petition, 20-22;

’116 Paper 21 at 10-11

“The Board thus relied on ‘ordinary creativity’ as a wholesale substitute for reasoned analysis and evidentiary support… Without a reasoned explanation that avoids conclusory generalizations, this was not sufficient.”

DSS Tech. Mgmt., Inc. v. Apple Inc., 885 F.3d 1367, 1377 (Fed. Cir. 2018) (cited ’116 Reply, 10-11)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

'116 Ground 1: SFI's Argument #2 ("complexity") fails because including Tajima's keys in the "key information" would avoid the need for a separate keystore or additional key management

SFI's POR

Nor does Petitioner demonstrate a POSITA would be motivated to make the proposed alteration. As Petitioner acknowledges, a POSITA would understand there are significant security drawbacks to including the encryption key itself in the "key information" because "[i]t was known that storing plaintext (i.e., unencrypted) keys with encrypted data results in a vulnerability." Pet., 33-34; EX1003, ¶140, *citing, e.g.*, EX1021, 1:45-65 (explaining that methods "where the data encryption keys are stored or transmitted together with the encrypted data" are potentially defeatable). Encrypting the data fragment encryption key with yet another key before storing it with the fragment would require additional key management, contrary to what the Petition asserts as Tajima's goal: "a system that needs only the key information to decrypt data fragments and not additional key management infrastructure." *See* Pet.,

'116 Paper 17 at 6-7

Reply

C. SFI's Lack-of-Motivation Argument Fails

SFI alleges that the Petition failed to demonstrate motivation and points to Petitioner's acknowledgement that storing plaintext keys has risks. POR, 6-7. But SFI ignores that Tajima addresses some of that risk by storing key information with a data fragment it cannot decrypt. Petition, 33; Bhattacharjee, ¶140. The fact that POSAs knew that storing plaintext keys with encrypted data, even when obfuscated, still has risk does not make that known implementation non-obvious. Bhattacharjee-Reply, ¶¶32. Obviousness does not require the "most desirable" implementation." *In re Fulton*, 391 F.3d 1195, 1200 (Fed. Cir. 2004).

Including keys in Tajima's key information meant the authorized service site "need only retrieve the stored encrypted data fragments to acquire the keys needed to decrypt them, [] rather than implementing a separate keystore or key management system" to store them (Bhattacharjee, ¶104), while Tajima's obfuscation protects against attackers that acquire some but not all encrypted data fragments. Bhattacharjee, ¶141; Bhattacharjee-Reply, ¶33. This beneficial implementation is not rendered non-obvious by other obvious modifications that could improve Tajima's security. *Intel Corp. v. PACT XPP Schweiz AG*, 61 F.4th 1373, 1379-81 (Fed. Cir. 2023)

'116 Paper 21 at 11-12

"The motivation-to-combine analysis is a flexible one…. It's enough for Intel to show that there was a known problem of cache coherency in the art, that Bauman's secondary cache helped address that issue, and that combining the teachings of Kabemoto and Bauman wasn't beyond the skill of an ordinary artisan. Nothing more is required to show a motivation to combine under *KSR* …."

Intel Corp. v. PACT XPP Schweiz, 61 F.4th 1373, 1379-81 (Fed. Cir. 2023) (cited '116 Reply, 12)

SFI's Sur-Reply

DOES NOT DISPUTE

See '116 Paper 28 at 14

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 1 (Anticipation): Tajima’s “key information” includes the key itself

POSAs knew Tajima’s “key information” already includes the encryption key based on:

1. What Tajima expressly says about how “key information” is used

2. Tajima’s claims expressly reciting an “encryption key” that maps to “key information”

3. Multiple corroborating references that use the term “key information” to - include (at least) the corresponding encryption key

’116 Pet., 18-22
’116 Reply, 1-12

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 1 (Anticipation): Tajima only uses the “key information” to decrypt data fragments

Dr. Bhattacharjee

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.

’116 EX1003, ¶97 (cited Pet., 21)

Tajima

[0114] When combining data fragment X and data fragment Y that have been divided in this way, 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. Then, if information relating to the

’116 EX1005, [0114]
(cited ’116 Pet., 8, 18-21)

SFI’s POR

DOES NOT DISPUTE

See ’116 Paper 28 at 1-8

’116 Ground 1: Tajima’s claims teach that Tajima uses the term “key information” to include the encryption key itself

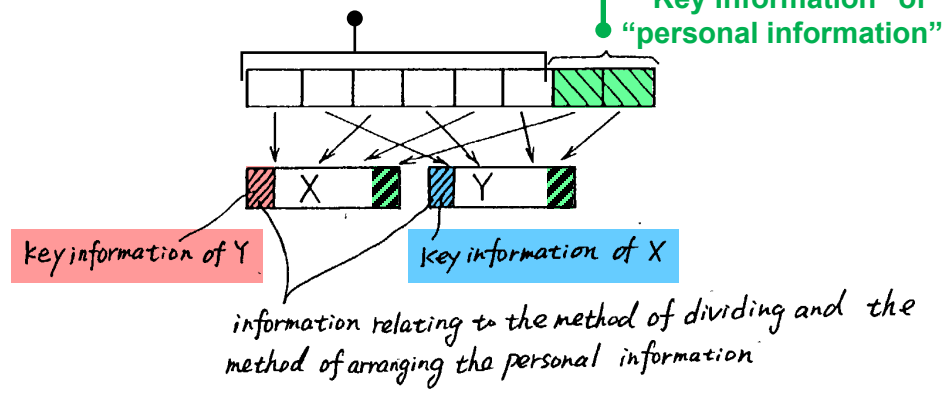
Tajima

4. A personal information management system according to claim 1, wherein said authentication means: divides said personal information into a plurality of data portions each of a predetermined fixed length, arranges the plurality of divided data portions in at least two data fragments according to a set method of arranging, and registers each data fragment with a database provided in said authentication means or in said other areas.

8. A personal information management system according to claim 4, wherein said authentication means encrypts said personal information and divides the encrypted personal information and the encryption key used in the encryption into a plurality of data portions.

’116 EX1005, Claims 4, 8
(cited ’116 Pet., 21-22; ’116 Reply, 6-8)

Fig. 13



’116 EX1005, FIG. 13 (cited ’116 Pet., 12-25)

[0110] When combining data fragment X and data fragment Y that have been divided in this way, data fragment X and data fragment Y are combined based on information relating to the method of dividing and the method of arranging if information relating to the method of dividing and the method of arranging has been attached to data fragments X and Y, and the key information that was attached to the encrypted personal information is then used to decrypt the encrypted personal information.

[0113] As yet another method of dividing personal information, as shown in **FIG. 13**, personal information that the user of user terminal **10** has submitted to certificate authority **40***a* 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, each of fixed length as shown in **FIG. 8** or of arbitrary length as shown in **FIG. 9** or **FIG. 10**. This

[0114] When combining data fragment X and data fragment Y that have been divided in this way, 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. Then, if information relating to the

’116 EX1005, [0110], [0113]-[0114]
(cited ’116 Pet., 21-22; ’116 Reply, 1-2, 6)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 1: SFI’s own patents use the term “key information” to be information that includes the encryption key itself

’854 Patent

The workgroup key concept allows for enhanced protection of information assets by encrypting key information stored within the shares. Once this operation is performed, even if all required shares and external keys are discovered, an attacker has no hope of recreating the information without access to the workgroup key.

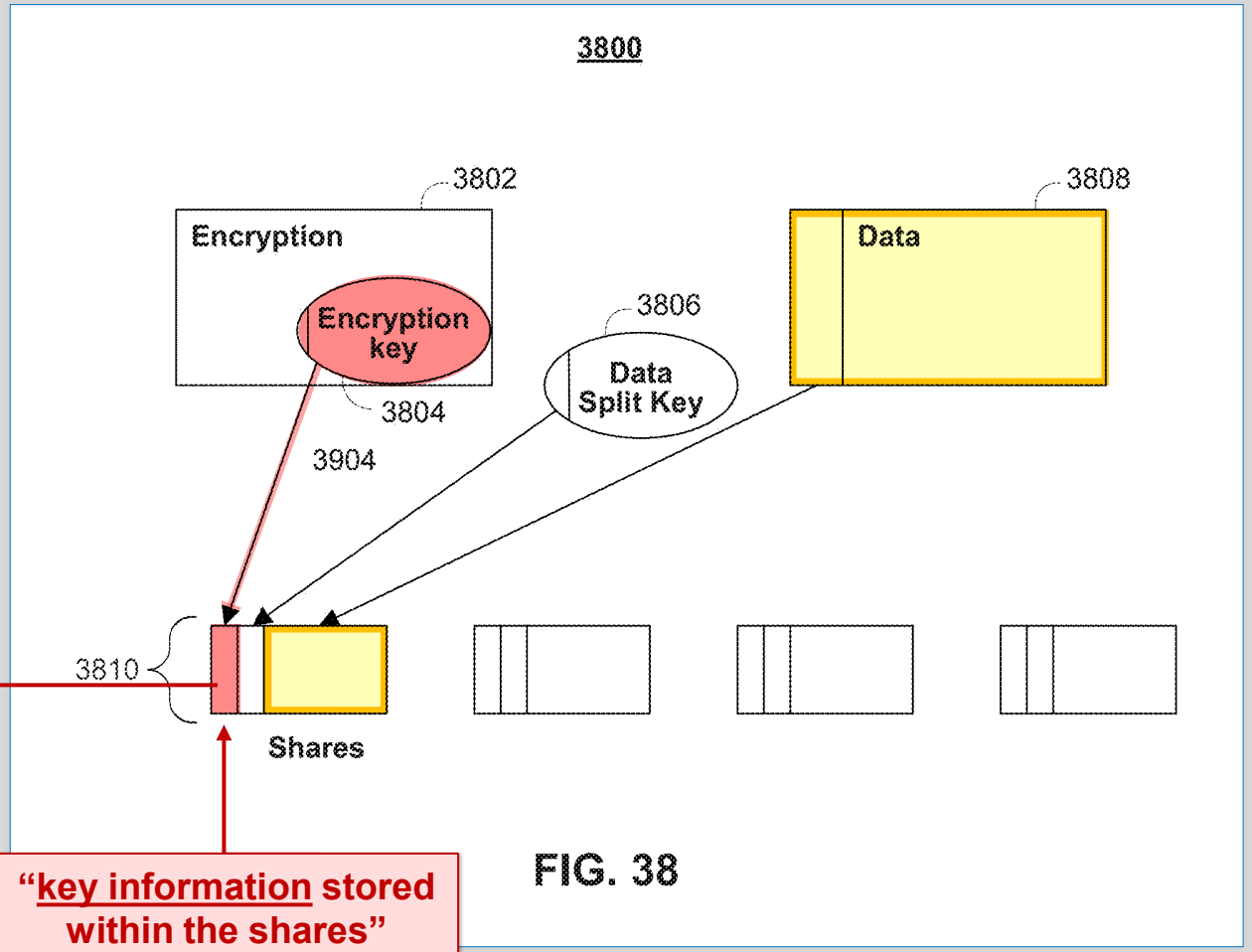
FIG. **38** shows illustrative block diagram **3800** for storing key and data components within the shares. In the example of diagram **3800**, the optional pre-encrypt and post-encrypt steps are omitted, although these steps may be included in other embodiments.

The simplified process to split the data includes encrypting the data using encryption key **3804** at encryption stage **3802**. Portions of encryption key **3804** may then be split and stored within shares **3810** in accordance with the present invention. Portions of split encryption key **3806** may also be stored within shares **3810**. Using the split encryption key, data **3808** is then split and stored in shares **3810**.

In order to restore the data, split encryption key **3806** may be retrieved and restored in accordance with the present invention. The split operation may then be reversed to restore the ciphertext. Encryption key **3804** may also be retrieved and restored, and the ciphertext may then be decrypted using the encryption key.

When a workgroup key is utilized, the above process may be changed slightly to protect the encryption key with the workgroup key. The encryption key may then be encrypted with the workgroup key prior to being stored within the shares. The modified steps are shown in illustrative block diagram **3900** of FIG. **39**.

’116 EX1015, 79:66-8:23 (cited ’116 Reply, 6)



’116 EX1015, FIG. 38 (cited ’116 Reply, 6)

’116 Ground 1: It was conventional for “key information” to include (at least) the key itself

EX1114

[0022] Unlike the two aforedescribed traditional encryption techniques (i.e. public-key and private-key encryption) and the Infosafe system, in the present invention the key information preferably comprises at least a public key which is accessible to both the owner of the data and the third party, and a master key the details of which are known only to the data owner. The master key is preferably used in order to generate the public key. The encrypting step therefore preferably also comprises the step of generating the public key.

’116 EX1114, [0022] (cited ’116 Reply, 6)

EX1113

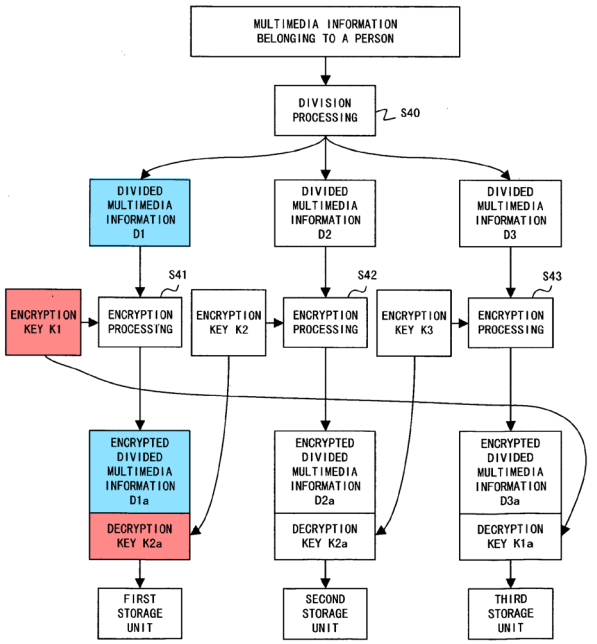


FIG. 13

[0092] Then the encrypted divided multimedia information D1a is stored in the first storage unit 53 along with the decryption key K2a (i.e., the encryption key K2); the encrypted divided multimedia information D2a is stored in the second storage unit 31 along with the decryption key K3a (i.e., the encryption key K3); and the encrypted divided multimedia information D3a is stored in the third storage unit 41 along with the decryption key K1a (i.e., the encryption key K1).

[0098] As described thus far, encrypting the dividedly stored multimedia information enhances the security. Storing the key information for decrypting the encrypted multimedia information in a storage unit separately from the one storing the encrypted multimedia information itself in the process prevents an illegitimate decrypting if the encrypted multimedia information is stolen, since the key for decrypting the encryption is stored in a different storage unit.

’116 EX1113, FIG. 13, [0092], [0098] (cited ’116 Reply, 9)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 1: Tajima anticipates based on its express, not inherent, disclosures

SFI’s POR

corresponding key.” Pet., 20 (citing only EX1003 ¶ 96).[1] To show anticipation by Tajima, Petitioner’s claim about what a POSITA “understood” must meet the requirements for an inherent disclosure. *Microsoft Corp. v. Biscotti, Inc.*, 878 F.3d 1052, 1065 (Fed. Cir. 2017) (statements that “a [POSITA] would understand the [evidence] as disclosing [the limitation]” could support an argument that a limitation was “inherently disclose[d] …”). Inherency requires demonstrating that a limitation is “‘necessarily present,’ not merely probably or possibly present, in the prior art.” *Trintec Indus., Inc. v. Top U.S.A. Corp.*, 295 F.3d 1292, 1295 (Fed. Cir. 2002). But Petitioner fails to demonstrate that “information regarding the key” would “necessarily” include the key. As Dr. Rubin explains:

’116 Paper 17 at 3

SFI’s Sur-Reply

draw from other disclosures Tajima. Reply, 2-7. This argument fails because the Federal Circuit has repeatedly explained that “[t]o anticipate a claim, a single prior art reference must disclose every limitation of the claimed invention either expressly or inherently.” *HTC Corp.* v. *Cellular Commc’ns Equip. LLC*, 877 F.3d 1361, 1368-69 (Fed. Cir. 2017); *see also Arbutus Biopharma Corp.* v. *ModernaTX, Inc.*, 65 F.4th 656, 661-65 (Fed. Cir. 2023).[2] It is undisputed that, in the embodiments on which the Petition relies, Tajima does not expressly state that the key itself is attached to data fragments. Thus, Petitioner is necessarily relying on an inherent disclosure, but cannot dispute that the Petition does not make the required showing. Reply, 2; POR, 2-3.

[2] Petitioner disputes the requirements for anticipation, relying on a 57-year-old case and a 20-year-old case that generally describe the same requirements using somewhat different language. *See* Reply, 2.

’116 Paper 29 at 13

Reply

1. Petitioner Did Not Rely On Inherency

An argument based on what POSAs “understood from [Tajima]’s teachings” is “***not*** [an] inherency”[1] argument (*InductEV Inc. v. Witricity Corp.*, IPR2021-01167, Paper 31, 45 (Nov. 22, 2022)) as SFI erroneously alleges. POR, 3.

A reference discloses a limitation by the “specific teachings of the reference [and] the inferences which [a POSA] would reasonably draw therefrom.” *In re Preda*, 401 F.2d 825, 826 (C.C.P.A. 1968); *see also ArthroCare Corp. v. Smith & Nephew, Inc.*, 406 F.3d 1365, 1373-74 (Fed. Cir. 2005) (“even if a piece of prior art does not expressly disclose a limitation, it anticipates if a [POSA] would understand the prior art to disclose the limitation and could combine the prior art description with his own knowledge to make the claimed invention”); *see also Microsoft v. Biscotti* at 1065, 1068 (cited POR, 3).

’116 Paper 21 at 2

“Turning to our analysis, we agree with Petitioner that the Petition relies on what one of ordinary skill in the art would have understood from Sato’s teachings, not inherency, by arguing that “[a] POSA would have understood that when the transmission distance changes, the coupling coefficient changes.” Pet. 27 (citing, e.g. Ex. 1003 ¶¶ 108-110). Thus, Patent Owner’s argument attacking Petitioner for relying on “inherency” misses the mark.”

InductEV Inc. v. Witricity Corp., IPR2021-01167, Paper 31 at 45 (PTAB) (cited ’116 Reply, 2)

Overview of the ’116 Petition’s Grounds

GROUND	CLAIMS	BASIS
1. Tajima (EX1105)	1, 5-8	Obviousness/ Anticipation
2. Tajima, Foster (EX1106)	**1-2, 9, 11-14**	**Obviousness**
3. Tajima, Rivest (EX1034)	9-14	Obviousness
4. Tajima, Rivest, Shyu (EX1016)	1-3, 6	Obviousness
5. Foster, Hayhurst (EX1007)	1-2, 5, 9, 11	Obviousness

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 2 (Tajima-Foster): Encrypts Tajima’s keys with a conventional KEK and stores the encrypted keys with data fragments, as Foster taught

’116 Petition

POSAs would have recognized that Tajima-Foster would have beneficially provided even further security for storage technique (2) above because encrypted data fragments/key pairs ***and*** the KEK needed to decrypt the keys are each stored on different databases under separate control of respective service sites—i.e., the KEK at the service site that divided the personal information, and the resulting encrypted data fragments and encrypted keys at other sites—requiring an attacker to not only acquire all data fragments and encryption keys stored on multiple different databases, but to also gain unauthorized access to another system to acquire the KEK. Bhattacharjee, ¶159.

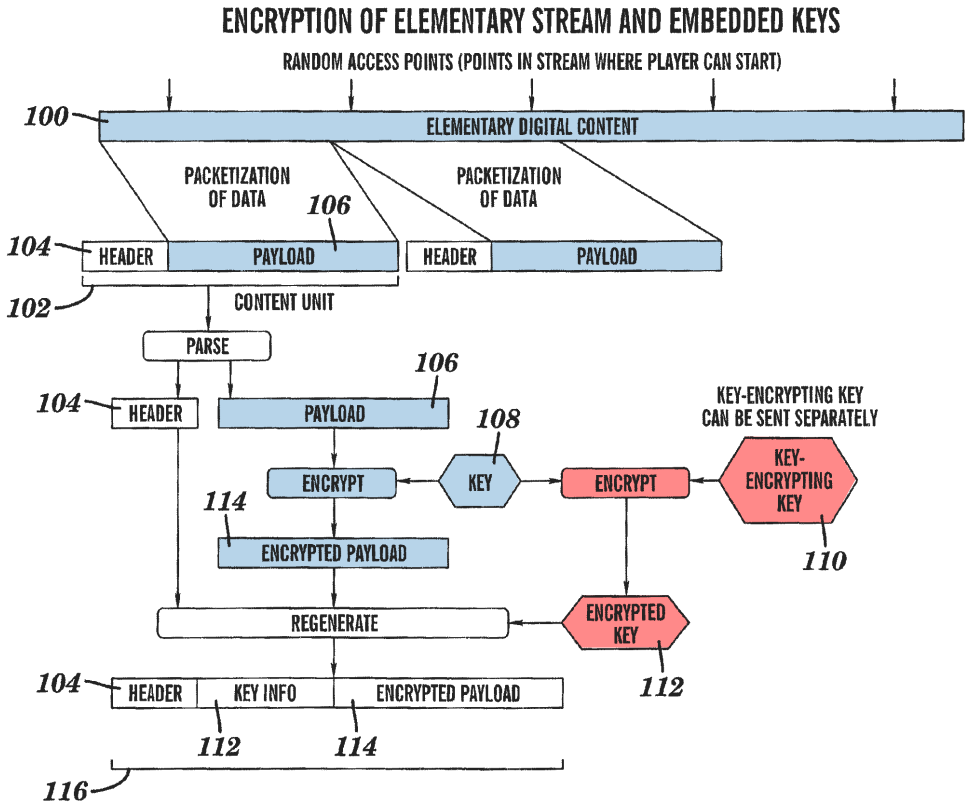
’116 Paper 2 at 39

SFI’s Sur-Reply

In fact, Foster discourages the use of its KEK technique with Tajima, by stressing the importance of keeping the title key with the content it encrypts. POR, 27. The only reason for using the KEK in Foster is ***because*** the title key is kept with the content it decrypts; the KEK provides security because it prevents someone with access to all of the content from decrypting it. *See* POR, 15-16;

’116 Paper 29 at 5

Foster



’116 EX1006, FIG. 8 (cited ’116 Pet., 36-37)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 2 (Tajima-Foster): KEKs addressed the known risk existing in Tajima from storing / transmitting plaintext (unencrypted) encryption keys

’116 Petition

A. Security Risk of Plaintext Keys

It was known that storing plaintext (i.e., unencrypted) keys with encrypted data results in a vulnerability. Bhattacharjee, ¶140, *citing* EX1021, 1:45-65 (explaining that simple-to-use methods “where the data encryption keys are stored or transmitted together with the encrypted data” are likewise potentially simple to defeat), EX1023, 17:44-18:8 (storing plaintext keys with the information that they protect “is somewhat akin to locking a house and then putting the keys under the doormat”); EX1022, 10:42-48 (“cryptographic keys should never occur in clear form outside the cryptographic device”).

While Tajima achieves some protection by storing the key used to encrypt a data fragment with a different data fragment, POSAs understood that by storing the key in plaintext, Tajima’s approach subjects the data fragments 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 possess all the plaintext keys used to encrypt the entire data set. Bhattacharjee, ¶141. 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. Bhattacharjee, ¶¶141-43 (describing known techniques to address the vulnerability of plaintext keys).

’116 Paper 2 at 33-34

EX1021

Security 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. This approach means that the data encryption keys will be encrypted by a key encryption key. This in turn leads to the

’116 EX1021, 1:66-2:4
(cited ’116 Pet., 33-34)

EX1022

Because the ciphering algorithm used is not secret, the degree of protection that can be derived from a cryptographic system ultimately depends upon the security of the cryptographic keys. Therefore, the objectives of key management are: (1) cryptographic keys should never occur in clear form outside the cryptographic device, except under secure conditions during the period when keys are originally distributed and installed or when stored in a secure place such as a safe, vault or similar location for backup or recovery and (2)

’116 EX1022, 10:42-51
(cited ’116 Pet., 33-34)

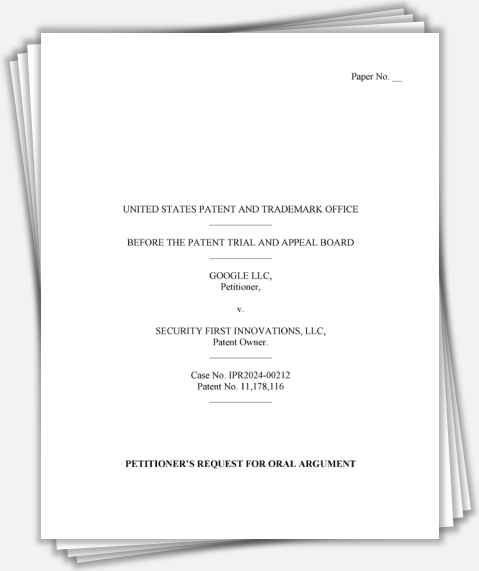
EX1023

Another scheme is strong media key management (“MM_KM_STRONG”). In this embodiment, data encryption keys are stored on the media, but we additionally encrypt them with the user’s pass-phrase. The pass-phrase

’116 EX1023, 18:28-31
(cited ’116 Pet., 33-34)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

The ’116 Institution Decision found a reasonable likelihood that claims 1-2, 9, 11-14 would have been obvious over Tajima-Foster



Paper No. __

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner,

v.

SECURITY FIRST INNOVATIONS, LLC,
Patent Owner.

Case No. IPR2024-00212
Patent No. 11,178,116

PETITIONER'S REQUEST FOR ORAL ARGUMENT

Institution Decision

3. Summary and Conclusion – Tajima Grounds

We have considered Petitioner’s analysis of the Tajima grounds and Patent Owner’s response. We determine that for the reasons given, Petitioner has demonstrated that there is a reasonable likelihood of prevailing on at least the Tajima-Foster challenge to independent claims 1 and 9. Patent Owner does not separately argue dependent claims 2 and 11–14 challenged in this ground. Therefore, for the reasons given for claims 1 and 9 and in the Petition, we determine also that Petitioner has also demonstrated that there is a reasonable likelihood of prevailing on the Tajima-Foster challenge to those claims. *See* Petitioner’s analysis of claims 2 and 11–14 at pages 42–46 of the Petition.

At this stage, for the purpose of institution, we do not see the need to address the other Tajima grounds that were not separately addressed by Patent Owner.

’116 Paper 11 at 19

’116 Ground 2: The Parties’ Disputes Are Narrow

- No dispute that Tajima-Foster meets all limitations of claims 1, 2, 9, and 11-14

- **Only dispute is whether POSAs had motivation to use a conventional key-encryption key (KEK), as taught by Foster (and other corroborating references), to further secure Tajima’s encryption keys**

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 2 (Tajima-Foster): SFI’s Arguments

SFI argues it would not have been obvious to encrypt Tajima’s keys based on Foster’s KEK because:

1. A KEK would have been redundant security to Tajima’s obfuscation

2. Tajima’s obfuscation technique cannot use Foster’s synchronization technique

3. Tajima and Foster are in different fields and address different problems

4. Tajima and Foster are each missing a claimed feature: Tajima doesn’t disclose attaching a key to a “data chunk”; Foster doesn’t disclose “data chunks”

’116 POR, 9-28
’116 Sur-Reply, 2-9

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 2: SFI’s Arguments Fail

1. A KEK provides additional, independent security over Tajima’s obfuscation

2. Tajima-Foster does not use or require Foster’s synchronization

3. Tajima and Foster are in the same field of securing data, but even if they were in different fields, a POSA had motivation to encrypt keys with a KEK

4. The combination attaches encryption keys to data chunks and encrypts Tajima’s data, not Foster’s data

’116 Pet., 36-40
’116 Reply, 2-9

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 2: SFI’s Argument #1 (“redundancy”) fails because POSAs would have recognized that a KEK would provide additional, independent security to Tajima’s encryption keys

SFI’s POR

(ii) Combining Tajima and Foster Renders the Key-Securing Method of Tajima Redundant.

A POSITA would not be motivated to combine Foster to Tajima for an additional reason: Tajima’s key information is already secured by hiding it—by attaching it to a data fragment it cannot decrypt, and storing the different data fragment/key information combinations in different locations. *See, e.g.*, EX1005 at

’116 Paper 17 at 17

Reply

Tajima-Foster increases security in this way. Requiring an attacker to obtain KEKs to decrypt Tajima’s encryption keys ***in addition to*** de-obfuscating them is not “redundant” to only having to de-obfuscate them. Bhattacharjee-Reply, ¶¶41-44.

SFI’s redundancy argument (POR, 24 citing EX2033, ¶75) never refutes that applying Foster’s key-encryption to Tajima’s keys provided more security.

’116 Paper 21 at 16

SFI’s Dr. Rubin

48. The inventions-at-issue concern cryptography, which is a technologically-involved and complex field involving incredibly high stakes and safeguarding against a wide variety of highly sophisticated and ever-evolving cybersecurity attacks. Thus, a POSITA would be hesitant to modify existing practices in favor of untested techniques with no history of peer review.

’116 EX2033, ¶48 (cited ’116 POR, 6)

SFI’s Sur-Reply

EX1005 [Tajima], [0048]. Tajima recognized that storing the information in different, separately controlled locations meant that an attacker would be very unlikely to get access to every portion of the personal information because it is unlikely the same security flaw would exist at two locations separately controlled by different entities, or that an insider at one location could get access to a data portion stored in a database controlled by another entity. *See id.*, [0044]-[0046] (describing the improvement). As Petitioner’s expert, Dr. Bhattacharjee, explained at his deposition:

> The idea is that if data is stored in different places, then ideally, if one site compromised, that does not automatically, you know, lead to a compromise at another site. One way to potentially decrease that chance is to make them have different control. Potentially they will use different methods, different security protocols and that could reduce the chance of one compromise spilling over to another.

EX2046 [Bhattacharjee-2nd-Depo.], 91:20-92:3; *see also id.*, 58:23-59:5.

Thus Tajima’s system offers greater security over systems that use key encryption, to avoid the only situation Petitioner has identified in which it claims Foster’s KEK technique might “provide more security.” Tajima discloses an improvement specifically to overcome that problem, by preventing an attacker from obtaining all of the data portions. To add another layer of KEK encryption on top of Tajima’s invention, when Tajima avoids the need for KEKs, is the very definition of a “redundant” combination.

’116 Paper 29 at 7-8

Q. So using non-cryptographic protection could be useful when you're combining it with cryptographic protection?

A. Everything is layers in security. **So you have a layer of protection, which is the encryption, and you can add a layer of protection, which is obfuscation** of putting things in separate.

And we say that in security you want to provide defense in depth, meaning the more layers of security you have, the better.

’116 EX1099, 145-146

’116 Ground 2: SFI’s Argument #1 (“redundancy”) fails because even Tajima recognizes (and POSAs knew) that it is possible an attacker obtains all shares

SFI’s Sur-Reply

by storing each data portion in a different, separately controlled location. This is critical to Tajima’s invention. As Tajima explains, in “the present invention”:

> because personal information that is registered with areas that are connectible to the Internet is divided into a plurality of data portions and then registered with areas that are each under separate control, the personal information cannot be viewed unless all of the areas in which data are registered are exposed, and an improvement can therefore be obtained in the security of personal information that is registered with areas that are connectible to the Internet.

EX1005 [Tajima], [0048]. Tajima recognized that storing the information in different, separately controlled locations meant that an attacker would be very unlikely to get access to every portion of the personal information because it is unlikely the same security flaw would exist at two locations separately controlled by different entities, or that an insider at one location could get access to a data portion stored in a database controlled by another entity. *See id.*, [0044]-[0046] (describing the improvement). As Petitioner’s expert, Dr. Bhattacharjee, explained

“While Runte does disclose that its table top gaming device already provides comfortable seating for its players, it does not discourage the provision of additional legroom that an LCD or plasma display may provide. Thus, we see no error in the Board's determination that it would have been obvious … to replace Runte's CRT with an LCD or plasma display.’”

Cole Kepro Int’l, LLC v. VSR Indus., Inc., 695 F. App’x 566, 570 (Fed. Cir. 2017) (cited ’116 Reply, 17)

Tajima

SUMMARY OF THE INVENTION

[0044] It is an object of the present invention to provide a personal information management system, a personal information management method, and a server that can improve the security of personal information that is registered with areas that are connectible to the Internet.

[0045] In the present invention, when a user registers personal information with an area that is connectible to the Internet, the personal information that is to be registered and a public key with a public key system are submitted to an authentication means or a service provider. The authentication means checks whether the submitted personal information is correct or not, and if the personal information is determined to be correct, the personal information and the public key are certified to be the user’s.

[0046] The authentication means or service provider divides the submitted personal information into a plurality of data portions, registers at least one of the plurality of data portions with a database that is provided in the authentication means or service provider, and registers the other data portions with other areas that are connectible to the Internet and that are under control that is separate from the authentication means or service provider. Here, the authentication means or service provider either saves link information that indicates the registration destinations of the other data portions or attaches link information to data portions that are registered with the database that is provided in the authentication means or service provider.

[0048] Thus, because personal information that is registered with areas that are connectible to the Internet is divided into a plurality of data portions and then registered with areas that are each under separate control, the personal information cannot be viewed unless all of the areas in which data are registered are exposed, and an improvement can therefore be obtained in the security of personal information that is registered with areas that are connectible to the Internet.

’116 EX1005, [0044]-[0046], [0047] (cited ’116 Sur-Reply, 8-9)

[0042] However, if the above-described personal information relating to a user is registered with one area that is connectible to the Internet, there is the danger that, because the information registered with one area as a contiguous data file, this information may be viewed by unauthorized access from the outside through a security hole or by unauthorized access by someone inside the system, even though the information is registered with an area that is closed to the outside by means of the access control function of a server or a firewall.

’116 EX1005, [0042] (cited ’116 Pet., 34-35)

authority 40*b* or certificate authority 140*a*. However, it is also possible for the personal information to be divided among three or more data fragments and for each of the data fragments to be registered with different certificate authorities or service providers and then later combined.

’116 EX1005, [0156] (cited ’116 Pet., 39)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 2: SFI’s Argument #1 (“redundancy”) fails because even Tajima recognizes (and POSAs knew) that it is possible an attacker obtains all shares

’116 Petition

But Tajima then discloses further levels of security (e.g., 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]. This demonstrates to POSAs that Tajima recognized the possibility of failure of only storing its data fragments on different databases, such that additional techniques would be desirable to increase security. Bhattacharjee, ¶148. POSAs would have been motivated to apply known key-encryption techniques to Tajima so that an attacker that has acquired all the data fragments would not have all the information needed to decrypt the individual data fragments. This multi-level protection of Tajima’s keys is consistent with Tajima’s teaching to employ multiple layers of security in protecting users’ personal information and such “belt-and-suspenders” approaches to security were well-known. Bhattacharjee, ¶¶148-149 (citing EX1016, 3:12-60, 4:21-27; EX1036, [0003]-[0009], [0053]-[0054]; EX1026, 10:7-28 for corroboration).

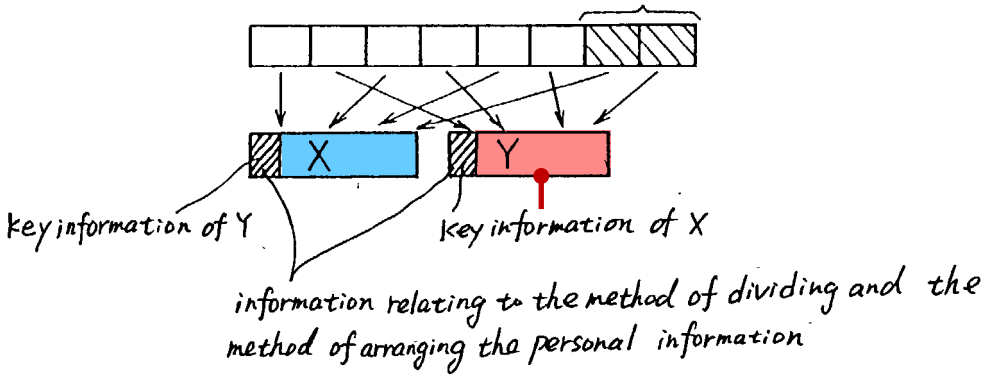
’116 Paper 2 at 35

Tajima

[0072] If it is confirmed that the personal information received from the user is correct in Step **S4**, certificate authority **40***a* divides, by means of a prescribed dividing method, the personal information that have been received from the user into two data fragments in Step **S5**. The division of the personal information is implemented such that each data fragment is completely unintelligible when taken independently. The details of the dividing method will be described hereinbelow.

’116 EX1005, [0072] (cited ’116 Pet., 8, 15, 16, 34)

Fig. 13



’116 EX1005, FIG. 13 (cited ’116 Pet., 19-21)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 2: SFI’s Argument #2 (“synchronization”) fails because the Tajima-Foster combination does not use Foster’s synchronization

Reply

SFI’s argument (2) above fails because the claimed *data chunks* are met in Tajima-Foster by ***Tajima’s data fragments*** (not Foster’s packets), which SFI does not dispute are *data chunks*. Petition, 37-38, 40-41. Allegedly “significant” differences between Foster’s packets and *data chunks* (POR, 20-23) are irrelevant in Tajima-Foster. SFI never explains how any alleged differences between Foster’s “packet payloads,” Tajima’s data fragments, and the claimed *data chunks* (POR, 20-21) would have dissuaded POSAs from using Foster’s well-known KEK technique to encrypt Tajima’s encryption keys. Those alleged distinctions are irrelevant because POSAs knew the enhanced security provided by a KEK is not dependent on the content of data being secured. Moreover, Foster’s KEK encrypts ***encryption keys***, not any data SFI alleges have “significant” differences. Bhattacharjee-Reply, ¶39; Rubin-Depo., 22-25, 154-156.

’116 Paper 21 at 15

SFI’s Sur-Reply

Here, a POSITA would not recognize that the KEK technique of Foster could be used to improve Tajima, much less to improve it “in the same way.” Foster’s KEK technique does not improve data security as compared to the prior art, which also used encrypted packets. *See* EX1006 [Foster], [0004]-[0005]; POR, 12-13. Foster’s technique simply allows synchronous transmission of a key with the content it can decrypt, in order to reduce the delays associated with prior art media technologies. EX1006 [Foster], [0063]. Such delays are not present, and need not be avoided, in Tajima.

In fact, Foster discourages the use of its KEK technique with Tajima, by stressing the importance of keeping the title key with the content it encrypts. POR, 27. The only reason for using the KEK in Foster is ***because*** the title key is kept with the content it decrypts; the KEK provides security because it prevents someone with access to all of the content from decrypting it. *See* POR, 15-16;

’116 Paper 29 at 5



“A conclusion of **obviousness ‘cannot be overcome by attacking references individually** where the rejection is based upon the teachings of a combination of references.’”

In re CSP Techs., Inc., 840 F. App'x 577, 585 (Fed. Cir. 2021) (quoting *Bradium Techs. LLC v. Iancu*, 923 F.3d 1032, 1050 (Fed. Cir. 2019)) (cited ’116 Reply, 14)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 2: SFI’s Argument #3 (“different fields”) fails because the known benefit of using a KEK is independent of Tajima’s and Foster’s fields of invention

Dr. Bhattacharjee

36. I have been informed that, in any event, obvious combinations are not limited to references in the same field, addressing the same problems. Nonetheless, both Tajima and Foster are in the same field of securing data and both encrypt different pieces of data with different encryption keys to do so. Tajima uses the known technique of obfuscating its encryption keys and Foster uses the known technique of encrypting the encryption keys with a key-encrypting key to allow storing keys with the encrypted data. The Tajima-Foster combination straightforwardly uses “Foster’s key-encryption technique” to “encrypt Tajima’s keys before attaching them to different encrypted fragments to benefit from the increased level of security that would prevent an attacker who successfully acquired all of Tajima’s data fragment[s] from possessing the plaintext key information need to decrypt the data fragments” (*see* First Declaration, ¶154)—i.e., Tajima-Foster uses Foster’s known KEK technique to address the same known security risk of storing plaintext keys with encrypted data that Foster addresses, thus increasing the security of Tajima’s system.

’116 EX1111, ¶36 (cited ’116 Reply, 13)

Wyers v. Master Lock Co.

“The Supreme Court's decision in *KSR* … directs us … that **‘familiar items may have obvious uses beyond their primary purposes**, and a person of ordinary skill often will be able to fit the teachings of multiple patents together like pieces of a puzzle.’”

616 F.3d 1231, 1238 (Fed. Cir. 2010) (quoting *KSR*, 550 U.S. 398, 402 (2007)) (cited ’116 Reply, 13)

KSR Int’l Co. v. Teleflex

“When a work is available in one field of endeavor, design incentives and other market forces can **prompt variations of it, either in the same field or a different one**. If a person of ordinary skill can implement a predictable variation, §103 likely bars its patentability.”

550 U.S. 398, 417 (2007) (cited ’116 Reply, 13)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 2: SFI’s Argument #3 (“different fields”) also fails because Tajima and Foster are both in the field of “securing data”

SFI’s POR

As Dr. Rubin explains, Tajima and Foster are directed to entirely different fields and to different problems, and in the normal course of research and development would not have considered combining the two references:

’116 Paper 17 at 9

SFI’s Sur-Reply

and EX1006 [Foster], [0003]).[1] This is incorrect; Tajima relates methods and systems for improving the security of personal information in the on-line shopping context, while Foster addresses media content but does not purport to improve the security of such content as compared to the prior art. Foster teaches, not improved data security, but rather a way of avoiding the “inherent latencies (*e.g.* delays)” in providing random access to content associated with prior art systems that lacked synchronized transmission of keys and content in the streaming context. EX1006 [Foster], [0063], Fig. 8; POR, 13-15.

’116 Paper 29 at 4

Tajima

[0044] It is an object of the present invention to provide a personal information management system, a personal information management method, and a server that can improve the security of personal information that is registered with areas that are connectible to the Internet.

’116 EX1005, [0044] (cited ‘116 Pet., 14, 19; ’116 Reply, 13)

Foster

[0003] 2. Background Art

[0004] In the transmission of digital signals, content such as video and audio data is often transmitted from a signal source to a receiver. In transmitting such content, however, the security thereof must be ensured. In general, security of the content is provided by encrypting the content with a key, and then transmitting the encrypted content to the receiver.

’116 EX1006, [0003]-[0004] (cited ’116 Pet., 52-53)

Dr. Rubin

57. Foster, by contrast, is directed towards a system wherein a content owner (*e.g.*, a movie studio) provides content to Foster’s system for encryption so that the content can be streamed (*e.g.*, via the internet) to a recipient.

’116 EX2033, ¶57 (EX2033 cited ’116 POR)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 2: SFI’s Argument #4 (“missing features”) fails because the Tajima-Foster combination encrypts Tajima’s data (not Foster’s), and uses a conventional KEK to encrypt Tajima’s keys

Reply

SFI’s argument (1) above fails if the Board agrees that Tajima’s “key information” includes Tajima’s keys as discussed *supra* §I.A. But even if the Board disagrees, SFI’s argument still fails because the Tajima-Foster combination uses “***Foster’s key-encryption*** technique to encrypt ***Tajima’s keys*** before ***attaching them*** to different encrypted fragments” (Petition, 37), which the Board recognized at institution. DI, 18-19 (“Foster’s key-encryption technique…involves transmitting and storing the encrypted key itself”); *see also* Petition, 38-42 (Tajima-Foster ***combination*** stores encrypted keys with the data); Bhattacharjee-Reply, ¶¶37-38.

SFI’s argument (2) above fails because the claimed *data chunks* are met in Tajima-Foster by ***Tajima’s data fragments*** (not Foster’s packets), which SFI does not dispute are *data chunks*. Petition, 37-38, 40-41. Allegedly “significant” differences between Foster’s packets and *data chunks* (POR, 20-23) are irrelevant in Tajima-Foster. SFI never explains how any alleged differences between Foster’s “packet payloads,” Tajima’s data fragments, and the claimed *data chunks* (POR, 20-21) would have dissuaded POSAs from using Foster’s well-known KEK technique to encrypt Tajima’s encryption keys. Those alleged distinctions are irrelevant because POSAs knew the enhanced security provided by a KEK is not dependent on the content of data being secured. Moreover, Foster’s KEK encrypts ***encryption keys***, not any data SFI alleges have “significant” differences. Bhattacharjee-Reply, ¶39; Rubin-Depo., 22-25, 154-156.

’116 Paper 21 at 14-15

SFI’s Sur-Reply

DOES NOT DISPUTE

In fact, Foster discourages the use of its KEK technique with Tajima, by stressing the importance of keeping the title key with the content it encrypts. POR, 27. The only reason for using the KEK in Foster is ***because*** the title key is kept with the content it decrypts; the KEK provides security because it prevents someone with access to all of the content from decrypting it. *See* POR, 15-16;

’116 Paper 29 at 5

a USB drive) provides “nearly perfect security.”); *see also infra* at 6-8. The reason Foster uses its KEK technique is simply absent from Tajima and a POSITA would not have considered combining them in the normal course of research and development. *See* POR, 26-27.

’116 Paper 29 at 6

Agenda

1. Tajima-based Grounds: ’116 Grounds 1-2 (independent claim 1 + claims 5-8)
2. **Foster-Hayhurst Grounds: ’116 Ground 5 (independent claims 1 & 9 + claims 2, 5, 11)**
   ’854 Ground 2 (all challenged claims 1-10)
3. Foster Ground: ’609 Ground 2 (all challenged claims 1-10)
4. Orsini Grounds: ’854 Ground 1 (all challenged claims 1-10)
   ’609 Ground 1 (all challenged claims 1-10)
5. Tajima-Rivest-based Grounds: ’116 Ground 3 (independent claim 9 + claims 10-14)
   ’116 Ground 4 (independent claim 1 + claims 2-3, 6)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Overview of the ’116 & ’854 Petitions’ Grounds

’116 Patent

GROUND	CLAIMS	BASIS
1. Tajima (EX1105)	1, 5-8	Obviousness/Anticipation
2. Tajima, Foster (EX1106)	1-2, 9, 11-14	Obviousness
3. Tajima, Rivest (EX1034)	9-14	Obviousness
4. Tajima, Rivest, Shyu (EX1016)	1-3, 6	Obviousness
5. Foster, Hayhurst (EX1007)	**1-2, 5, 9, 11**	**Obviousness**

’854 Patent

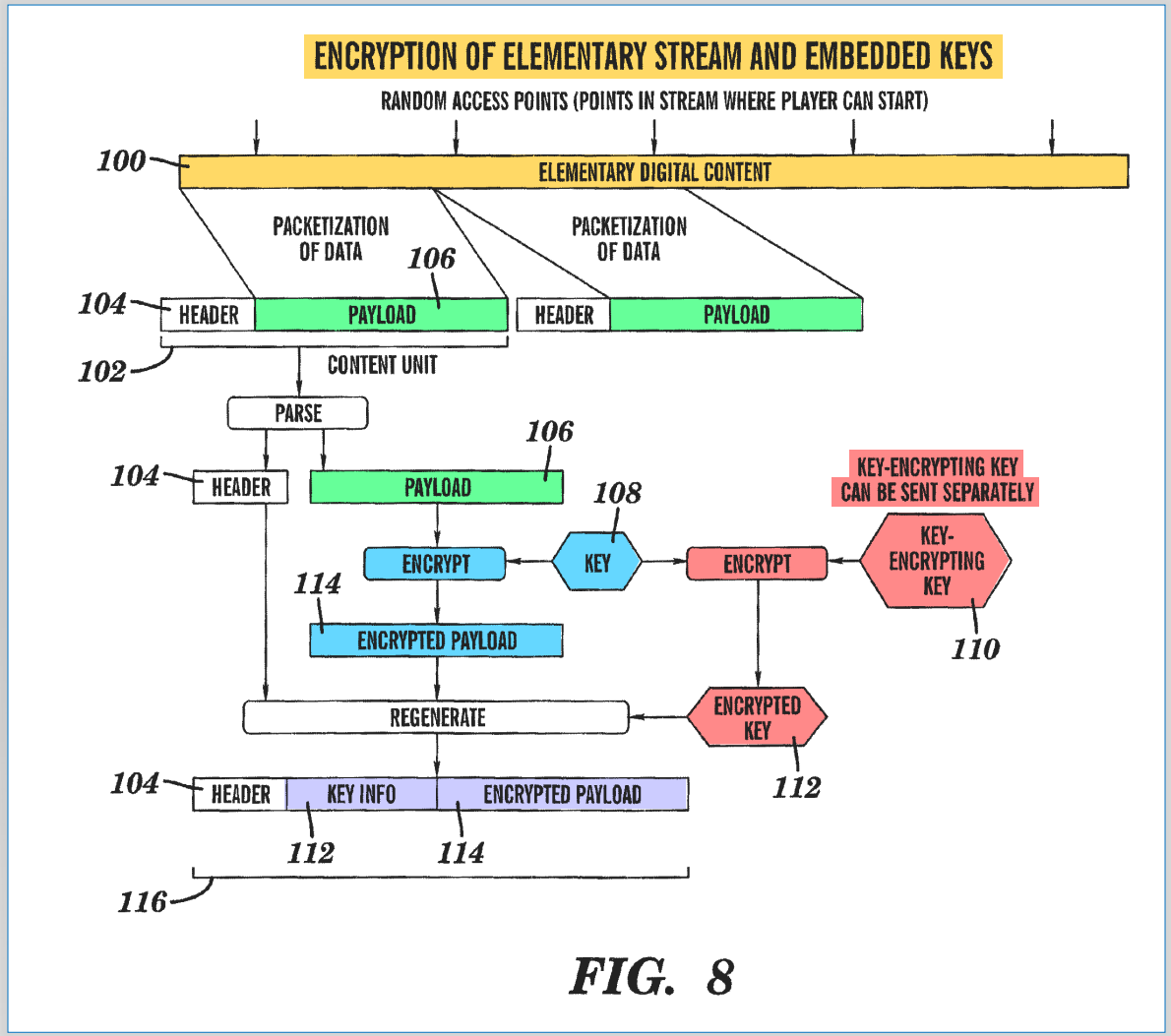
GROUND	CLAIMS	BASIS
1. Orsini	1-10	Obviousness/Anticipation
2. Foster, Hayhurst (EX1007)	**1-10**	**Obviousness**

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 5 (claim 1): Foster’s encryption process

	’116 Patent, Claim 1	Foster
[1PRE]	A method for securing a data set, the method comprising:	[0064] ... [A]n elementary content stream 100 comprising of compressed content is parceled into content units 102. Each content unit includes a header 104 and a content packet 106.… Once parceled, header 104 of each content unit 102 is parsed from content packet 106 and content packet 106 is encrypted with title key 108 to yield encrypted content packet 114. Then, key encrypting key 110 is used to encrypt title key 108, which is then attached to content packet 106 along with header 104. The resulting processed content unit ’116 can then be transmitted to a recipient. That is, the encrypted content unit, the header and the header extension (including the encrypted title key) will be delivered to, or stored by, a recipient as a single, integral data stream. Thus, a recipient will receive or store encrypted content along will all information necessary to decrypt the content.
[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.	[0065] … [E]ach content packet could be encrypted with a different title key.

