

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. IPR2024-00215  
Patent No. 10,452,854

---

**REPLY DECLARATION OF SAMRAT BHATTACHARJEE, PH.D.**

|                                      |
|--------------------------------------|
| Google Exhibit 1111<br>Google v. SFI |
|--------------------------------------|

## TABLE OF CONTENTS

|      |                                                                                                                                                                        |    |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| I.   | Materials Reviewed And Considered .....                                                                                                                                | 1  |
| II.  | My Reply to SFI Regarding Ground 2: IT IS STILL MY OPINION<br>THAT ALL CHALLENGED Claims WOULD HAVE BEEN<br>OBVIOUS TO POSAs over THE Foster-Hayhurst COMBINATION..... | 3  |
|      | A. Response to Argument that Hayhurst (EX1007) is Not Analogous<br>Art to the '854 Patent .....                                                                        | 5  |
|      | 1. Response to Argument that POSAs would find Hayhurst to be<br>in the same field of endeavor as the '854 patent. ....                                                 | 7  |
|      | 2. Response to Argument that Hayhurst is not reasonably<br>pertinent to at least one of the problems to which the '854<br>patent relates. ....                         | 15 |
|      | a. Hayhurst is reasonably pertinent to the problem of user-<br>independent security. ....                                                                              | 15 |
|      | b. Hayhurst is reasonably pertinent to the problem of<br>computer-network security. ....                                                                               | 20 |
|      | c. Hayhurst is reasonably pertinent to problems addressed by<br>distributed-storage (e.g., the problem of data remaining<br>whole and subject to attack).....          | 21 |
|      | B. Response to Argument that POSAs Would Not Have Had<br>Motivation For the Foster-Hayhurst Combination .....                                                          | 25 |
|      | 1. It is clear to POSAs from Foster's disclosure that Foster stores<br>encrypted content.....                                                                          | 31 |
|      | 2. The Foster-Hayhurst combination discussed in my First<br>Declaration is consistent with Foster's goals. ....                                                        | 51 |
|      | C. Response to Argument that Foster's Encrypted Payloads Are Not<br>" <i>Data Chunks</i> " .....                                                                       | 74 |
| III. | My Reply to SFI Regarding Ground 1: IT IS STILL MY OPINION<br>THAT ALL CHALLENGED Claims WOULD HAVE BEEN<br>OBVIOUS over Orsini .....                                  | 85 |
|      | A. Response to Argument that It Would Not Have Been Obvious to<br>POSAs to Encrypt Orsini's Share Keys Using a Known Key-<br>Encrypting Key ("KEK").....               | 88 |

|                                                                                                                                                                                                                    |     |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 1. My KEK-based Obviousness Opinion was Independent of, and based on independent corroborating evidence from, my opinion that Orsini’s public-key encryption disclosures provided an anticipating embodiment. .... | 107 |
| 2. My KEK-based Obviousness Opinions Did Not Rely on Common Sense to Supply a Missing Limitation in the Prior Art. ....                                                                                            | 109 |
| a. POSAs would not have agreed that limitation [1C] was missing in the prior art. ....                                                                                                                             | 109 |
| b. My opinions on the obviousness of encrypting Orsini’s share keys with a KEK was not based on common sense. ....                                                                                                 | 112 |
| 3. My reply to SFI’s Challenges to my opinions regarding POSAs’ Motivation-to-Combine Orsini with a conventional KEK ....                                                                                          | 113 |
| a. It is still my opinion that POSAs Would Have Wanted to Make Orsini’s System More Secure by using a KEK. ....                                                                                                    | 114 |
| b. In my opinion, any alleged Drawbacks to the modifying Orsini to use a KEK that SFI Identifies would have been Easily Overcome ....                                                                              | 120 |
| c. My opinion KEK-based Obviousness Opinion was not based on hindsight, but was instead grounded in evidence that I cited.....                                                                                     | 125 |
| B. Response to Argument that POSAs would not have a reasonable expectation of success in modifying Orsini to meet limitation [1C].....                                                                             | 127 |

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

1. I have been retained by Wolf, Greenfield & Sacks, P.C., counsel for Petitioner Google LLC, to assess claims 1-10 (the “challenged claims”) of U.S. Patent No. 10,452,854 (“the ’854 patent”) (EX1001). I am being compensated for my time at my standard rate of \$700.00 per hour, plus actual expenses. My compensation is not dependent in any way upon the outcome of the *inter partes* review of the ’854 patent.

## **I. MATERIALS REVIEWED AND CONSIDERED**

2. My findings, as explained below, are based on my years of education, research, experience, and background in field of securing data from unauthorized access or use, as well as my investigation and study of relevant materials for this declaration. *See also* EX1070 (Supplemental Declaration of Samrat Bhattacharjee, Ph.D.). When developing the opinions set forth in this declaration, I assumed the perspective of a person having ordinary skill in the art (“POSA”), as set forth in Section IV of my initial declaration submitted in this proceeding (EX1003, “First Declaration”).

3. In forming my opinions set forth in this declaration, in addition to the references I discussed and reviewed for my First Declaration, I have reviewed SFI’s POR (Paper 21), Dr. Rubin’s declaration (EX2033), the exhibits cited by SFI and Dr. Rubin, and the following exhibits and papers as well as any papers or

exhibits explicitly cited herein that I cite in response to SFI's POR and/or Dr.

Rubin's declaration:

| <b>Exhibit</b> | <b>Description</b>                                                                                                                                                                                                          |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1072           | U.S. Patent Application Publication No. 2004/0019900                                                                                                                                                                        |
| 1073           | U.S. Patent Application Publication No. 2002/0083438                                                                                                                                                                        |
| 1074           | U.S. Patent No. 7,076,661                                                                                                                                                                                                   |
| 1075           | U.S. Patent No. 6,771,657                                                                                                                                                                                                   |
| 1076           | U.S. Patent No. 6,389,473                                                                                                                                                                                                   |
| 1077           | U.S. Patent No. 6,662,329                                                                                                                                                                                                   |
| 1078           | International Application Publication No. WO 2003/026220                                                                                                                                                                    |
| 1079           | Trappe, Key Management and Distribution for Secure Multimedia Multicast (IEEE Dec. 2003)                                                                                                                                    |
| 1080           | D. McGovern, "Pushing the internet direct to the user - security issues," IEE Seminar on IP Over Satellite - The next Generation: MPLS, VPN and DRM Delivered Services, 2003., 2003, pp. 122-138, doi: 10.1049/ic:20030063. |
| 1081           | U.S. Patent No. 7,277,941                                                                                                                                                                                                   |
| 1082           | Mike Knee, MPEG Video                                                                                                                                                                                                       |
| 1083           | U.S. Patent No. 7,266,613                                                                                                                                                                                                   |
| 1084           | U.S. Patent Application Publication No. 2004/0202320                                                                                                                                                                        |
| 1085           | U.S. Patent No. 7,823,009                                                                                                                                                                                                   |
| 1086           | U.S. Patent No. 8,098,819                                                                                                                                                                                                   |
| 1087           | U.S. Patent Application Publication No. 2004/0236958                                                                                                                                                                        |
| 1088           | International Application Publication No. WO 2001/015162                                                                                                                                                                    |
| 1089           | U.S. Patent Application Publication No. 2004/0107356                                                                                                                                                                        |
| 1090           | U.S. Patent No. 7,739,501                                                                                                                                                                                                   |
| 1091           | U.S. Patent Application Publication No. 2004/0162871                                                                                                                                                                        |
| 1092           | U.S. Patent Application Publication No. 2002/0129383                                                                                                                                                                        |
| 1093           | U.S. Patent Application Publication No. 2004/0005061                                                                                                                                                                        |
| 1094           | U.S. Patent Application Publication No. 2005/0160292                                                                                                                                                                        |
| 1095           | U.S. Patent No. 6,959,086                                                                                                                                                                                                   |
| 1096           | RFC 3394: Advanced Encryption Standard (AES) key wrap algorithm                                                                                                                                                             |
| 1097           | U.S. Patent No. 7,257,227                                                                                                                                                                                                   |
| 1098           | U.S. Patent No. 6,993,508                                                                                                                                                                                                   |
| 1099           | Deposition of Aviel D. Rubin, Ph.D. (Dec. 12, 2024)                                                                                                                                                                         |
| 1100           | U.S. Patent Application Publication No. 2003/0140257                                                                                                                                                                        |

|      |                                                                                                                                                   |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| 1101 | U.S. Patent Application Publication No. 2004/0199771                                                                                              |
| 1102 | U.S. Patent Application Publication No. 2005/0177740                                                                                              |
| 1103 | U.S. Patent Application Publication No. 2004/0199667                                                                                              |
| 1104 | U.S. Patent Application Publication No. 2005/0050218                                                                                              |
| 1105 | U.S. Patent Application Publication No. 2003/0187799                                                                                              |
| 1106 | National Institute of Standards and Technology, NIST Special Publication 800-57, Recommendations for Key Management – Part 1: General (Aug. 2005) |
| 1107 | U.S. Patent Application Publication No. 2004/0236958                                                                                              |
| 1108 | U.S. Patent No. 6,169,843                                                                                                                         |
| 1109 | MPEG-2 - a tutorial introduction to the systems layer (IEEE 1995)                                                                                 |
| 1110 | U.S. Patent Application Publication No. 2003/0145329                                                                                              |
| 1111 | Reply Declaration of Samrat Bhattacharjee, PH.D.                                                                                                  |
| 1112 | U.S. Patent Application Publication No. 2004/0010699                                                                                              |
| 1113 | European Patent Application Publication No. EP1,564,686                                                                                           |
| 1114 | U.S. Patent Application Publication No. 2004/0010699                                                                                              |
| 1115 | U.S. Patent Application Publication No. 2004/0187014                                                                                              |
| 1116 | RESERVED                                                                                                                                          |
| 1117 | U.S. Patent Application Publication No. 2006/0143350                                                                                              |

## **II. MY REPLY TO SFI REGARDING GROUND 2: IT IS STILL MY OPINION THAT ALL CHALLENGED CLAIMS WOULD HAVE BEEN OBVIOUS TO POSAS OVER THE FOSTER-HAYHURST COMBINATION<sup>1</sup>**

4. As I discussed in my First Declaration (¶¶166-247), Ground 2 combines Foster (EX1006) with Hayhurst (EX1007). In particular, the Foster-Hayhurst combination combines Foster's encryption methods with Hayhurst's distributed-storage technique. First Declaration, ¶¶182-190. As I discussed in my

---

<sup>1</sup> I have added all emphases added throughout this declaration unless I have noted otherwise.

First Declaration (¶¶167-177), Foster divides a *dataset* (e.g., a movie file) into segments (which Foster calls “payloads”), encrypts each segment with a respective key, encrypts those keys with the same key-encryption key (KEK), and stores all encrypted segments plus corresponding encryption keys on a storage device.

Foster, [0064]-[0072], FIG. 8. As I also discussed in First Declaration (¶¶178-181), Hayhurst, like Foster, also “segments” a “media file (movie),” encrypts each segment, and then separately distributes those encrypted segments to different storage devices for secure storage (and other benefits that Hayhurst identifies and that POSAs would have understood from Hayhurst’s disclosure). Hayhurst, [0006]-[0007].

5. As I discussed in my First Declaration (¶¶182-190), POSAs would have recognized that utilizing Hayhurst’s distributed-storage technique in Foster would have provided several benefits to Foster’s system, such as (1) reducing bandwidth requirements (and increasing the speed of delivery) for a content service provider delivering content to customers, (2) adding redundancy and backup to the storage and distribution of data, and (3) increasing the “security and a deterrence to piracy” (Hayhurst, [0008]).

6. My opinions in that regard were corroborated not only by Hayhurst’s explicit disclosure but also by prior art I cited that that demonstrated POSAs background knowledge about the benefits of using a distributed storage technique,

both in terms of storing and delivering content. First Declaration, ¶¶182-190 (citing EX1062, EX1037, EX1043, EX1050). As I discuss below (§§II.A-II.C), in response to SFI's and Dr. Rubin's arguments otherwise, my opinion regarding POSAs' motivation to pursue Foster-Hayhurst and their reasonable expectation of success in doing so are corroborated by the evidence I cited in my First Declaration and the additional corroborating evidence I cite in this declaration in response to SFI's arguments and Dr. Rubin's opinions.

7. As I discuss further below (§§II.A-II.C), in reviewing SFI's POR and the Rubin Declaration, SFI appears to have three primary arguments challenging how Foster-Hayhurst meets the challenged claims: (1) SFI argues that Hayhurst is not analogous art (POR, 32-41); (2) SFI argues that POSAs would have lacked motivation to pursue the Foster-Hayhurst combination (POR, 41-63); and (3) SFI argues that Foster does not disclose claim 1's *shares* or *data chunks* (POR, 63-71). As I discuss in the sections below (§§II.A-II.C), in my opinion POSAs would disagree with each of these arguments.

**A. Response to Argument that Hayhurst (EX1007) is Not Analogous Art to the '854 Patent**

8. I have been informed that, in an obviousness analysis, prior art must be analogous to the patent being considered. I have been informed that there are two ways by which a prior art reference can be considered to be analogous, or in the same field of art: the reference can **either** be (1) in the same field of endeavor

as the challenged patent, regardless of the problems the challenged patent and the prior art address; **or** (2) reasonably pertinent to at least one of the problems being solved by the challenged patent. I have been informed that a prior art reference may be both in the same field of endeavor as the patent and reasonably pertinent to a problem in the patent, but the reference only has to meet one or the other to qualify as analogous art (i.e., in the same field **or** reasonably pertinent to one problem faced by the inventors). I have been informed that a prior art reference can be in the same field of endeavor even if it addresses different problems than the patent. I have also been informed that a reference is reasonably pertinent if, though in a different field, it would have commended itself to an inventor's attention in considering one of the problems the inventor is addressing. I have further been informed that in determining whether a reference is reasonably pertinent to a problem, one should consider a problem faced by the inventor, as reflected either explicitly or implicitly, in the specification. I have further been informed that courts have been directed to construe the scope of analogous art broadly because POSAs understand that familiar items may have obvious uses beyond their primary purposes, and POSAs often will be able to fit the teaching of multiple patents together like pieces of a puzzle. I have been informed that the analysis of whether a prior art reference is analogous art should take into account

any relevant evidence in the record cited by the parties to demonstrate the knowledge and perspective of POSAs.

9. SFI argues, and Dr. Rubin opines, that Hayhurst is not analogous art because it is allegedly not in the same field of endeavor as the '854 patent and allegedly not relevant to any of the problems addressed by the '854 patent. POR, 32-41.

10. As I explain below (§§II.A.1-II.A.2), it is my opinion that POSAs would have disagreed with SFI and Dr. Rubin on both accounts: in my opinion, POSAs would have recognized that Hayhurst is **both** in the same field of endeavor as the '854 patent **and** reasonably pertinent at least one problem (and in fact several programs) addressed by the inventors of the '854 patent.

**1. Response to Argument that POSAs would find Hayhurst to be in the same field of endeavor as the '854 patent.**

11. SFI argues “the '854's field is securing data,” including securely storing and communicating the secured data. POR, 33-35. I have not been asked to opine on whether SFI has identified the correct field of endeavor for the '854 patent. That is not necessary to my opinion because, for the reasons I discuss below, it is my opinion Hayhurst is (at least) in the same field of endeavor that SFI has identified for the '854 patent.

12. My opinion that Hayhurst is in the field of securing data is supported by Hayhurst's explicit disclosure. For example, Hayhurst's “Technical Field”

section explicitly says that Hayhurst “relates to the field of electronic media delivery...including the **secure handling and storage** of such media.” Hayhurst, [0002]. That Hayhurst explicitly says its “Technical Field” includes **secure** data handling (*e.g.*, data communication/transmission) and **secure** data storage would be dispositive to POSAs’ recognition that Hayhurst is in the field of securing data.

13. But even if Hayhurst had not explicitly identified its own field of endeavor as including secure handling and storage of data, the remainder of Hayhurst’s disclosure reinforces that Hayhurst’s teachings are in the field of securing data. *See also* First Declaration, ¶¶178-181 (describing how Hayhurst’s technique provides security for stored data).

14. Hayhurst provides (increased) **security** and deterrence via its segmented and distributed storage of the content. At [0008], Hayhurst says that its “segmented and distributed storage of the media file **provides a high level of security** and a deterrence to piracy.” POSA would agree because, in Hayhurst’s storage technique, “[n]o single subscriber unit has more than a segment of a media file, which segment, by itself, has little value.” Hayhurst, [0008]. That is, in Hayhurst data has “a high level of security” (Hayhurst, [0008]) and are protected from privacy because there is not a single location where the data is stored and no single storage device / user can, by itself, undermine the secure storage of the file overall because each storage device has only a small portion of the data set that, on

its own, cannot be used to recover the data set. For example, even if one storage device / user were subject to attack, other segments would still be secured at the other storage devices. As I discuss further below (§II.B), the security that Hayhurst provides via its distributed storage would have provided a benefit to Foster's system because Foster generally contemplates storing the encrypted data segments contiguously (i.e., the segments are stored with each other) at Foster's media server and then again at an end-user. Thus, unlike in Hayhurst, an attacker would only have to breach the security of a single storage device in Foster to obtain access to all *data chunks* of Foster's dataset.

15. Consistent with the above explicit disclosures in Hayhurst (and POSAs' understand thereof), Hayhurst expressly teaches that its "approach to segmenting the movie [data] provides **increased security** from piracy." Hayhurst, [0021]. Hayhurst further demonstrates that it is in the field of securing data because it teaches that its "segments are each **encrypted**...to prevent unauthorized viewing." Hayhurst, [0007]. Similarly, Hayhurst explicitly says that its technique provides additional data-storage security (i.e., Hayhurst's technique is in the field of securing data) because restoring the data in Hayhurst also requires knowledge of "the correct method for recombining the segments," which is "stored in a password-protected file, linked to an encryption key." Hayhurst, [0021]. In other words, Hayhurst's technique would provide additional security for and is

consistent with a system (like Foster's) that is encrypting segments of data but not using a distributed storage technique (like Hayhurst's).

16. Thus, as I've discussed above, Hayhurst explicitly and repeatedly provides disclosures in its specification that demonstrates that Hayhurst's field is securing data, which is the same field as the '854 patent that SFI alleges.

17. Despite the explicit and repeated disclosures in Hayhurst demonstrating that its field is securing data, SFI and Dr. Rubin attempt to limit Hayhurst to the field of "decentralized media delivery." POR, 35-36; Rubin-Decl., ¶¶100-104. To the extent SFI is alleging that is outside the field of securing data, then in my opinion POSAs would disagree because decentralized delivery and distributed storage of media is the primary way **how** Hayhurst provides "**security** and [] deterrence to piracy" of stored data. Hayhurst, [0008]. Hayhurst generates content segments and stores the segments on separate storage devices for the purposes of securely storing the data by making it more challenging for an attacker to obtain all of the *chunks* derived from the original data set. Indeed, like Hayhurst, the '854 patent **itself** says that one way that it secures data is via "geographical separation of the data storage facilities." EX1001, 19:64-20:3. In other words, the '854 patent itself recognizes that using distributed storage devices (as in Hayhurst) is within the field of securing data. Again, increasing security of

the data is one of the main reasons Hayhurst proposes to store segments of encrypted data on different storage devices.

18. I further note that the '854 patent also says that its disclosures are useful in “peer-to-peer file-sharing” and “video broadcasting systems” (EX1001, 81:40-50), which are two of the very same applications to which Hayhurst’s teachings are also directed, reinforcing that their techniques apply to securing data in similar contexts. *See* Hayhurst, [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.**”), [0010] (“It is contemplated that the request for the media file and subsequent responses for directing the segments to the requester can be handled without the intervention of the network’s main server **by using peer-to-peer architecture** or the like.”). POSAs would have known that peer-to-peer / distributed storage and distribution networks like Hayhurst’s could be used to deliver live television broadcasts. Indeed, I was co-author of several peer-reviewed journal articles that were intended for POSAs in the field of securing data that described techniques for using systems like Hayhurst’s to distribute live (and stored) media. Thus, the fact that Hayhurst uses peer-to-peer / distributed storage and distribution certainly does not take Hayhurst outside the field of securing data and, if anything, places it

squarely within that field because, as I've discussed, Hayhurst's distributed storage is done for the purpose of adding security (among other benefits).

19. I understand that SFI's argument is that notwithstanding Hayhurst's (repeated) "disclosures about the security of the media files" those disclosures should be ignored because those disclosures allegedly are only "tangentially relate[d]" to securing data in Hayhurst. POR, 36. In my opinion, POSAs would disagree with SFI's argument. As I discussed above, Hayhurst **explicitly** says that its "Technical Field" includes "the **secure** handling and storage of [] media." Hayhurst, [0002]. As I also discussed above, Hayhurst provides numerous disclosures explaining how Hayhurst's technique provides "**security** and [] deterrence to piracy" of stored data. Hayhurst, [0008]. In my opinion, POSAs would find that SFI's allegation that data security is "tangential" to Hayhurst (POR, 36) is irreconcilable with what Hayhurst itself says is its "Technical Field" and what Hayhurst says about how the security of storing data is enhanced by Hayhurst's distributed storage techniques. Again, POSAs would recognize that Hayhurst's "segmented and distributed storage" of data is not tangential to securing data—it is **how** Hayhurst secures data to "**provide[] a high level of security**." Hayhurst, [0008].

20. My opinion that Hayhurst's distributed storage and distribution technique is in the field of securing data is consistent with testimony that Dr. Rubin

provided at deposition. Dr. Rubin testified that to secure data from unauthorized access it “means that parties that shouldn’t – that haven’t been given access or haven’t purchased access and are not parties that should have the access can’t get access to the data.” Rubin-Depo., 108-109. He further testified that “[s]ecur[ing] data” means “protect[ing] data” and to “[p]rotect the data” means “[t]o 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.” Rubin-Depo., 108-109.

21. The same is true in Hayhurst as a result of Hayhurst’s distributed-storage technique. For example, Hayhurst’s technique provides “**security** and [] deterrence to **piracy**” (i.e., preventing parties that haven’t been given access or purchased access from accessing the secured data). Hayhurst, [0008].

22. SFI further argues Hayhurst that does not teach that security was a problem in prior-art systems “or that its invention addresses a need for more security.” POR, 36. As I discussed above (§II.A), I have been informed that under the field-of-endeavor inquiry, the problems being addressed are irrelevant because the field-of-endeavor test does not look to the problem that the patent purports to address.