Foster



’116 EX1006, FIG. 8 (annotated) (cited ’116 Pet., 51-54, 67-74)

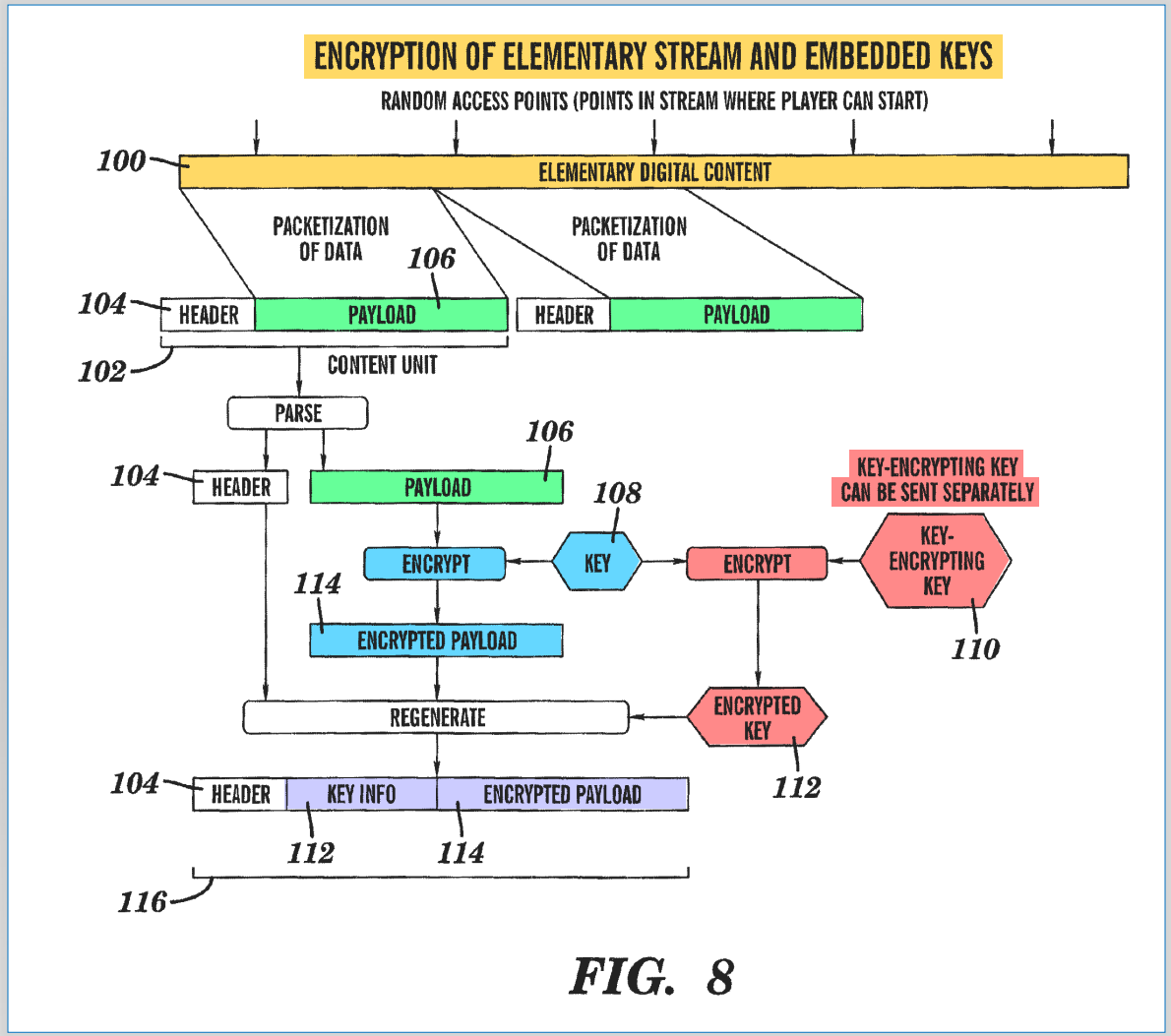
’116 EX1006, [0064]-[0065] (cited ’116 Pet., 51-54, 67-74)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 5 (claim 9): Foster’s encryption process

’116 Patent, Claim 9	
[1PRE]	1. A method for securing a data set, the method comprising:
[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]	transforming 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 transformed different encryption keys on a plurality of different storage devices.

Foster



’116 EX1006, FIG. 8 (annotated) (cited ’116 Pet., 51-54, 67-77)

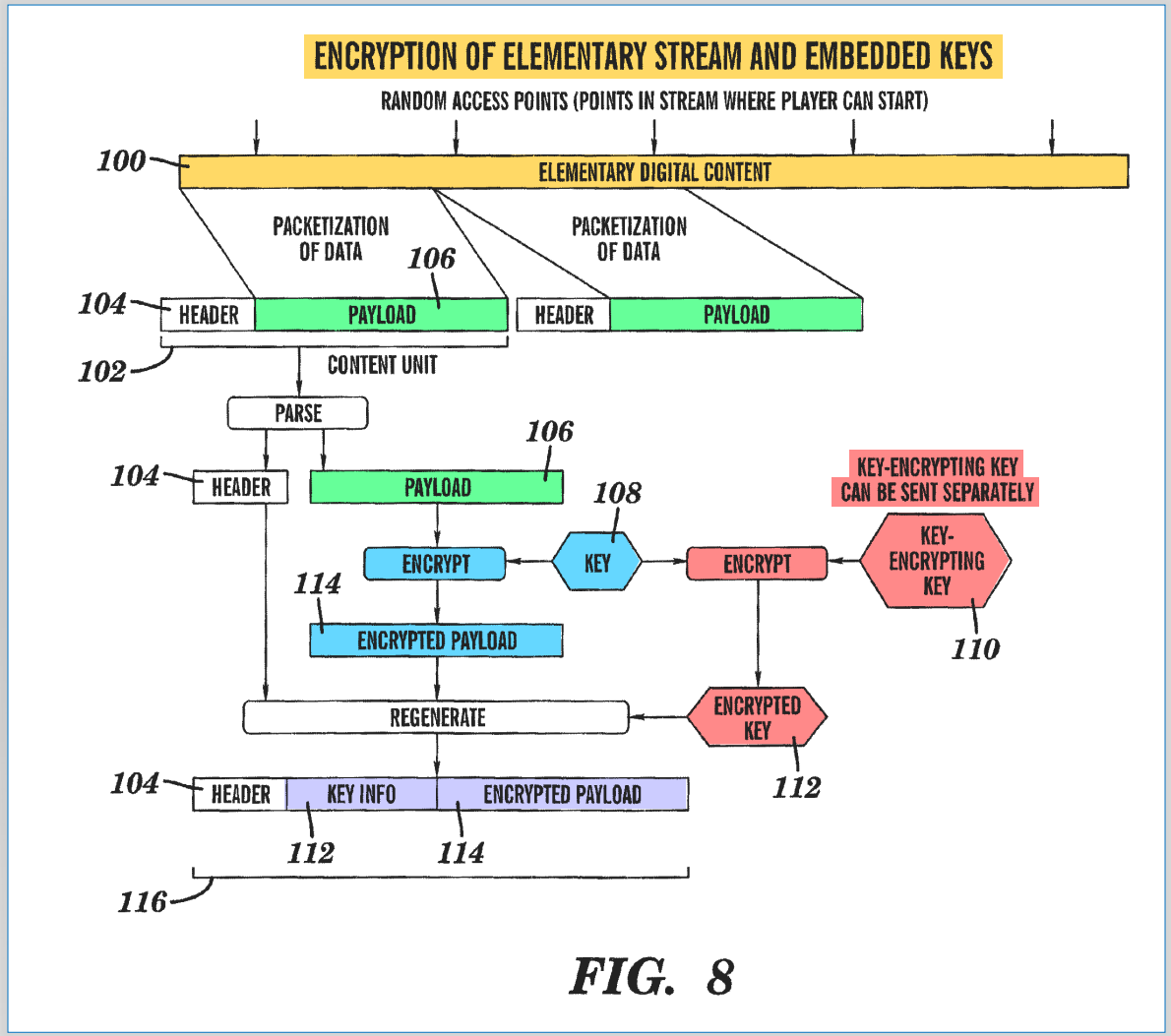
See ’116 Pet., 75-77

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’854 Ground 2 (claim 1): Foster’s encryption process

’854 Patent, Claim 1	
[1PRE]	1. A method for securely storing a data set, the method comprising:
[1A]	receiving an external key from an external storage system;
[1B]	generating a plurality of data chunks based on the data set, such that the data set can be reconstructed using at least a minimum number of the plurality of data chunks, wherein generating the data chunks comprises:
[1B1]	distributing the data set into a plurality of shares, wherein each of the shares comprises less than all of the data set,
[1B2]	accessing a plurality of distinct encryption keys,
[1B3]	encrypting each of the shares with a respective one of the plurality of distinct encryption keys,
[1D]	performing an encryption operation based on the external key to further secure the plurality of data chunks; and
[1E]	storing with the plurality of data chunks data indicative of at least one of the distinct encryption keys on a plurality of different storage devices.

Foster



’854 EX1006, FIG. 8 (annotated) (cited ’854 Pet., 56)

See ’854 Pet., 56-59

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116/’854 Foster-Hayhurst Grounds: Foster’s “synchronization” and “random access”

Foster

[0070] Referring now to **FIG. 12**, the receipt and processing of content units **102** and **202** under the present invention are shown in greater detail. As depicted, content units **102** and **202**, including headers **104** and **204**, header extensions **118** and **218** and content packets **106** and **206** are received. Encrypted keys **112** and **212** are recovered from header extensions **118** and **218**, respectively. Key encrypting key **110** is then used to decrypt both encrypted keys **112** and **212** to yield title keys **108** and **208**. Once recovered, title keys **108** and **208** are used to decrypt the corresponding content packets **106** and **206** to yield content **140** and **240**.

[0071] Because the title keys **108** and **208** are attached and transmitted with the content packets they were used to encrypt, synchronization under the present invention is inherent. In contrast, related systems generally rely on separate transmissions and/or bit flags for synchronization. Moreover, synchronized transmission and delivery of title keys with encrypted content allows random access of the content. Specifically, because each content packet is accompanied with the correct title key, the content stream can be accessed at any packet. Related systems fail to provide such a feature. 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 is was attached. Also, because the title key and encrypted content are transmitted as a single stream, the present invention is compatible and compliant with current front-end and back-end standards.

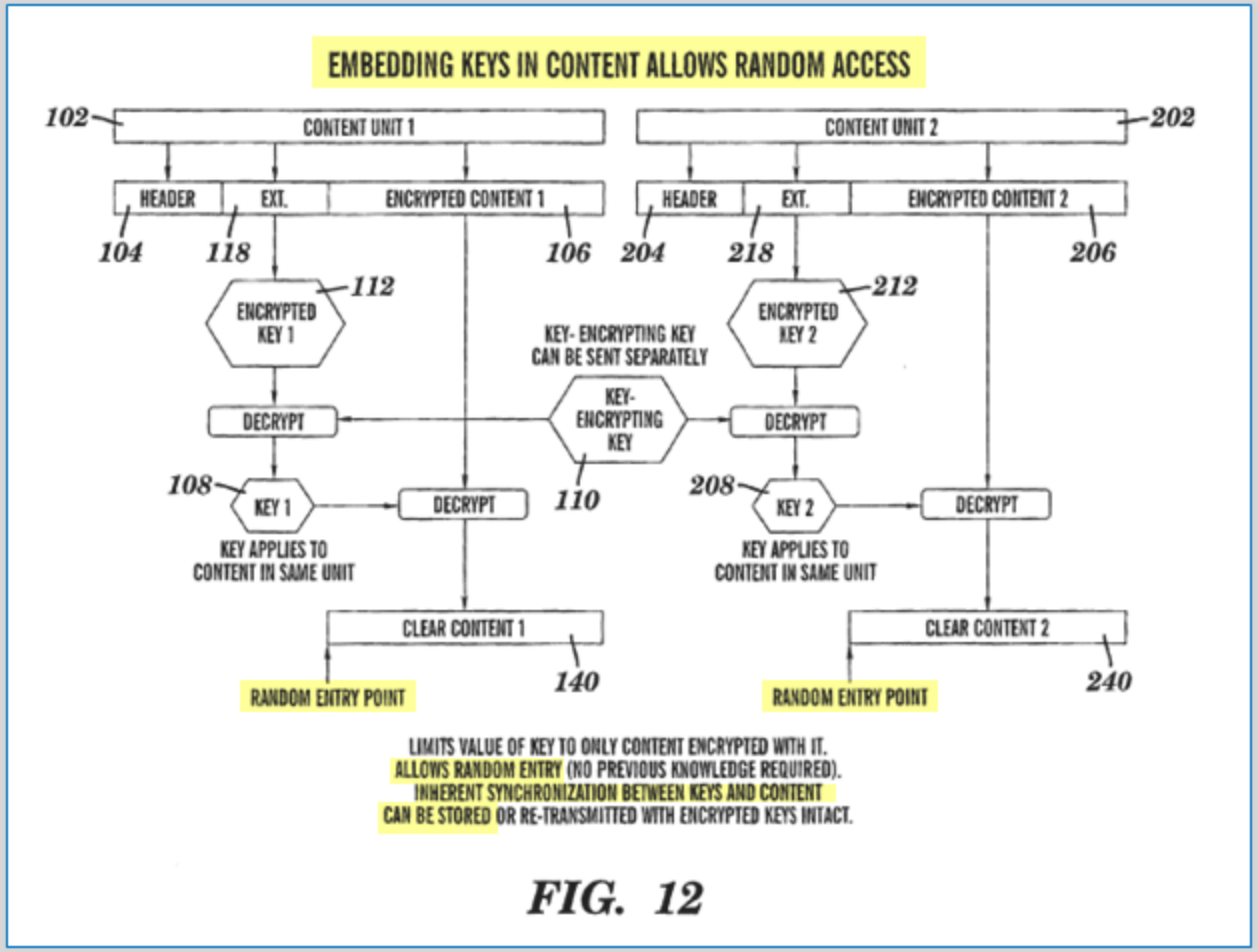


FIG. 12

EX1006, [0070]-[0071], FIG. 12
(cited ’854 Pet., 73-74; ’609 Pet., 66-67, 69, 73; ’116 Pet., 36-37, 51-52
’609 Reply, 5; ’854 Reply, 15; ’116 Reply, 30-31)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116/’854 Foster-Hayhurst: Hayhurst’s distributed storage technique

Hayhurst

TECHNICAL FIELD

[0002] This application relates to the field of electronic media delivery, such as video-on-demand, including the secure handling and storage of such media.

SUMMARY OF THE INVENTION

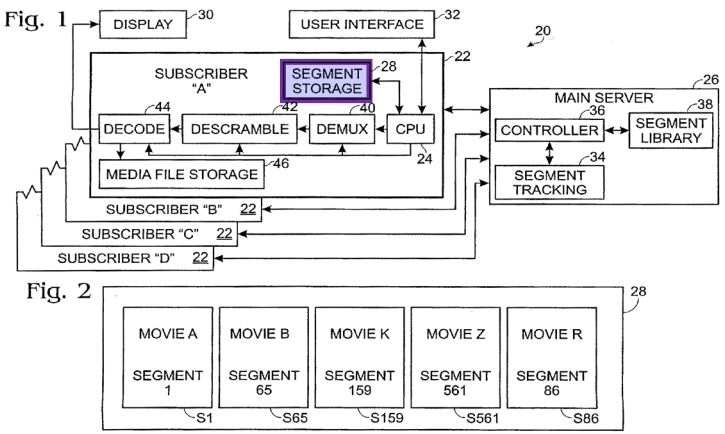
[0005] The present invention addresses the problem of prohibitive bandwidth requirements for a content provider in delivering video-on-demand or other large media files. In short, there is provided a system that shifts the bandwidth requirements away from the content provider and to the network of subscriber units.

[0006] In one embodiment, a media file representing, for example, a movie is segmented. These segments are distributed among the network of subscriber units. Any one of the subscribers may submit a request to view a movie. In response to such a request, the other subscriber units direct to the requesting subscriber all of the segments, which are then assembled into the complete media file (movie) for display, or stored for later display, on the requesting subscriber’s unit.

[0008] The segmented and distributed storage of the media file provides a high level of security and a deterrence to piracy. No single subscriber unit has more than a segment of a media file, which segment, by itself, has little value.

[0021] **FIG. 3** depicts an alternative way of segmenting a media file (movie) for storage in the segment storage **28** of a subscriber unit. Here, several separate segments **S1**, **S5**, **S9**, and **S13** of a movie (“Movie A”) are bundled together for storage. The segments are separate in the sense that they represent discontinuous or non-sequential portions of the movie and must be soiled and integrated with the remaining segments (distributed elsewhere in the network) to provide the complete movie. This approach to segmenting the movie provides increased security from piracy since each subscriber A, B, C, etc, will store only very small portions of the movie on the storage device of the unit. If the segments are made very small, to the scale of a few bytes or single bits, decoding of any individual bundle will be virtually impossible. Further, access to the correct method for recombining the segments to recreate the original media file could be stored in a password-protected file, linked to an encryption key, or locked to all devices except the specific hardware device that created the bundled segments.

[0027] As noted, the main server **26**, via the controller **36** and segment-tracking module **34**, distribute **102** the movie segments (including copies of segments to establish the redundancy described above) to all of the participating subscriber units **22**. Such distribution and storage of the segments on the units may be performed in a manner that is automatic and transparent to the users of the units.



EX1007, [0002], [0005], [0006], [0008], [0021], [0027], Fig. 1, (cited ’854 Pet., 59-61; ’116 Pet., 54-58, 61-66)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116/’854 Foster-Hayhurst: POSAs had motivation to store and deliver Foster’s encrypted content using Hayhurst’s distributed storage devices

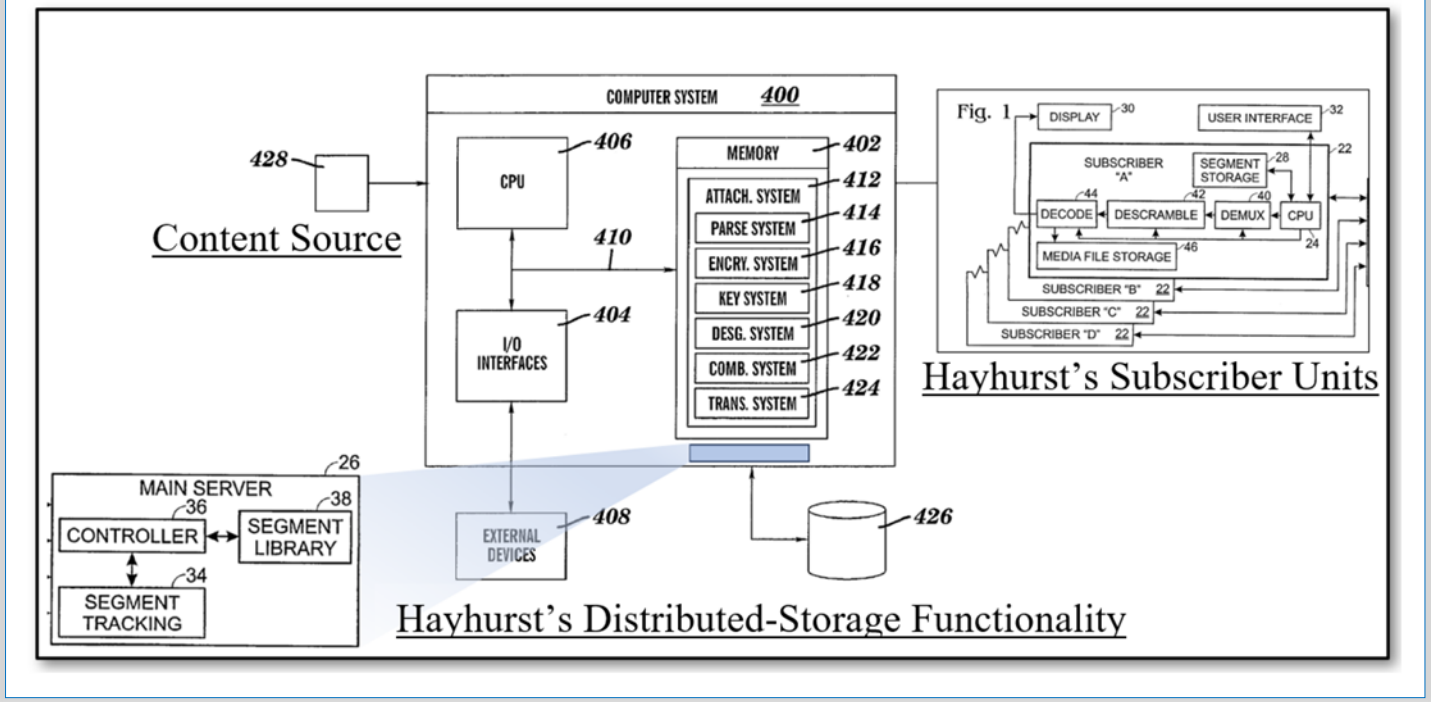
’116 Petition

¶240. 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. *Id.*

First, Foster’s content provider is responsible for delivering content to consumer devices. Bhattacharjee, ¶222-224, 241; §IX.A. Hayhurst teaches that such systems can be improved for the content provider by storing media-file segments (like Foster’s content units) on multiple subscriber units (like Foster’s receivers) because Hayhurst’s technique: (1) solves problems with on-demand delivery of a media 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). Bhattacharjee, ¶¶241-242; §IX.B.1 *supra*. Thus, Hayhurst explicitly motivates a POSA to apply Hayhurst’s distributed-storage technique to Foster’s system to improve content delivery in multiple ways. Bhattacharjee, ¶¶241-242.

’116 Paper 2 at 62; ’854 Pet., 61-65

Foster-Hayhurst Combination



’116 Paper 2 at 64; ’854 Pet., 65

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116/’854 Foster-Hayhurst: The Parties’ disputes are narrow

- No dispute that Foster-Hayhurst meets all but “data chunks” limitations (’116 limitations [1A] and [9A]; ’854 limitations [1B]-[1B1])

- **Only disputes are whether:**
  (1) Hayhurst is analogous art to the ’116 and ’854 patents;
  (2) POSAs had motivation to combine Foster & Hayhurst; and
  (3) Foster discloses “data chunks” / “shares”*

*** SFI raises the same dispute for ’609 Ground 2 (Foster)**

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116/’854 Foster-Hayhurst: Analogous art standard

In re Wood

“The determination that a reference is from a nonanalogous art is therefore two-fold. First, we decide if the reference is within the field of the inventor's endeavor. If it is not, we proceed to determine whether the reference is reasonably pertinent to the particular problem with which the inventor was involved.’”

In re Wood, 559 F.2d 1032, 1036 (CCPA 1979) (cited ’854 Reply, 2; ’116 Reply, 22)

Wyers v. Master Lock Co.

“*KSR International Co. v. Teleflex, Inc.*, [citation omitted], directs us to construe the scope of analogous art broadly, stating that ‘*familiar items may have obvious uses beyond their primary purposes*, and a person of ordinary skill often will be able to fit the teachings of multiple patents together like pieces of a puzzle.’”

Wyers v. Master Lock Co., 616 F.3d 1231, 1238 (Fed. Cir. 2010) (italics original) (cited ’854 Reply, 4; ’116 Reply, 22)

Netflix, Inc. v. DivX, LLC

“The field of endeavor is ‘not limited to the specific point of novelty, the narrowest possible conception of the field, or the particular focus within a given field. [citation omitted] (stating analogous art test merely gauges whether prior art is “too remote” to be treated as analogous art) [citation omitted] (“[C]ourts generally take an expansive view of what constitutes analogous art.”) [citation omitted].… [T]he field-of-endeavor test does not look to the problem that the patent purports to address. [citation omitted]. It is enough that the prior art reference falls within the relevant field of endeavor of the patent-in-suit. We have affirmed findings of analogous art where the references shared a general field of endeavor.”

Netflix, Inc. v. DivX, LLC, 80 F.4th 1352, 1359 (Fed. Cir. 2023) (cited ’854 Reply, 4; ’116 Reply, 22)

’116/’854 Foster-Hayhurst: The Petitions did not have to anticipate a dispute over analogous art; SFI cites no law to the contrary

SFI’s POR

A. Petitioner Fails To Demonstrate That Hayhurst Is Analogous Art.

The Foster/Hayhurst ground fails because Petitioner fails to prove Hayhurst is analogous art. Petitioner’s combination relies upon “Hayhurst’s techniques for

’854 Paper 21 at 32-33; ’116 POR. 38

SFI’s Sur-Reply

II. FOSTER-HAYHURST FAILS (GROUND 2, ALL CLAIMS).

A. Petitioner Fails To Prove Hayhurst Is Analogous Art (Ground 2, All Claims).

Petitioner has the burden of proving Hayhurst is analogous art and did not attempt to carry that burden in the Petition. POR, 32-41. The Reply fares no

’854 Paper 32 at 2; ’116 Sur-Reply, 14

Contrary to the law

’854 Reply

[2] SFI suggests the Petition had to demonstrate that Hayhurst is analogous art (POR, 33), but a “petitioner is not required to anticipate and raise analogous art arguments in its petition.” *Sanofi-Aventis Deutschland Gmbh v. Mylan Pharms. Inc.*, 66 F.4th 1373, 1379 (Fed. Cir. 2023).

’854 Paper 2 n.2

“A petitioner is not required to anticipate and raise analogous art arguments in its petition.”

Sanofi-Aventis Deutschland GmbH v. Mylan Pharms. Inc., 66 F.4th 1373, 1379 (Fed. Cir. 2023) (cited ’854 Reply, 2)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116/’854 Foster-Hayhurst: SFI’s Sur-Reply argument that Petitioner did not identify a field of endeavor is refuted by SFI’s POR

SFI’s Sur-Reply

II. FOSTER-HAYHURST FAILS (GROUND 2, ALL CLAIMS).

A. Petitioner Fails To Prove Hayhurst Is Analogous Art (Ground 2, All Claims).

Petitioner has the burden of proving Hayhurst is analogous art and did not attempt to carry that burden in the Petition. POR, 32-41. The Reply fares no better. To meet its burden, Petitioner was required to identify the field of endeavor to which the ’854 relates, and the problem or problems that Hayhurst attempted to solve. POR, 33 (citing cases). Petitioner declined to do either. Reply, 2, 5-6; Ex. 1111 [Bhattacharjee-2nd-Decl.] ¶11 (“I have not been asked to opine on whether SFI has identified the correct field of endeavor for the ’854 patent.”); Ex. 2046 [Bhattacharjee-2nd-Depo.] 102:3-103:13 (Dr. Bhattacharjee did not determine field of endeavor); *see also* Reply, 5-6 (not identifying problem Hayhurst sought to address); Ex. 1111 [Bhattacharjee-2nd-Decl.] ¶¶27-47 (same). Thus, the Reply cannot rectify Petitioner’s failure to carry its burden.

’854 Paper 32 at 2;’ 116 Sur-Reply, 14

SFI’s POR

1. Hayhurst And The ’854 Patent Do Not Share The Same Field Of Endeavor.

Petitioner makes no attempt to show that Hayhurst and the ’854 share the same field of endeavor. Nor could it. As Petitioner and Dr. Bhattacharjee admit, the ’854’s field is securing data while Hayhurst’s is decentralized electronic media delivery. Pet., 4 (“[t]he ’854 patent relates to techniques for securing electronic data from unauthorized access.”); Ex. 1003 [Bhattacharjee-Decl.] ¶ 33 (same); Pet., 59 (“Hayhurst concerns ‘***electronic media delivery, such as video-on-demand***’”); Ex. 1003 [Bhattacharjee-Decl.] ¶ 178 (same); Ex. 2034 [Bhattacharjee-Depo.] 101:19-25.

’854 Paper 21 at 32-33; *see also* ’116 POR. 39

Dr. Bhattacharjee

Q. In paragraph 11 have you identified the field of endeavor for the Hayhurst reference?

A. **Well, Hayhurst is at least in the same field of endeavor that SFI has identified for the ’854 patent.**

Q. And what field of endeavor is that?

A. **The field of securing data, including securely storing and communicating the secured data.**

EX2046, 102-103
(cited ’854 Sur-Reply, 2)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116/’854 Foster-Hayhurst: Hayhurst expressly says its field “includ[es] the secure handling and storage of” data

SFI’s POR

The ’854’s Field Is “Securing Data”

* * *

Hayhurst’s field, per these same traditional sources, is “decentralized media delivery.”

Hayhurst’s Field Is “Decentralized Media Delivery”	
Title	“**Decentralized Media Delivery**”
Abstract	“**A decentralized approach to delivering a media file** ...”
Technical Field	“This application **relates to the field of electronic media delivery** ...” *Id.*, [0002].

’854 Paper 21 at 35-36; ’116 POR, 41-42

Hayhurst

TECHNICAL FIELD

[0002] This application relates to the field of electronic media delivery, such as video-on-demand, including the secure handling and storage of such media.

* * *

[0008] The segmented and distributed storage of the media file provides a high level of security and a deterrence to piracy. No single subscriber unit has more than a segment of a media file, which segment, by itself, has little value.

* * *

the complete movie. This approach to segmenting the movie provides increased security from piracy since each subscriber A, B, C, etc, will store only very small portions of the movie on the storage device of the unit. If the segments are

’854 EX1007, [0002] (cited ’854 Pet., 59; ’116 Pet., 55-57, 62-63

SFI’s Expert Dr. Rubin

Q. Well, you said before that you split it and you stored one piece in one location and another piece in another location; is that right?

A. I said that.

Q. And that provides security by having one piece in one location and the other piece in the other location; is that right?

A. That makes it harder for an attacker to get the data.

’854 EX1099, 130 (cited ’854 Reply, 3; ’116 Reply, 8)

’854 Patent

does not necessarily compromise the sensitive data. Moreover, in the embodiment having geographical separation of the data storage facilities, a compromise of multiple geographically remote facilities becomes increasingly difficult.

’854 EX1001,19:64-67 (cited ’854 Reply, 3’ ’116 Reply, 21)

’116/’854 Foster-Hayhurst: “Securing data” is a broad field that includes Hayhurst

Dr. Bhattacharjee

24. As I discussed above, the ’854 patent and Hayhurst describe overlapping techniques for securing data, which the ’854 patent itself recognizes is a broad field:

> The secure data parser of the present invention may be advantageously used in **a number of applications and technologies**. For example, email system, RAID systems, video broadcasting systems, database systems, or **any other suitable system** may have the secure data parser integrated **at any suitable level**. As previously discussed, it will be

’854 EX1111, ¶24 (cited ’854 Reply, 5) (quoting ’854 Patent, 81:20-50); ’116 EX1111, ¶65 (cited ’116 Reply, 22)

SFI’s Expert Dr. Rubin

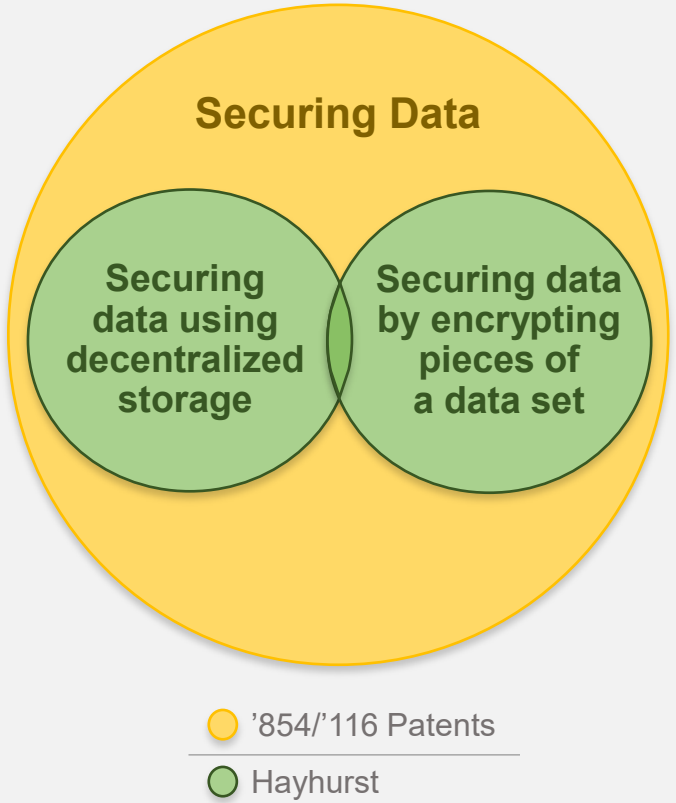
Q. Okay. And what does it mean to secure data?

A. It means to protect the data.

Q. What do you mean by “protect the data”?

A. To provide security for the data such that it isn’t available to people who are not supposed to have it and that they can't be changed.

’854 EX1099, 108-109 (cited ’854 Reply, 2)
’116 EX1099, 108-109 (cited ’116 Reply, 22)



“The field of endeavor is ‘not limited to the specific point of novelty, the narrowest possible conception of the field, or the particular focus within a given field….[T]he field-of-endeavor test does not look to the problem that the patent purports to address. [citation omitted]. It is enough that the prior art reference falls within the relevant field of endeavor of the patent-in-suit. We have affirmed findings of analogous art where the references shared a general field of endeavor.”

Netflix, Inc. v. DivX, LLC, 80 F.4th 1352, 1359 (Fed. Cir. 2023) (cited ’854 Reply, 4; ’116 Reply, 22)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116/’854 Foster-Hayhurst: Contrary to SFI, Petitioner relied on Hayhurst’s disclosure to demonstrate that Hayhurst is within the field of securing data

SFI’s Sur-Reply

Petitioner similarly ignores that each of Hayhurst’s title, abstract, technical field, background, summary, specification/figures, and claims—traditionally used to discern the field of endeavor—***all*** confirm that Hayhurst’s field is “decentralized media delivery.” *See* POR, 41-43.

Petitioner instead asserts that that “decentralized media delivery” not Hayhurst’s field, but merely “is how Hayhurst provides ‘security,’” Reply, 21, and “deterrence to piracy.” Reply, 31. But this gets Hayhurst backwards; security and deterrence to piracy is a by-product of its decentralized media delivery, which seeks to overcome bandwidth inefficiencies associated with prior art video-on-demand systems. EX1007 [Hayhurst] [0004]-[0006]. Solely as a by-product, Hayhurst suggests, the “segmented and distributed storage of the media file” can provide an additional “level of security and a deterrence to piracy.” *Id.*, [0008].

But Hayhurst’s focus is achieving a “decentralized approach to delivering a media file.” POR, 41-42 (discussing EX1007 [Hayhurst], Title, Abstract, [0002], [0003]-[0004], [0005], [0012], [0015]).

’116 Paper 29 at 15-16; ’854 Sur-Reply, 3-4

’116 Reply

1. Hayhurst Has the Same Field of Endeavor.

SFI argues “the ’116’s field is securing data,” including securely storing and communicating data. POR, 39-40. Hayhurst is in the same field.

Hayhurst’s “***Technical Field***” is “electronic media delivery…including the ***secure handling and storage*** of such media.” Hayhurst, [0002]; DI, 10. That should be dispositive that Hayhurst is analogous. *In re Wood*, 599 F.2d 1032, 1036 (C.C.P.A. 1979). Hayhurst’s specification reinforces this. Hayhurst, [0021] (“provid[ing] increased ***security*** from piracy”), [0007] (“***encrypt[ing]***” “***segments***” further “prevent[s] unauthorized viewing”); Bhattacharjee-Reply, ¶¶60-68.

Hayhurst’s field is not limited to “decentralized media delivery” (POR, 41)—that is ***how*** Hayhurst provides “***security***” of stored data. Hayhurst, [0008]. Like Hayhurst, the ’116 patent ***itself*** says it secures data via “geographical separation of…storage facilities.” EX1001, 20:28-38. Bhattacharjee-Reply, ¶69.

’116 Paper 21 at 21; ’854 Reply, 4)

SFI’s Sur-Reply

DOES NOT DISPUTE

“The field of endeavor is ‘not limited to the specific point of novelty, the narrowest possible conception of the field, or the particular focus within a given field.”

Netflix, Inc. v. DivX, LLC, 80 F.4th 1352, 1359 (Fed. Cir. 2023) (cited ’854 Reply, 4; ’116 Reply, 22)

“Asset Guard has not supported its contention that…the reference must be concerned with a problem that was of ‘central importance’ to the inventors.”

In re Asset Guard Prods., Inc., 2022 WL 1097363, at *4 (Fed. Cir. Apr. 13, 2022) (cited ’854 Reply, 5-6)

’116/’854 Foster-Hayhurst: Hayhurst is also analogous art because it is reasonably pertinent to several problems SFI’s patents are concerned with

SFI’s POR

The ’854 is directed towards creating a cryptographic system whose security is user independent, while still supporting mobile users. Ex. 1001 [’854] 2:42-44. The ’854 notes the “ever-increasing need ...

’854 Paper 21 at 38; ’116 POR, 44-45

SFI’s Sur-Reply

DOES NOT DISPUTE

And Petitioner cannot dispute Hayhurst’s plain statement that its “invention addresses ***the problem*** of prohibitive bandwidth requirements for a content provider in delivering video-on-demand or other large media files.” Ex. 1007 [Hayhurst] [0004]-[0005]; POR, 39-40; Ex. 2033 [Rubin-2nd-Decl.] ¶107. Petitioner does not dispute that Hayhurst’s limited bandwidth problem is not reasonably pertinent to the ’854’s problem of eliminating user-related vulnerabilities in storing sensitive information, nor does it contend that limited bandwidth is reasonably pertinent to any “additional” problems Petitioner purports the ’854 addresses. POR, 40-41; Ex. 2033 [Rubin-2nd-Decl.] ¶¶108-109; Reply, 5-6.

’854 Paper 32 at 4-5; ’116 Sur-Reply, 16

Reply

User-Independent Security.

* * *

POR, 38-39 (quoting EX2033, ¶106). Hayhurst is reasonably pertinent to that problem and addresses it by segmenting and distributing data without user action. Hayhurst, [0027] (cited Petition, 61, 65). As the Board correctly observed, users cannot undermine Hayhurst’s security because “no one subscriber unit has more than one media segment, and a single segment has little value.” DI, 22; Hayhurst, [0021],

* * *

Computer-Network Security.

* * *

the patent relates to, not its “central” problem). The patent identifies the problem of “ensuring [that] data stored and transmitted over” “computer networks” “cannot be read or otherwise compromised.” EX1001, 1:30-39; Rubin-Depo., 109-110. Hayhurst similarly addresses problems “relat[ing] to…**electronic** media **delivery**…including the **secure handling and storage** of” data ([0002]) and “deterr[ing] [] piracy” ([0008]). Bhattacharjee-Reply, ¶¶37-39.

* * *

Distributed Storage.

The patent splits and “stor[es]…data in geographically remote secure data storage facilities.” EX1001, 5:53-56. It does that to address

* * *

them by “splitting the data set into shares”). Hayhurst’s distributed-storage technique addresses those problems and solves them in the same way. *Supra* §I.A.1. Rubin-Depo., 124-135; Bhattacharjee-Reply, ¶¶40-47.

’854 Reply Paper 24 at 5-6; ’116 Reply, 23-24

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116/’854 Foster-Hayhurst: The Parties’ disputes are narrow

- No dispute that Foster-Hayhurst meets all but “data chunks” limitations (’116 limitations [1A] and [9A]; ’854 limitations [1B]-[1B1])

- **Only disputes are whether:**
  (1) Hayhurst is analogous art to the ’116 and ’854 patents;
  (2) POSAs had motivation to combine Foster & Hayhurst; and
  (3) Foster discloses “data chunks” / “shares”*

*** SFI raises the same dispute for ’609 Ground 2 (Foster)**

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116/’854 Foster-Hayhurst: A POSA had motivation to combine Foster’s encryption technique with Hayhurst’s distributed storage

1. POSAs would have recognized several benefits of Hayhurst to Foster’s storage and delivery of content, including adding security that Foster lacks

2. Foster is for use in a variety of systems that store encrypted content both before and after delivering to end-users

’116 Pet., 61-67; ’116 Reply, 24-29;
’854 Pet., 61-65; ’854 Reply, 6-13

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116/’854 Foster-Hayhurst: SFI’s “no motivation” argument fails because Hayhurst provides benefits to storing and delivering Foster-encrypted data that Foster lacks

’116 Petition

For example, in Foster’s embodiments in which the content service provider implements Foster’s encryption system (*supra* §IX.A), POSAs would have been motivated to configure Foster’s content 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) in improving delivery of digital content (e.g., “delivery-on-demand [of] a large number of movie selections” Hayhurst, [0004]-[0010]) and 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 single-source bottlenecks; and/or (2) efficiency by intelligently selecting storage device(s) that would provide the best delivery based on availability, location, etc. Bhattacharjee, ¶¶242-244 (*citing* as corroboration EX1028, [0022]-[0023]; EX1031, [0003], [0012]-[0013], [0015], [0018]-[0024]; EX1030, [0005]-[0010]; EX1029, 1:19-22, 2:31-38, 11:11-26, 15:39-16:-20).

’116 Paper 2 at 62-63; *see also* ’854 Pet., 62

’116 EX1031

[0018] The invention provides reliable, high throughput, and low latency data transfer by downloading data from

* * *

works or peer-to-peer networks. Some advantages that may be achieved by the invention are as follows:

[0019] Speed—It includes optimizations such as Dynamic Range Requests, Pipelining, and Range Preemption to download the data as absolutely fast as possible.

[0020] Intelligence—The engine will automatically discover and download from the most desirable mirrors or sources. It constantly monitors and responses to changing network conditions, optimizing the download on the fly.

[0021] Fault Tolerance—The majority of software is not robust in the face of network problems, they hit a snag and immediately throw an error or hang. In contrast, the invention automatically compensates for network problems when they arise. It routes around problems, retries, and will faithfully deliver the data.

[0022] Streaming—The engine was specifically designed to provide the fastest progressive downloads. Using the technology, one can actually watch a video as it is being downloaded from multiple sources in parallel. This is perfect for Video-On-Demand, P2P, and distance learning applications.

EX1031, [0018]-[0022] (cited ’116 Pet., 63; ’854 Pet., 62)

’116 EX1028

[0022] As mentioned previously one embodiment of the invention uses ‘shotgun’ downloading, where each media file desired by a user is divided into predefined segments, such as by time, and the segments are received from several different peers. In this manner, the bandwidth needed to download content in such a system is vastly reduced when compared to a traditional client/server download. For

EX1028, [0022] (cited ’116 Pet., 63; ’854 Pet., 62)

Hayhurst

[0004] It is usually not economically feasible for the operator of, for example, a cable television (CATV) system (or other content provider) to provide for delivery-on-demand a large number of movie selections to subscribers because such a system requires a extensive amount of bandwidth between the content provider and each of the subscriber units (televisions or other display devices) of the network.

Summary of the Invention

[0005] The present invention addresses the problem of prohibitive bandwidth requirements for a content provider in delivering video-on-demand or other large media files. In short, there is provided a system that shifts the bandwidth requirements away from the content provider and to the network of subscriber units.

EX1007 (cited ’116 Pet., 55-57, 62-63; ’854 Pet., 59-61)

’116/’854 Foster-Hayhurst: SFI’s “no motivation” argument fails because Hayhurst adds redundancy to storage and delivery of Foster-encrypted data, which Foster lacks

’116 Petition

First, Foster’s content provider is responsible for delivering content to consumer devices. Bhattacharjee, ¶222-224, 241; §IX.A. Hayhurst teaches that such systems can be improved for the content provider by storing media-file segments (like Foster’s content units) on multiple subscriber units (like Foster’s receivers) because Hayhurst’s technique: (1) solves problems with on-demand delivery of a media 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). Bhattacharjee, ¶¶241-242; §IX.B.1 *supra*. Thus, Hayhurst explicitly motivates a POSA to apply Hayhurst’s distributed-storage technique to Foster’s system to improve content delivery in multiple ways. Bhattacharjee, ¶¶241-242.

’116 Paper 2 at 62

’116 EX1029

This scheme protects the system against data loss by adding redundancy to the original data. The basic idea of the scheme is to fragment a stored object (e.g., file) into blocks and distribute the blocks across available servers divided among servers that hold original data and servers that hold redundant data. If the total number of failed servers is less than the number of servers holding the redundant data, the actual data can be restored. However, the way these blocks are distributed over the different servers is left open. It is the responsibility of the application to choose a proper data layout to maximize parallelism.

EX1029, 2:31-41
(cited ’116 Pet., 63; ’854 Pet., 62)

SFI’s POR

discussed previously. Further, Foster’s system has no need for redundancy as Foster’s system can simply resend dropped packets, nor any need for efficiency/intelligence when sending on a one-to-one basis.

Moreover, any benefits are easily outweighed by the drawbacks. Again, Foster is directed towards reducing delays, particularly delays frustrating random access to content. The Foster/Hayhurst combination, however, would introduce enormous delays preventing random access to content and rendering Foster’s system unsuitable for streaming live television. Further, the combination would also decrease

’116 Paper 17 at 67-68

’116 Reply

But even if Foster-Hayhurst were less secure than Foster in ways SFI alleges (it is not), Foster-Hayhurst added a security technique that Foster lacks (distributed storage) ***and*** provided the other (undisputed) benefits (e.g., speed, intelligence, efficiency, and redundancy) that also motivated Foster-Hayhurst. Petition, 62-63 (citing EX1028-EX1031); Bhattacharjee-Reply, ¶¶174-176 (citing corroboration); *Corephotonics*, 2021 WL 4944471, at *6 (“design decisions entail making tradeoffs among multiple objectives”).

SFI argues Foster “has no need for redundancy” or “efficiency/intelligence” that Foster-Hayhurst provided. POR, 67-68. But, compared to Foster’s single source, Foster-Hayhurst’s redundancy addressed known problems (e.g., power/network failure at one source) and efficiency/intelligence allowed for selecting better source(s). Bhattacharjee, ¶¶241-244; Rubin-Depo., 89-92, 135-137. Bhattacharjee-Reply, ¶¶177-182.

’116 Paper 21 at 29

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116/’854 Foster-Hayhurst: SFI’s “no motivation” argument fails because multiple pieces of corroboration demonstrates the known benefits of Hayhurst’s storage technique

’116 Petition

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 unit has more than a segment of a media file” ([0008]). Each segment is also duplicated and stored on multiple subscriber units, which beneficially “provide[s] redundancy” – each segment is available even if “a given subscriber unit is unable to provide the…segment stored on that given unit.” Hayhurst, [0009]; *see also* [0025]. Bhattacharjee, ¶179.

’854 Paper 2 at 59-60; ’116 Pet. 57

In addition to Hayhurst’s explicit teachings, POSAs knew of those benefits from background knowledge, and knew that storing segmented content units on multiple devices in a distributed fashion provided additional advantages in terms of:

- aggregate speed (*e.g.*, by allowing segments to be retrieved from multiple sources as opposed to one source); and
- efficiency/intelligence (*e.g.*, by being able to select storage device(s) that would provide the best delivery, based on availability, location, etc.).

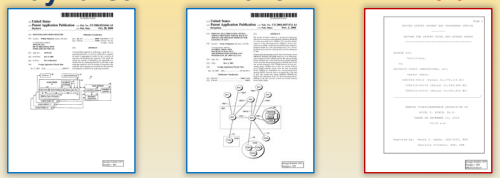
Bhattacharjee, ¶184 (citing as corroboration EX1062, [0022]-[0023]; EX1037, [0003], [0012]-[0013], [0015], [0018]-[0024]; EX1043, [0005]-[0010]; EX1050, 1:19-22, 2:31-38, 11:11-26, 16:6-20). POSAs thus had well-known reasons to incorporate Hayhurst’s distributed-storage technique into Foster’s system, in addition to motivation from Hayhurst itself. Bhattacharjee, ¶184.

’854 Paper 2 at 62; ’116 Pet. 61-63

’854 and ’116 Petitions*

Security

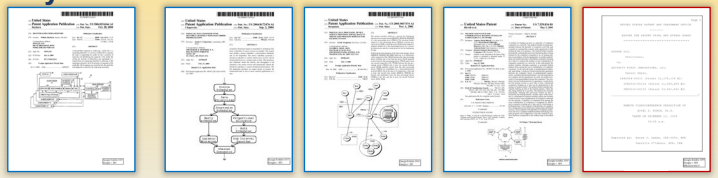
Hayhurst | EX1043 | Dr. Rubin



Cited ’854 Pet., 59-60, 62; ’116 Pet., 57, 62; ’854 Reply, 12; ’116 Reply 27-29

Redundancy / Fault tolerance

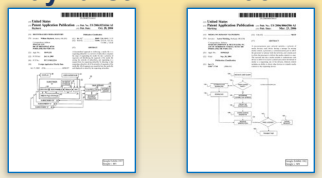
Hayhurst | EX1037 | EX1043 | EX1050 | Dr. Rubin



Cited ’854 Pet., 59-60, 62; ’116 Pet., 57, 62; ’854 Reply, 12-13; ’116 Reply 27-29

Shifts bandwidth

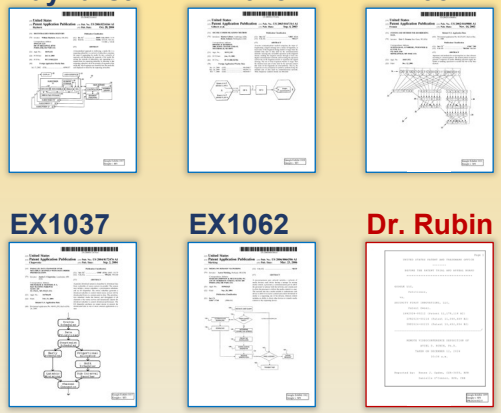
Hayhurst | EX1062



Cited ’854 Pet., 59-60, 62; ’116 Pet., 57, 62

Faster delivery / aggregate speed

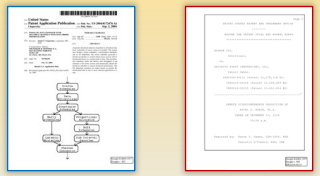
Hayhurst | EX1028 | EX1031 | EX1037 | EX1062 | Dr. Rubin



Cited ’854 Pet., 59; ’116 Pet., 59-60; ’854 Reply, 12-13; ’116 Reply 27-29

Efficient, intelligent delivery

EX1037 | Dr. Rubin



Cited ’854 Pet., 62; ’116 Pet., 62; ’854 Reply, 12-13; ’116 Reply 27-29

* Exhibit numbers are as identified in IPR2024-00215 (’854 Patent)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116/’854 Foster-Hayhurst: SFI’s Arguments

SFI further argues a POSA would not have had motivation to combine Foster with Hayhurst because:

1. Foster is only about real-time streaming
2. Hayhurst is incompatible with real-time streaming
3. Hayhurst would make Foster slower
4. Hayhurst would make Foster less secure

’116 POR, 47-68; ’116 Sur-Reply, 14-26
’854 POR, 41-62; ’854 Sur-reply, 2-19

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Foster's encryption technique is for use with various systems that store encrypted data

Foster

[0005] Current technologies that utilize key-based security practices include a personal computer-based Digital Rights Management System (DRMS) and a Conditional Access System (CAS). In the case of the former, content is packaged and loaded onto a web server. The keys used to encrypt/decrypt the content are downloaded to the personal computer, but not as an integral part of the content. Rather, either at download or rendering time, the receiver must access a license server to receive permission to access the content and a description of any corresponding usage rules.

[0007] In view of the foregoing, there exists a need for a method, system and program product for attaching a key to encrypted content for synchronized transmission to, or storage by, a receiver. That is, a need exists for a key used to encrypt content to be itself encrypted, and transmitted as an integral part of the content. By transmitting the encrypted key as an integral part of the encrypted content, a receiver would receive the encrypted content as well as all information necessary to decrypt the content in a single stream. Moreover, by transmitting the encrypted key and encrypted content as a single stream, compatibility and compliance with existing front-end and back-end standards would be maintained. In addition, synchronous transmission to (and storage by) a receiver fosters random access to the content.

[0074] It should be understood that the elements of FIGS. 1-15 used to receive/transmit transmissions, encrypt/decrypt content and/or keys, attach keys to content etc. can be implemented as hardware, software or as a combination of hardware or software. As such, any kind of computer/server system(s)—or other apparatus adapted for carrying out the methods described herein—is suited. A typical combination

DRMS

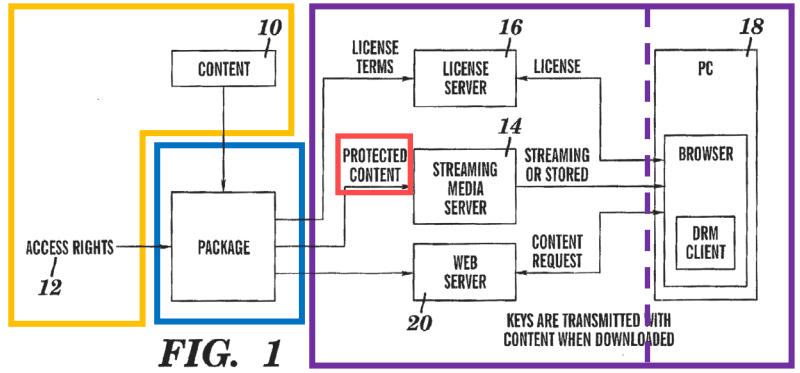


FIG. 1

CAS

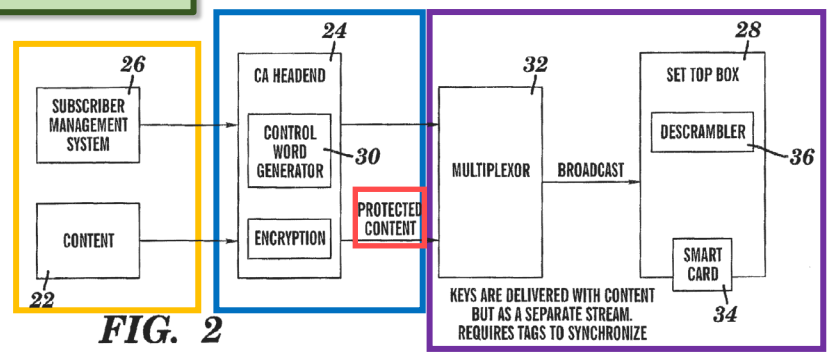


FIG. 2

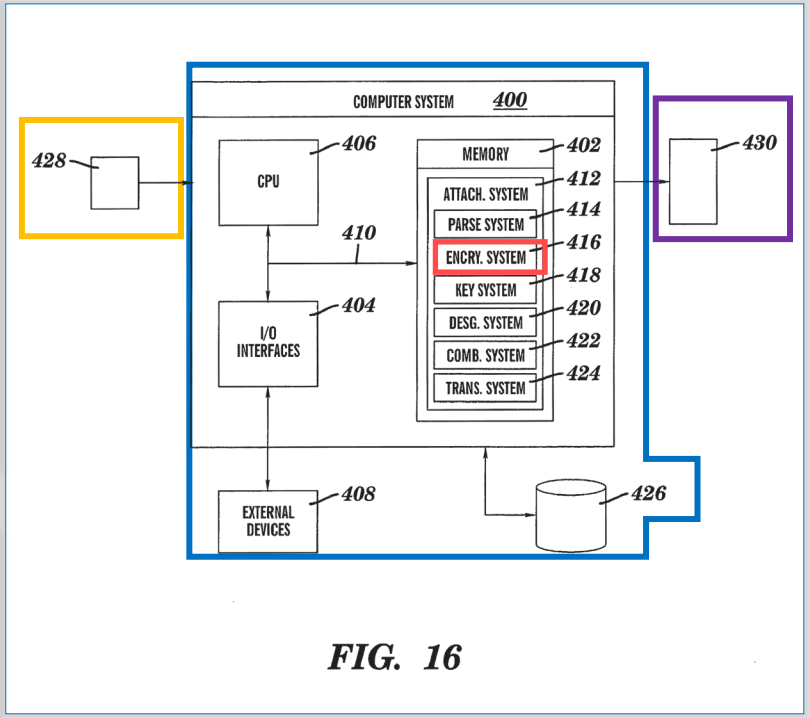


FIG. 16

EX1006, FIG. 16 (cited '854 Pet., 69)

EX1006, FIGS. 1-2 (cited '854 Reply, 8; '854 Sur-Reply, 15; '609 Reply, 2, 9, 13-14, 16; '609 Sur-Reply, 11; '116 Reply, 2, 25-26; '116 Sur-Reply, 24)

EX1006, [0005], [0007], [0074] (cited '854 Pet., 65; '854 Reply, 7-9; '854 Sur-Reply, 7; '609 Reply, 1-2, 6; '609 Sur-Reply, 10-11; '116 Pet., 52-53, 66; '116 Reply, 24-25, 27; '116 Sur-Reply, 18, 24)

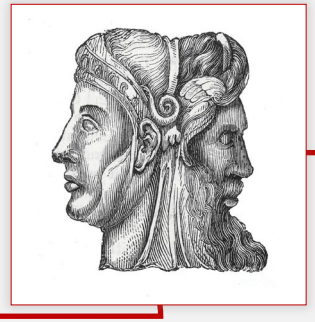
DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Foster: SFI's Argument #1 ("streaming") fails because SFI admits "Foster is directed towards 'Conditional Access System[s],'" including VOD systems that store encrypted content

SFI's POR

Foster teaches that its system processes and transmits content units between a "Content Service Provider" "such as a cable service provider" (*id.*, [0039]) and a "Receiver" "such as a set-top box" (*id.*, [0040]). *Id.*, Fig. 16, [0078]. It seeks to improve upon prior art cable systems providing real-time television. *Id.*, [0052] (Foster is directed towards "Conditional Access System[s] (CAS)," which are "cable or satellite television system[s]"); [0006]. As Dr. Rubin explains, Foster's

'854 Paper 21 at 45; *see also* '116 POR, 12; '609 POR, 45



SFI's Sur-Reply

1006 [Foster] [0051]; Ex. 2034 [Bhattacharjee-1st-Depo.] 32:20-23. Foster's inventive system is intended to replace both DRMS and CAS. *See id.*, [0003]-[0007] (criticizing DRMS and CAS, concluding a need exists for a system "for synchronized transmission").

'854 Paper 21 at 45; *see also* '116 Sur-Reply, 24; '609 Sur-Reply, 11

Reply

Finally, even Foster's implementation in conditional-access systems ("CASs") is not limited to "live" broadcasting. POSAs knew content service providers used CAS in video-on-demand (VOD), like Foster describes ([0005], [0049]-[0052]). In VOD systems, "content **is stored pre-encrypted** on hard drives in the VOD server" (EX1110, [0007]) and again in "**storage** media in which the delivered **content can be stored**" (EX1072, [0043]). Bhattacharjee-Reply, ¶¶74-85 (citing EX1073, [0001]-[0004], [0043], [0098], FIG. 3; EX1097, 2:31-53, FIG. 1; EX1075, 1:51-67, FIG. 2; EX1076, 9:53-10:5, FIG. 2; EX1077, 1:12-21, 1:48-65, 3:56-5:8, FIG. 1; EX1092, [0079]).

'854 Paper 24 at 9; '116 Reply, 26; '609 Reply, 2-3

SFI's Dr. Rubin

Q. What's the difference between DRM and CAS?

A. **So CAS is kind of a broader category, and DRM is a particular kind of CAS.**

EX1099, 96 (cited '854 Reply, 2; '609 Reply, 2; '116 Reply, 8)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Foster: SFI's Argument #1 ("streaming") fails because POSAs knew CAS-based systems store encrypted content

Reply

Finally, even Foster's implementation in conditional-access systems ("CASs") is not limited to "live" broadcasting. POSAs knew content service providers used CAS in video-on-demand (VOD), like Foster describes ([0005], [0049]-[0052]). In VOD systems, "content **is stored pre-encrypted** on hard drives in the VOD server" (EX1110, [0007]) and again in "**storage** media in which the delivered **content can be stored**" (EX1072, [0043]). Bhattacharjee-Reply, ¶¶74-85 (citing EX1073, [0001]-[0004], [0043], [0098], FIG. 3; EX1097, 2:31-53, FIG. 1; EX1075, 1:51-67, FIG. 2; EX1076, 9:53-10:5, FIG. 2; EX1077, 1:12-21, 1:48-65, 3:56-5:8, FIG. 1; EX1092, [0079]).

'854 Paper 24 at 9; '116 Reply, 26; '609 Reply, 2-3

EX1072

[0043] Video on demand (VOD) **345** enables the subscribers to select a particular content type from a list of available content stored on a plurality of servers at the head-end to be viewed by the subscribers. The selected content type could be television programs, movies, events, sporting events, music or some other form of stored media or real-time encoded broadcast. The VOD servers send the requested content to the subscriber over a delivery mechanism such as cable, satellite, or other transmission infrastructure. The client device may have some storage media in which the delivered content can be stored and viewed by the users. The user may also be provided with the transport control functions such as controlling the delivered video. This may include pause, fast forward, and so forth. The transport control functions are typically supported at the headend where the content is located/stored.

'854 EX1072, [0043] (cited '854 Reply, 9; '116 Reply, 26; '609 Reply, 3)

EX1073

[0003] Conditional Access

[0004] The system implementing VOD provides the capability to limit content access to authorized subscribers only, as the contents delivered as part of the service are generally considered valuable intellectual properties by their owners. In cable and satellite television, such capability is known as conditional access. Conditional access requires a trustworthy mechanism for classifying subscribers into different classes, and an enforcement mechanism for denying access to unauthorized subscribers. Encryption is typically the mechanism used to deny unauthorized access to content (as opposed to carrier signal).

'854 EX1073, [0003]-[0004] (cited '854 Reply, 9; '116 Reply, 26; '609 Reply, 3)

EX1075

It would be advantageous to provide a bandwidth-efficient system for conveying digital television programs and/or other content (e.g., multimedia) in their original format, which may comprise, for example, MPEG-2 transport packets. The system should allow the programs and other content to be provided on a non-real-time basis for later use by a recipient. Such content should be able to be easily received, recorded on a digital VCR or otherwise stored, and played back at the user's convenience. It would be still further

'854 EX1075, 1:51-59 (cited '854 Reply, 9; '116 Reply, 26; '609 Reply, 3)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Foster: SFI's Argument #1 ("streaming") fails because POSAs knew CAS-based systems store encrypted content

EX1097

Disadvantageously, for VOD, real-time encryption poses much greater cost and space issues. A medium-sized cable system may have, for example, 50,000 subscribers. Using a common estimate of 10% peak simultaneous usage, there can be up to 5000 simultaneous VOD sessions during the peak hours. A typical encryption device can process a small number of transport multiplexes (digital carriers). Over 300 such real-time encryption devices will be needed to handle the peak usage in the example system. Such a large amount of equipment not only adds significantly to the system cost, but also poses a space requirement challenge.

One solution to the aforementioned problem is disclosed in copending related application entitled, "SYSTEM FOR SECURELY DELIVERING PRE-ENCRYPTED CONTENT ON DEMAND WITH ACCESS CONTROL," Ser. No. 09/898,184, filed Jul. 3, 2001, which is hereby incorporated by reference in its entirety. In U.S. Ser. No. 09/898,184, a system is disclosed that encrypts content offline (typically before the content is requested by the user) before it is distributed to point-to-point systems such as cable systems. The system allows content to be encrypted once, at a centralized facility, and to be useable at different point-to-point systems. Advantageously, the pre-encrypted contents in the present invention have indefinite lifetimes. The

'854 EX1097, 2:31-54, FIG. 1
(cited '854 Reply, 9; '116 Reply, 26; '609 Reply, 3)

EX1097

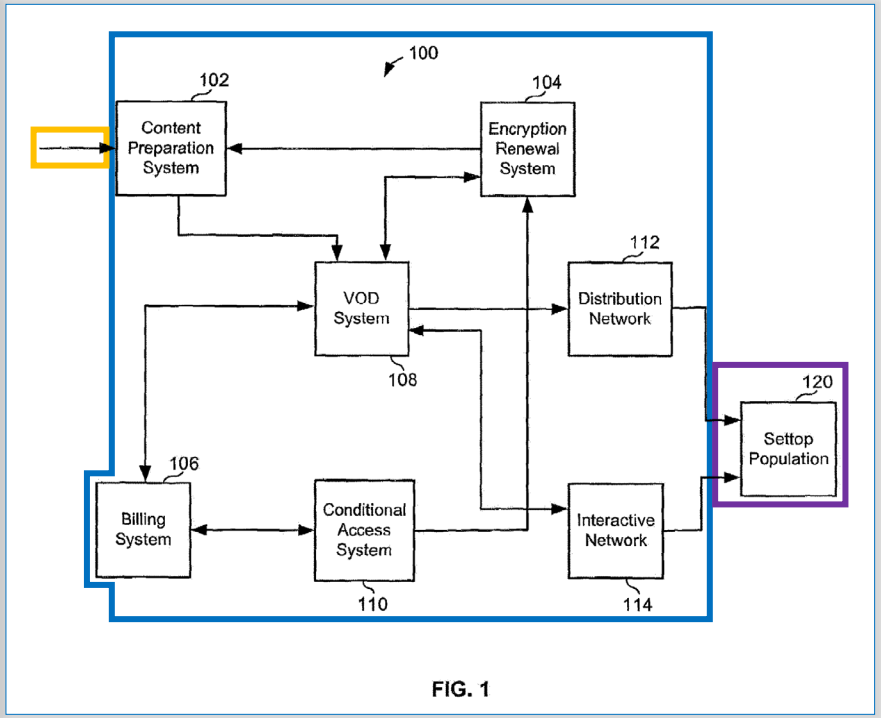


FIG. 1

Foster

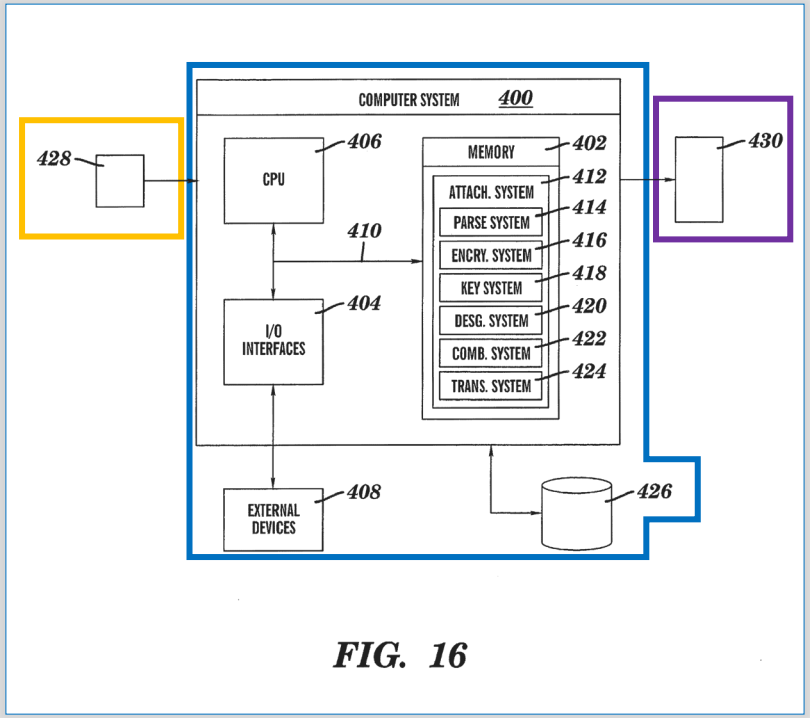


FIG. 16

EX1006, FIG. 16 (cited '854 Pet., 62; '116 Pet., 53-54; '609 Pet., 64)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Foster: SFI's Argument #1 ("streaming") fails because POSAs knew CAS-based systems store encrypted content

EX1076

FIG. **5** is a flow chart that schematically shows an overview of operations of computer **34** in preparing and transmitting data stream **40** over network **28**, in accordance with a preferred embodiment of the present invention. Details of some of the steps in the operation are shown in FIGS. **7–9** and described further with reference thereto. Exemplary programs for carrying out the functions illustrated in FIG. **5** are incorporated herein in a software appendix, which is described further hereinbelow.

To begin the broadcast, computer **34** connects to server **36**, optionally opening the plurality of links shown in FIG. **4**. Broadcast data are then input to the computer, for example, from input devices **22**, or from a video, audio or animation sequence stored on disk or tape. The data are compressed at step **80**, and are then "sliced" at step **82** into files **42**, **44**, **46**, **48**, etc., as shown in FIG. **3A**. Computer **34** conveys file **40** to server **36** over links **60**, **62**, **64**, **66** and **68**, as described above, preferably using FTP, at step **84**. Each time a new file is uploaded to the server, index file **50** (FIG. **3B**) is updated, at step **86**.

EX1076, 9:53-10:5 (cited '854 Reply, 9; '116 Reply, 26; '609 Reply, 3)

EX1092

[**0079**] **FIG. 9** illustrates a communications system **900** that includes a storage device **905**, such as a video-on-demand server. It will be appreciated that the video-on-demand server can store other types of information, such as audio and data, and are not constrained to storing video only.

EX1092, [0079] (cited '854 Reply, 9; '609 Reply, 3)

EX1076

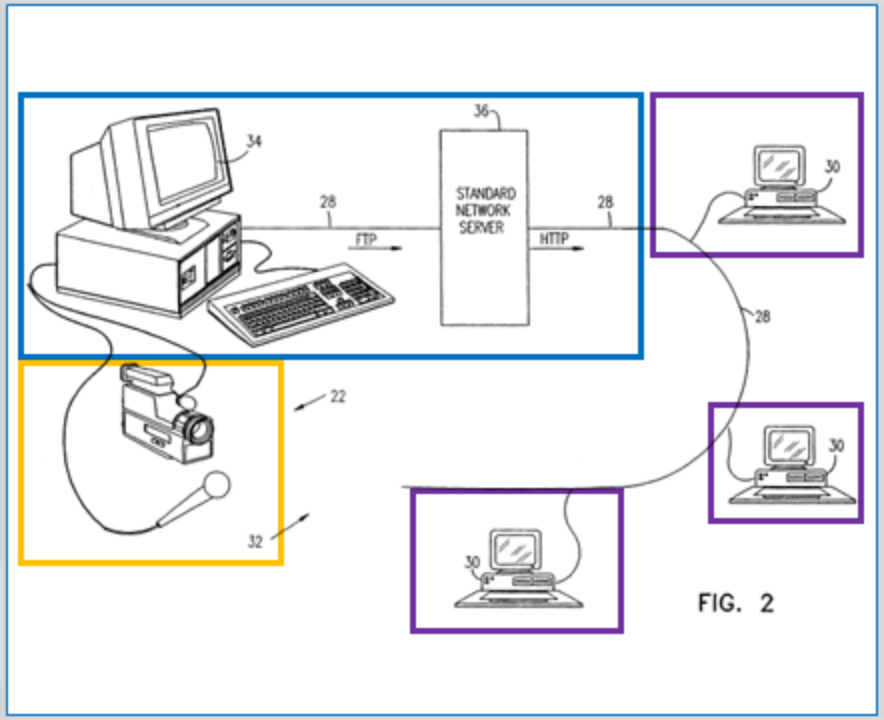


FIG. 2

EX1076, FIG. 2
(cited '854 Reply, 9; '116 Reply, 26; '609 Reply, 3)

Foster

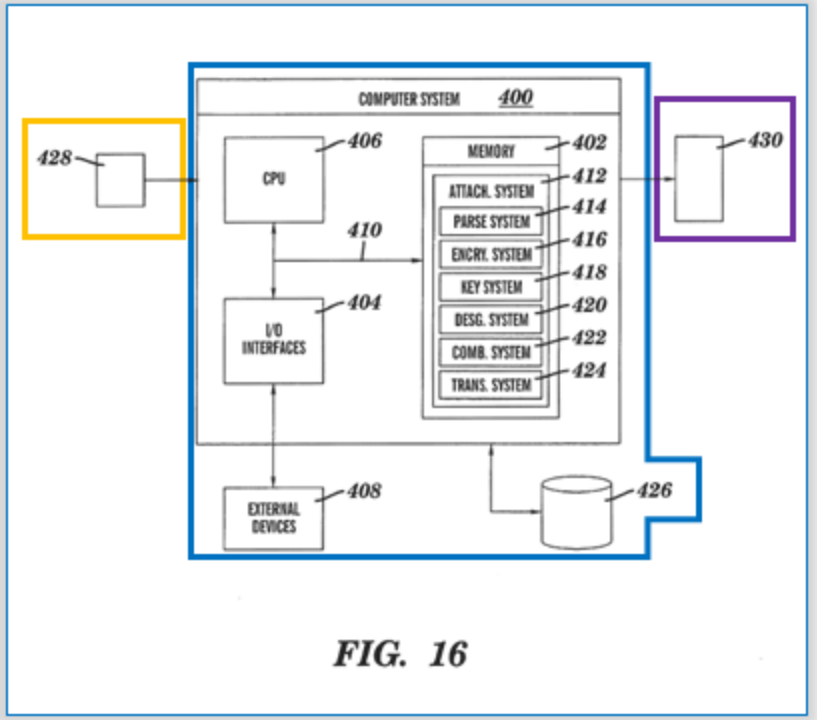


FIG. 16

EX1006, FIG. 16
(cited '854 Pet., 62; '116 Pet., 53-54; '609 Pet., 64)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Foster: SFI's Argument #1 ("streaming") fails because POSAs knew CAS-based systems store encrypted content

SFI's Expert Dr. Rubin

Q. Okay. And you wrote in your paragraph 131 of your '854 declaration, you say: "CAS, conditional access system, used for OTT digital video streaming service 'such as live sports.'" Do you see that?

A. I do.

Q. Is your opinion that CAS, conditional access systems, are only used for live sports?

A. I said "such as live sports."

Q. Is it used for anything else other than live sports?

A. It is.

Q. Such as? Can you give me an example?

A. Now, when you're asking -- when you say "is it used," are you talking about CAS, or is CAS for digital video streaming services?

Q. Well, I'm asking for CAS more broadly, not just for digital video streaming service.

A. **Yeah. It's used in a lot of places.**

Q. Can you name any examples?

A. **File systems.**

Q. Any other examples?

A. I think basically **any place that you have permissions and you want to prevent people who aren't authorized from getting to something, you could use conditional access system.**

Q. And would that have been true in the 2004-2005 time frame?

A. Yes.

Q. Do you know if conditional access systems were used for video-on-demand services in the 2004-2005 time frame?

A. **They were.**

Q. Do you know how a video-on-demand system works?

A. Yes, I do.

* * *

A. Sure. In that time frame, you asked for video-on-demand?

Q. Yes

A. Right. So there would be a collection of **content that would be hosted by a hosting site. AT&T was an example of that. And there would be content producers that would license the content to the hosting sites.** And then users would subscribe to the service.

Q. Okay. Let's drill down a little bit. So there was content made by a content producer or content owner, correct?

A. Yes.

Q. Like a movie studio, correct?

A. Yes.

Q. And they have a movie, correct?

A. Yes.

Q. Okay. And then give that movie to a service provider; is that correct?

A. **Yes**

Q. The service provider stores that content on a server; is that correct?

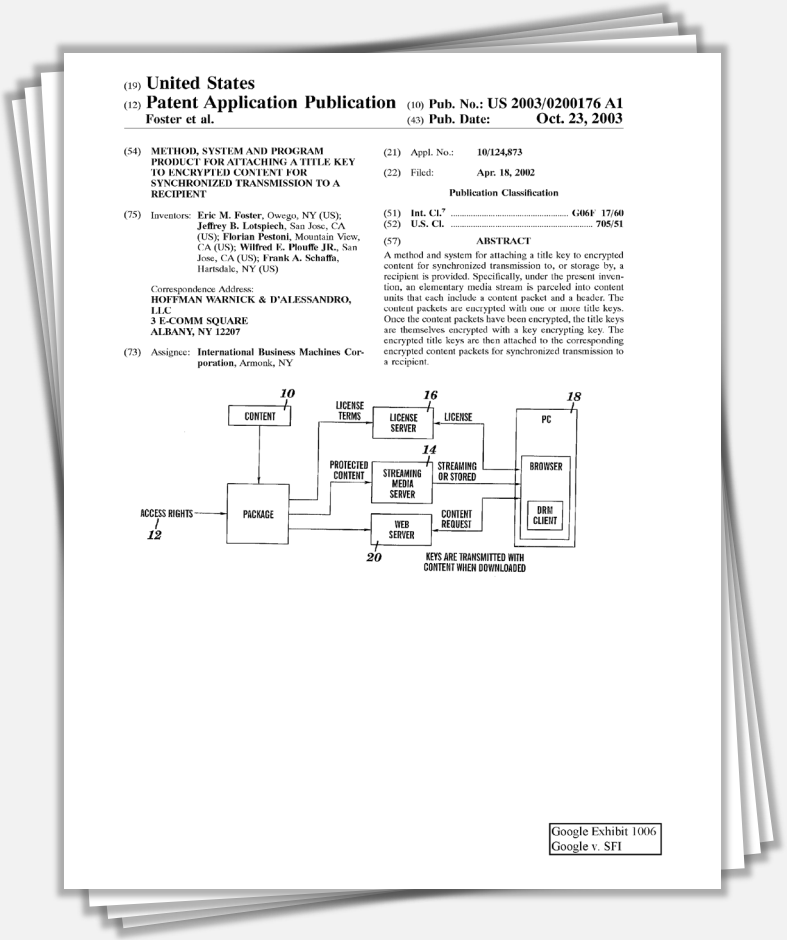
A. So there were different models. Obviously, we're talking about one particular example here of how this happened. But there would be **service providers that would keep the content, the movie, or whatever, themselves.**

EX1099, 85-88 (cited '854 Reply, 10; '116 Reply, 26; '609 Reply, 3)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Foster: SFI's Argument #1 ("streaming") fails because Foster is also directed towards Digital Rights Management Systems (DRMS), which POSAs knew store encrypted content

Foster



(19) **United States**
(12) **Patent Application Publication** (10) **Pub. No.: US 2003/0200176 A1**
Foster et al. (43) **Pub. Date: Oct. 23, 2003**

(54) **METHOD, SYSTEM AND PROGRAM PRODUCT FOR ATTACHING A TITLE KEY TO ENCRYPTED CONTENT FOR SYNCHRONIZED TRANSMISSION TO A RECIPIENT**

(75) Inventors: **Eric M. Foster**, Owego, NY (US); **Jeffrey B. Lotspiech**, San Jose, CA (US); **Florian Pestoni**, Mountain View, CA (US); **Wilfred E. Plouffe JR.**, San Jose, CA (US); **Frank A. Schaffa**, Hartsdale, NY (US)

Correspondence Address:
HOFFMAN WARNICK & D'ALESSANDRO, LLC
3 E-COMM SQUARE
ALBANY, NY 12207

(73) Assignee: **International Business Machines Corporation**, Armonk, NY

(21) Appl. No.: **10/124,873**

(22) Filed: **Apr. 18, 2002**

Publication Classification

(51) **Int. Cl.**[7] **G06F 17/60**
(52) **U.S. Cl.** **705/51**

(57) **ABSTRACT**

A method and system for attaching a title key to encrypted content for synchronized transmission to, or storage by, a recipient is provided. Specifically, under the present invention, an elementary media stream is parceled into content units that each include a content packet and a header. The content packets are encrypted with one or more title keys. Once the content packets have been encrypted, the title keys are themselves encrypted with a key encrypting key. The encrypted title keys are then attached to the corresponding encrypted content packets for synchronized transmission to a recipient.

Google Exhibit 1006
Google v. SFI

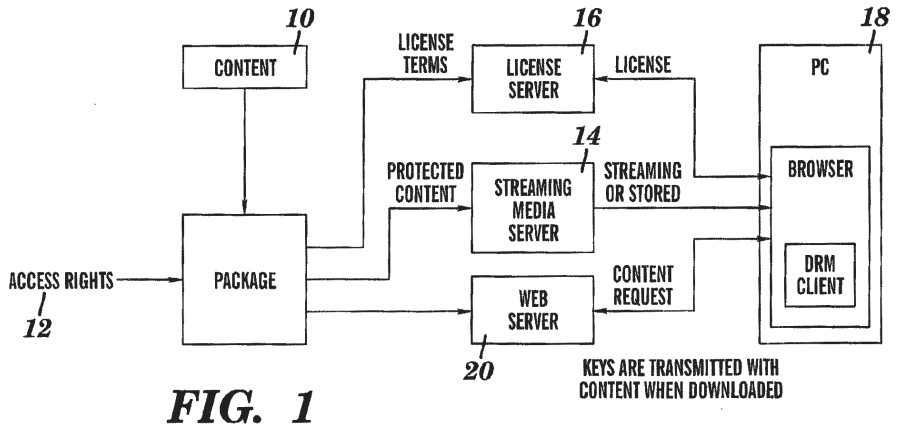


FIG. 1

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] These and other features of this invention will be more readily understood from the following detailed description of the various aspects of the invention taken in conjunction with the accompanying drawings in which:

[0015] **FIG. 1** depicts a flow diagram of a Digital Rights Management System (DRMS).

EX1006, Cover, FIG. 1, [0014]-[0015] (cited '854 Pet., 3, ; '609 Pet.,3; '116 Pet., 1, 51; '854 POR, 42;'854 Reply, 8; '609 Reply, 15-16)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Foster: SFI's Argument #1 ("streaming") fails because POSAs knew DRMSs store encrypted content

SFI's POR

1. Foster Provides Synchronized Transmission To Avoid Delays.

Foster's system is directed towards synchronizing the transmission of encrypted content packets with the keys used to encrypt those packets, so that each packet can be viewed as soon as it is received. This fundamental goal is reflected

'854 Paper 21 at 42; '116 POR, 48; '609 POR, 33-34

Reply

Foster recognizes that synchronization reduces latency for storage-based purposes, like for Digital Rights Management Systems ("DRMSs"). Foster, [0064] ("[A] recipient will receive or **store** encrypted content…."). Rubin-Depo., 83-85, 96-99. In Foster's DRMS, "content is packaged and **loaded** [i.e., stored] onto a web server" and subsequently "**downloaded to** [i.e., stored to a] personal computer" for on-demand access (i.e., viewing stored content whenever the user wants to). Foster, [0005], [0049] (users access "content on [the] media server"); FIG. 1 (content is "stream[ed] **or stored**"); EX2039, 6-10 (DRMS "safeguards **content** when **at rest** [i.e., stored]," and when "**stored** on a receiving device"). Foster's synchronization also facilitates "random access" to stored chunks of "protected content" (e.g.,

'854 Paper 24 at 8; '116 Reply, 25; '609 Reply, 2

SFI's EX2039

What is digital rights management (DRM)?

Digital rights management (DRM) refers to the tools, standards and systems that are used to protect and monetize intellectual property and copyrighted materials from misuse or theft in the digital sphere, including Internet Protocol (IP) video pay-TV services transmitted over the internet.

Traditionally, digital content has been protected with encryption. Using a secret key, content is encrypted (and thus made unwatchable) so that only somebody who has the key can decrypt and view it. But because key, like all digital information, is easy to copy and share, on its own it is not sufficient to protect the content.

DRM technology can be very flexible and may include business rules for secure download and offline playback, enabling a viewer on-the-go without internet access to enjoy subscribed content.

CAS was designed to protect pay-TV content during transport from the digital video processing head-end (a.k.a. network operations center, or NOC) to STBs over various kinds of broadcast or "one-way" networks. DRM, on the other hand, is a broader asset protection mechanism that also safeguards content when at rest in addition to the transport phase, made possible by IP-based "two-way" networks such as the internet.

EX2039 (cited '116 POR, 61; '854 POR, 55-56; '609 POR, 55; '116 Reply, 25; '854 Reply, 8; '609 Reply, 2)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Foster: SFI's Argument #1 ("streaming") fails because POSAs knew DRMSs store encrypted content

SFI's Expert Dr. Rubin

Q. Okay. So on page 7 of Exhibit 2039. It says, effectively, DRM functionality is a superset of CAS.
Do you see that?

A. Yes.

Q. What's the difference between DRM and CAS?

A. So CAS is kind of a broader category, and DRM is a particular kind of CAS

Q. Does -- is that what Exhibit 2039 says? It seems to be saying to me that it's a -- DRM is a superset of CAS or CAS broader than DRM?

A. So I guess **I'm viewing DRM as being a particular implementation of CAS. So it has all the functionality of CAS, but it has other things as well.**

Q. What other things does it have? DRM?

A. Right.
It says it in this paragraph: So **DRM protects content on any device with various distribution models, offline, online**, with or without return channel. And it can also be applied to other types of digital content such as ebooks, BICODE, and more.

Q. What does it mean by distribution models offline?

A. Like, for example, if I give you **a disk with some data** instead of using the internet to it to you.

Q. Okay. And then would distribution models online be using the internet to transmit data to a recipient?

A. **I would think so.**

Q. This Exhibit 2039 says, towards the bottom of page 7, "A DRM license is typically a small file that contains the content encryption key in encrypted form along with some rules, policies, defining how the contents may be used."
Do you see that?

A. Yes.

Q. That's saying the content is encrypted before it's transmitted to the consumer?

A. I mean, I don't see that saying in the sentence, but **I believe that is the case.**

EX1099, 85-88 (cited '854 Reply, 10; '116 Reply, 26; '609 Reply, 3)

Foster: SFI's Argument #1 ("streaming") fails because Foster encrypts for a "program stream," which Foster says (and POSAs knew) is "generally used" "in fixed storage"

Reply

Consistent with its DRMS disclosures, Foster discloses processing datasets for an MPEG "program stream," which Foster says (and POSAs knew) is intended for storing "**in fixed storage**." Foster, [0073]; *see also* EX2014, 1 ("[p]rogram streams are **designed for**…**storage**"); EX1108, 1:65-2:31; EX1109, 1-9, 13); EX1072, [0002]-[0004], [0011], [0042]-[0043]; Bhattacharjee-Reply, ¶¶67-73; Rubin-Depo., 82-85 (admitting there are "different ways of encoding MPEG data" but he did not "discuss[] the distinctions"). Foster's text makes clear that Foster includes storage-based embodiments.

'854 Paper 24 at 8; '116 Reply, 26; '609 Reply, 2

SFI's EX2039

The MPEG-2 standard defines two ways to deliver packetized elementary streams: *program streams* and *transport streams*.

Program Streams

Program streams are designed for environments that are relatively error-free, such as local file storage. In a program stream, the PES packets are multiplexed and organized into units called *packs*. All of the PES streams in a program stream are synchronized to the same clock.

EX2039, 1 (cited '854 Reply, 8; '116 Reply, 26; '609 Reply, 2)

Foster

[0073] Once the encrypted key (and any optional identifier information and/or verifier) has been attached to PES packet **306**, the resulting PES unit **302** can then be multiplexed in two different system options. The first is shown in **FIG. 14** and is known as a transport stream. A transport stream is typically used in error-prone broadcast environments such as digital television. As depicted, transport stream packets **350**A-F are relatively small, and it generally takes multiple transport stream packets **350**A-C to contain a single video PES unit **302**. The second option is a program stream, which is shown in **FIG. 15**. A program stream is generally used in more reliable transmissions such as those with error correcting codes or in fixed storage. Unlike transport stream packets, however, a single program stream packet **360** can hold multiple video PES units **302** and **362**.

'854 EX1006, [0073] (cited '854 Reply, 9; '116 Reply, 26; '609 Reply, 2, 10, 15)

EX1108

and presented together in a coherent fashion. The program and transport streams may be recorded on or played back from a digital video disc (DVD), video tape, magnetic or optical disk drive or other suitable storage device. It should

EX1108, 2:1-4 (cited '854 Reply, 9; '116 Reply, 26; '609 Reply, 2)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Foster: SFI's Argument #1 ("streaming") fails because Foster encrypts for a "program stream" to store "in fixed storage" (e.g., a DVD)

SFI's POR

Foster's synchronization is directed towards reducing delays in viewer access to content as it is transmitted by a provider. Per Dr. Rubin:

'116 Paper 17 at 49

Reply

SFI argues Foster uses "synchronization" to "reduc[e] delays in viewer access to content **as it is transmitted**" in real-time. POR, 42-43 (citing Foster, [0008]). But the same sentence SFI cites also teaches that Foster attaches a key to encrypted data "for synchronized…**storage**." Foster, [0008]. Foster repeatedly says synchronization improves data for "transmission…**or storage**." Foster, Abstract; *see also* [0007]-[0008], [0048], [0064], [0079], claim 11. The Board recognized this at institution. DI, 17 (Foster "synchronously **stor[es]** the encrypted content"). SFI admitted in IPR2024-00214 (Paper 22, 44, 51) that Foster "**teaches**...the 'recipient 430 **can**….**store** the encrypted content.'" Similarly, SFI's expert admitted Foster's "recipients can determine whether to **store**." Rubin-Depo., 187-191 ("The recipient…could decide to **store** it."); EX2033, ¶140 ("Foster's system focuses upon," *e.g.*, "content **stored** on a DVD"). That Foster stores its encrypted data should not be genuinely disputed.

'854 Paper 24 at 7-8;
'116 Reply, 10-11

SFI's Dr. Rubin

140. Foster's system, however, has no need for any of these alleged benefits. Foster's system simply processes content and transmits streamed content to a particular recipient. Ex. 1006 [Foster] Fig. 16; [0078]. Because Foster's system focuses upon streamed content (or content stored on a DVD) and provides a

'854 EX2033, ¶140
(cited '854 POR, 62; '116 POR, 68; '854 Reply, 9; '116 Reply, 25)

SFI's Sur-Reply

Petitioner also contends that Foster describes an MPEG program stream used for fixed storage. That a content file may be stored in "fixed storage" does not indicate that Foster teaches that the entire content file is stored on a personal computer in its system. Rather, the only embodiment in Foster that Petitioner alludes to as teaching "fixed storage" is "content stored on a DVD." *See* Reply, 8. But Petitioner never relied on Foster's DVD embodiment in the Petition. *See* Pet.,

'854 Paper 32 at 16; '116 Sur-Reply, 25

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116/’854 Foster-Hayhurst: SFI’s Argument #2 (“incompatibility”) fails because Foster’s use cases (DRMS; CAS; MPEG “program stream”) were commonly used in various storage-based applications

’854 Reply

CAS

Finally, even Foster’s implementation in conditional-access systems (“CASs”) is not limited to “live” broadcasting. POSAs knew content service providers used CAS in video-on-demand (VOD), like Foster describes ([0005], [0049]-[0052]). In VOD systems, “content **is stored pre-encrypted** on hard drives in the VOD server” (EX1110, [0007]) and again in “**storage** media in which the delivered **content can be stored**” (EX1072, [0043]). Bhattacharjee-Reply, ¶¶74-85 (citing EX1073, [0001]-[0004], [0043], [0098], FIG. 3; EX1097, 2:31-53, FIG. 1; EX1075, 1:51-67, FIG. 2; EX1076, 9:53-10:5, FIG. 2; EX1077, 1:12-21, 1:48-65, 3:56-5:8, FIG. 1; EX1092, [0079]).

’854 Paper 24 at 9; ’116 Reply, 26; ’609 Reply, 2-3

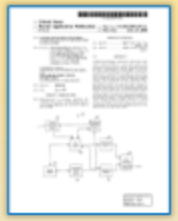
Foster



EX1110



EX1073



EX1092



EX1075



EX1076



EX1077



Dr. Rubin



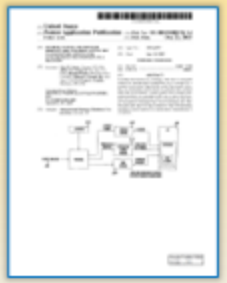
Cited ’854 Reply, 9; ’116 Reply, 26; ’609 Reply, 2-3

DRMS

Foster recognizes that synchronization reduces latency for storage-based purposes, like for Digital Rights Management Systems (“DRMSs”). Foster, [0064] (“[A] recipient will receive or **store** encrypted content….”). Rubin-Depo., 83-85, 96-99. In Foster’s DRMS, “content is packaged and **loaded** [i.e., stored] onto a web server” and subsequently “**downloaded to** [i.e., stored to a] personal computer” for on-demand access (i.e., viewing stored content whenever the user wants to). Foster, [0005], [0049] (users access “content on [the] media server”); FIG. 1 (content is “stream[ed] **or stored**”); EX2039, 6-10 (DRMS “safeguards **content** when **at rest** [i.e., stored],” and when “**stored** on a receiving device”). Foster’s synchronization also facilitates “random access” to stored chunks of “protected content” (e.g., starting a stored movie at any point). Foster, [0005], [0064] (random access is “access[ing] and decrypt[ing]…**at any packet**”). Bhattacharjee-Reply, ¶¶59-66.

’854 Paper 24 at 8-9; ’116 Reply, 25-26; ’609 Reply, 2

Foster



EX2039



Dr. Rubin



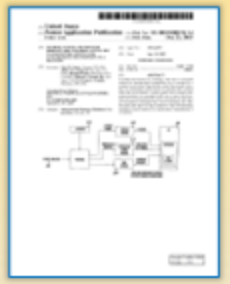
Cited ’854 POR, 8-9; ’854 Reply, 8-9; ’116 Reply, 25-26; ’609 Reply, 2

MPEG “program stream”

Consistent with its DRMS disclosures, Foster discloses processing datasets for an MPEG “program stream,” which Foster says (and POSAs knew) is intended for storing “**in fixed storage**.” Foster, [0073]; *see also* EX2014, 1 (“[p]rogram streams are **designed for**…**storage**”); EX1108, 1:65-2:31; EX1109, 1-9, 13); EX1072, [0002]-[0004], [0011], [0042]-[0043]; Bhattacharjee-Reply, ¶¶67-73; Rubin-Depo., 82-85 (admitting there are “different ways of encoding MPEG data” but he did not “discuss[] the distinctions”). Foster’s text makes clear that Foster includes storage-based embodiments.

’854 Paper 24 at 9; ’116 Reply, 26; ’609 Reply, 2

Foster



EX1108



EX1109



EX1072



EX2014



Cited ’854 POR, 9; ’854 Reply, 9; ’116 Reply, 26; ’609 Reply, 2

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116/’854 Foster-Hayhurst: SFI’s Arguments

SFI argues a POSA would not have had motivation to combine Foster with Hayhurst because:

1. Foster is only about real-time streaming
2. **Hayhurst is incompatible with real-time streaming**
3. Hayhurst would make Foster slower
4. Hayhurst would make Foster less secure

’116 POR, 47-68; ’116 Sur-Reply, 17-20
’854 POR, 41-63; ’854 Sur-reply, 5-10

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116/’854 Foster-Hayhurst: SFI’s Argument #2 (“incompatibility”) fails even if Foster were limited to real-time streaming because SFI’s patents say (and POSAs knew) that distributed storage/delivery supports streaming

SFI’s Sur-Reply

Foster’s emphasis is avoiding delays to allow swift random access to content. As Petitioner’s expert concedes, “synchronous transmission” of a series of packets and the keys used to encrypt them to “foster random access to content” at a (single) receiver was “sort of the whole invention in Foster.” EX2034 [Bhattacharjee-Depo.], 32:20-23. The Foster-Hayhurst combination, however, would thwart these aims because applying Hayhurst’s distribution regime to Foster would massively increase delays (the very thing Foster sought to avoid) and preclude real-time streaming and/or random access to content. *Id.*, 57-64. The synchronization of keys and content required by Foster is irrelevant to Hayhurst. *Id.*, 57-58.

In its Reply, Petitioner argues that Foster is not “limited to streaming ‘live’ broadcasts” and “also concerns storing secured data.” Reply, 26-27. But Patent Owner has never argued that Foster is so limited. Rather, the POR demonstrated that Foster seeks to improve upon prior-art systems (DRMS and CAS) used for live or real-time television/streaming by synchronizing content packets and the keys used to encrypt each. POR, 12-13; *see also* EX1006 [Foster], [0006]-[0007], [0050], [0052].

’116 Paper 29 at 17-18; ’854 Sur-Reply, 7

Dr. Bhattacharjee

105. In my opinion, based on my experience and research in this space, I do not agree and POSAs would not agree that there would be “enormous delays” over Foster. *See supra* §VI.B.2 (discussing how Foster-Hayhurst would reduce delays over Foster alone).

106. For one, I note that the ’116 patent itself recognizes that sending data via “multiple parallel streams” (as would have occurred in Hayhurst-Foster but not Foster alone) provides for “**increasing speed and efficiency**.” EX1001, 70:10-12. Benefits of increased speed and efficiency are **exactly** two of the benefits that I opined would have motivated POSAs to pursue the Foster-Hayhurst combination, which I corroborated with independent prior art (*i.e.*, my opinion was not based on hindsight from the ’116 patent’s teachings).

’854 EX1111 ¶¶105-106 (cited Reply, 11)
’116 EX1111 ¶¶51-52 (cited Reply, 19-20)

’116 Patent

Similarly, in the case of streaming broadcasts of data (e.g., audio, video), the outgoing data may be parsed and separated into multiple streams each containing a portion of the parsed data. The multiple streams may be transmitted along one or more paths and recombined at the recipient’s location in accordance with the present invention. One of the benefits of this approach is that it avoids the relatively large overhead associated with traditional encryption of data followed by transmission of the encrypted data over a single communications channel. The secure parser of the present invention allows data in motion to be sent in multiple parallel streams, increasing speed and efficiency.

’116 EX1001, 70:1-12 (cited Reply, 27)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116/’854 Foster-Hayhurst: SFI’s Argument #2 (“incompatibility”) fails because Hayhurst, like Foster, explicitly says it can be implemented in a cable television (CATV) network

SFI’s ’116 POR

(b) *Hayhurst’s Delivery Method Is Incompatible with Foster’s System.*

The principal application for Foster’s method is real-time cable television broadcasting to a set-top box.

As Dr. Rubin explains, Foster seeks to securely stream content, *e.g.*, live television, to a set-top box without the delay found in previous systems due to lack of synchronization between each packet and the key needed to decrypt it:

’116 Paper 17 at 61; ’854 POR, 55

SFI’s ’116 POR

Petitioner contends that “POSAs would have had numerous reasons” to store Foster’s encrypted content units on multiple storage devices (*e.g.*, consumer set-top boxes like Hayhurst’s subscriber units and Foster’s recipients 430) among the various subscribers in Hayhurst’s subscriber network. Pet., 62. But doing so would render the combination inoperable for viewing streamed content, Foster’s intended purpose, because Hayhurst requires that all segments be delivered to a requesting unit before the content can be reassembled and playback can begin. As Dr. Rubin explains:

’116 Paper 17 at 62; ’854 POR, 56

Foster

[0049] Referring now to **FIG. 1, a** depiction of a Digital Rights Management System (DRMS) is shown. As depicted, content **10** is packaged with access rights/content usage conditions **12** and loaded onto media server **14**. The keys used to access the content are downloaded to PC **18** as a separate component from the content. Specifically, either at download or at rendering time, license server **16** must be accessed to receive permission to access the content and a description of any corresponding usage conditions. Web server **20** provides the web pages that walk the customer through the steps to access the content on media server **14**.

[0050] As shown, the content can be delivered through streaming delivery (e.g., if the file is large).

* * *

[0052] Referring now to **FIG. 2, a** Conditional Access System (CAS), such as a cable or satellite television system, is shown in greater detail. In a CAS, two entitlement messages are delivered to receiver **28** (e.g., a set-top box).

’116 EX1006, [0049]
(cited ’116 Reply, 28; ’854 Reply, 9; ’609 Reply, 2-3)

Hayhurst

[0016] With reference to Fig. 1, a preferred embodiment of the present invention is includes a network 20 of electronically connected subscribers A, B, C, and D. The small number of subscribers depicted here is intended to be representative of, for example, the thousands of subscribers that may be part of a typical cable television (CATV) network. Alternatively, the network can be any other communications network of users or subscribers, such as those linked by a local area network (LAN) or a wide area network (WAN).