23. But even if the problem(s) being addresses mattered, in my opinion, POSAs would understand that SFI is incorrect that Hayhurst did not recognize a need for more security in the prior art. Hayhurst explicitly says its “approach” to

storing and delivering data “provides **increased security.**” Hayhurst, [0021]. By recognizing that its technique provides increased security, Hayhurst was clearly recognizing and telling POSAs that Hayhurst believed there was a **need** for that “increased security” that Hayhurst sought to provide. As I’ve discussed above, Hayhurst explicitly teaches (several times) that its storage technique is designed to meet that need for increased security (including for already encrypted data).

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 understood that the secure data parser may also be integrated for protection and fault tolerance of any type of data in motion through any transport medium, including, for example, wired, wireless, or physical transport mediums.... [T]he secure data parser may be integrated with a file-sharing application (e.g., a peer-to-peer file-sharing application), a video broadcasting application, an electronic voting or polling application (which may implement an electronic voting protocol and blind signatures, such as the Sensus protocol), an email application, **or any other network application** that may require or desire secure communication.

*See, e.g.,* EX1001, 81:20-50

25. I note again that Hayhurst **expressly** says its Technical Field “relates to” the “**secure** handling and storage” of data (Hayhurst, [0002]) and Hayhurst achieves that secure handling of data using a technique the ’854 patent **itself** says secures data (including already-encrypted data), *i.e.*, distributed data storage.

26. Thus, in my opinion, POSAs would find that Hayhurst is analogous art to the ’854 patent at least because it is plainly within the field of securing data, which is the field of endeavor that SFI has identified for its ’854 patent.

**2. Response to Argument that Hayhurst is not reasonably pertinent to at least one of the problems to which the ’854 patent relates.**

27. Even if Hayhurst were (somehow) found not to be in the same field of endeavor as the ’854 patent, in my opinion POSAs would find that Hayhurst is also reasonably pertinent to one or more of the problems to which the ’854 patent relates.

**a. Hayhurst is reasonably pertinent to the problem of user-independent security.**

28. SFI argues that the problem to which the ’854 patent relates is providing a “cryptographic system whose security is user independent.” POR, 38-39 (quoting EX2033, ¶106). SFI does not explain what it means by “user independent” security. At deposition, Dr. Rubin opined that user-independent security means “[t]he data is secured in a way that is not dependent on a particular

user; it's dependent on the data." Rubin-Depo., 115-116. Dr. Rubin identified an example of user-independent security in the patent is its disclosure of "encrypting the data, splitting it into shares," which Dr. Rubin said "is not going to depend on a particular user who's going to be using the data." Rubin-Depo., 116. Dr. Rubin also said, for example, user-independent security includes when "no user will have enough of the data to be able to obtain the original data." Rubin-Depo., 116. Dr. Rubin further testified that an example of that includes when "the data is stored such that in no user, at any particular location where a piece of the data is, can reconstruct the whole data." Rubin-Depo., 115-120. As I discuss below, POSAs would recognize that all of those examples of user-independent security are found in Hayhurst.

29. In particular, Hayhurst's distributed-storage technique secures data in a user-independent manner because Hayhurst segments and distributes data without requiring user action or user knowledge for either the storing or recovering of the secured data. For example, Hayhurst discloses a "main server 26" that "distributes [] the movie segments...to all of the participating subscriber units []." Hayhurst, [0027]. Hayhurst continues that "[s]uch 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.**" Hayhurst, [0027]. That is, the users play no part in the segmenting, distribution, or storage of Hayhurst's segments (beyond

having a storage device that is connected to the necessary network). Users do not know what segments are being stored, what other users have segments for the same data set (or what segments any individual has), or how any segment has been encrypted. Users do not select the encryption keys or encryption methods, nor is an encryption key ever revealed to a user. Users do not know how a file was segmented or how many segments exist for any given file. Users also cannot undermine the security that Hayhurst's system provides (and thus the security is not dependent upon users) because no one subscriber unit has more than one segment, and a single segment has little value on its own. Hayhurst, [0021] (disclosing that "each subscriber...will store only very small portions of the movie on the[ir] storage device" and thus "decoding of any individual bundle will be virtually impossible"), [0029] (explaining that the main server has to "authoriz[e]" users to recover the data).

30. Hayhurst further teaches that additional security is provided because "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.**" Hayhurst, [0021]. That further reinforces that the security provided by Hayhurst's technique is user-independent because users have no idea how to recombine segments even if they were given all of the segments

(and even then they would need to be given the encryption key(s) that are not dependent on, selected by, or revealed to users).

31. Thus, in my opinion, POSAs would have considered Hayhurst when considering the problem of providing a cryptographic system whose security is user independent, which is the only problem SFI identifies in alleging Hayhurst is not analogous art.

32. In discussing the above problem that the inventors were allegedly involved with, SFI's POR quotes Dr. Rubin's declaration, which, in turn, quotes the patent alleging that its disclosure includes "provid[ing] a cryptographic system whose security is user-independent while still supporting mobile users." POR, 38-39 (quoting EX2033, ¶106 (quoting EX1001, 1:35-37, 2:4-44)). If SFI and Dr. Rubin are arguing that the '854 patent's claims are limited to or require supporting mobile users, then that is a new claim construction that no party had previously identified. In my opinion, POSAs would also disagree with that construction because there is no indication in the plain language of the claims or intrinsic evidence that would limit and/or require mobile devices.

33. I note too that at deposition Dr. Rubin admitted that the independent claims of the '854 patent do not refer to a mobile user, that it was not his opinion that the disclosed system or independent claims required mobile users (because he said he "ha[s]n't studied that" and "it's not an opinion that's in [his] report"), and

he further testified that he “ha[d] not looked at th[e] question” of whether a system without mobile users is outside the scope of claim 1 of the ’854 patent. Rubin-Depo., 115-120. Indeed, Dr. Rubin testified that he was just “quoting from the summary of the invention, but [he] didn’t elaborate or give it further thought as to what the supporting mobile user – how that might relate to the claims or anything like that.” Rubin-Depo., 119-120. I have given that some thought, and I disagree that POSAs would think that supporting mobile users was relevant to a problem the patent was allegedly trying to address. At a minimum, to the extent SFI has correctly identified user-independent security as a problem that the ’854 patent concerns itself with, POSAs would recognize that that would be a consideration with or without the need to support mobile users.

34. In any event, in my opinion, POSAs would understand that Hayhurst **does** support mobile users because POSAs knew that distributed/peer-to-peer storage techniques (like Hayhurst’s) also supported and were beneficial to mobile users. In response to SFI’s and Dr. Rubin’s arguments otherwise, my opinion that Hayhurst’s technique would support mobile users is corroborated by EX1091, [0010], which discloses a “peer-to-peer (p2p) computing model [that] lends itself to the dynamic environments of **wireless devices** including, but not limited to, **cell phones, pagers, PDAs, and other consumer devices.**”

35. My opinion is further corroborated by prior art papers in the field of securing data that I co-authored myself that discussed using distributed storage and/or distribution networks like Hayhurst's that supported mobile users.

36. Moreover, as I discussed above (§II.A), I understand that a prior art reference can be analogous art if it is reasonably pertinent to any **one** of the problems addressed by the patent's inventors. Though SFI's allegation that Hayhurst is not analogous art only identifies one problem allegedly addressed by the '854 patent, in my opinion POSAs would understand that the '854 patent identifies other alleged problems that were allegedly being addressed by the patent. In my opinion, POSAs would have found that Hayhurst is also reasonably pertinent to several of those other problems the '854 patent discusses.

**b. Hayhurst is reasonably pertinent to the problem of computer-network security.**

37. The '854 patent expresses concern with the problem of how to “ensur[e] [that] data stored and transmitted over” “computer networks” “cannot be read or otherwise compromised.” EX1001, 1:30-39. *See also* Rubin-Depo., 109-110 (testifying that “ensuring data stored and transmitted over these systems cannot be read or otherwise compromised” was “part of the problem that the invention solves”).

38. In my opinion, Hayhurst is reasonably pertinent to that problem because Hayhurst too “relates to the field of **electronic** media **delivery**...including

the **secure handling and storage** of” data (Hayhurst, [0002]) and “deter[ing] [] piracy” (i.e., unauthorized access) (Hayhurst, [0008]). *See also* Hayhurst, [0016] (describing various computer networks in which Hayhurst’s technique can be integrated including a ”cable television (CATV) network,” or “any other communications network of users or subscribers, such as those linked by a local area network (LAN) or a wide area network (WAN)”). That is, Hayhurst is also addressing the problem of how to ensure that data stored and transmitted over a computer network maintains “increased security from piracy” (*i.e.*, unauthorized reading or comprising of security).

39. Thus, in my opinion, POSAs would have considered Hayhurst as reasonably pertinent to the problem the ’854 patent identifies of ensuring that data stored and transmitted over computer networks cannot be read or otherwise compromised.

**c. Hayhurst is reasonably pertinent to problems addressed by distributed-storage (e.g., the problem of data remaining whole and subject to attack).**

40. The ’854 patent also discusses problems that occur with data “remain[ing] whole and intact and subject to attack” (EX1001, 56:47-50) and how to “protect[.]...against compromise of any individual data storage facility” (EX1001, 14:4-51, 19:60-64).

41. While the section of SFI's POR that alleges Hayhurst is not analogous art focuses only on the alleged problem of providing user-independent security, elsewhere in its POR, SFI quotes Dr. Rubin's opinion that the patent explains that "when 'data ... remains whole and intact,' it is 'subject to attack'" and that "[t]he '854 addresses this concern, in part, by 'performing a cryptographic parsing and splitting of the encrypted file'" and thus "teaches that splitting the data set into shares helps secure the data." POR, 68-69 (quoting EX2033, ¶¶149-151 (quoting EX1001, 56:47-61)). Thus, SFI recognizes that the '854 patent also alleges there were problems arising from secured data remaining whole and intact and subject to attack. SFI and Dr. Rubin do not discuss whether Hayhurst is reasonably pertinent to that problem, let alone allege and demonstrate that Hayhurst is not reasonably pertinent to that problem. As I discuss further below, POSAs would find that Hayhurst is reasonably pertinent to that problem as well.

42. At deposition, Dr. Rubin testified that the patent addresses a "concern" of "when data remains whole or intact it's subject to attack" because, in that case, when "an adversary breaks into the site where the data is, then all the data's there." Rubin-Depo., 124-126. The '854 addresses those concerns / problems by splitting and "storing cryptographic data in geographically remote secure data storage facilities" (i.e., distributed storage facilities). EX1001, 5:53-56; *see also* EX1001, 3:34-44, 34:53-55, 35:8-32, 36:42-59. That is, the '854

patent addresses an alleged problem that arose in prior art techniques that kept encrypted data together in one piece and stored it in one place. The patent addresses that problem by breaking the data apart (before or after encrypting) and storing the pieces (with or without individually encrypting the pieces) on different storage devices.

43. I note that Dr. Rubin provided deposition testimony that is consistent with my opinion regarding the patent's purported solution to that problem. Rubin-Depo., 126-128 (testifying that the patent addresses the concern of data remaining whole and intact by "put[ting] one of those shares, for example, in one place and another share in another place" "[s]o now the data is not whole or intact, so if an attacker comprises [sic: compromises] one of the sites, but not the other sites, then they haven't gotten the data"); Rubin-Depo., 129-130 (by having one piece in one location and the other piece in the other location "[t]hat makes it harder for an attacker to get the data"); Rubin-Depo, 130-132 ("I'm not offering an opinion at this point the case on th[e] question" of whether Claim 1 requires cryptographic splitting of the data); Rubin-Depo., 133-135 (testifying that the objective of preventing an employee from obtaining more than one portion of a data set is met by having "geographically remote" storage devices so that "a physical compromise of a system would only yield the portion that's there, but then you would need the other portion at the other remote location to actually get any useful data").

44. In my opinion, POASs would find that Hayhurst is not only reasonably pertinent to those same problems discussed above but solves them the same way the '854 patent alleges it addresses those problems: via segmenting encrypted data and storing the segments on distributed storage devices, as I discussed above (§II.A.1). Hayhurst, [0021].

45. Indeed, limitation [1D] requires “storing” portions of data “on a **plurality of different storage devices**,” which is one of the teachings from Hayhurst that I opined POSAs would have used in the Foster-Hayhurst combination. By claiming as a limitation, the very thing that Hayhurst teaches, it reinforces to POSAs that Hayhurst is reasonably pertinent to one of the problems discussed in the '854 patent and, in fact, solves that problem in the same way. POSAs would plainly find that a prior art reference that teaches segmenting data and storing it on distributed storage devices for the purposes of increasing security, like Hayhurst (*e.g.*, Hayhurst, [0008]), was reasonably pertinent to a patent whose claims require storing segments of data on multiple storage devices and whose specification alleges that that segmenting and distributed storage was done for the purposes of increasing security.

46. Thus, in my opinion, POSAs would have also considered Hayhurst reasonably pertinent to the problem the '854 patent identifies of security concerns when data remains whole and intact and subject to attack and the related problem

of preventing an overall failure of security for a data file even if one (or some subset) of the data storage facilities become insecure and/or fail.

47. Thus, for the reasons I discussed above in this section, it is my opinion that Hayhurst is also analogous art because POSAs would understand that Hayhurst is reasonably pertinent to at least one (and in my opinion several) of the problems to which the '854 patent's inventors were involved.

**B. Response to Argument that POSAs Would Not Have Had Motivation For the Foster-Hayhurst Combination**

48. My First Declaration provided my opinions (backed by citations to Foster, Hayhurst, and several corroborating references) that POSAs would have been motivated to pursue the Foster-Hayhurst combination that I described in my First Declaration with a reasonable expectation of success because POSAs would have recognized that implementing Foster's system using Hayhurst's distributed-storage technique would have provided **numerous** benefits for Foster's system. First Declaration, ¶¶182-185; *see also id.* ¶186 (explaining my opinion that POSAs would have found that Foster-Hayhurst "would have also been an application of Hayhurst's known distributed storage technique to Foster's known encryption system and method").

49. Any one of the benefits that I identified (summarized again below) would have motivated POSAs to pursue the Foster-Hayhurst combination, as I corroborated with evidence from Hayhurst itself and from POSAs' background

knowledge about the benefits of a distributed storage and delivery system, like Hayhurst's:

- “advantageously reduc[ing] the content provider’s bandwidth requirements” (citing Hayhurst, [0005]);
- “provid[ing] a high level of security and a deterrence to piracy” (quoting Hayhurst, [0008]; citing also, EX1043, [0005]-[0010]);
- “support[ing] redundancy in segment available from storage (citing Hayhurst, [0009]); and
- “benefits in bandwidth, security, and redundancy (also referred to as ‘fault tolerance’),” “aggregate speed,” “efficiency / ‘intelligence’” (citing and describing EX1062, [0022]-[0023]; EX1037, [0003], [0012]-[0013], [0015], [0018]-[0024]; EX1043, [0005]-[0010]; EX1050, 1:19-22, 2:31-38, 11:11-26, and 15:39-16:20).

50. In its POR, SFI argues that POSAs would not have sought to achieve any of the benefits that I identified because, according to SFI and Dr. Rubin, using Hayhurst’s distributed-storage technique in Foster’s system would have allegedly been incompatible with Foster’s goal of “avoiding delays” to improve “streaming live television.” POR, 42. Dr. Rubin similarly asserts that the combination “would introduce enormous delays...rendering Foster’s system unsuitable for streaming live television.” EX2033, ¶141. Dr. Rubin does not explain what he meant by “enormous delays,” nor how he (let alone a POSA) would quantify “enormous

delays,” nor explain why there would have been “enormous delays,” nor did Dr. Rubin cite any evidence in support of that assertion that there would have been enormous delays. (Dr. Rubin’s opinion in ¶141 is also undermined directly by his immediately preceding paragraph ¶140 because in ¶140 he recognizes that Foster **also** includes “content **stored** on a DVD” and thus includes embodiments that are not about streaming live television, as I discuss further below.)

51. 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” in Foster-Hayhurst compared to Foster alone. *See supra* §II.B.2 (discussing how Foster-Hayhurst would reduce delays over Foster alone); First Declaration, ¶184.

52. For one, I note that the ’854 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, 69:56-58. 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 ’854 patent’s teachings).

53. Below is a list of several, peer-reviewed articles and/or papers relating to presentations given that relate to downloading data from replicated sources and the benefits thereof. Each of these papers were published and publicly available in

the prior art that I was a co-author on that are in the field of securing data and which POSAs would have been familiar with:

- Bobby Bhattacharjee, Mostafa Ammar, Ellen Zegura, Viren Shah, and Zongming Fei. “Application-Layer Anycasting”, In Proceedings of INFOCOM’97, pages 1388–1396, Kobe, Japan, 1997.
- Zongming Fei, Bobby Bhattacharjee, Ellen W. Zegura, and Mostafa Ammar. “A Novel Server Selection Technique for Improving the Response Time of a Replicated Service”, In Proceedings of INFOCOM’98, pages 783–791, San Francisco, CA, 1998.
- Ellen W. Zegura, Mostafa Ammar, Zongming Fei, and Samrat Bhattacharjee. “Application-Layer Anycasting: A Server Selection Architecture and Use in Replicated Web Service”, Transactions on Networking, Vol. 8, Issue 4, pages 455-466, August 2000.
- Suman Banerjee, Seungjoon Lee, Bobby Bhattacharjee, and Aravind Srinivasan. “Scalable Resilient Multicast”, Proceedings of ACM SIGMETRICS, pages 102–113, 2003.

54. Below is a list of several, peer-reviewed articles and/or papers relating to presentations given that relate to large-scale streaming of data (including live streaming thereof) from multiple sources and the benefits thereof. Each of these papers were published and publicly available in the prior art that I was a co-author on that are in the field of securing data and which POSAs would have been familiar with:

- Suman Banerjee and Samrat Bhattacharjee. “Scalable Application-Layer Multicast for Content Distribution”, Computer Communications Workshop, 2001.
- Suman Banerjee, Bobby Bhattacharjee, and Christopher Kommareddy. “Scalable Application- Layer Multicast”, Proceedings of ACM SIGCOMM, pages 205–217, 2002.
- Suman Banerjee, Christopher Kommareddy, Koushik Kar, Bobby Bhattacharjee, and Samir Khuller. “Construction of an Efficient Overlay Multicast Infrastructure for Real-time Applications”, Proceedings of IEEE Infocom, pages 1521–1531, 2003.
- Suman Banerjee, Seungjoon Lee, Bobby Bhattacharjee, and Aravind Srinivasan. “Scalable Resilient Multicast”, Proceedings of ACM SIGMETRICS, pages 102–113, 2003.
- Suman Banerjee, Seungjoon Lee, Ryan Braud, Bobby Bhattacharjee, and Aravind Srinivasan. “Scalable Resilient Media Streaming”, In Proceedings of ACM NOSSDAV’04, pages 4–9, 2004.
- Suman Banerjee, Christopher Kommareddy, Koushik Kar, Bobby Bhattacharjee, and Samir Khuller. “OMNI: An Efficient Overlay Multicast Infrastructure for Real-time Applications”, Special Issue of Computer Networks on Overlay Distribution Structures and their Applications, Vol 50:6, pages 826-842, 2005.

- Suman Banerjee, Seungjoon Lee, Bobby Bhattacharjee, and Aravind Srinivasan. “Resilient Multicast using Overlays”, IEEE/ACM Transactions of Networking, Vol. 14, No. 2, pages 237–248, April 2006.

55. As I discuss further below (§§II.B.1-II.B.2), POSA would not agree with SFI’s argument (at least) because SFI’s argument is premised on an interpretation of Foster’s disclosure that is limited to “streaming live television.” However, POSAs understand that Foster disclosure is relevant to much more than that, including the **storage** of Foster’s encrypted content (both before and after delivering to end-users, whether via streaming or otherwise).

56. POSAs would have understood that reducing latency (and thus improving the delivery of streaming of live television) is **one** of the benefits Foster says its system and encryption technique can provide, in some use cases. But as I discuss further below (§§II.B.1-II.B.2), Foster’s teachings are not limited to that exemplary benefit of reducing latency for live streaming of television.

57. POSAs would read Foster for all that it teaches. As I discussed in my First Declaration (¶167, ¶174, ¶177, ¶182) and further below (§§II.B.1-II.B.2), Foster teaches embodiments and implementations where Foster’s encrypted content is not streamed live but rather **stored** for and by recipient(s) for later playback (*e.g.*, a movie downloaded from a video-on-demand server). That is a context explicitly taught by Foster and one that is relied upon in my First

Declaration and reinforced by POSAs' background knowledge about the systems and content discussed in Foster. It is in (at least) that storage context that POSAs would have recognized that Hayhurst's distributed-storage technique would have offered benefits to both the storage and delivery of Foster's encrypted content. In my opinion, once Foster's teachings are correctly read as not being limited to live streaming, SFI's arguments against POSAs' motivation-to-combine Foster and Hayhurst fall apart because they depend on an incorrectly confining Foster to streaming live broadcasts, which directly contradicts Foster's broader teachings that include storing. (In fact, as I discuss further below, §II.B.1, POSAs would understand that even Foster's streaming context typically includes storage, both before and after delivery to end users.)

**1. It is clear to POSAs from Foster's disclosure that Foster stores encrypted content.**

58. As I discussed in my First Declaration (¶¶167-169), Foster describes benefits to attaching an encryption key to the data that that encryption key encrypts, which Foster describes that as "synchronization" (because the data and keys can be stored and delivered at the same time). Foster, [0002]-[0007]. Foster explains that attaching an encryption key to the data that that encryption key encrypts provides several benefits. For example, by keeping the encryption key with the *share* of Foster's data that it encrypted, the risk of applying the wrong key to the wrong share of data is reduced (because the system would not have to

“search” for the correct key, such as in systems where keys are stored and/or delivered separately from the data they encrypt). Foster, [0002]-[0007]. Placing the encryption key with the *share* of Foster’s data that it encrypts **also** reduces the time it takes to locate the correct key for decrypting a portion of the dataset (i.e., reduces latency from having to locate the correct key), thereby facilitating “random access” to portions of “protected content.” Foster, [0002]-[0007]. Foster is clear (and POSAs would have recognized) that the benefit of reducing latency for random access to content is **not** limited to streaming content, let alone streaming “live” content. Instead, as I explained in my First Declaration (§168), Foster’s technique allows for random access to stored content (e.g., movie) at a particular (“random”) *chunk*: for example, starting a stored movie at any point (say the middle where your favorite scene is), rather than have to start the movie at the beginning every time. Foster, [0002]-[0007]. Foster calls the technique of attaching an encryption key with the encrypted data “synchroniz[ation].” Foster, [0002].

59. In its POR, SFI argues (and Dr. Rubin opines) that Foster’s “fundamental goal” is to use “synchronization” to “reduc[e] delays in viewer access to content **as it is transmitted**” in real-time. POR, 42-43. In my opinion, POSAs would disagree that reducing delay in accessing real-time content as it is transmitted is the **fundamental** goal of synchronization in Foster and to the extent

SFI is arguing that that is the **only** goal of synchronization in Foster, POSAs would disagree in my opinion. Using synchronization for the benefit of streaming live content is **a** benefit in Foster, when Foster is applied to streaming live content (but that is just one of the applications Foster discusses). For example, Foster certainly teaches “synchronized transmission to...a receiver.” Foster, [0008]. But in that very same sentence, Foster **also** teaches attaching an encryption key to encrypted data to beneficially provide “for synchronized...**storage** by a receiver.” Foster, [0008].

60. That is, Foster is **explicit** that synchronization can be used to benefit transmission **or storage** of content (or both in combination). Synchronization benefits apply to applies to **stored** data as well because, for example, as I explained above, it supports random access to a movie at a random point in the movie, rather than have to start the movie at the beginning and wait for the movie to reach the portion you’re interested in viewing.

61. My opinion that Foster’s synchronization is not limited to live-streaming applications is reinforced by Foster repeatedly saying synchronization is useful for either data “transmission to, **or storage by**, a recipient.” Foster, Abstract; *see also* [0007]-[0008], [0048], [0064], [0079], claim 11. Indeed, I understand that in the IPR of the ’609 patent (which I understand is IPR2024-00214 and for which I am also submitting a reply declaration), SFI **admits** that

Foster “**teaches**...the ‘recipient 430 can....**store** the encrypted content.’” 609-POR, 44 (quoting Foster, [0079]), 51 (similar). At deposition, Dr. Rubin acknowledged that, in Foster, “[t]he recipient...could decide to **store** it [i.e., received content].” Rubin-Depo., 191. As I noted above, in EX2033 ¶140, even Dr. Rubin admits that “Foster’s system focuses upon streamed content [] or **content stored** on a DVD.”

62. Thus, it is readily apparent to POSAs and made clear by multiple disclosures in Foster that Foster includes processing a data set for the purposes of **storing** a Foster-encrypted version of the data set.

63. As demonstrated by the following quote from Foster, Foster is even explicit that synchronization reduces latency for streaming purposes **and** storage-based purposes, as would be used for the Digital Rights Management Systems (“DRMS”) implementations discussed in Foster:

[A] recipient will receive or **store encrypted content** along with all information necessary to decrypt the content. This not only **prevents the synchronization problems** associated with previous systems, **but also provides random access of content**. As indicated above, both DRMS and CAS have inherent latencies (e.g., delays) in providing random access of content. Specifically, because each content packet transmitted under the present invention is “packaged” with the title key that was used to encrypt it, a recipient can access and decrypt the content stream at any packet.

Foster, [0064].

64. As Foster explains, in Foster’s DRMS implementations, “content is packaged and loaded [i.e., stored] onto a web server” and subsequently “**downloaded to** [i.e., stored at a] personal computer” for on-demand accessing at “rendering time” (i.e., when the user wants to view stored content at some indefinite time after downloading/storing). Foster, [0005], [0049] (explaining how users access “content on media server,” which POSAs understood is **storing** the content before it has been purchased so it could be downloaded by multiple users at various times); FIG. 1 (content is accessed for “streaming **or stored**”).

65. Foster at [0050] says that “the content **can** be delivered through streaming delivery (e.g., if the file is large).” Foster says it “can” be delivered via streaming “**if**” the file is large because the file would not **have** to be delivered via streaming (e.g., if the file were not too large or when the storage devices had sufficient memory to store the file). POSAs would understand that one of the things that Foster at [0050] is suggesting is that if someone is downloading a large file (e.g., a lengthy, high-definition movie) it may take a long time to download and store the entire file. Thus, rather than wait for the entire file to be downloaded (i.e., stored) before viewing it, a user could elect to start viewing the content via streaming while it is also being downloaded and stored. (The content is also being stored because it was purchased, or could be viewed again over the course of several days, or to support interactive television, like pausing and rewinding, etc.)

66. In response to SFI's and Dr. Rubin's arguments that Foster's synchronization is only relevant to streaming live television, my opinion that POSAs would have understood Foster's DRMS embodiments to be for the purposes of **storing** content is also corroborated by SFI's EX2039, which recognizes that DRMS "safeguards content when **at rest** [i.e., stored] in addition to the transport phase," and extends to content "**stored** on a receiving device" and for "offline playback," *i.e.*, not just for live streaming (*i.e.*, "offline"). EX2039, 6-10. POSAs would have had the same understanding of DRMS in the prior art.

67. Consistent with Foster's DRMS disclosures, Foster also teaches that its payloads can be processed according to the MPEG-standard's so-called "program stream," which Foster says (and POSAs knew) is intended for use "**in fixed storage.**" Foster, [0073]. My opinion that POSAs knew as part of their background knowledge that an MPEG program stream is designed and intended for storage is supported by the following corroborating references.

68. SFI's EX2014, for example, acknowledges that the MPEG "[p]rogram streams are **designed for...storage.**" EX2014, 1.

69. As another example I cite in response to SFI's arguments and Dr. Rubin's opinions, EX1108, teaches that **both** "[t]he program and transport **streams** may be recorded on [i.e., stored on] or played back from a digital video disc (DVD), video tape, magnetic or optical disk drive or other

suitable storage device” (EX1108, 1:65-2:31), which are all examples of storage devices in the ’854 patent (EX1001, 66:53-59, 67:15-19, 70:19-25). EX1108 further discloses that it was known that even with the MPEG transport stream it was conventional that “a prior art video server may interact with a peripheral **such as a storage device** in a television set-top box **to download a transport stream corresponding to a video-on-demand selection.**” EX1108, 1:65-2:31. EX1108 is thus consistent with my opinion in my First Declaration (and discussed further below) that Foster is consistent with a video-on-demand (VOD) system, where encrypted content is both stored at the provider (e.g., a cable company and/or movie studio) **and** at the recipient (e.g., in a cable customer’s set-top box). *See also* Foster, [0052] (disclosing that content is “delivered to receiver 28 (e.g., a set-top box)”).

70. As another example I cite in response to SFI’s arguments and Dr. Rubin’s opinions, EX1109 (an IEEE article from 1995) teaches that the “MPEG-2 System layer describes how elementary streams comprising one or more programmes are multiplexed together to form a single data stream suitable for digital transmission **or storage.**” EX1109, 1.<sup>2</sup> EX1109 further taught that “[t]he

---

<sup>2</sup> The IEEE Xplore website indicates that EX1109 was published in “IEE Colloquium on MPEG-2 - What it is and What it isn’t,” in connection with a

programme stream.... is **intended for the storage** and retrieval of programme materials from **digital storage media**.” EX1109, 2; *id.*, 3-9 (explaining how “data bytes [are] taken sequentially from the original elementary stream” to form a “PES-packet”), 13 (“The ‘programme stream’ is biased **towards the storage** and replay of a single programme **from a digital storage device**....”).

71. As another example I cite in response to SFI’s arguments and Dr. Rubin’s opinions, EX1072 discloses as background information that content distributors (e.g., cable and satellite companies) had been developing and releasing “[i]nteractive television (iTV)” that provided, among other features, “2) personal video recorders (PVRs) which enable users to record, store, pause, and playback programming using hard-drive storage mechanisms, as opposed to conventional video tape, 3) video on demand (VOD) which enables viewers to order any type of

---

conference on “24 January 1995”; IEEE Xplore also indicates that EX1109 was added to IEEE Xplore on “06 August 2002.”

<https://ieeexplore.ieee.org/document/478032>. Based on the above evidence and my review of the indicia appearing on EX1109 and my understanding of the IEEE and IEEE Xplore, it is my opinion that EX1109 was publicly accessible in the prior art by being disseminated or otherwise made available to POSAs exercising reasonable diligence by February 1995, or at least by August 2002.

content on demand, such as movies, television shows, special events, sports programs, and other content, and to control its delivery on their receiving devices by pausing, rewinding, or fast-forwarding them.” EX1072, [0002]-[0003].

EX1072 further recognized as background information, that the “content distributor’s access control system, often referred to as a Conditional Access system” would interact with “a user’s set-top box” having “the appropriate middleware software that is compatible with the iTV applications being deployed at the content distributor's head-end.” EX1072, [0004].

72. EX1072 further discusses as background information that it was “desirable to have methods and systems that allow **a service provider to devise, deploy and manage ... video-on-demand service packages.**” EX1072, [0011]. In EX1072’s example, prior-art system, included numerous places where content would be stored – both before delivery and after delivery to customers:

[0042] ... The system operates with **a built-in storage device in the client device for storing the programs, for instance, a hard disk drive or any other storage media or by storing programs on a server at the head-end.** Exemplary personal video recorders are designed by companies such as TiVo, Inc. and manufactured by companies such as Sony Corporation and U.S. Phillips, Inc.

[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.**

73. Accordingly, that Foster's teachings include storage-based embodiments is readily apparent from Foster's text, as explicitly stated in Foster and POSAs would independently understand from their background knowledge of the DRMS implementations Foster explicitly identifies for loading (storing) encrypted content to web servers for download and storing by recipients.

74. Finally, even in the implementations of Foster that are for streaming content for use in a conditional access system ("CAS"), those systems do not only deliver "live" broadcasts. And to the extent SFI meant real-time broadcasts – like a live sporting event – POSAs would know that a percentage (if not a majority) of television at the time was not streaming live events, but rather streamed from content that has been pre-recorded and stored by the content provider (e.g., a television sitcom, a movie, a documentary, etc.).

75. In response to SFI's arguments and Dr. Rubin's opinions regarding CAS in Foster, POSAs knew that CAS is used in video-on-demand (VOD) systems, where "content **is stored pre-encrypted** on hard drives in the VOD server." EX1110, [0008]. POSAs similarly knew that the content delivered by VOD was commonly stored again at a "client device[']s" or in "**storage** media in which the **delivered content can be stored**" (EX1072, [0043]). For example, my opinion in that regard is corroborated by EX1100, which notes as background information that prior art services, like "video on demand (VOD)" had "need for secure delivery of content," where "a content provider must encrypt the content" first. EX1100, [0012]-[0013]. EX1100 further notes as background information that "encrypt[ing] content in real time as it is delivered to the customer" was "not always desirable or feasible" and so "there is a need...for content provider to be able to encrypt content before it is transmitted over the Internet as opposed to encrypting it in real time," which "would reduce cost and overhead that are associated with real time encryption." EX1100, [0013].

76. In response to SFI's arguments and Dr. Rubin's opinions regarding the use of a CAS in Foster, my opinion that POSAs understood that Foster's CAS embodiments were broader than "live" streaming and also relevant to the **storing** of encrypted content is corroborated by the following references (along with the references I discussed above).

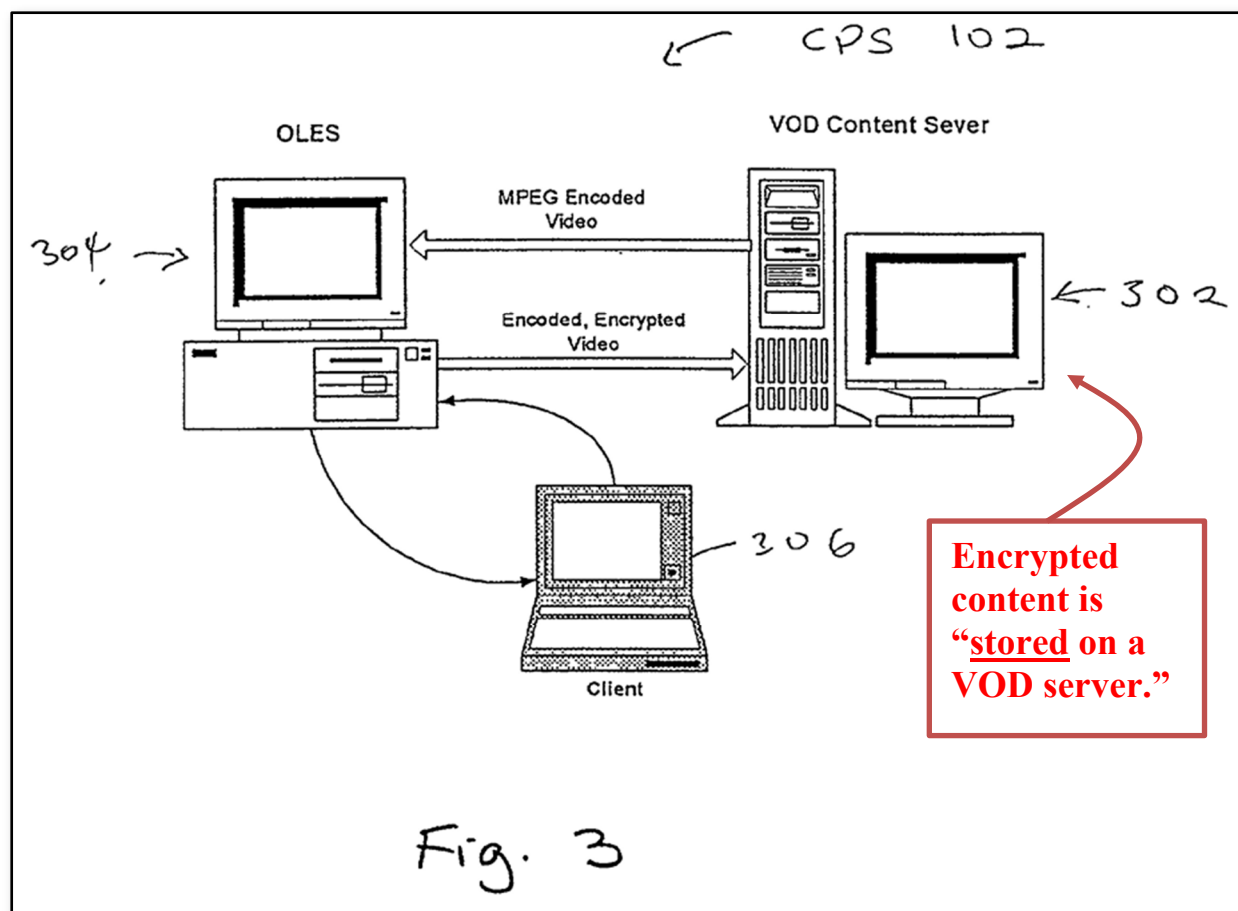
77. For one, Hayhurst itself recognizes that in “deliver-on-demand,” such as for “cable television (CATV) system (or for other content provider[s]),” it was desirable for the end-user consumer (e.g., the user that has purchased and received an encrypted movie file), to “assemble[ the segments] into the complete media file (movie) for display, **or stored for later display**, on the requesting subscriber's unit.” Hayhurst, [0004]-[0006].

78. As another example in response to SFI’s arguments and Dr. Rubin’s opinions, I cite EX1062 (cited First Declaration, ¶184), discloses as background information that “[v]ideo on demand, such as through cable and satellite providers” include “head end hardware” ([0002]), which is where content is stored from VOD. EX1062 shows “a traditional client/server type of media-on-demand network,” where the central server “may be a regional server or local hub” that “delivers the content to the requesting user.” EX1062, [0005]. EX1062 further notes that the “content [is] available from peers and centralized storage.” EX1062, [0018]. *See also* EX1062, [0043] (noting that “[t]he user may store the content on the device for any period of time the user desires”).

79. As another example I cite in response to SFI’s arguments and Dr. Rubin’s opinions is EX1037 (cited First Declaration, ¶184). EX1037 notes as background information that the prior art includes “Content Delivery Networks,” that POSAs knew where systems having servers that stored content to support the

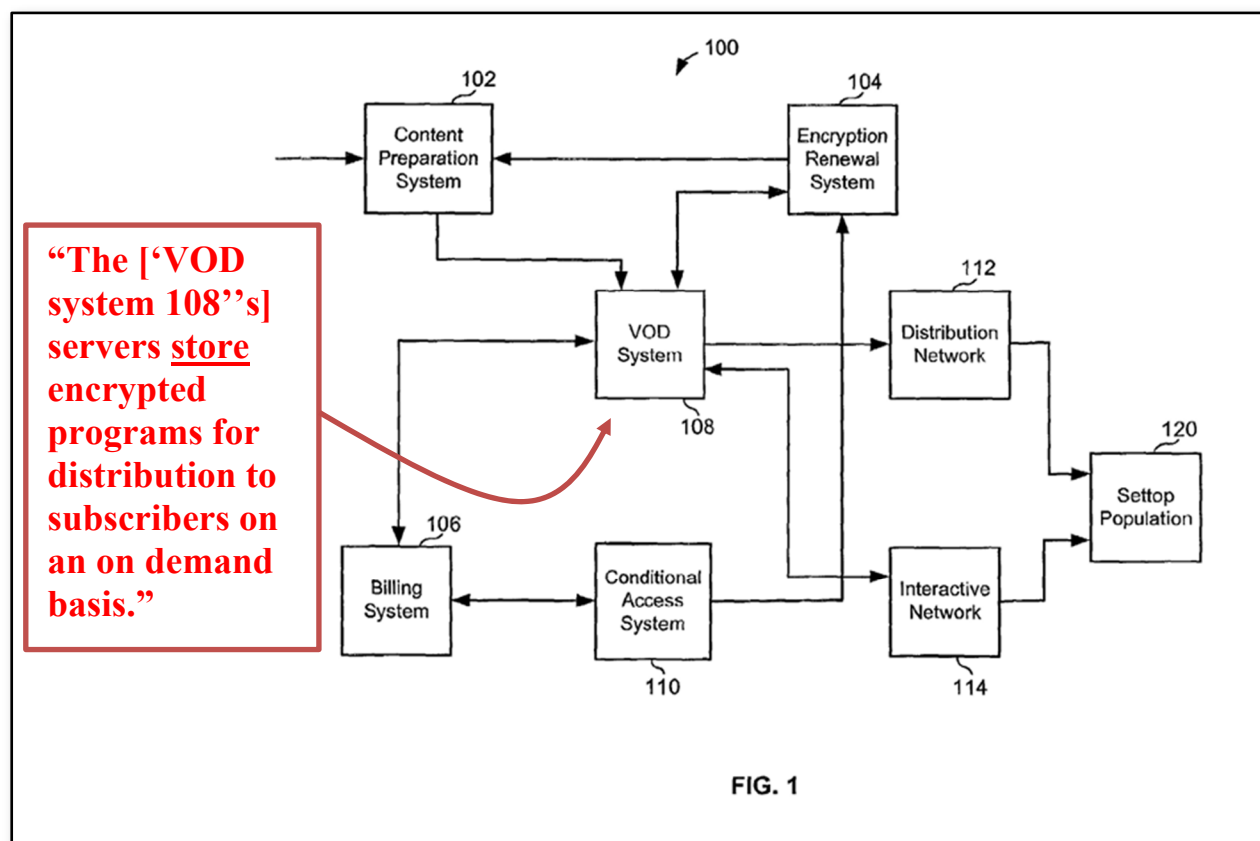
delivery of content-service-provider content. EX1037, [0005]. EX1037 at [0019]-[0025] notes benefit of “transfer[ring] by downloading data from multiple sources or channels in parallel” (as in Hayhurst). One of the (several) benefits of those delivery systems include the support for “Streaming” where “one can actually watch a video as it is being downloaded from multiple sources in parallel,” such as for “Video-on-Demand.” EX1037, [0022].

80. As another example I cite in response to SFI’s arguments and Dr. Rubin’s opinions, EX1073 at [0001]-[0004] corroborates my opinion that POSAs knew that “conditional access” systems (like those discussed in Foster) were used for VOD system. EX1073 at [0043] further corroborates that it was known that VOD systems had “pre-encrypt[ed]” content and “**stores** the content for forwarding...when authorized” for delivery to a storage device. *See also* EX1073, [0098] (“The encoded file is encrypted and verified prior to writing the encrypted material to VOD content server 302.... A streaming mode of pre-encryption is also possible in which content is ‘streamed’ from a source of raw content such as a video (possibly analog tape) through an MPEG encoder, sent to the OLES to **perform encryption and finally stored on a VOD server.**”), FIG. 3 (annotated below).



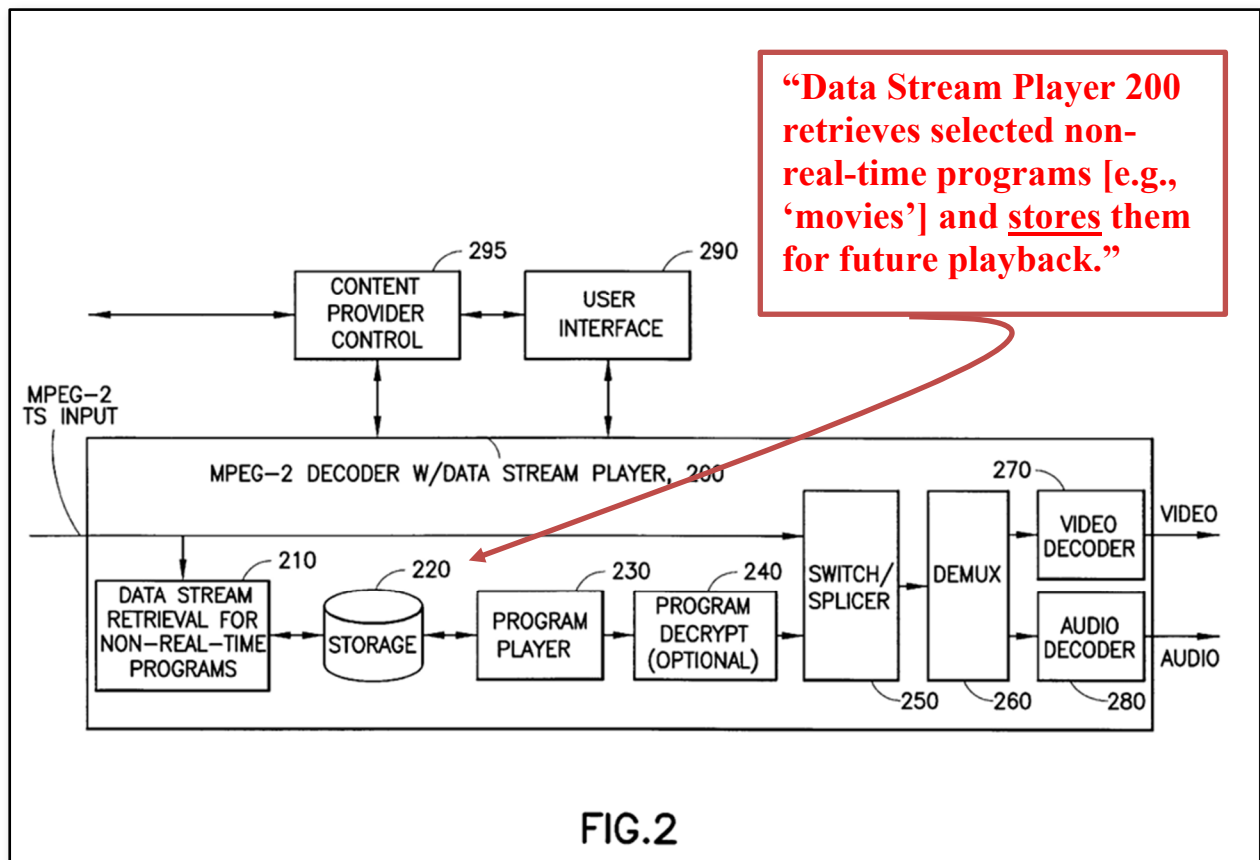
81. As another example I cite in response to SFI's arguments and Dr. Rubin's opinions, EX1097 at 2:31-53 describes as background knowledge that using "real-time encryption" in VOD system is problematic and "poses much greater cost and space issues." Thus, rather than use real-time encryption, it was known to implement content-delivery systems (like Foster and Hayhurst) that "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." EX1097, 2:31-53, FIG. 1

(annotated below). That is, the content is encrypted and **stored** at the “centralized facility” so it could be delivered (again and again if popular content). Thus, EX1097 is also consistent with my opinion in my First Declaration that POSAs would have wanted to pursue Foster-Hayhurst for the various benefits that POSAs would have recognized, such as improving the aggregate speed of content delivery and fault tolerance. First Declaration, ¶¶182-185.