’854/’116 EX1007, [0016]
(cited ’116 Pet., 64; ’854 Pet., 60)
’609 EX1116
(cited ’609 Reply, 3)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116/’854 Foster-Hayhurst: SFI’s Arguments

SFI argues a POSA would not have had motivation to combine Foster with Hayhurst because:

1. Foster is only about real-time streaming
2. Hayhurst is incompatible with real-time streaming
3. **Hayhurst would make Foster slower**
4. Hayhurst would make Foster less secure

’116 POR, 47-68; ’116 Sur-Reply, 17-20
’854 POR, 41-63; ’854 Sur-reply, 5-10

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116/’854 Foster-Hayhurst: SFI’s Argument #3 (“slow speed”) fails because Hayhurst would have improved the speed of delivering Foster-encrypted data

SFI’s ’116 POR

(a) *Adding Hayhurst’s Delivery Method to Foster’s System Exacerbates the Underlying Problem Of Delay Foster Seeks to Solve.*

As explained above, in Foster’s system, because each packet in an elementary content stream can be generated as the content is created, and then sent directly to the recipient with its title encryption key, each packet can be viewed without delay, allowing for immediate, random access to the content. *See supra* Sections III.A.1, III.B.1.

Replacing Foster’s direct distribution of content streams with Hayhurst’s decentralized delivery and storage of media files would introduce the delays it was Foster’s goal sought to avoid. Hayhurst, again, concerns a decentralized delivery system where a movie file is divided up into segments stored amongst thousands of subscribers. By its logic, Hayhurst’s system inherently requires significant delays before content can be played. *See supra* Section IV.B.2. As Dr. Rubin explains:

’116 Paper 17 at 58; ’854 POR, 52-53

’116 Reply

SFI speculates Foster-Hayhurst would have excessive delays when a movie file is distributed amongst “thousands” of recipients. POR, 58-60. But Hayhurst’s distributed system ***reduces*** delay by increasing aggregate speed and efficiency of content storage/delivery by having multiple devices store and deliver content. Hayhurst, [0005], [0008]-[0009] (cited Petition, 62); Bhattacharjee, ¶¶241-243; Bhattacharjee-Reply, ¶¶145-158; EX1001, 70:10-12; Rubin-Depo., 89-93.

’116 Paper 21 at 27; ’854 Reply, 11

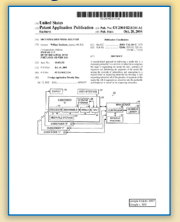
SFI’s ’116 Sur-Reply

[Bhattacharjee-Depo.], 32:20-23. The Foster-Hayhurst combination, however, would thwart these aims because applying Hayhurst’s distribution regime to Foster would massively increase delays (the very thing Foster sought to avoid) and preclude real-time streaming and/or random access to content. *Id.*, 57-64. The synchronization of keys and content required by Foster is irrelevant to Hayhurst. *Id.*, 57-58.

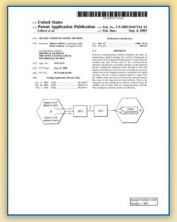
’116 Paper 29 at 18; ’854 Sur-Reply, 7

Reply & Petition

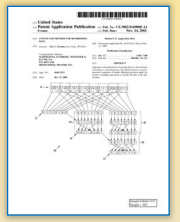
Hayhurst



EX1028



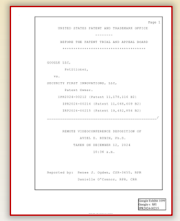
EX1031



Dr. Bhattacharjee



Dr. Rubin



’116/’854 Foster-Hayhurst: SFI’s Arguments

SFI argues a POSA would not have had motivation to combine Foster with Hayhurst because:

1. Foster is only about real-time streaming
2. Hayhurst is incompatible with real-time streaming
3. Hayhurst would make Foster slower
4. Hayhurst would make Foster less secure

’116 POR, 47-68; ’116 Sur-Reply, 18-20
’854 POR, 51-63; ’854 Sur-reply, 8-9

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Foster-Hayhurst: SFI's Argument #4 ("less secure") fails because Hayhurst's distributed storage would have added a layer of security that Foster lacks

SFI's POR

c. *Hayhurst's Methods Make Foster's System Less Secure.*

Foster's content units are encrypted with title keys which are, in turn, encrypted with a key-encrypting-key ("KEK"). Pet., 64. In Petitioner's proposed Foster/Hayhurst combination, the encrypted content units would be, as in Hayhurst, distributed across thousands of subscribers. *Id.* A POSITA would not be motivated to do this because it would compromise the security of the system, as compared to Foster, by making Foster's KEKs more vulnerable while eliminating the ability (disclosed in Foster) to change the KEK as necessary.

'854 Paper 21 at 58; '116 POR, 64

Reply

Moreover, independent of where the KEK is stored, Foster-Hayhurst would still reduce the risk of unauthorized access compared to Foster because an attacker must collect chunks from Foster-Hayhurst's multiple storage devices, rather than from one storage device, as in Foster. Petition, 59-60 (citing Hayhurst, [0005], [0008]); Bhattacharjee-Reply, ¶¶118-119.

Thus, Foster-Hayhurst does not diminish Foster's security. *Contra* POR, 58. Rather, Hayhurst adds "a **high level of security**." Hayhurst, [0008]. Bhattacharjee-Reply, ¶¶118-120; Rubin-Depo., 124-126.

'854 Paper 24 at 12; '116 Reply, 28

SFI's Dr. Rubin

Q. Whatever you wrote here. You said the '854 addresses this concern. What concern are you referring to?

A. That when data remains whole or intact it's subject to attack.

Q. Why is it a concern of the '854 patents?

A. Because the '854 patent is about securing data.

Q. Why is data remaining whole and intact a concern?

A. An adversary would have access. Let's say an adversary breaks into the site where the data is, then all the data's there.

Q. And so it's problematic if you keep all the data in one location; is that correct?

A. It is a risk.

EX1099, 17 (cited '854 Reply, 14; '609 Reply, 4; '116 Reply, 30)

SFI's Sur-Reply



DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Foster-Hayhurst: SFI's Argument #4 ("less secure") fails because Foster's encryption technique does not require changing the KEK

SFI's POR

c. *Hayhurst's Methods Make Foster's System Less Secure.*

Foster's content units are encrypted with title keys which are, in turn, encrypted with a key-encrypting-key ("KEK"). Pet., 64. In Petitioner's proposed Foster/Hayhurst combination, the encrypted content units would be, as in Hayhurst, distributed across thousands of subscribers. *Id.* A POSITA would not be motivated to do this because it would compromise the security of the system, as compared to Foster, by making Foster's KEKs more vulnerable while eliminating the ability (disclosed in Foster) to change the KEK as necessary.

Because Foster teaches attaching the title key used to encrypt content to the content unit, the title key itself must also be encrypted with a key-encrypting-key. *See* Ex. 1006 [Foster] [0053]. Critically, however, the key-encrypting-key may be replaced if, *e.g.*, it is compromised. As Dr. Rubin explains:

> In Foster, should a key-encrypting-key be compromised or "protection on the content [need] to be varied," "an encrypted title key 30 can be re-encrypted with a different key encrypting key 32 without having to re-encrypt the encrypted content 26." Ex. 1006 [Foster] [0054].

'854 Paper 21 at 58-59; '116 POR, 64

Reply

SFI cites Foster, [0053]-[0054] but that only says Foster's key "**can** be re-encrypted with a different" KEK. Contrary to SFI's arguments (POR, 59), Foster does not **require** changing the KEK. Rather, Foster recognizes "re-encrypting the content itself," rather than changing the KEK, is an alternative. Foster, [0053]-[0054]. Bhattacharjee-Reply, ¶¶106-109.

'854 Paper 24 at 11; '116 Reply, 27-28

Foster

[0053] Referring to **FIG. 3**, the encryption of content **22** using title key **24** is depicted in greater detail. As shown, content **22** is encrypted with title key **24** to yield encrypted content **26**. Then, title key **24** is itself encrypted with key encrypting key **28**. The encrypted title key **30** can then be transmitted along with the encrypted content **26** to a recipient.

[0054] **FIG. 4** demonstrates that an encrypted title key **30** can be re-encrypted with a different key encrypting key **32** without having to re-encrypt the encrypted content **26**. As shown, encrypted key **30** is received and decrypted with key encrypting key **28**. Title key **24** is then re-encrypted with a new key encrypting key **32**. The re-encrypted title key **34** is then transmitted along with the undisturbed encrypted content **26**. By re-encrypting title key **24**, the protection on the content can be varied without having to re-encrypt the content itself, which could be significantly more time-consuming and/or consume more resources than re-encrypting title key **24**.

'116 EX1006, [0054] (cited '116 Reply, 27-28; '854 Reply, 11)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Foster-Hayhurst: SFI's Argument #4 ("less secure") fails because changing a KEK is not claimed or disclosed by SFI's patents or required by Foster

Dr. Bhattacharjee

106. In my opinion, POSAs would have also disagreed with SFI's argument that the Foster-Hayhurst combination is less secure than Foster's unmodified system. POR, 58-61. That argument is premised on the idea that (**i**) changing or replacing Foster's key-encryption-key ("KEK") would have been more difficult in Foster-Hayhurst because the KEK would have (according to SFI) been located in more than one storage location, and (**ii**) Foster-Hayhurst would have stored the in more locations, which SFI alleges would have made it more vulnerable to attack. POR, 58-61.

107. As to SFI's first allegation regarding the need to update or replace the KEK, I note that the claims do not have any limitations directed to updating or replacing any of the encryption keys, in particular the *external key* that is met by Foster's KEK under SFI's claim constructions. The Foster-Hayhurst combination that I described in my First Declaration did not require updating or replacing the KEK as part of the combination, and thus the combination would not have had to update or replace the KEK. Thus, even if it were true that Foster by itself preferred to have the ability to update / change the KEK, that was not a feature that POSAs would have needed in Foster-Hayhurst for the combination's system to have worked properly, and thus it would not have discouraged POSAs from pursuing the combination. Moreover, the combination had other benefits over Foster alone that would have also motivated the combination (e.g., efficiency, intelligent content delivery, enhanced security from distributed storage, etc.).

108. Furthermore, Foster itself does not **require** updating or replacing the KEK. Instead, Foster explicitly says the KEK "**can** be" (i.e., does not **have** to be) changed in Foster's system. Foster, [0054]. Thus, to the extent SFI is arguing Foster (by itself) required the ability to change its KEK, POSAs would disagree as Foster says otherwise.

109. Foster also recognizes that KEKs "are inherently more secure than title keys, and **do not need to be changed as frequently**." Foster, [0055]. That disclosure in Foster further reinforces my opinion that the ability to change / update the KEK would not have dissuaded POSAs from pursing Foster-Hayhurst because the fact that the KEK did not have to be changed frequently meant it would have been an easier problem to solve compared to an encryption key that had to be changed more frequently.

110. But even though changing and updating the KEK is not required by Foster, Foster-Hayhurst, **or** the challenged claims, POSAs would still disagree with SFI's argument because POSAs would have known how to update and/or replace KEKs that have been distributed across many devices/users. In addition, POSAs would have recognized that whether the encrypted content was stored in one location (as in Foster alone) or across distributed storage devices (as in Foster-Hayhurst), changing the KEK would have to be propagated across the same number of segments. Accordingly, there would be no meaningful difference (let alone insurmountable different requiring more than ordinary skill) to POSAs in complexity between Foster's updating of the KEK and Foster-Hayhurst's updating the KEK. (I note, however, that as I discuss below, Foster-Hayhurst would not have stored the KEK on every storage device in the system anyway.)

'854 EX1111, ¶¶106-110 (cited Reply, 11)
'116 EX1111, ¶¶159-163 (cited Reply, 28)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Foster-Hayhurst: SFI's Argument #4 ("less secure") fails because POSAs knew how to change KEKs in a distributed environment like Foster-Hayhurst

Reply

Moreover, POSAs knew changing a KEK in Foster-Hayhurst would be comparable to doing so in Foster alone because both would involve changing a KEK used for multiple segments. Furthermore, POSAs knew how to change/replace KEKs across many devices/users. Bhattacharjee-Reply, ¶¶104-105, ¶¶110-115 (citing EX1078, 4:5-28; EX1079, 544-546; EX1080, 2-7).

'854 Paper 24 at 11-12;'116 Reply, 28

SFI's Sur-reply

Petitioner also argues, without explanation, that "POSAs knew how to change/replace KEKs across many devices/users." Reply, 11-12 (citing new Exhibits 1078-1080). This argument exemplifies why Petitioner's unexplained citations to new references should be rejected. Aside from being untimely, none of the newly cited exhibits involve changing/replacing KEKs, much less across many devices/users. *See* Ex. 1078, 4:5-28 (concerning information delivery method in peer-to-peer network, not KEKs); Ex. 1079, 544-546 (replacing session keys, not KEKs); Ex. 1080, 2-7 (replacing traffic encryption keys, not KEKs).

'854 Paper 32 at 8-9; '116 Sur-Reply, 19

EX1079

to groups of users. Multicast communications is efficient since it reduces the demands on network and bandwidth resources. It will play a key role in delivering services shared by many users, such as pay-per-view broadcasts of sporting events, as well as allowing for interactive multimedia applications such as interactive television, video conferencing, and communal gaming.

II. KEY MANAGEMENT IN MULTICAST COMMUNICATIONS

* * *

Due to the dynamic nature of the group, and the possible expiration of keying material, it is necessary to update both the SK and KEKs using rekeying messages. The three operations involved are key refreshing, key updating when a new user joins the service, and key updating when a user departs the service.

EX1079, 544-46 (cited '854 Reply, 11-12; '116 Reply, 28)

EX1080

In a broadcast-based network, multicast traffic will become the norm rather than the exception. Random browsing is supported in unicast, via the broadcast channel for large files, and via the narrowband circuit for delay-sensitive and low-volume traffic. Multicast traffic comprises the main volume, and takes two forms – streaming for all real-time multipoint content, and offline for all file transfer content. Any web site may have a combination of all three – for example browsing

Two forms of multicast traffic are defined in the pushed Internet – streaming and offline. Streamed content may be live (and usually has a limited shelf-life), or may be playout of pre-recorded content. It is intended to be used in real time, that is, at the time of reception only. If user recording of streamed content is an issue for content owners, they need to examine options to manage this. Offline multicast content comprises files for local storage and use some time after reception. Security of offline content is

To send the TEK securely it must itself be encrypted, and for this an agreement is made between the source and each user on a Key Encryption Key (KEK). In order to manage the various problems involved in large multicasts, the KEK is part of a hierarchy of frequently-changed keys, but is based on unique information about each user subscription known to the source. Protection of subscriber information is important but is outside the scope of this paper.

Solutions to the various complications of large-scale secure multicast are discussed below:

EX1080, 2-6 (cited '854 Reply, 11-12; '116 Reply, 28)

Foster-Hayhurst: SFI's Argument #4 ("less secure") also fails because changing KEKs in Foster-Hayhurst would have been comparable to changing KEKs in Foster alone

'116 Reply

Moreover, POSAs knew changing a KEK in Foster-Hayhurst would be comparable to doing so in Foster because both would involve changing the KEK used for multiple segments. Bhattacharjee-Reply, ¶163. Independently, POSAs also knew how to change/replace KEKs across many devices/users. Bhattacharjee-Reply, ¶¶164-168 (citing EX1078, 4:5-28; EX1079, 544-546; EX1080, 6-7).

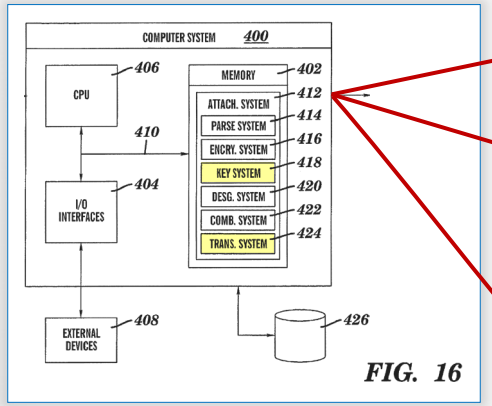
'116 Paper 21 at 28; '854 Reply, 11-12

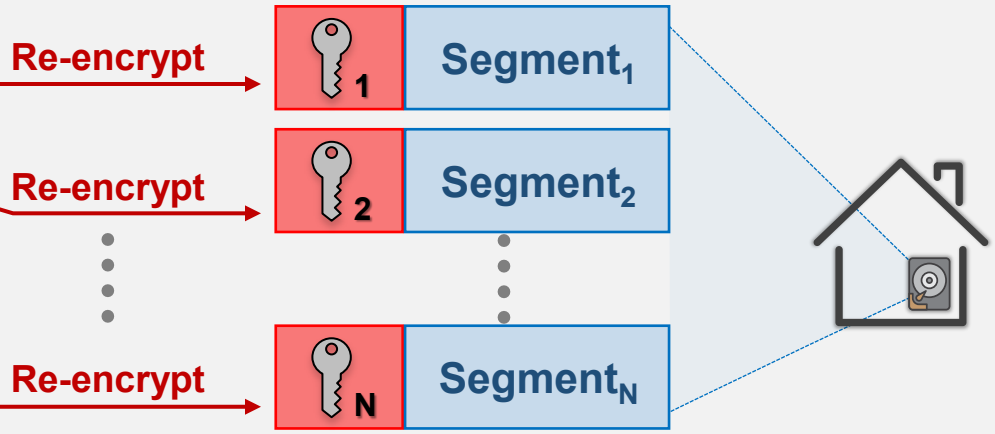
SFI's Sur-Reply

DOES NOT DISPUTE

'116 Paper 21 at 17-20;
'854 Sur-Reply, 5-10

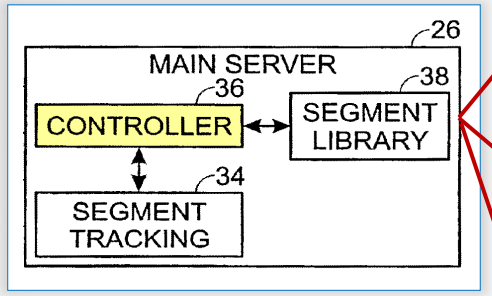
Foster

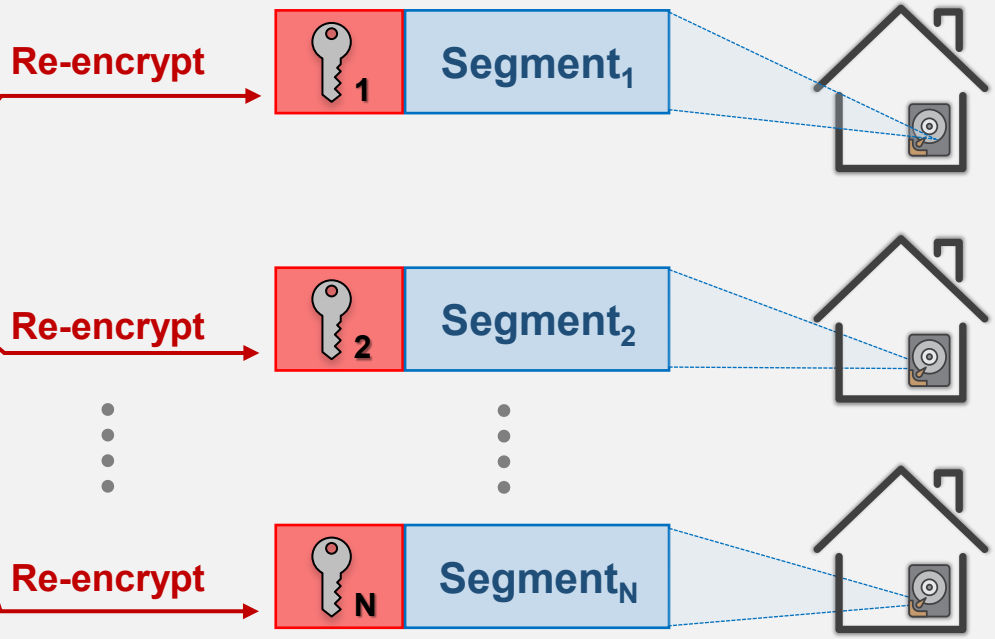




EX1006, FIG. 16 (partial) (cited '116 Pet., 53; '116 Reply, 27-28; '854 Pet. 69; '854 Reply, 11-12)

Foster-Hayhurst





EX1007, FIG. 1 (partial)
(cited '116 Pet., 56; '116 Reply, 27-28;
'854 Pet. 60; '854 Reply, 11-12)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116/’854 Foster-Hayhurst: SFI’s Sur-Reply argument fails because Foster-Hayhurst has undisputed benefits over Foster alone, and tradeoffs are commonplace

’116 Sur-Reply

Ultimately, “[t]he Board ***must*** weigh the benefits and drawbacks of the modification against each other, to determine whether there would be a motivation to combine.” *Arctic Cat Inc.* v. *Polaris Indus.*, 795 Fed. App’x 827, 833 (Fed. Cir. 2019); POR 56-57, 35 (citing additional cases). But Petitioner does not perform any such weighing in its Reply, and does not dispute Patent Owner’s showing that the numerous drawbacks described in the POR outweigh the benefits alleged in the Petition. *See* Reply, 24-29.

’116 Paper 29 at 20; ’854 Sur-Reply, 10

Reply

But even if Foster-Hayhurst were less secure than Foster in ways SFI alleges (it is not), Foster-Hayhurst added a security technique that Foster lacks (distributed storage) ***and*** provided the other (undisputed) benefits (e.g., speed, intelligence, efficiency, and redundancy) that also motivated Foster-Hayhurst. Petition, 62-63 (citing EX1028-EX1031); Bhattacharjee-Reply, ¶¶174-176 (citing corroboration); *Corephotonics*, 2021 WL 4944471, at *6 (“design decisions entail making tradeoffs among multiple objectives”).

’116 Paper 21 at 29; ’854 Reply, 12

Dr. Bhattacharjee

174. But even if Foster-Hayhurst were less secure than Foster in some of the ways SFI asserts, in my opinion POSAs would have been motivated to pursue Foster-Hayhurst for the reasons I identified in my First Declaration because POSAs would have recognized that the combination would have added various security techniques that Foster lacks, as my First Declaration discussed (and I reiterated above and further below).

175. Separate from improvements in security that I reiterated above, Foster-Hayhurst would have **also** provided other (undisputed) benefits in terms of increased speed, intelligence, efficiency, bandwidth requirements, as I discussed in my First Declaration (¶¶242-246) and above (§VI.B.1). Those benefits would have also motivated POSAs to pursue Foster-Hayhurst, with a reasonable expectation of success. Indeed, POSAs knew that designing an encryption and storage system for encrypted content was a matter of tradeoffs:

> **Data protection** in storage management systems is **a tradeoff** between user’s convenience and security, speed of operation and capabilities of the encryption algorithm, length of the encryption keys, government restrictions, and other elements known in the art. There are many encryption algorithms available that vary by strength, speed and other parameters.

’116 EX1111, ¶¶174-75 (cited Reply, 29); ’854 EX1111, ¶¶121-122 (cited Reply, 12-13)



“[I]t is a commonplace fact that design decisions entail making tradeoffs among multiple objectives. *Allied Erecting and Dismantling Co. v. Genesis Attachments, LLC*, 825 F.3d 1373, 1381 (Fed. Cir. 2016) (‘A given course of action often has simultaneous advantages and disadvantages, and this does not necessarily obviate motivation to combine.’).”

Corephotonics, Ltd. v. Apple Inc., 2021 WL4944471, at *6 (Fed. Cir. Oct. 25, 2021) (cited ’116 Reply, 29; ’854 Reply, 13)

’116/’854 Foster-Hayhurst: SFI’s Sur-Reply argument fails because the Foster-Hayhurst combination does not deliver a KEK until a user is authorized to view content

’116 Sur-Reply

Petitioner does not dispute that Foster-Hayhurst, by distributing KEKs among thousands of subscribers, would compromise the security of the combination as compared to Foster alone. POR, 64-67. In reply, Petitioner argues for the first time that “Foster-Hayhurst does not distribute the KEK to user devices” because “[a]s in Foster alone, Foster-Hayhurst would only give access to a KEK once authorized to decrypt the content.” Reply, 28 (citing EX1006, [0049], [0052]). Petitioner, however, relies on Foster’s description of prior art, rather than Foster’s system. EX1006 [Foster] [0049], [0052] (describing prior-art DRMS); *see also* Section II.B.3. In Foster’s actual invention, the KEK is delivered to the user device “prior to[] the encrypted content.” *Id.*, [0056]. Moreover, even if (contrary to Foster’s teachings), Foster-Hayhurst only delivered the KEK to authorized users, that would still leave the KEKs far more exposed. It is undisputed that in these circumstances, Foster’s KEKs can no longer be unique to a user and are at most unique to a movie. POR, 65-67. Thus, rather than having only one copy of a KEK exposed, as many KEK copies are exposed as there are reassembled content files. *Id.*

’116 Paper 29 at 19-20; ’854 Sur-Reply, 9

’116 Reply

SFI’s third critique—that Foster-Hayhurst is less secure than Foster—improperly considers the references individually and is premised on the mistaken idea that (i) changing/replacing the key-encryption-key (KEK) as taught by Foster is required by Foster and would be more difficult if stored in multiple locations, and (ii) storing the KEK in more locations makes it more vulnerable.

’116 Paper 21 at 27; ’854 Reply, 11

Regardless, Foster-Hayhurst does not distribute the KEK to user devices. Rather, as in Foster alone, Foster-Hayhurst would only give access to a KEK once authorized to decrypt the content. Foster, [0049], [0052]. Based on Hayhurst, the “encryption key to unlock the movie” (Foster-Hayhurst’s KEK) would be stored on the “main server,” not at each subscriber. Hayhurst, [0007] (cited Petition, 57-58), [0021]; Bhattacharjee, ¶251 (cited Petition, 66); Bhattacharjee-Reply, ¶¶169-170.

Thus, Foster-Hayhurst does not diminish Foster’s security. *Contra* POR, 64-68. Rather, using Hayhurst’s distributed storage in Foster “provides a ***high level of security***.” Hayhurst, [0008]. Hayhurst reduces unauthorized access in Foster because an attacker must collect multiple chunks from multiple storage devices, rather than from one storage device. Petition, 57-58 (citing Hayhurst, [0005], [0008]); Bhattacharjee-Reply, ¶171-173; Rubin-Depo., 125-130.

’116 Paper 21 at 28; ’854 Reply, 12

Dr. Bhattacharjee

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.

’116 EX1003, ¶251 (cited Pet. 66)
’854 EX1003, ¶188 (cited Pet., 64)

’116/’854 Foster-Hayhurst: Petitioner’s Reply did not change the combination to rely on Foster’s storage – it pointed to Foster’s storage context only to support a reason to combine with Hayhurst

SFI’s Sur-Reply

Second, implicitly admitting that a POSITA would not be motivated to make its originally-proffered combination, Petitioner shifts to a new combination, arguing for the first time that Foster-Hayhurst is limited to Foster’s alleged “storage context,” something not even mentioned in the Petition. In the Petition,

’854 Paper 32 at 5; ’116 Sur-Reply, 21

Reply

B. POSAs Had Motivation For Foster-Hayhurst

The Petition established POSAs had reason to implement Foster’s system using Hayhurst’s distributed-storage technique because it would, e.g., reduce a content provider’s bandwidth requirements and increase security by storing portions of a *data set* in different storage devices. Petition, 61-65.

In response, SFI argues Hayhurst’s distributed-storage technique is incompatible with Foster’s goal of “avoiding delays” for “streaming live television.” POR, 42. But Foster must be “read for **all** that it teaches.” *In re Mouttet*, 686 F.3d 1322, 1331 (Fed. Cir. 2012). Foster’s teachings are not limited to live-streaming, as Foster teaches **storing** encrypting content for and by recipient(s) (*e.g.*, video-on-demand). Foster, [0064]. In (at least) Foster’s storage context—relied upon in the Petition (65, 80-82)—POSAs would have recognized the benefits of combining Foster with Hayhurst. Bhattacharjee-Reply, ¶¶48-57.

’854 Paper 24 at 6; ’116 Reply, 24

Petition

source 428 and recipient 430, POSAs understood Foster to be disclosing that system 400 can perform functions of the “Content Service Provider” that is acting as a “pipeline” to deliver data from content owner 428 to recipients 430 (*e.g.*, residential customers that rent digital movies from cable (content service) providers). Bhattacharjee, ¶177.

’854 Paper 2 at 58-59; ’116 Pet., 53-54

POSAs understood that Foster’s content units are individually accessible and agnostic to where and how they are stored compared to other content units, and thus were compatible with being stored on multiple, separate storage devices (like Hayhurst’s subscriber storage devices). Bhattacharjee, ¶¶188-89; Foster, [0040]-

* * *

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. Bhattacharjee, ¶190.

’854 Paper 2 at 64-65; ’116 Pet., 67

As illustrated in Foster’s Figure 8 (color-annotated below), Foster’s method generates and stores a “processed content unit 116,” which includes three pieces of

’854 Paper 2 at 80; ’116 Pet., 69, 72-74

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116/’854 Foster-Hayhurst Grounds: The Parties’ disputes are narrow

- No dispute that Foster-Hayhurst meets all but “data chunks” limitations (’116 limitations [1A] and [9A]; ’854 limitations [1B]-[1B1])

- **Only disputes are whether:**
  (1) Hayhurst is analogous art to the ’116 and ’854 patents;
  (2) POSAs had motivation to combine Foster & Hayhurst; and
  (3) Foster discloses “data chunks” / “shares”**

*** SFI raises the same dispute for ’609 Ground 2 (Foster) regarding the claimed “shares”**

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Foster: The claimed “data chunk” (’116 | ’609) and “share” (’854) are just the “the thing that” the data set “gets split into”

’854 Patent, Claim 1

[1B] generating a plurality of data chunks based on the data set, such that the data set can be reconstructed using at least a minimum number of the plurality of data chunks, wherein generating the data chunks comprises:

[1B1] distributing the data set into a plurality of shares, wherein each of the shares comprises less than all of the data set,

’116 Patent, Claim 1

[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;

’609 Patent, Claim 1

[1C1] generating a plurality of data chunks based on a data set, wherein each data chunk of the plurality of data chunks comprises less than an entirety of data of the data set, and wherein the data set can be reconstructed using at least a minimum number of the plurality of chunks;

SFI’s Dr. Rubin

Q. So a – what does the term “share” represent?

A. So the share is the thing that something gets split into.

EX1099, 17 (cited ’854 Reply, 14; ’609 Reply, 4; ’116 Reply, 30)

Q. Is it your opinion that Claim 1 of the ’854 patent uses the term “share” and “data chunk” interchangeably?

A. I -- I don’t believe I set out any opinions specifically on that in my report, but that is how I read these claims -- this claim.

Q. So in your opinion, the Claim 1 uses two different terms referring to the same thing?

A. I wouldn’t say that. In order to provide, you know, that type of claim instruction information, I would want to study this some more, but, in essence, it's referring to **the pieces that have been split off from the original data set when it's referring to data chunks and when it’s referring to shares.**

EX1099, 158 (cited ’854 Reply, 14; ’609 Reply, 4; ’116 Reply, 30)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Foster has “data chunks” / “shares” because Foster’s digital content (data set) is split into payloads that each comprise less data than the data set

SFI’s Dr. Rubin

Q. So a – what does the term “share” represent?

A. **So the share is the thing that something gets split into.**

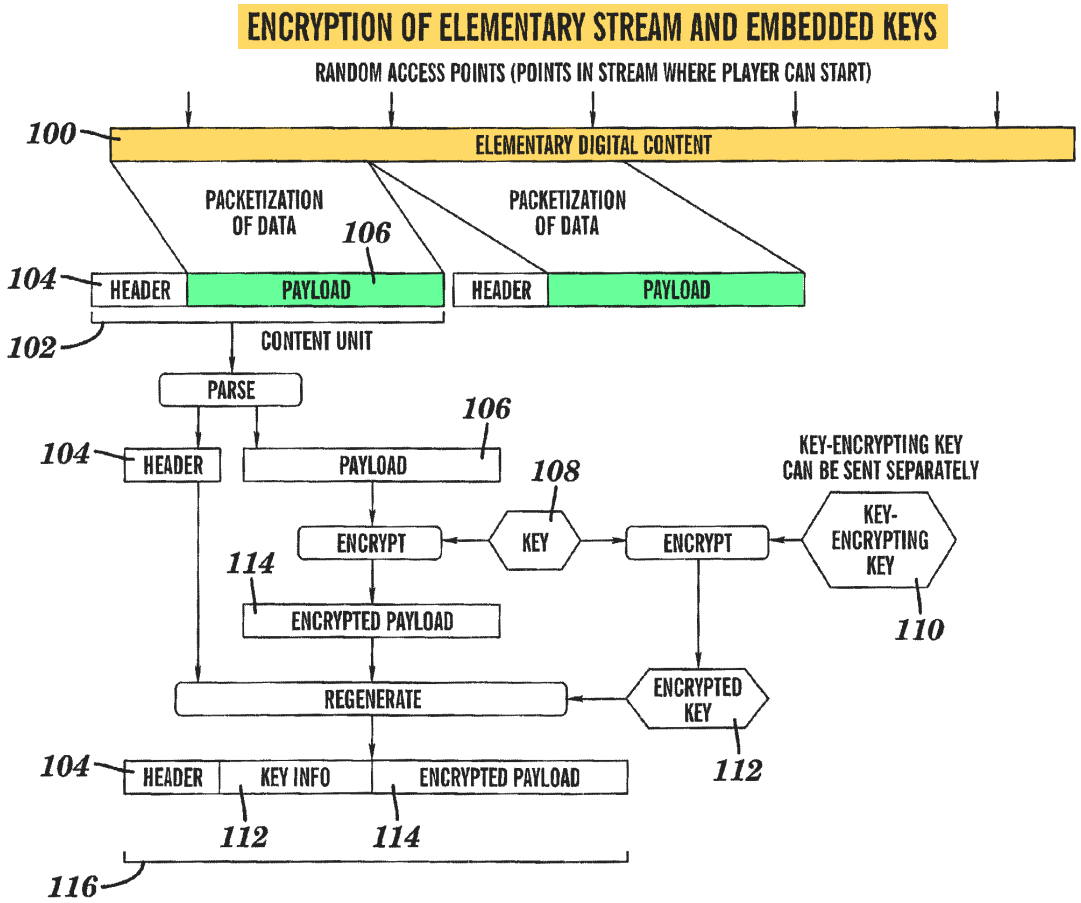
EX1099, 17 (cited ’854 Reply, 14; ’609 Reply, 4; ’116 Reply, 30)

Foster

[0064] Referring now to **FIG. 8**, the encryption and attachment of a title key to encrypted content under the present invention is shown. As depicted, an elementary content stream **100** comprising of compressed content is parceled into content units **102**. Each content unit includes a header **104** and a content packet **106**. Header **104** includes

’116 EX1006, [0064] (cited ’116 Pet., 52; ’854 Pet., 56; ’609 Pet., 57)

Foster



’116 EX1006, FIG. 8 (annotated) (cited ’116 Pet., 51; ’854 Pet., 56; ’609 Pet., 57)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Foster: SFI's "packets" argument fails because a POSA understands that Foster's "packets"/"payloads" meet the claimed "data chunks"

SFI's POR

when encrypted). Certainly, Foster never refers to its packet payloads as "data chunks" or "shares." At best, Petitioner's assertion is conclusory and does not

'854 Paper 21 at 66; '116 POR, 20-21; '609 POR, 37

deemed a chunk. As Dr. Rubin explains, the literature frequently refers to a packet or packet payload as something that is divided into "chunks" or "shares":

'854 Paper 21 at 66; '116 POR, 20; '609 POR, 37

This usage, wherein a packet is divided into data chunks is, moreover, reflected in the '854 patent itself. Per Dr. Rubin:

> Consistent with this usage in the literature, the '854 identifies a data packet (or packet payload) as something that is divided into shares. Ex. 1001 ['854] 82:21-24 ("At step 4106, ***the data packet or data block may then be partitioned into two or more data shares*** ..."), 81:36-37 ("***Packets of data (e.g., network packets) may also be efficiently split*** and restored"), 81:54-58 ("…simplified ***data partitioning process 4010*** … ***may be performed on network packets***, …"). Therefore, the '854

'854 Paper 21 at 67-68; '116 POR, 21-22; '609 POR, 38



"The prior inventor does not need to…conceive of its invention using the same words as the patentee would later use to claim it."

Teva Pharm. Indus. Ltd. v. AstraZeneca Pharms. LP, 661 F.3d 1378, 1384 (Fed. Cir. 2011) (cited '854 Reply, 16; '609 Reply, 5)

Reply

It is also irrelevant that some non-prior-art "literature" says a "payload" or "packet" can be "divided into 'chunks' or 'shares.'" POR, 66.[4] As discussed, neither the challenged claims nor the specification preclude a payload packet from (**i**) being a single *share* (or *chunk* when encrypted) or (**ii**) being divisible into multiple *shares*/*chunks*. The patent only requires that a *share* contain less than the entire *data set*. Foster's payloads are each less than Foster's entire *data set*. Bhattacharjee-

'854 Paper 24 at 16; '116 Reply, 29-31; '609 Reply, 5

EX1082

3.2.1 Transport Stream

MPEG video, audio and data streams are usually combined together to form a *Transport Stream*. A Transport Stream consists of short, fixed-length (188 byte) *packets* concatenated together. Each packet contains a *header* followed by a *payload* which is a chunk of a particular elementary stream.

EX1082, 8 (cited, e.g., '854 Reply, 16 n.2)
'854 Reply, 16 n.4 (citing also EX1083-1085); '116 Reply, 31)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Foster: SFI's "packet" argument also fails because cited literature that SFI ignores says a "packet" can be a "chunk" of a data set

EX1082

3.2.1 Transport Stream

MPEG video, audio and data streams are usually combined together to form a *Transport Stream*. A Transport Stream consists of short, fixed-length (188 byte) *packets* concatenated together. Each packet contains a *header* followed by a *payload* which is a chunk of a particular elementary stream.

EX1082, 8 (cited, e.g., '854 Reply, 16 n.2)

EX1083

A packet is a chuck of data provided by the application program. UDP typically sends a single "application-level packet" as a single UDP packet. However, TCP may break a single application-level packet into multiple smaller TCP "segments", each of which is treated as a separate "packet" at the TCP layer. The Nagle Algorithm (discussed below) does the opposite: It takes multiple small application packets and combines them into a single larger TCP segment.

EX1083, 7:32-40 (cited, e.g., '854 Reply, 16 n.2)

EX1084

[0026] RTP packet: A chunk or portion of multimedia data as sent over an IP network, prepended with certain information, such as a time stamp and a sequence number.

EX1084, [0026] (cited, e.g., '854 Reply, 16 n.2)

EX1085

In one aspect of the invention, there is provided a method for data storage in a distributed data storage system with redundancy, including disassembling a data set into a plurality of data blocks; defining a number of distinct data chunks N; defining a minimal number of data chunks K≦N needed to restore one data block; transforming each of the data blocks into L data chunks, wherein K≦L≦N; and distributing the data chunks to storage elements of the distributed storage system. The storage elements may be different storage for each of data block, or for some of the data blocks. Filler can be added to any data block. Filler can be added to the data set so that the data set is an integral number of data blocks. Each data chunk can include an identifier derived from some function used for its generation. A size of each data chunk can be defined based on an integral number of Message Transfer Units. A size of each data chunk can correspond to packet size, or an integral number of packets. A number L of chunks is chosen such that L≧M+K, where M is maximum number of storage elements that can fail simultaneously.

EX1085, 4:7-25 (cited, e.g., '854 Reply, 16 n.2)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Foster: SFI's "securing data" argument fails because SFI's patents do not require the data chunks / shares to be split "as a means of protecting a data set"

SFI's POR

Nor does the '854's discussion of shares suggest that a packet payload would be deemed a "data share" or a "data chunk" when encrypted. As the '854 explains, partitioning the data set into chunks increases security because when "data ... remains whole and intact," it is "subject to attack." The '854 addresses this concern, in part, by "performing a cryptographic parsing and splitting of the encrypted file." Ex. 1001 ['854] 56:47-61. Thus, the '854 teaches that splitting the data set into shares helps secure the data.

Thus, the '854 splits a data set into shares or chunks as a means of protecting a dataset. This is why a packet in the '854 is an example of a data set that is split into shares. *See* Ex. 1001 ['854] 82:21-24. By contrast, Foster's packet payloads exist not to protect data but, rather, because it is just a by-product of a pre-existing standard. Consequently, a POSITA reading the '854 patent would not consider a packet payload to be a data share (or a data chunk when encrypted) in view of the '854's disclosure.

'854 Paper 21 at 68-69; '116 POR, 20-22; '609 POR, 40

'854 Patent

Traditional encryption technologies known in the art rely on one or more key used to encrypt the data and render it unusable without the key. The data, however, remains whole and intact and subject to attack. The secure data parser of the present invention, in one embodiment, addresses this problem by performing a cryptographic parsing and splitting of the encrypted file into two or more portions or shares, and in another embodiment, preferably four or more shares, adding another layer of encryption to each share of the data, then storing the shares in different physical and/or logical locations. When one or more data shares are physically removed from the system, either by using a removable device, such as a data storage device, or by placing the share under another party's control, any possibility of compromise of secured data is effectively removed.

'854 EX1001, 56:47-61

At step **4106**, the data packet or data block may then be partitioned into two or more data shares using one of the data splitting algorithms described above in accordance with the present invention. In some embodiments, the data packet or

'854 EX1001, 82:21-24

Foster: The Board already rejected SFI's "securing data" argument that generating chunks / shares must be for the purpose of protecting the data

'854 Patent

sizes. In addition, varying the sizes of the data units resulting from the splitting process may render the data more difficult to restore to a useable form, thereby increasing security of sensitive data. It is readily apparent to those of ordinary skill in the art that the split data unit sizes may be a wide variety of data unit sizes or patterns of sizes or combinations of sizes. For example, the data unit sizes may be selected or predetermined to be all of the same size, a fixed set of different sizes, a combination of sizes, or randomly generates sizes. Similarly, the data units may be distributed into one or more shares according to a fixed or predetermined data unit size, a pattern or combination of data unit sizes, or a randomly generated data unit size or sizes per share.

'854 EX1001, 21:25-37 (cited Reply, 14-15);
'116 EX1001, 21:59-22:4 (cited Reply, 30); '609 EX1001, 21:25-37 (cited Reply, 4)

Reply

Furthermore, Foster's payloads "are generated to further secure the data" (POR, 68-69), because each is encrypted with a different key to "provide **enhanced security**" (Foster, [0004], [0071]). Bhattacharjee-Reply, ¶147.

'854 Paper 24 at 15; '116 Reply, 30-31; '609 Reply, 5

Institution Decision

but are not chunks themselves. *Id.* at 43–46. Patent Owner argues that the purpose of generating the data chunks is to secure the data. *Id.* at 46.

Patent Owner mischaracterizes Petitioner's argument. Petitioner does not identify Foster's payloads as the claimed data chunks. *See* Prelim. Resp. 42–46. Rather, Petitioner identifies payload 106 as the claimed "shares" and *encrypted* payload 114 as the "data chunks." *See, e.g.*, Pet. 72 ("[T]he payloads ([1B1]'s '*shares*') are encrypted to generate encrypted payloads, which are '*a plurality of data chunks based on the data set*,' as claimed."), 73 ("POSAs had reason to distribute Foster-Hayhurst's '*data set*' into encrypted payloads ('*data chunks*') . . . .") (emphasis in original). The two payloads are not the same: Payload 106 is the result of packetizing the elementary digital content 100, and encrypted payload 114 is generated by encrypting payload 106. *See, e.g.*, Ex. 1006, Fig. 6. Thus, even if the claim requires generating data chunks to secure the data, as Patent Owner argues (Prelim. Resp. 45–46), the record at this stage shows that Foster secures the digital content by generating encrypted payload 114 (Pet. 72–73).

'854 Paper 15 at 21; '116 Paper 11 at 23-24; '609 Paper 16 at 27-28

Foster: SFI's '854 and '609 Sur-Replies (but not SFI's '116 Sur-Reply) still argue chunks / shares must be "generated to secure data"

SFI's Sur-Reply

Exs. 2012-2013, 2036-2038, 2041-2043). The POR further explained that the difference between the '854's chunks/shares and Foster's packets/payloads is significant as the '854's data chunks or shares are generated to further secure the data, and Foster's packet payloads merely exist as a by-product of the MPEG standard. *Id.*, 68-69.

In reply, Petitioner does not dispute the '854's clear teaching that packets or packet payloads themselves are split into chunks or shares. *See* Reply, 13-16. Petitioner ignores these portions and alleges that "SFI suggests a share must be 'cryptographically generated' so the splitting of the data set by itself makes it less susceptible to 'attack.'" *Id.*, 15. But, as seen above, this is not and has never been Patent Owner's position. Rather, the '854's specification gives several examples of how a dataset may be split and is clear that a packet or its payload is itself a dataset that is split into shares or chunks, regardless of whether the split is performed cryptographically. Ex. 1001 ['854] 82:21-24, 81:36-37, 81:54-58.

'854 Paper 32 at 17-19; '609 Sur-Reply, 3-4

one. *See id.* Petitioner contends that "Foster's payloads 'are generated to further secure the data' [], because each is encrypted with a different key." *Id.*, 15. But this misapprehends to purpose of packetization which, as both experts agree, is a part of the MPEG standard. Ex. 2046 [Bhattacharjee-2nd-Depo.] 34:2-20, 35:23-36:15 (MPEG packetization for "optimiz[ation]" such as "random access"). Petitioner, moreover, never argued that encryption is part of MPEG packetization. *See* Reply, 15. Indeed, Petitioner argued that Foster's encryption of the shares teaches "encrypting each of the shares" rather than "distributing the data set into a plurality of shares." Pet., 74-75, 77. Petitioner thus fails to rebut the POR's showing that MPEG packets are not generated to secure data. POR, 68-69.

Finally, Petitioner argues that the POR's literature demonstrating a known distinction between packets and/or their payloads and chunks/shares is not relevant as they do not relate to MPEG packets. Reply, 16. But the POR's literature discusses data packets and MPEG packets are a type of data packet. Ex. 2011 [Tsai] [0001]. Petitioner fails to provide any evidence showing that an MPEG packet/payload or any other type of packet/payload is terminology used to refer to a share/chunk. The POR's showing is thus unrebutted.

Reply

[4] Other literature says a packet can contain a chunk. EX1082, 8; EX1083, 7:32-40; EX1084, [0026]; EX1085, 4:7-25.

'854 Paper 24 at 16 n.4; '116 Reply, 31

Foster: SFI's "securing data" argument fails because SFI's patents do not exclude a "packet" from being a "share" / "chunk"

SFI's POR

Consistent with this usage in the literature, the '854 identifies a data packet (or packet payload) as something that is divided into shares. Ex. 1001 ['854] 82:21-24 ("At step 4106, ***the data packet or data block may then be partitioned into two or more data shares*** ..."), 81:36-37 ("***Packets of data (e.g., network packets) may also be efficiently split*** and restored"), 81:54-58 ("…simplified ***data partitioning process 4010***

'854 Paper 21 at 67-68; '116 POR, 20-22; '609 POR, 38

'854 Patent

random numbers. The data may be split into any desired, selected, predetermined, or randomly assigned size unit, including but not limited to, a bit, bits, bytes, kilobytes, megabytes or larger, or any combination or sequence of sizes. In addition, varying the sizes of the data units resulting from the splitting process may render the data more difficult to restore to a useable form, thereby increasing security of sensitive data. It is readily apparent to those of ordinary skill in the art that the split data unit sizes may be a wide variety of data unit sizes or patterns of sizes or combinations of sizes. For example, the data unit sizes may be selected or predetermined to be all of the same size, a fixed set of different sizes, a combination of sizes, or randomly generates sizes. Similarly, the data units may be distributed into one or more shares according to a fixed or predetermined data unit size, a pattern or combination of data unit sizes, or a randomly generated data unit size or sizes per share.

'854 EX1001, 21:21-37 (cited '854 Reply, 15; '609 Reply, 4; '116 Reply, 30)

'854 Patent

At step **4106**, the data packet or data block may then be partitioned into two or more data shares using one of the data splitting algorithms described above in accordance with the present invention. In some embodiments, the data packet or

'854 EX1001, 82:21-24

a predetermined number of shares. Packets of data (e.g., network packets) may also be efficiently split and restored

'854 EX1001, 81:36-37

and a data partitioning phase. Simplified header generation process **4000** and simplified data partitioning process **4010** are shown in FIGS. **40**A and **40**B, respectively. One or both of these processes may be performed on network packets, file system blocks, or any other suitable information.

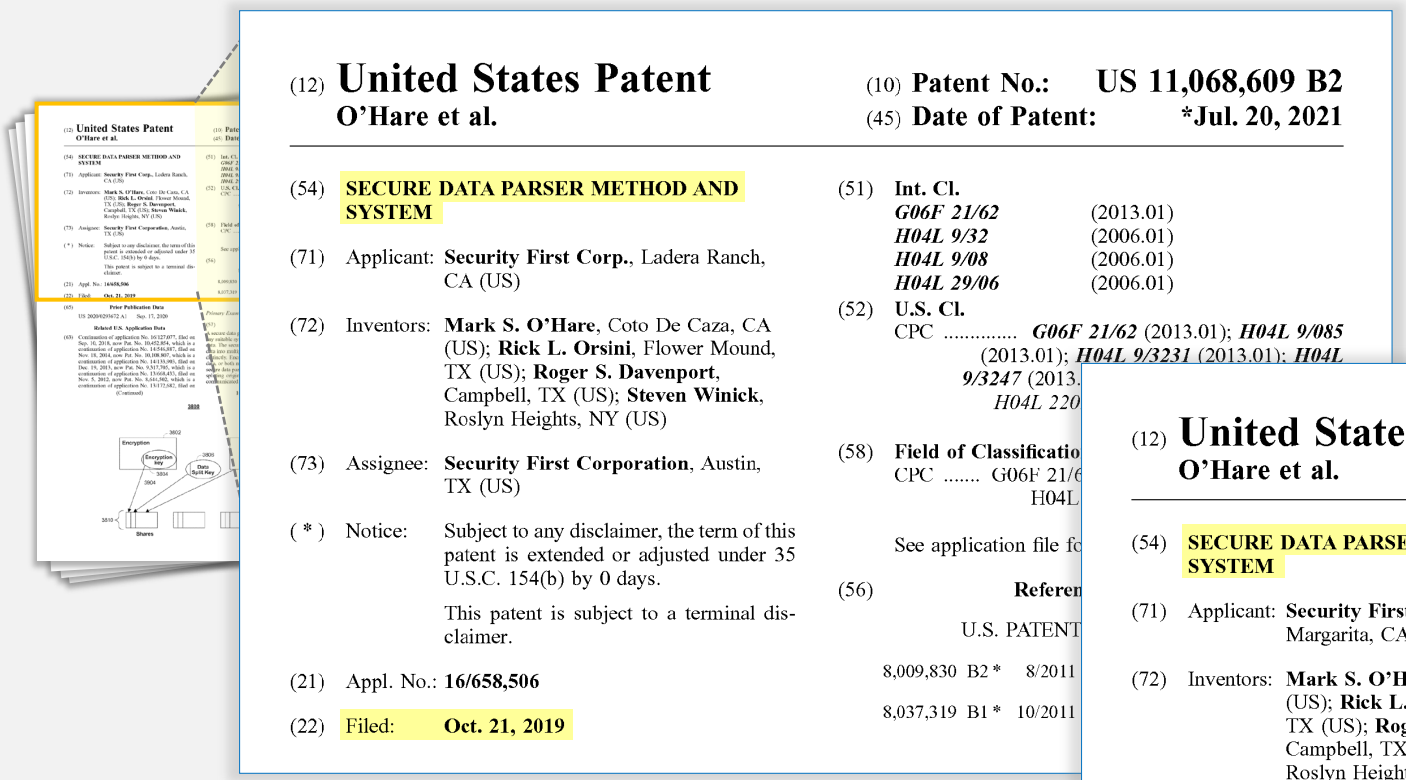
'854 EX1001, 81:54-58

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’609 and ’854 Patent

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Challenged U.S. Patent Nos. 11,068,609 & 10,452,854



(12) **United States Patent**
O'Hare et al.

(10) **Patent No.: US 11,068,609 B2**
(45) **Date of Patent: *Jul. 20, 2021**

(54) **SECURE DATA PARSER METHOD AND SYSTEM**

(71) Applicant: **Security First Corp.**, Ladera Ranch, CA (US)

(72) Inventors: **Mark S. O'Hare**, Coto De Caza, CA (US); **Rick L. Orsini**, Flower Mound, TX (US); **Roger S. Davenport**, Campbell, TX (US); **Steven Winick**, Roslyn Heights, NY (US)

(73) Assignee: **Security First Corporation**, Austin, TX (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **16/658,506**

(22) Filed: **Oct. 21, 2019**

(51) **Int. Cl.**
G06F 21/62 (2013.01)
H04L 9/32 (2006.01)
H04L 9/08 (2006.01)
H04L 29/06 (2006.01)

(52) **U.S. Cl.**
CPC ..... ***G06F 21/62*** (2013.01); ***H04L 9/085*** (2013.01); ***H04L 9/3231*** (2013.01); ***H04L 9/3247*** (2013.
H04L 220

(58) **Field of Classificatio**
CPC G06F 21/6
H04L

See application file fo

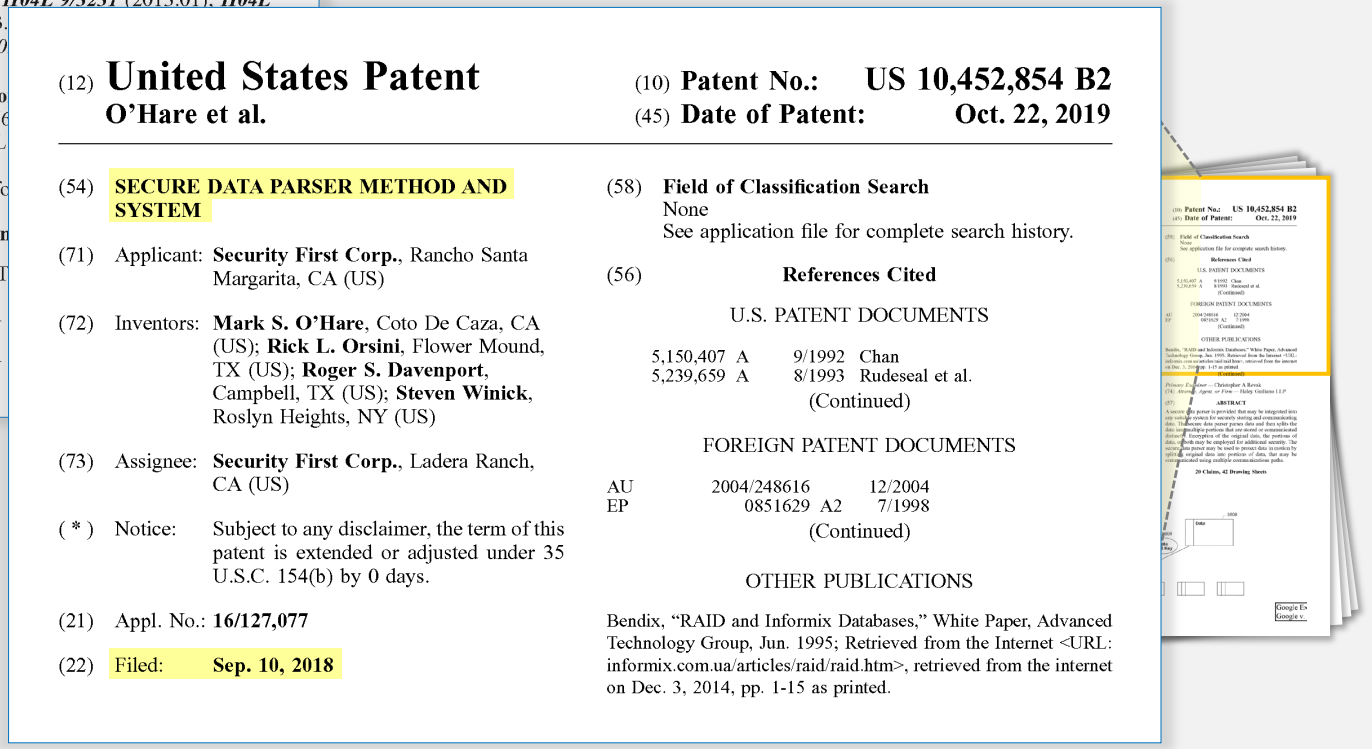
(56) **Referen**

U.S. PATENT

8,009,830 B2 * 8/2011

8,037,319 B1 * 10/2011

'609 Patent (EX1001)



(12) **United States Patent**
O'Hare et al.

(10) **Patent No.: US 10,452,854 B2**
(45) **Date of Patent: Oct. 22, 2019**

(54) **SECURE DATA PARSER METHOD AND SYSTEM**

(71) Applicant: **Security First Corp.**, Rancho Santa Margarita, CA (US)

(72) Inventors: **Mark S. O'Hare**, Coto De Caza, CA (US); **Rick L. Orsini**, Flower Mound, TX (US); **Roger S. Davenport**, Campbell, TX (US); **Steven Winick**, Roslyn Heights, NY (US)

(73) Assignee: **Security First Corp.**, Ladera Ranch, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **16/127,077**

(22) Filed: **Sep. 10, 2018**

(58) **Field of Classification Search**
None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,150,407 A 9/1992 Chan
5,239,659 A 8/1993 Rudeseal et al.
(Continued)

FOREIGN PATENT DOCUMENTS

AU 2004/248616 12/2004
EP 0851629 A2 7/1998
(Continued)

OTHER PUBLICATIONS

Bendix, "RAID and Informix Databases," White Paper, Advanced Technology Group, Jun. 1995; Retrieved from the Internet <URL: informix.com.ua/articles/raid/raid.htm>, retrieved from the internet on Dec. 3, 2014, pp. 1-15 as printed.

'854 Patent (EX1015)

Independent claims of the ’609 and ’854 patents

’854 Claim 1

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

[1A] receiving an external key from an external storage system;

[1B] generating a plurality of data chunks based on the data set, such that the data set can be reconstructed using at least a minimum number of the plurality of data chunks, wherein generating the data chunks comprises:

[1B1] distributing the data set into a plurality of shares, wherein each of the shares comprises less than all of the data set,

[1B2] accessing a plurality of distinct encryption keys,

[1B3] encrypting each of the shares with a respective one of the plurality of distinct encryption keys,

[1D] performing an encryption operation based on the external key to further secure the plurality of data chunks; and

[1E] storing with the plurality of data chunks data indicative of at least one of the distinct encryption keys on a plurality of different storage devices.

’609 Claim 1

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

[1A] executing code by a processor to perform:

[1B] receiving a first key from a storage system;

[1C1] generating a plurality of data chunks based on a data set, wherein each data chunk of the plurality of data chunks comprises less than an entirety of data of the data set, and wherein the data set can be reconstructed using at least a minimum number of the plurality of chunks;

[1D] encrypting each respective data chunk of the plurality of data chunks with a respective second key, wherein each of the respective second keys are distinct from each other;

[1E] performing a cryptographic operation based on the first key to further secure the plurality of data chunks; and

[1F] storing, in a memory coupled to the processor, at least one data chunk of the plurality of data chunks with data indicative of at least one of the distinct encryption keys on at least one storage device.

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’854 Patent – Claim 1 generates “shares”; encrypted shares are “data chunks”

’854 Patent, Claim 1

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

[1A] receiving an external key from an external storage system;

[1B] generating a plurality of data chunks based on the data set, such that the data set can be reconstructed using at least a minimum number of the plurality of data chunks, wherein generating the data chunks comprises:

[1B1] distributing the data set into a plurality of shares, wherein each of the shares comprises less than all of the data set,

[1B2] accessing a plurality of distinct encryption keys,

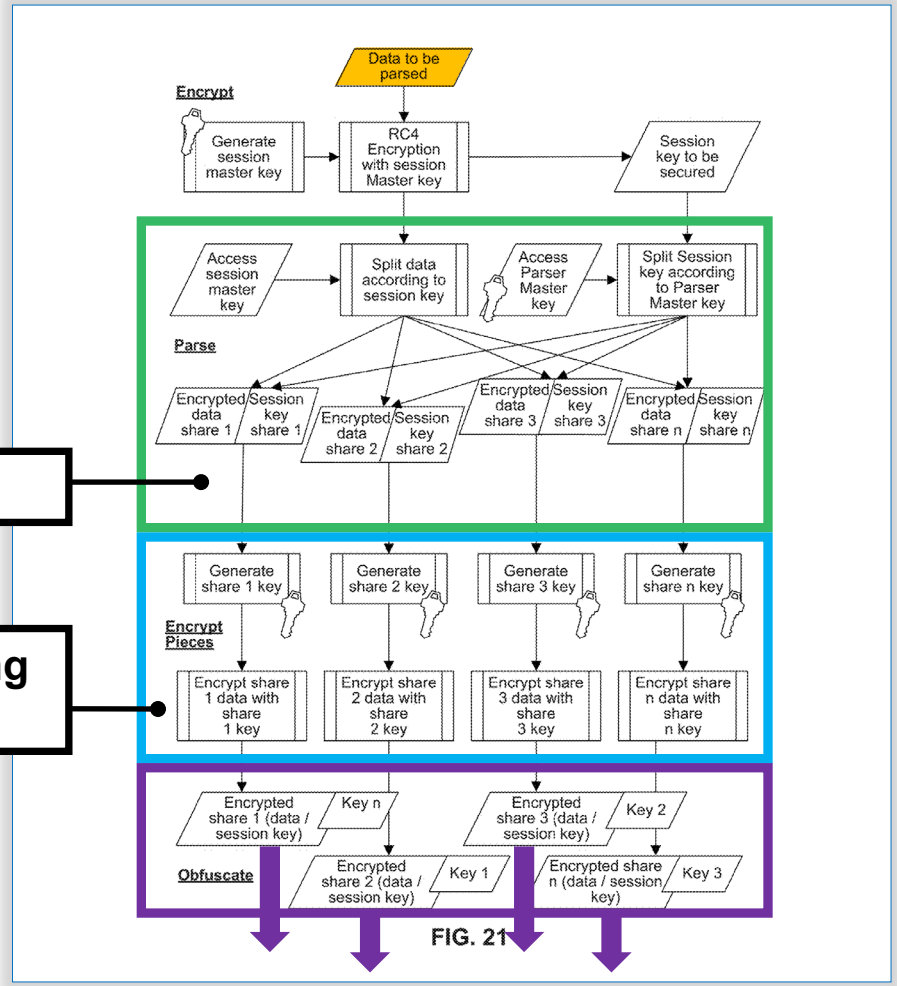
[1B3] encrypting each of the shares with a respective one of the plurality of distinct encryption keys,

[1D] performing an encryption operation based on the external key to further secure the plurality of data chunks; and

[1E] storing with the plurality of data chunks data indicative of at least one of the distinct encryption keys on a plurality of different storage devices.

“shares”

“encrypting shares”



’854 EX1001, FIG. 21 (annotated)
(cited ’854 Pet., 9)

’609 Patent – Claim 1 generates and encrypts “data chunks”

’609 Patent, Claim 1

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

[1A] executing code by a processor to perform:

[1B] receiving a first key from a storage system;

[1C1]-[1C2] generating a plurality of data chunks based on a data set, wherein each data chunk of the plurality of data chunks comprises less than an entirety of data of the data set, and wherein the data set can be reconstructed using at least a minimum number of the plurality of chunks;

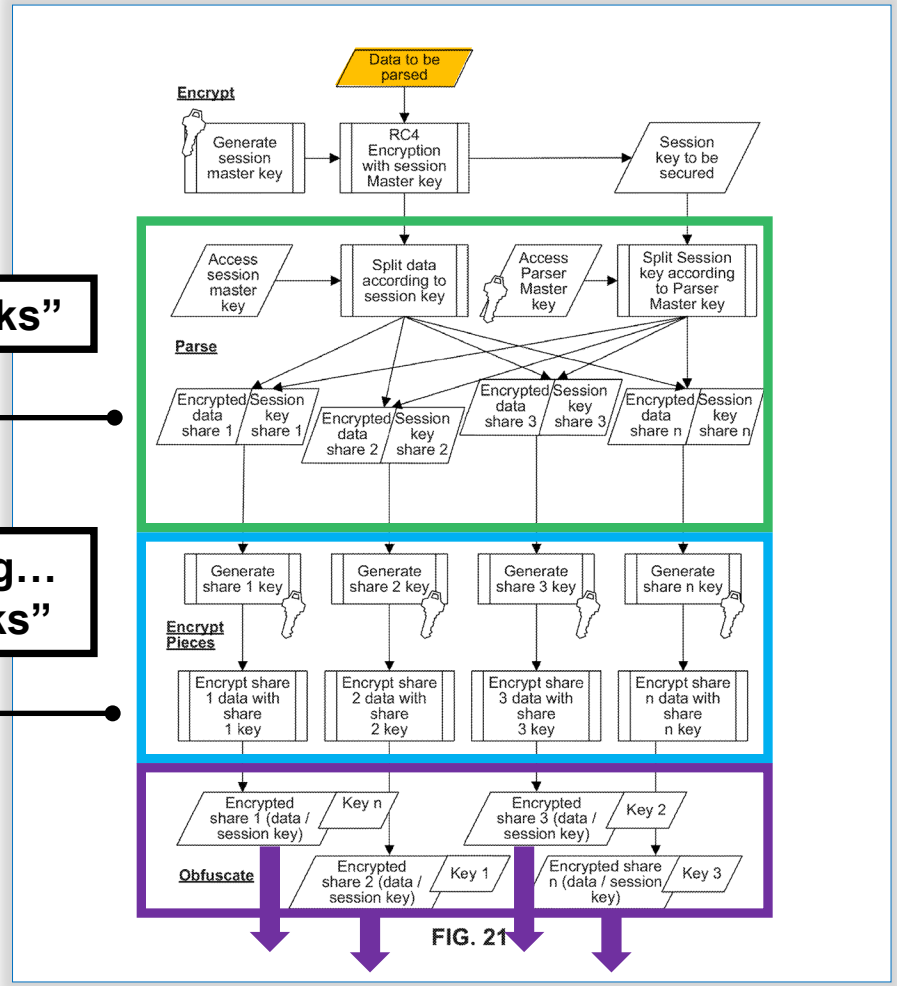
[1D] encrypting each respective data chunk of the plurality of data chunks with a respective second key, wherein each of the respective second keys are distinct from each other;

[1E] performing a cryptographic operation based on the first key to further secure the plurality of data chunks; and

[1F] storing, in a memory coupled to the processor, at least one data chunk of the plurality of data chunks with data indicative of at least one of the distinct encryption keys on at least one storage device.

“data chunks”

“encrypting… data chunks”



’609 EX1001, FIG. 21 (annotated)
(cited ’609 Pet., 10)

Agenda

1. Tajima-based Grounds: ’116 Grounds 1-2 (independent claim 1 + claims 5-8)
2. Foster-Hayhurst Grounds: ’116 Ground 5 (independent claims 1 & 9 + claims 2, 5, 11)
   ’854 Ground 2 (all challenged claims 1-10)
3. **Foster Ground: ’609 Ground 2 (all challenged claims 1-10)**
4. Orsini Grounds: ’854 Ground 1 (all challenged claims 1-10)
   ’609 Ground 1 (all challenged claims 1-10)
5. Tajima-Rivest-based Grounds: ’116 Ground 3 (independent claim 9 + claims 10-14)
   ’116 Ground 4 (independent claim 1 + claims 2-3, 6)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

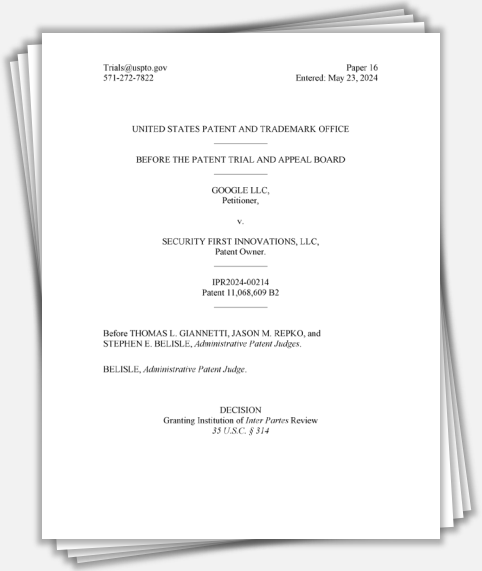
Overview of the ’609 Petition’s Grounds

’609 Patent

GROUND	CLAIMS	BASIS
1. Orsini (EX1005)	1-10	102/103
2. Foster (EX1006)	1-3, 7-8 4-6, 9-10	Obviousness/ Anticipation Obviousness

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’609 Ground 2 (Foster): At institution, the Board found that Foster anticipates and renders obvious independent claim 1 (and dependent claims 2-3, and 8)



Trials@uspto.gov
571-272-7822

Paper 16
Entered: May 23, 2024

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner,

v.

SECURITY FIRST INNOVATIONS, LLC,
Patent Owner.

IPR2024-00214
Patent 11,068,609 B2

Before THOMAS L. GIANNETTI, JASON M. REPKO, and STEPHEN E. BELISLE, *Administrative Patent Judges*.

BELISLE, *Administrative Patent Judge*.

DECISION
Granting Institution of *Inter Partes* Review
35 U.S.C. § 314

Institution Decision

D. Alleged Anticipation or Obviousness of Claims 1–3, 7, and 8 by or over Foster

Petitioner contends claims 1–3, 7, and 8 would have been unpatentable under 35 U.S.C. § 102 as anticipated by Foster (Ex. 1006), or, alternatively, under 35 U.S.C. § 103 as obvious over Foster. Pet. 3, 56–79, 83–84. Patent Owner opposes Petitioner’s contentions. Prelim. Resp. 36–66. Based on our review of the record before us, we determine that Petitioner has established a reasonable likelihood that it would prevail in showing that claims 1–3, 7, and 8 are unpatentable under 35 U.S.C. § 102 as anticipated by Foster, as discussed below.

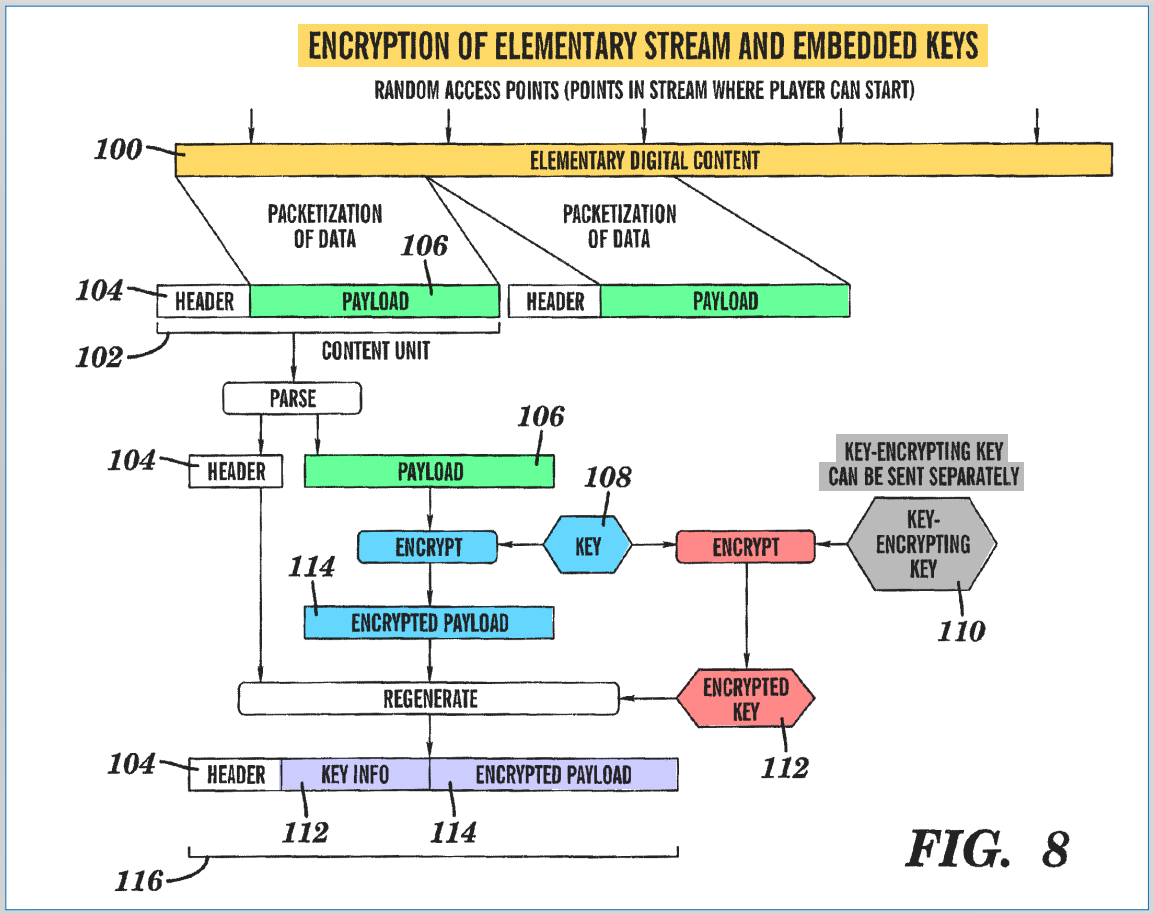
A fortiori, we also determine that Petitioner has established a reasonable likelihood that it would prevail in showing that claims 1–3, 7, and 8 would have been unpatentable under 35 U.S.C. § 103 as obvious over Foster. *See Cohesive Techs., Inc. v. Waters Corp.*, 543 F.3d 1351, 1364

’609 Paper 16 at 19

’609 Ground 2 (anticipation and obviousness): Mapping to claim 1 under the Petition’s view of claim scope for limitations [1A] and [1F]

	’609 Patent, Claim 1	Foster
[1PRE]	A method for securing data, the method comprising:	**[0064]** ... [A]n **elementary content stream 100** comprising of compressed content is **parceled into content units 102.** Each content unit includes a header 104 and a content packet 106.… Once parceled, header 104 of each content unit 102 is parsed from content packet 106 and **content packet 106 is encrypted with title key 108** to yield encrypted content packet 114. Then, **key encrypting key 110 is used to encrypt title key 108, which is then attached to content packet 106** along with header 104. The resulting processed content unit ’116 can then be transmitted to a recipient. That is, the encrypted content unit, the header and the header extension (including the encrypted title key) **will be delivered to, or stored by, a recipient** as a single, integral data stream. Thus, **a recipient will receive or store encrypted content** along will all information necessary to decrypt the content.
[1A]	executing code by a processor to perform;	
[1B]	receiving a first key from a storage system;	
[1C1]-[1C2]	generating a plurality of data chunks based on a data set, wherein each data chunk of the plurality of data chunks comprises less than an entirety of data of the data set, and wherein the data set can be reconstructed using at least a minimum number of the plurality of chunks;	
[1D]	encrypting each respective data chunk of the plurality of data chunks with a respective second key, wherein each of the respective second keys are distinct from each other;	**[0065]** … [E]ach content packet could be **encrypted with a different title key.**
[1E]	performing a cryptographic operation based on the first key to further secure the plurality of data chunks; and	
[1F]	storing, in a memory coupled to the processor, at least one data chunk of the plurality of data chunks with data indicative of at least one of the distinct encryption keys on at least one storage device.	

Foster



[0046] Key Management Block (KMB)—a data structure containing multiple encryptions of a key encrypting key, and that excludes non-compliant

’609 EX1006, [0046], [0064]-[0065], FIG. 8 (annotated) (cited ’609 Pet., 64)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’609 Ground 2 (anticipation & obviousness): Foster discloses a processor executing code to implement Foster’s encryption process

’609 Patent, Claim 1

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

[1A] executing code by a processor to perform;

…

[1F] storing, in a memory coupled to the processor, at least one data chunk of the plurality of data chunks with data indicative of at least one of the distinct encryption keys on at least one storage device.

Foster

[0074] It should be understood that the elements of FIGS. **1-15** used to receive/transmit transmissions, encrypt/decrypt content and/or keys, attach keys to content etc. can be implemented as hardware, software or as a combination of hardware or software. As such, any kind of computer/server system(s)—or other apparatus adapted for carrying out the methods described herein—is suited. A typical combination of hardware and software could be a general purpose (computer) system with a computer program that, when loaded and executed, carries out the methods described herein.

’609 EX1006, [0074] (cited ’609 Pet., 62)

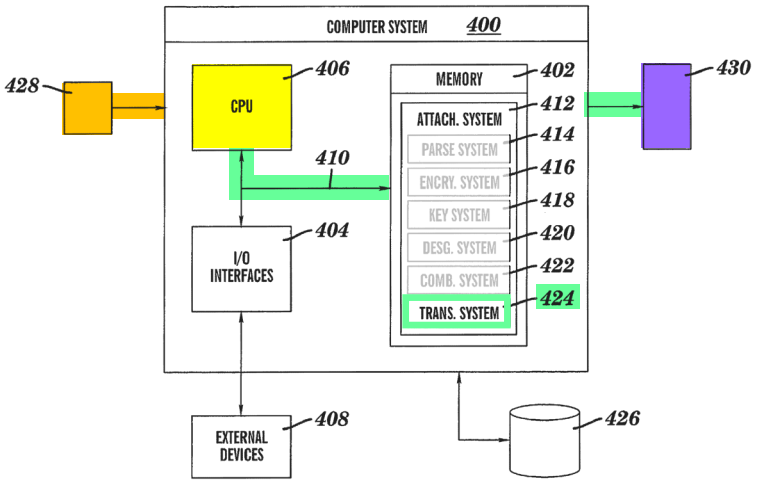


FIG. 16

’609 EX1006, FIG. 16 (annotated) (cited ’609 Pet., 62, 75-76)

Petition

2. [1A]

Foster “*execut[es] code by a processor to perform*” Foster’s method, as claimed.

Foster’s methods are implemented by executing “a computer program,” which comprises “program *code*,” using a computer system’s CPU (central processing unit), which is a “*processor*.” Foster, [0074]-[0075], FIG. 16 (annotated below), [0012] (“execut[ing]” “program code”); Bhattacharjee, ¶¶176-77.

’609 Paper 2 at 62

SFI’s POR

DOES NOT DISPUTE

See ’609 Paper 22 at 32-48

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’609 Ground 2 (anticipation): Foster also anticipates because Foster’s processor executes code that “results in storing” in the “memory” of storage devices “coupled to” the processor

Reply

SFI does not dispute that, as Petitioner argued, Foster’s processor “execut[es] ‘a computer program’” to implement Foster’s methods. Petition, 62 (quoting Foster, [0074]-[0075]). SFI’s expert testified that the claimed *code* need only “be code executed on the processor **that results in storing** something on – **in memory that’s coupled to the processor**.” Rubin-Depo, 189-191. *Infra* §I.D (discussing *coupled to* limitation). The Board properly found Foster’s processor does that because Foster’s methods store encrypted content in memories of storage devices (*e.g.*, on servers and at recipients) *coupled to* the processor. DI, 24-25; Petition, 72-76. Thus, Foster’s processor executes code that is “relat[ed] to storing content units on” (POR, 43) those storage devices. Bhattacharjee, ¶¶200-206; Bhattacharjee-Reply, ¶¶58-60.

’609 Paper 25 at 6

SFI’s Dr. Rubin

Q. So your opinion is that the processor in the claim has to instruct the storage device to store?

A. Well, I mean, **what the claim says**, executing code via processor to perform those first steps and then to perform storing in a memory coupled to the processor at least one data chunk, et cetera. So **it has to be code executed on the processor that results in storing something on -- in memory that’s coupled to the processor.** In Foster, the, you know, set-top box memory is not coupled to the processor.

’609 EX1099, 190 (cited Reply, 6)

SFI’s POR

generally comprises,” not including recipient 430). Consequently, for Foster’s CPU 406 to “execute code … to perform” “storing … at least one data chunk,” Foster’s CPU 406 must at least instruct recipient 430 to store or perform some action related to storing a content unit.

’609 Paper 22 at 44

’609 Ground 2 (anticipation and obviousness): Foster’s processor is “coupled to” storage devices under limitation [1F], which SFI does not dispute is a broad limitation

Petition

8. [1F]

As discussed below, Foster discloses and renders obvious “*storing, in a memory coupled to the processor, at least one data chunk of the plurality of data chunks with data indicative of at least one of the distinct encryption keys on at least one storage device*,” as claimed.

As illustrated in Foster’s Figure 8 (color-annotated below), Foster’s method generates and stores a “processed content unit 116,” which includes three pieces of data: (1) an encrypted **payload** 114 (an encrypted “*data chunk*”); (2) “encrypted **title key** 112” (a “*distinct encryption key*”), which is a **title key** “after it has been encrypted with the **KEK**” (the “***first key***”); and (3) header 104. Foster, [0065]-[0066], [0071] (“[T]itle keys 108 and 208 are attached and transmitted with the content packets they were used to encrypt.”), [0023], FIG. 9; Bhattacharjee, ¶¶200-01. Each “processed content unit 116” is stored on a receiver at a different “recipient 430.” POSAs understood such a receiver (*e.g.*, consumer set-top box) can store the content units ***because*** each includes a “*storage device*” (*e.g.*, a hard drive). Foster, [0058] (“receiver may ***cache***” (store) “KMBs…in **memory**” (a storage device)); Bhattacharjee, ¶201.

’609 Paper 2 at 72-73

As discussed above (§VI.B.8), POSAs understood the ’609 patent uses “coupled to” broadly to include a connection using any communication interface that allows two components to exchange information. EX1001, 66:49-55 (describing a component “**coupled to** [a] storage area [] **using any suitable communication interface** [] (e.g., USB, serial, parallel, Bluetooth, IR, IEEE 1394, Ethernet, **or any other suitable communications interface**”); Bhattacharjee, ¶204.

Each content unit 116 discussed above is stored on receiver at “receipt 430”; that receiver is a “*storage device*” (*e.g.*, consumer set-top box) with “*memory*” (*e.g.*, a hard drive). Foster, [0040], [0052], [0058]. As illustrated in Figure 16 (color-annotated below) recipient 430 (and its memory) is in communication with CPU 406 (the “*processor*”) via “transmission system 424” and “bus 410.” Foster, [0078]-[0079]. Thus, the memory in recipient 430’s storage device is “*coupled to*” Foster’s CPU (“*processor*”). They are also coupled to each other under the claim scope that SFI asserts for [1F] in related district-court litigation. EX1009, 10-11 (alleging limitation is met by storage in “storage systems” remote from the alleged processor performing the encryption); Bhattacharjee, ¶205.

’609 Patent, Claim 1

[1A] “executing code by a processor”

* * *

[1F] “storing, in memory coupled to the processor … on at least one storage device”

SFI’s Claim Constructions

NO CONSTRUCTION

EX1065 (cited ’609 Pet., 8-9)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’609 Ground 2 (obviousness): Corroborated expert testimony of three independent reasons POSAs had motivation for Foster’s CPU to execute code to store encrypted data in a memory coupled to the CPU

Dr. Bhattacharjee

207. In my opinion, POSAs had reason to use one or more of those conventional storage devices that are coupled to Foster’s processor to store data sets encrypted by Foster’s processor. Doing so would have beneficially allowed Foster’s system to store copies of encrypted content for multiple beneficial uses.

208. First, storing encrypted content in a memory coupled to Foster’s processor would have facilitated backing up encrypted content, which was conventional and also would have been useful, for example, to resend encrypted content to recipients (*e.g.*, if the content were lost or requested again). My opinion in this regard is corroborated by EX1050 at 1:19-22, which discloses as background information “[s]afeguarding a computer file system prevents irrevocable loss of data” and “is essential for ensuring resiliency of the data.” EX1050 at 2:31-38 similarly teaches the benefit of maintain redundant copies of data to “protect[] the system against data loss.” EX1050 at 11:11-16 and 16:6-20 similarly teaches the benefit of maintaining data distributed against multiple storage devices to “maintain[] data reliability in the event of failure.”

209. Second (and similarly), storing encrypted content in memory coupled to Foster’s processor would have beneficially facilitated video-on-demand services. As I discussed above (§VIII.A), Foster’s computer system is or can be part of a content service provider. By the mid-2000s, POSAs understood that content service providers desired the ability to provide video-on-demand service, where content stored by the content provider (*e.g.*, a movie) could be delivered to recipients on demand (*e.g.*, for a fee). My opinion in this regard is corroborated by EX1059 at 2:9-4:10, which discloses an invention “for handling the dissemination and/or purchase of programs, e.g., streaming digital content on demand, or purchases … via the Internet.” EX1059 at 3:32-4:10 describes how a “content provider, e.g., (Disney, Warner, etc.[])” are providing the video-on-demand services. EX1059 at 5:4-23 also describe how “a consumer [can] purchase[] content from a movie studio (Disney)” or a “user may purchase a recent Time Warner movie” via video-on-demand services. My opinion is further corroborated by EX1060 at [0007]-[0008], which notes as background information that “[c]ustomer appeal of ‘video on demand’ is well known” and in such systems “a server or set of servers will store all movies, will enable customers to select movies, and will stream the movies to the consumer” and that “On Demand provides the ability to instantly download movies for playback.”

210. Third, storing encrypted content in memory coupled to Foster’s processor would have also beneficially facilitate distributing the encrypted content to future recipients (*e.g.*, future subscribers of the content service providers). *See* EX1007 (discussed above).

’609 EX1003, ¶¶207-210 (cited ’609 Pet., 77)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’609 Ground 2: The Parties’ disputes are narrow

- No dispute that Foster discloses limitations 1[PRE]-[1B] and [1C2]-[1D]

- **Only disputes are whether:**
  (1) Foster discloses “data chunks” ([1C1])*
  *** This is the same dispute that is discussed on slides 92-100**
  (2) Foster discloses or renders obvious “executing code” to store a data chunk in a “storage device” memory “coupled to” the processor ([1F])

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’609 Ground 2 (Foster): SFI’s arguments for limitation [1F] (executing code for “storing” a data chunk in “storage device” “memory coupled to the processor”)

For obviousness, SFI’s POR makes a single argument for [1F]:

> SFI argues a POSA would not have been motivated to store data chunks in a memory coupled to the processor

’609 POR, 42-47

For anticipation, SFI’s POR makes a single argument for [1F]:

> Under SFI’s new construction, SFI argues Foster does not disclose the processor executing code that sends storage instructions to a storage device memory coupled to the processor

’609 POR, 47-48

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’609 Ground 2: SFI’s arguments for [1F] fail

1. A POSA would have had motivation to store Foster-encrypted data in a storage device coupled to the processor for several reasons (e.g., backup, VOD, etc.)

---

2. Even if SFI’s new claim construction were adopted, it would have at least been obvious to have Foster’s processor execute code to store a data chunk in a storage device coupled to the processor under SFI’s construction (e.g., backup, VOD, etc.)

’609 Pet., 76-78
’609 Reply, 6-18

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’609 Ground 2: For obviousness, SFI disputes whether a POSA would have had motivation to store Foster-encrypted content in a storage device coupled to Foster’s processor

Petition

POSAs had reason to use one of the conventional storage devices coupled to Foster’s processor to store data sets encrypted by Foster’s processor. Doing so would have beneficially allowed Foster’s system to store copies of encrypted content for (1) backup, (2) video-on-demand distribution, and/or (3) distribution to later-identified recipients (*e.g.*, new subscribers), without having to repeat the encryption processing every time digital content (*e.g.*, a movie) needed to be encrypted and delivered to recipients. Bhattacharjee, ¶¶207-11 (citing as corroboration EX1045, FIGS. 5-6, [0109]-[0111]; EX1050, 1:19-22, 2:31-38, 11:11-26, 16:6-20; EX1059, 2:9-4:10, 5:4-23; EX1060, [0007]-[0008]).

Storing encrypted content in memory coupled to Foster’s processor on at least one storage device would have also been applying a known technique to solve the known problem of how to store encrypted content for delivery to multiple recipients at varying times, and thus was obvious. *Intel*, 61 F.4th at 1380 (“[I]f there’s a known technique to address a known problem using prior art elements according to their established functions, then there is a motivation to combine.”) (cleaned up); Bhattacharjee, ¶212 (citing EX1045).

’609 Paper 2 at 77

POR

Petitioner offers an alternative, obviousness theory attempting to account for the fact that recipient 430 is not “coupled” to CPU 406. Pet., 77 (“POSAs had reason to use one of the conventional storage devices coupled to Foster’s processor to store data sets encrypted by Foster’s processor.”). This argument is discussed in Section III.D, *infra*. But even if Petitioner’s modification were accepted, Petitioner presents no argument as to how (or why) a POSITA would be motivated to modify CPU 406 to execute code to store encrypted content units on these “conventional storage devices.” *Id.*, 76-78. At best, Petitioner’s argument is vague and conclusory. But obviousness may not be shown through conclusory statements.

’609 Paper 22 at 47

cl. 1. Petitioner, however, fails to show that the alleged “memory” (an alleged hard-drive on Foster’s recipient 430) is “coupled” to the alleged processor. *See* Section III.D.1. Petitioner, moreover, fails to demonstrate a POSITA would be motivated to modify Foster to address this shortcoming. *See* Section III.D.2.

’609 Paper 22 at 49

But Petitioner’s proposed modification does not actually solve any problem. As Dr. Rubin explains, a POSITA would recognize that the purported benefits Petitioner identifies vis-à-vis “(1) backup, (2) video-on-demand distribution, and/or (3) distribution to later-identified recipients,” are already solved by Foster’s system because those benefits are provided by the content owner:

’609 Paper 22 at 52

Reply

2. Obviousness

Petitioner also demonstrated POSAs had reason to implement Foster’s processor to execute code to store encrypted content in “conventional storage devices coupled to [the] processor,” *e.g.*, to beneficially “allow[] Foster’s system to store copies of encrypted data,” for “backup” and VOD/subsequent “distribution,” etc., “without having to repeat the encryption processing” multiple times for one *dataset*. Petition, 77. SFI does not dispute that this implementation meets [1F]’s *storing* limitation.

’609 Paper 25 at 11

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’609 Ground 2: SFI’s attack on obviousness fails because a POSA had motivation to execute code to store Foster-encrypted content in a storage device for multiple reasons

EX1059: On-demand Services

Figure 1 depicts a business environment 10 employing the discounting reward methodology of the invention. As shown in Figure 1, there is shown a content provider 30 which may be any commercial enterprise having a web-presence (e.g., web-site 31) enabling on-line purchase of content, whether it be streaming digital content 15 provided on-demand to the customer’s home 20 via a streaming connection over the Internet 12, or an on-line purchase of content, e.g., books, analog or digital storage media, for immediate downloading via an Internet connection or, subsequently via conventional delivery systems (mail) to the customer’s home 20. As shown in Figure 1, a gateway

’609 EX1059, 2:29-3:13 (cited ’609 Pet., 77)

1. If a consumer purchases content from a movie studio (Disney), and he/she is also paying a subscription to view programs from the same company (Disney), that consumer will benefit from special discounts. For example, the user may purchase a recent Time Warner movie and subscribe to Time Warner TV channels, or may

’609 EX1059, 5:6-10 (cited ’609 Pet., 77)

EX1050: Data Backup

2. Background Art

Safeguarding a computer file system prevents irrevocable loss of data stored in the file system. Creating backup copies of the file system data is essential for ensuring resiliency of the data. Traditionally, backup involves creating a copy of the

’609 EX1050, 1:19-22 (cited ’609 Pet., 77)

This scheme protects the system against data loss by adding redundancy to the original data. The basic idea of the scheme is to fragment a stored object (e.g., file) into blocks and distribute the blocks across available servers divided among servers that hold original data and servers that hold redundant data. If the total number of failed servers is less than the

’609 EX1050, 2:31-36 (cited ’609 Pet., 77)

EX1060: On-demand & Later Downloads

[**0007**] Customer appeal of “video on demand” is well known. Generally, true video on demand can be characterized as instantaneous viewing of a movie (or other content) which may be selected from a list of preferably all movies available. Ideally, a server or set of servers will store all movies, will enable customers to select movies, and will stream the movies to the consumers while the customers view the movies without network interruption. However, given the many shortcomings of today’s technology and network-related infrastructure, true video on demand is currently unavailable to the general public. Communication

* * *

[**0008**] In limited circumstances, true VOD is offered today using specialized, reliable networks capable of delivering and maintaining high capacity and speed. Cable “On Demand” is one such service. On Demand provides the ability to instantly download movies for playback, but only if the user is connected to a high speed digital network and if the service provider can support VOD. This service is not available over traditional broadband connections.

’609 EX1060, [0007]-[0008] (cited ’609 Pet., 77)

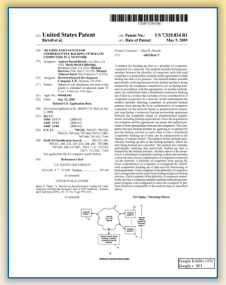
’609 Ground 2: SFI’s attack on obviousness fails because multiple pieces of corroborating evidence support that a POSA had motivation to store Foster-encrypted content

SFI’s POR & Sur-Reply

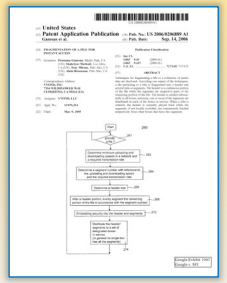
DOES NOT DISPUTE

Storing for Data Backup

EX1050



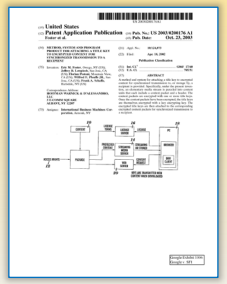
EX1060



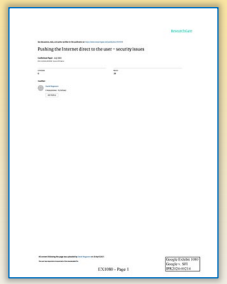
Cited ’609, Pet. 77; ’609 Reply 15

Storing for Streaming

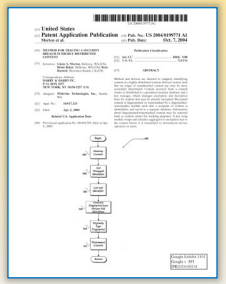
Foster



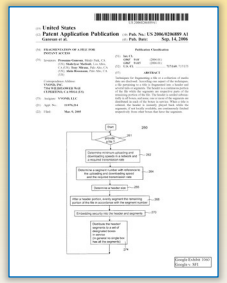
EX1080



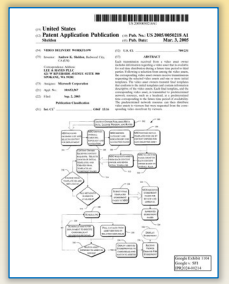
EX1101



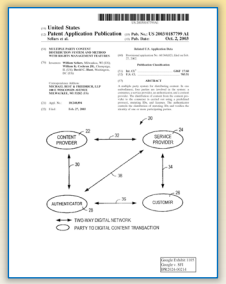
EX1060



EX1104



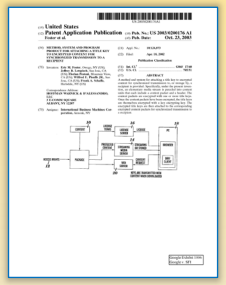
EX1105



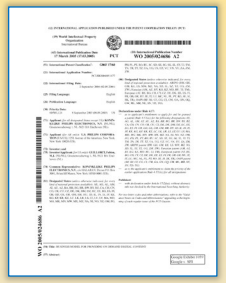
Cited ’609 Pet., 77; ’609 Reply 15-18, 23)

Storing for On-Demand / Distribution

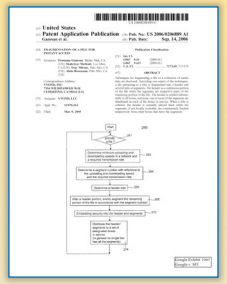
Foster



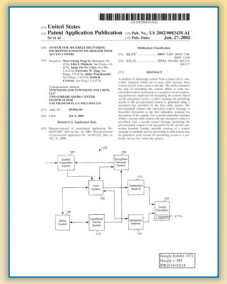
EX1059



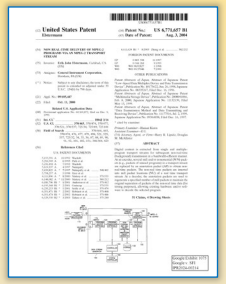
EX1060



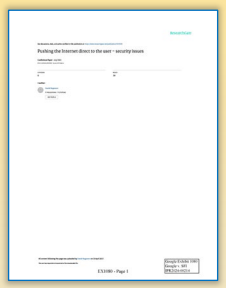
EX1073



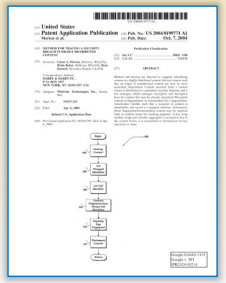
EX1075



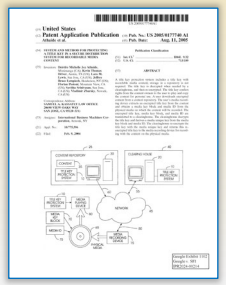
EX1080



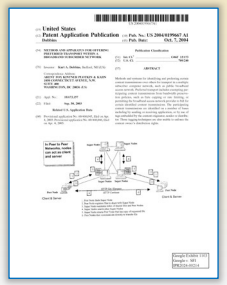
EX1101



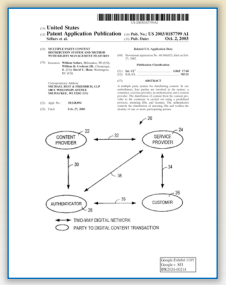
EX1102



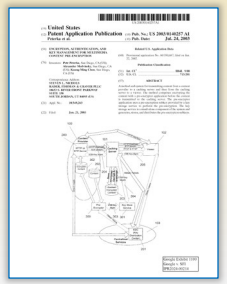
EX1103



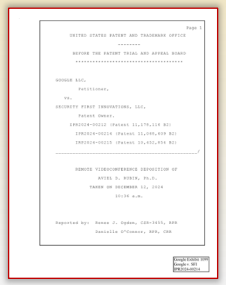
EX1105



EX1100



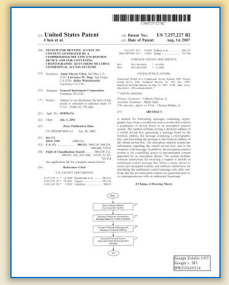
Dr. Rubin



Cited ’609 Pet., 77; ’609 Reply, 15-18

Avoiding always encrypting on-the-fly

EX1097



Cited ’609 Reply, 15

’609 Ground 2 (obviousness): SFI’s Sur-Reply allegation that the Petition’s obviousness argument did not address Foster’s code is a new argument and fails

Sur-Reply

Petitioner’s obviousness arguments are also deficient. Petitioner disputes that it presented no argument as to how (or why) a POSITA would be motivated to modify CPU 406 to execute code to store encrypted content units on “conventional storage devices,” citing the Petition at 76-78. But this section of the Petition does not address ***the processor*** executing code at all. *See* Pet., 76-78 (addressing only how to modify Foster’s memory 402 to store content). Petitioner’s argument that a POSITA “would have known how to implement the feature” because its modification purportedly “relied on conventional storage devices” cannot sustain its argument. Reply, 11-12. As in the Petition, Petitioner’s vague and conclusory

’609 Paper 33 at 13-14

’609 Patent

26 illustrates this application. Storage area **2500** includes a portion of parsed data **2502**. Physical token **2504**, having a portion of parsed data **2506** would need to be coupled to storage area **2500** using any suitable communications interface **2508** (e.g., USB, serial, parallel, Bluetooth, IR, IEEE 1394, Ethernet, or any other suitable communications interface) in order to access the original data. This is useful in a

EX1001, 66:49-55 (cited Reply, 42)

The secure data parser of the present invention may be used to implement a secure authentication system whereby user enrollment data (e.g., passwords, private encryption keys, fingerprint templates, biometric data or any other suitable user enrollment data) is parsed and split using the secure data parser. The user enrollment data may be parsed and split whereby one or more portions are stored on a smart card, a government Common Access Card, any suitable physical storage device (e.g., magnetic or optical disk, USB key drive, etc.), or any other suitable device. One or more

EX1001, 66:62-67:4 (cited Reply, 15)

Orsini

[0063] According to one embodiment of the invention, the user system **105** comprises a conventional general-purpose computer having one or more microprocessors, such as, for example, an Intel-based processor. Moreover, the user system **105** includes an appropriate operating system, such as, for example, an operating system capable of including graphics or windows, such as Windows, Unix, Linux, or the like. As shown in **FIG. 1**, the user system **105** may include

Orsini, [0063] (cited ’609 Pet., 18); *see* Pet., 9 (demonstrating same disclosure in ’609 patent) (citing EX1010)

"Publicover argues that Venable and Kiderman's disclosures are too sparse to adequately explain to a skilled artisan how to modify Venable's system to identify vestibulo-ocular movement. But as the examiner and Board correctly found, Publicover's specification is just as sparse on how a system would identify this type of eye movement.”