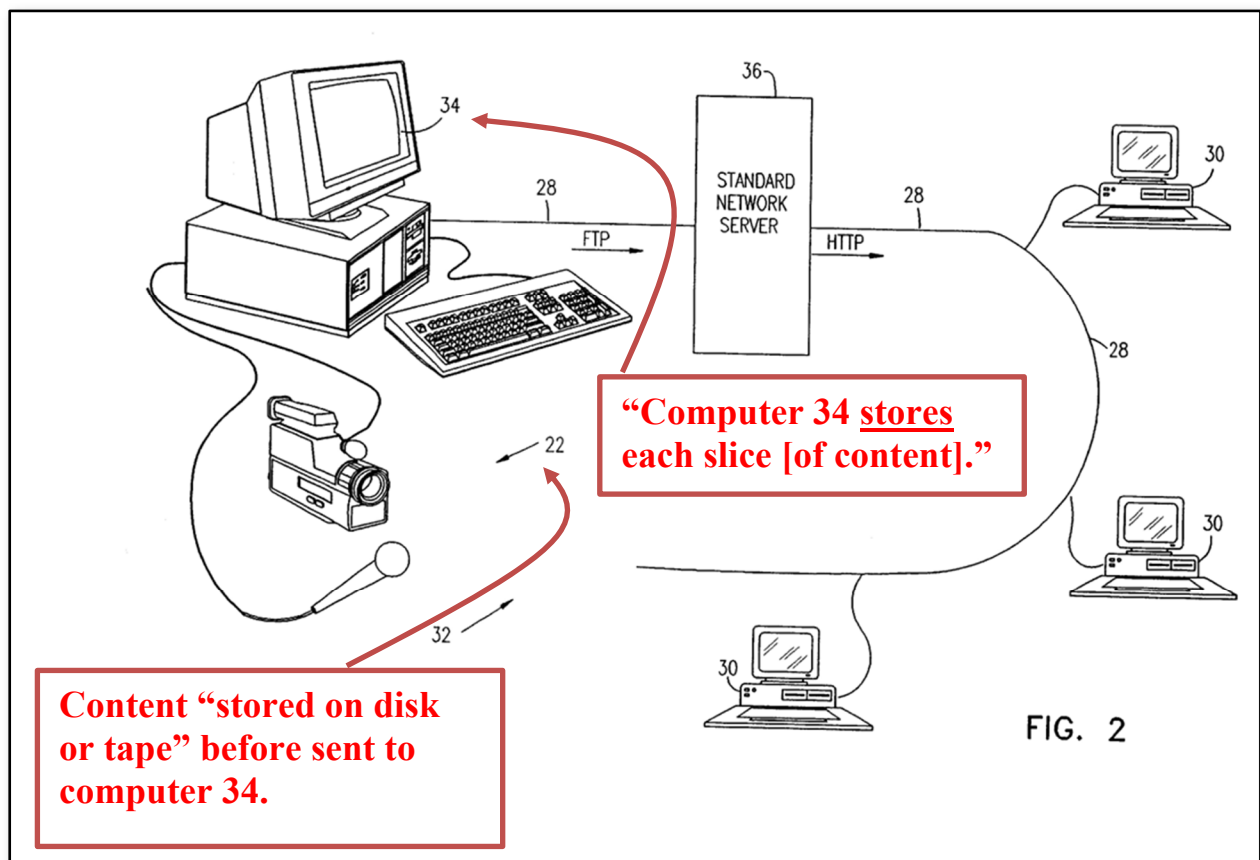


82. As another example I cite in response to SFI’s arguments and Dr. Rubin’s opinions is EX1075. EX1075 at 1:51-67 teaches that it was known to implement the “MPEG-2 transport” stream (discussed in Foster) in a “system [that] should allow the programs and other content to be provided **on a non-real-time**

**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." EX1075 at 1:51-67 further confirms that it was desirable to have "the delivery of **pre-encrypted content ... in non-real-time.**" *See also* EX1075, FIG. 2 (annotated below). Thus, EX1075 further corroborates my opinion that POSAs not only understood Foster's program stream to be beneficially and primarily directed to storage, but that it was beneficial to **non-real-time** storage and delivery of encrypted content (i.e., **not** live streaming).



83. As another example I cite in response to SFI's arguments and Dr. Rubin's opinions is EX1076. EX1076 at 9:53-10:5 discloses a prior art system where even "Broadcast data are [] input to the computer, for example, from input devices 22, or from a video, audio or animation sequence *stored on disk or tape*." That is, even "live" broadcasting of content is rarely real-time, but instead often retrieved from a *stored device* that stores the content for some later use.

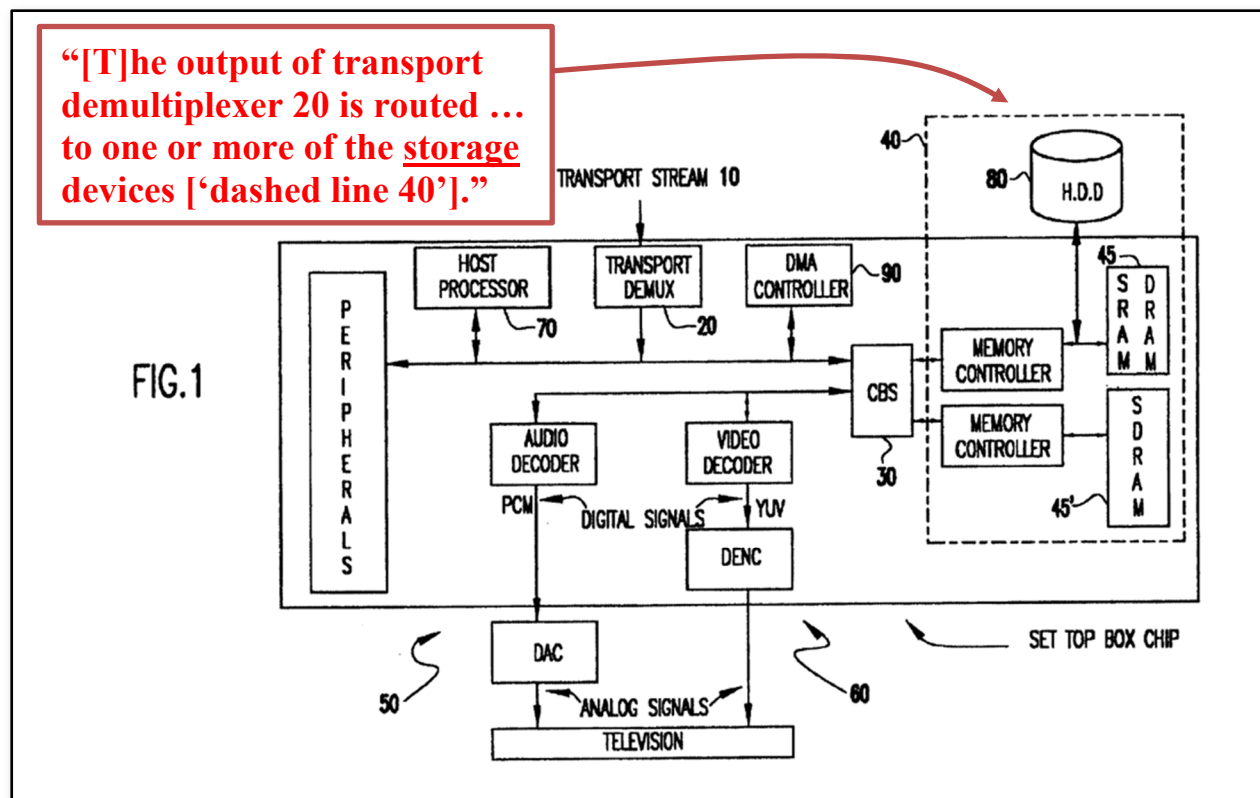


84. As another example I cite in response to SFI's arguments and Dr. Rubin's opinions is EX1077 (which is another patent by several of the Foster inventors). EX1077 discloses that it was known in the prior art to have "pay-per-

view movies” that were stored for viewing “at a time convenient” to the user.

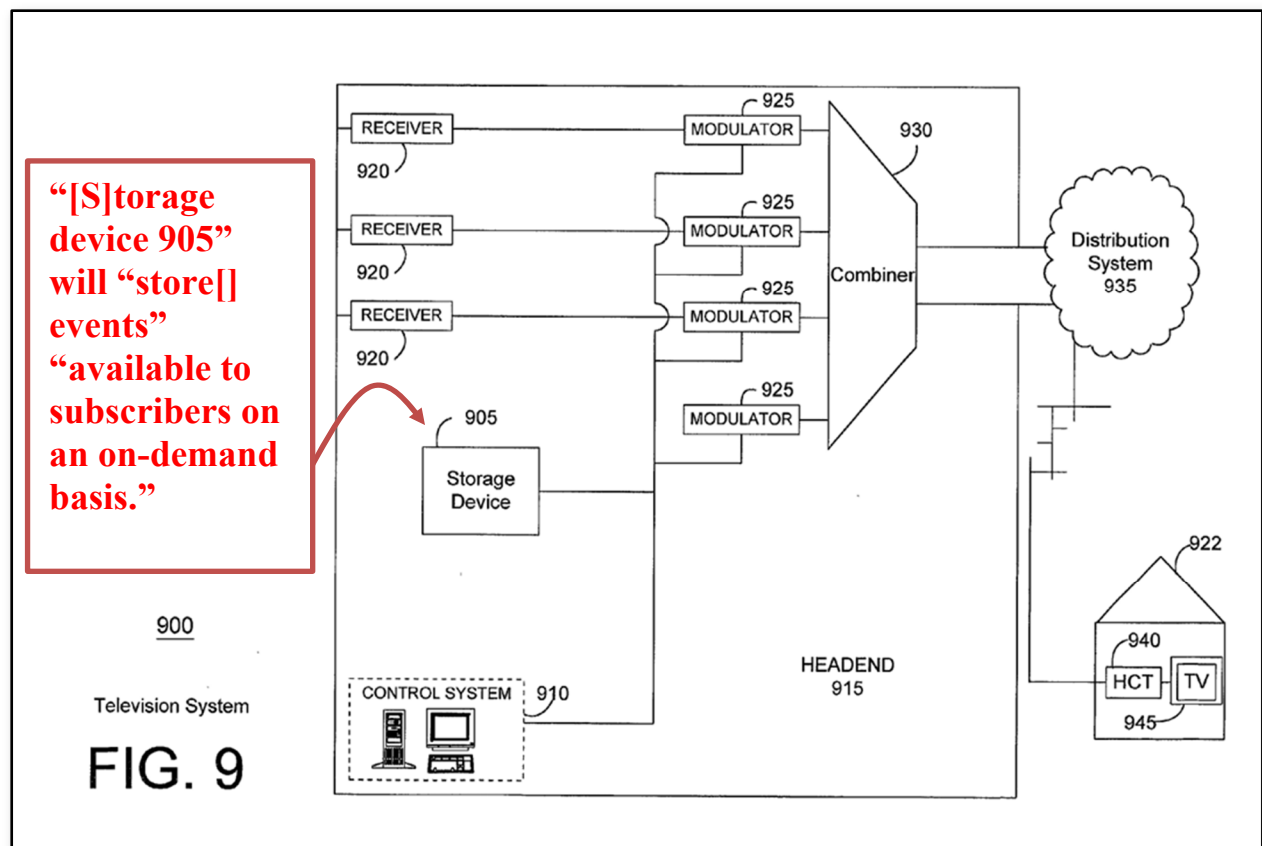
EX1077, 1:12-21. In those system, users typically had a “set-top box (STB)” (as in Foster, [0040], [0052]), that was expected to have “substantial storage within the STB” and/or paired with “further storage on an onboard mass storage device such as a hard disk drive or compact disk recorder/playback device.” EX1077, 1:48-65.

EX1077 at 3:56-5:8 goes on to describe processing of an MPEG “packetized elementary stream” (as in Foster) for “stor[ing] to mass storage.”



85. In response to SFI’s arguments and Dr. Rubin’s opinions that Foster’s CAS implementation would only be for use in streaming live television, I cite EX1092. EX1092’s Figure 9 (annotated below) illustrates “a storage device 905,

such as a video-on-demand server.” EX1092, [0079]. “A control system 910 within the headend 915 is programmed to store all the streams included within popular events that are received from one of the receivers 920” (where the receivers are receiving content from content providers). That is, even in cable / satellite company systems, a conventional video-on-demand server for CAS (like those discussed in Foster) will have a processor that instructs storage devices to store content for later downloading and storage to another storage device.



Thus, it is clear from Foster’s disclosure (and reinforced POSAs’ background knowledge) that Foster’s technique was designed for use in CAS-based systems

that included the storage of content both before and after delivery of the encrypted content to end-user recipients (both for streaming and non-streaming applications).

86. I note that at his deposition, Dr. Rubin provided various testimony about the wide-spread use of CAS in the prior art, such as for use in the security of “[f]ile systems” and “video-on-demand” or “basically any place that you have permissions and you want to prevent people who aren’t authorized from getting to something.” Rubin-Depo., 85-86; *see also* Rubin-Depo., 87-89 (testifying that “content producers [] would license the content to the [VOD] hosting sites” and that it was known to have “service providers that would keep the content” that “would be stored,” for example, “[o]n a server”), 90-94 (testifying that VOD system included “intermediaries that would house the actual content” for distribution to end users). I agree that POSAs would have known that CAS was widely used including VOD / VOD-like systems that are discussed in Foster and Hayhurst.

87. Foster’s compatibility and suggestion of CAS-based and/or VOD-based system thus corroborates the opinions I expressed in my First Declaration that POSAs had motivation to combine Foster and Hayhurst because “[l]ike Foster, Hayhurst concerns ‘electronic media delivery, such as video-on-demand.’” First Declaration, ¶178 (quoting Hayhurst, [0002]). Indeed, Hayhurst explicitly says its technique is beneficial for “cable television (CATV) system[s]” and “a content

provider in **delivering video-on-demand.**” E.g., Hayhurst, [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.**”), [0005]-[0006] (“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 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.”).

88. Thus, as I explained in my First Declaration (¶¶182-192), POSAs had motivation to pursue the Foster-Hayhurst combination for purposes of storing encrypted content in Hayhurst distributed storage devices, including for streaming applications (e.g., streamed VOD) and non-streaming applications (e.g., VOD for downloading and storage).

**2. The Foster-Hayhurst combination discussed in my First Declaration is consistent with Foster’s goals.**

89. Based on SFI’s narrow interpretation of Foster’s teachings being limited to streaming of live broadcasts, SFI argues Hayhurst’s distributed-storage technique would “thwart Foster’s goals, make Foster inoperable for its intended purpose, and compromise any security benefits otherwise provided by Foster alone.” POR, 51-61. In my opinion, none of these three critiques would have had

merit to POSAs, let alone discouraged POSAs from pursuing the Foster-Hayhurst combination.

90. The first two critiques – based on Foster’s alleged “goals” and “intended purpose” – depend on narrowly interpreting Foster as being concerned **only** with transmission of synchronized data for real-time video streaming purposes, *e.g.*, live video of a sporting event. POR, 52-58 (“Foster seeks to securely stream content, *e.g.*, live television, without delay...Foster’s system is able to deliver live or real-time content...”). However, as I discussed above (§II.B.1), Foster (like Hayhurst) is also concerned with **storage** of data for later/repeated playback. *See also, e.g.*, Hayhurst, [0002]-[0006].

91. SFI argues that POSAs would have been discouraged from the potential for excessive delays in delivering content to users in the Foster-Hayhurst system because a movie file would purportedly be distributed amongst “thousands” of recipients. POR, 52-55. In my opinion, POSAs would disagree with SFI’s argument for multiple reasons.

92. First, SFI ignores my opinion, supported by multiple corroborating references, that POASs knew distributed distribution systems like Hayhurst’s would have **reduced** delays in Foster’s system because it would have increased the “aggregate speed” and the “efficiency/intelligence” of content delivery. First Declaration, ¶¶183-184 (citing EX1062, EX1037, EX1043, EX1050). For

example, in Foster alone, content is delivered from one server, e.g., at the content service provider's system. Foster, [0005] (In DRMS, "content is packaged and loaded onto a web server."), [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."), [0052] ("[A]s depicted in FIG. 2, content 22 is provided to a head-end server 24 and encrypted using a key.").

93. Thus, as I explained in my First Declaration (with corroboration), that could sometimes create a potential bottleneck (e.g., if the server is overloaded or its network connection (or the server itself) fails). First Declaration, ¶184.

94. In contrast, in Foster-Hayhurst, based on Hayhurst's teachings, the content service provider will no longer "require[] a[n] extensive amount of bandwidth" (Hayhurst, [0004]) because the overall data set is delivered from multiple storage devices (each effectively a server of the content that it stores). Thus, as I explained in my First Declaration (with corroboration), POSAs understood that a delivery network like Hayhurst would, on average, allow for *faster* delivery of content to a user. First Declaration, ¶184 (citing, e.g., EX1037, EX1062). For example, EX1037 at [0019]-[0025] (cited First Declaration, ¶184), notes benefit of "transfer[ring] by downloading data from multiple sources or

channels in parallel” (as in Hayhurst). One of those benefits is “**Speed**” (optimizing to “download the data as absolutely fast as possible”) and “**Intelligence**” (“download[ing] from the most desirable mirrors or sources”). EX1037, [0019]-[0020].

95. Again, I note that the ’854 patent itself recognizes (as POSAs already knew) 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, 69:56-58.

96. Thus, Hayhurst’s technique would not only **not** add excessive delays, as SFI alleges, but POSAs would have reasonably expected Foster-Hayhurst to **reduce** delays over Foster alone. Similarly, by having redundant copies of encrypted segments as Hayhurst teaches, it would also be reasonably expected to **reduce** delays overall (and provided added protection against corrupted and/or lost data) because the system could intelligently determine which storage devices were better suited to deliver the content (e.g., selecting storage devices that were closest to the end-user and/or less busy).

97. In his declaration, Dr. Rubin opines that “there is no indication that Foster’s system suffers from the bandwidth constraints confronted by Hayhurst’s video-on-demand system.” Rubin-Decl., ¶140. Even if that were true, I disagree that that would have dissuaded POSAs from pursuing the Foster-Hayhurst

combination. For one, Hayhurst teaches that bandwidth constraints were a concern for content service providers (Hayhurst, [0004]-[0005]), whether or not Foster itself recognized it. Limited bandwidth would have also been recognized as a universal consideration for any content storage and/or content delivery system. Indeed, as I discuss below, at deposition, Dr. Rubin recognized that the prior art comprised content delivery networks. The very purpose of which was to address, among other considerations, bandwidth limitations of a content service provider: as the number of users in the system increases, and/or the size of the data being stored and delivered increases, and/or the speed by which you want to stored and/or retrieve stored content, the available bandwidth can become a limiting factor that POSAs would have known was a consideration in the design of systems like Foster's. For example, beyond Hayhurst, EX1037 similarly recognizes as background information that "for many applications and file transfers taking many hours [e.g., for video files], even on a broadband connection, there exists **a need to optimize the delivery of such** files and make that delivery **robust in the face of network changing** conditions." EX1037, [0004]. EX1037's solution to that known need and problem is similar to Hayhurst's: "transferring data over multiple channels in parallel, possibly from multiple servers." EX1037, [0011].

98. Moreover, Foster does not discourage the use of techniques that reduce bandwidth considerations for content service providers, and SFI and Dr.

Rubin did not identify such disclosure in Foster. Foster also does not expressly identify any concerns with the potential for power failures and the need for data backup, but Foster did not need to identify those concerns because those were considerations well-known to POSAs in designing systems like Foster's. Thus, even if it were true that Foster itself did not recognize a need to improve bandwidth constraints at a content service provider, POSAs would have recognized that that would have been a potential concern that would have been addressed by the Foster-Hayhurst combination.

99. Dr. Rubin offered testimony at his deposition that is consistent with my opinion that POSAs would have recognized that Hayhurst would have provided benefits to Foster (including in terms of speed and intelligence). For example, Dr. Rubin testified about prior art content delivery networks, where intermediaries would store content on servers that “would be closer to the consumer than the servers of either the service provider or the content owner” and would delivery content over networks using intelligence, such as by having “servers based on population density.” Rubin-Depo., 89-93. That testimony is consistent with the testimony I offered in my First Declaration (with corroborating evidence) that POSAs had motivation for Foster-Hayhurst because it would have beneficially allowed for “efficiency / ‘intelligence’” in the delivery of Foster's encrypted content, such as “by being able to select storage device(s) that would provide the

best delivery, based on availability, **location**, etc., [which] would have improved the reliability and speed of retrieval.” First Declaration, ¶184 (citing EX1037).

100. Second, in my opinion POSAs would disagree with the premise of SFI’s argument that Foster-Hayhurst necessarily has “thousands” of users delivering any single piece of content. POR, 40, 48. Hayhurst says a cable system “**may**” have “thousands of subscribers” (Hayhurst, [0016]), but my opinions regarding the Foster-Hayhurst combination did not require thousands of storage devices because the challenged claims only require a plurality of storage devices, not thousands. Indeed, Hayhurst’s exemplary embodiment has only **four** storage nodes.

101. Moreover, Hayhurst also says “not all of the subscriber units” must participate “at all times” in storing/distributing data, which teaches that even when content is stored at end-users, fewer than all the users (i.e., fewer than “thousands”) can store chunks of any given dataset. Hayhurst, [0023]. The number of users that would store *data chunks* of any given *dataset* would be a matter of design choice, balancing the benefits of having more storage devices (e.g., additional security) against considerations like the complexity of managing storage and delivery of each data set increases as the number of storage devices increases. For example, even if Foster-Hayhurst had a million storage devices, POSAs would have recognized that there would have not been a need to distribute a single data set to

one million storage devices. In the same way, POSAs would have recognized that even in Foster-Hayhurst systems with “thousands” of storage devices, there would have been reasons to distribute each data set to far fewer than a thousand storage devices (under some design considerations, a single-digit number of storage devices could be sufficient and/or beneficial).

102. Regardless, even in implementations of Foster-Hayhurst that were using “thousands” of storage devices, POSAs would have still recognized the combination had the same benefits over Foster alone that I identified in my First Declaration (e.g., using smaller segments across more storage devices increased security and flexibility in the intelligent delivery of the content).

103. Furthermore, Hayhurst says rather than use a cable network or even a wide-area network for the storage and delivery of encrypted content, Hayhurst explicitly teaches that “any other communications network,” like “a local area network (LAN),” could be used. Hayhurst, [0016]. POSAs knew that LANs typically had far fewer than thousands of nodes. In other words, Hayhurst is teaching that its storage and delivery system is beneficial even in systems that distribute the encrypted data segments to only a handful of storage devices (e.g., as a conventional tradeoff in speed and complexity).

104. In addition to the evidence noted above that I cited in my First Declaration (§184), in response to SFI’s arguments and Dr. Rubin’s opinions that

POASs would not have pursued Foster-Hayhurst, my opinion about Foster-Hayhurst increasing the speed of delivery over Foster alone is further corroborated by EX1078. EX1078 discloses that peer-to-peer like delivery network will reduce the load on the server that originates the content. EX1078 says that its “present invention is

to improve an information delivery rate on a network such as Internet, reduce load occurring in a server and the network and shorten a data delivery path, by **enabling information to be delivered in parallel by means of use of a distributed computing technology** utilizing idle resources including CPUs and hard discs of general user computers (hereinafter, referred to as ‘peers’) such as PCs and a bandwidth of the network, and a peer-to-peer technology for enabling data to be delivered between general user computers without passage through the central server.

EX1078, 4:5-12.

105. EX1078 further teaches using a peer-to-peer delivery system:

to enable clients to make **use of rapid delivery, to reduce load** in a central server and a network and **to shorten a data delivery path** by means of delivery of data from adjacent peers rather than the central server, **by dividing information to be serviced and redundantly storing the divided data in a plurality of peers** on behalf of the central server for entirely storing and delivering the information to be serviced, searching peers storing the divided data corresponding to information requested by the clients and **selecting peers capable of providing the**

**most rapid delivery rate** when the clients request the information, and **causing the data to be delivered in parallel to the clients requesting the information.”**

EX1078, 4:13-28.

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 pursuing 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 difference 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.)

111. In response to SFI's POR and Dr. Rubin's opinions regarding updating KEKs in distributed networks, my opinion regarding POSAs' background knowledge that KEKs could be changed and/or updated even when distributed to multiple storage device is corroborated by the following references.

112. First, I note that I am co-author on several prior art, peer-reviewed journal articles that were intended for POSAs in the field of securing data that addressed techniques for updating encryption keys in large and/or distributed systems. Below is a list of several, peer-reviewed articles and/or papers relating to presentations given that relate to key management with large userbases and the benefits thereof. Each of these papers were published and publicly available in the prior art that I was a co-author on that are in the field of securing data and which POSAs would have been familiar with:

- Suman Banerjee and Bobby Bhattacharjee. “Scalable Group Communication over IP Multicast”, In Proceedings of the International Conference on Network Protocols, pages 261–271, 2001.
- Suman Banerjee and Samrat Bhattacharjee. “Scalable Secure Group Communications over IP-multicast”, IEEE Journal of Selected Areas in Communications, JSAC, Vol. 20, No. 8, pages 1511 - 1527, October 2002.

113. In response to SFI’s POR and Dr. Rubin’s opinions about the alleged complexity in changing a KEK across multiple devices, I further cite, EX1079, which is a 2003 IEEE article entitled “Key Management and Distribution for Secure Multimedia Multicast.”<sup>3</sup> EX1079 describes using “[m]ulticast communications” because it “is efficient since it reduces the demands on the network and bandwidth resources.” EX1079, 544 (identifying potential applications in “pay-per-view” and “interactive television”). EX1079 (at 546)

---

<sup>3</sup> IEEE Xplore’s website lists EX1079 as having been published on “31 December 2003” in “IEEE Transactions on Multimedia (Volume: 5, Issue: 4, December 2003).” See <https://ieeexplore.ieee.org/document/1244812>. Based on the above evidence and my review of the indicia appearing on EX1079 and my understanding of the IEEE and IEEE Xplore, it is my opinion that EX1079 was publicly accessible in the prior art by being disseminated or otherwise made available to POSAs exercising reasonable diligence by 2004.

noted that “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.**” EX1079 (at 544-546) then goes on to provide a prior art procedure for updating / changing KEKs that have been distributed to multiple users, which corroborates my opinion that POSAs were aware of techniques for updating KEKs across many devices.

114. In response to SFI’s POR and Dr. Rubin’s opinions about the alleged complexity in changing a KEK across multiple devices, I cite EX10180. EX1080 is another journal article from 2003 obtainable from IEEE Xplore, entitled “Pushing the internet direct to the user - security issues.”<sup>4</sup> It describes a system where “[d]ata is encrypted at the source by the TEK” and the system “distribute[s]

---

<sup>4</sup> IEEE Xplore’s website lists EX1080 as having been added to IEEE Xplore on “24 November 2003” and published in “IEE Seminar on IP Over Satellite - The next Generation: MPLS, VPN and DRM Delivered Services, 2003.” *See* <https://ieeexplore.ieee.org/abstract/document/1246598>. Based on the above evidence and my review of the indicia appearing on EX1080 and my understanding of the IEEE and IEEE Xplore, it is my opinion that EX1080 was publicly accessible in the prior art by being disseminated or otherwise made available to POSAs exercising reasonable diligence by 2004.

the key to each receiver,” and notes that the TEK “must itself be encrypted” using a KEK. EX1080, 126. EX1080 further notes that “[i]n multicast sessions users who leave a session and are no longer entitled to take part force the changing of the TEK, and a multicast with a flat Key Management system and a large number of subscribers” there can be “a huge **rekey problem**,” but techniques for addressing that problem already had solutions that were known to POSAs, such as “described in a number of IETF MSEC Internet Drafts.” EX1080, 126-127. Thus, EX1080 further corroborates my opinion that POASs would not have been dissuaded from Foster-Hayhurst because of a purported challenge in changing the KEK(s).

115. As to SFI’s argument that Foster-Hayhurst would make the KEK less secure (POR, 58-61), it is my opinion that POSAs would disagree. POSAs would disagree with SFI’s argument (at least) because Foster-Hayhurst would not have had to give every device a copy of a KEK and I did not opine otherwise in my First Declaration. Rather, as in Foster alone, Foster-Hayhurst would only give a user access to a KEK after that user has been authorized to receive and decrypt the encrypted content. Foster, [0052] (“Upon receiving encrypted content, **if the entitlement management message indicates that a subscriber is authorized** to access a certain channel, [then] the control words in the entitlement control message **will be converted into a key** that is passed to descrambler 36.”), [0055] (“If, however, the DVD player was a known non-compliant or circumvention

device, its internal device keys **would be unable to recover the correct key encrypting key 28**, because the KMB would have been modified to exclude the device”). *See* First Declaration, ¶188.

116. Similarly, in Foster-Hayhurst, the plurality of storage devices would not each store a KEK; instead, the KEK would only be available to the end-user(s) that have been authorized to receive and decrypt an entire data set. For example, based on Hayhurst’s teachings, the “encryption key to unlock the movie” (the KEK in Foster-Hayhurst) would have been stored on the “main server,” not at each subscriber. Hayhurst, [0007], [0021] (cited First Declaration, ¶188). As I explicitly opined in my First Declaration (¶188), POSA would have found it obvious “[i]n the Foster-Hayhurst combination, [that] the KEK **would be maintained in a storage device (e.g., Foster’s database 426)** until required by a subscriber....consistent with Foster’s providing the KEK to authorized recipients.” In Foster-Hayhurst, Foster’s database is part of the main server coordinating the storage and delivery of content by the storage devices.

117. That obvious implementation of Foster-Hayhurst is also consistent with Foster’s teachings to decouple the transferring of information needed to obtain the KEK from the transferring of the encrypted data and encryption keys. *See, e.g.*, Foster, [0055] (“It should be understood that key encrypting key 28 is generally determined based on a key management block (KMB), which is typically

delivered to a recipient concurrently with or prior to the delivery of encrypted content 26....”), [0056] (“the KMB should be delivered synchronously with, or prior to, the encrypted content”).

118. Finally, regardless of where the KEK is stored in Foster-Hayhurst, I also disagree with SFI that POSAs would have found Foster-Hayhurst to be less secure than Foster alone. POR, 58. Rather, as I explained in my First Declaration (with corroborating evidence, including from Hayhurst’s own disclosure), POSA would have found that Foster-Hayhurst would have **increased** the overall security of Foster’s system. First Declaration, ¶184. Again, Hayhurst explicitly says it “provides a **high level of security**” via its distributed storage technique. Hayhurst, [0008]. *See supra* §II.A, II.B.1.

119. In my opinion, POSAs would have agreed that Foster-Hayhurst was more secure than Foster alone in several ways. As I discussed in my First Declaration (¶184) and above (§II.B.1), Hayhurst would have **reduced** the risk of unauthorized access to content compared to Foster because an attacker of Foster-Hayhurst would have had to collect multiple encrypted *data chunks* from multiple storage devices, rather than from one storage device, as in Foster alone. Hayhurst further teaches that security is increased by its distributed-storage technique because an attacker would also have to know how to reassemble the Foster-Hayhurst *data chunks*. Security is added because by Hayhurst’s technique “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.” Hayhurst, [0021].

120. Moreover, even if SFI were arguing (or at suggesting) that Foster taught distributing access to its KEK to end users even before users were authorized to access the encrypted content, Foster-Hayhurst would not have done that. That is because (as I discuss above) Hayhurst teaches to keep the KEK at the main server that coordinates the storage devices, rather than at one of the distributed storage devices. As I discussed in my First Declaration (§§188-189), Foster-Hayhurst would similarly use a main server to coordinate distribution of content to and from the storage devices and stored the KEK at a server, rather than at the user devices.

121. But even if Foster-Hayhurst were less secure than Foster in some of the ways SFI asserts, in my opinion POSAs still would have been motivated to pursue Foster-Hayhurst for the reasons I identified in my First Declaration. That is 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) **and** additional, beneficial non-security feature that Foster lacks.

122. 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, and reducing bandwidth requirements, as I discussed in my First Declaration (§184) and above (§II.B.1). Those benefits would have motivated POSAs to pursue Foster-Hayhurst, with a reasonable expectation of success, even if it were true that it would have added complexity in managing and distributing a KEK. 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.

EX1081, 15:6-25.

123. As I discussed above, I note that at deposition, Dr. Rubin testified that it was known in the prior art that content owners would use intermediaries like content providers and content delivery networks (working with content providers) to improve the delivery of content “because they could optimize for sharing content in many different locations” such as having more servers in higher “population density” locations. Rubin-Depo., 90-93. Dr. Rubin further testified that by having storage devices that are “remote from each other” and storing different portions of data “a physical compromise of the system would only yield

the portion that's there [i.e., at the storage device that was compromised]." Rubin-Depo, 134-137 (further testifying that "redundancy" could be helpful "[i]n the event that there would be problems at some of the sites"). All of that testimony is consistent with my opinions in my First Declaration identifying several benefits POSAs would have recognized from Foster-Hayhurst, compared to Foster alone, which I corroborated by several prior art references.

124. That deposition testimony also directly contradicts Dr. Rubin's declaration testimony that "Foster's system has no need for redundancy as Foster's system can simply resend dropped packets" (EX2033, ¶140): POSAs understood that the one of the reasons why redundancy would have been beneficial to Foster is that it would have provided a way to store and/or deliver content in more robust ways than would have been possible when the single source of content in Foster – what Dr. Rubin called Foster's "one-on-one transmission" (EX2033, ¶140) – failed. POSAs would have recognized that designs with a single point for storing and/or delivering data could be fragile, *e.g.*, due to component and/or network disruptions. Unlike Foster alone, Foster-Hayhurst would have been obvious to POSAs as a more robust solution that was less susceptible to those problems because it provided redundant sources for storing and delivering data.

125. As with Dr. Rubin's testimony alleging that Foster itself did not express a need for the benefits that Hayhurst's technique would have provided,

POSAs knew that systems like Foster’s were not designed for operation under the assumption that optimal conditions would always persist. Rather, POSAs designed those system to account for the real-world where problems could (and often did) arise, such as when bandwidth becomes limited (e.g., at times of high demand, like holiday periods), where data storage and/or delivery systems failed, where data could be lost or subject to attack if maintained in a single location and without backup, etc. Indeed, at deposition, Dr. Rubin admitted that POSAs knew that redundancy was helpful when “one of the sites went down, like a power failure.” Rubin-Depo., 134-137.

126. Dr. Rubin opines in his declaration that “Foster’s system has no ... need for efficiency/intelligence when sending on a one-to-one basis.” EX2033, ¶140. As an initial matter, I am unsure exactly what Dr. Rubin meant by “when sending on a one-to-one basis.” By prefacing it with “when,” Dr. Rubin appears to admit that there may be cases **when** a one-to-one basis is not used (or at least it was known to have systems that were not on a one-to-one basis). In any event, Foster only discusses an example system where content is delivered from one server – e.g., Foster’s media server – to recipients. *See, e.g.*, Foster, FIG. 1. That appears to be what Dr. Rubin means by a “one-to-one basis.” As I demonstrated with corroboration in my First Declaration, POSAs would have recognized that having **more than one** source to deliver content (i.e., what Hayhurst teaches),

allows for several benefits in terms of speed, intelligence, and efficiency (in addition to security and fault tolerance benefits) that are not possible with a single source for the delivery of content. First Declaration, ¶184 (citing Hayhurst itself, EX1062, EX1037, EX1043, EX1050).

127. As just one of the examples I identified in detail in my First Declaration, by having multiple sources of storage devices storing and delivering the encrypted content, Foster-Hayhurst's system could select the storage devices that were best equipped to delivery content to a particular user compared to other storage devices in the network **and** that, in many cases, would have been better than Foster's system having only one server of the content. For example, consider a system where one recipient was in New York and another recipient for the same content was in California. In Foster alone, both recipients would be served by the same media server that might be equidistant from both recipients or (more likely) much closer to one recipient over the other. In Foster-Hayhurst, storage devices in (or closer to) New York could deliver the content to the New York recipient, whereas storage device in (or closer to) California could deliver the content to the California recipient. That is a form of intelligence and efficiency that I identified in my First Declaration, with support from the prior art.

128. I note too that at deposition Dr. Rubin recognized that POSAs were aware of situations similar to the one I just described. Dr. Rubin testified that so-

called content delivery networks were known and were used by content owners and/or content service providers to deliver content “because they could optimize for sharing content with customers in many different locations.” Rubin-Depo., 91. For example, he testified that “instead of AT&T having to build warehouses all over the place where their customers lives, they could rely on the CDN,” who would “have servers based on population density” so “they could service [a] city **optimally**.” Rubin-Depo., 89-92. That is precisely an example of efficient / intelligent storage and delivery of content that POSAs would have recognized that Foster-Hayhurst provided that Foster’s one-to-one delivery could not.

129. In summary, it is my opinion that POSAs would have disagreed with SFI’s argument that POSAs would not have had reason to pursue Foster-Hayhurst with a reasonable expectation of success. That is because SFI’s arguments are largely (if not exclusively) premised on incorrectly limiting Foster to streaming of live broadcasts. SFI’s argument fails because POSAs would have recognized that Hayhurst would have **improved** Foster in several ways (including in terms of security but also intelligence, speed, redundancy, etc.). SFI’s argument also fails because even if Foster-Hayhurst would have led to any decrease in security over Foster alone (which POSAs would have disagreed with), POSAs would have recognized that Hayhurst would have provided other security and non-security benefits that Foster by itself did not achieve.

**C. Response to Argument that Foster’s Encrypted Payloads Are Not “*Data Chunks*”**

130. Claim 1 requires “*distributing [a] data set into a plurality of shares*” and then encrypting those shares, with an encrypted share called a *data chunk*. SFI acknowledges the Petition (and similarly my First Declaration) “map[ped] the claimed ‘plurality of shares’ to Foster’s” “MPEG packet payloads, [*i.e.*, content units],” the “‘data chunks to Foster’s encrypted content units..., and” “the ‘data set’ to Foster’s content.” POR, 63-65 (citing, *inter alia*, Petition, 72-74); see First Declaration, ¶¶201-209.

131. SFI does not dispute that Foster’s content meets the claimed *data set*. Rather, in its POR, SFI argues that Foster’s MPEG payloads (which comprise Foster’s content units) are not *shares* and that Foster’s encrypted content units are therefore not *data chunks*. POR, 63. According to SFI, a payload of a *data set* processed consistent with (or in ways like) the MPEG standard cannot be a *share*, and thus an encrypted MPEG payload cannot be a *data chunk*. In my opinion, as I explain further below, POSAs would not agree with SFI’s arguments and that there is nothing in the intrinsic record of SFI’s patent to support SFI’s narrow interpretations of *share* and *data chunk* to exclude Foster’s packets of payloads split off from a common *data set* (e.g., movie file).

132. SFI argues there was no explanation of “what ‘data chunks’ or ‘data shares’ are or why [Foster’s] packet ‘payloads’ are ‘data shares’ (or ‘data chunks’ when encrypted).” POR, 66; POPR, 42.

133. As an initial matter, I note that neither party has construed *data chunk* or *share*. In my opinion, the plain language of claim 1 provides that a *data set* is the set of data to be securely stored, a *share* is merely a portion (or part) of that *data set*, and a *data chunk* is an encrypted *share*. The claim’s meaning is apparent on its face—a *share* is just a collection of data that is some amount less than the entire *data set*. Foster’s payloads (content units) are exactly that, as each payload is a collection of data less than the complete set of data for Foster’s content.

134. In my opinion, POSAs would understand that the patent’s specification places no limit on the relationship between a *data set* and a *share*. Instead, the specification has multiple disclosures that inform POSAs that the pieces of data that are split from the data set (i.e., *shares* in the ’854 patent’s claims) can be of any size of data from the data set or can each be the same size or different sizes of data from the data set.

135. For example, the patent says POSAs “will recognize from the disclosure herein, a wide number of alternatives for the data splitting process,” and that “data may be split **into any desired, selected, predetermined, or randomly assigned size unit.**” EX1001, 21:9-25. The patent similarly says that a “data unit

sizes **may be...all of the same size**, [or] a combination of sizes....” EX1001, 21:30-34.

136. Elsewhere, the patent says “[i]t 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.” EX1001, 21:27-31. Similarly, the patent says “[i]t will be understood that any other suitable parsing and splitting approach may be used to generate portions of data from an original data set in accordance with the present invention.” EX1001, 74:56-75:10.

137. Thus, the patent explicitly (and repeatedly) allows for *shares* to take any form or size of data so long as they are a subset of data split from the *data set*.

138. At deposition, Dr. Rubin testified that in his opinion “the share is the thing that something gets split into” (Rubin-Depo., 17) and that the terms share and chunk “refer[] to the pieces that have been split from the original data set” (Rubin-Depo., 157-158). POSAs would have recognized that Foster’s payloads (content units) meet Dr. Rubin’s view of what the *shares* / *chunks* are in SFI’s patent because each of Foster’s payload’s data is a split portion of the complete *data set* being secured by Foster.

139. The opinions I offered in my First Declaration (¶¶201-209) and discuss further below in response to SFI’s arguments, demonstrate that Foster’s payloads in its content units are the data that Foster’s data set is split into.

140. In its POR, SFI suggests a *share* must be cryptographically generated, e.g., the originally data set must be randomly parsed/split, thereby making the data within the “*share*” less susceptible to “attack.” POR, 68-69 (quoting EX2033, Rubin-Decl., ¶¶149-151 (citing EX1001, 56:47-61)). If SFI is arguing that the claimed *shares* must be randomly split from the *data set* or cryptographically split from the *data set*, then in my opinion POSAs would find that that construction is inconsistent with the plain meaning of the claims and the patent’s specification.

141. As I discussed above in this section, the plain meaning of claim 1 does not restrict the manner by which *shares* are generated from the *data set* beyond reciting that each *share* that is split from the *data set* must comprise less than the entirety of the *data set*.

142. Similarly, the patent’s specification says “data units may be distributed into one or more shares” in several different ways, including simply “**fixed** or predetermined data unit size...**or** a randomly generated data unit size,” as an **alternative**. EX1001, 21:33-37. That is, randomly generated data unit size is an alternative **option** to fixed sizes but it is not required. Instead, as shown in this quoted language (and other language I quoted above), the patent is explicit that each *share* can be a “fixed or predetermined data unit size.” That means, contrary to SFI’s arguments, the patent does **not** require “splitting the data into shares” to

“help[] secure the data.” POR, 69. Instead, the patent discloses that the sizing of the splitting can be done to provide no additional security.

143. (I note, however, that one of my obviousness opinions for Ground 2 – Orsini – demonstrated that it would have been obvious to cryptographically adjust the sizes of Orsini’s *data chunks* to further secure the data set based on an encryption key that defined the *chunk* sizes. First Declaration, ¶¶126-129 (citing Orsini, [0281] and additional corroboration)); *see also* Orsini, [0292] (which include the same disclosure as EX1001, 56:47-61).

144. The dependent claims of the ’854 patent are further evidence that SFI is incorrect in its attempt to narrow the scope of claim 1’s *shares* to provide protection by being cryptographically split from the *data set* (as opposed to being some portion split from the *data set* that is less than the entirety of the *data set*). Dependent claims 4-6 limit claim 1’s *shares* to ones that are generated using cryptographic techniques like a “random technique.” I understand that a dependent claim is narrower than the independent claim from which it depends. Thus, the fact dependent claims from claim 1 require cryptographic techniques for generating the *shares*, but claim 1 does not have any such limitation about the splitting in and of itself providing security, further corroborates my opinion that POSAs understood the independent claim is broader and is not limited to

generating *shares* using particular techniques that might further increase security of the *shares*.