In re Publicover, 813 F. App'x 527, 532 (Fed. Cir. 2020) (responsive to ’609 Sur-Reply, 13-14)

"A patent specification need not describe what already is well known. Patent Owner does not purport to have invented a microprocessor or a microcontroller.”

Square, Inc. v. 4361423 Canada Inc., IPR2019-01651, Paper 35 (PTAB Apr. 27, 2021) (FWD) (responsive to ’609 Sur-Reply, 13-14)

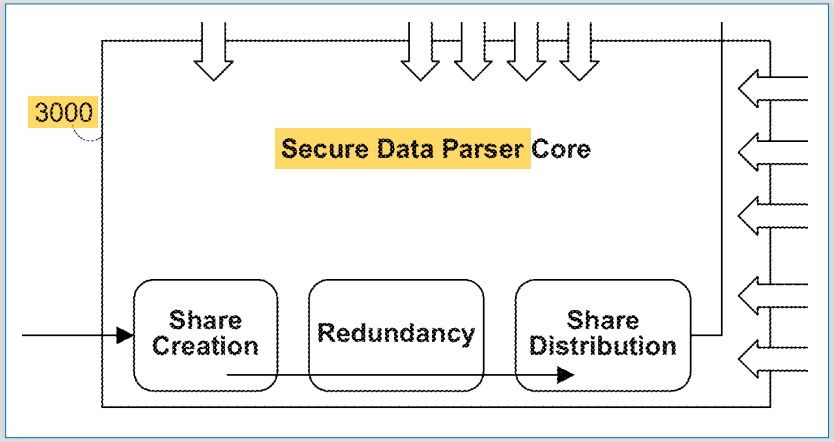
DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’609 Ground 2 (obviousness): SFI’s new Sur-Reply allegation that the Petition did not sufficiently address Foster’s code for storing fails because SFI’s Patent relies on conventional processors and code for storing

’609 Patent

According to one embodiment, the transaction engine **205** also transmits a confirmation message to the user. Although the signing process **1100** is disclosed with reference to its preferred and alternative embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for the signing process **1100**. For example, the vendor may be replaced with a user application, such as an email application. For example, the user may wish to digitally sign a particular email with his or her digital signature. In such an embodiment, the transmission throughout the signing process **1100** may advantageously include only one copy of a hash of the message. Moreover, a skilled artisan will recognize from the disclosure herein that a wide number of client applications may request digital signatures. For example, the client applications may comprise word processors, spreadsheets, emails, voicemail, access to restricted system areas, or the like.

EX1001, 32:23-39 (responsive to Sur-Reply, 13-14)



EX1001, FIG. 31 (partial) (highlighted)
(responsive to Sur-Reply, 13-14)

In addition, a skilled artisan will recognize that the cryptographic system **100** or the trust engine **110** may include the ability to recognize any type of devices, such as, but not limited to, a laptop, a cell phone, a network, a biometric device or the like. According to one embodiment, such recognition may come from data supplied in the request for a particular service, such as, a request for authentication leading to access or use, a request for cryptographic functionality, or the like. According to one embodiment, the foregoing request may include a unique device identifier, such as, for example, a processor ID. Alternatively, the

EX1001, 37:14-24 (responsive to Sur-Reply, 13-14)

FIG. **31** illustratively shows how the arrangement of FIG. **30** may be used when a write (e.g., to a storage device), insert (e.g., in a database field), or transmit (e.g., across a network) command is issued in application layer **3024**. At step **3100** data to be secured is identified and a call is made to the secure data parser. The call is passed through wrapper layer **3022** where at step **3102**, wrapper layer **3022** streams the input data identified at step **3100** into assembled data buffer **3008**. Also at step **3102**, any suitable share information, filenames, any other suitable information, or any combination thereof may be stored (e.g., as information **3106** at wrapper layer **3022**). Secure data processor **3000** then parses and splits the data it takes as input from assembled data buffer **3008** in accordance with the present invention. It outputs the data shares into split shares buffers **3010**. At step **3104**, wrapper layer **3022** obtains from stored information **3106** any suitable share information (i.e., stored by wrapper **3022** at step **3102**) and share location(s) (e.g., from one or more configuration files). Wrapper layer **3022** then writes the output shares (obtained from split shares buffers **3010**) appropriately (e.g., written to one or more storage devices, communicated onto a network, etc.).

EX1001, 72:33-54
(responsive to Sur-Reply, 13-14)

known as “metadata”, related to that password attempt. This circumstantial data includes information such as: the time at which the particular authentication instance took place, the network address from which the authentication information was delivered, as well as any other information as is known to those of skill in the art which may be determined about the origin of the authentication data (the type of connection, the processor serial number, etc.).

In many cases, only a small amount of circumstantial metadata will be available. For example, if the user is located on a network which uses proxies or network address translation or another technique which masks the address of the originating computer, only the address of the proxy or router may be determined. Similarly, in many cases information such as the processor serial number will not be available because of either limitations of the hardware or operating system being used, disabling of such features by the operator of the system, or other limitations of the connection between the user’s system and the trust engine **110**.

EX1001, 38:62-39:14
(responsive to Sur-Reply, 13-14)

sic: “Secure data parser 3000”

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’609 Ground 2: SFI’s Argument #2 (processor “instructions”) raises a new claim construction dispute about the scope of limitation [1F]

SFI’s POR

Petitioner, however, does not attempt to show that Foster’s CPU 406, executes any code relating to storing content units on recipient 430 or instructing recipient 430 to store content units. Petitioner does not mention—let alone explain—how Foster’s CPU 406, which is part of Foster’s computer system 400, executes code to store the content units on the separate recipient 430 (*e.g.*, a set-top box). *See* Pet., 72-75. Nor is there any reason to believe that Foster’s CPU 406 executes code to perform storing the content units on Foster’s recipient 430 because, as Dr. Rubin explains, Foster’s CPU 406 simply transmits the content units to the recipient 430 and is agnostic as to what the recipient 430 does with the content units after transmission:

’609 Paper 22 at 43

Reply

SFI argues Foster’s processor is “agnostic as to what the recipient does with” content and fails to “instruct the recipient to store or perform some action related to storing.” POR, 43-44. This is a new construction (EX1060) and SFI identifies no intrinsic evidence requiring the processor to instruct a storage device. *Wasica Finance GmbH v. Continental Automotive Sys., Inc.*, 853 F.3d 1272, 1281-82 (Fed. Cir. 2017). Claim 1 only requires executing code *to perform storing* in memory coupled to the processor. As discussed, Petitioner shows that Foster’s processor does that because it executes code that encrypts content for storing in storage-device memories *coupled to* Foster’s processor. DI, 25-26; Bhattacharjee-Reply, ¶61.

But even under SFI’s new, narrowing construction, Foster anticipates, in several ways.

’609 Paper 25 at 6-7

SFI’s Sur-Reply

In reply, Petitioner contends that the POR’s showing “is a new construction (EX1060) and SFI identifies no intrinsic evidence requiring the processor to instruct a storage device.” Reply, 6-7.[2] But the POR did not offer a new claim construction. Rather, the claim language on-its-face requires that the “processor” “execut[e] code” “***to perform***” storing at least one data chunk on at least one storage device. As the claim says, the processor must execute code that performs storing the data chunks. And, as the POR explained, there is no such code executed on Foster’s processor that “performs” storing at least one data chunk on the recipient or the recipient’s hard drive. POR, 42-47.

’609 Paper 33 at 5-6

Claim construction can be an evolving issue.… [W]e are not persuaded that the subsequent development of both parties’ positions on the interpretation of this term throughout the course of this proceeding is contrary to our rules. ‘There is no rule, for example, requiring a petition to describe all possible or reasonable claim constructions and to present invalidity theories under those constructions.’ *Axonics, Inc. v. Medtronic, Inc.*, 75 F.4th 1374, 1380–81 (Fed. Cir. 2023). The *Axonics* court further opined that ‘we have continued to recognize that a petitioner is entitled under the APA to respond to new claim construction arguments made by a patent owner.’ *Id.* at 1382.… The fact remains that limitation 1[e] is in dispute as to its scope and application.… Notwithstanding the lack of a ‘conventional’ claim construction proposal from either party, we review the parties’ contentions, which we view as properly and timely submitted, as follows.

Apple Inc. v. Immersion, Inc., IPR2023-00472, Paper 22 (FWD) at 21-23 (July 29, 2024) (responsive to ’609 Sur-Reply, 5-6)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

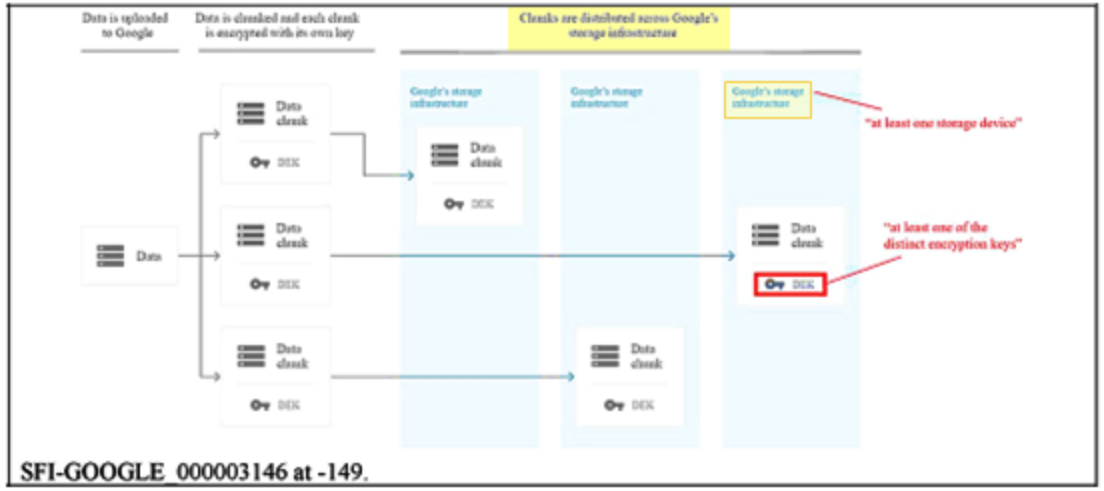
'609 Ground 2: That SFI's Argument #2 raises a new claim construction is demonstrated by SFI's pre-institution infringement allegations, which say nothing about processor instructions

SFI's Infringement Contentions

executing code by ocessor to perform:	Google's "Encryption at Rest" comprises executing code by at least one processor. For example, Google's default encryption at rest requires executing code by a processor such as the associated with the wrapping and unwrapping of keys on Google's computer systems. Google has a fourteen data centers in the U.S., each of which contains "many computers that store and process amounts of information."[3] Google's encryption at rest technology occurs on machines, executing co processors, within Google's data centers.[4] Below is an image depicting servers for, *e.g.*, storage and processing of encrypted data within a Googl center. 

EX1009, 2 (EX1009 cited '609 Pet., 43-44, 75)

No Allegations that Processor Must Instruct a Storage Device to Store

U.S. Patent No. 11,068,609	Google's "Encryption at Rest"[1]
1[f] storing, in a memory coupled to the processor, at least one data chunk of the plurality of data chunks with data indicative of at least one of the distinct encryption keys on at least one storage device.	Google's "Encryption at Rest" comprises storing, in a memory coupled to the processor, at least one data chunk of the plurality of data chunks with data indicative of at least one of the distinct encryption keys on at least one storage device. For example, Google stores, in a memory coupled to a processor within its data centers, the data chunks with data indicative of at least one of the distinct encryption keys on one or more storage devices, at least when it stores DEKs with the data chunks throughout Google's storage infrastructure. After the original data is split and encrypted, each encrypted chunk is stored in a storage system with its data encryption key.[8] Google explains that "[e]ach chunk is distributed across our storage systems and is replicated in encrypted form for backup and disaster recovery." SFI-GOOGLE_000003146 at -149. Google states, for example, that in its "data centers, [encrypted] data belonging to Google Workspace customers is stored at rest in two types of systems: disks and backup media." SFI-GOOGLE_000003287 at -291; *see also* SFI-GOOGLE_000005743 (Cloud Key Management Service). Below is an exemplary annotated image from the Google Cloud webpage showing an example of the at least one storage device and data indicative of at least one of the distinct encryption keys:  SFI-GOOGLE_000003146 at -149.

EX1009, 10-11 (cited '609 Pet. 43-44, 75)

’609 Ground 2: That SFI’s Argument #2 raises a new claim construction is demonstrated by SFI’s POPR, which had no argument requiring the processor to instruct a storage device to store

SFI’s POPR

III. THE FOSTER GROUND FAILS (ALL CLAIMS, GROUND 2).....36

A. The ’609 And Foster Fundamentally Differ.....36

B. Petitioner Fails To Show Foster Discloses “Generating A Plurality Of Data Chunks Based On A Data Set” (Limitation 1[c][1]).....39

C. Petitioner Fails To Show That Foster Renders Obvious “The Data Set Can Be Reconstructed Using At Least A Minimum Number Of The Plurality Of Chunks” (Limitation 1[c][2]).....45

D. Petitioner Fails To Show That Foster Discloses Or Renders Obvious “Storing, . . . At Least One Data Chunk . . . On At Least One Storage Device” (Limitation 1[f]).....54

E. Petitioner Fails To Show That Foster Teaches The Claimed Steps In Order.....62

Paper 10 at i-ii

D. Petitioner Fails To Show That Foster Discloses Or Renders Obvious “Storing, . . . At Least One Data Chunk . . . On At Least One Storage Device” (Limitation 1[f]).

Limitation 1[f] requires “***storing,*** in a memory coupled to the processor, ***at least one data chunk*** of the plurality of data chunks with data indicative of at least one of the distinct encryption keys ***on at least one storage device***.” Ex. 1001

Paper 10 at 54-55

No Allegations that Processor Must Instruct a Storage Device to Store

[’609] cl. 1. Even assuming, *arguendo*, that the data packets of Foster could correspond to the chunks of Claim 1, this limitation is not satisfied by Foster because Foster does not ***store*** its data packets on a ***storage device***, as required by Claim 1. In Foster, data is streamed to a “receiver” for immediate decoding and play on a playback device such as a set top box, not for secure long-term storage. As previously explained, this distinction goes to the fundamental difference between the ’609’s inventions—for secure storage of data—and Foster’s invention—for encrypting and streaming content from a content owner to a consumer. *See* Section III.A.

Petitioner argues that Foster discloses a “memory” on a “storage device” that is also “coupled to the processor” because Foster teaches transmitting content units to a receiver. Pet., 75. Petitioner’s argument, however, fails because Foster’s “receiver” is neither a “storage device” nor used to store chunks, as those terms are used in Claim 1.

Although Petitioner never explains what a “storage device” is, Petitioner asserts that Foster’s “receiver” (also referred to a “recipient”) “is a ‘storage device’” because data packets can be “stored on [the] receiver.” Pet., 75. Petitioner’s argument thus rests upon the unsubstantiated premise that any device that has a memory is a claimed “storage device.” This is wrong, as Dr. Rubin explains:

Paper 10 at 54-55

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’609 Ground 2: Because SFI did not raise Argument #2’s new claim construction pre-institution, Petitioner’s reply was permitted to prove that Foster meets or render obvious under SFI’s claim scope

Institution Decision

C. Claim Construction

* * *

4. Limitation [1F]: “storage device”

Claim 1 recites “storing, in a memory coupled to the processor, at least one data chunk of the plurality of data chunks with data indicative of at least one of the distinct encryption keys on at least one *storage device*.” Ex. 1001, 83:27–30 (emphasis added).

Patent Owner argues that the recited “storage device” is for long-term storage:

> [The skilled artisan] reviewing the ’609 patent would not understand a “storage device” to encompass something like Foster’s receiver (*e.g.*, a set-top box or a DVD player) because the examples of storage devices given in the ’609—“magnetic or optical disk, USB key drive, etc.”—are each devices primarily dedicated to long-term storage.

* * *

We note that claim 1 does not use the term “long-term.” Nonetheless, we need not resolve whether the claim requires long-term storage at this stage of the proceeding because we disagree with Patent Owner that this feature distinguishes Foster. *See infra* § III.D.2.a.

’609 Paper 16 at 18-19

SFI’s Dr. Rubin

A. So the word “agnostic” does not appear in the claims here…

’609 EX1099, 189 (cited Reply, 6)

SFI’s POR

Petitioner, however, does not attempt to show that Foster’s CPU 406, executes any code relating to storing content units on recipient 430 or instructing recipient 430 to store content units. Petitioner does not mention—let alone explain—how Foster’s CPU 406, which is part of Foster’s computer system 400, executes code to store the content units on the separate recipient 430 (*e.g.*, a set-top box). *See* Pet., 72-75. Nor is there any reason to believe that Foster’s CPU 406 executes code to perform storing the content units on Foster’s recipient 430 because, as Dr. Rubin explains, Foster’s CPU 406 simply transmits the content units to the recipient 430 and is agnostic as to what the recipient 430 does with the content units after transmission:

No citations to ’609 patent

Paper 22 at 43; *see also id.*, 42-56

“Barring argument and evidence in a reply directed to a new claim construction proposed by the patent owner would create opportunities for sandbagging by the patent owner in order to create an estoppel. A patent owner has the opportunity in a preliminary response to oppose institution on the ground that the claim construction relied on by the petitioner is incorrect, and the Board may adopt the patent owner's proposed claim construction and deny institution if the petition fails to demonstrate a reasonable likelihood that a claim would be unpatentable under the correct claim construction. See 35 U.S.C. § 314(a). The result would be that there would be no institution and no estoppel pursuant to 35 U.S.C. § 315(e). But if instead of raising the issue in a preliminary response, a patent owner sits on its strongest claim construction arguments before institution and then raises them in response after institution, a patent owner could obtain a favorable final IPR decision and an estoppel without the Board's reaching the merits of any invalidity arguments under the newly adopted claim construction.… We are reluctant to adopt a construction of the APA and the Board's rules that would permit such gamesmanship.

Axonics, Inc. v. Medtronic, Inc., 75 F. 4th 1374, 1384 (Fed. Cir. 2023) (cited ’609 Reply, 4; ’854 Reply, 14)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’609 Ground 2: SFI’s Assertion that the claims require that the processor send “instructions” to a storage device is refuted by the ’609 patent’s specification

SFI’s POR

Petitioner, however, does not attempt to show that Foster’s CPU 406, executes any code relating to storing content units on recipient 430 or instructing recipient 430 to store content units. Petitioner does not mention—let alone explain—how Foster’s CPU 406, which is part of Foster’s computer system 400, executes code to store the content units on the separate recipient 430 (*e.g.*, a set-top box). *See* Pet., 72-75. Nor is there any reason to believe that Foster’s CPU 406 executes code to perform storing the content units on Foster’s recipient 430 because, as Dr. Rubin explains, Foster’s CPU 406 simply transmits the content units to the recipient 430 and is agnostic as to what the recipient 430 does with the content units after transmission:

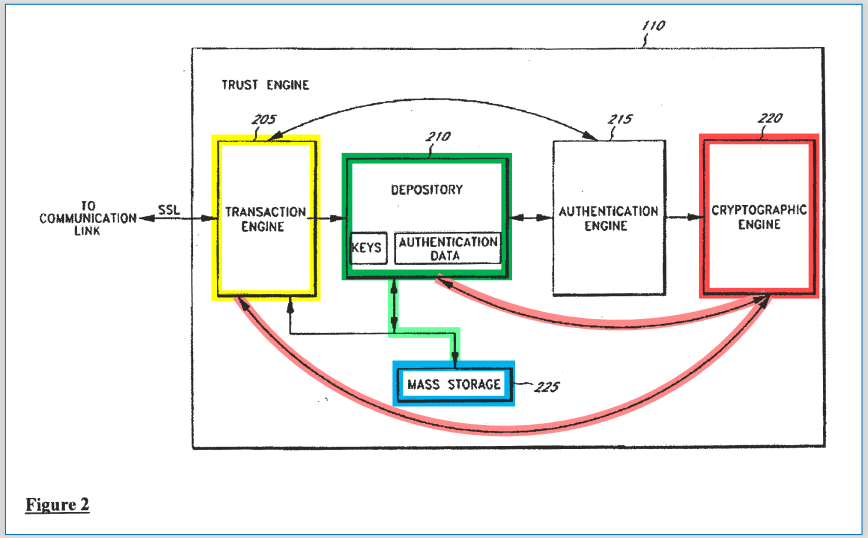
Paper 33 at 43

SFI’s Sur-reply

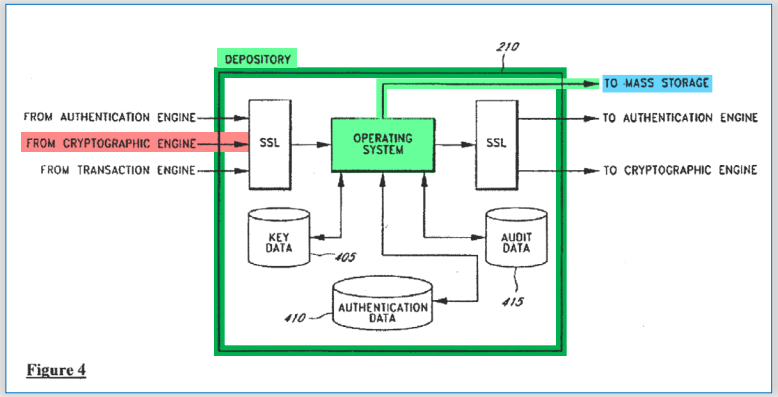
instruct a storage device.” Reply, 6-7.[2] But the POR did not offer a new claim construction. Rather, the claim language on-its-face requires that the “processor” “execut[e] code” “***to perform***” storing at least one data chunk on at least one storage device. As the claim says, the processor must execute code that performs storing the data chunks. And, as the POR explained, there is no such code executed on Foster’s processor that “performs” storing at least one data chunk on the recipient or the recipient’s hard drive. POR, 42-47.

Paper 33 at 6

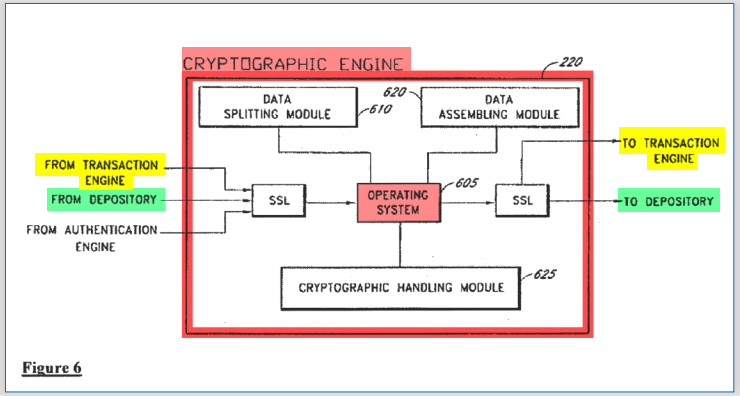
Orsini



EX1005, FIG. 2 (annotated) (cited Pet., 20) (also responsive to Sur-Reply, 6, 13-14); *see* Pet., 9 (demonstrating same figure is in the ’609 patent); *see also* ’609 Reply, 4 (citing ’609 patent, 15:51-16:3)



EX1005, FIG. 4 (annotated) (cited Pet., 40) (responsive to Sur-Reply, 6, 13-14); *see* Pet., 9 (demonstrating same figure is in the ’609 patent)



EX1005, FIG. 6 (annotated) (cited Pet., 19) (responsive to Sur-Reply, 6, 13-14); *see* Pet., 9 (demonstrating same figure is in the ’609 patent); *see also* ’609 Reply, 4 (citing ’609 patent, 15:51-16:3)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’609 Ground 2: Foster meets SFI’s narrow construction because it would have at least been obvious for Foster’s processor to execute code to stores encrypted data in the RAM coupled to Foster’s processor

Reply

First, Foster’s processor executes code that encrypts and stores data on storage-device memory (RAM) directly *coupled to* it. Bhattacharjee-Reply, ¶¶62-67 (citing EX1117, [0003]-[0005]); Foster, [0075]-[0078], FIG. 16 (annotated below); EX1001, 80:63-81:2 (“RAM” is “local [storage] hardware”); DI, 25 (rejecting SFI’s pre-institution argument – which SFI’s POR does not repeat – that claim 1 requires long-term storage devices).

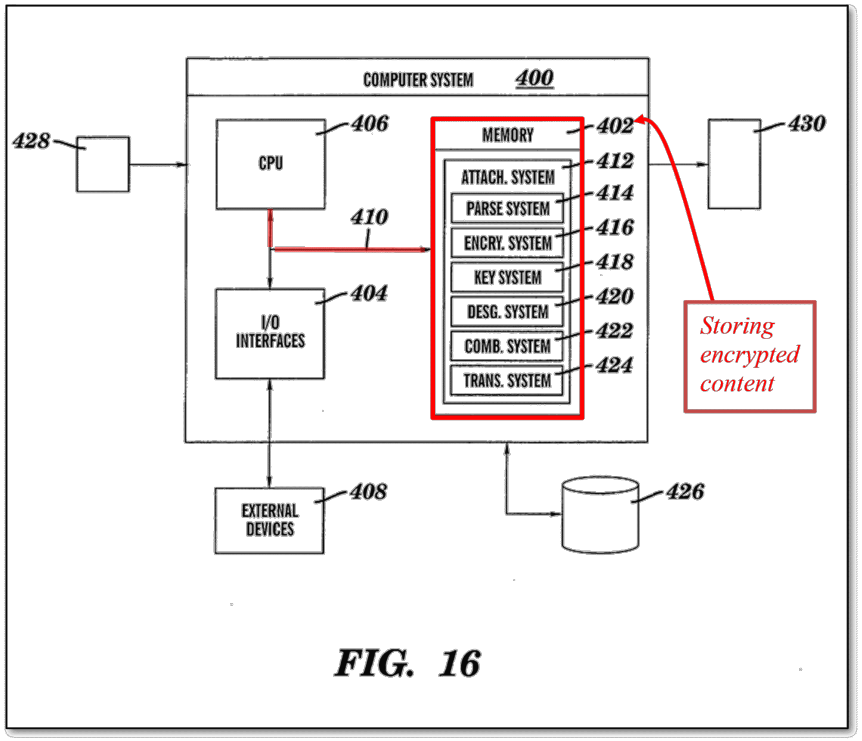


FIG. 16

’609 Paper 25 at 7-8

Petition

Alternatively, POSAs also had reason to store Foster’s encrypted content in the memory of storage devices coupled to the processor (*e.g.*, through a bus, direct wired/wireless connection, and/or a network-based connection). As discussed above (§VII.A), Foster’s system 400 is (or can be part of) a content service provider. As illustrated in Figure 16 (above), Foster’s system 400 includes memory 402, which comprises “any known type of data storage,” including “one or more types of data storage” or “distributed [storage] across a plurality of physical systems.” Foster, [0075]. Thus, Foster’s memory 402, which is coupled to CPU 406 via bus 410, includes storage devices. Bhattacharjee, ¶206.

’609 Paper 2 at 76-77

SFI’s Sur-reply

Even if Petitioner’s untimely argument were to be considered, it would still fail. Foster does not disclose storing “content units” (as required by the claims) on “memory 402.” Rather, Foster teaches that “[s]tored in memory 402 is attachment system 412, which includes parsing system 414, encryption system 416, key system 418, designation system 420, combination system 422 and transmission system 424.” Ex. 1006, [0078]. Foster thus teaches that the systems that execute Foster’s methods are stored on memory 402. Foster does not teach that the content units are stored on memory 402. *See id.*

Paper 33 at 9

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’609 Ground 2: SFI’s “instructions” argument fails because it was at least obvious to store encrypted data in RAM before sending to another device

Reply

First, Foster’s processor executes code that encrypts and stores data on storage-device memory (RAM) directly *coupled to* it. Bhattacharjee-Reply, ¶¶62-67 (citing EX1117, [0003]-[0005]); Foster, [0075]-[0078], FIG. 16 (annotated below); EX1001, 80:63-81:2 (“RAM” is “local [storage] hardware”); DI, 25 (rejecting SFI’s pre-institution argument – which SFI’s POR does not repeat – that claim 1 requires long-term storage devices).

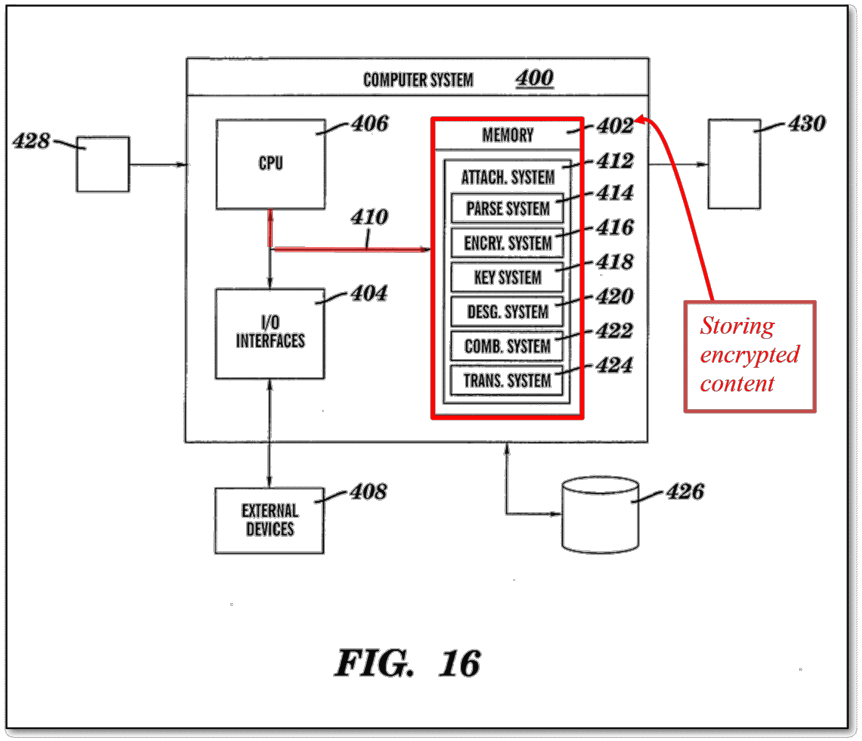


FIG. 16

’609 Paper 25 at 7-8

EX1117

[0003] The basic components of a conventional computer typically include one or more processors for processing program instructions; a temporary memory device (such as random access memory, or RAM) for storing program data and instructions that are being processed by the processors; a permanent memory device (such as a magnetic disk) for storing the computer’s operating system, programs and data;

EX1117, [0003] (cited Reply at 7)

Foster

content units to a recipient **430**. Stored in memory **402** is attachment system **412**, which includes parsing system **414**, encryption system **416**, key system **418**, designation system **420**, combination system **422** and transmission system **424**.

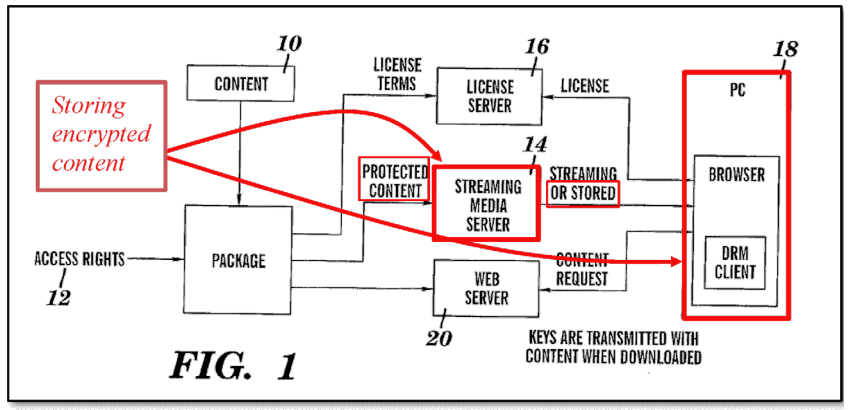
EX1006, [0078] (cited Reply at 7)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’609 Ground 2: An obvious VOD implementation of Foster meets SFI’s new construction because the processor execute codes to store data in a storage device for later download

Reply

Then, for Foster’s DRMS (*see supra* §I.A), Foster’s processor executes code to “package[]…and **load**[encrypted data] onto media server 14,” which is later “**downloaded** to PC 18.” Foster, [0049], FIG. 1 (annotated below); DI, 25 (“Foster teaches a PC…**storing**…the data set….”).



Thus, Foster’s processor is also executing code that instructs the media server to load (**store**) encrypted content on the media server’s storage device *coupled to* Foster’s processor, for the purpose of subsequent **storing** to (downloading to) another storage device *coupled to* Foster’s processor (i.e., FIG. 1’s “protected content” is “stored” at Foster’s server **and** PC). Bhattacharjee-Reply, ¶¶68-75 (citing EX1098, 5:44-65); DI, 23-25; Rubin-Depo., 83-89, 96-99 (admitting DRM “encrypt[s]” data “before it’s transmitted to the consumer”).

’609 Paper 25 at 9

SFI’s Sur-reply

In addition to being untimely, this new argument is wrong. The “DRMS” described in Foster is Foster’s prior art, not Foster’s inventive system. *See* Reply, 9; Ex. 1006 [Foster] [0005] (in “Background Art,” “Current technologies … include a personal computer-based Digital Rights Management System (DRMS)”), [0049] (“Referring now to FIG. 1, a depiction of a Digital Rights Management System (DRMS) is shown.”), Fig. 1; Ex. 2046 [Bhattacharjee-2nd-Depo.] 24:15-25. Indeed, Foster criticizes the DRMS depicted in Foster’s Figure 1 because it “fails to provide synchronized delivery of the content with the keys used to encrypt the content”—which, as Petitioner’s expert admits, is “sort of the whole invention in Foster.” Ex. 1006 [Foster] [0051]; Ex. 2034 [Bhattacharjee-1st-Depo.] 32:20-23. Foster’s inventive system is intended to replace both DRMS and CAS. *See id.*, [0003]-[0007] (criticizing DRMS and CAS, concluding a need exists for a system “for synchronized transmission”). The DRMS depicted in Figure 1 is, therefore, a prior-art system that Foster’s system is meant to replace and not an embodiment of or foundation for Foster’s inventive system that Petitioner otherwise relies upon.

Thus, whether DRMS allegedly stores content units on a streaming media server or personal computer has no bearing on whether Foster’s system does so. This alone should defeat anticipation as, under such a ground, Foster’s inventive system may not be combined with its admitted prior art. *BlephEx, LLC v. Myco*

Paper 33 at 10-11

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’609 Ground 2: An obvious VOD implementation of Foster meets SFI’s new construction because the processor execute codes to store data in a storage device for later download

Reply

Hypothetical content-owner storage would not negate benefits of content-service providers (“CSPs”) storing Foster-encrypted content. POSAs knew CSPs (e.g., cable-TV companies) often purchased/licensed content from content owners and then securely stored it before delivering to recipients. Rubin-Depo., 84-94. CSPs typically and beneficially encrypted and stored that content for repeated use (e.g., downloads by different customers) – rather than repeatedly encrypting it on-the-fly – and for backup purposes. Petition, 77-78 (citing EX1045, EX1050, EX1059-EX1060; Bhattacharjee, ¶¶206-213); Bhattacharjee-Reply, ¶¶102-114 (citing EX1073, [0043] (VOD “pre-encrypt[s]” and “stores the content”); EX1097, 2:31-53 (“real-time encryption [for VOD] poses much greater cost and space issues”); EX1075, 1:51-67; EX1105, [0034]-[0037]; EX1080, 2-7).

’609 Paper 25 at 15

SFI’s Sur-reply

Does Not Dispute

SFI’s Sur-reply

Rather, Petitioner argues that there are benefits of “content-service providers (‘CSPs’)” storing content, and CSPs were known to have stored encrypted content. But Foster’s system is not a CSP. Rather, Foster teaches that a CSP “provides the ‘pipeline’ through which content is delivered from a content owner to a consumer” (Ex. 1006, [0039]) and that the CSP is a “source” of content for Foster’s system which provides encryption and synchronized transmission (*id.*, [0042], [0078]). Thus, Foster’s system receives content ***from*** a CSP; it is not a CSP storing content. *Id.* That CSPs securely store content does not provide motivation for Foster’s system to store content. And Petitioner’s “corroborating evidence,” old and new, still does not demonstrate that a POSITA would be motivated to modify an intermediary such as Foster’s system, meant only for encryption or securing content, such that it would store the underlying content. As Dr. Rubin testified, using intermediaries allows each to “focus on what they were good at” (Ex. 1099, 93:13-22): intermediary servers can focus on storing content and Foster’s system can focus on encryption and synchronized transmission.

Paper 33 at 17-18

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’609 Ground 2: SFI’s “instructions” argument fails also because when Foster’s encrypts content for an MPEG “program stream” the processor executes code for storing data in “fixed storage”

Reply

Additionally, Foster’s processor executes code to encrypt content for one of “two different system options”: “transport stream” (FIG. 14) and “program stream” (FIG. 15). Foster, [0073]. POSAs knew these were two processing options for the MPEG-2 standard. Rubin-Depo., 82-83 (admitting these are “different ways of encoding MPEG data,” but he did not “discuss[] the distinctions”). Foster explicitly says the program stream “is generally used in more reliable transmissions such as those…**in fixed storage**.” Foster, [0073]; EX2014, 1 (“[p]rogram streams are designed for…**local file storage**”). SFI’s expert similarly admits “Foster’s system focuses upon streamed content [] or **content stored on a DVD**” (IPR2024-00215, EX2033, ¶140), which the ’609 patent says is a storage device (EX1001, 66:36-45 (“optical disk”); Paper 3, 56). Bhattacharjee-Reply, ¶¶76-77. Thus, when Foster’s processor executes code to produce encrypted program-stream segments it is **not** “agnostic” to whether that data is for storage (POR, 43-44) because Foster says the program stream is “used in…**fixed storage**” (Foster, [0073]).

’609 Paper 25 at 10

EX2014

Program Streams

Program streams are designed for environments that are relatively error-free, such as local file storage. In a program stream, the PES packets are multiplexed and organized

EX2014, 1 (cited Reply, 9)

SFI’s Sur-reply

Petitioner’s new reliance on the “program stream” also fails. As Foster explains, an MPEG stream can be encoded into a program stream or transport stream, which are each used for different environments. Ex. 1006 [Foster] [0073]. Encoding a stream such that it is optimized for a certain environment, however, is still not the same as the processor executing code ***to perform*** storing the content units. Encoding a stream of packets does not equate to storing them. Moreover, the “program stream” is irrelevant to Petitioner’s original theory because, as Dr. Bhattacharjee confirms, in the cable environment that set-top boxes are in, a transport stream—which does not require storage of the packets being transported—is used rather than a program stream. Ex. 1006 [Foster] [0073] (“A transport stream is typically used in error-prone broadcast environments such as digital television.”); Ex. 2046 [Bhatttacharjee-2nd-Depo.] 30:16-31:10.

Paper 33 at 12-13

SFI’s Dr. Rubin

140. Foster’s system, however, has no need for any of these alleged benefits. Foster’s system simply processes content and transmits streamed content to a particular recipient. Ex. 1006 [Foster] Fig. 16; [0078]. Because Foster’s system focuses upon streamed content (or content stored on a DVD) and provides a

’854 EX2033, ¶140 (cited ’609 Reply, 10)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Agenda

1. Tajima-based Grounds: ’116 Grounds 1-2 (independent claim 1 + claims 5-8)
2. Foster-Hayhurst Grounds: ’116 Ground 5 (independent claims 1 & 9 + claims 2, 5, 11)
   ’854 Ground 2 (all challenged claims 1-10)
3. Foster Ground: ’609 Ground 2 (all challenged claims 1-10)
4. **Orsini Grounds: ’854 Ground 1 (all challenged claims 1-10)**
   ’609 Ground 1 (all challenged claims 1-10)
5. Tajima-Rivest-based Grounds: ’116 Ground 3 (independent claim 9 + claims 10-14)
   ’116 Ground 4 (independent claim 1 + claims 2-3, 6)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Overview of the ’609 & ’854 Petitions’ Grounds

’609 Patent

GROUND	CLAIMS	BASIS
1. Orsini (EX1005)	**1-10**	**Obviousness/ Anticipation**
2. Foster (EX1006)	1-3, 7-8 4-6, 9-10	102/103 103

’854 Patent

GROUND	CLAIMS	BASIS
1. Orsini	**1-10**	**Obviousness/ Anticipation**
2. Foster, Hayhurst (EX1007)	1-10	102/103

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’854/’609 Ground 1: The parties’ remaining dispute is narrow – whether a POSA had motivation to “further secure” Orsini’s “data chunks” under SFI’s claim scope

’854 Claim 1

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

[1A] receiving an external key from an external storage system,;

[1B] generating a plurality of data chunks based on the data set, such that the data set can be reconstructed using at least a minimum number of the plurality of data chunks, wherein generating the data chunks comprises:

[1B1] distributing the data set into a plurality of shares, wherein each of the shares comprises less than all of the data set,

[1B2] accessing a plurality of distinct encryption keys,

[1B3] encrypting each of the shares with a respective one of the plurality of distinct encryption keys,

[1C] **performing an encryption operation based on the external key to further secure the plurality of data chunks;** and

[1D] storing with the plurality of data chunks data indicative of at least one of the distinct encryption keys on a plurality of different storage devices.

’609 Claim 1

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

[1A] executing code by a processor to perform:

[1B] receiving a first key from a storage system;

[1C1] generating a plurality of data chunks based on a data set, wherein each data chunk of the plurality of data chunks comprises less than an entirety of data of the data set, and wherein the data set can be reconstructed using at least a minimum number of the plurality of chunks;

[1D] encrypting each respective data chunk of the plurality of data chunks with a respective second key, wherein each of the respective second keys are distinct from each other;

[1E] **performing a cryptographic operation based on the first key to further secure the plurality of data chunks;** and

[1F] storing, in a memory coupled to the processor, at least one data chunk of the plurality of data chunks with data indicative of at least one of the distinct encryption keys on at least one storage device.

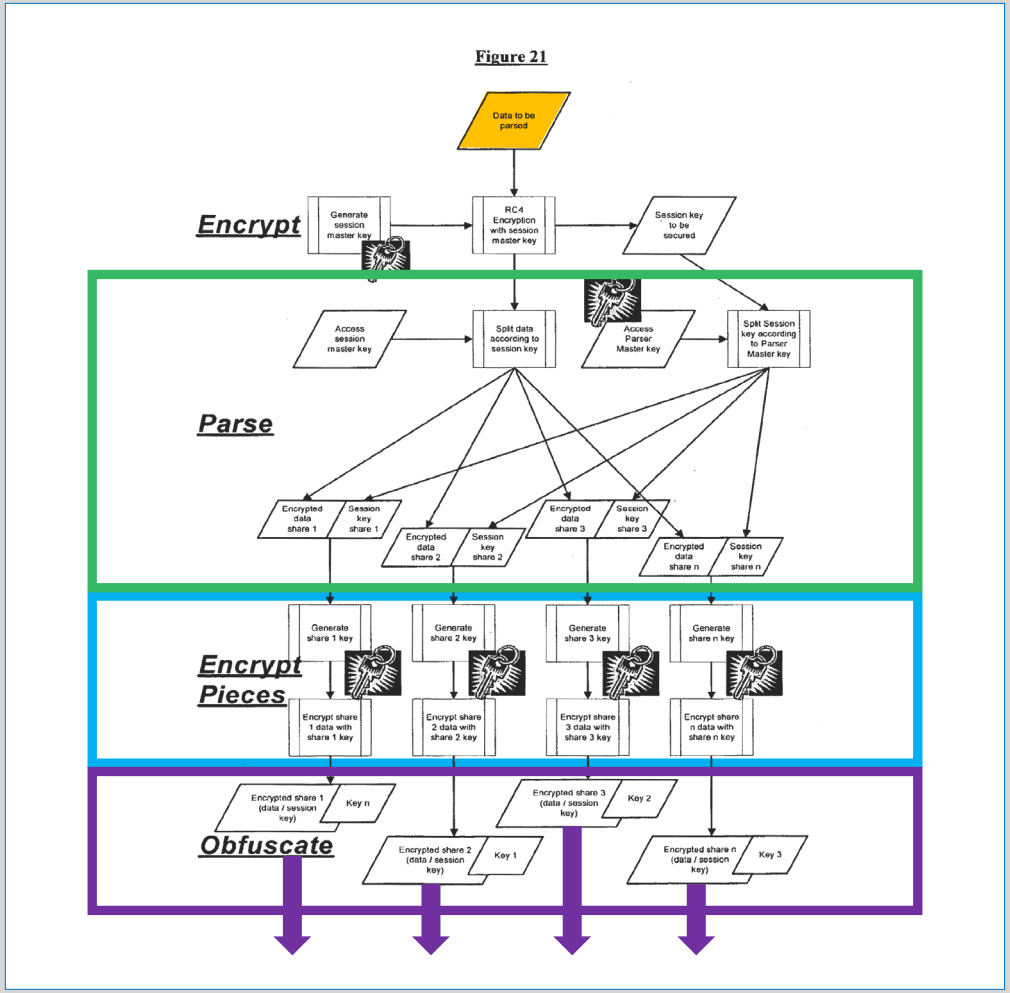
’116 Paper 2 at 13

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’854/’609 Ground 1: Orsini renders all challenged claims obvious

	’854 Patent, Claim 1	Orsini
[1PRE]	1. A method for securely storing a data set, the method comprising:	[0294] As shown in FIG. 21, this embodiment of the present invention shows the steps of the process performed by the parser software suite on data to store the session master key with the parsed data:
[1A]	receiving an external key from an external storage system;	[0295] 1. Generating a session master key and encrypt the data using RSl stream cipher.
[1B]	generating a plurality of data chunks based on the data set, such that the data set can be reconstructed using at least a minimum number of the plurality of data chunks, wherein generating the data chunks comprises:	[0296] 2. Separating the resulting encrypted data into four shares or portions of parsed data according to the pattern of the session master key.
[1B1]	distributing the data set into a plurality of shares, wherein each of the shares comprises less than all of the data set,	[0297] 3. In this embodiment of the method, the session master key will be stored along with the secured data shares in a data depository. Separating the session master key according to the pattern of the Parser Master Key and append the key data to the encrypted parsed data.
[1B2]	accessing a plurality of distinct encryption keys,	[0298] 4. The resulting four shares of data will contain encrypted portions of the original data and portions of the session master key. Generate a stream cipher key for each of the four data shares.
[1B3]	encrypting each of the shares with a respective one of the plurality of distinct encryption keys,	[0299] 5. Encrypting each share, then 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 gets Key 2, Share 4 gets Key 3.
[1D]	performing an encryption operation based on the external key to further secure the plurality of data chunks; and	
[1E]	storing with the plurality of data chunks data indicative of at least one of the distinct encryption keys on a plurality of different storage devices.	

Orsini



EX1005, FIG. 21, [0294]-[0299]
(cited ’854 Petition 11; ’609 Pet., 12)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’854/’609 Ground 1: The Petition’s argument that it was obvious to use a KEK was independent of Orsini’s public-key disclosures

’609 Patent, Claim 1

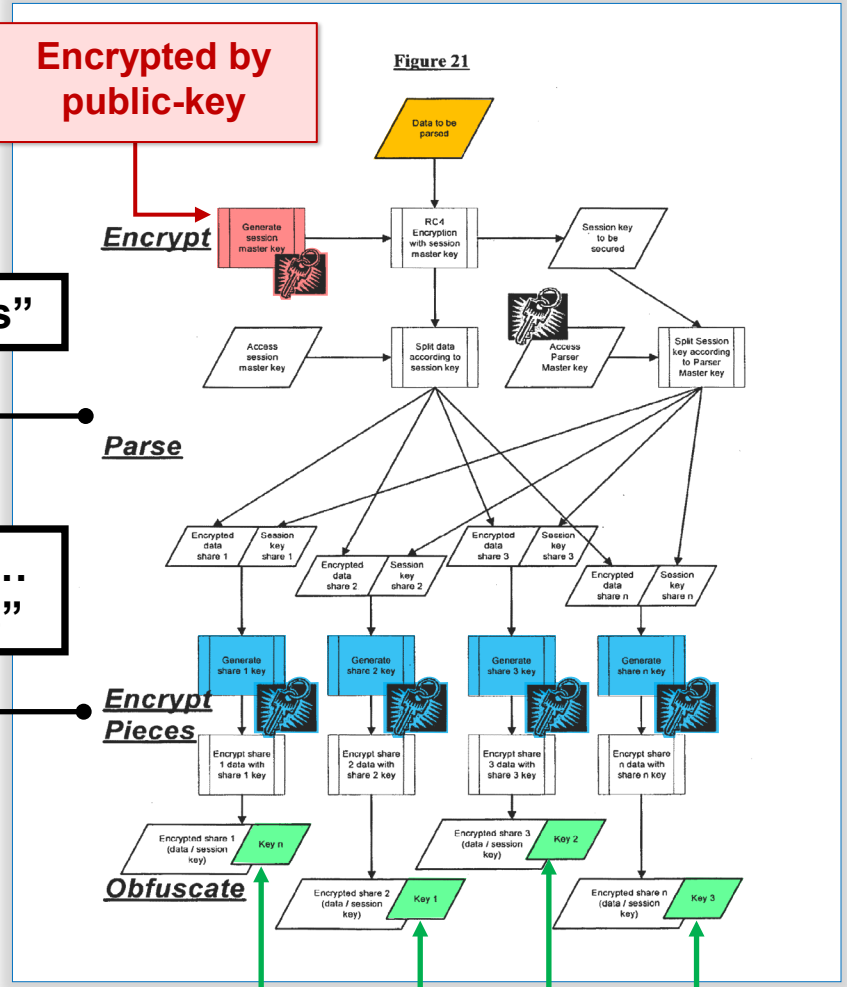
[1PRE]	1. A method for securing data, the method comprising:
[1A]	executing code by a processor to perform:
[1B]	receiving a first key from a storage system;
[1C1]-[1C2]	generating a plurality of data chunks based on a data set, wherein each data chunk of the plurality of data chunks comprises less than an entirety of data of the data set, and wherein the data set can be reconstructed using at least a minimum number of the plurality of chunks;
[1D]	encrypting each respective data chunk of the plurality of data chunks with a respective second key, wherein each of the respective second keys are distinct from each other;
[1E]	performing a cryptographic operation based on the first key to further secure the plurality of data chunks; and
[1F]	storing, in a memory coupled to the processor, at least one data chunk of the plurality of data chunks with data indicative of at least one of the distinct encryption keys on at least one storage device.