145. In its reply, SFI does not dispute that Foster’s payloads (content units) each comprise less than all of the data in Foster’s content, *i.e.*, *data set*. In my opinion, POSAs understand that that is all that the claims require of the *shares*. The fact that “Foster never refers to its packet payloads as ‘data chunks’ or ‘shares’” (POR, 66) is irrelevant because a POSA would understand that Foster’s payloads (content units) meet the claimed *shares* even if Foster did not label them “shares” or “data chunks.”

146. For (at least) the above reasons, POSA would disagree with SFI that “Foster’s packet payloads [or Hayhurst’s segments] exist not to protect data.” POR, 69 (quoting EX2033, ¶¶149-151). But even if that were true, it is my opinion that POSAs would have still understood Foster to include *data chunks*, for the reasons I identified in my First Declaration (¶¶201-209) and above.

147. That said, the POR at one point suggests that in the ’854 patent it is merely the act of “splitting the data set into shares [that] helps secure the data.” POR, 68-69. That is, SFI is saying that just by splitting the *data set* into *shares* security has been increased. If that is SFI’s argument, then Foster clearly provides that same security based on generating its payloads. That is at least because in Foster each share can be separately encrypted with its own encryption key, which

Foster says “provides **enhanced security** preventing an entire content stream from being accessed with a single key.” Foster, [0004]; Foster, [0071] (“In addition, because each content packet can be encrypted with a different title key, a party who illicitly obtains a title key cannot access the entire stream. In contrast, the pirated key is only good for decrypting the content packet to which is was attached.”). That supports my opinion that generating Foster’s packet payloads from Foster’s *data set* is **done** to protect data, not merely as a function of the MPEG standard, because Foster splits the data set into *shares* for the purpose of protecting the individual pieces of data and overall *data set*. Foster individually encrypts each *share* (packet payload) with an encryption key (potentially a different encryption key). First Declaration, ¶¶217-220 (citing Foster, [0064]-[0065], FIG. 8); Foster, [0071]. Contrary to SFI, Foster’s splitting and encryption is **not** “just a by-product of a pre-existing standard.” POR, 69. The MPEG standard, for example, does not require encrypting each of the packet payloads (let alone encrypting them with respective, different encryption keys as taught for one embodiment of Foster). It is Foster’s encryption technique, not a requirement of the MPEG standard, that encrypts each of the individual *shares* for the purpose of protecting the *data set*.

148. Furthermore, based on Hayhurst’s teachings, in the Foster-Hayhurst **combination**, the data set is split into smaller portions (*shares*) for purposes of

storing the encrypted segments on distributed storage devices. As I discussed in my First Declaration (§§221-225) and again above (§II.B), splitting Foster-Hayhurst’s *data set* was done for several benefits, including for the same reason SFI alleges is the reason why its patent “splits a data set into shares or chunks as a means of protecting a data set” (POR, 69): to address the problem when “data...remains whole and intact’ [and] it is ‘subject to attack’” (POR, 68, quoting EX1001, 56:47-61). *See supra* §II.A.2.c (discussing how Hayhurst is reasonably pertinent to this problem). That is, in Foster-Hayhurst, the splitting of the *data set* into *shares* (packet payloads) is done for the purposes of adding security by, among other things, separately storing the segments on geographically dispersed storage devices.

149. SFI’s POR identified several exhibits, several of which do not appear to be prior art, that allegedly teach that a “payload” or “packet” can be “divided into ‘chunks’ or ‘shares.’” POR, 66-67 (citing EX2012-EX2013, EX2041-EX2043, EX2036-EX2038). I have reviewed these exhibits. They generally concern generic “packets” (e.g., an IP packet), and are not about packets produced

according to the MPEG standard, let alone according to Foster's or Hayhurst's disclosures (or the combination thereof).<sup>5</sup>

150. In my opinion, the exhibits SFI cites are not inconsistent with my opinion that POSAs' would have understood that Foster-Hayhurst meets the claimed *shares* and *data chunks*. As I discussed above, neither the challenged claims nor the patent specification preclude a packet from either (i) being a single *share* (or single *chunk* when encrypted) or (ii) being divisible into multiple *shares* (and thus multiple *chunks*). The issue is whether the data being carried by that packet comprises data from the *data set* that has been generated from a *data set*

---

<sup>5</sup> I note too that Foster itself is not limited to using MPEG data, as "Content" is defined as "**any** data such as digital image, sound or binary data deliverable from a source to a recipient." Foster, [0037]. Thus, Foster (let alone the Foster-Hayhurst combination) would not be limited to dividing its *data set* into MPEG "packets" (or any other packet). Regardless, as I discuss in this section of my reply declaration, I do not agree and a POSA would not agree that a packet of MPEG-processed data cannot be a *share/data chunk*; and a POSA would find that Foster's processing in particular generates *shares/data chunks* for the reasons I discussed in my First Declaration (¶¶167-175, ¶¶210-212) and here in response to SFI's and Dr. Rubin's arguments.

and comprises less data than the entire *data set* (which the patent claims call a *share* of that *data set*). Foster’s packet payloads (and Hayhurst’s segments – which are needed to use Hayhurst’s segmented storage technique) are each less than an entire *data set*, as I demonstrated in my First Declaration (§§210-212) and discuss above in response to SFI’s POR and Dr. Rubin’s declaration.

151. Moreover, other literature demonstrates that POSAs knew that even something referred to as a “packet” could also be called a “chunk” (in the language of the ’854 claims a *share*) of a data set if (as in Foster, Hayhurst, and Foster-Hayhurst) the packet comprises a less-than-entire portion of the data set.

152. In response to SFI’s and Dr. Rubin’s argument regarding POSAs’ understanding of Foster’s content units, I cite EX1082, which unlike SFI’s exhibits **is** in the context MPEG-encoded data.<sup>6</sup> EX1082 says that with MPEG, each

---

<sup>6</sup> I note that a document sharing the same author, title, and publisher as EX1082 was cited on page 2 of U.S. Patent 7,898,951 (filed Aug. 6, 2004; issued Mar. 1, 2011). *See* <http://bit.ly/49OIc3o> (PDF of US7,898,951 on USPTO’s website). The ’951 patent lists as a reference titled “Mike Knee, ‘MPEG Video’” and indicates it was published “Mar. 2002” by “Snell & Wilcox,” and was available from <http://www.bretl.com/mpeghtml/codecdial.HTM>. I have also reviewed the Internet Archive’s Wayback Machine and identified archived copies of Snell & Wilcox’s

packet's "payload [] is a **chunk** of a particular elementary stream." EX1082, 8.

The "elementary [content] stream" is an example *data set* in Foster. Foster, [0064], FIG. 8; First Declaration, ¶¶170-171, ¶¶201-203.

153. And even as to generic computer "packets," prior art recognizes a "packet" can be a chunk (a "share") of a dataset. For example, EX1083 at 7:32-40 gives an example of "[a] packet [that] is a chunk of data provided by the application program." Thus, EX1084 contradicts SFI's argument that something called a "packet" cannot also be "chunk of data."

---

website, including an archive from May 15, 2006 of a "White Papers" web page.

<https://bit.ly/4fuXZWb>. That archived web page provides a link to download "MPEG Video: Mike Knee," which links to a November 9, 2006 archived PDF of EX1082. <https://bit.ly/3ZRVAiu>. See also the following link to a 2006 archived copy of Snell & Wilcox's home page, which provides a link to a "Knowledge Center" web page that links to the "White Papers" web page I identified above.

<https://bit.ly/4iPP58B>. See also <https://bit.ly/49P0xxe> (link to IEEE Xplore's page for an article added to IEEE Xplore on January 3, 2006, which lists "Mike Knee, 'MPEG Video,' Snell and Wilcox" as a reference cited in that article, which I have confirmed by reviewing a copy of that January 2006 article).

154. Similarly, EX1084 at [0026] teaches that an “RTP **packet**” can be “A **chunk or portion of multimedia data** as sent over an IP network...” Thus, EX1084 contradicts SFI’s argument that something called a “packet” cannot also be “chunk or portion of” a data set.

155. As another example, EX1085 (at 4:7-25) describes a system that “disassembling a data set into a plurality of data blocks...transforming each of the data blocks into L **data chunks**,” where “[a] size of **each data chunk can correspond to packet size**” (i.e., there is only one chunk-per-packet). Thus, EX1085 contradicts SFI’s argument that something called a “packet” cannot also be “data chunk” of a data set or that a packet is only divisible into data chunks rather than itself being the data chunk.

156. Thus, the above references corroborate my opinion that SFI is incorrect in its argument that POSAs would understand that packets cannot be *chunks* merely because packets could, in some systems, **also** be “divided into data chunks” (POR, 67)). The above references demonstrate that POSAs would disagree, as do I.

### **III. MY REPLY TO SFI REGARDING GROUND 1: IT IS STILL MY OPINION THAT ALL CHALLENGED CLAIMS WOULD HAVE BEEN OBVIOUS OVER ORSINI**

157. For Ground 1 (the Orsini-based Ground), SFI only disputes whether Orsini discloses, or an obvious implementation of Orsini would have performed,

limitation [1C]: “*an encryption operation based on the external key to further secure...data chunks.*” POR, 6-32.

158. My First Declaration advanced several alternative arguments demonstrating how Orsini both anticipates and/or renders obvious claim 1 (including limitation [1C]) based on disclosures in Orsini, POSAs understanding thereof, and/or use of conventional, well-known encryption techniques (all of which I corroborated with multiple prior art references). First Declaration, ¶¶107-134. In my opinion, those opinions are still correct, as SFI’s POR and Dr. Rubin’s declaration did not demonstrate that [1C] (and any other limitations of the challenged claims) was not disclosed and rendered obvious over Orsini.

159. For Ground 1, I have been asked to focus this Reply Declaration on one of my obviousness opinions for Ground 1. Specifically, I have been asked to focus on responding to SFI’s arguments regarding my opinion that POSAs had reason to encrypt Orsini’s share keys with a key-encryption-key (“KEK”). As I discussed in my First Declaration, POSAs had reason to do that to safeguard the share keys from detection, as multiple corroborating references I cited demonstrated was a well-known, conventional technique in the prior art. First Declaration, ¶¶131-134 (citing corroboration discussed again below, §III.A). SFI does not appear to dispute that encrypting encryption keys with KEKs was known

or that encrypting Orsini's share keys with a KEK meets [1C] under SFI's constructions.<sup>7</sup>

160. In my First Declaration (¶¶78-86, ¶¶118-119), I also discussed Orsini's disclosures regarding public-key encryption. As I discuss further below (§III.A), the reasoning behind my opinions that it would have been obvious to POSAs to use a conventional KEK to encrypt Orsini's share key was fundamentally different from, and not dependent on, testimony regarding Orsini's disclosure of using public-key encryption. Instead, my obviousness opinions were and are premised on POSAs' background knowledge **and** prior-art references that **explicitly** provided POSAs with motivation to encrypt Orsini's share keys and a reasonable expectation of success in doing so. That obviousness opinion was independent of Orsini's disclosures involving public-key encryption. In other words, even if Orsini said nothing at all about public-key encryption (which Orsini does not use to encrypt share keys), it was and still is my opinion that POSAs would have had reason to, and a reasonable expectation of success to, encrypt Orsini's share keys with a convention key-encryption key (KEK), for its well-known benefits that I demonstrated with corroboration in the prior art. *See also*

---

<sup>7</sup> I have offered alternative opinions as to how Orsini, and Foster, meet and / or render obvious [1C] under a proper claim scope. *See* First Declaration, ¶¶107-134.

*infra* §III.A (responding to SFI’s and Dr. Rubin’s arguments regarding the obviousness of using a KEK in Orsini).

**A. Response to Argument that It Would Not Have Been Obvious to POSAs to Encrypt Orsini’s Share Keys Using a Known Key-Encrypting Key (“KEK”)**

161. In its POR, SFI admits that the ’854 copies Orsini’s “Figures 1-25 and paragraphs 1-[4]23.” POR, 1-2. SFI does not dispute that that overlapping disclosure meets most of the limitations of claim 1. (As discussed above (§III), SFI only disputes whether Orsini teaches or renders obvious [1C]).

162. SFI alleges that the ’854 patent added adds “17 new figures and...~10,500” words of “significantly different technology.” POR, 1-2; EX2003, ¶33. SFI does identify which “new” disclosures in the ’854 patent are allegedly additional support for a single embodiment having all the features of claim 1, including the feature in limitation [1C]. I have reviewed the ’854 patent, and at most, there only appears to be some cursory, additional disclosure of using a “workgroup key” that encrypts another “encryption key” -- the encryption key that encrypts the entire data set – for the predictable result of (marginal) additional security. EX1001, 77:20-23, 80:24-38, 81:62-65. That is not “new” or “significant” because as I discussed in my First Declaration (¶¶121, ¶¶131-134)

using KEKs were well-known.<sup>8</sup> And Orsini itself discloses the desire to provide different levels of access to secured data to different groups of people. First Declaration, ¶¶124-126.

163. SFI and Dr. Rubin do not dispute that the encryption method described in reference to Orsini's Figure 21 results in an encrypted *share* of data being stored with a share-encryption key that is itself **not** encrypted (i.e., a plaintext key is stored with an encrypted share). First Declaration, ¶¶131-134. SFI and Dr. Rubin do not dispute that each unencrypted share key in Orsini that is stored with an encrypted share is not the key to unlock that share (e.g., Orsini's obfuscation step in Figure 21). Orsini, [0294]-[0299], FIG. 21; First Declaration, ¶¶51-54.

164. Even with Orsini's obfuscation step, SFI itself recognizes that an attacker nonetheless could "reconstruct [Orsini's] encrypted dataset if the attacker

---

<sup>8</sup> Orsini also discloses implementing protections for workgroups. *See* Orsini, [0293], [0408]-[0413]. To the extent Orsini discloses workgroup-based protection any differently than in the '854 patent, my First Declaration identified several ways by which there were conventional, known ways to implement workgroup-based protection that would have met [1C] when used in Orsini to perform the workgroup-based protection it sought. First Declaration, ¶¶124-125.

has access to each of the shares.” POR, 4; EX2033, ¶41. That is because Orsini’s various share keys are not encrypted, but rather have only been non-cryptographically obfuscated (replaced) by another share’s key. Thus, once an attacker has obtained all of the encrypted shares, the attacker has also obtained all of the keys in plaintext form.

165. As I discuss further below (§III.A.1), using the unencrypted share keys to decrypt the data would have been within the skill of a conventional determined attacker using conventional techniques and computing power at the time, even if the attacker was unaware that Orsini’s method used non-cryptographic obfuscation for the share keys. POSAs would have recognized that the same would not have been true if a conventional encryption technique, like a KEK, is used to prevent decryption of encrypted shares. As I discuss further below, POSAs knew that the best practices for security systems were to be able to provide security even when the attacker knows the algorithm for the encryption (because better security comes from keeping the key secret, not the algorithm).

166. In my First Declaration, I demonstrated in detail, with multiple pieces of *corroborating evidence*, that POSAs had reason to encrypt Orsini’s share keys with a KEK to prevent an attacker from reconstructing the encrypted dataset, even when the attacker had access to all of the encrypted shares. First Declaration, ¶121, ¶¶131-133. The corroborating evidence that I cited 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 First Declaration, ¶¶131-132).

167. As the corroborating prior art explains, POSAs knew that when an encryption key is transmitted or stored outside of the encrypting device (and often even when stored in a local storage device), a well-designed encryption system should “encrypt the encryption keys themselves...[with] a key encryption key.” EX1036, 1:66-2:6 (cited First Declaration, ¶132). As I discussed in my First Declaration (¶¶131-132) and further below, POSAs understood that the benefits of and desire to encrypt encryption keys with KEKs (so as not to have unencrypted keys outside the computer performing the encryption) applied whether the “encryption keys are included with the encrypted message” (EX1036, 1:66-2:6) or are stored separately from and/or obfuscated with encrypted data-portions.

168. My opinion regarding POSAs’ background knowledge that it was known (and preferred) to encrypt encryption keys with KEKs, even when the encryption key is not stored with the data it encrypts, is corroborated by several prior art references discussed below.

169. For example, I cited EX1035 (cited First Declaration, ¶¶131-132). EX1035 (at 10:42-62) discloses that the “objectives of key management are: (1) cryptographic keys should **never** occur in clear form outside the cryptographic

device, except under secure conditions” and “(2) no cryptographic operation, or combination thereof, using any cryptographical quantities that are routinely stored or routed through the system, or derived therefrom, should permit clear keys to be recoverable outside the cryptographic device.” EX1035 continues that “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.**” EX1035, 10:56-62. *See* First Declaration, ¶¶131-132 (discussing EX1035).

170. As another example, I cited EX1036 (cited First Declaration, ¶132). EX1036 at 1:66-2:6 taught as background information that “[s]ecurity for such a system, where the encryption keys are included with the encrypted message, is not to store or transmit the encryption keys in an easily readable format but rather to encrypt the encryption keys themselves.” That is, “the data encryption key will be encrypted by a key encryption key.” EX1036, 2:2-6.

171. In response to SFI’s arguments and Dr. Rubin’s opinions challenging my opinions on the obviousness of using a KEK in Orsini, I cite as another example EX1086. EX1086 at 7:3-22 discloses a system with an “archive station 4 [that] is usually a server or consists of several servers, i.e. computers providing service,” including at least, “a first and a second server... wherein encrypted data 10 are stored on the first server and **the second keys 7 belonging thereto are stored on a second server.**” That is, the encrypted data and the encryption key for

that data are stored in separate locations. EX1086 at 7:3-22 discloses that

**“Further the second keys 7 can be stored encrypted on the archive station 4.”**

That is, even though the encryption keys are stored separately from the encrypted data, EX1086 corroborates my opinion in my First Declaration that POSAs would have known how to and would have wanted to encrypt those keys. Encrypting those keys teaches or at least suggests using a KEK for that encryption of keys because KEKs, by definition, are keys that encrypt other encryption keys.

172. In response to SFI’s arguments and Dr. Rubin’s opinions challenging my opinions on the obviousness of using a KEK in Orsini, I cite as another example EX1087. EX1087 teaches a “method for securely storing data,” where the “data are encrypted using a backup key” and “[t]he backup key is protected as a protected backup key” (i.e., the backup encryption key is encrypted with a KEK). EX1087, [0017]. Rather than be stored with the data, “the backup key is protected by being stored” elsewhere, such as “in an internet server, in a pocketable medium or in a trusted computer.” EX1087, [0019]. “Preferably, protecting the backup key **includes encrypting the backup key.**” EX1087, [0020]. Thus, EX1087 is another example of prior art demonstrating that POSAs knew that even when encrypted data was stored separately from its encryption key, the encryption key should be protected by a KEK when stored outside the encrypting device (which

itself was known to be used in a conventional key management systems). EX1087, [0020]-[0025].

173. In response to SFI's arguments and Dr. Rubin's opinions challenging my opinions on the obviousness of using a KEK in Orsini, I cite as another example EX1112. EX1112 at [0025] discusses the use of a public encryption key, which it says "is also known as a 'session key,'" and it is "used to encrypt each individual field of data stored in [a] data store." EX1112 at [0026] discloses that there is also "master key" that is "protected by a password." The master key is a key "used" "to generate the public key." EX1112, [0022]. EX1112 further discloses that "[t]he session key may be stored in the same data store used to store the encrypted data, or in a different data store." EX1112, [0027]. EX1112 adds that "[w]herever the session key is stored, it is preferable that it is encrypted beforehand." EX1112, [0027] (e.g., "[t]he session key may be encrypted using the master key"). Thus, EX1112 corroborates my opinion that POSAs understood that it was conventional to encrypt an encryption key "wherever" it is stored, whether stored with data it is encrypted or stored somewhere else (like a conventional data store).

174. Moreover, Orsini itself says "[i]t 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 several well-

known, prior art encryption techniques / standards, such as “Triple DES and AES.” Orsini, [0378]. Those were examples of prior art commercialized and/or standardized encryption techniques. POSAs knew that Triple DES and AES, for example, described using KEKs and the storage / management thereof.

175. My opinion in that regard is corroborated by several prior art references discussed below.

176. In response to SFI’s arguments and Dr. Rubin’s opinions challenging my opinions on the obviousness of using a KEK in Orsini, I cite as another example EX1094. EX1094 at [0056] notes that “[s]tandardized algorithms,” like “Triple-DES **Key Wrap**” were compatible with the referenced “exemplary key management mechanisms.” POSAs understood that “Key Wrap” in this context refers to wrapping (encrypting) an encryption key with another encryption key (i.e., a KEK). EX1094 at [0057] further notes that “[v]arious algorithms can be used to encrypt the content with the encryption key, including the Data Encryption Standard (DES), Triple-DES, RC2 and AES algorithms.”

177. In response to SFI’s arguments and Dr. Rubin’s opinions challenging my opinions on the obviousness of using a KEK in Orsini, I cite as another example EX1090. EX1090 at 19:47-61 discusses “deriv[ing] a key to ‘wrap’ (e.g., encrypt) the data encryption key (DeK),” by “generating a Key Encrypting Key (KeK) 430 to protect the Data Encrypting Key (DeK) 412, that, in turn, is used to

encrypt the plaintext data 402.” And at 21:1-22:54, EX1090 discusses using the “AES[] key wrapping function” (i.e., a KEK function) to encrypt the DeKs.

178. In response to SFI’s arguments and Dr. Rubin’s opinions challenging my opinions on the obviousness of using a KEK in Orsini, I cite as another example EX1096.<sup>9</sup> EX1096 is a copy of an AES key-wrap algorithm, and it notes

---

<sup>9</sup> EX1096 is a document published by the IETF – the Internet Engineering Task Force, which is the “standards development organization (SDO) for the Internet.” <https://www.ietf.org/about/>. EX1096 provides on its face a “Copyright Notice” “Copyright (C) The Internet Society (2002)” and lists a date of “September 2002.” In my opinion, POSAs would have been aware of RFC (Requests for Comments) published by the IETF, like EX1096, at the time of the alleged invention of the ’854 patent and would have known how to locate RFCs like EX1096 by navigating to the IETF’s RFC database. As the IETF’s website notes, the “IETF publishes its technical documentation as RFCs,” which “are freely available to download, copy, publish, display and distribute, in a variety of formats, under a license granted by the IETF Trust.” <https://www.ietf.org/process/rfc/>. *See also* <https://bit.ly/3Bw9Rtr> (a February 6, 2007 archived copy of EX1096 from the Internet Archive’s Wayback Machine). Based on the above evidence and my review of the indicia appearing on EX1096 and my understanding of the IETF and

that “[t]his key wrap algorithm **needs to provide ample security to protect keys in the context of prudently designed key management architecture.**” EX1096, 1-3.

179. Orsini’s use of standardized encryption techniques that already described and accounted for use of KEKs (like AES and DES), reinforces my opinion that it would have been obvious to use a KEK in Orsini **and** that POSAs would have had a reasonable expectation of success doing so in Orsini. First Declaration, ¶134, ¶138. *See also infra* §III.B (where I reply to SFI’s arguments regarding POSAs’ expectation of success in modifying Orsini).

180. In contrast to the known cryptographic protection that would have been provided using the known, conventional cryptographic protection of a KEK, Orsini’s non-cryptographic obfuscation only provides (some minor) security when an attacker obtains fewer-than-all shares. In contrast, POSAs would have recognized that a KEK would have provided cryptographic security (generally stronger than non-cryptographic security and certainly stronger than Orsini’s obfuscation) **even if** an attacker obtains all of the encrypted shares.

---

its RFC database, it is my opinion that EX1096 was publicly accessible in the prior art by being disseminated or otherwise made available to POSAs exercising reasonable diligence by 2003.

181. For example, with Orsini's technique, once an attacker recovered all of the encrypted shares (and thus all of Orsini's plaintext share keys), it would have been within the capabilities of a conventional attacker (i.e., an attacker having the level of sophistication that POSAs would have wanted to protect against) to reverse Orsini's obfuscation to put the share keys back together.

182. Indeed, in Orsini's Figure 21 embodiment, there are only four shares and four share keys, and thus (at most) only 24-different combinations of shares and keys.<sup>10</sup> Moreover, neither Orsini's Figure 21 method nor the challenged claims require four shares, but instead allow for as few as two shares, which would have only **one way to non-cryptographically obfuscate** in the manner Figure 21 shows, as Dr. Rubin admitted at his deposition. Rubin-Depo., 24-25; *See, e.g.*, Orsini, [0013] ("The method [of the present invention] comprises one or more steps of parsing, splitting or separating the data to be secured *into two* or more parts or portions."). Thus, POSAs understood that whatever additional security was provided by Orsini's obfuscation step, was clearly not significant security, and

---

<sup>10</sup> Because Orsini's Figure 21 embodiment with the (optional) ability to obfuscate does not store encrypted shares with their share keys, there are even fewer than 24 combinations (e.g., the six combinations where share 1 is stored with key 1 would not be allowed).

certainly not comparable to the type of cryptographic security that could be provided by a KEK (whether in place or in addition to obfuscation).

183. In contrast to Orsini's non-cryptographic obfuscation, with the cryptographic protection of a KEK, even if an attacker obtained all of Orsini's encrypted shares and keys, the attacker would have **still** needed to know **where** the KEK was stored, how to **breach** the security of that key store, **and then determine** how to find the right KEK **and break the cryptographic protection of the KEK**.

184. Using the cryptographic protection of KEK would have also facilitated providing all encrypted shares to trusted parties, while still maintaining stronger control over when and how the parties could access the secured data. In contrast, with Orsini's obfuscation, once a party obtained all of the encrypted shares (which are stored with the keys), POSAs would have recognized that it would have been within the capabilities of a determined attacker to reverse the obfuscation to decrypt the shares. POSAs would have also known that the time it would have taken an attacker to break the security in the shares would have been significantly shorter than the time it would have taken to break KEK-based encryption (which, depending on what encryption was used, may not even have been possible at the time using then-existing encryption techniques for KEKs).

185. My opinion that POSAs' background knowledge would have made it clear to POSAs that Orsini's non-cryptographic obfuscation step provided little, and certainly less, security compared to the cryptographic security of a conventional KEK is corroborated by several references that I cited and discussed in my First Declaration (§§121, §§131-134) and the additional ones I cite and discuss below in response to SFI's arguments and Dr. Rubin's opinions.

186. In response to SFI's arguments and Dr. Rubin's opinions challenging my opinions on the obviousness of using a KEK in Orsini, I cite SFI's EX2007. EX2007 teaches that even "26 possible keys" is subject to "a *brute-force attack*" and that's poor security because "any secure encryption **must not be vulnerable to such brute-force attack.**" EX2007, 11. As I discussed above, Orsini's example embodiment includes (at most) 24 different combinations (and can be as few as one way to obfuscate, when there are only two shares). POSAs would have recognized that such a small number of combinations would have also been subject to a (relatively simple) brute-force attack when all the shares were obtained, and thus provided substantially weaker security compared to the cryptographic protection of a KEK (whose security would not have been meaningfully reduced, if at all, even if all encrypted shares were obtained).

187. In response to SFI's arguments and Dr. Rubin's opinions challenging my opinions on the obviousness of using a KEK in Orsini, I further cite EX1089.

EX1089 at [0569] discusses “a non-cryptographic security approach” (e.g., one that “applies a 32-byte XOR mask, repeatedly, to the entire data object”), which is substantially more secure than Orsini’s obfuscation. Nonetheless, EX1089 says that non-cryptographic security does “serve[] to obscure the data so it cannot be used directly, **but is generally more open to attack than is the cryptographic security approach.**” EX1089, [0569]. That is, EX1089 corroborates my opinion that POSAs would have known that non-cryptographic security (like Orsini’s obfuscation, which POSA would have recognized as being weaker than EX1089’s 32-byte XOR mask) would also have been not as secure as conventional cryptographic security techniques (like using a KEK).

188. In response to SFI’s arguments and Dr. Rubin’s opinions challenging my opinions on the obviousness of using a KEK in Orsini, I further cite EX1090. As EX1090 (at 2:39-41) discloses (as POSAs knew), “[t]he more difficult it is for an unauthorized entity to obtain a valid key, the more secure the cryptographic scheme.” There can be no reasonable dispute that POSAs would have known that using the cryptographic protection of a KEK could have made Orsini’s data more secure (regardless of what security was provided by Orsini’s non-cryptographic obfuscation step). Indeed, that is what EX1089 teaches, as I discussed above.

189. In response to SFI's arguments and Dr. Rubin's opinions challenging my opinions on the obviousness of using a KEK in Orsini, I further cite EX1106.<sup>11</sup> EX1106 is a document published by the U.S. Department of Commerce's National Institute of Standards and Technology ("NIST"), titled "Recommendation for Key Management – Part 1: General." It is NIST Special Publication 800-57, published August, 2005. *See* EX1106, pdf page 2. This document provided "basic key management guidance" to advise on the "'best practices' associated with key

---

<sup>11</sup> NIST's website indicates that EX1106 was published "August 2005."

<https://csrc.nist.gov/pubs/sp/800/57/pt1/final>. I have also located a September 11, 2005 archived copy of EX1106 from the Internet Archive's Wayback Machine. *See* <https://bit.ly/3ZQiYwR>; *see also* <https://bit.ly/4fElcFO> (September 3, 2005 archive of the webpage for "NIST Special Publications"); <https://bit.ly/3P9wM0x> (August 31, 2005 archive of NIST's homepage – <http://csrc.nist.gov> – which had a link to the "Security Guidelines" page for NIST's Special Publications, and a direct link to copies of EX1106). Based on the above evidence and my review of the indicia appearing on EX1106 and the U.S. Government's NIST website stating when EX1106 was published, it is my opinion that EX1106 was publicly accessible in the prior art by being disseminated or otherwise made available to POSAs exercising reasonable diligence by August / September 2005.

management.” EX1106, 5. As EX1106 states (and POSAs knew), “Keys are analogous to the combination of a safe” and “the security of information protected by cryptography directly depends on the strength of the keys” and “the protection afforded to the keys” themselves. EX1106, 5.

190. EX1106 discusses “Symmetric key algorithms” and states that the symmetric “key **shall not** be disclosed to entities that are not authorized access to the data protected by that algorithm.” EX1106, 34 (emphasis original); *see also* EX1106, 35-36 (discussing various “Symmetric Key Algorithms used for Encryption and Decryption,” including AES and “Triple DEA (TDEA)” (aka Triple DES)). As discussed above, Orsini’s obfuscation step violates best practices in the field of securing data, as recognized by NIST, because Orsini exposes a plaintext share key to anyone that obtains an encrypted share. EX1106 at 61 further notes that “[t]he security strength of an algorithm for a given key size is traditional described in terms of the amount of work it takes to try all keys for a symmetric algorithm.” EX1106, 61. As discussed above, with respect to Orsini’s obfuscation, the amount of time it would take to reverse the obfuscation step would have been well-within the capabilities of a determined attacker. (As discussed below, Dr. Rubin admits, Orsini’s obfuscation is “not really involving a lot of math,” Rubin-Depo., 154)

191. And if the attacker knew that Orsini was performing obfuscation by pairing encrypted shares with unencrypted keys (e.g., something that would likely be hard to prevent a rogue insider from discovering), reversing Orsini's obfuscation step would have been even easier for an attacker to reverse. *See, e.g.,* EX1044, [0002] (noting as background information that while there has been a "great deal of focus" on "improving computer security" over the Internet, there was a need to "address the problem of providing security once someone has accessed the computer," like when "many employees may have access to a computer system" and thus there was still a need "to provide a level of security for information on the computer system").

192. The same would not have been true with a conventional KEK: even if the attacker knew a KEK was used to protect Orsini's share keys, it would have still been extremely challenging (and possibly beyond a conventional attacker's skill level) to break the encryption of a KEK. EX1106, 123 ("Cryptographic integrity mechanisms...protect against an active, intelligent attacker.").

193. 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).<sup>12</sup> 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.

194. My opinion that POSAs would have recognized that Orsini’s non-obfuscation step would not have provided anything close to the security provided by the cryptographic security provided by a conventional KEK is consistent with testimony Dr. Rubin gave at deposition, as I discuss below.

195. At deposition, Dr. Rubin testified that figuring out the different combinations of encrypted shares and share keys in Orsini’s obfuscation step “is a pretty simple math problem” such that “it wouldn’t take long” just to write out the combinations by hand. Rubin-Depo., 22-25. Dr. Rubin further testified that in his

---

<sup>12</sup> 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).

opinion he didn't "even **think it's really involving a lot of math**" because "[y]ou're just putting something with other things and not putting it with other things..." Rubin-Depo., 154-156. Dr. Rubin further testified that POSAs knew "**you should put your security in the keys** and not in the hiding of your algorithms" but "performing an **obfuscation is [] not putting the protection in the keys**, so I would say **this does not on its own provide security.**" Rubin-Depo., 154-156. I agree that Orsini's obfuscation step is not putting the security of the encrypted shares in a key (i.e., not encrypting the keys) and that the obfuscation does not on its own provide much (if any) security to prevent an encrypted share from being decrypted.

196. As demonstrated in my First Declaration (¶¶131-132) and above, POSAs were well aware that it was undesirable to leave an encryption key unencrypted (as Orsini does with its share key) because it unnecessarily exposed the key (and thus the data it unlocks, in this case the encrypted share) to the risk of a successful attack that could (and POSAs knew would) have been substantially reduced by using the known, conventional cryptographic protection of a KEK. *See, e.g.,* First Declaration, ¶132.

197. Therefore, as I opined in my First Declaration (¶¶131-132), encrypting Orsini's share keys with a KEK would have at least been obvious to POSAs because it would have been little more than using a known technique

(encrypting encryption keys for storage) to solve a known problem (securing the encryption key used to encrypt a piece of data). Doing so required only ordinary skill and there would not have been any unexpected results because using a KEK involved precise mathematics, and, therefore, POSAs would have had a reasonable expectation of success.

**1. My KEK-based Obviousness Opinion was Independent of, and based on independent corroborating evidence from, my opinion that Orsini's public-key encryption disclosures provided an anticipating embodiment.**

198. SFI and Dr. Rubin argue that my opinions regarding the obviousness of using a KEK to encrypt Orsini's share keys fail for the same reason they dispute my opinion that Orsini anticipates based on Orsini's disclosure to use public-key encryption. POR, 17-18; EX2033, ¶¶53-63. Though I continue to believe that Orsini's public-key disclosures are included in a single embodiment that contains Orsini's encryption method disclosed in Orsini's Figure 21, my anticipation opinions (including based on Orsini's public-key encryption disclosures) are separate from and were supported by different evidence than my opinions of the obviousness of using a conventional, externally received KEK to encrypt Orsini's share keys.

199. Completely independent of Orsini's disclosures about encrypting data to and at data depositories using public-key encryption, my opinion was (and is) that it would have been obvious to POSAs to encrypt Orsini's share keys with a

(in several ways) was premised on Orsini’s express disclosures alone, as that disclosure would have been understood by POSAs.

200. However, as I discussed above (§III), it is undisputed that Orsini’s embodiments, in particular the ones based on Figure 21’s method, do not encrypt Orsini’s share keys. Thus, I did not opine that Orsini anticipates by encrypting Figure 21’s share keys with another key. Instead, I opined that it would have been obvious for POSAs to do so, using a conventional externally-stored and -retrieved KEK. As I discussed above (§III.A) and further below (§§III.A.2-III.A.3), the opinion in my First Declaration on the obviousness of using a KEK was based on, for example, POSAs’ background knowledge (which I corroborated with independent evidence) that encryption keys—like Orsini’s share keys—“**should never**” (EX1035, 12:15-23) be transmitted or stored in unencrypted fashion. That evidence – not Orsini’s public-key disclosures – was the corroborating support for my opinion that POSAs had reason to encrypt Orsini’s share keys with a conventional KEK. *Compare* First Declaration, ¶¶78-86, ¶¶118-119 (discussing my anticipation arguments for public-key encryption of already-encrypted data (**not** the share keys)) *with* First Declaration, ¶¶131-134 (discussing my obviousness opinions based on a conventional KEK encryption to encrypt Orsini’s share keys).

**2. My KEK-based Obviousness Opinions Did Not Rely on Common Sense to Supply a Missing Limitation in the Prior Art.**

201. I understand that SFI argues that the Petition's KEK-based obviousness argument relied on common sense to supply limitation [1C], which is allegedly missing in the prior art. POR, 5-7. As I discuss below, SFI's argument fails on (at least) two accounts: (1) POSAs would not agree that limitation [1C] is missing in the prior art (as demonstrated by multiple corroborating prior art references) and (2) my opinions demonstrating the obviousness of encrypting Orsini's share keys with a KEK did not rely on POSAs' bare common sense.

**a. POSAs would not have agreed that limitation [1C] was missing in the prior art.**

202. As an initial matter, in my opinion, POSAs would not agree that limitation [1C] is missing in the prior art. As I discussed above (*supra* §III.A.1), my opinions on the obviousness of using a KEK were supported by (at least) EX1035 and EX1036, both of which disclose the known desirability of using a KEK to encrypt an encryption key for storage. First Declaration, ¶¶131-134.

203. Moreover, as I discussed in my First Declaration (¶¶217-220), Foster is another example of prior art discloses limitation [1C]. Indeed, SFI and Dr. Rubin do not dispute that Foster's content units are individually encrypted with title keys that are, in turn, encrypted with a KEK.

204. My First Declaration also identified several other disclosures in Orsini and / or obvious variations thereof using well-known encryption techniques that teach and / or render obvious limitation [1C] under Google’s and/or SFI’s constructions. *See, e.g.*, First Declaration, ¶¶107-134. Though SFI and Dr. Rubin dispute whether POSAs had reason to include those other conventional encryption techniques in Orsini (e.g., cryptographically reordering the data within each encrypted share), SFI and Dr. Rubin do not dispute that those were known encryption techniques that would meet limitation [1C] if performed in Orsini. Thus, those other known encryption techniques are further evidence of limitation [1C] being known in the art under one or more of the parties’ claim constructions.

205. That limitation [1C] is not missing in the prior art is further corroborated by additional prior art I discuss below.

206. In response to SFI’s and Dr. Rubin’s arguments that limitation [1C] is missing in the prior art, I cite EX1088. EX1088 at 2:20-3:16 teaches to use a KEK to encrypt keys that were used to encrypt “partitions[s]” of “digital content.”

207. In response to SFI’s and Dr. Rubin’s arguments that limitation [1C] is missing in the prior art, I cite EX1080. EX1080 at pdf pages 2-7 describes a system wherein “[d]ata is encrypted at the source by the TEK” and the system “distribute[s] the key to each receiver.” EX1080 further disclosed that the TEK **“must itself be encrypted”** using a KEK. EX1080 at 6.

208. In response to SFI's and Dr. Rubin's arguments that limitation [1C] is missing in the prior art, I cite again EX1106. As I discussed above (§III.A), EX1106 was published by the US Department of Commerce as a guidelines for best practices in the field of securing data, and it taught that when a symmetric encryption key is used (like Orsini's share keys), and those key "exist[] external to the cryptomodule" (as in Orsini) "confidentiality protection **shall be provided either by encryption (e.g., key wrapping).**" EX1106, 72 (emphasis of "shall" original).

209. As I discussed above, Orsini itself says the encryption keys in its system are generated according to several standardized encryption techniques, including DES and AES. Orsini, [0378] (cited, *e.g.*, First Declaration, ¶138). As I discussed above with corroboration (§III.A), both DES and AES disclosed using key wrappers (KEKs) to encrypt their encryption keys. *Infra* §III.B (citing additional corroboration). Thus, POSAs' knowledge of the very same encryption techniques that Orsini explicitly identifies would have included knowledge of using a KEK encryption key to encrypt the very keys Orsini uses (e.g., using an AES key-wrapping key to protect Orsini's share keys based on AES encryption).

210. Thus, in my opinion POSAs would disagree with SFI and Dr. Rubin that limitation [1C] is missing from the prior art. Instead, as I demonstrated in my First Declaration and above in response to SFI's POR, limitation [1C] was not only

in the prior art, it was long-known by POSAs in the prior art, and had even been incorporated into the same standardized encryption techniques that Orsini relies on.

**b. My opinions on the obviousness of encrypting Orsini's share keys with a KEK was not based on common sense.**

211. But even if it were true that limitation [1C] were not disclosed in the prior art, my opinions in my First Declaration on the obviousness of encrypting Orsini's share keys did not rely only POSAs' "common sense." Rather, I cited multiple corroborating references (e.g., EX1035 and EX1036) to support my testimony that explained in detail why POSAs had reason to further secure Orsini's shares by using the well-known (not SFI-invented) technique of encrypting encryption keys using conventional KEKs. First Declaration, ¶¶131-134.

212. In response to SFI's argument that limitation [1C] is missing in the prior art, I have cited and discussed additional prior art references above (§§III.A-III.A.2.a) that refute SFI's argument. Similarly, in response to SFI's and Dr. Rubin's argument that POSAs lacked motivation to encrypt Orsini's share keys with a KEK, I cited several additional corroborating references in this reply declaration. *See infra* §§III.A.3-III.B; *supra* §§III.A-III.A.2.a.

213. Thus, my obviousness opinions were and are not based on common sense, but are instead based on my (corroborated) opinion that limitation [1C] was not only in the prior art, but was also within POSAs' general knowledge

(corroborated by evidence) of how to secure encryption keys that are used to secure stored and/or transmitted data. Specifically, as I discussed in my First Declaration (¶¶131-134) and further below in response to SFI's and Dr. Rubin's arguments, POSAs would have wanted to encrypt Orsini's share keys because encrypting keys was a known solution to the problem of how to secure encryption keys stored with encrypted data **and** Orsini's obfuscation step, to the extent it addressed that problem, would have been recognized as being less secure than using a conventional KEK.

214. Thus, POSAs' motivation to use a KEK in Orsini that I have identified is not based on "common sense," but is based on well-understood, documented principles in the field of securing data to encrypt plaintext encryption keys (like Orsini's) with a KEK.

**3. My reply to SFI's Challenges to my opinions regarding POSAs' Motivation-to-Combine Orsini with a conventional KEK**

215. In its reply, SFI offers three basic challenges to the identified motivations to combine Orsini with a conventional KEK: **(1)** SFI argues that Orsini already provides sufficient security via its obfuscation step and thus POSAs had no reason to add additional security by encrypting Orsini's share keys with a KEK, **(2)** SFI argues that encrypting Orsini's share keys with a KEK would have introduced complications that would have dissuaded POSAs from pursuing the

combination, and (3) SFI argues the KEK-based obviousness ground is hindsight driven. Consistent with the opinions offered in my First Declaration (§§131-134) and discussed further below (§§III.A.3.a-III.A.3.c), in my opinion POSAs would have disagreed with each of SFI's arguments.

**a. It is still my opinion that POSAs Would Have Wanted to Make Orsini's System More Secure by using a KEK.**

216. In its POR, SFI argues that POSAs would have had no reason to further secure Orsini's system because (according to SFI) Orsini itself was satisfied with the security its system provided. POR, 21-24 ("It is not obvious to modify a reference that already achieves the benefit supposedly served by the proposed modification.")). In my opinion, SFI's argument misunderstands how POSAs would read Orsini, in light of what Orsini itself says and further in light of POSAs' background knowledge of cryptography techniques and the field of securing data.

217. I first note that the fact that Orsini's used non-cryptographic obfuscation after encrypting the original data set and each individual share confirms that Orsini itself recognized that there was still a need to further secure Orsini's encrypted shares. That is, Orsini itself wanted to secure the encrypted shares further. The way Orsini did so was by making it (slightly) harder to access the share encryption keys, by not storing a share key with the share that it encrypted.

218. Orsini **also** discloses that “certain steps of” Figure 21’s method “may be performed in different order, or repeated multiple times, as desired.” Orsini, [0301]. Orsini further discloses in connection with Figure 21’s method that “[e]ach portion of parsed data may be uniquely secured in any desirable way provided only that the data may be reassembled, reconstituted, reformed, decrypted or restored to its original or other usable form.” Orsini, [0301]. Orsini would not have disclosed multiple alterations to Figure 21’s method that added varying forms of security if Orsini (let alone POSAs) actually thought Figure 21’s method was already “sufficient” (POR, 21-24), let alone sufficient for all types of data sets, no matter how sensitive the data set being secured.

219. If SFI and Dr. Rubin were correct that Orsini believed that the method illustrated in Figure 21 was “sufficient” (POR, 21-24), then Orsini would not have **repeatedly** identified several **additional** steps that **could** be layered on top of Figure 21’s method to provide **additional** security (e.g., repeating a Figure 21, uniquely securing each share, etc.) That is, rather than teach any of its disclosed methods were sufficient, POSAs would have recognized that Orsini itself recognized that its obfuscation – splitting encrypted portions of data and storing those encrypted portions of data on different storage devices – was potentially insufficient security. POSAs would have agreed that additional (cryptographic) security was often desirable.

220. But rather than use the well-known, pre-existing and conventional cryptographic-based approach of using a KEK, Orsini elected to perform what it termed obfuscation. Even if Orsini thought otherwise (as SFI alleges), POSAs would have recognized that Orsini’s non-cryptographic obfuscation (unlike KEK-based security) provided little additional security over even having no obfuscation at all for a determined attacker. *See also supra* §III.A. That is, POSAs would have recognized that Orsini’s obfuscation provided **some** additional security, under **some** circumstances (e.g., when an attacker did not possess all encrypted shares), but using a KEK encryption would have provided (at least) the same security in some cases and additional security in other cases that obfuscation did not provide (e.g., even when an attacker **did** possess all encrypted shares). *See* EX1100, [0006]-[0011]; [0010] (“Encryption’s purpose is to ensure privacy by keeping information hidden from anyone for whom it is not intended, *even those who have access to the encrypted data.*”).

221. As I discussed in my First Declaration (¶¶120-134) and above (§III-§III.A.2), even if Orsini had taught or suggested that Figure 21’s method was secure on its own, POSAs would not have agreed. POSAs knew that Orsini’s non-cryptographic technique of storing the share keys separately from the shares they encrypted was **less secure** than doing so while **also** (or alternatively) encrypting the share keys using a KEK. That is because, as I discussed above, unlike using a

KEK, Orsini's obfuscation provided little-to-no additional protection once all of the encrypted shares are collected. *Supra* §III.A (*citing, e.g.*, EX1089, [0569]; EX1090, 2:39-41 ("The more difficult it is for an unauthorized entity to obtain a valid key, the more secure the cryptographic scheme.")).

222. SFI and Dr. Rubin do not dispute my opinion in my First Declaration that using a KEK would have provided additional security in Orsini's method. SFI and Dr. Rubin also do not dispute that cryptographic security would have been almost always (and certainly could have been made) stronger than Orsini's non-cryptographic obfuscation (e.g., by using the encryption techniques available at the time of the invention that were known to be, or at least believed to be, unbreakable in any reasonable amount of time with then-existing technologies).

223. As I discussed above (§III.A), Orsini's Figure 21 divides data into four shares/chunks and stores the share encryption keys with those shares/chunks. Orsini, [0298]-[0301]. First Declaration, ¶¶50-54. There are (at most) only twenty-four different chunk-key pairings that are possible when there are four chunks. POSAs would have understood that computers could easily overcome this obfuscation using a conventional brute-force attack that tried all twenty-four possibilities until the correct one is identified.

224. For example, SFI's EX2007 (at 11) discloses that even "26 possible keys" is subject to "a *brute-force attack*" but "any secure encryption **must not be**

**vulnerable to such brute-force attack”** (italicized emphasis original). In the same way, Orsini’s 24 different obfuscation possibilities would be subject to a conventional brute-force attack that would have been more vulnerable than using the conventional cryptographic-based security of a KEK.

225. Moreover, as I discussed above (§III.A), neither Orsini nor the patent’s claims required four shares and four share keys, but rather allow for as few as two shares and two share keys, which has only **one** way to non-cryptographically obfuscate by swapping one share’s key for the other, as Dr. Rubin agreed at deposition (Rubin-Depo., 24-25). Plainly, only a single way to obfuscate is not providing the same level of protection that conventional encryption techniques with a KEK would have provided by the mid-2000s.

226. Thus, POSAs would have recognized that SFI’s assertion that Orsini’s non-cryptographic obfuscation “achieves the benefit...served by” a KEK’s cryptographic protection (POR, 23) is incorrect. Rather, POSAs would have understood that using a KEK would have provided independent, and almost certainly in every case (but at least in many cases) better, security than Orsini’s obfuscation, whether both were used together or only KEK-based security was used in place of Orsini’s non-cryptographic obfuscation.

227. SFI alleges that Orsini teaches that its obfuscation was “sufficient” to protect its data. POR, 24 (citing Orsini, [0353]). As I discussed above, in my

opinion, those boilerplate statements in Orsini would not have in any way discouraged POSAs from modifying Orsini to use better and/or alternative cryptographic techniques (like a KEK) to further increase security against unauthorized attack. Indeed, the '854 patent, which allegedly recognized the inadequacies of Orsini's technique and allegedly solved it by using a KEK (a well-known technique) for its well-known purpose, has that same disclosures as Orsini. *Compare* Orsini, [0353] *with* EX1001, 60:17-20.

228. But even if SFI were right that Orsini itself thought that obfuscation was "sufficient" (POR, 24) for some of Orsini's *data sets*, that would not have made using the known KEK-based cryptographic technique non-obvious to POSAs to use in Orsini's system for all *data sets*, as an alternative and/or additional (and more-secure) way to add security, and *further secure the data chunks* (under SFI's view of how that limitation can be met). I have been informed that POSAs are hypothetical people that are presumed to be aware of all the pertinent prior art. As I discussed above (§III-§III.A.2), my First Declaration and this reply have cited numerous prior art references in the field of securing data that unambiguously taught the benefits of and the risks in not encrypting encryption keys with a conventional KEK.

229. POSAs would have known that even when a prior art reference (like Orsini) discloses a certain level of a solution to a problem (e.g., Orsini's use of

obfuscation to provide some additional security against discovering encryption keys and decrypting shares), POSAs in the field of securing data would have sought out another known (and in my opinion would have been recognized as an obviously more secure) solution to the same problem (in this case, using a KEK to provide meaningful cryptographic protection against discovering Orsini's share keys). That is especially true to POSAs in the field of securing data because POSAs knew "[t]he more difficult it is for an unauthorized entity to obtain a valid key, the more secure the cryptographic scheme." EX1090, 2:39-41. POSAs would have recognized that for (at least) high importance *data sets* and/or risk levels, additional security would have been desired and worth whatever complexity and/or cost using a KEK would have required. (And given the ubiquitousness of using KEKs in the prior art, as I've demonstrated in my First Declaration and above, POSAs would have found the complexity and/or cost to be negligible or at least resolvable using their ordinary skill with a reasonable expectation of success.)

**b. In my opinion, any alleged Drawbacks to the modifying Orsini to use a KEK that SFI Identifies would have been Easily Overcome**

230. SFI argues that alleged "drawbacks associated with managing and securing" KEKs would have dissuaded POSAs from using KEKs in Orsini. POR, 24-32. As I explained in my First Declaration (¶¶131-134) and in response to SFI's arguments, it is my opinion that POSAs would not have been dissuaded from

using KEKs in Orsini. Instead, POSAs would have had clear reason to use a KEK in Orsini and a reasonable expectation of success in doing so. I also note that SFI does not allege that addressing these alleged “drawbacks” was beyond POSAs’ skill. As I discuss below, these alleged drawbacks would not have been insurmountable, but instead would have required only ordinary skill to consider and deal with. That is why POSAs would have had (at least) a reasonable expectation of success in implementing a KEK in Orsini.

231. I note first that Orsini **already** taught techniques for managing and securing KEKs, and certainly could have been modified in a conventional, straightforward way to use conventional techniques for doing so. For example, as I discussed above (*supra* §III.A), Orsini contemplates using and supports “Triple DES and AES” (Orsini, [0378]), which already taught the use and management of KEKs, as I demonstrated above with corroborating evidence.

232. In addition, Orsini also **explicitly** says it supports “a wide number of alternative key management systems.” Orsini, [0107]. Several prior art references I cited in my First Declaration (and additional corroborating evidence I cite in response to SFI’s reply arguments) demonstrate POSAs knew that using and managing (e.g., creating and storing) KEKs was conventional by the time of the patent’s alleged priority date.

233. In response to SFI's arguments and Dr. Rubin's opinions challenging POSAs' expectation of success in modifying Orsini to use and manage KEKs I cited EX1036 (cited First Declaration, ¶132). EX1036 discloses a "system for **managing keys**," including KEKs (2:29-61) and says the "prior art" already included "security infrastructure to solve the key management problem[]" (2:7-22).

234. In response to SFI's arguments and Dr. Rubin's opinions challenging POSAs' expectation of success in modifying Orsini to use and manage KEKs, I further cite SFI's EX2032. EX2032 discloses that IBM had already "designed and developed" a commercialized "Common Cryptographic Architecture...to provide" for, for example, "key management." EX2032, 683. Among those keys that were managed by that prior art (commercial) system were "key-encrypting keys (KEKs)" that were "used to export and import encrypted keys between systems." EX2032, 683. EX2032 further discloses "encrypt[ing] session keys using [a] master key." EX2032, 683-684.

235. In response to SFI's arguments and Dr. Rubin's opinions challenging POSAs' expectation of success in modifying Orsini to use and manage KEKs, I further cite EX1017 (cited First Declaration, ¶93). EX1017 (at 1:50-1:69) discloses as background information that that "[t]here are generally two categories of encryption key variables in a secure communication system.... The second type is called a **key encryption key (KEK)**, and is used to encrypt encryption key

**variables for** transit, transmission or **storage.**” See also First Declaration, ¶93 (discussing EX1017). EX1017 further discloses as background information that “[i]n more advanced secure communication systems, encryption **key variables are stored and distributed electronically to communication units using a key management controller or KMC.**” EX1017, 1:50-1:68. That is, it was background knowledge as of 1993 how to store, distribute, manage, and use KEKs for secure storage.

236. In response to SFI’s arguments and Dr. Rubin’s opinions challenging POSAs’ expectation of success in modifying Orsini to use and manage KEKs, I further cite EX1106. As I discussed above (§III.A), EX1106 is a document published by the U.S. Department of Commerce’s National Institute of Standards and Technology (“NIST”), titled “**Recommendation for Key Management – Part 1: General.**” This document provided “basic key management guidance” to advise on the “‘best practices’ associated with key management.” EX1106, 5. The various key-management principles discussed in EX1106 include for KEKs for the standardized encryption techniques discussed in Orsini, like AES and DES. EX1106, 35-36.

237. Thus, the above corroborating evidence supports my opinion in my First Declaration that any potential drawbacks of using having to manage KEKs

were not beyond POSAs' skill and thus would not have discouraged POSAs from pursuing the use of a conventional KEK to secure Orsini's share keys.

238. But even if the alleged "drawbacks" to KEKs that SFI identifies were true, that would still not defeat have dissuaded POSAs from using this known technique. SFI does not dispute that using the known KEK technique in Orsini's system would have increased security because Orsini stores its share keys outside of Orsini's encryption device in unencrypted form, which the prior art taught "should **never** occur" for security reasons, as I discussed in my First Declaration (¶¶131-134) and above (§III-§III.A.3.a). *E.g.*, EX1035, 12:15-23.

239. Indeed, POSAs would have known that securing data often involved tradeoffs in "convenience and security, speed," etc. EX1081, 15:6-25. For example, for particularly if data to be secured were of high value, POSAs would have known that any (marginal) increase in cost or complexity of using the cryptographic protection of a KEK would have been desirable nonetheless given the (relatively weak) additional protection that Orsini's non-cryptographic obfuscation would have provided. Thus, even if using KEKs to encryption Orsini's share keys would have added some complexity (and I don't believe it would have given the already-existing structures in Orsini's system) that still would not have dissuaded POSAs from pursuing the use of a KEK in Orsini to secure its share key. That is because using and managing KEKs were well-known

to POSAs and were known to have practical application in a system (like Orsini's) that stores and transmits encryption keys.

240. Thus, in my opinion, encrypting Orsini's share keys would not have been beyond POSAs' capability, at least because it was nothing more than using a known technique (KEK-based encryption of keys) to address a known problem (the security risk of storing an unencrypted encryption key outside the cryptographic device), using prior art elements of KEK-based encryption according to their established function. Moreover, as I discussed in my First Declaration (§§131-134) and above (§III-§III.A.3.a), using the cryptographic protection of a KEK would have obtained security benefits that were not possible from Orsini's Figure 21 in the absence of using a KEK.

**c. My opinion KEK-based Obviousness Opinion was not based on hindsight, but was instead grounded in evidence that I cited.**

241. SFI contends that the Petition's KEK-based obviousness argument is driven by hindsight. POR, 29-32. To the extent SFI is contending that my opinions were also driven by hindsight, I do not agree that my opinions on the obviousness of using a KEK in Orsini's system to encrypt Orsini's share keys was based on hindsight. Instead, the combination of Orsini with conventional KEK-based protection that I discussed in my First Declaration (§§131-134) and further herein in response to SFI's argument, is based on a specific motivation—that I

supported by multiple *corroborating evidence*—to use a KEK for its well-known purpose to avoid the well-known (and unnecessary and avoidable) risk of having Orsini’s share keys stored outside the cryptographic device in non-encrypted form. *See* First Declaration, ¶132 (citing **EX1035**, 12:15-23, 10:42-62; **EX1036**, 1:66-2:6; *supra* §III-§III.A.3.b (citing additional corroborating evidence in response to SFI’s arguments and Dr. Rubin’s opinion demonstrating that POASs would have recognized based on already-existing principles and techniques the benefits of using a KEK to secure plaintext keys like Orsini’s share keys)).

242. Again, Orsini itself recognized a need to protect its share keys. Orsini elected to add some protection for its share keys by using a non-cryptographic obfuscation step, rather a conventional KEK technique. However, POSAs would have recognized that an alternative (or additional) and conventional well-known way to protect encryption keys like Orsini’s share keys would have been to use the cryptographic protection provided by a KEK. As I’ve discussed above (§III.A), POSAs would have recognized that using a KEK would have provided at least as much (and in many situations **more**) security as Orsini’s non-cryptographic obfuscation. POSAs would have recognized that even in an implementation that continued to use non-cryptographic obfuscation, adding the cryptographic protection of a KEK would have provided additional security. Indeed, the alleged invention of the ’854 is patent over Orsini is to further secure the shares with an

encryption operation based on an encryption key. If that limitation can be met by encrypting the share keys with a KEK (as SFI's infringement contentions allege), my First Declaration and this Reply Declaration have demonstrated with **multiple pieces of corroborating evidence** that using a KEK was already well-known by POSAs to provide the same additional security that SFI alleges its patent added over Orsini.

**B. Response to Argument that POSAs would not have a reasonable expectation of success in modifying Orsini to meet limitation [1C].**

243. In its POR, SFI asserts that POSAs would not have had a “reasonable expectation of success in modifying Orsini to achieve claim 1.” POR, 4. Though SFI makes that broad statement for all of the different obvious implementations of Orsini that meet the challenged claims, I note that SFI never identifies any specific reason **why** POSAs allegedly would not have had a reasonable expectation of success in using a conventional external KEK to encrypt encryption keys (like Orsini's), as was known in the art. SFI cites no independent evidence (beyond Dr. Rubin's opinions) to substantiate SFI's unexplained reasons why POSAs would not have reasonably expected success in modifying Orsini to use a conventional KEK to encrypt Orsini's share keys.

244. As I opined in my First Declaration, POSAs would have reasonably expected success in using a KEK in Orsini because doing was “conventional and

well-known to POSAs” and because Orsini itself says that additional steps may be performed. First Declaration, ¶134. That opinion was backed by the multiple corroborating references I cited that demonstrated that using and managing KEKs was well known and conventional in the prior art. First Declaration, ¶¶93, ¶¶131-132 (citing EX1035, EX1036).

245. In response to SFI’s argument that POSAs would not have reasonably expected success in using a KEK in Orsini, I note again that Orsini itself uses “Triple DES and AES” encryption techniques to encrypt the shares. Orsini, [0378]. As I discussed above (§III.A.2.a, §III.A.3.b), these were standardized encryption techniques that taught encrypting keys using KEKs to secure those keys for storage. *See* EX1094, [0056]-[0057] (“Triple-DES Key Wrap” (i.e., KEK)); EX1090, 19:47-61, 21:1-22:54 (noting “the National Institute of Standards and Technology (NIST) Advanced Encryption Standard (AES)” provides a “key wrapping function” (i.e., a KEK function)).

246. In response to SFI’s argument about POSAs expectation of success, my continuing opinion that POSAs would have reasonably expected success in using, managing, and storing KEKs in Orsini because it was a well-known, conventional technique is demonstrated by additional corroborating evidence.

247. In response to SFI’s argument that POSAs would not have reasonably expected success in using a KEK in Orsini, I cite EX1017. EX1017 (at 1:50-68)

(also discussed *supra* §III.A.3.b) discloses as background information that “**a key encryption key (KEK), and is used to encrypt encryption key variables for transit, transmission or storage**” and that “[i]n more advanced secure communication systems, encryption **key variables are stored and distributed electronically to communication units using a key management controller or KMC.**”

248. In response to SFI’s argument that POSAs would not have reasonably expected success in using a KEK in Orsini, I further cite EX1093. EX1093 (at [0013]) discloses an example of using “secure key management when separate devices are used for generating and utilizing the keys” (e.g., externally retrieved keys), and as part of that prior art key management there would be “a key manager that facilitates exchange of the KEK used to encrypt and decrypt the keys generated by the host processor.” *See also* EX1093, [0035]-[0039] (similar), [0057]-[0060].

249. In response to SFI’s argument that POSAs would not have reasonably expected success in using a KEK in Orsini, I further cite EX1095. EX1095 (at 1:32-47) discloses that it was POSAs’ “BACKGROUND:” to have “[c]ryptographic key managements schemes [that] are built on **commonly used...concepts, such as key wrapping**” (i.e., KEKs). EX1095 (at 3:10-15) recognizes that “KEK[s] allows keys to be exported **for storage.**”

250. Thus, consistent with my First Declaration, numerous prior art references corroborate my opinion that POSAs would have had a reasonable expectation of success in using a KEK in Orsini's system to encrypt Orsini's share keys. Indeed, using a KEK was based on deterministic mathematical processes (not guess work) that would have further bolstered POSAs' reasonable expectation of success.

251. In his declaration, Dr. Rubin asserts that Orsini's "*session* key management is very different than key management for longer-lived keys such as the other 'conventional encryption keys' Petitioner relies upon." EX2033, ¶78. Dr. Rubin asserts that the "longer lifespan" of longer-lived keys "makes potential compromise of a key-encryption key all the more consequential" and that a "key management solution for session keys would not be appropriate for the 'group key,' 'encryption key,' or 'key-encryption key'" and would require "necessarily" "different" "key management" than for the "session" keys. EX2033, ¶¶79-80. I disagree and in my opinion POSAs would disagree. As I discussed above, POSAs would have been well-aware of numerous solutions for key-management systems for a wide-variety of encryption keys, including conventional key-encryption keys (KEKs). As I discussed above (with corroboration), Orsini itself discloses using pre-existing encryption standards, such as AES and DES, which **already** included key management solutions for their encryption keys, which included KEK.

252. Moreover, one of Orsini's keys is a "Parser Master Key," which Orsini says is "an individual key associated with the installation of the data parser" that "is installed on the server" and "secur[ed]" in various ways including "separate hardware key store, standard key stores, customer key stores or within a secured database table." Orsini, [0399]-[0400]. That is, the Parser Master Key is an example of what Dr. Rubin is alleging is a longer-lived key and Orsini discloses that it **already** relies on existing, conventional key management systems to manage that key. POSAs would have known that a KEK could have (at least) been managed using the same systems. Orsini similarly notes that there are "a wide number of alternative key management systems," including those for "public-key system[s]." Orsini, [0107]. POSAs knew that public-private encryption key pairs were (and are) intended to be long-lived keys and that the key-management of public-key encryption systems was substantially more challenging than the management of a conventional, externally retrieved and stored (symmetric) KEK. But as even Orsini recognizes, the management of public-encryption keys had already been solved in the prior art. As I demonstrated above with corroborating evidence, the same was true of conventional KEKs like the ones I opined would have been used in an obvious modification of Orsini.

253. Moreover, Dr. Rubin did not identify where the '854 patent discusses any of the alleged challenges in (let alone provide a solution to) key-management

problems for a key-encryption key, or any of the '854 patent's alleged external keys. That reinforces to POSAs that the '854 patent is relying on **existing** key-management systems and techniques for its external key, thus reinforcing their expectation of success in using the same or similar available key management systems for a KEK. In fact, Dr. Rubin testified at deposition that one of the patent's disclosed external keys is the session master key: "I agree that the session key is an external key." Rubin-Depo., 28. That admission contradicts his declaration opinion that Orsini's sessions keys are "very different than key management for longer-lived keys" (EX2033, ¶78) because at deposition he admitted that Orsini is **already** disclosing examples of an external key, and thus management thereof, and SFI's infringement contentions have mapped the external key to a KEK encryption key. *See, e.g.*, EX 1071, pdf. 47. That is, to the extent the '854 patent has described a key-management system for an *external key* that could be a KEK, POSAs would understand the same would have been true of Orsini, as I see no meaningful, additional disclosures of key management for external keys in the '854 patent compared to Orsini.

254. (I note, for clarity, that I have not been asked to consider whether the '854 patent discloses using a KEK in the manner required of the claimed *external key*, but that is a different question from whether POSAs would have known how

to manage such keys, whether or not those management systems were explicitly disclosed in Orsini.)

255. To POSAs, the management of a KEK was, from a technical point of view, little different from the management of any other encryption key in Orsini's system. In conventional key management of a KEK, the KEK would have been retrieved from an external storage system (e.g., a conventional key store from within or from outside the system),<sup>13</sup> and after being applied to Orsini's share keys, a record would have been kept (likely in an encryption form) about what KEK was used to encrypt those share keys, and then the KEK could have been returned / placed into storage or destroyed, with a record of where the KEK could be retrieved from when needed.

256. Thus, POSAs would not have agreed with Dr. Rubin's opinions on the alleged complexities and differences in management of allegedly shorter-lived sessions keys versus allegedly longer-lived KEK encryptions. Rather, to the extent Orsini did not already have in place key-management systems designed to include

---

<sup>13</sup> When retrieved from an outside system (and possible even when retrieved from an inside system), the KEK would itself be encrypted by another encryption key (e.g., the recipient's public key so that only that recipient could decrypt it using their private key).

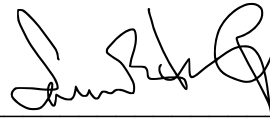
management of KEKs (and I believe they would have recognized that Orsini's key-management systems were already designed for that), POSAs would have recognized that it would have required only ordinary skill to do so using well-known, conventional key-management techniques and systems (much like SFI's patents rely on existing key-management solutions).

\* \* \*

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

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

Dated: December 23, 2024

A handwritten signature in black ink, appearing to read 'Samrat Bhattacharjee', is written above a horizontal line.

**Samrat Bhattacharjee, Ph.D.**