“data chunks”

“encrypting… data chunks”

Orsini

Encrypted by public-key



Left Unencrypted

EX1005, FIG. 21 (cited ’854 Pet., 10; ’609 Pet., 11)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’854/’609 Ground 1: The Petition’s argument that it was obvious to use a KEK was independent of Orsini’s public-key disclosures

’854 Petition

iv. POSAs had reason to encrypt [1B3]’s distinct encryption keys with an externally-stored key-encryption key

As discussed above (§VI.B.7.a.iii, [1C]), Orsini discloses (or alternatively renders obvious) encrypting session keys with a public-key encryption key. [0172]. Thus, Orsini discloses embodiments that encrypt encryption keys with other encryption keys, the latter of which are known in the art as key-encrypting keys (“KEKs”). Bhattacharjee, ¶131 (citing EX1035, EX1036).

As discussed below (§VI.B.8, [1D]), Orsini stores [1B3]’s distinct encryption keys (with “*data chunks*”) on storage systems. However, POSAs knew, in general, “cryptographic keys should never occur in clear form outside the cryptographic device.” EX1035, 12:15-23, 10:42-62; *see also* EX1036, 1:66-2:6 (when “encryption keys are included with the encrypted message” “[s]ecurity” comes from “encrypt[ing] the encryption keys themselves” with “a key encryption key”). Thus, to avoid a potential security risk in storing [1B3]’s encryption keys in clear (unencrypted) form on storage devices (especially those accessible from the public Internet), POSAs had reason to encrypt [1B3]’s keys with a KEK. Bhattacharjee, ¶132 (citing EX1035 at 8:15-22, which explains that there are “two categories of cipher keys”: “operational keys (KO) and key encrypting keys (KEK)”).

’854 Paper 2 at 43

Institution Decision

iv. Encrypting the Distinct Encryption Keys with Externally-Stored KEK

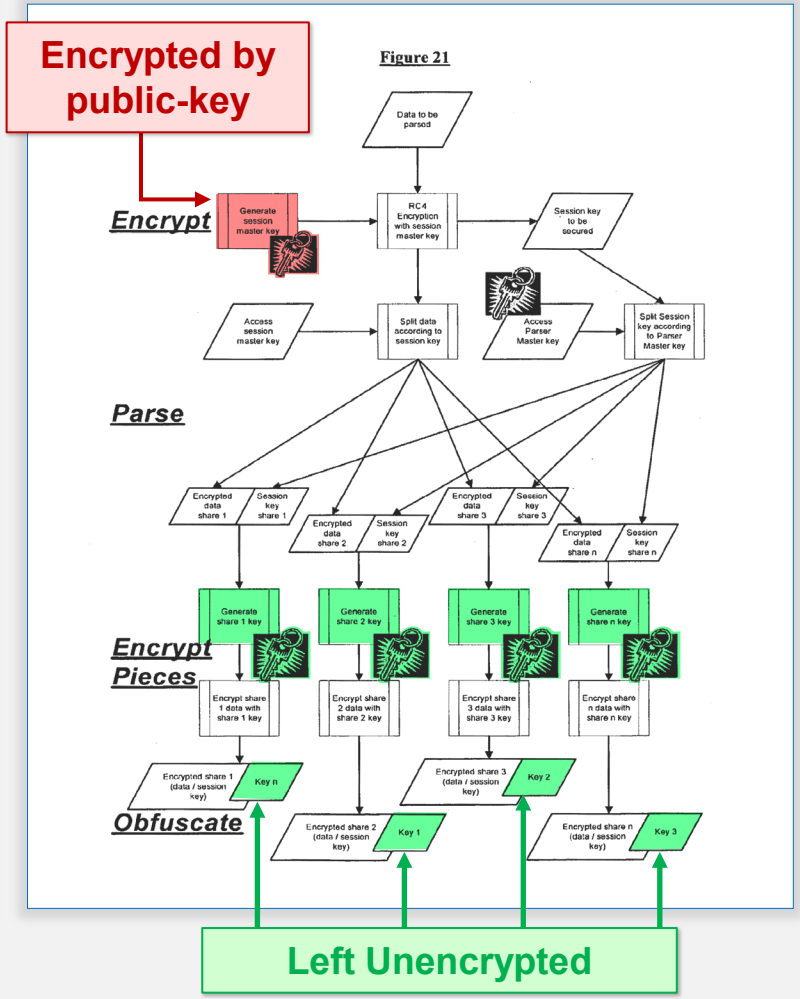
Petitioner argues that a person of ordinary skill in the art had reason to encrypt the distinct encryption keys with an externally stored KEK. Pet. 43–45.

Petitioner argues that Orsini discloses or renders obvious encrypting session keys with a public-key encryption key. *Id.* at 43. For the reasons discussed in § II.E.2.a.iii, Petitioner’s arguments about the public-key encryption key are deficient. In particular, Orsini does not show a public key in Figure 21. *See* Prelim. Resp. 19–20. Petitioner’s argument about the public-key encryption key is based on the features from multiple embodiments. *See* Pet. 38–39. Yet Petitioner does not provide a sufficient rationale for combining these multiple embodiments. *See id.*

Petitioner’s analysis in § VI.B.b.iv of the Petition does not remedy the deficiencies in Petitioner’s analysis of the public-key encryption key. *See* § II.E.2.a.iii *supra*. Rather, the reasoning in § VI.B.b.iv depends on the Petition’s analysis about Orsini disclosing or rendering obvious encrypting session keys with a public-key encryption key. *See* Pet. 43 (citing Ex. 1005 ¶ 172). Thus, Petitioner has not shown a reasonable likelihood in demonstrating that claim 1 would have been obvious under its rationale about encrypting the distinct encryption keys with an externally stored KEK. *See* § II.E.2.a.iii *supra*.

’854 Paper 15 at 38-39

Orsini



EX1005, FIG. 21 (cited ’854 Pet., 10; ’609 Pet., 11)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’609/’854 Ground 1: The Parties’ disputes are narrow

- No dispute that Orsini is prior art and discloses all limitations of ’854 claim 1 and ’609 claim 1 except the “further secur[ing]” limitations (’854, [1C]; ’609 [1E])

- **Only dispute is whether a POSA would have had motivation to encrypt Orsini’s cleartext (unencrypted) share keys with a conventional KEK**

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Orsini (Obviousness): POSAs knew encrypting keys with a KEK added additional security beyond what Orsini's obfuscation provides

'609 Reply

Orsini's obfuscation also disregards industry-wide best practices. The National Institute of Standards and Technology ("NIST") advised POSAs that when, as in Orsini, a "key exists external to the cryptomodule, **confidentiality protection shall be provided either by encryption (e.g., key wrapping)** [*i.e.*, a KEK]" or physically protecting keys (like using a safe, which would be inconsistent with Orsini's **electronic** storage of keys). EX1106, 34, 61, 123-125. Bhattacharjee-Reply, ¶¶171-176.

'609 Paper 25 at 26 (citing EX1106, [sic:] 72);* '854 Paper 24 at 19 (citing EX1106)

[12] Because Orsini is electronically storing key information with the data, using physical security (like storing the plain text keys in a safe) is not compatible with Orsini's system (let alone a practical solution for most electronic communication and storage systems).

Dr. Bhattacharjee

176. EX1106 further directs POSAs that the "*confidentiality* of all symmetric and private keys **shall** be protected." EX1106, 72 (all emphasis original). It further states that "[w]hen the symmetric or private key exists external to the cryptomodule, confidentiality protection **shall be provided either by encryption (e.g., key wrapping)** or by controlling access to the key via physical means (e.g. storing the keying material in a safe with limited access)." EX1106, 72 (emphasis of "shall" original).[12] Orsini's method stores the share encryption keys electronically with the data, and so storing the keying material physically in a safe would be contrary to Orsini's system. Thus, EX1106 informs POSAs (as they already knew and consistent with the opinions I offered in my First Declaration), to have "confidentiality protection" of the share keys "provided [] by encryption (e.g., key wrapping)" (i.e., a KEK). EX1106, 72.

'609 EX1111, ¶176 (cited POR, 26); '854 EX1111, ¶193 (cited POR, 19)

Orsini (Obviousness): A POSA would have had motivation to encrypt Orsini's share keys with a KEK because it was conventional and had known benefits

'854 Petition

iv. POSAs had reason to encrypt [1B3]'s distinct encryption keys with an externally-stored key-encryption key

As discussed above (§VI.B.7.a.iii, [1C]), Orsini discloses (or alternatively renders obvious) encrypting session keys with a public-key encryption key. [0172]. Thus, Orsini discloses embodiments that encrypt encryption keys with other encryption keys, the latter of which are known in the art as key-encrypting keys ("KEKs"). Bhattacharjee, ¶131 (citing EX1035, EX1036).

As discussed below (§VI.B.8, [1D]), Orsini stores [1B3]'s distinct encryption keys (with "*data chunks*") on storage systems. However, POSAs knew, in general, "cryptographic keys should never occur in clear form outside the cryptographic device." EX1035, 12:15-23, 10:42-62; *see also* EX1036, 1:66-2:6 (when "encryption keys are included with the encrypted message" "[s]ecurity" comes from "encrypt[ing] the encryption keys themselves" with "a key encryption key"). Thus, to avoid a potential security risk in storing [1B3]'s encryption keys in clear (unencrypted) form on storage devices (especially those accessible from the public Internet), POSAs had reason to encrypt [1B3]'s keys with a KEK. Bhattacharjee, ¶132 (citing EX1035 at 8:15-22, which explains that there are "two categories of cipher keys": "operational keys (KO) and key encrypting keys (KEK)").

'854 Paper 2 at 43; '609 Pet., 40

Does Not Dispute

EX1036

Security 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. This approach means that the data encryption keys will be encrypted by a key encryption key. This in turn leads to the

'854 EX1036, 1:66-2:4 (cited '854 Pet., 43; '609 Pet., 40)

EX1035

Because the ciphering algorithm used is not secret, the degree of protection that can be derived from a cryptographic system ultimately depends upon the security of the cryptographic keys. Therefore, the objectives of key management are: (1) cryptographic keys should never occur in clear form outside the cryptographic device, except under secure conditions during the period when keys are originally distributed and installed or when stored in a secure place such as a safe, vault or similar location for backup or recovery and (2) no cryptographic operation, or combination thereof, using any cryptographic quantities which are routinely stored or routed through the system, or derived therefrom, should permit clear keys to be recoverable outside the cryptographic device. Therefore, in keeping with the first objective, if the system generated terminal keys are to be stored at the host system they must be protected by being enciphered under another key. One

'854 EX1035, 10:42-59 (cited '854 Pet., 18; '609 Pet., 40)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’854/’609 Orsini Ground 1: SFI’s Arguments

SFI argues POSAs would not have had motivation to encrypt Orsini’s share keys with a conventional KEK because:

1. Further securing data based on an encryption key (’854 [1C]; ’609 [1E]) was a “missing limitation”

2. Encrypting Orsini's keys would have no benefit in view of Orsini’s non-cryptographic obfuscation

3. Encrypting Orsini's keys would have drawbacks in managing and securing the KEK

’854 POR, 19-32; ’854 Sur-Reply, 19-26
’609 POR, 17-32; ’609 Sur-Reply, 21-29

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’854/’609 Orsini (Obviousness): SFI’s arguments fail

1. Encrypting keys with a KEK was not missing in the prior art; it was well-known and conventional

2. The cryptographic protection of a KEK is independent of and non-redundant with Orsini’s non-cryptographic obfuscations

3. Managing KEKs was well-known and conventional in the prior art

’609 Pet., 36-37, 39-40, 42-43; ’609 Reply, 18-30
’854 Pet., 39-40, 43-45; ’854 Reply, 16-29

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’854/’609 Orsini (Obviousness): SFI’s Argument #1 (“missing limitation”) fails because encrypting keys with a KEK is not a missing limitation in the prior art

SFI’s POR

But “common sense” can be used to supply a missing limitation “***only*** when ‘the limitation in question [is] unusually simple and the technology particularly straightforward.’” *DSS Tech. Mgm’t v. Apple Inc.*, 885 F.3d 1367, 1374-75 (Fed. Cir. 2018) (citations omitted). And where, as here, a “missing limitation goes to the heart of an invention” a court’s review of any purported “reasoned basis for resort to common sense” must be especially “searching” *Arendi S.A.R.L. v. Apple Inc.*, 832 F.3d 1355, 1363 (Fed. Cir. 2016).

’854 Paper 21 at 19-20;
’609 POR, 18

SFI’s Sur-Reply

DOES NOT DISPUTE

’854 Paper 32 at 19-26;
’609 Sur-Reply, 21-29

Reply

[1C] is not missing in the prior art. As discussed *supra* §II.A, the Petition cited EX1035’s **and** EX1036’s disclosures of the known desirability of using a KEK to encrypt an encryption key for storage. Petition, 43; *see infra* §II.A.3. As the Board found, Foster also discloses [1C]. DI, 13, 24. Other references also teach using KEKs to encrypt keys that encrypt “partitions[s]” of “digital content.” EX1088, 2:20-3:16; *see also* EX1080, 2-7; EX1106, 34, 61, 123-125. Bhattacharjee-Reply, ¶¶200-210.

Moreover, the Petition did not rely on only “common sense.” The Petition provided reasoned analysis — supported by multiple pieces of corroborating evidence and expert testimony — detailing why POSAs would further secure Orsini’s share keys using the well-known (not SFI-invented) technique of encrypting encryption keys. Bhattacharjee-Reply, ¶¶211-214.

’854 Paper 24 at 21-22
’609 Reply, 23

“**That the proposed *combination* of James and POH—rather than one of the individual references—discloses the disputed claim limitations does not defeat the Board's conclusion of obviousness**….[I]t is sufficient that a person of ordinary skill in the art would have been motivated to combine the prior art in a way such that the combination discloses the claim limitations.”

Fleming v. Cirrus Design Corp., 28 F.4th 1214, 1222 (Fed. Cir. 2022) (cited ’854 Reply, 22; ’609 Reply, 24)

“**In *Arendi***, the Board relied on **nothing more than conclusory statements and unspecific expert testimony** in finding that it would have been common sense to supply a limitation that was admittedly missing from the prior art. Conversely, here **the Board relied on expert evidence, which was corroborated by Hua, in concluding that pipelining was not only in the prior art, but also within the general knowledge of a skilled artisan**.”

Koninklijke Philips N.V. v. Google LLC, 948 F.3d 1330, 1339 (Fed. Cir. 2020) (cleaned up) (cited ’854 Reply, 22; ’609 Reply, 24)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’854/’609 Orsini (Obviousness): SFI’s Argument #1 (“missing limitation”) fails because using a KEK to encrypt keys was conventional and had well-known benefits

’854 Reply

The Petition demonstrated, supported by expert testimony **and multiple pieces of corroborating evidence**, that POSAs would have encrypted Orsini’s keys with a KEK to prevent an attacker from reconstructing the encrypted dataset, even when they possessed every encrypted share. Petition, 43-44. The Petition’s corroborating evidence was direct in its teaching: “cryptographic keys should **never** occur in clear [*i.e.*, unencrypted] form outside the cryptographic device.” EX1035, 12:15-23 (cited Petition, 43).

Petitioner’s cited prior art explains that POSAs knew that when an encryption key is transmitted or stored (as in Orsini), the system should “encrypt the encryption keys themselves…[with] a key encryption key” EX1036, 1:66-2:6 (cited Petition, 43). That was true whether the “encryption keys were included with the encrypted message” (*id.*) or stored separately from and/or obfuscated with encrypted data-portions (EX1035, 10:42-62; EX1036, 1:66-2:6; EX1086, 7:3-22; EX1087, [0017]-[0025]; EX1112, [0022]-[0027] (“Wherever the session key is stored, it is preferable that it is encrypted beforehand.”). Petition, 43-45. Bhattacharjee-Reply, ¶¶157-179.

’854 Paper 24 at 18; ’609 Reply, 20

EX1087

[0017] A first aspect of the present invention is a method of securely storing data. The data are stored initially in a first portable storage device. Then, for security, the data are encrypted using a backup key, thereby transforming the data into encrypted backup data. The backup key is protected as a protected backup key. When the data need to be restored, the protected backup key is retrieved and the encrypted backup data are decrypted.

[0020] Preferably, protecting the backup key includes encrypting the backup key, and retrieving the protected backup key includes decrypting the protected backup key. More preferably, the backup key is encrypted using an encryption password that is stored in a backup key storage device that is used to protect the backup key by storing the backup key therein; and the method of the present invention also includes providing an alleged encryption password to the backup key storage device. Examples of backup key storage devices include an internet server and a trusted computer that are used to store the backup key. The backup

EX1087, [0017], [0020]
(cited ’854 Reply, 18; ’609 Reply, 20)

EX1112

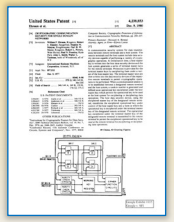
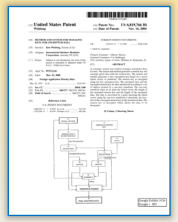
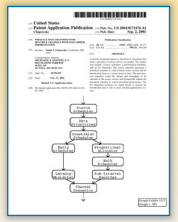
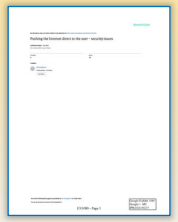
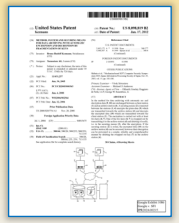
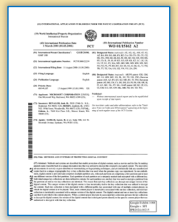
session, the public key is also known as a “session key”. As a unique session key is used to encrypt each individual field of data stored in the data store, a recipient in possession of one particular session key will not be able to access other encrypted data which is stored at the data store.

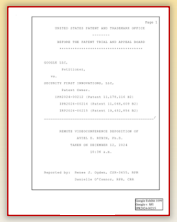
[0027] The session key may be stored in the same data store used to store the encrypted data, or in a different data store. Wherever the session key is stored, it is preferable that it is encrypted beforehand. The session key may be encrypted using the master key. Again, the advantage of

EX1112, [0025], [0027[(cited ’854 Reply, 18; ’609 Reply, 20)

Additional Corroborating Evidence

EX1035 EX1036 EX1037 EX1080 EX1086 EX1088 EX1106 Dr. Rubin



Cited ’854 Pet. 43; ’854 Reply, 18, 21-22;
’609 Pet., 40; ’609 Reply, 20-21

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’854/’609 Orsini (Obviousness): SFI’s Argument #2 (“no benefits”) is contradicted by SFI’s own argument and SFI’s own expert

’854 POR

Petitioner’s purported motivation should be rejected because adding an “encryption operation based on the external key” to Orsini is useless because, unlike the ’854’s claim 1, Orsini ***already*** uses a ***non***-cryptographic operation ***that is not based on any key*** to further secure the data chunks. Ex. 1005 [Orsini] Fig. 21, [0299]. As Dr. Rubin explains:

> Orsini already teaches hiding the encryption key used to encrypt each data chunk with a different chunk through a **non-cryptographic process**, illustrated in Figure 21 below.
>
> * * *
>
> Petitioner fails to explain why a POSITA would be motivated to modify Orsini using other allegedly “known” methods, particularly when it non-cryptographically hides each share key with a share it did not encrypt, and teaches that this provides a “comprehensive package for data security.” Ex. 1005 [Orsini] [0380]; Ex. 2033 [Rubin-2nd-Decl.] ¶ 71. It is not obvious to modify a reference that already achieves the benefit supposedly served by the proposed modification.

’854 Paper 21 at 21-23; ’609 Reply, 20-22



SFI’s Expert Dr. Rubin

41. Orsini’s Figure 21 merely hides the keys used to encrypt the data chunks by storing each key with a chunk it was not used to encrypt. Ex. 1005 [Orsini] [0299]. Orsini’s non-cryptographic approach, however, cannot protect a dataset if an attacker has access to all of the shares (and thus access to all of the share keys) and can therefore reveal the share keys and use them to decrypt the corresponding shares. In that case, reconstruction of the original dataset becomes straight-forward. Orsini’s Figure 21 does not teach any method of “further securing” the dataset after the data shares have been encrypted with the share keys other than by hiding the share keys with different shares.

’854 EX2033, ¶41 (cited ’854 Reply, 18; ’609 Reply, 20)

SFI’s ’116 Sur-Reply

In fact, Foster discourages the use of its KEK technique with Tajima, by stressing the importance of keeping the title key with the content it encrypts. POR, 27. The only reason for using the KEK in Foster is ***because*** the title key is kept with the content it decrypts; the KEK provides security because it prevents someone with access to all of the content from decrypting it. *See* POR, 15-16;

’116 Paper 29 at 5

’854/’609 Orsini (Obviousness): SFI’s Argument #2 (“no benefits”) fails because SFI’s expert admits that Orsini does not protect a dataset when “an attacker has access to all of the shares”

SFI’s Expert Dr. Rubin

41. Orsini’s Figure 21 merely hides the keys used to encrypt the data chunks by storing each key with a chunk it was not used to encrypt. Ex. 1005 [Orsini] [0299]. Orsini’s non-cryptographic approach, however, cannot protect a dataset if an attacker has access to all of the shares (and thus access to all of the share keys) and can therefore reveal the share keys and use them to decrypt the corresponding shares. In that case, reconstruction of the original dataset becomes straight-forward. Orsini’s Figure 21 does not teach any method of “further securing” the dataset after the data shares have been encrypted with the share keys other than by hiding the share keys with different shares.

’854 EX2033, ¶41 (cited ’854 Reply, 18; ’609 Reply, 20)

Q. What does it mean by a "rogue employee"?

A. Somebody that's trying to bypass the security of the system or that's willing to break the rules.

Q. But it's someone who has access to the system, but is trying to break it?

A. Well, **employees in general have more** access to systems than people who don't work at a facility. And I think this is referring to the idea one of the threats that is mentioned in the patent, that **even a rogue employee with physical access needs to be protected against.**

’854 EX1099, 138:6-18 (cited ’854 Reply, 18; ’609 Reply, 21)

Q. Does the non-cryptographic step violate Kerckhoffs's Principle?

A. Kerckhoffs's Principle basically says that you should not put your security in anything other than the keys. And so obfuscating -- **performing an obfuscation is a -- is not putting the protection in the keys, so I would say this does not on its own provide security** --

’854 EX1099, 155 (cited ’854 Reply, 17; ’609 Reply, 19)

’854/’609 Orsini (Obviousness): SFI’s Argument #2 (“no benefits”) fails because SFI’s own expert admits that, unlike a KEK, Orsini’s “obfuscation” “does not on its own provide security”

SFI’s Expert Dr. Rubin

Q. So back to your page 25 of your '854 declaration. So that's a non-cryptographic obfuscation, correct?

A. It's a non-cryptographic step, and it's an obfuscation, yes.

Q. Would a person skilled in the art think that is a computationally intense obfuscation?

A. I don't think that the step would be computationally intensive.

Q. And I think, as we talked about before, it's simple math, right?

A. I don't even think it's really involving a lot of math. You're just putting something with other things and not putting it with other things.

’854 EX1099, 154 (cited ’854 Reply, 23; ’609 Reply, 21)

Q. Does the non-cryptographic step violate Kerckhoffs's Principle?

A. Kerckhoffs's Principle basically says that you should not put your security in anything other than the keys. And so obfuscating -- **performing an obfuscation is a -- is not putting the protection in the keys, so I would say this does not on its own provide security --**

’854 EX1099, 155 (cited ’854 Reply, 17; ’609 Reply, 19)

Q. And how long do you think it would take a person of skill to write out those combinations if there’s four shares and four keys?

A. Four shares, it wouldn't take long.

’854 EX1099, 24 (cited ’854 Reply, 23; ’609 Reply, 26)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’854/’609 Orsini (Obviousness): SFI’s Sur-Reply “security” argument fails because SFI’s own expert admits that Orsini’s obfuscation “cannot protect a data set if an attacker has access to all of the share”

’609 Sur-Reply

Moreover, the Reply does nothing to remedy the Petition’s failure to demonstrate that modifying Orsini to “also (or alternatively) encrypt[] Orsini’s share keys using a conventional KEK” (Reply, 25) would have increased security. Petitioner continues to ignore the POR’s demonstration that adding an additional encryption layer here would likely diminish Orsini’s overall system’s security by adding targets that an attacker could use to compromise the system.[6] POR, 24-25; Ex. 2006, 2; Ex. 2008, 1.

And Petitioner’s newly proposed modification would do exactly that, decreasing the security of the system as a whole. Petitioner’s original modification was to use a KEK with Orsini’s “***session*** key management” which could be modified to “***generat[e]***” the KEK. POR, 25-27; Pet., 23-24. But session key management is very different from KEK management, especially for storage and transport of the keys. POR, 25-27. As the POR demonstrated, the session key management techniques described in Orsini would not adequately protect KEKs. *Id.*

[6] Dr. Rubin’s testimony regarding increasing security through multiple encryption layers referred to the security of an encryption protocol, not the overall security of a system, contrary to Petitioner’s suggestion. *See* Ex. 2033 [Rubin-2nd-Decl.] ¶¶74-76.

’609 Paper 33 at 26-27; ’854 Sur-Reply, 24-25

SFI’s Expert Dr. Rubin

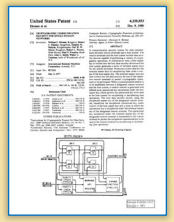
41. Orsini’s Figure 21 merely hides the keys used to encrypt the data chunks by storing each key with a chunk it was not used to encrypt. Ex. 1005 [Orsini] [0299]. Orsini’s non-cryptographic approach, however, cannot protect a dataset if an attacker has access to all of the shares (and thus access to all of the share keys) and can therefore reveal the share keys and use them to decrypt the corresponding shares. In that case, reconstruction of the original dataset becomes straight-forward. Orsini’s Figure 21 does not teach any method of “further securing” the dataset after the data shares have been encrypted with the share keys other than by hiding the share keys with different shares.

’854 EX2033, ¶41 (cited ’854 Reply, 18; ’609 Reply, 20)

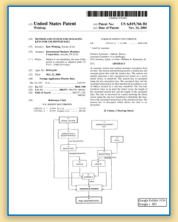
’854 Pet., 43; ’854 Reply, 18-19
’609 Pet. 40, ’609 Reply, 20-21

Corroborating evidence cited in Petitions and Replies

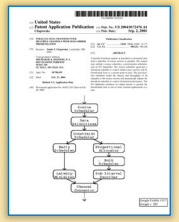
EX1035



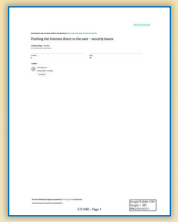
EX1036



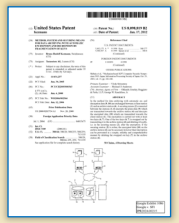
EX1037



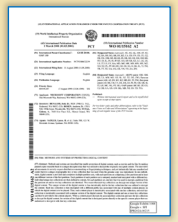
EX1080



EX1086



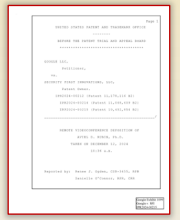
EX1088



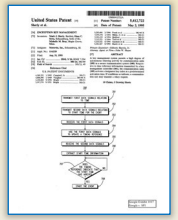
EX1106



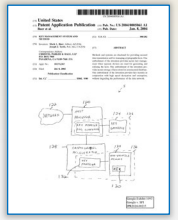
Dr. Rubin



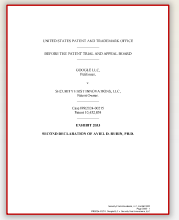
EX1017



EX1093



EX2003



’854/’609 Orsini (Obviousness): SFI’s Argument #3 (“key management”) fails because using and managing KEKs was well-known and conventional, as corroborated by multiple references

’609 Reply

b. The Alleged Drawbacks to KEKs Were Easily Overcome and Are Legally Irrelevant.

SFI argues “drawbacks associated with managing and securing” KEKs would have dissuaded POSAs from using KEKs in Orsini. POR, 25-32. But SFI does not allege that addressing these alleged “drawbacks” was beyond POSAs’ skill.

Orsini **already** taught techniques for managing and securing other KEKs, or (at least) could have been modified in a simple, straightforward way to use conventional techniques for managing KEKs. For example, Orsini supports “Triple DES and AES” (Orsini, [0378]), which taught using and managing KEKs. *Infra* §II.B; Bhattacharjee-Reply, ¶¶213-215. Orsini also supports “a wide number of alternative key management systems.” Orsini, [0107].

Petition-cited prior art demonstrates that using/managing KEKs was conventional. Petition, 43 (citing EX1035-EX1036). EX1036, e.g., discloses a “system for **managing keys**,” including KEKs (2:29-61) and says the “prior art” included “security infrastructure to solve the key management problem[]” (2:7-22). *See also* EX2032, 683-84 (“key management” for “KEKs”); EX1017, 1:50-1:68 (similar); EX1093, [0013], [0035]-[0039] (similar), [0057]-[0060]. As discussed above (§II.A.3.b), NIST urged POSAs to use KEKs and discussed related key-management issues. EX1106, 34, 61, 123-125. The evidence is clear that any drawbacks in managing KEKs were (at most) a conventional tradeoff and not beyond POSAs’ skill, and thus do not defeat obviousness. *Intel*, 61 F.4th at 1381; Rubin-Depo., 185-186. Bhattacharjee-Reply, ¶¶216-221.

’609 Paper 25 at 27-28; ’854 Reply, 20-22

SFI’s Sur-Reply

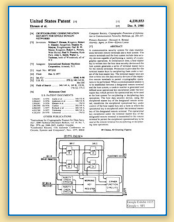
DOES NOT DISPUTE

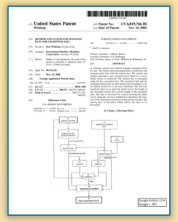
’609 Paper 25 at 20-22;
’854 Paper 32 at 25-26

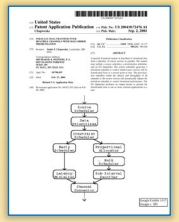
“[I]t is a commonplace fact that design decisions entail making tradeoffs among multiple objectives. *Allied Erecting and Dismantling Co. v. Genesis Attachments, LLC*, 825 F.3d 1373, 1381 (Fed. Cir. 2016) (‘A given course of action often has simultaneous advantages and disadvantages, and this does not necessarily obviate motivation to combine.’).”

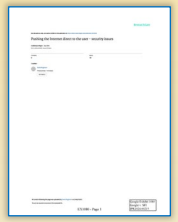
Corephotonics, Ltd. v. Apple Inc., 2021 WL4944471, at *6 (Fed. Cir. Oct. 25, 2021) (cited ’609 Reply, 18; ’854 Reply, 13)

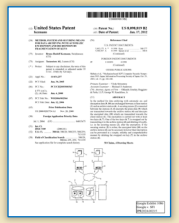
Evidence cited in Petitions and Replies

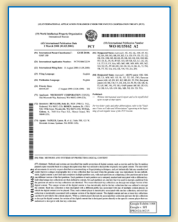
EX1035 

EX1036 

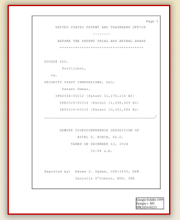
EX1037 

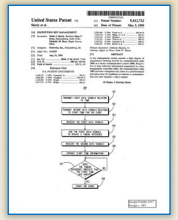
EX1080 

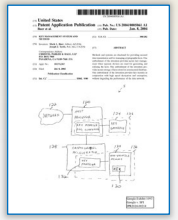
EX1086 

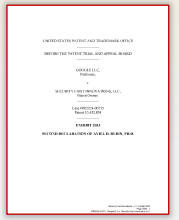
EX1088 

EX1106 

Dr. Rubin 

EX1017 

EX1093 

EX2003 

Agenda

1. Tajima-based Grounds: ’116 Grounds 1-2 (independent claim 1 + claims 5-8)
2. Foster-Hayhurst Grounds: ’116 Ground 5 (independent claims 1 & 9 + claims 2, 5, 11)
   ’854 Ground 2 (all challenged claims 1-10)
3. Foster Ground: ’609 Ground 2 (all challenged claims 1-10)
4. Orsini Grounds: ’854 Ground 1 (all challenged claims 1-10)
   ’609 Ground 1 (all challenged claims 1-10)
5. **Tajima-Rivest-based Grounds: ’116 Ground 3 (independent claim 9 + claims 10-14)**
   ’116 Ground 4 (independent claim 1 + claims 2-3, 6)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Overview of the ’116 Petition’s Grounds

GROUND	CLAIMS	BASIS
1. Tajima (EX1105)	1, 5-8	102/103
2. Tajima, Foster (EX1106)	1-2, 9, 11-14	103
3. Tajima, Rivest (EX1034)	**9-14**	**Obviousness**
4. Tajima, Rivest, Shyu (EX1016)	**1-3, 6**	**Obviousness**
5. Foster, Hayhurst (EX1007)	1-2, 5, 9, 11	103

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Grounds 3-4: Google’s constructions of “transform” / “transforming” and “data indicative …”

	’116 Patent, Claim 1	’116 Patent, Claims 2-3	’116 Patent, Claim 9	
[1PRE]	A method for securing a data set, the method comprising:	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.	9. A method for securing a data set, the method comprising:	[1PRE]
[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;	3. The method of claim 2, wherein **performing a transform** 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.	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;	[1A]
[1B]	encrypting each of the data chunks with a respective one of a plurality of encryption keys;		encrypting each of the data chunks with a respective one of a plurality of encryption keys;	[1B]
[1C]	**obfuscating** each of the plurality of different encryption keys; and;		**transforming** each of the plurality of different encryption keys; and	[1C]
[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.		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.	[1D]

Claim Constructions

“transform” / “transforming”	
Google	**SFI**
Changing without the use of a secret key the bits of a key / data indicative of a key	No construction necessary Alternatively: “changing the form”

EX1042, 5-7, 8-9; EX1043, 3-4, 5-6 (cited ’116 Pet., 47, 50)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

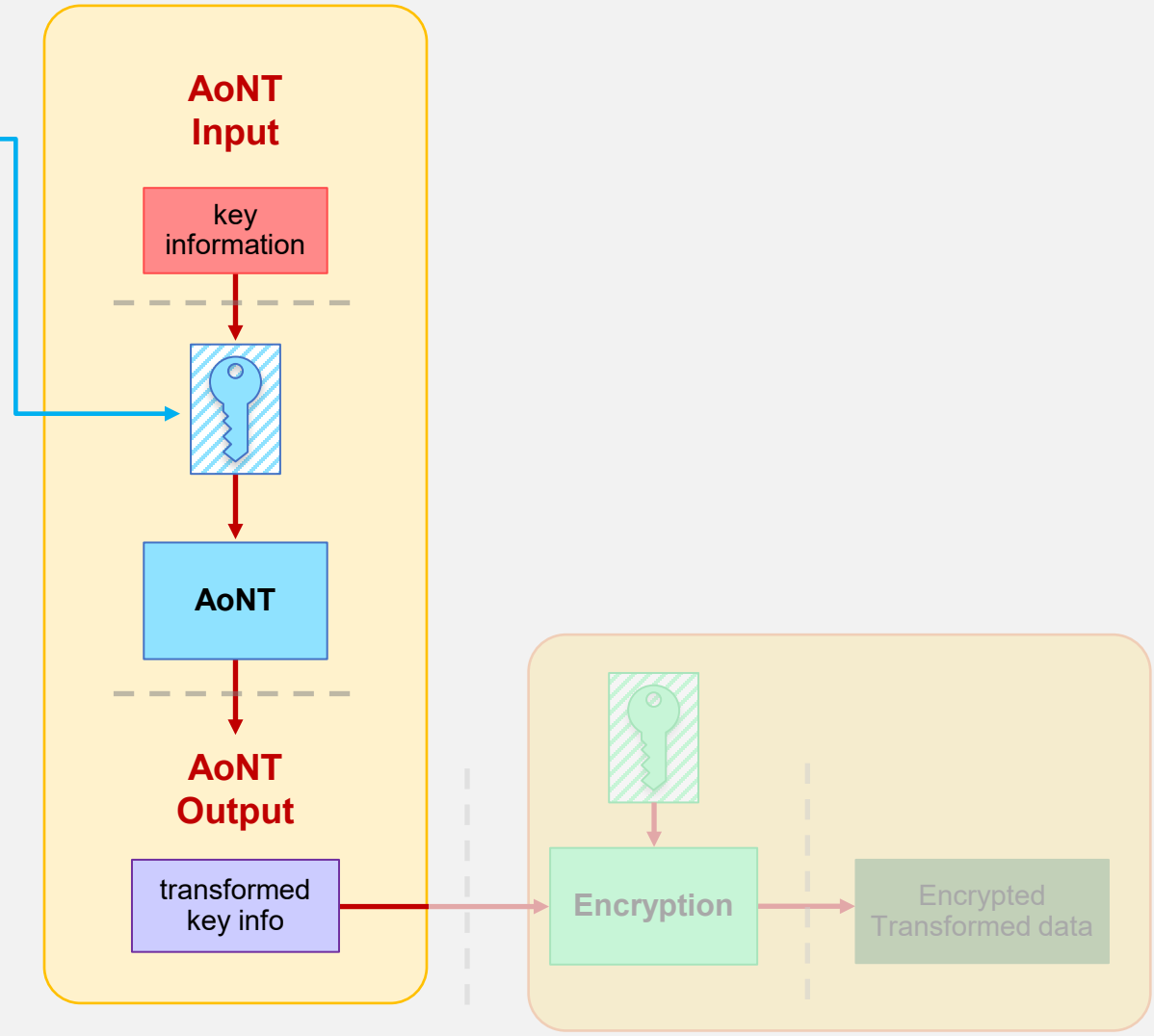
’116 Grounds 3-4: Using Rivest’s AoNT to transform Tajima’s “key information” based on Tajima’s public key

’116 Petition

Tajima-Rivest applies Rivest’s AoNT using the user’s public key provided to Tajima’s service sites (Tajima, [0070]-[0071], Fig. 6 (S2-S3) and thus beneficially avoids using secret keys. Rivest, 213-214; Bhattacharjee, ¶¶186-192.

C. Claim 9

Tajima-Rivest (§VII.B) transforms Tajima’s key information using AoNT and thereby changes the bits (and likewise changes the form) of Tajima’s keys where “no secret keys are used” (Rivest, 213), meeting [9C]’s “*transforming*” limitation under both Google’s construction (“Changing without the use of a secret key the bits of a key”—EX1042, 8) and SFI’s construction (“changing the form”—EX1043, 5-6). Bhattacharjee, ¶193; §VI.G *supra*. Claim 9’s other limitations are met for the same reasons discussed in §§VI.H.1 and VI.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 by Tajima-Rivest. Bhattacharjee, ¶194.



’116 Paper 2 at 47

’116 Ground 3: The Parties’ disputes are narrow

- No dispute that Tajima-Rivest meets challenged claims 9-14

- **Only dispute concerns whether POSAs had motivation to add Rivest’s AoNT to Tajima**

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 3: SFI’s arguments fail because they assume an attacker knew that AoNT would be used and which public key would be input to the AoNT algorithm, which are kept secret in Tajima-Rivest

SFI’s POR

A POSITA would not be motivated to apply “Rivest’s AoNT using the user’s public key” as Petitioner and its expert suggest. As Dr. Rubin explains,

> Rivest says nothing about the use of public keys and is not designed to be used with public keys. Rivest is generally used in connection with symmetric encryption systems that require a secret key. Public key encryption systems generally use different types of transformations. Instead of a public key, Rivest requires the use of a random number as the basis for its AoNT transformation. *See* EX1034 [Rivest], 213-14.

EX2033 ¶84.

Petitioner’s proffered motivation should also be rejected because it is contradicted by Rivest itself. Rivest explains that the AoNT standing alone does not require the use of a secret key only because, standing alone, it does not “protect” anything. EX1034, 212. Rather, the AoNT adds additional protection to a data set that is otherwise separately encrypted or secured. *Id.* Because Tajima does not

’116 Paper 17 at 28-29

’116 Patent

It will be understood that the AoNT may be applied to encryption keys used to encrypt the data portions following parsing and splitting either in place of or in addition to the encryption and AoNT of the respective encryption key corresponding to the data set prior to parsing and splitting. Likewise, AoNT may be applied to splitting keys.

EX1001, 77:37-42 (cited Pet., 3-4)

Reply

transformation. POR, 29-30. That argument fails because inverting Tajima’s transformed data would also require knowing (1) that AoNT was applied in the first place and (2) that a user’s public key, and which user’s public key, was used in the transformation—none of which is known by an attacker or apparent from the data fragment itself. Bhattacharjee-Reply, ¶¶48-50.

’116 Paper 21 at 28-29

SFI’s Sur-reply

DOES NOT DISPUTE

SFI’s Sur-reply

In Petitioner’s proposed combination, the opposite is true: The key is public knowledge, but use of AoNT is purportedly kept secret. But Rivest’s transformation algorithm is known from Rivest itself, and there are only a limited number of algorithms that might be used; an attacker could simply try them all. *Id.* Public keys are, by definition, public, accompanied by certificates that identify the key’s holder. EX1005 [Tajima], [0075]. A POSITA would recognize that the

’116 Paper 29 at 10

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

'116 Ground 3: SFI's argument also fails because Tajima's public key is a random number (consistent with Rivest), but in Tajima-Foster the particular public key that is used is kept secret, thereby providing security

'116 Patent

According to one embodiment of the present invention, an All or Nothing Transform (AoNT), such as a Full Package Transform, may be used to further secure data by transforming splitting keys, encryption keys, any other suitable data elements, or any combination thereof. For example, an encryption key used to encrypt a data set prior to parsing and splitting in accordance with the present invention may be transformed by an AoNT algorithm. The transformed encryption key may then be distributed among the data shares according to, for example, the Shamir algorithm or any other suitable algorithm. In order to reconstruct the encryption key, the encrypted data set must be restored (e.g., not necessarily using all the data shares if redundancy was used in accordance with the present invention) in order to access the necessary information regarding the transformation in accordance with AoNTs as is well known by one skilled in the art. When the original encryption key is retrieved, it may be used to decrypt the encrypted data set to retrieve the original data set. It will be understood that the fault tolerance features of the present invention may be used in conjunction with the AoNT feature. Namely, redundancy data may be included in the data portions such that fewer than all data portions are necessary to restore the encrypted data set.

It will be understood that the AoNT may be applied to encryption keys used to encrypt the data portions following parsing and splitting either in place of or in addition to the encryption and AoNT of the respective encryption key corresponding to the data set prior to parsing and splitting. Likewise, AoNT may be applied to splitting keys.

'116 EX1001, 77:13-42 (cited Pet., 3-4)

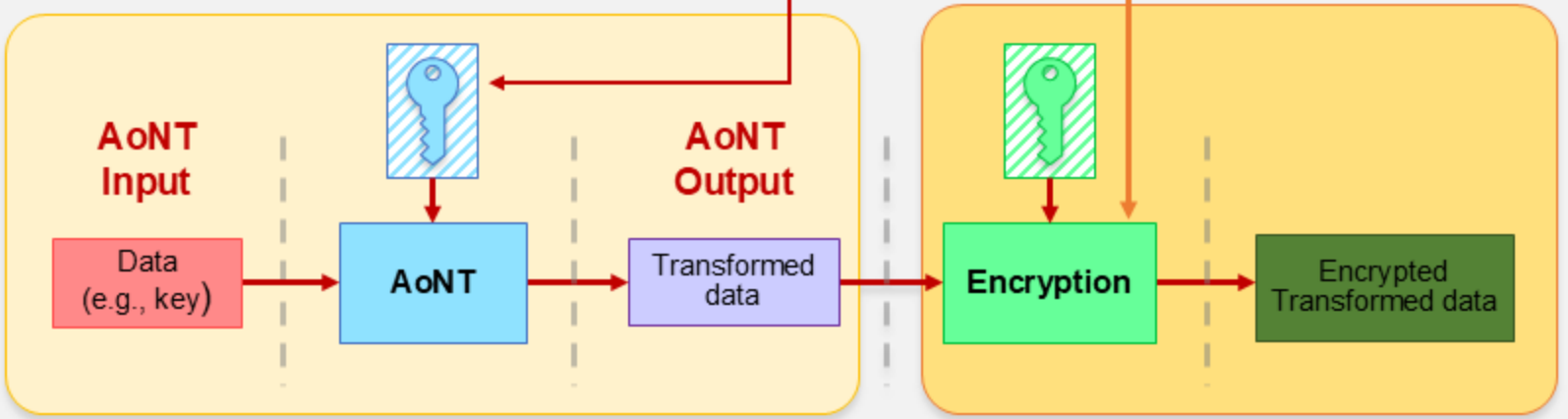
Rivest

We note that the all-or-nothing transformation is not itself "encryption," since it makes no use of any secret key information. It is merely an invertible "preprocessing" step that has certain interesting properties. The actual encryption in an all-or-nothing encryption mode is the operation that encrypts the pseudo-message resulting from the all-or-nothing transform. An all-or-nothing transform is a fixed public transform that anyone can perform on the message to obtain the pseudo-message, or invert given the pseudo-message to obtain the message.

We propose here a particular all-or-nothing transform, which we call the "package transform." We note that while it uses a block cipher itself as a primitive, no secret keys are used. (Instead, a randomly chosen key is used, and this key can be easily determined from the pseudo-message sequence.) The block cipher

searching for a key is infeasible. The scheme also uses a fixed publically-known key K_0 for the package transform block cipher.

'116 EX1034, 212-213 (cited Pet., 3-4)



DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Grounds 3-4: SFI alleges AoNT causeses “error-propagation” but Rivest teaches that ways to minimize error propagation were already known to POSAs

SFI’s POR

Petitioner additionally ignores the substantial drawback associated with using AoNT. As Dr. Rubin explains:

> Rivest also points out that “all-or-nothing encryption has ***terrible*** error-propagation properties” because if any pseudo-message block is damaged “then it is likely that ***every*** message block will be damaged.” EX1034, 216. Accordingly, redundancy information should be included in any data set before it is transformed using the AoNT, a practice that is suggested in the ’116 (*see* 77:23-36) but which would have to be added to Tajima.

’116 Paper 17 at 31

SFI’s Sur-Reply

EX1034, 216. Petitioner admits that in its proposed combination, “individual transformed message blocks are kept ***and transmitted*** together,” Reply, 20, the same action that Rivest discloses can lead to damage that would prevent inversion of the transformation. Thus Petitioner’s Tajima-Rivest would require additional, failure-proof error-correction techniques not disclosed in either Tajima nor Rivest.

’116 Paper 29 at 11

Dr. Bhattacharjee

C. Response to Argument That AoNT Has Terrible Error Propagation Properties

51. Dr Rubin argues that a POSA would not have applied AoNT because Rivest says “all-or-nothing encryption has terrible error propagation properties” because if any message block is damaged “then it is likely that every message block will be damaged” (Rivest, 216). EX2033, ¶88. I disagree that a POSA would understand Rivest to be discouraging the use his own transform. Rivest is simply pointing out the importance or using reliable transmission means to reduce data corruption and/or to use error-correction techniques. EX1034, 216. Second, in the Tajima-Rivest combination the transformed data—both the first implementation where the key information is transformed, and in the second implementation in which the data fragment is transformed using the key information (see First Declaration, ¶¶189-190)—the individual transformed message blocks are kept together, transmitted and stored together. Thus, there is no “error propagation” in Tajima-Rivest.

’116 EX1111, ¶51 (cited ’116 Reply, 19-20)

“Although the prior art that can be considered in inter partes reviews is limited to patents and printed publications, it does not follow that we ignore the skilled artisan’s knowledge when determining whether it would have been obvious to modify the prior art.

Koninklijke Philips N.V. v. Google LLC, 948 F.3d 1330, 1337 (Fed. Cir. 2020) (cited ’609 Reply, 24; ’854 Reply, 22) (quoting 35 U.S.C. §103) (pre-AIA)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Overview of the ’116 Petition’s Grounds

GROUND	CLAIMS	BASIS
1. Tajima (EX1105)	1, 5-8	102/103
2. Tajima, Foster (EX1106)	1-2, 9, 11-14	103
3. Tajima, Rivest (EX1034)	9-14	Obviousness
4. Tajima, Rivest, Shyu (EX1016)	**1-3*, 6**	**Obviousness**
5. Foster, Hayhurst (EX1007)	1-2, 5, 9, 11	103

*** Only Ground that covers dependent claim 3, but SFI raises no arguments specific to claim 3**

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 4: A POSA would have had motivation to use Shyu’s variable-encryption technique to add security to Tajima-Rivest

’116 Petition

VIII. GROUND 4: CLAIMS 1-3, 6 WOULD HAVE BEEN OBVIOUS OVER TAJIMA, RIVEST AND SHYU

A. Tajima-Rivest-Shyu Combination

POSAs would have recognized the benefit of applying different encryption algorithms to different data fragments as taught by Shyu (EX1016) to avoid known security risks of using a fixed encryption algorithm. Shyu, 1:24-42; 3:12-38, 3:56-60; Bhattacharjee, ¶¶201-204 (*citing also* EX1036, [0003]-[0009]). Moreover, it would have obvious to vary the encryption algorithm in Tajima’s system as doing so is merely the use of a known technique according to its intended purpose (increasing security) to yield the predictable result of making it more difficult for an attacker to decrypt different data fragments to achieve Tajima’s objective of securing a user’s personal information. *KSR*, 550 U.S. at 416-417; Bhattacharjee, ¶205.

In addition to including in Tajima’s key information metadata identifying what encryption algorithm the corresponding key is used with as taught by Shyu (Shyu, 3:12-26, 3:56-60), POSAs would have known to include additional metadata, such as the length of the key. Bhattacharjee, ¶¶129-133, 206-207 (citing Shyu, 3:12-26; EX1036, [0053]-[0054], [0077]). Tajima-Rivest-Shyu thus (1) encrypts Tajima’s data fragments using Shyu’s variable-encryption technique with Tajima’s different keys and stores the encryption type (and optionally additional metadata) with the corresponding key in Tajima’s key information (Shyu, 3:12-60), and (2) transforms the resulting key information using AoNT as in Tajima-Rivest (§VII.B *supra*), beneficially providing multiple protections for Tajima’s keys. Bhattacharjee, ¶¶206-207; §VI.A *supra*.

’116 Paper 2 at 48-49

Shyu

It is noted that the scrambling algorithms employed in conventional data encryption systems are fixed throughout the software or the medium used to convey or store the encrypted data. As such, the protection against software piracy is weak once the scrambling algorithm has been broken by hackers.

data segment. Preferably, the scrambling type and the key word of each of the segment headers (SH) are selected randomly. In addition, a pseudo key is recorded in the key

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.

’116 EX1016, 3:56-59 (cited Pet., 48-49)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

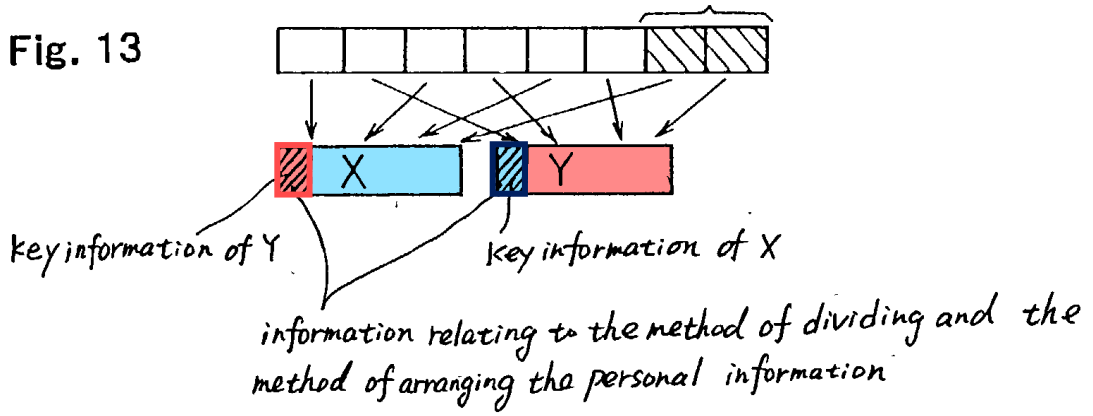
’116 Ground 4: A POSA would have had motivation to use Shyu’s variable-encryption technique to add security to Tajima-Rivest

’116 Petition

Shyu, 3:12-26; EX1036, [0053]-[0054], [0077]). Tajima-Rivest-Shyu thus (1) encrypts Tajima’s data fragments using Shyu’s variable-encryption technique with Tajima’s different keys and stores the encryption type (and optionally additional metadata) with the corresponding key in Tajima’s key information (Shyu, 3:12-60), and (2) transforms the resulting key information using AoNT as in Tajima-Rivest (§VII.B *supra*), beneficially providing multiple protections for Tajima’s keys. Bhattacharjee, ¶¶206-207; §VI.A *supra*.

’116 Paper 2 at 49

Tajima



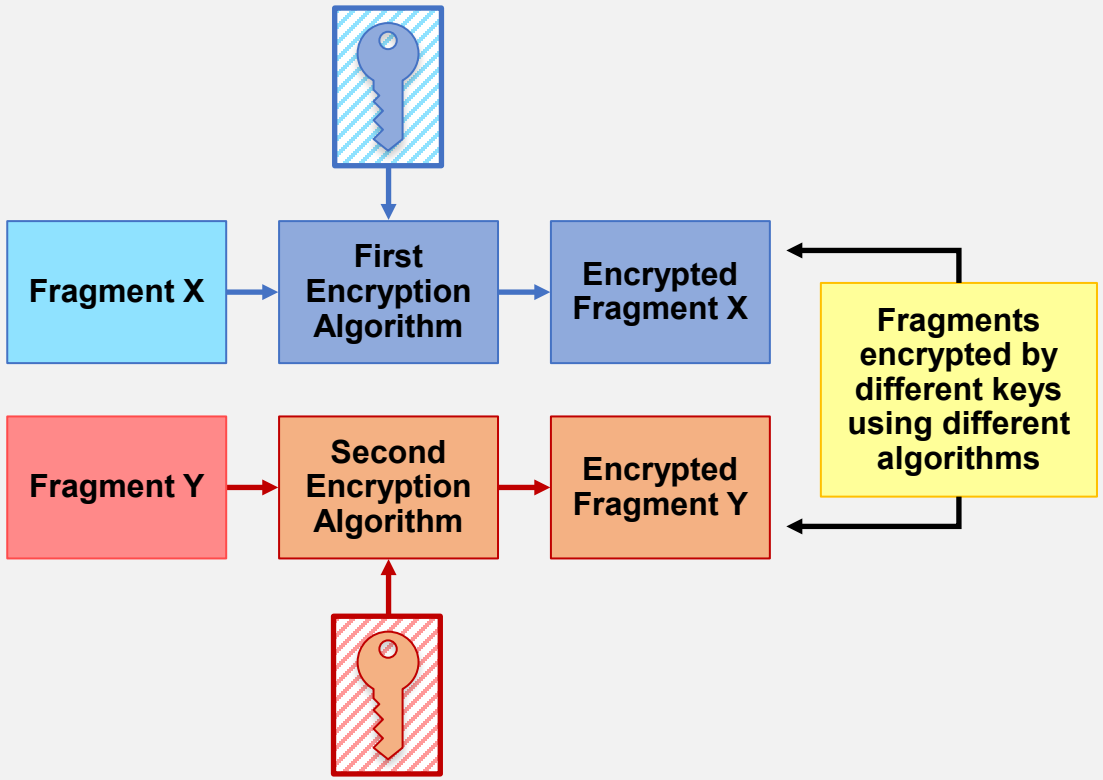
’116 EX1005, FIG. 13 (cited ’116 Pet., 19-21)

Shyu

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.

’116 EX1016, 3:56-59 (cited Pet., 48-49)

Tajima-Rivest-Shyu



DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 4: The Parties’ Disputes Are Narrow

- No disputes that Tajima-Rivest meets challenged claims 9-14

- **Only dispute concerns whether POSAs had motivation to add Shyu’s variable-encryption technique to Tajima-Rivest**

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 4: Even if SFI were correct that Shyu’s violates Kerckhoff’s Principle (Shyu does not) SFI’s own expert admits “Kerckhoff’s Principle” is just “a guideline”

SFI’s POR

First, ***no*** POSITA would be motivated believe security would be improved by applying of different encryption algorithms to different pieces of the same data set (per Shyu) because doing so would violate Kerckhoff’s Principle, a long-standing and fundamental tenet of cryptography. As Dr. Rubin explains,

> Kerckhoff’s Principle…is one of the foundational rules of cryptography…. [T]he rule states that “[t]he cipher method must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience.” EX2007, 3. In other words, it holds that holds that a cryptosystem should be secure, even if everything about the system, except the key, is public knowledge. *Id.* Stated differently, Kerckhoff’s principle demands that security rely solely on the secrecy of the key. Kerckhoff’s Principle has been widely
>
> * * *

Shyu’s anti-piracy system is a prototypical attempt to achieve security through obscurity, to the extent it proposes to conceal or obscure the encryption algorithms (Pet., 48-49) used on any data set or piece of a data set. Because it violates Kerckhoff’s principle, an POSITA would be discouraged from following the path it discloses, much less motivated to combine it with Tajima and Rivest.

’116 Paper 17 at 32-34

Shyu

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.

’116 EX1016, 3:56-59 (cited Pet., 48-49)

SFI’s Dr. Rubin

Q. Does the non-cryptographic step violate Kerckhoffs's Principle?

A. Kerckhoffs's Principle basically says that you should not put your security in anything other than the keys. And so obfuscating -- performing an obfuscation is a -- is not putting the protection in the keys, so I would say this does not on its own provide security --

Q. Okay.

A. -- based on Kerckhoffs.

Q. Does Kerckhoffs say that -- let me take us back. So Kerckhoffs's Principle says you should not put security in anything other than the keys, correct?

A. That's part of it. It also says your algorithms should be public, and the attacker should know exactly what you're doing, and you should still be secure if your keys are secure.

Q. Does Kerckhoffs's Principle say you cannot use security that isn't based on a key?

A. Kerckhoffs's Principle doesn't say you can or can’t do anything. It says that as a guideline, you should put your security in the keys and not in the hiding of your algorithms.

’116 EX1099, 155-156 (cited Reply, 20)

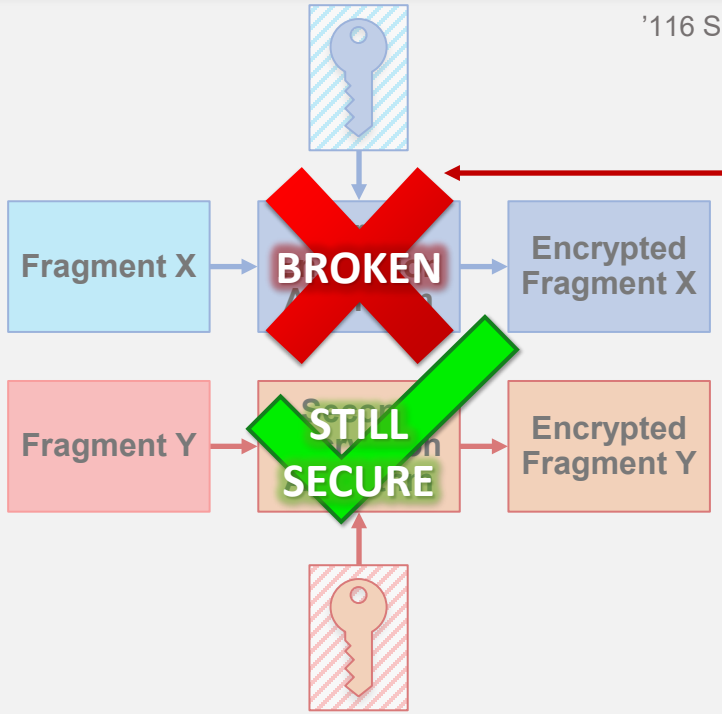
DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 4: SFI’s argument fails because the protection in Shyu comes from using different secure algorithms on different pieces of data, not the secrecy of which algorithm is used on which piece of data

SFI’s Sur-reply

Contrary to Petitioner’s argument, Shyu makes clear that whatever security it may provide ***is*** due to the secrecy of its algorithms. Shyu teaches that the use of a single algorithm may be inadequate to prevent software piracy because “the protection…is weak once the scrambling algorithm has been broken by hackers.” EX1016 [Shyu], 1:27-29. A system in which the algorithms must be “broken” by an attacker is a system that uses secret algorithms. Shyu’s purports to solve the

’116 Sur-reply, 21



Shyu

It is noted that the scrambling algorithms employed in conventional data encryption systems are fixed throughout the software or the medium used to convey or store the encrypted data. As such, the protection against software piracy is weak once the scrambling algorithm has been broken by hackers.

’116 EX1016, 1:24-29 (cited Pet., 48-49)

EX1036

[0008] According to the present invention, there is provided a ciphering comprising: dividing plaintext into blocks; setting a ciphering attribute for use in ciphering each of the blocks; ciphering each of the blocks in accordance with a ciphering attribute set for the block to obtain a ciphertext; and outputting the ciphertext and the ciphering attribute used for obtaining the ciphertext.

[0009] According to the present invention, the ciphering attribute of each part of the ciphertext can be changed. Therefore, even if part of the ciphertext can be deciphered, the rest cannot be deciphered, and a more reliable ciphering technique can be provided. Particularly, a user appropriately determines the ciphering attribute, and it is thereby possible to perform ciphering which meets the user’s needs.

’116 EX1046, [0008]-[0009] (cited Pet., 48)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 Ground 4: SFI’s Exhibit 2010 recognizes that using different encryption algorithms on different pieces of data provides security

EX2010

Multiple encryption is one combination technique: using an algorithm to encrypt the same plaintext block multiple times with multiple keys. Cascading is like multiple encryption, but uses different algorithms. There are other techniques.

Encrypting a plaintext block twice with the same key, whether with the same algorithm or a different one, is not smart. For the same algorithm, it does not affect the complexity of a brute-force search. (Remember, you assume a cryptanalyst knows the algorithm including the number of encryptions used.) For different algorithms, it may or may not. If you are going to use any of the techniques in this chapter, make sure the multiple keys are different and independent.

’116 EX2010, 357 (EX2010 cited ’116 POR, 6)

15.7 Cascading Multiple Block Algorithms

What about encrypting a message once with Algorithm A and key K_A, then again with Algorithm B and key K_B? Maybe Alice and Bob have different ideas about which algorithms are secure: Alice wants to use Algorithm A and Bob wants to use Algorithm B. This technique is sometimes called **cascading**, and can be extended far beyond only two algorithms and keys.

Pessimists have said that there is no guarantee that the two algorithms will work together to increase security. There may be subtle interactions between the two algorithms that actually *decrease* security. Even triple encryption with three different algorithms may not be as secure as you think. Cryptography is a black art; if you don’t know what you are doing, you can easily get into trouble.

Reality is much rosier. The previous warnings are true only if the different keys are related to each other. If all of the multiple keys are independent, then the resultant cascade is at least as difficult to break as the first algorithm in the cascade [1033]. If the second algorithm is vulnerable to a chosen-plaintext attack, then the first algorithm might facilitate that attack and make the second algorithm vulnerable to a known-plaintext attack when used in a cascade. This potential attack is not limited to encryption algorithms: If you let someone else specify any algorithm which is used on your message before encryption, then you had better be sure that your encryption will withstand a chosen-plaintext attack. (Note that the most common algorithm used for compressing and digitizing speech to modem speeds, used before any encryption, is CELP—designed by the NSA.)

This can be better phrased: Using a chosen-plaintext attack, a cascade of ciphers is at least as hard to break as any of its component ciphers [858]. A previous result showed that the cascade is at least as difficult to break as the strongest algorithm, but that result is based on some unstated assumptions [528]. Only if the algorithms commute, as they do in the case of cascaded stream ciphers (or block ciphers in OFB mode), is the cascade at least as strong as the strongest algorithm.

’116 EX2010, 367 (EX2010 cited ’116 POR, 6)

ADDITIONAL SLIDES

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

A Petition is not required to demonstrate a reference is analogous art; a Petitioner can respond to a POR's analogous art challenge in the Reply



"A petitioner is not required to anticipate and raise analogous art arguments in its petition; instead a petitioner can use its reply to 'respond to arguments raised in the corresponding opposition, patent owner preliminary response, patent owner response, or decision on institution.' *See* 37 C.F.R. § 42.23."

Sanofi-Aventis Deutschland GmbH v. Mylan Pharms. Inc., 66 F.4th 1373, 1379 (Fed. Cir. 2023) (cited '854 Reply, 2)



"[A] reference can be analogous art with respect to a patent even if there are significant differences between the two references. Indeed, there will frequently be significant differences between a patent and a reference from a different field of endeavor. But it does not follow that such a reference is, for that reason alone, not reasonably pertinent to one or more problems to which the claimed invention relates.… The relevant question is whether a PHOSITA would reasonably have consulted the reference in solving the relevant problem.… [I]f the two references have pertinent similarities such that [prior art reference] Mullen is reasonably pertinent to one or more of the problems to which the '023 patent pertains, then Mullen is analogous art.

Donner Tech., LLC v. Pro Stage Gear, LLC, 979 F.3d 1353, 1361 (Fed. Cir. 2020) (internal quotation marks and citations omitted) (cited '854 Sur-Reply, 4)

SFI's Expert Admits Tajima's FIG. 13 Shows "Key Information"

SFI's Dr. Rubin

Q. So let me ask again. In Figure 13 there's encrypted personal information and key information on the key that was used in encrypting the personal information, correct?

A. I'm sorry. Go ahead, Nate.

Q. And the key information of that first key is shown by the two hash marks in Figure 13, correct?

A. When you say the two boxes that are shaded?

Q. Yes.

A. That represents information about the key that was used to encrypt the personal information.

Q. And if you look to Figure 11, it has a very similar figure, and it calls those two boxes key information. Do you see that?

A. Yes.

Q. And Figure 13, it's just left out that label. Do you see that?

A. The label is not there.

Q. Right. But the key information is what those two hash marks are in Figure 13, correct?

A. I think that's what is explained in the text, and I believe that's the case.

'116 EX1099, 172-173 (objections omitted)
(responsive to '116 Sur-Reply, 14)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE



"First, Philips argues…§ 311(b) prohibits use of general knowledge to supply a missing claim limitation in an inter partes review. We disagree.… Here, Google properly alleged that although SMIL 1.0 did not disclose each and every element of the claimed invention, the differences between the claimed invention and SMIL 1.0 are such that the claimed invention would have been obvious to a person having ordinary skill in the art when considering SMIL 1.0. In particular, Google properly alleged that a skilled artisan would have known about pipelining and been motivated to combine pipelining with SMIL 1.0."

Koninklijke Philips N.V. v. Google LLC, 948 F.3d 1330, 1337-38 (Fed. Cir. 2020) (cited '609 Reply, 24; '854 Reply, 22)



"Because [prior art references] Oinonen and Whitley disclose the environmental devices and commands, it does not avail [] to complain that [reference] Menard does not. **The Board did not rely on either common sense or ordinary creativity to supply any missing environmental teaching, but rather pointed to the Oinonen and Whitley references**."

Koninklijke Philips N.V. v. Google LLC, 948 F.3d 1330, 1337-38 (Fed. Cir. 2020)



"As in *Arendi*, the limitation at issue here is not unusually simple, and the technology is not particularly straightforward. The '290 patent devotes the bulk of its written description to the complex communications protocol that enables the claimed 'low duty cycle" mode of operation. … [T]he question of whether a transmitter is "energized in low duty cycle RF bursts" is not an easy one. The missing limitation, moreover, plays a major role in the subject matter claimed… [A] stated object of the patent is for this server to have 'extremely low power consumption.' The '290 patent explains that the low duty cycle pulsed mode of operation is critical to achieving this goal."

DSS Tech. Mgmt., Inc. v. Apple Inc., 885 F.3d 1367, 1375 (Fed. Cir. 2018) (citations and quotation marks omitted) (cited '116 POR, 6; '609 POR, 18; '854 POR, 20)

’854 EX1035

Because the ciphering algorithm used is not secret, the degree of protection that can be derived from a cryptographic system ultimately depends upon the security of the cryptographic keys. Therefore, the objectives of key management are: (1) cryptographic keys should never occur in clear form outside the cryptographic device, except under secure conditions during the period when keys are originally distributed and installed or when stored in a secure place such as a safe, vault or similar location for backup or recovery and (2) no cryptographic operation, or combination thereof, using any cryptographic quantities which are routinely stored or routed through the system, or derived therefrom, should permit clear keys to be recoverable outside the cryptographic device. Therefore, in keeping with the first objective, if the system generated terminal keys are to be stored at the host system they must be protected by being enciphered under another key. One

’854 EX1035, 10:42-59 (cited ’854 Pet., 18; ’609 Pet., 40)
’116 EX1022 (cited ’116 Pet., 33)

’854 EX1036

Security 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. This approach means that the data encryption keys will be encrypted by a key encryption key. This in turn leads to the

’854 EX1036, 1:66-2:4 (cited ’854 Pet., 43; ’609 Pet., 40)
’116 EX1021 (cited ’116 Pet., 33-34)

’854 EX1087

[0020] Preferably, protecting the backup key includes encrypting the backup key, and retrieving the protected backup key includes decrypting the protected backup key. More preferably, the backup key is encrypted using an encryption password that is stored in a backup key storage device that is used to protect the backup key by storing the backup key therein; and the method of the present invention also includes providing an alleged encryption password to the backup key storage device. Examples of backup key storage devices include an internet server and a trusted computer that are used to store the backup key. The backup

’854 EX1037, [0020]
(cited ’854 Reply, 18; ’609 Reply, 20)

’854 EX1112

[0027] The session key may be stored in the same data store used to store the encrypted data, or in a different data store. Wherever the session key is stored, it is preferable that it is encrypted beforehand. The session key may be encrypted using the master key. Again, the advantage of

’854 EX1112, [0027]
(cited ’854 Reply, 18; ’609 Reply, 20)

'116 EX2010

Alice could send Bob the symmetric key over their communications channel—the one they are going to encrypt. This is foolish; if the channel warrants encryption, sending the encryption key in the clear over the same channel guarantees that anyone eavesdropping on the channel can decrypt all communications.

The X9.17 standard [55] specifies two types of keys: key-encryption keys and data keys. **Key-Encryption Keys** encrypt other keys for distribution. **Data Keys** encrypt message traffic. These key-encrypting keys have to be distributed manually (although they can be secured in a tamperproof device, like a smart card), but only seldomly. Data keys are distributed more often. More details are in [75]. This two-tiered key concept is used a lot in key distribution.

'116 EX2010, 176 (EX2010 cited '116 POR, 6)

Ideally, a key should never appear unencrypted outside the encryption device. This isn't always possible, but it is a worthy goal.

'116 EX2010, 181 (EX2010 cited '116 POR, 6)

enhance security in any way; it just gives a cryptanalyst more to work with. One solution might be to encrypt each file with a unique file key, and then encrypt all the file keys with a key-encryption key. The key-encryption key should then be

'116 EX2010, 184 (EX2010 cited '116 POR, 6)

'116 EX2008

Attack Prevention vs. Attack Detection

Most cryptographic systems rely on prevention as their sole means of defense: the cryptography keeps people from cheating, lying, abusing, or whatever. Defense should never be that narrow. A strong system also tries to detect abuse and to contain the effects of any attack. One of our fundamental design principles is that sooner or later, every system will be successfully attacked, probably in a completely unexpected way and with unexpected consequences. It is important to be able to detect such an attack, and then to contain the attack to ensure it does minimal damage.

'116 EX2008, 5 (EX2008 cited '116 POR, 6)

Building Secure Cryptographic Systems

Security designers occupy what Prussian general Carl von Clausewitz calls "the position of the interior." A good security product must defend against every possible attack, even attacks that haven't been invented yet. Attackers, on the other hand, only need to find one security flaw in order to defeat the system. And they can cheat. They can collude, conspire, and wait for technology to give them additional tools. They can attack the system in ways the system designer never thought of.

'116 EX2008, 5-6 (EX2008 cited '116 POR, 6)

'116 EX2029

3. *Importance for study:* Given two encryption schemes, how can we compare them? Without any definition of security, the only point of comparison is efficiency; but efficiency alone is a poor criterion since a highly efficient scheme that is completely insecure is of no use. Precise specification of the level of security achieved by a scheme offers another point of comparison. If two schemes are equally efficient but the first one satisfies a stronger definition of security than the second, then the first is preferable.[4] Alternately, there may be a trade-off between security and efficiency (see the previous two points), but at least with precise definitions we can understand what this trade-off entails.

'116 EX2029, 19 (EX2010 cited '116 POR, 6)

Like SFI's patents, Orsini relies on existing encryption standards that included KEKs and key-management systems for KEKs

Orsini

[0107] The cryptographic engine **220** also comprises a cryptographic handling module **625** configured to perform one, some or all of a wide number of cryptographic functions. According to one embodiment, the cryptographic handling module **625** may comprise software modules or programs, hardware, or both. According to another embodiment, the cryptographic handling module **625** may perform data comparisons, data parsing, data splitting, data separating, data hashing, data encryption or decryption, digital signature verification or creation, digital certificate generation, storage, or requests, cryptographic key generation, or the like. Moreover, a skilled artisan will recognize from the disclosure herein that the cryptographic handling module **825** may advantageously comprises a public-key infrastructure, such as Pretty Good Privacy (PGP), an RSA-based public-key system, or a wide number of alternative key management systems. In addition, the cryptographic handling module **625** may perform public-key encryption, symmetric-key encryption, or both. In addition to the foregoing, the cryptographic handling module **625** may include one or more computer programs or modules, hardware, or both, for implementing seamless, transparent, interoperability functions.

'609 EX1005, [0107] (cited '609 Pet. 22; '854 Pet. 20)
See also '116 patent, 19:13-34; '854 patent, 18:46--67; '609 patent, 18:46--67

EX1005

[0378] In this stream cipher RS1 algorithm, the required computations needed are nearly as simple and fast as the OTP algorithm. The benefit in this example using the RS1 Stream Cipher is that the system needs to store and transmit on average only about 16 bytes more than the size of the original data to be secured per share. When the size of the original data is more than 16 bytes, this RS1 algorithm is more efficient than the OTP algorithm because it is simply shorter. It is readily apparent to those of ordinary skill in the art that a wide variety of encryption methods or algorithms are suitable for use in the present invention, including, but not limited to RS1, OTP, RC4™, Triple DES and AES.

'609 EX1005, [0378] (cited '609 Pet. 44; '854 Pet. 46)
See also
'116 patent, 19:13-34;
'854 patent, 18:46--67;
'609 patent, 18:46--67

EX1094

to encrypt the encryption key. Standardized algorithms that can be used to implement the above described exemplary key management mechanisms can include the Diffie-Hellman, RSA, Triple-DES Key Wrap, RC2 Key Wrap, and RSAOEAP algorithms. Once the encryption key has been

EX1090

where wKi is the data encryption key **412** wrapped with the ith KeK **430**, Wkeki is a key wrapping function **432**, here the National Institute of Standards and Technology (NIST) Advanced Encryption Standard (AES) key wrapping function, applied to the DeK **412** using key the KeKi **430**, and i=1 to c. Similarly,

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Arctic Cat Inc. v. Polaris Indust. Inc.

"When several prior art references are combined a claimed invention is not obvious unless a person having ordinary skill in the art would have been motivated to combine the references. The important inquiry is whether an ordinarily skilled artisan is able to recognize, based on her background knowledge, a technique's potential to improve the device and be able to apply the technique. This court has previously held that a challenger need not prove that there was a known problem with the prior art in order to demonstrate that there was a motivation to combine prior art references.

Arctic Cat argues that the Board misapplied the law by requiring Arctic Cat to prove that there was a known problem in the prior art. However, the Board did not misapply the law here. Arctic Cat argued to the Board that there were problems with the Sunsdahl reference that could be improved through modifying Sunsdahl with Suzuki and Brown. Arctic Cat claimed that there was a blockage problem with the Sunsdahl air inlets and that Sunsdahl's air intake was open to the sky, creating a problem with rain and snow ingestion. The Board rejected these arguments because Arctic Cat failed to prove that these alleged problems with Sunsdahl existed such that a skilled artisan would been motivated to combine Sunsdahl with the other prior art to overcome those alleged problems. In this case, Arctic Cat tried to create a problem with the prior art in hopes of creating a motivation to combine references, but it failed. Although a challenger to a patent is not required to prove there was a known problem with the prior art the Board did not misapply the law by requiring Arctic Cat to prove the facts that it alleged. Arctic Cat's case was doomed when it failed to prove the premise it offered for combining Sunsdahl with Suzuki.

Arctic Cat also argues that the Board improperly required it to explain how drawbacks of that combination would be overcome. The relevant inquiry is whether or not a person having ordinary skill in the art would be motivated to combine references despite drawbacks that the combination might present. The Board must weigh the benefits and drawbacks of the modification against each other, to determine whether there would be a motivation to combine.

The Board stated that Arctic Cat failed to explain how the drawbacks of the modifications would be overcome, rather than weighing the benefits and drawbacks of the modifications against each other. However, the Board also made factual findings regarding the benefits and drawbacks of modifying Sunsdahl with Suzuki. The Board found that the air inlets of Sunsdahl were not open to the sky as Arctic Cat claimed, and therefore moving the inlets to the side did not provide Arctic Cat's alleged benefits of preventing inlet blockage or rain and snow ingestion. The Board further found Arctic Cat's expert evidence supporting these claimed benefits of moving the air inlets to be unpersuasive.

Additionally, the Board found that the modification of Sunsdahl with Suzuki would be unlikely to reduce noise and vibration due to the longer ducts that would be required to place the inlet on the side of the vehicle."

Arctic Cat Inc. v. Polaris Indust. Inc., 795 Fed. Appx. 827, 832-33 (Fed. Cir. 2019 (cleaned up) (cited '116 Sur-Reply, 20)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Dr. Bhattacharjee

Q. So in paragraph 51 do you see where it says, "As indicated above, the DRMS fails to provide synchronized" -- sorry – "provide synchronized delivery of the content with the keys used to encrypt the content." Do you see that?

A. I do see that.

Q. In that sentence, would it be fair to say "the DRMS" refers to the DRMS depicted in Figure 1?

A. I think if it's showing -- it is talking about a way the DRMS can fail to synchronize delivery of content. But the goal of Foster is to take systems such as this and enable the synchronization along with other things that Foster teaches. And if we take a look at, for instance, the end of paragraph 48, Foster says that "previous systems failed to provide such synchronized transmission. Moreover, by synchronously transmitting the title key with the encrypted content, the present invention maintains compatibility and compliance with current front-end and back-ends standards." And so the systems that are being shown in Figure 1 and Figure 2 are compatible with Foster, but Foster shows how in some wats they can be changed.

Q. What is a front-end standard?

A. It depends on the context. But the front-end standard can refer to the processing that is done at the recipient as defined by Foster.

Q. And what is a back-end standard?

A. The back-end standard, again, depends on context. But the back-end standard would be, for instance, as an example, the systems and techniques that are used, for instance, to store, encrypt, and transmit through a front end.

Q. When you say "front end," what are you referring to?

A. Yeah, one system's front end could be another system's back end and so on. So we have to take a look at the context in which these words are being used. But in general, a front end can be a recipient in Foster, and the back end, correspondingly, depending on perspective, would be the set of systems that are used to transmit data to the front end.

Q. Is DRMS a standard?

A. "DRMS" is a broad term. There are standards for DRMS. For instance, again, going back to DVDs, there is a DRMS standard called CAS that encrypts contents of DVDs. That's one standard. But by itself, "DRMS" is a broad term that just refers to techniques for digital rights management.

Q. Does Figure 1 depict a DRMS that fails to provide synchronized delivery of the content with the keys used to encrypt the content?

A. Figure 1 is a schematic of a system that implements DRMS. And that system may or may not do synchronized delivery of keys with content. Foster describes how such a system could be used to synchronize delivery, but it doesn't have to always synchronize delivery of keys with content in such a system at that level of abstraction.

EX2046, 39-42 (EX2046 cited '116 Sur-reply, 2; '609 Sur-reply, 24; '854 Sur-reply, 2)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’609 Patent

In one embodiment of the present invention, encryption keys, splitting keys, or both as used in accordance with the present invention may be further encrypted using, for example, a workgroup key in order to provide an extra level of security to a secured data set.

’609 EX1001, 77:11-15 (cited ’854 POR, 20; ’609 POR, 19)

The workgroup key concept allows for enhanced protection of information assets by encrypting key information stored within the shares. Once this operation is performed, even if all required shares and external keys are discovered, an attacker has no hope of recreating the information without access to the workgroup key.

’609 EX1001, 79:60-65 (cited ’854 POR, 20; ’609 POR, 19)

Orsini

[0389] Key Management

* * *

[0391] 1. The Parser Master Key

[0392] This key is an individual key associated with the installation of the data parser. It is installed on the server on which the parser has been deployed. There are a variety of options suitable for securing this key including, but not limited to, a smart card, separate hardware key store, standard key stores, custom key stores or within a secured database table, for example.

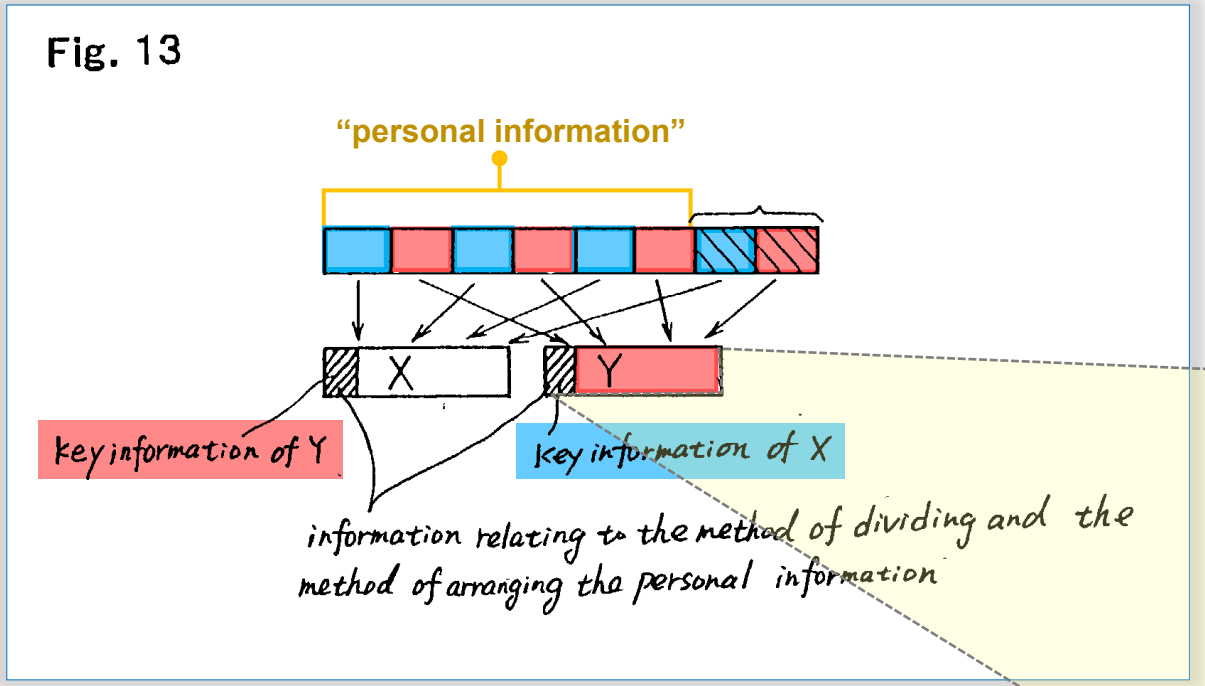
[0393] 2. The Session Master Key

[0394] A Session Master Key may be generated each time data is secured. The Session Master Key is used to encrypt the data prior to the parsing operation. It may also be incorporated (if the Session Master Key is not integrated into the parsed data) as a means of parsing the encrypted data. The Session Master Key may be secured in a variety of manners, including, but not limited to, a standard key store, custom key store, separate database table, or secured within the encrypted shares, for example.

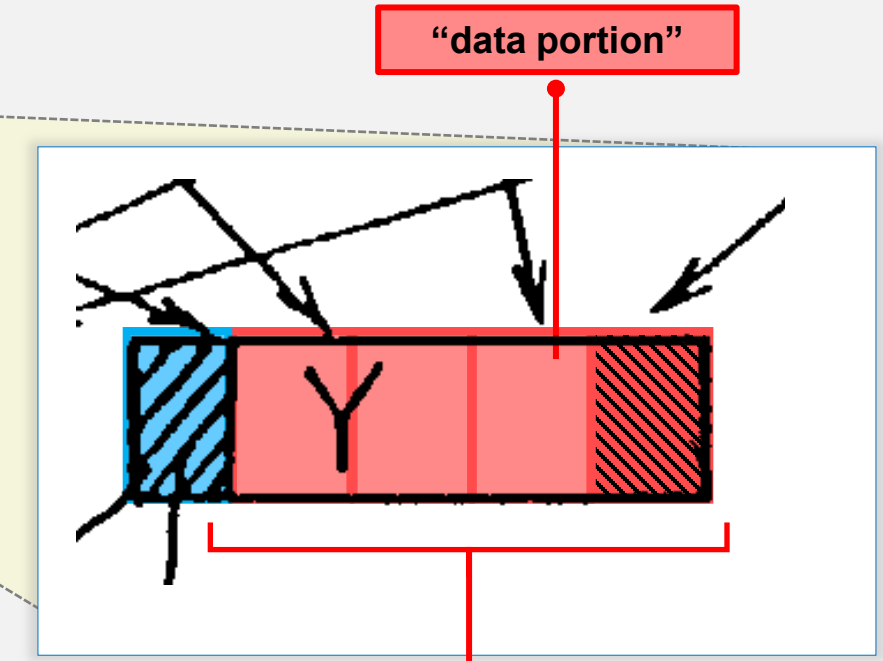
’609 EX1005, [0389]-[0394] (cited ’609 Pet., 18-19); *see* Pet., 9 (demonstrating same disclosure in ’609 patent) (citing EX1010)

’116 Ground 1: Tajima’s personal information has “data portions” that are distributed into two “data fragments” (the claimed “data chunks”)

Tajima



’116 EX1005, FIG. 13 (annotated)
(cited ’116 Pet., 16-22)



DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Reply

User-Independent Security. SFI only addresses the '854 patent's alleged problem of providing a "cryptographic system whose security is user independent." POR, 38-39 (quoting EX2033, ¶106). Hayhurst is reasonably pertinent to that problem and addresses it by segmenting and distributing data without user action. Hayhurst, [0027] (cited Petition, 61, 65). As the Board correctly observed, users cannot undermine Hayhurst's security because "no one subscriber unit has more than one media segment, and a single segment has little value." DI, 22; Hayhurst, [0021], [0027]-[0029]. Bhattacharjee-Reply, ¶¶27-36. Rubin-Depo., 115-120.

Computer-Network Security. Hayhurst is also reasonably pertinent to other problems the patent discusses. *In re Asset Guard Prods., Inc.*, 2022 WL 1097363, *4 (Fed. Cir. Apr. 13, 2022) (prior art need only be reasonably pertinent to ***a*** problem the patent relates to, not its "central" problem). The patent identifies the problem of "ensuring [that] data stored and transmitted over" "computer networks" "cannot be read or otherwise compromised." EX1001, 1:30-39; Rubin-Depo., 109-110. Hayhurst similarly addresses problems "relat[ing] to…**electronic** media **delivery**…including the **secure handling and storage** of" data ([0002]) and "deterr[ing] [] piracy" ([0008]). Bhattacharjee-Reply, ¶¶37-39.

Distributed Storage. The patent splits and "stor[es]…data in geographically remote secure data storage facilities." EX1001, 5:53-56. It does that to address problems with "protecting…against compromise of any individual data storage facility" (5:53-56) and data "remain[ing] whole and intact and subject to attack" (56:47-50). POR, 68-69 (admitting the patent has these "concern[s]," and addresses them by "splitting the data set into shares"). Hayhurst's distributed-storage technique addresses those problems and solves them in the same way. *Supra* §I.A.1. Rubin-Depo., 124-135; Bhattacharjee-Reply, ¶¶40-47.

'854 Reply, 5-6;
'116 Reply, 23-24

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

’116 POR

> Hayhurst’s system is not compatible with real-time streaming or random-access viewing of live television. Hayhurst requires that an entire movie file be segmented, and the segments transmitted to, and stored among, the members of a subscriber network before the movie can even be requested for viewing by any individual subscriber. Beyond that, because the segments are not stored in or received in any particular order, the movie cannot be viewed at all by the individual subscriber until (i) ***all*** segments are received from the subscriber network by the subscriber, (ii) the requesting unit obtains authorization, and (iii) the packets are descrambled and then reassembled into the complete movie file. EX1007, [0029]; [0030]; EX2034, 101:19-25. Thus, Hayhurst’s system requires all segments to be received and for the entire file to be reassembled the system can play the movie.

EX2033 ¶125-26. As others in the field have recognized, in “download and play,” systems such as Hayhurst “the client has to download and store first ***the whole media content*** on his PC and only then he/she will be able to view and use it.” EX2035, 219. This requirement “***exclud[es] the use [of such systems] for covering live events***, as the multimedia information that is to be transmitted cannot be ‘a-priori’ available.” *Id.*

’116 Paper 17 at 54-55

’116 POR

3. Petitioner’s Combination Involves Storing and Distributing All of Foster’s Content Units per Hayhurst’s Method.

Petitioner posits a combination of Foster and Hayhurst in which Foster’s media ***content units*** are transmitted and then stored in Hayhurst’s decentralized media ***file*** storage system. Specifically, “Foster’s encrypted content units” would be stored “on multiple storage devices...based on Hayhurst’s techniques” “for storing [and distributing] encrypted data segments” of a media file. Pet., 67, 63. In

’116 Paper 17 at 55

’116 POR

(a) *Adding Hayhurst’s Delivery Method to Foster’s System Exacerbates the Underlying Problem Of Delay Foster Seeks to Solve.*

As explained above, in Foster’s system, because each packet in an elementary content stream can be generated as the content is created, and then sent directly to the recipient with its title encryption key, each packet can be viewed without delay, allowing for immediate, random access to the content. *See supra* Sections III.A.1, III.B.1.

’116 Paper 17 at 58

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE



"Patent Owner asserts that Petitioner's Reply brief "presents new evidence and arguments beyond the scope of a proper reply" and "[t]he Board should refuse to consider" it. Sur-reply 9–10. This generic objection to the entire Reply fails to provide adequate notice regarding which of Petitioner's theories and arguments Patent Owner contends are allegedly new and improper. See, e.g., Rembrandt Diagnostics, LP v. Alere, Inc., 76 F.4th 1376, 1383 (Fed. Cir. 2023) (rejecting generic objection to alleged new theories in reply as failing to provide adequate notice of which theories are allegedly new). Accordingly, we reject Patent Owner's request that we "refuse to consider" the entirety of the Reply. Where Patent Owner's Sur-reply raises more particularized complaints regarding Petitioner's allegedly improper new arguments, we address those complaints as appropriate below.

Life Spine, Inc. v. Globus Med., Inc., IPR2022-01599, Paper 39 at 24 n.10 (FWD) (PTAB)

"Mr. Hatch's testimony in paragraphs 53–55 is persuasive to support the argument for which Petitioner cites them and the mere fact that Mr. Hatch provides additional detail to support that argument does not amount to an improper incorporation by reference. Indeed, expert testimony is generally more persuasive when it does not merely parrot the arguments in the briefs, but provides additional detail supporting the argument for which it is cited, as it does here."

Life Spine, Inc. v. Globus Med., Inc., IPR2022-01599, Paper 49 at 45 (FWD) (PTAB)

"Patent Owner asserts that Petitioner's Reply "exceeds the proper scope of reply by explaining its protrusion theory for the first time via lengthy new testimony (Ex. 1072 ¶¶ 122–131), despite Mr. Drewry's original declaration never referencing the issue." Sur-reply 27 (citing Ex. 1002 (Drewry Decl.) ¶¶ 461–66). We disagree. We find that Petitioner's discussion of the protrusion problem is both a permissible expansion on its arguments made in the Petition, and response to Patent Owner. See Corephotonics, Ltd. v. Apple Inc., 84 F.4th 990, 1008 (Fed. Cir. 2023) ("'there is no blanket prohibition against the introduction of new evidence during an IPR,' provided it is an 'expan[sion] on and . . . fair extension of . . . [a] previously raised . . . argument' and has a 'nexus' (and is therefore responsive) to an argument made by the patent owner or the Board") (quoting Rembrandt Diagnostics, 76 F.4th at 1384–85) (alterations in original). The Petition identifies a concern with the actuator "stick[ing] out the far end of the central ramp," and Mr. Drewry's original declaration stated that the combination would "allow a surgeon to use a shorter actuator." Pet. 109; Ex. 1002 (Drewry Decl.) ¶ 462. As we discuss above, in attempting to rebut Petitioner's argument, Patent Owner misdescribed Petitioner's proposed combination. Under the circumstances here, we find that Petitioner's Reply and Mr. Drewry's Reply declaration appropriately elaborate on the Petition and respond to Patent Owner's counterarguments related thereto.

Life Spine, Inc. v. Globus Med., Inc., IPR2022-01599, Paper 39 at 95 n.20 (FWD) (PTAB)



"The facts in Gurley's record are that this use of epoxy was known, the structure of these circuit boards was known, and epoxy had been used for Gurley's purpose. We share Gurley's view that a person seeking to improve the art of flexible circuit boards, on learning from Yamaguchi that epoxy was inferior to polyester-imide resins, might well be led to search beyond epoxy for improved products. However, Yamaguchi also teaches that epoxy is usable and has been used for Gurley's purpose. The Board recognized Yamaguchi's teaching of the deficiencies of epoxy-impregnated material, but observed that Gurley did not distinguish his epoxy product from the product described by Yamaguchi. On the facts of this case, Gurley's "teaching away" argument was insufficient to establish patentability. Gurley did not offer specific epoxies, or improved properties, and we are not presented with the question of whether any such products might meet the requirements of patentability. Even reading Yamaguchi's description as discouraging use of epoxy for this purpose, Gurley asserted no discovery beyond what was known to the art."

In re Gurley, 27 F.3d 551, 554 (Fed. Cir. 1994) (responsive to, e.g., '116 Sur-Reply, 14; '609 Sur-Reply, 26-30; '854 Sur-Reply, 19-26)

"Appellants argue that "the prior art disclosed alternatives to each of the claimed elements A [the perimeter], B [the shape of the surface], and C [the orientation of the surface]. Choosing one alternative necessarily means rejecting the other, i.e., following a path that is 'in a divergent direction from the path taken by the applicant.' " This interpretation of our case law fails. The prior art's mere disclosure of more than one alternative does not constitute a teaching away from any of these alternatives because such disclosure does not criticize, discredit, or otherwise discourage the solution claimed in the '198 application. Indeed, in the case cited by appellants, In re Gurley, we held that the invention claimed in the patent application was unpatentable based primarily on a prior art reference that disclosed two alternatives, one of which was the claimed alternative. Accordingly, mere disclosure of alternative designs does not teach away."

In re Fulton, 391 F.3d 1195, 1201 (Fed. Cir. 2004)
(responsive to, e.g., '116 Sur-Reply, 14; '609 Sur-Reply, 16-21, 26-30; '854 Sur-Reply, 7-9)

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

SFI's Dr. Rubin

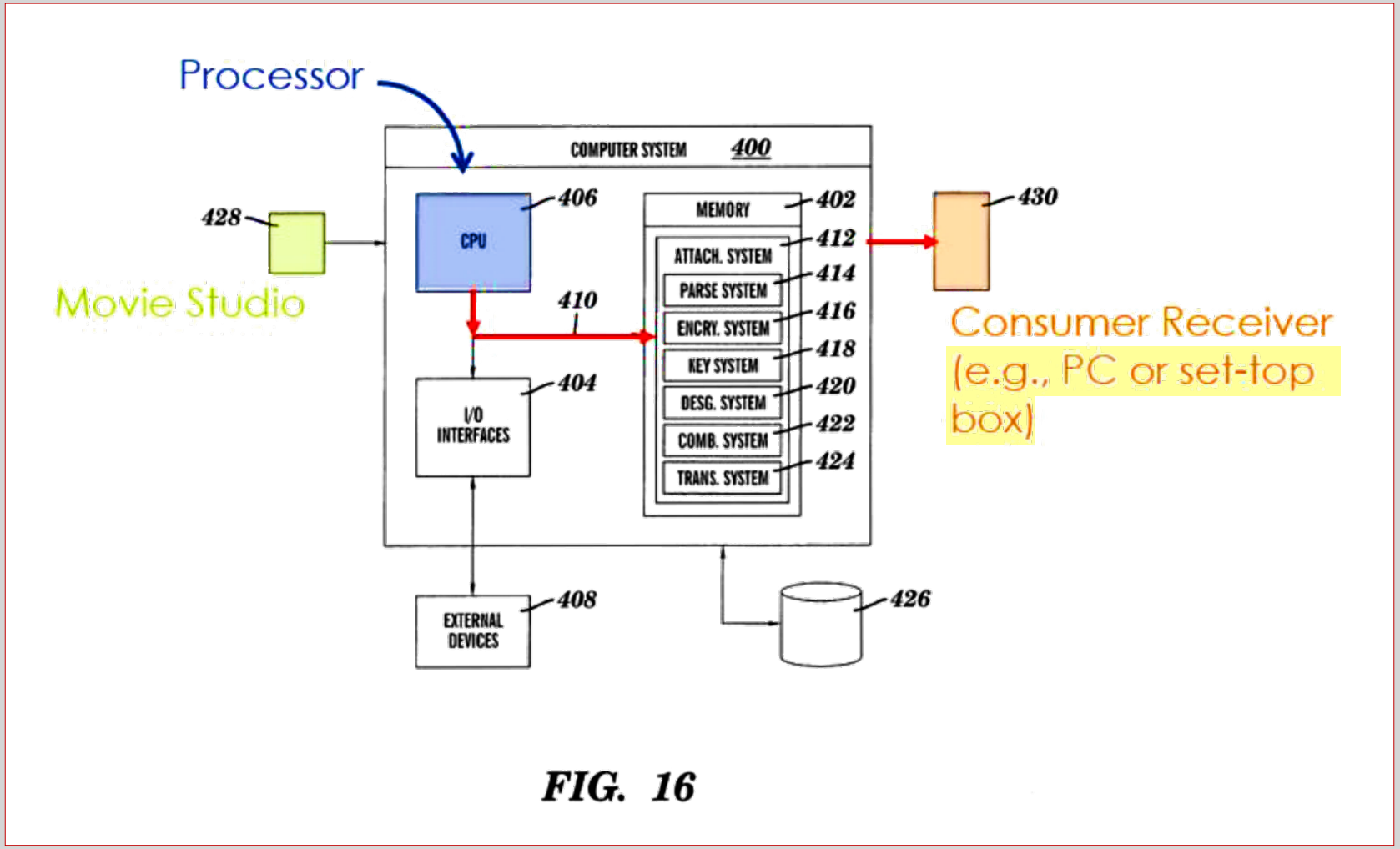


FIG. 16

'854 EX2003, ¶83 (cited DI at 17) (annotated); *see also* '854 EX2044 at 17;
'609 EX2003, ¶83 (cited DI at 17) (annotated); *see also* '609 EX2044 at 17

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE