

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

INTERNATIONAL BUSINESS MACHINES CORPORATION
Petitioner

v.

SECURITY FIRST INNOVATIONS, LLC
Patent Owner

Case No. IPR2025-01200
U.S. Patent No. 8,271,802

PETITION FOR *INTER PARTES* REVIEW OF
U.S. PATENT NO. 8,271,802

CHALLENGING CLAIMS 1-27
UNDER 35 U.S.C. §312 AND 37 C.F.R. §42.104

TABLE OF CONTENTS

	<u>Page</u>
I. Introduction.....	1
II. Mandatory Notices.....	1
A. Real Party-In-Interest (37 C.F.R. §42.8(b)(1))	1
B. Related Matters (37 C.F.R. §42.8(b)(2)).....	1
C. Lead And Back-up Counsel (37 C.F.R. §42.8(b)(3)).....	1
D. Service Information (37 C.F.R. §42.8(b)(4))	2
III. Fees	3
IV. Certification Of Grounds For Standing	3
V. Overview Of Challenge And Relief Requested.....	3
A. Prior Art References	3
B. Public Accessibility of Krawczyk	4
C. Grounds For Challenge	4
VI. The '802 Patent.....	5
A. Prosecution History	7
VII. Level Of Ordinary Skill In The Art.....	7
VIII. Claim Construction.....	8
IX. Specific Grounds For Challenge.....	8
A. Grounds I : Torre And Tajima Render Obvious Claims 1-3, 5, 9-12, 14, 18-21, 23, 27.	8
1. Torre.....	8
2. Tajima	14
3. Torre In Combination With Tajima (“Torre/Tajima”)	16

Petition for *Inter Partes* Review of U.S. Patent No. 8,271,802
Claims 1-27

4.	Claim 1	29
5.	Claim 2	50
6.	Claims 3, 5	54
7.	Claim 9	56
8.	Claim 10	57
9.	Claim 19	58
10.	Claims 11-12, 14, 18, 20-21, 23, 27	59
B.	Ground II: Torre, Tajima, And Aldridge Render Obvious Claims 4, 13, 22.....	60
1.	Aldridge	60
2.	Torre/Tajima In Combination With Aldridge ("Torre/Tajima/Aldridge").....	62
3.	Claim 4	64
4.	Claims 13 and 22.....	66
C.	Ground III: Torre, Tajima, And Krawczyk Render Obvious Claims 6, 15, 24.....	67
1.	Krawczyk	67
2.	Torre/Tajima In Combination With Krawczyk ("Torre/Tajima/Krawczyk")	68
3.	Claim 6	72
4.	Claims 15 and 24.....	73
D.	Ground IV: Torre, Tajima, And Foster Render Obvious Claims 7-8, 16-17, 25-26.....	73
1.	Foster.....	73

2.	Torre/Tajima In Combination With Foster ("Torre/Tajima/Foster")	74
3.	Claim 7	78
4.	Claim 8	80
5.	Claims 16-17, 25-26	81
E.	Ground V: Orsini And Torre Render Obvious Claims 1-5, 9-14, 18-23, 27	81
1.	Orsini	81
2.	Orsini In Combination With Torre ("Orsini/Torre")	85
3.	Claim 1	90
4.	Claims 2-3, 5	96
5.	Claim 4	97
6.	Claim 9	97
7.	Claim 10	98
8.	Claim 19	99
9.	Claims 11-14, 18, 20-23, 27	99
F.	Ground VI: Orsini, Torre, And Krawczyk Render Obvious Claims 6, 15, 24	99
1.	Claim 6	99
2.	Claims 15 and 24	103
G.	Ground VII: Orsini, Torre, And Foster Render Obvious Claims 7-8, 16-17, 25-26	103
1.	Orsini/Torre In Combination With Foster	103
2.	Claim 7	106

Petition for *Inter Partes* Review of U.S. Patent No. 8,271,802
Claims 1-27

3.	Claim 8	107
4.	Claims 16-17, 25-26.....	108
X.	Conclusion	109

TABLE OF AUTHORITIES

Page(s)

Cases

<i>Keynetik, Inc. v. Samsung Elecs. Co., Ltd.</i> , No. 2022-1127, 2023 WL 2003932 (Fed. Cir. Feb. 15, 2023).....	passim
<i>Sequoia Tech., LLC v. Dell, Inc.</i> , 66 F.4th 1317 (Fed. Cir. 2023)	92

Statutes

35 U.S.C. § 102	3, 82
35 U.S.C. § 103	3, 5
35 U.S.C. §314(a)	5

Rules

37 C.F.R. § 42.104(b)(1)-(2).....	3
37 C.F.R. § 42.22(a)(1)	3

PETITIONER’S TABLE OF EXHIBITS

Exhibit No.	DESCRIPTION
1001	U.S. Patent No. 8,271,802 (’802 Patent)
1002	Declaration of Dr. Erez Zadok
1003	U.S. Patent Publication No. 2003/0065656 (“Torre”)
1004	U.S. Patent Publication No. 2003/0028493 (“Tajima”)
1005	U.S. Patent Publication No. 2004/0030921 (“Aldridge”)
1006	“Secret Sharing Made Short,” Hugo Krawczyk, Advances in Cryptology - CRYPTO ’93, 13th Annual International Cryptology Conference, LNCS 773, 136-146, 1993 (“Krawczyk”)
1007	U.S. Patent Publication No. 2003/0200176 (“Foster”)
1008	U.S. Patent Publication No. 2004/0049687 (“Orsini”)
1009	File History of U.S. Patent No. 8,271,802
1010	File History of U.S. Patent Application No. 11/258,839
1011	File History of U.S. Patent Application No. 13/371,363
1012	Comparison Of The ’802 Patent’s Specification And Orsini
1013	<i>Google, LLC v. Security First Innovations, LLC</i> , IPR2024-00212, Exhibit 1043 (Patent Owner’s Proposed Claim Constructions In <i>Security First Innovations, LLC v. Google, LLC</i> , No. 2:23-cv-00097 (E.D. Va.)) (“PO’s Google Litigation Construction”)
1014	U.S. Provisional Application No. 60/622,146
1015	U.S. Provisional Application No. 60/718,185
1016	Microsoft Computer Dictionary, 5 th ed., 2002, excerpts (“Microsoft Computer Dictionary”)
1017	Bruce Schneier, Applied Cryptography, 2 nd ed., 1996, excerpts (“Schneier”)
1018	Curriculum Vitae of Dr. Erez Zadok
1019	Yitzhak Birk, “Random RAIDs with Selective Exploitation of Redundancy for High Performance Video Servers,” IEEE 1997 (“Birk”)
1020	U.S. Patent Publication No. 2003/0016596A1 (“Chiquoine”)
1021	Declaration of Dr. Hall-Ellis (“Hall-Ellis Decl.”)
1022	John Kubiawicz, et al., “OceanStore: An Architecture for Global-Scale Persistent Storage,” ACM 2000 (“Kubiawicz”)
1023	U.S. Patent No. 6,226,618 (“Downs”)

Petition for *Inter Partes* Review of U.S. Patent No. 8,271,802
Claims 1-27

Exhibit No.	DESCRIPTION
1024	P. Venkat Rangan et al., “Designing File Systems for Digital Video and Audio,” ACM SIGOPS Operating Systems Review, Volume 25, Issue 5, 81–94 (1991) (“Rangan”)

Throughout the Petition, all annotations, coloring, and emphases have been added unless indicated otherwise.

LISTING OF CLAIMS

Claim 1

[1Pre] A method for securing a data set, the method steps implemented by a programmed computer system, the method steps comprising:

[1A] creating integrity information using the data set;

[1B] encrypting the data set based on an encryption key to produce an encrypted data set;

[1C] generating data splitting information, wherein the data splitting information is usable to determine into which of a plurality of shares of data a unit of data of the encrypted data set will be placed;

[1D] separating the encrypted data set into the plurality of shares based on the data splitting information;

[1E] including in the plurality of shares data indicative of (a) the encryption key and (b) the integrity information; and

[1F] causing the plurality of shares to be stored in respective separate storage locations;

[1G] wherein the data set is restorable by accessing less than all, but at least a threshold number of, the plurality of shares.

Claim 2

[2] The method of claim 1, wherein the step of separating the encrypted data set comprises using a deterministic technique.

Claim 3

[3] The method of claim 1, wherein the step of separating the encrypted data set comprises using a substantially random technique.

Claim 4

[4] The method of claim 1, further comprising causing a plurality of data units in each of the shares to be rearranged relative to one another after the separating step.

Claim 5

[5] The method of claim 1, wherein the step of separating the encrypted data into the plurality of shares comprises causing the plurality of shares to have a substantially random distribution of the encrypted data set.

Claim 6

[6] The method of claim 1, wherein the data indicative of the encryption key comprises data created using a Shamir algorithm.

Claim 7

[7] The method of claim 1, wherein the step of including data indicative of the encryption key comprises:

[7A] encrypting the encryption key with a second key; and

[7B] including in the plurality of shares data indicative of the encrypted encryption key.

Claim 8

[8] The method of claim 7 further comprising the method step of storing the second key outside of the plurality of shares, wherein the data set is restorable by accessing less than all, but at least a threshold number of, the plurality of shares, and the second key.

Claim 9

[9] The method of claim 1, wherein the step of storing the shares in respective separate storage locations comprises storing the shares on at least two separate storage devices.

Claim 10

[10Pre] A non-transitory computer readable medium storing computer executable instructions that, when executed by at least one processor, cause a computer system to carry out a method for securing a data set, the method comprising the steps of:

[10A] creating integrity information using the data set;

[10B] encrypting the data set based on an encryption key to produce an encrypted data set;

[10C] generating data splitting information, wherein the data splitting information is usable to determine into which of a plurality of shares of data a unit of data of the encrypted data set will be placed;

[10D] separating the encrypted data set into the plurality of shares based on the data splitting information;

[10E] including in the plurality of shares data indicative of (a) the encryption key and (b) the integrity information; and

[10F] causing the plurality of shares to be stored in respective separate storage locations;

[10G] wherein the data set is restorable by accessing less than all, but at least a threshold number of, the plurality of shares.

Claim 11

[11] The non-transitory computer readable medium of claim 10, wherein the step of separating the encrypted data set comprises using a deterministic technique.

Claim 12

[12] The non-transitory computer readable medium of claim 10, wherein the step of separating the encrypted data set comprises using a substantially random technique.

Claim 13

[13] The non-transitory computer readable medium of claim 10, wherein the method further comprises causing a plurality of data units of the encrypted data set to be rearranged relative to one another after the separating step.

Claim 14

[14] The non-transitory computer readable medium of claim 10, wherein the step of separating the encrypted data into the plurality of shares comprises causing the plurality of shares to have a substantially randomly distribution of the encrypted data set.

Claim 15

[15] The non-transitory computer readable medium of claim 10, wherein the data indicative of the encryption key comprises data created using a Shamir algorithm.

Claim 16

[16] The non-transitory computer readable medium of claim 10, wherein the step of including data indicative of the encryption key comprises:

[16A] encrypting the encryption key with a second key; and

[16B] including in the plurality of shares data indicative of the encrypted encryption key.

Claim 17

[17] The non-transitory computer readable medium of claim 16 wherein the method further comprises storing the second key outside of the plurality of shares, wherein the data set is restorable by accessing less than all, but at least a threshold number of, the plurality of shares, and the second key.

Claim 18

[18] The non-transitory computer readable medium of claim 10, wherein the step of storing the shares in respective separate storage locations comprises storing the shares on at least two separate storage devices.

Claim 19

[19Pre] A computer system for securing a data set, the system comprising:

[19A] at least one processor;

[19B] a non-transitory computer readable medium storing computer executable instructions that, when executed by the at least one processor, cause the computer system to carry out a method for securing a data set, the method comprising the steps of:

[19C] creating integrity information using the data set;

[19D] encrypting the data set based on an encryption key to produce an encrypted data set;

[19E] generating data splitting information, wherein the data splitting information is usable to determine into which of a plurality of shares of data a unit of data of the encrypted data set will be placed;

[19F] separating the encrypted data set into the plurality of shares based on the data splitting information;

[19G] including in the plurality of shares data indicative of (a) the encryption key and (b) the integrity information; and

[19H] causing the plurality of shares to be stored in respective separate storage locations;

[19I] wherein the data set is restorable by accessing less than all, but at least a threshold number of, the plurality of shares.

Claim 20

[20] The system of claim 19, wherein the step of separating the encrypted data set comprises using a deterministic technique.

Claim 21

[21] The system of claim 19, wherein the step of separating the encrypted data set comprises using a substantially random technique.

Claim 22

[22] The system of claim 19, wherein the method further comprises causing a plurality of data units of the encrypted data set to be rearranged relative to one another after the separating step.

Claim 23

[23] The system of claim 19, wherein the step of separating the encrypted data into the plurality of shares comprises causing the plurality of shares to have a substantially random distribution of the encrypted data set.

Claim 24

[24] The system of claim 19, wherein data indicative of the encryption key comprises data that was created using a Shamir algorithm.

Claim 25

[25] The system of claim 19, wherein the step of including data indicative of the encryption key comprises:

[25A] encrypting the encryption key with a second key; and

[25B] including in the plurality of shares data indicative of the encrypted encryption key.

Claim 26

[26] The system of claim 25 wherein the method further comprises storing the second key outside of the plurality of shares, wherein the data set is restorable by

accessing less than all, but at least a threshold number of, the plurality of shares, and the second key.

Claim 27

[27] The system of claim 19, wherein the separate storage locations are located on at least two separate storage devices.

I. Introduction

Petitioner respectfully requests cancellation of all claims in U.S. Patent No. 8,271,802 (“’802 Patent”).

II. Mandatory Notices

A. Real Party-In-Interest (37 C.F.R. §42.8(b)(1))

Petitioner, International Business Machines Corporation (“IBM”), identifies itself as a real party-in-interest (“RPI”).

B. Related Matters (37 C.F.R. §42.8(b)(2))

The ’802 Patent is currently asserted against Petitioner by Patent Owner (*Security First Innovations, LLC*) in *Security First Innovations, LLC v. International Business Machines Corp.*, No. 1:25-cv-00514 (EDVA) (“Litigation”). Petitioner has filed or plans to file petitions for *inter partes* review of U.S. Patent Nos. 8,904,194 and 9,135,456, each of which is being asserted against Petitioner in the Litigation.

C. Lead And Back-up Counsel (37 C.F.R. §42.8(b)(3))

Petitioner is filing a Power of Attorney appointing the practitioners associated with Customer Number 132,593. Petitioner designates the following lead and back-up counsel:

Lead Counsel	First Back-up Counsel
Taeg Sang Cho (Reg. No. 69,618) Desmarais LLP 230 Park Ave New York, NY 10169 Telephone: (212) 351-3400 Email: tcho@desmaraisllp.com	Benjamin Rodd (Reg. No. 79,158) Desmarais LLP 230 Park Ave New York, NY 10169 Telephone: (212) 351-3400 Email: brodd@desmaraisllp.com
Back-up Counsel	Back-up Counsel
Lindsey Miller (<i>pro hac vice</i> forthcoming) Desmarais LLP 230 Park Ave New York, NY 10169 Telephone: (212) 351-3400 Email: lmiller@desmaraisllp.com	Laura Avena (Reg. No. 78,920) Desmarais LLP 230 Park Ave New York, NY 10169 Telephone: (212) 351-3400 Email: lavena@desmaraisllp.com

D. Service Information (37 C.F.R. §42.8(b)(4))

Post and hand delivery: Desmarais LLP

230 Park Ave, New York, NY 10169

Telephone: 212-351-3400

Email: IBM-SFI-IPR-Service@desmaraisllp.com

Please address all correspondence to counsel identified above. Petitioner consents to electronic service by email at:

tcho@desmaraisllp.com

brodd@desmaraisllp.com

lmiller@desmaraisllp.com;

lavena@desmaraisllp.com; and

IBM-SFI-IPR-Service@desmaraisllp.com

III. Fees

Petitioner is concurrently electronically submitting the required fees for this Petition. The Board is authorized to charge Desmarais LLP's deposit account, No. 50-6822, for any fee deficiency.

IV. Certification Of Grounds For Standing

Petitioner certifies that the '802 Patent is available for *inter partes* review and that Petitioner is not barred or estopped from requesting *inter partes* review.

V. Overview Of Challenge And Relief Requested

Under 37 C.F.R. §§42.22(a)(1) and 42.104(b)(1)-(2), Petitioner requests cancellation of claims 1-27 ("Challenged Claims") of the '802 Patent.

A. Prior Art References

This Petition applies pre-AIA provisions of 35 U.S.C. §§ 102, 103. The following references are pertinent to the grounds of unpatentability:

Pat. Pub. No. or Title	Publication/Priority Date	Prior Art Under At Least (35 U.S.C.)
US 2003/0065656 ("Torre") (EX1003)	published April 3, 2003	102(b)
US 2003/0028493 ("Tajima") (EX1004)	published February 6, 2003	102(b)
US 2004/0030921 ("Aldridge") (EX1005)	published February 12, 2004	102(a)
"Secret Sharing Made Short" ("Krawczyk") (EX1006)	published as early as January 4, 1994	102(b)
US 2003/0200176 ("Foster") (EX1007)	published October 23, 2003	102(b)

Pat. Pub. No. or Title	Publication/Priority Date	Prior Art Under At Least (35 U.S.C.)
US 2004/0049687 ("Orsini") (EX1008)	published March 11, 2004	102(a)

B. Public Accessibility of Krawczyk

Krawczyk was publicly available at least since January 4, 1994—more than one year before the earliest possible priority date of the '802 Patent (October 25, 2004). EX1021, ¶¶44-51.

For instance, as explained by Dr. Sylvia Hall-Ellis, Krawczyk was part of volume 773 of the Lecture Notes in Computer Science (LNCS) series, titled *Advances in Cryptology: CRYPTO '93*, found at the Linda Hall Library of Science, Engineering & Technology, which was "publicly available ... as of January 4, 1994." EX1021, ¶46. Indeed, the print version of this book "had been received, cataloged, and indexed at the Library of Congress, made part of the OCLC bibliographic database, and received in the Linda Hall Library of Science, Engineering & Technology" "on or shortly after January 4, 1994." EX1021, ¶50. *See also* EX1021, ¶¶46-49. Krawczyk is also available online from the publisher Springer-Verlag and in the Association for Computing Machinery (ACM) Digital Library, which indicates that the Krawczyk reference published on August 22, 1993. EX1021, ¶45.

C. Grounds For Challenge

Petitioner requests cancellation of Claims 1-27 of the '802 Patent under 35

U.S.C. §103 based on the following Grounds.

Grounds	Claims	Prior Art References
I	1-3, 5, 9-12, 14, 18-21, 23, 27	Torre, Tajima
II	4, 13, 22	Torre, Tajima, Aldridge
III	6, 15, 24	Torre, Tajima, Krawczyk
IV	7-8, 16-17, 25-26	Torre, Tajima, Foster
V	1-5, 9-14, 18-23, 27	Orsini, Torre
VI	6, 15, 24	Orsini, Torre, Krawczyk
VII	7-8, 16-17, 25-26	Orsini, Torre, Foster

This Petition demonstrates that there is a reasonable likelihood that Petitioner would prevail with respect to at least one of the challenged claims. 35 U.S.C. §314(a).

VI. The '802 Patent

The '802 Patent is directed to a “secure data parser” that “parses data and then splits the data into multiple portions that are stored or communicated distinctly.” EX1001, Abstract. Relevant to the '802 Patent's claims, Figure 21 (below) “show[s] the steps of the process performed by the secure data parser on data to store the session master key with the parsed data.” EX1001, 55:22-25. For example, Figure 21 discloses: (1) generating a “session master key” (shown in magenta); (2) encrypting a data set using the session master key (shown in lime green); (3) “[s]eparating the resulting encrypted data [set] into four shares or portions of parsed data” (shown in blue); (4) similarly separating the session master key into “key shares” (shown in red); and (5) “appending the key data [shares] to the encrypted

parsed data” (shown in orange and magenta) to form data shares. EX1001, 55:22-39. The data shares with key data are further processed and stored in storage devices. EX1001, 58:13-43, 58:55-62. EX1002, ¶¶148-149.

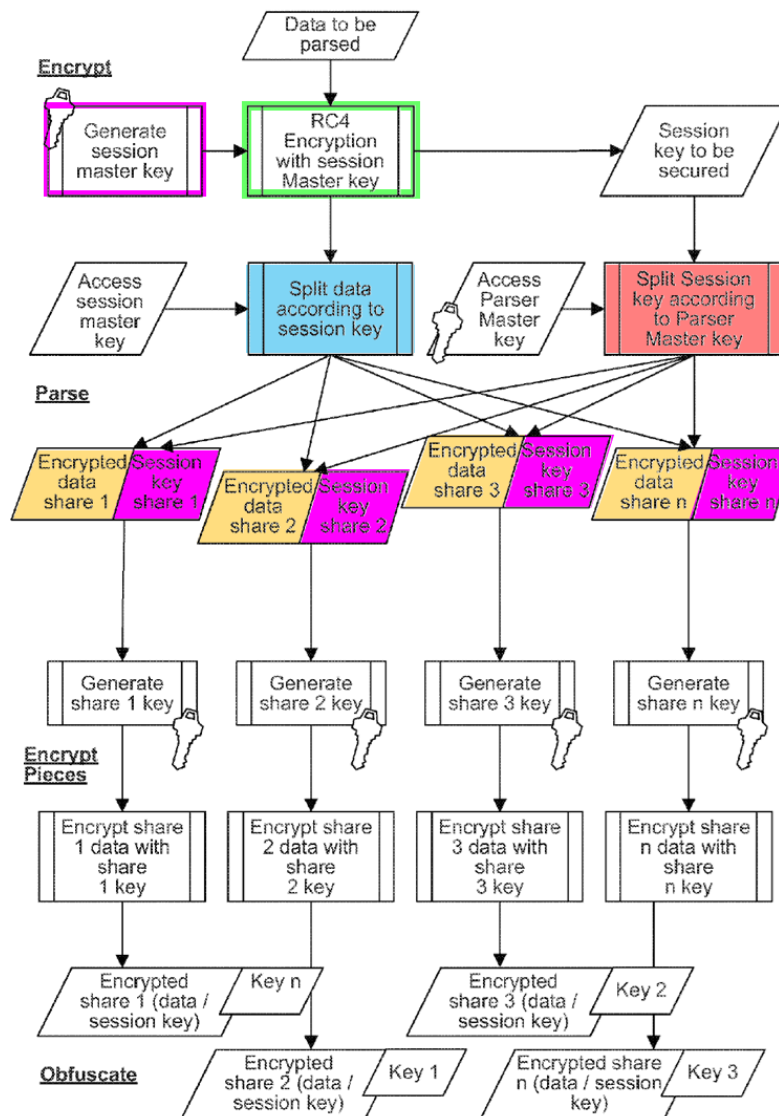


FIG. 21

EX1001, Figure 21.

The '802 Patent refers to the process of splitting the data into shares as “[a] cryptographic split (cryptosplit).” EX1001, 52:40-41. Its cryptosplit process

“provide[s] sufficient redundancy in the shares such that only a subset of the shares are needed to reassemble or restore the data to its original or useable form.” EX1001, 53:15-19. *See also* EX1001, 53:28-54:40. EX1002, ¶¶150-151.

A. Prosecution History

The '802 Patent originated from U.S. Application No. 13/371,361 (“’361 App.”), filed February 10, 2012, which is a continuation of U.S. Application No. 11/258,839 (“’839 App.”), filed October 25, 2005, which, in turn, claims priority to Provisional Applications 60/622,146 (filed October 25, 2004) and 60/718,185 (filed September 16, 2005). EX1009, 249; EX1010. EX1002, ¶152.

The '802 Patent’s claims issued after one round of Office Action without any amendment. The Examiner issued a Non-Final Office Action rejecting the originally-filed claims. EX1009, 324-328. In response, Applicant argued without any amendment that the claims “improve[] upon [the prior art] by including, in a plurality of shares, data indicative of integrity information created from a data set.” EX1009, 561. Subsequently, the Examiner allowed the claims unamended. EX1009, 569-576. EX1002, ¶¶153-154.

VII. Level Of Ordinary Skill In The Art

A person of ordinary skill in the relevant field or art (“POSITA”) as of the earliest claimed priority date of the '802 Patent would have had a Bachelor’s degree in Computer Science, Computer Engineering, Electrical Engineering, or an

equivalent field, and about 2-3 years of experience in the fields of data storage and security. Less professional experience can be substituted by additional education, and vice versa. EX1002, ¶155.

VIII. Claim Construction

Petitioner submits that no terms require construction for the purposes of this IPR.¹ EX1002, ¶¶157-159.

IX. Specific Grounds For Challenge

A. Grounds I : Torre And Tajima Render Obvious Claims 1-3, 5, 9-12, 14, 18-21, 23, 27.

1. Torre

Torre discloses “[a] system and method for shredding and deshredding [data] ... to store and retrieve [the] data while emphasizing considerations such as redundancy, integrity, and security.” EX1003, [0049]. Specifically, Torre teaches transforming the input data “into smaller pieces of data called shreds” (EX1003, [0051]) and “writing the shredded input data ... [in] multiple storage units” (EX1003, [0052]). “Shredding is done in a manner that adds redundancy to the shreds so that the input data can be recovered from a fewer number of shreds than the total number stored.” EX1003, [0051]. EX1002, ¶163.

¹ Petitioner reserves all rights to raise claim construction arguments in other forums.

Torre explains that its “shredder” shreds input data using a series of processing steps, as shown in Figure 4. EX1003, [0063]. *See also* EX1003, [0055]-[0057]. For example, when the shredder receives input data (step 122), it “performs first transformations (step 132).” EX1003, [0061]. The shredder then “allocates data transformed by the first transformations into various different blocks of data (step 133)” called “pre-shred[s]” (EX1003, [0061], [0080]) by “split[ting] up the [transformed] data” (EX1003, [0062]). Torre subsequently performs a second transformation (step 134) and “further splits the units of data into smaller sized units of data and generates additional redundant data (step 135) so that fewer pieces of stored data are necessary to restore the stored input data.” EX1003, [0061]. The shredder then completes the shredding process by performing a third transformation (step 136). EX1003, [0061]-[0062]. EX1002, ¶¶164-168.

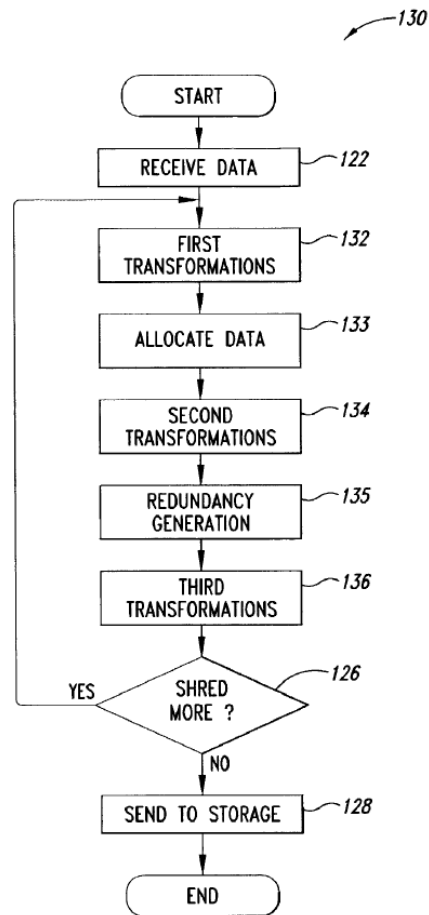


Fig. 4

EX1003, Figure 4.

As the shredder processes input data using various steps, the shredder appends “tags” to the processed data “to keep track of what is been done to the pieces of the input data 139 as the pieces of the input data move through the shredder.” EX1003, [0078]. “Information [in the tags] describing how the shreds were created is used to reassemble the shreds through processes inverse to those used to create the shreds.” EX1003, [0053]. EX1002, ¶¶169-170.

Torre provides one implementation of a shredder in Figure 34. There, the first transformer 174 “uses a SHA-1 signature generation function to generate a signature associated with the received data to generate a pre-shred 466 containing the received data and a tag of the SHA-1 signature” (EX1003, [0131]); the allocator 176 “splits the pre-shred 466 into three data blocks 494, which are sent on to three redundancy generators [180/]468 each located in three of the second-tier components 164” (EX1003, [0135]); and the redundancy generator 180 generates six redundant pre-shreds 470 from each data block 494. EX1002, ¶171.

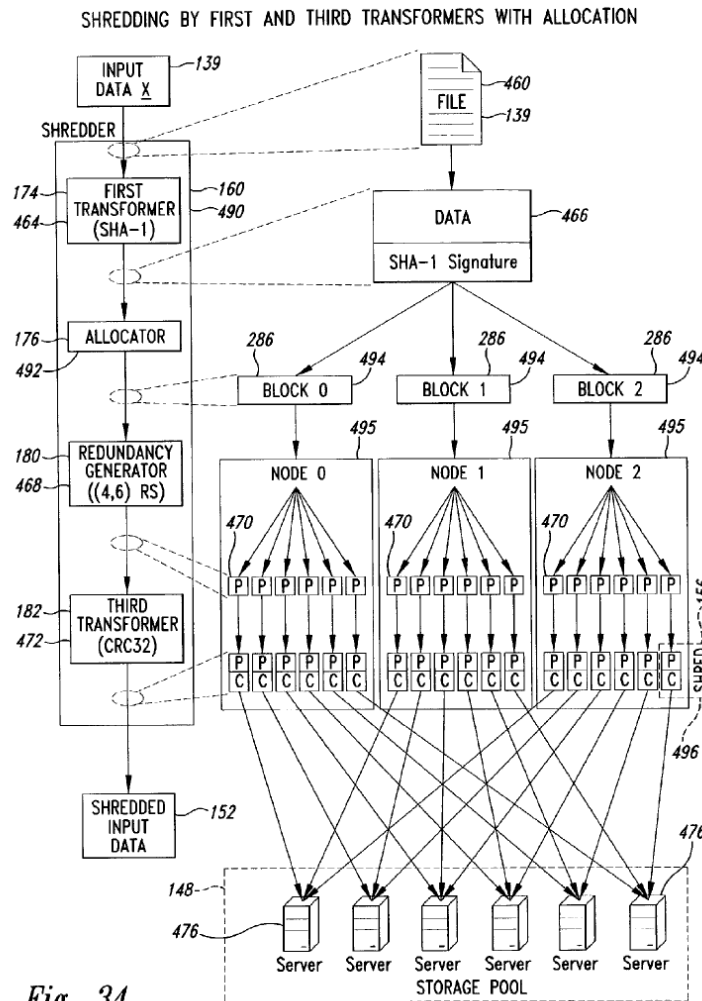


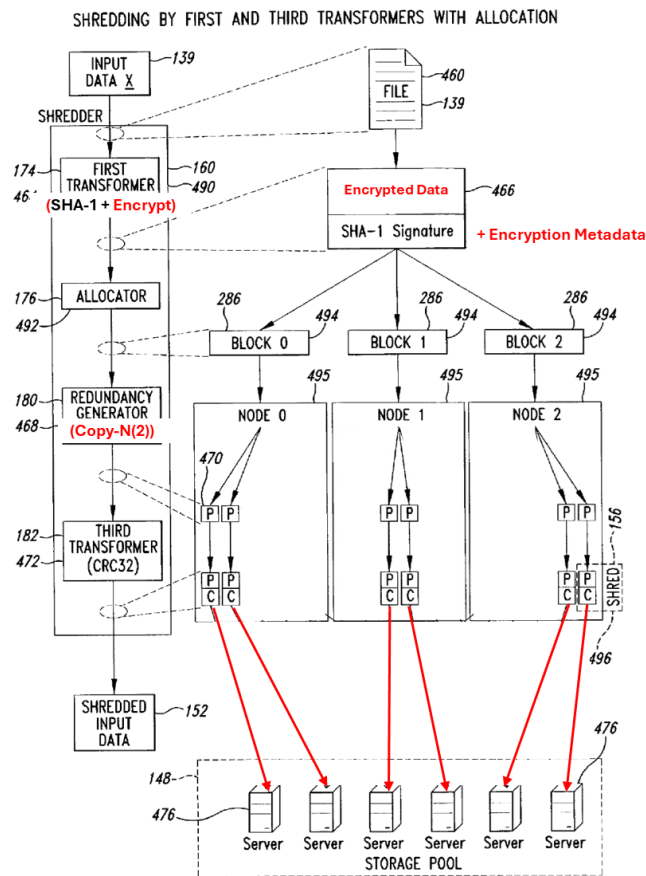
Fig. 34

EX1003, Figure 34.

Torre explains that the first transformer and the redundancy generator can be implemented in a variety of ways. For example, the first transformer can perform digital signature generation and encryption, and append a tag for both the signature generation and encryption steps. EX1003, Figure 11 (Rule 9), [0091]-[0092]. Also, the redundancy generator may use “[a] redundancy transformation called *Copy-N* ...

to simply make multiple exact copies of existing data.” EX1003, [0111], [0113].
EX1002, ¶172.

The following figure illustrates such variations to the first transformer and the redundancy generator in Figure 34, which is pertinent to the invalidity of the challenged claims. This variation is referred herein as the “Torre System.” EX1002, ¶¶173-174.



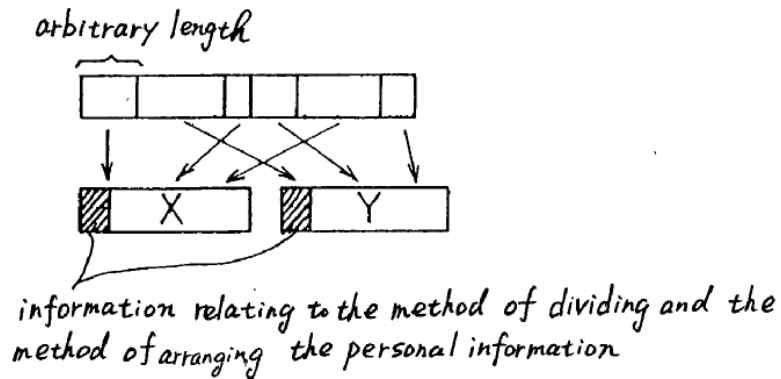
EX1003, Figure 34 (modified).

2. Tajima

Tajima discloses “a personal information management method ... that can improve the security of personal information that is registered with areas that are connectible to the Internet.” EX1004, [0044]; [0048]. Specifically, Tajima teaches “divid[ing] [] personal information into a plurality of data portions,” and storing the data portions in different servers “that are under [separate] control.” EX1004, [0046]. EX1002, ¶¶175-176.

Tajima discloses several “method[s] of dividing personal information.” EX1004, [0100], [0106]. For example, Tajima teaches, in connection with Figure 10 (below), that “personal information ... is first divided into a plurality of data portions *each of different arbitrary length.*” EX1004, [0106]. “Because the personal information is divided into data portions of different arbitrary lengths in this case, *the arbitrary lengths, which is information relating to the method of dividing and method of arranging*, must be attached to each of data fragments X and Y.” EX1004, [0107]. EX1002, ¶¶177-178.

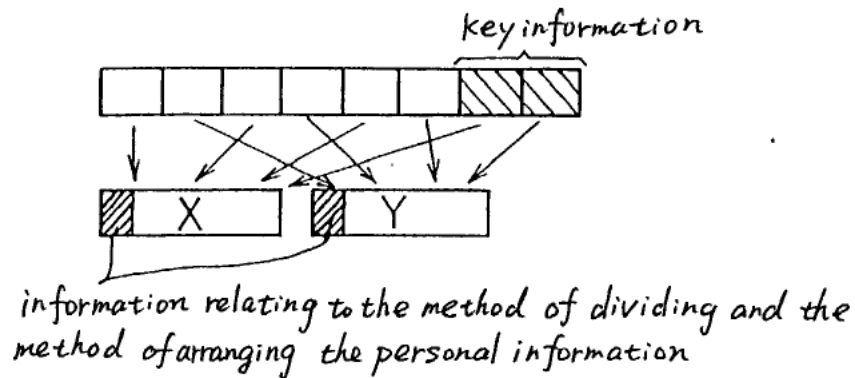
Fig. 10



EX1004, Figure 10.

The method of Figure 10 can be applied to encrypted data as well. For example, as illustrated in Figure 11, “personal information ... is first encrypted, and the *encrypted personal information* and *information regarding the key* that is used in the encryption are then divided into a plurality of data portions each of ... arbitrary length as shown in ... FIG. 10.” EX1004, [0109]. Claim 8 of Tajima explains that the “information regarding the key” may be the “*encrypted key*” used to encrypt the personal information. EX1004, Claim 8. EX1002, ¶¶179-180.

Fig. 11



EX1004, Figure 11.

3. Torre In Combination With Tajima (“Torre/Tajima”)

i. Torre System In View Of A Tag That Includes An Encryption Key, As Taught By Tajima

In Torre, when a shredder’s first transformer processes input data using a first transform function, it generates a **tag** describing that function and appends the **tag** to the **processed data** to form a **pre-shred**, as shown in Figure 14 (below). EX1003, [0078], [0102]. The **tag** is later used by the de-shredder to undo the first transform function and reconstruct the input data. EX1003, [0078]. “For instance, if [] encryption [] was done by [the first] transformer of the shredder 160, then [] decryption [] would be performed by the inverse transformer of the deshredder 390 corresponding to the [first] transformer of the shredder.” EX1003, [0123]. EX1002, ¶¶185-187.

ALLOCATOR TRANSFORMATION AND TAGGING

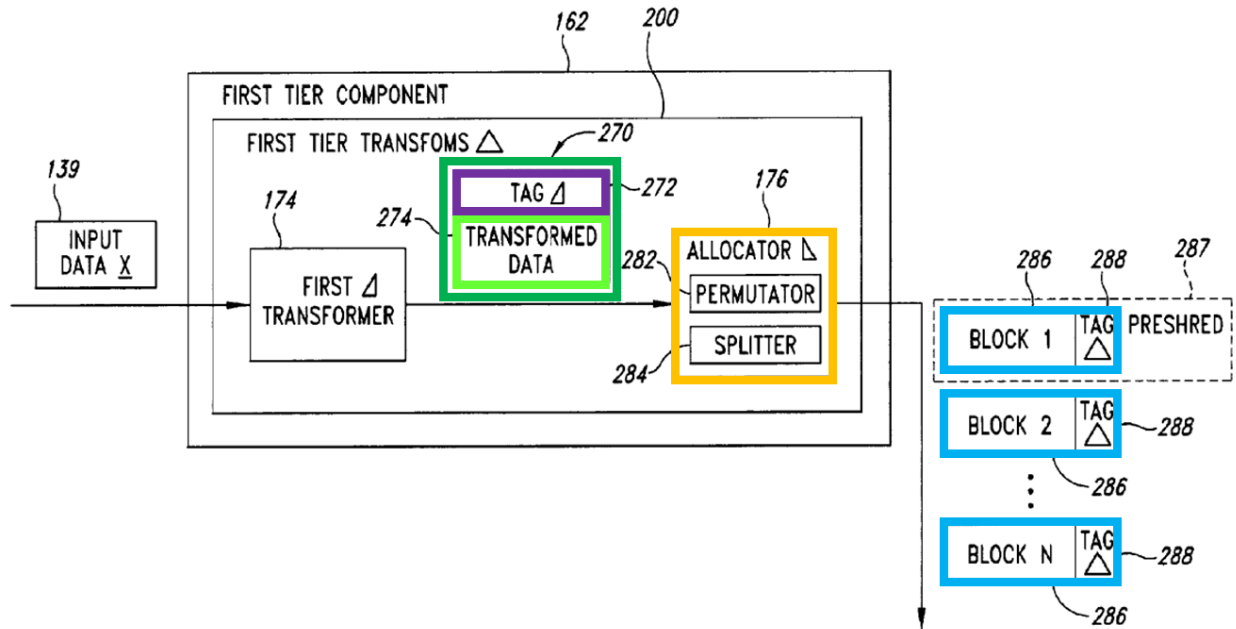


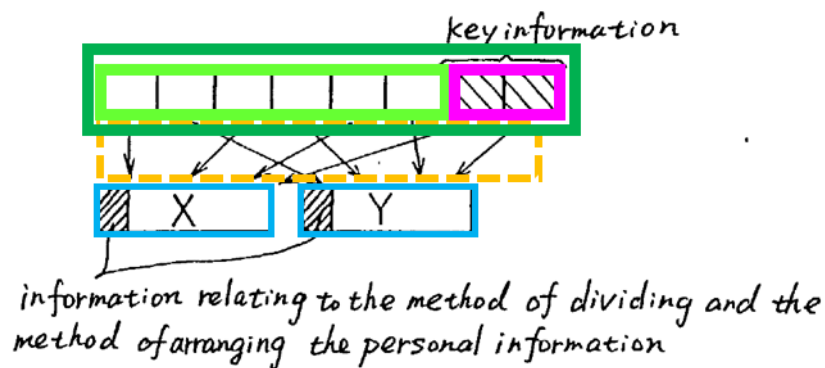
Fig. 14

EX1003, Figure 14.

If the first transformer encrypts the input data, the tag may contain an “identifier[]” of an encryption function. EX1003, [0079] (“Each of the tags 110 can also contain *identifiers of the particular functions* performed by the shredder 160 ... For example, an *encryption function* ...”). A POSITA would have understood that the tag identifying an encryption function is indicative of an encryption key because otherwise, it would not be possible to undo the encryption operation using the de-shredder as taught by Torre. EX1003, [0123]. EX1002, ¶188.

At a minimum, it would have been obvious to modify Torre to include the encryption key in the tag because that was well known. For example, Tajima teaches, in connection with Figure 11, encrypting input data with an **encryption key**; appending the **encrypted input data** with the **encryption key**; and splitting the **collection** of the **encrypted input data** and the **encryption key** into **data fragments**, as shown below. EX1004, [0109]-[0110]; Claim 8. EX1002, ¶¶189-190.

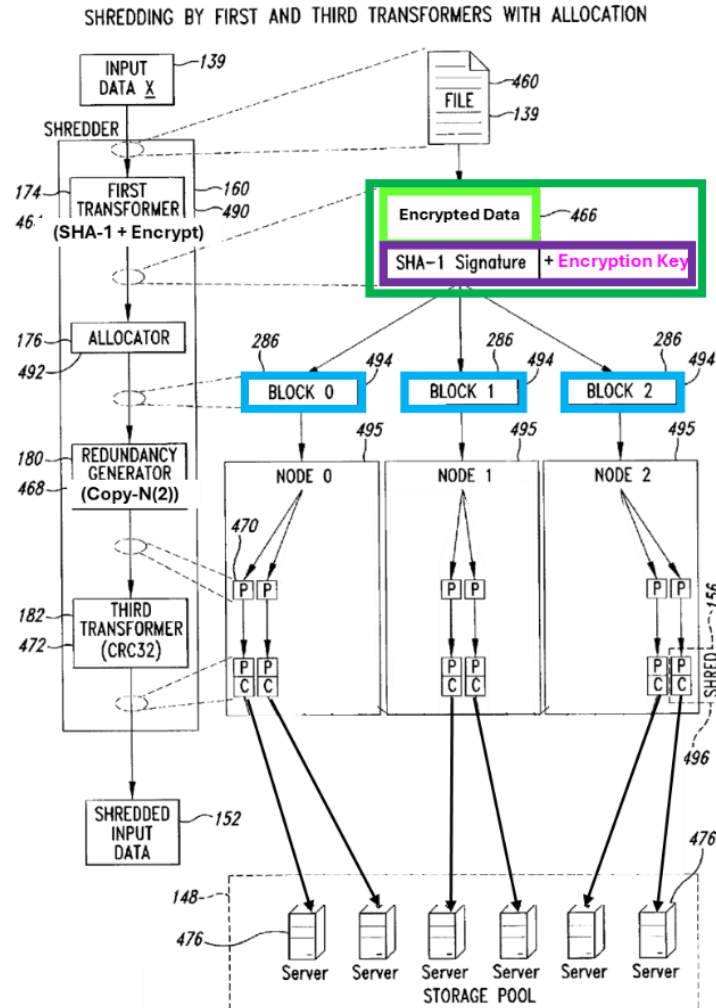
Fig. 11



EX1004, Figure 11.

It would have been obvious to combine Torre with Tajima so that when Torre's first transformer performs an encryption function, the **tag** includes the **encryption key**. This specific combination is referred to as the "Torre/Tajima System-1." EX1002, ¶¶191-192.

Torre/Tajima System-1



A POSITA would have been motivated to combine Torre with Tajima as shown in Torre/Tajima System-1. **First**, both Torre and Tajima are directed to solving a substantially similar problem in a substantially similar way. Both Torre and Tajima are directed to securely storing data in multiple storage devices. EX1003, [0049]; EX1004, [0044], [0048]. And both Torre and Tajima teach a substantially similar method to achieve that goal. EX1003, [0051]-[0052], [0061]-[0062];

EX1004, [0109]. Accordingly, a POSITA would have understood that Tajima's teachings are readily applicable to Torre's shredder. EX1002, ¶¶193-199.

Second, a POSITA would have been motivated to include the encryption key in the tag so that the de-shredder can use it later to decrypt the encrypted data (EX1003, [0078], [0123]). In Torre, the tag generated by the first transformer is used by the de-shredder to undo the operations previously performed on the input data to reconstruct the input data. EX1003, [0078], [0123]. Thus, when the first transformer encrypts data using an encryption key of, for example, "the Data Encryption Standard (DES)" (EX1003, [0085]), a POSITA would have understood that the deshredder will need the encryption key to decrypt the encrypted data because the DES is a symmetric encryption process that uses the same key for encryption and decryption. EX1023, 12:44-58. EX1002, ¶¶200-201.

This combination would have been particularly obvious because it would have merely amounted to using a known technique (Tajima's technique for appending encryption key to encrypted data) to improve a similar device (Torre System) in the same way of providing the encryption key as a tag appended to the encrypted data so that the deshredder can use it to decrypt the encrypted data. EX1002, ¶202.

Third, Torre explains that a tag "can be represented as metadata" and "can include the transformation steps taken, [and] *input parameters used*." EX1003, [0056]. Thus, when the first transformer performs encryption, a POSITA would

have sought to include, in the tag, input parameters associated with the encryption function, such as an encryption key. And Tajima teaches how to do so: including the encryption key as a tag to the encrypted data. EX1004, Figure 11, [0109]. EX1002, ¶203.

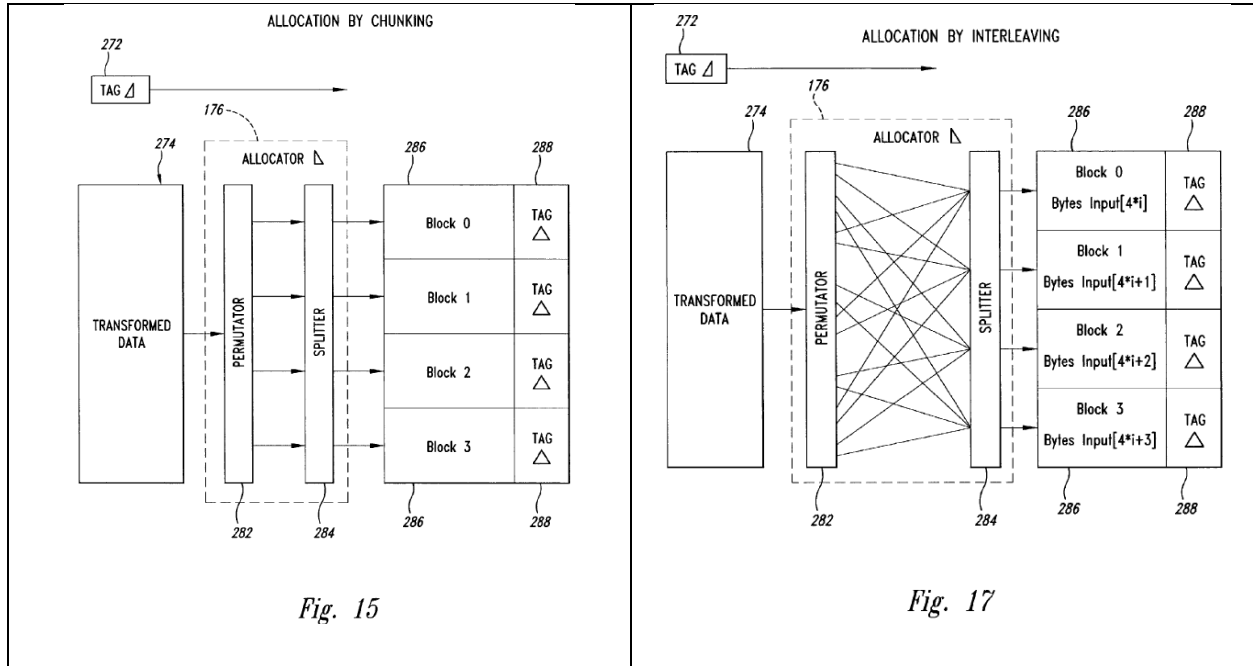
A POSITA would have had a reasonable expectation of success in providing an encryption key as part of a tag. **First**, Torre uses the tag to “indicate which functions should be performed in [the first transformer stage] of reconstructing the input data [] from the shredded input data [].” EX1003, [0078]. When Torre’s first transformer performs encryption, the tag for the first transformer would indicate how the input data was encrypted so that the first transformer in the deshredder can decrypt the encrypted data. EX1003, [0078]; [0123]. In fact, Tajima appends the encryption key to the encrypted data so that during the data reconstruction process, the method can use the encryption key to “decrypt the encrypted [data],” acting just like the tag in Torre. EX1004, [0110]. EX1002, ¶¶204-206, 208.

Second, Torre explains that “the embodiments disclosed herein ... can be equivalently implemented ... as one or more *computer programs* running on one or more computers.” EX1003, [0141]. Accordingly, implementing Tajima’s teachings in Torre System would have merely involved writing a program code. That was well within the skill of a POSITA. EX1003, [0141]; *Keynetik, Inc. v. Samsung Elecs. Co., Ltd.*, No. 2022-1127, 2023 WL 2003932, at *2 (Fed. Cir. Feb. 15, 2023)

(“Normally, once the function to be performed by software has been identified, writing code to achieve that function is within the skill of the art.”). EX1002, ¶207.

ii. Torre/Tajima System-1 In View Of Tajima’s Data Remapping And Splitting Method

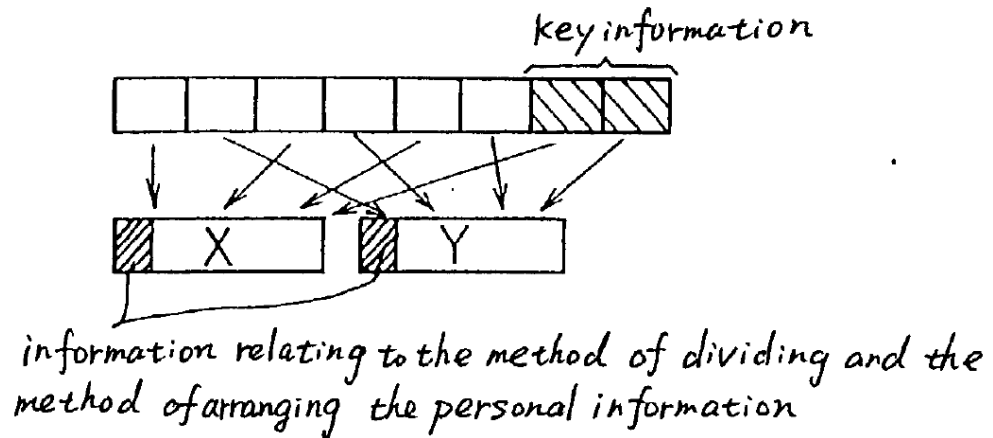
Torre’s shredder includes an allocator that “partition[s] the received data” and “split[s]” the partitioned data into data blocks. EX1003, [0099], [0102], Figure 34. As shown in Figures 15 and 17, the allocator includes a “permutator 282 [that] re-maps the ordering of the transformed [received] data 274 using” (EX1003, [0102]), for example, “chunking [and] interleaving” (EX1003, [0100]). The allocator also includes a “splitter 284 [that] splits the re-mapped data into data blocks 286.” EX1003, [0102]. Torre/Tajima System-1 uses Torre’s partitioning methods, such as chunking and interleaving, to split the encrypted data into multiple data blocks. EX1002, ¶209.



EX1003, Figures 15, 17.

Torre also teaches that the allocator may remap the received data using any suitable method “dependent upon particular situations.” EX1003, [0101]. Such flexible remapping methods were well known. For example, Tajima teaches in connection with Figure 11 (below): (1) receiving information to be secured, such as the encrypted data and the “encryption key used in the encryption”; (2) splitting the received data into “a plurality of data portions each of ... *arbitrary length* as shown in ... FIG. 10”; and (3) arranging the plurality of data portions into multiple data fragments X, Y. EX1004, [0109]; [0156]; Claim 8. See Section IX.A.2 [Tajima]. EX1002, ¶¶210-211.

Fig. 11



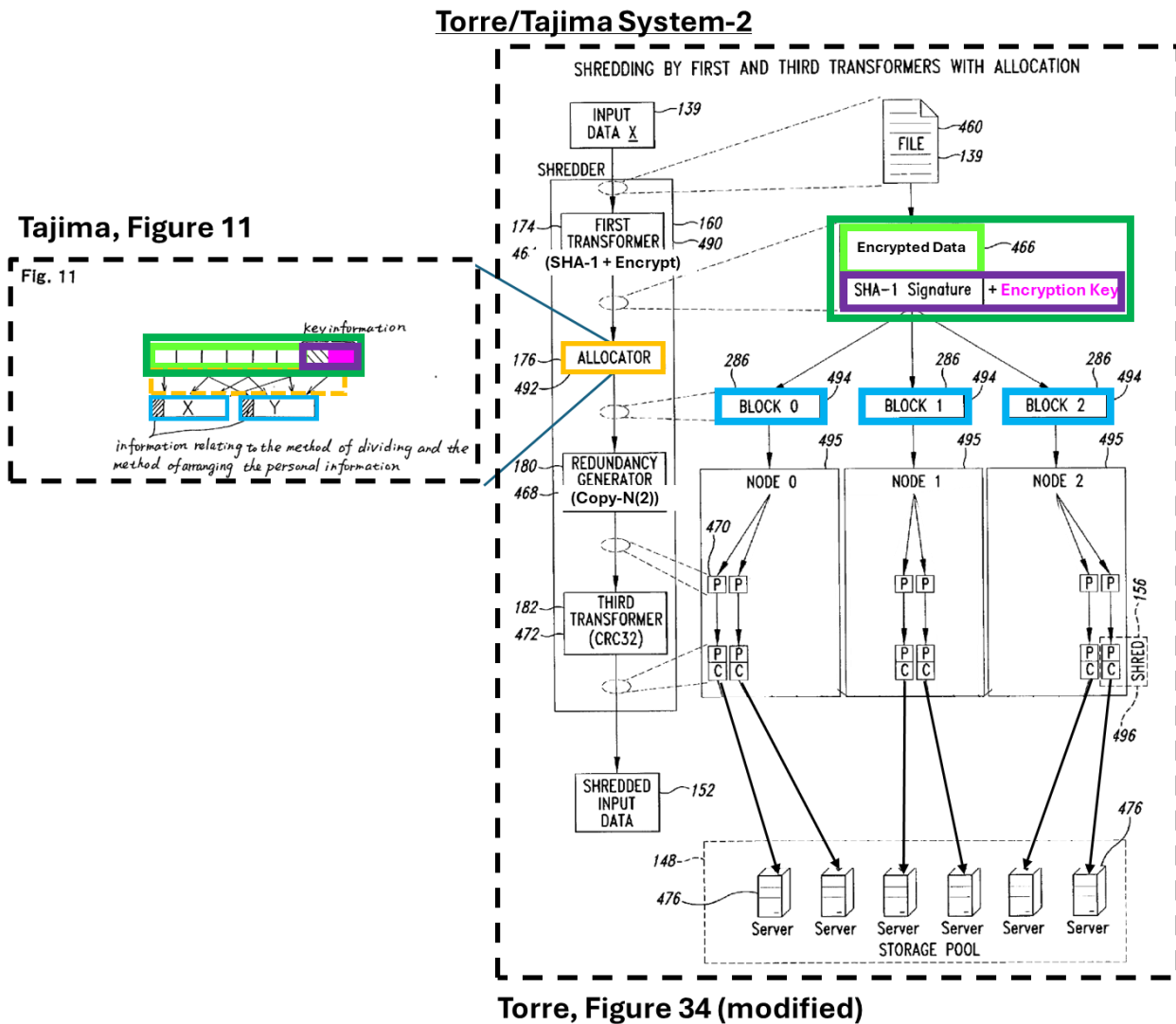
EX1004, Figure 11.

Thus, it would have been obvious to further modify Torre/Tajima System-1 to implement the **allocator 176** in Torre/Tajima System-1 to perform Tajima's **method of dividing and arranging** the **received data** to generate **data fragments** (i.e., **pre-shreds 287** in Torre/Tajima System-1), as in Tajima's Figure 11. EX1004, [0109]. EX1002, ¶212.

In this combination—referred to as “Torre/Tajima System-2”—the **allocator** determines “arbitrary length[s] according to a function of, for example, random numbers, time, or file capacity,” and uses the determined arbitrary lengths to divide the **collection** of the **transformed data** and the **encryption key** into data portions having the respective arbitrary lengths. EX1004, [0106], [0109]. The **allocator** can then assign each data portion to one of the **pre-shreds**, as in Tajima's Figure 11.

EX1004, [0106], [0109], [0156]. Torre/Tajima System-2 is illustrated below.

EX1002, ¶¶213-215.

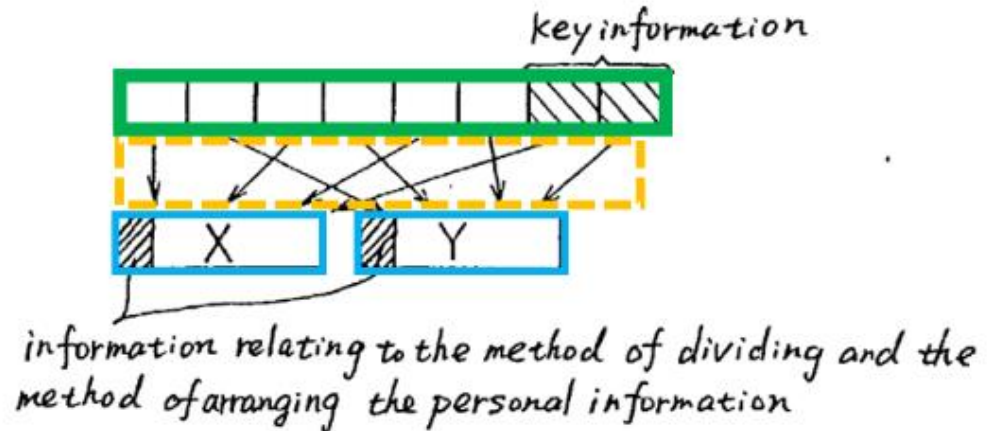


A POSITA would have been motivated to combine Torre/Tajima System-1 with Tajima's method at least because both Torre and Tajima are directed to solving a substantially similar problem in a substantially similar way. Section IX.A.3.i [Torre/Tajima System-1]. EX1002, ¶¶216-217.

Furthermore, a POSITA would have been specifically motivated to use Tajima's method in Torre/Tajima System-1 because Torre expressly motivates that combination. Torre teaches that the "scheme of remapping" is "not limited," and the transformed data and the tags can be "remapped" using any suitable method that "depend[s] upon particular situations." EX1003, [0101]. Accordingly, a POSITA would have been motivated to seek suitable methods for remapping and splitting received data, and would have understood that Tajima teaches such method that takes into consideration "particular situations." EX1004, [0106] ("different arbitrary length *according to a function of, for example, random numbers, time, or file capacity*"). EX1002, ¶¶218-219.

A POSITA would have also understood that Tajima's method improves security because by assigning data portions of "arbitrary length[s]" to different pre-shreds in round-robin (as shown in Tajima's Figure 11, below), it would be harder for a third-party to reconstruct the **data** from the **pre-shreds**. Accordingly, implementing Torre's allocator using Tajima's method would have amounted to using a known technique of rearranging and splitting received information (Tajima's Figure 11) to improve a similar device (Torre/Tajima's allocator) in the same way (increasing arbitrariness in pre-shreds/data fragments generated by the allocator). EX1002, ¶220.

Fig. 11



EX1004, Figure 11.

A POSITA would have had a reasonable expectation of success in implementing Torre/Tajima System-1's allocator using Tajima's method. **First**, there is nothing unexpected about the combination. As explained above, Torre/Tajima System-1 uses its allocator to remap and split the transformed data and the tag. EX1003, [0099], [0102]. Tajima's method, too, remaps and splits the transformed data (e.g., encrypted data) and the tag (e.g., encryption key). EX1004, [0109]; [0156]; Claim 8. Accordingly, a POSITA would have readily understood that the combination amounts to a simple substitution of one known technique (Torre's data allocation technique) for another (Tajima's method) to obtain predictable results of rearranging and splitting the received data and its tag according to Tajima's method. EX1002, ¶¶221-222.

Second, Torre explains that “the embodiments disclosed herein, in whole or in part, can be equivalently implemented ... as one or more computer programs running on one or more computers.” EX1003, [0141]. Accordingly, implementing Tajima’s method in Torre/Tajima System-1 would have merely involved writing a program code. That was well within the skill of a POSITA. EX1003, [0141]; *Keynetik*, 2023 WL 2003932, at *2. EX1002, ¶223.

Third, both Torre/Tajima System-1’s allocator and Tajima’s method process a substantially similar type of information (in Torre—the transformed (encrypted) data and its tag (including encryption key), and in Tajima—the encrypted data and its encryption key) and produce a substantially similar type of output (pre-shreds in Torre and data fragments in Tajima). Accordingly, a POSITA would have understood that Tajima’s method would readily replace Torre’s existing allocation method to perform the same function (splitting the received data) in a predictable manner (by assigning data portions of arbitrary lengths to pre-shreds as taught by Tajima). EX1002, ¶¶224-225.

At a minimum, it would have been obvious to try Tajima’s method of dividing and rearranging data. As Torre itself recognizes, the “scheme of remapping” is “not limited,” and the transformed data and the tags can be “remapped” using any suitable method that “depend[s] upon particular situations.” EX1003, [0101]. So, there was a design need to implement a remapping scheme that adapts to “particular situations.”

There would have been a finite number of predictable solutions for such remapping schemes, such as: (1) hybrid combinations of chunking and interleaving (EX1003, [0100]); (2) “divid[ing] [the received data] into a plurality of data portions each of equal arbitrary length” and “arrang[ing]” the data portions to pre-shreds (EX1004, [0103]); and (3) “divid[ing] [the received data] into a plurality of data portions each of different arbitrary length” and “arrang[ing]” the data portions to pre-shreds (EX1004, [0106]). Each of the identified solutions would have had a reasonable expectation of success for the reasons provided above. EX1002, ¶¶226-227.

*

*

*

Torre/Tajima System-1 and Torre/Tajima System-2 are identical except for the specific implementation of the allocator. Thus, in the forthcoming sections, when the analysis does not implicate the specific operation of an allocator, the analysis refers to Torre/Tajima System-1 and Torre/Tajima System-2 collectively as “Torre/Tajima System.” EX1002, ¶¶228-229.

4. Claim 1

Torre/Tajima System renders claim 1 obvious. EX1002, ¶230.

- i. [1Pre] A method for securing a data set, the method steps implemented by a programmed computer system, the method steps comprising:**

Torre discloses [1Pre]. EX1002, ¶¶231, 238.

First, Torre discloses “[a] method for securing [input data].”² Torre describes “[a] system and method for shredding and deshredding ... to **store and retrieve data** while emphasizing considerations such as redundancy, integrity, and **security of data**.” EX1003, [0049]. Torre’s “shredding system and method ... is envisioned to be used with any other encryption method that provides a **desired level of security** dependent upon such other factors as proper key management.” EX1003, [0085]. *See also* EX1003, [0095]-[0096], [0109]. EX1002, ¶¶232-233.

Second, the input data stored by Torre (and hence Torre/Tajima System) is the “*data set*.” Torre explains that “[t]he input data can be a **file or another form**” and can be generated “by receiving or recording data such as live audio or **video**, or audio or **video** broadcast streams.” EX1003, [0050]. Thus, the input data may be a video comprising a set of video frames. EX1016 (Microsoft Computer Dictionary), 350 (.mpeg, MPEG-2). The set of video frames satisfies the claimed “*data set*” because it is a set of data (frames). EX1024 (Rangan), 2. Indeed, in Patent Owner’s litigation against Google, Patent Owner broadly construed “*data set*” to mean “collection of information for storage.” EX1013, 8. EX1002, ¶¶234-236.

Third, Torre discloses that “*the method steps [are] implemented by a programmed computer system*.” Torre’s teachings may be implemented “as one or

² The quoted claim language is identified using *unbolded italics*.

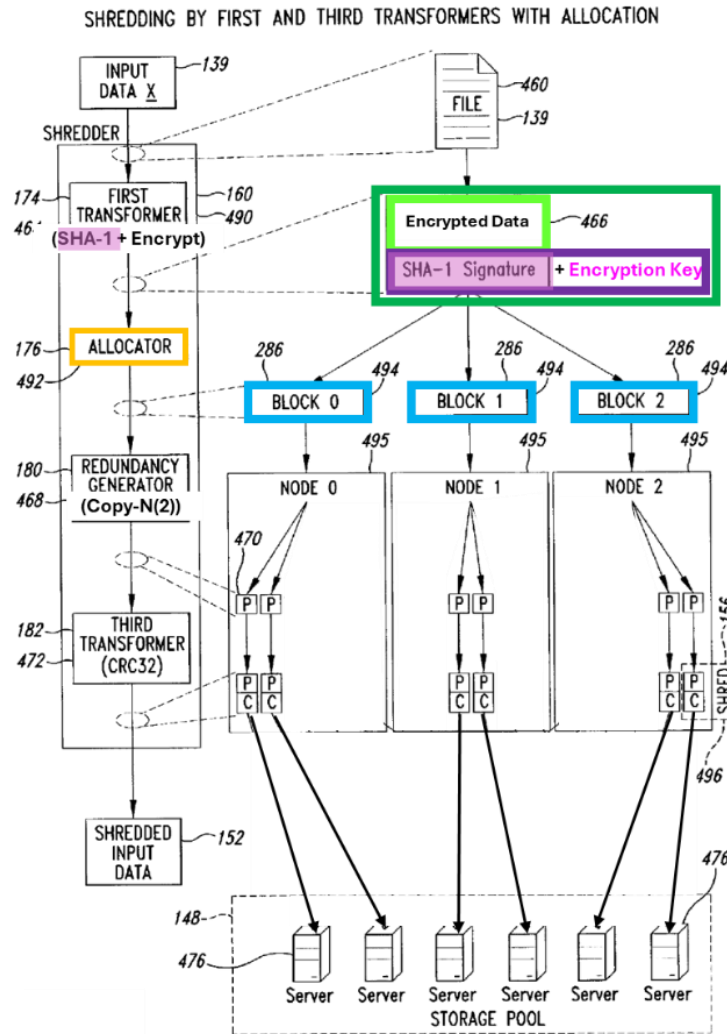
more *computer programs running on one or more computers.*” EX1003, [0141]; [0142]. EX1002, ¶237.

ii. [1A] creating integrity information using the data set;

Torre discloses [1A]. EX1002, ¶¶239, 244.

“In the process of shredding, input data [*data set*] is generally transformed into many smaller pieces of data called shreds.” EX1003, [0051]. As part of the transformations, “a first transformer 174 perform[s] the first transformation functions” on the input data. EX1003, [0073], [0062] (“data transformed by the first transformations”), Figure 34. One example of the first transformation functions is “*digital signature* generation.” EX1003, [0084]. For instance, Torre discloses that the first transformer 464 “receives the input data 139 ... and uses a *SHA-1 signature generation function* to generate a *signature associated with the received data* to generate a *pre-shred* 466 containing the received data and a tag of the *SHA-1 signature*,” as shown below. EX1003, [0131]. EX1002, ¶¶240-241.

Torre/Tajima System



The “digital signature” shows the “*integrity information*.” Torre explains that a digital signature—such as SHA-1—is used to check for “*data integrity*” of the information reconstructed from the shreds. EX1003, [0086] (“Implementations using *signature generation allows for data integrity checking*.”), [0088]. Torre’s teaching is consistent with the ’802 Patent which also states that “[e]xamples of

integrity information may include ... *digital signature* information.” EX1001, 64:30-35. EX1002, ¶¶242-243.

iii. [1B] **encrypting the data set based on an encryption key to produce an encrypted data set;**

Torre discloses [1B]. EX1002, ¶¶245, 252.

First, Torre teaches “*encrypting the data set ... to produce an encrypted data set.*” As explained in Section IX.A.1 [Torre], the first transformer can encrypt the input data (“*data set*”) after creating a digital signature (“*integrity information*”). For example, Torre’s Figure 11 (below) shows “[a] table of allowed transformation combinations” for the first transformer and the subsequent second transformer. EX1003, [0091], Figure 11. The table includes a “Fpre column [that] ... identifies which functions are performed by the first transformer 174 [in yellow] and the second transformer 178 [in green].” EX1003, [0091]. In this table, “‘E’ represents the *encryption functions*, and ‘S’ represents the signature generation functions.” EX1003, [0091]. Accordingly, row 9 of Figure 11—outlined in blue—shows that the first transformer can create a digital signature, and subsequently *encrypt* the input data to produce an *encrypted data*. *See also* EX1003, [0092]. EX1002, ¶¶246-248.

1st Transformer 2nd Transformer Advantages

TABLE OF ALLOWED TRANSFORM COMBINATIONS
FOR FIRST AND SECOND TRANSFORMERS

RULE	1st Transformer	2nd Transformer	Advantages
1	E		7 8
2	S		1
3	A		2
4	C		3
5	E A		2 7 8
6	E S		6 7 8
7	S A		1 2 12
8	S C		1 3
9	S E		1 7 8
10	A C		2 3 11
11	A E		2 4 7 9
12	A S		2 6 10
13	C A		2 3 5
14	C E		3 7 8
15	C S		3 6
16	E A S		2 6 7 8
17	E S A		2 6 7 8
18	S A C		1 2 3 11 12
19	S A E		1 2 4 7 9 12
20	S C A		1 2 3 5 12
21	S C E		1 3 7 8
22	S E A		1 2 7 8 12
23	A C E		2 3 4 7 9 11
24	A C S		2 3 6 10 11
25	A E S		2 4 6 7 9 10
26	A S C		2 3 6 11
27	A S E		2 4 6 7 9 10
28	C A E		2 3 4 5 7 9
29	C A S		2 3 5 6 10
30	C E A		2 3 5 7 8
31	C E S		3 6 7 8
32	C S A		2 3 5 6 12
33	C S E		3 6 7 8
34	S A C E		1 2 3 4 7 9 11 12
35	S C A E		1 2 3 4 5 7 9 12
36	S C E A		1 2 3 5 7 8 12
37	A C E S		2 3 4 6 7 9 10 11
38	A C S E		2 3 4 6 7 9 10 11
39	A S C E		2 3 4 6 7 9 10 11
40	C A E S		2 3 4 5 6 7 9 10
41	C A S E		2 3 4 5 6 7 9 10
42	C E A S		2 3 5 6 7 8 10
43	C E S A		2 3 5 6 7 8 12
44	C S A E		2 3 4 5 6 7 9 12
45	C S E A		2 3 5 6 7 8

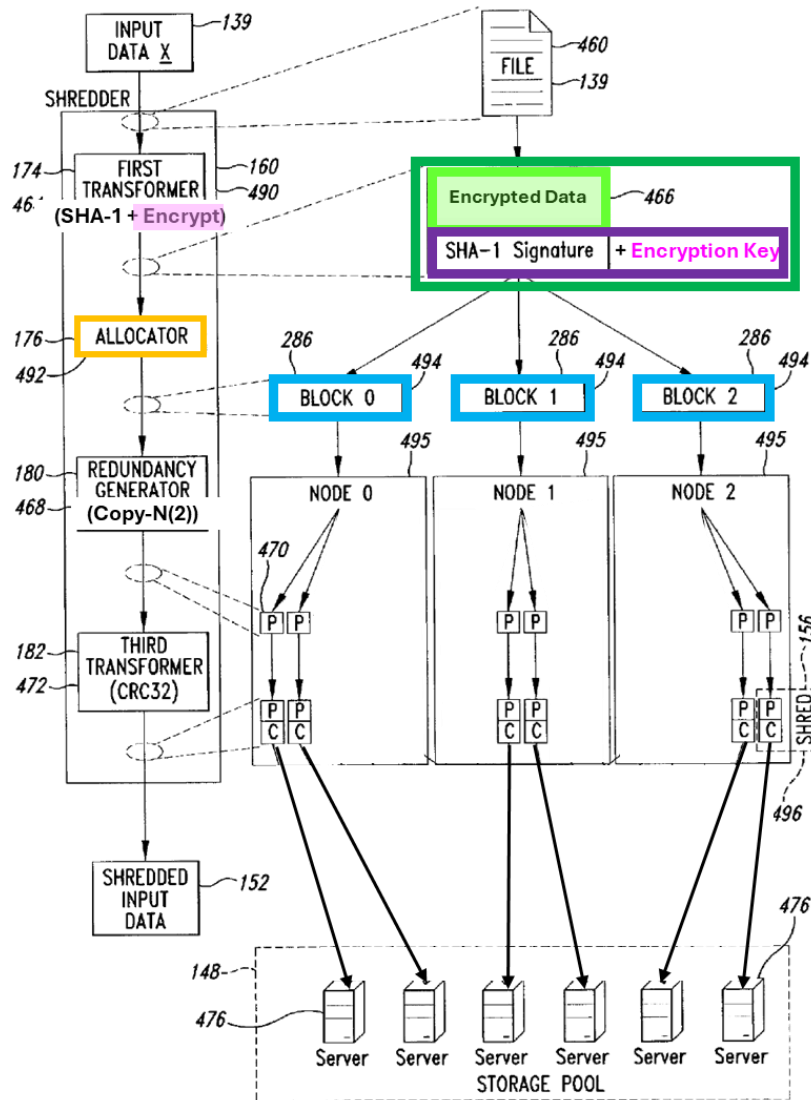
Fig. 11

EX1003, Figure 11.

The encryption operation of the first transformer and the corresponding encrypted data are illustrated below. EX1002, ¶249.

Torre/Tajima System

SHREDDING BY FIRST AND THIRD TRANSFORMERS WITH ALLOCATION



Second, Torre further teaches that the encryption step encrypts the input data “based on an encryption key.” “Implementations of encryption include ... those that use *block ciphers*,” such as “the *Data Encryption Standard (DES)*.” EX1003, [0085]. It was well known that a “block cipher” is a method of encrypting data in blocks using an encryption key, and that DES is an example of a block cipher that

encrypts the input data using a symmetric encryption key. EX1023, 12:44-58.
EX1002, ¶¶250-251.

- iv. [1C] generating **data splitting information**, wherein the **data splitting information** is usable to determine into which of a **plurality of shares of data** a unit of data of the **encrypted data set** will be placed;

- a. Torre/Tajima System-1

Torre/Tajima System-1 satisfies [1C] because Torre discloses or renders obvious [1C]. EX1002, ¶¶253, 267.

First, Torre discloses, or at a minimum renders obvious, “*generating data splitting information.*” As explained in Section IX.A.3 [Torre/Tajima], the “permutator 282 [in the allocator] re-maps the ordering of the transformed data 274 using various partitioning methods” (EX1003, [0102]), such as “*chunking, interleaving, and hybrid combinations* of these two types” (EX1003, [0100]), and the “splitter 284 [in the allocator] *splits* the re-mapped data into data blocks 286” (EX1003, [0102]). *See also* EX1003, [0099]. Torre explains that “specific remapping is *dependent upon particular situations.*” EX1003, [0101]. EX1002, ¶¶254-255.

A POSITA would have understood that Torre adapts the remapping to such “particular situations” by selecting between “various partitioning methods” (EX1003, [0102]), and such selection would entailed generating **selection**

information (“*data splitting information*”). Without such *selection information*, it would not have been possible to switch between remapping schemes “dependent upon particular situations.” EX1003, [0101]. EX1016, 87 (case statement). EX1002, ¶¶256-258.

To the extent Patent Owner argues that Torre does not teach generating such *selection information*, at a minimum, it would have been obvious to generate such *selection information* to switch between “various partitioning methods” based on “particular situations.” EX1003, [0100], [0102]. It was well known that mode switching may be performed by generating selection information. EX1016, 87 (case statement). Indeed, Torre already teaches “generat[ing] information describing [the allocator’s] processing,” which would have included information on the selected remapping scheme. EX1003, [0103]. Thus, the modification would have merely amounted to first “generat[ing] information describing [the allocator’s] processing,” i.e., *selection information*, and using it to select the remapping scheme. EX1003, [0103]. EX1002, ¶¶259-260.

Torre itself motivates the described modification because Torre teaches that the allocator would parse and split received data “dependent on particular situations.” EX1003, [0101], [0109]. EX1002, ¶261.

A POSITA would have had a reasonable expectation of success in generating and using the selection information in Torre/Tajima System-1 because it would have

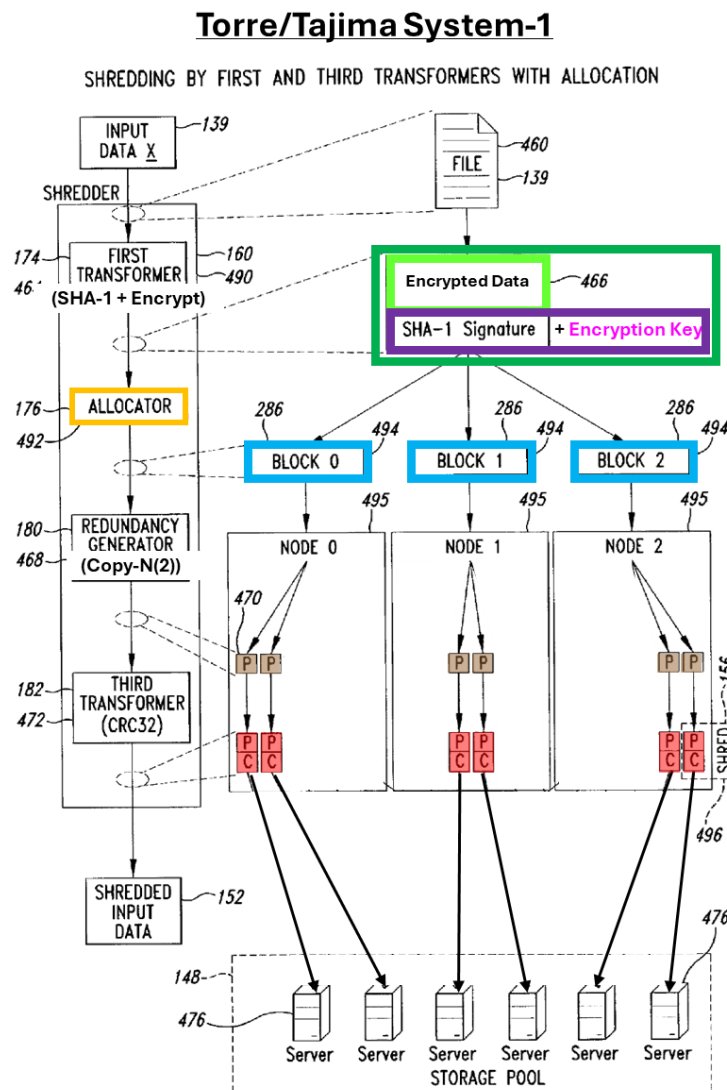
merely amounted to executing one remapping scheme over another based on the selection information. EX1002, ¶¶262-263.

Second, Torre teaches that such **selection information** “is usable to determine into which of a **plurality of shares of data** a unit of data of the **encrypted data set** will be placed.” Once the first transformer 174 performs its transformations to form a pre-shred 466, “[t]he allocator 492 **splits** the pre-shred 466 into **three data blocks** 494.” EX1003, [0135]. See also EX1003, [0061]. As part of the splitting process, the allocator uses the **selection information** to choose the partitioning method that would determine how a unit of data of the **encrypted data set** is assigned to one of the **data blocks**. EX1003, [0100]-[0102]. EX1002, ¶264.

By allocating a unit of data of the **encrypted data set** to one of the **data blocks**, the allocator has “determine[d] [using the **selection information**] into which of a **plurality of shares of data** a unit of data of the **encrypted data set** will be placed.” EX1002, ¶265.

For example, as shown below, once the allocator provides the **data blocks**, the redundancy generator 180 generates, under the Copy-N(2) redundancy scheme, **two copies of each unit of data** from the received **data block**, and provides the **two copies** to the third transformer, as illustrated in the diagram below. EX1003, [0113]. **Each copy** includes “exact copies” of the unit of data of the received **data block**. See EX1003, [0113] (“A redundancy transformation called Copy-N can be used by some

implementations to simply make *multiple exact copies of existing data.*”). The third transformer then processes *each unit of data* to form a *shred*, as shown below. EX1003, [0132]. Each *shred* thus includes a *unit of data* from the corresponding *data block*. EX1002, ¶¶265-266.



Accordingly, the allocator uses the *selection information* (“*data splitting information*”) to determine into which *data block* (and thus, which *shred*) a unit of

data of the **encrypted data set** 466 (“*encrypted data set*”) will be placed. EX1002, ¶267.

b. Torre/Tajima System-2

Torre/Tajima System-2 satisfies [1C] because Torre in view of Tajima teaches [1C]. EX1002, ¶¶268, 273.

First, Torre in view of Tajima teaches “*generating data splitting information.*” As explained in Sections IX.A.2 [Tajima] and IX.A.3.ii [Torre/Tajima], Tajima “divide[s]” “the **encrypted [data]** and information regarding the key [*encryption key*] ... into a plurality of data portions each of ... *arbitrary length* as shown in ... FIG. 10.” EX1004, [0109]; Claim 8 (clarifying that “information regarding the key” is “encryption key”). “[T]he *arbitrary length*” is “information relating to the method of *dividing* and method of *arranging*” (EX1004, [0107]), and is thus “*data splitting information.*” EX1002, ¶269.

Fig. 10

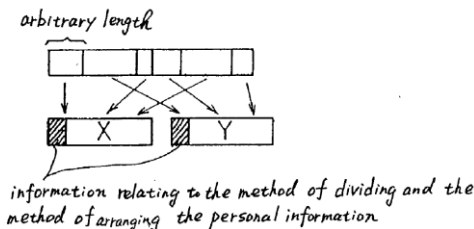
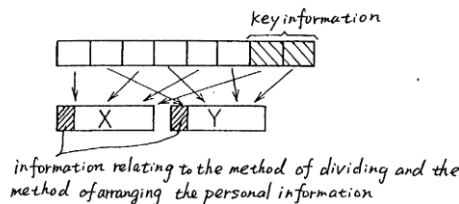


Fig. 11



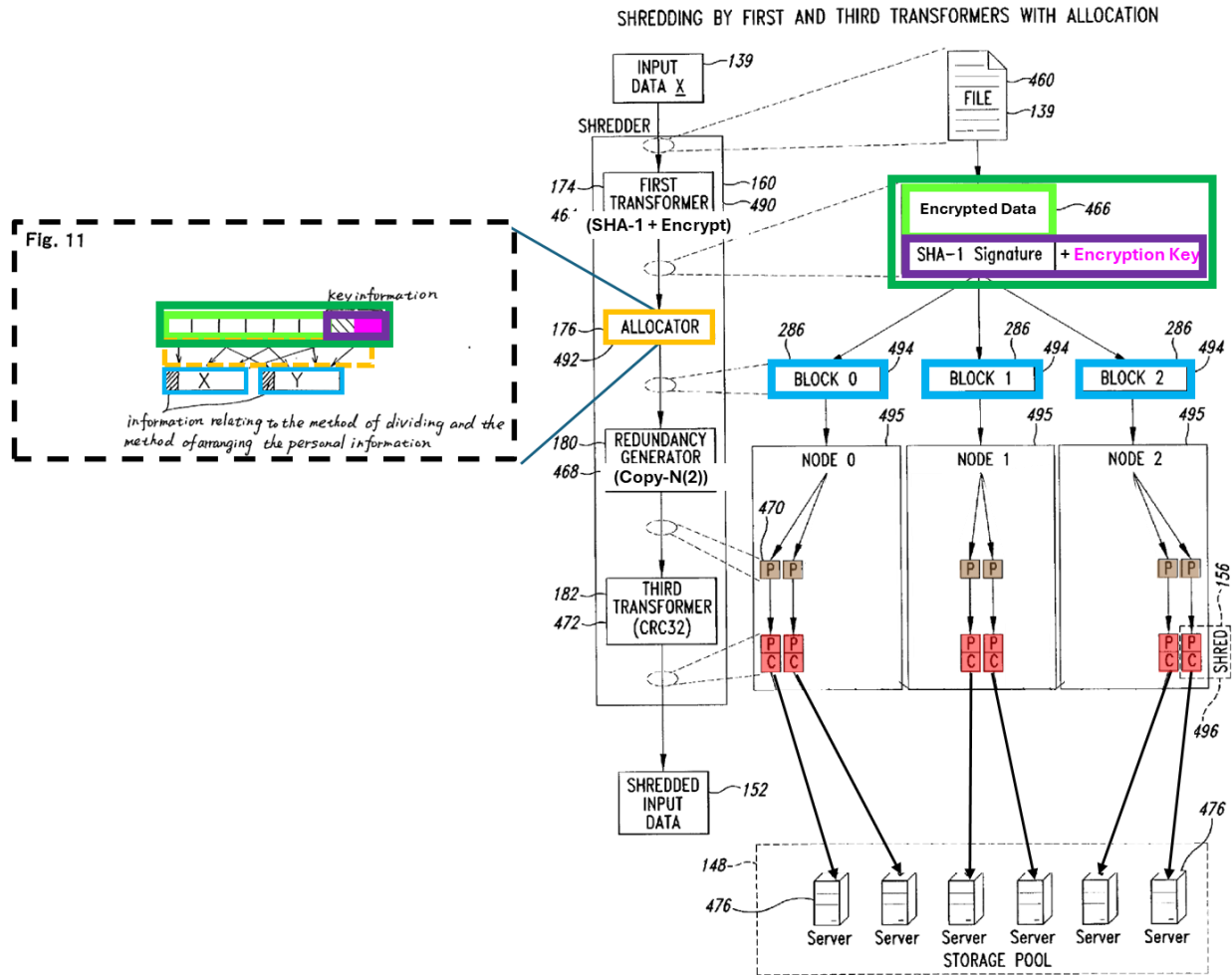
EX1004, Figures 10-11.

Tajima further teaches that the **arbitrary length information** is “*generat[ed]*.” In Tajima, data is “divided into a plurality of data portions each of different **arbitrary length according to a function of, for example, random numbers, time,** or file capacity.” EX1004, [0106]. Thus, the **arbitrary length information** depends on parameters that are determined at the time of executing the method of splitting/arranging, such as random numbers or time. Thus, the **arbitrary length information** is *generated* when the method is executed. EX1002, ¶270.

Such understanding is consistent with other teachings in Tajima. Tajima differentiates between splitting data into portions having “arbitrary length” and splitting data into portions having “*predetermined* fixed length.” EX1004, [0101], [0106]. Accordingly, the “arbitrary length” is not predetermined; it is generated at the time of execution. EX1002, ¶271.

Second, Torre/Tajima System-2 teaches that the **arbitrary length information** “*is usable to determine into which of a plurality of shares of data a unit of data of the encrypted data set will be placed*” for the reasons that the **selection information** in Torre/Tajima System-1 meets this portion of the limitation. See Section IX.A.4.iv.a. By allocating a unit of data of the **encrypted data set** to one of the **data blocks**, the allocator has “*determine[d] [using the arbitrary length information] into which of a plurality of shares of data a unit of data of the encrypted data set will be placed,*” as shown below. EX1002, ¶272.

Torre/Tajima System-2



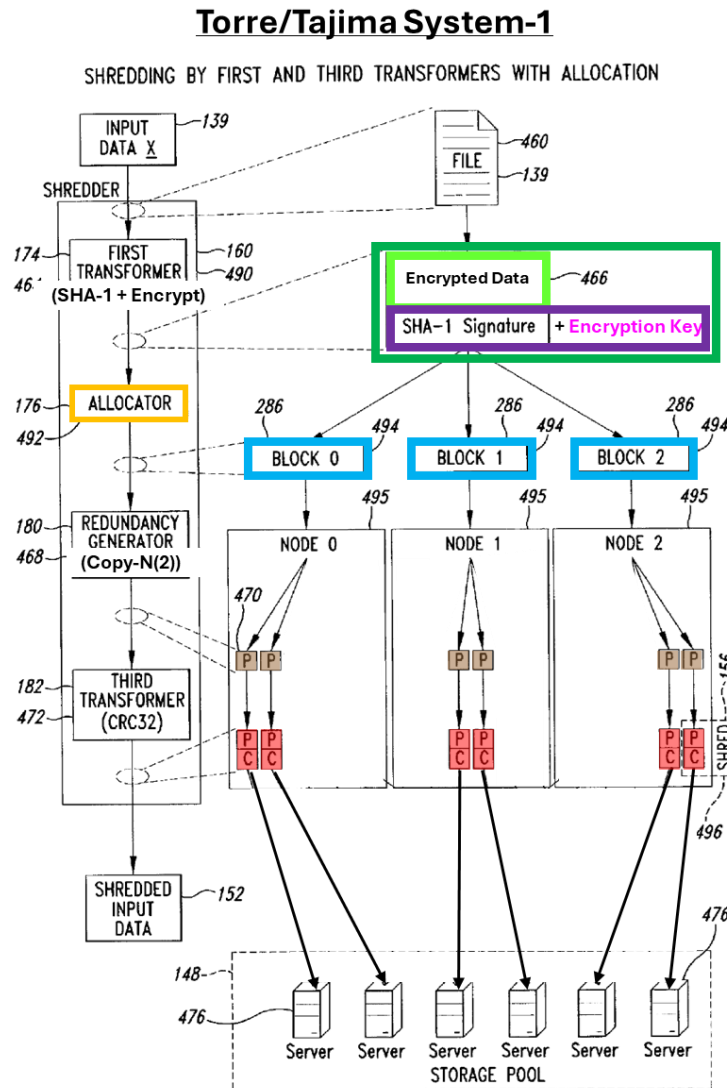
- v. [1D] separating the **encrypted data set** into the **plurality of shares** based on the **data splitting information**;

a. Torre/Tajima System-1

Torre discloses or renders obvious [1D]. EX1002, ¶¶274, 276.

As explained in Section IX.A.4.iv.a [1C, Torre/Tajima System-1], Torre discloses or renders obvious using **selection information** to separate the **encrypted data set** into **data blocks**, which are then further provided as **shreds** (“*plurality of*

shares”). See also EX1003, [0100]-[0102] (“The splitter 284 [in the allocator] then *splits* the re-mapped data into data blocks 286”). EX1002, ¶275.



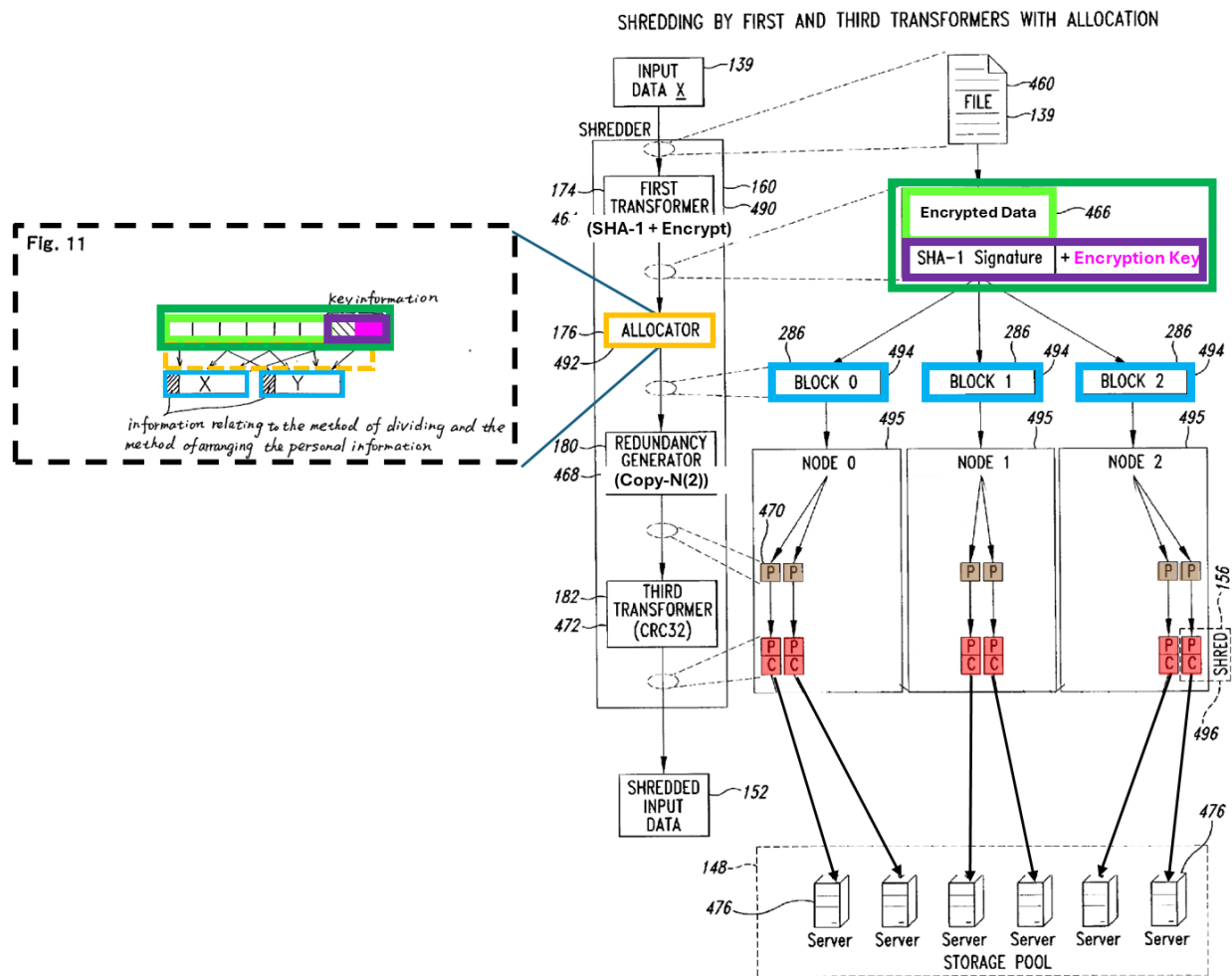
b. Torre/Tajima System-2

Torre in view of Tajima teaches [1D]. EX1002, ¶¶277, 279.

As explained in Section IX.A.4.iv.b [1C, Torre/Tajima System-2], Torre’s allocator uses the generated **arbitrary length information** (“data splitting

information”) to separate the encrypted data set into data blocks, which are then further provided as multiple shreds (“plurality of shares”), as illustrated below. EX1003, [0131]-[0136], Figure 34; EX1004, [0106]-[0107], Figure 11. EX1002, ¶278.

Torre/Tajima System-2

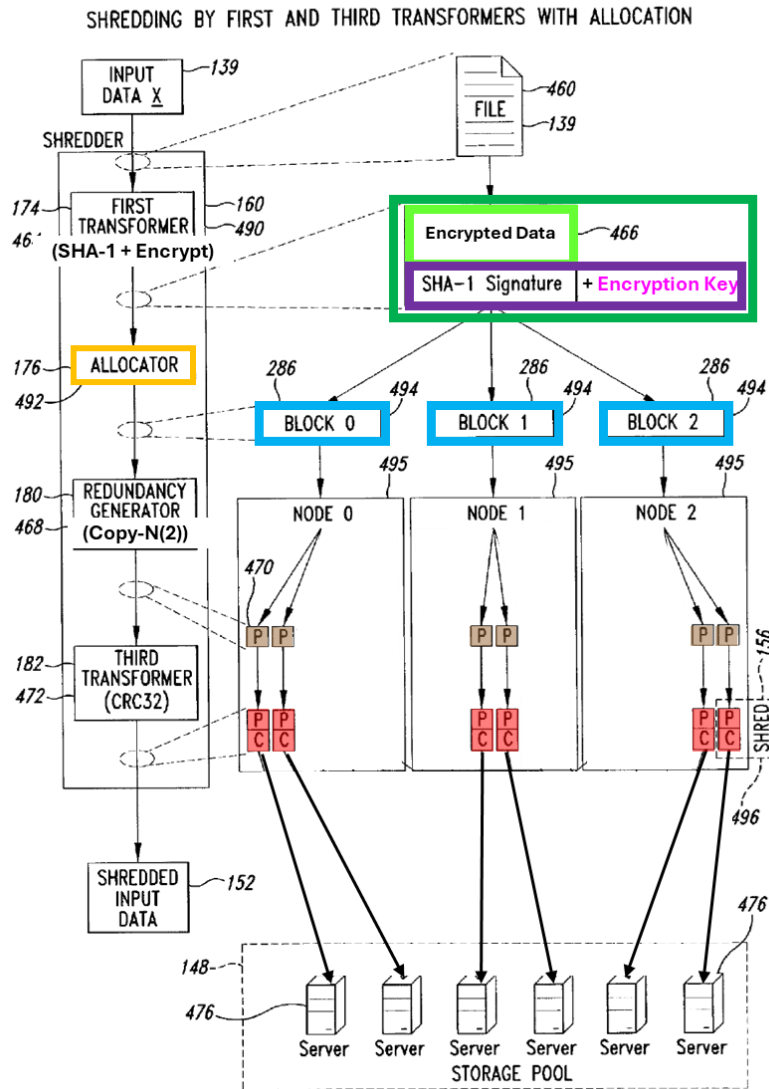


- vi. [1E] including in the plurality of shares data indicative of (a) the encryption key and (b) the integrity information; and

Torre in view of Tajima teaches [1E]. EX1002, ¶¶280, 285.

In Torre/Tajima System, the first transformer provides a **pre-shred output** that includes: (1) the **encrypted data**; and (2) a **tag** comprising the **integrity information (SHA-1 signature)** and the **encryption key**, as illustrated below. See Section IX.A.3.i [Torre/Tajima]; EX1003, Figure 11 (Rule 9), [0091], [0131], [0134], Figure 34; EX1004, [0078], [0085], [0123]. Subsequently, as explained in Section IX.A.4.iv [1C], the allocator in Torre/Tajima System divides and arranges the **pre-shred output** into **multiple pre-shreds** (EX1003, [0135]; EX1004, [0106], [0109]), and the redundancy generator and the third transformer further processes the **multiple pre-shreds** into a **plurality of shreds** (EX1003, [0131]-[0132], [0135]). EX1002, ¶¶281-282.

Torre/Tajima System



Because the first transformer's **pre-shred output**—which includes the **SHA-1 signature** and the **encryption key**—is rearranged and split into the **shreds**, the **shreds** include “data indicative of (a) the **encryption key** and (b) the **integrity information**.”
EX1002, ¶¶283-284.

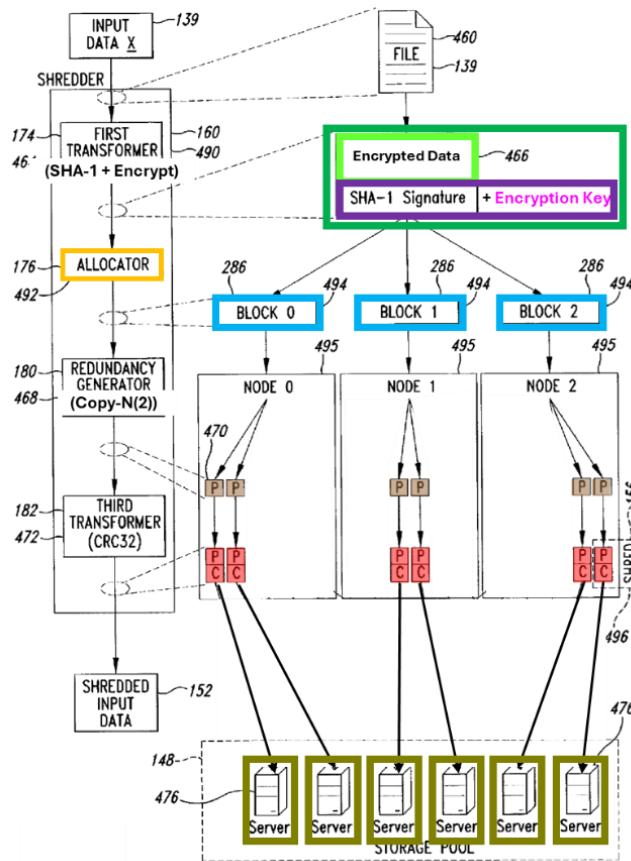
vii. [1F] causing the **plurality of shares** to be stored in **respective separate storage locations**;

Torre discloses [1F]. EX1002, ¶¶286, 288.

Torre teaches that the **shreds** are stored in **storage devices** provided in “*respective separate physical locations*.” In the context of the Torre/Tajima System, “[e]ach of the **three data blocks 494** results in **[two] shreds 156** being stored on the **[respective two] servers 476**” in a storage pool 148, as shown below. EX1003, [0135]. “For higher reliability, the **storage units** [such as the **servers 476** in the storage pool 148] could be *physically located separately from one another*.” EX1003, [0069]. *See also* EX1003, [0052]. Accordingly, the Torre/Tajima System causes the **plurality of shares** to be stored in **respective servers** that are “physically located separately from one another.” EX1003, [0069]. EX1002, ¶287.

Torre/Tajima System

SHREDDING BY FIRST AND THIRD TRANSFORMERS WITH ALLOCATION



viii. [1G] wherein the data set is restorable by accessing less than all, but at least a threshold number of, the plurality of shares.

Torre discloses [1G]. EX1002, ¶¶289, 294.

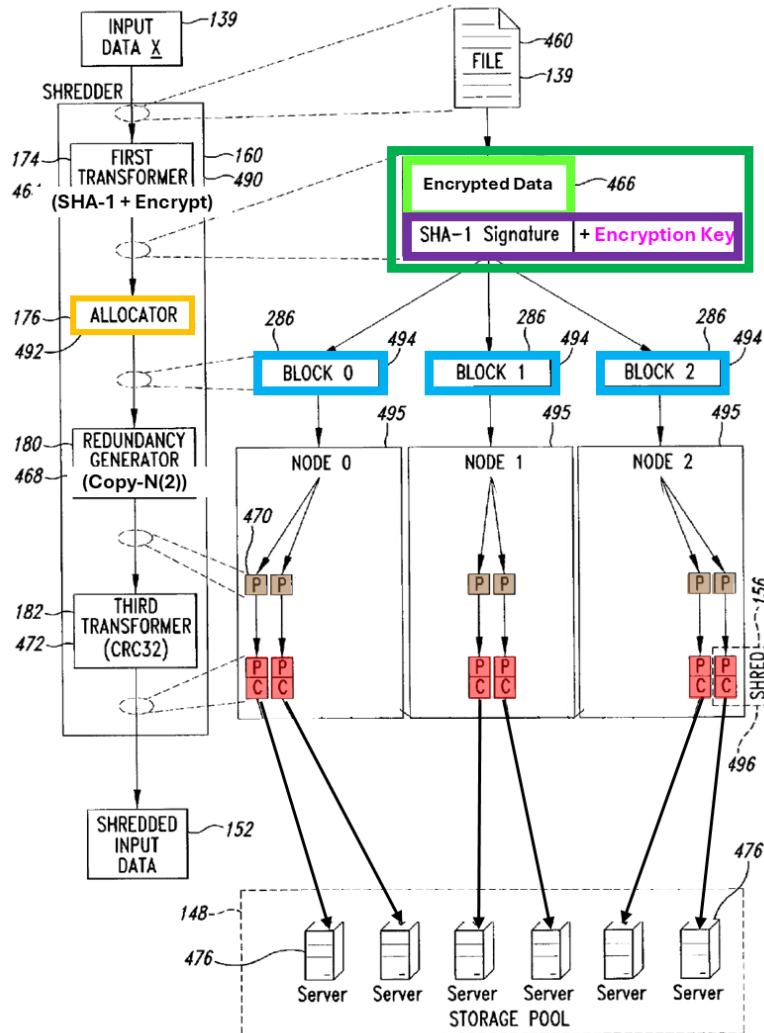
Torre explains that “[s]hredding is done in a manner that adds redundancy to the shreds so that the *input data* can be recovered from a *fewer number of shreds* than the total number stored.” EX1003, [0051]. EX1002, ¶290.

For example, Torre discloses a deshredder to reconstruct the input data from shreds. EX1003, [0121]. As an initial step of the deshredding process, “[t]he

deshredder obtains a regeneration *subset of shreds* 156 (step 411) being *enough shreds 156 from the shredded input data 152* to be able to reconstruct the input data 139.” EX1003, [0124]. If the obtained shreds are valid, “the deshredder ... performs functions inverse to those functions performed by the shredder” to reconstruct the input data. EX1003, [0121]. *See also* EX1003, [0125]-[0126]. Accordingly, Torre teaches that the input data (“*data set*”) is restorable by accessing a “*subset of shreds* 156 (step 411) being *enough shreds 156 from the shredded input data 152* to be able to reconstruct the input data 139” (EX1003, [0124]) (“*accessing less than all, but at least a threshold number of, the plurality of shares*”). *See also* EX1003, [0128]. In Torre/Tajima System shown below, the input data is restorable by accessing 5 of the 6 *shreds*. EX1002, ¶¶291-293.

Torre/Tajima System

SHREDDING BY FIRST AND THIRD TRANSFORMERS WITH ALLOCATION



5. Claim 2

- i. [2] The method of claim 1, wherein the step of separating the **encrypted data set** comprises using a deterministic technique.

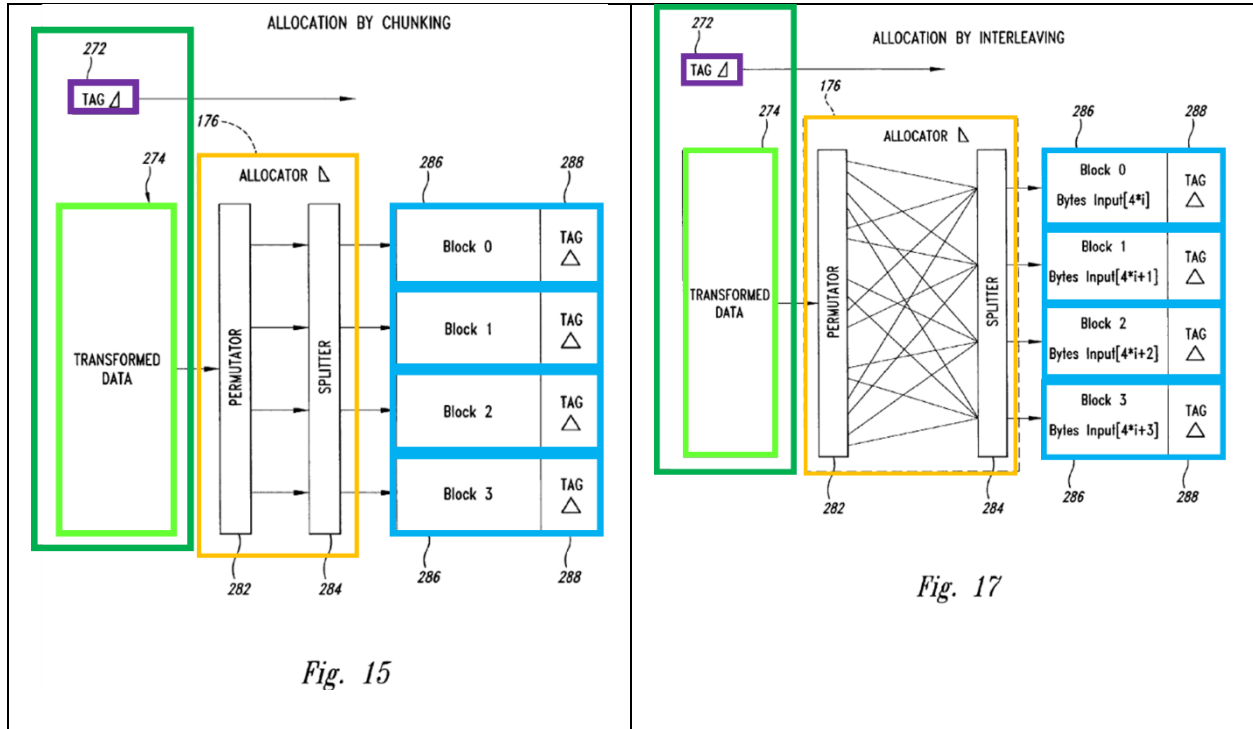
a. Torre/Tajima System-1

Torre discloses the additional limitation of [2]. EX1002, ¶¶295, 301.

The allocator in Torre/Tajima System-1 uses Torre's partitioning methods to split the **reshred** into multiple **data blocks**. EX1003, [0099]-[0102]. *See* Sections IX.A.3.i and IX.A.3.ii. EX1002, ¶295.

One such partitioning method is chunking. Torre's Figure 15 (below) shows “[a]n exemplary depiction of chunking ... in which the permutator 282 uses a pass-through re-map, which sequentially grabs blocks of data and passes the sequential blocks of data through the permutator without altering the order of the data.” EX1003, [0104]. *See also* EX1003, [0105], Figure 16. EX1002, ¶¶296-297.

Another such partitioning method is interleaving. Torre's Figure 17 (below) shows “[a]n exemplary depiction of interleaving ... in which the permutator 282 uses an interleaving re-map, which in some implementations is round-robin distributing of the data at the bit or byte level into blocks of data.” EX1003, [0107]. EX1002, ¶298.



EX1003, Figures 15, 17.

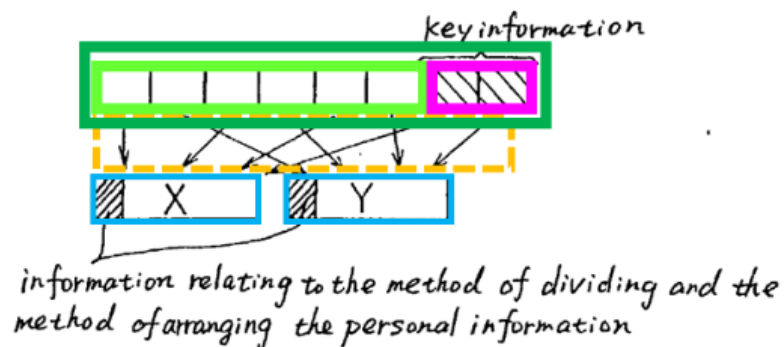
The chunking and interleaving methods are “*deterministic technique[s]*” for separating the **encrypted data** (“*encrypted data set*”) into **data blocks** because they assign a unit of data in the **encrypted data** to one of the **data blocks** in a predetermined manner. Indeed, the ’802 Patent suggests that chunking and interleaving are examples of a “deterministic algorithm.” EX1001, 72:65-73:4. EX1002, ¶¶299-300.

b. Torre/Tajima System-2

Torre in view of Tajima teaches the additional limitation of [2]. EX1002, ¶¶302, 305.

As explained in Section IX.A.3.ii, the **allocator** in Torre/Tajima System-2 determines “arbitrary length[s],” and uses the determined arbitrary lengths to divide the **collection** of the **transformed data** (“*encrypted data set*”) and the encryption key into a plurality of data portions. EX1004, [0106], [0109]. The **allocator** then assigns each data portion to one of the **data blocks** in a **round-robin manner**, as shown in Tajima’s Figure 11. Accordingly, Torre/Tajima System-2 separates the **transformed data** (“*encrypted data set*”) using a round-robin assignment of data portions (“*using a deterministic technique*”). EX1002, ¶¶303-304.

Fig. 11



EX1004, Figure 11.

6. Claims 3, 5

i. [3] The method of claim 1, wherein the step of separating the **encrypted data set** comprises using a substantially random technique.

[5] The method of claim 1, wherein the step of separating the **encrypted data**³ into the **plurality of shares** comprises causing the **plurality of shares** to have a substantially randomly distribution of the **encrypted data set**.

Torre/Tajima System-2 satisfies [3] and [5] because Torre in view of Tajima discloses the additional limitations of [3] and [5]. EX1002, ¶¶306, 310, 314.

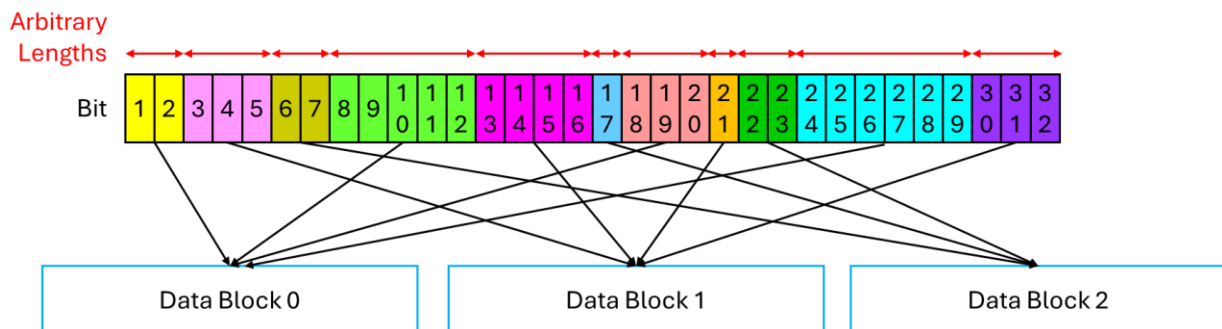
As explained in Section IX.A.3.ii, the **allocator** in Torre/Tajima System-2 determines “*arbitrary length[s]*” according to a function of, for example, *random numbers*,” uses the determined arbitrary lengths to divide the **collection** of the **transformed data** (“*encrypted data set*”) and the encryption key into a plurality of data portions, and assigns the data portions into **data blocks**. EX1004, [0106], [0109]. EX1002, ¶308.

³ The term “*encrypted data*” lacks antecedent basis. For purposes of this IPR only, Petitioner interprets “*encrypted data*” to mean “encrypted data set.” EX1002, ¶307.

Accordingly, the **allocator** separates the **transformed data** (“*encrypted data set*”) into **data blocks** using a “*substantially random technique*” that relies on “random numbers,” satisfying the additional limitation of [3]. EX1002, ¶309.

The **allocator** also satisfies the additional limitation of [5]. Because the sizes of the data portions—the arbitrary lengths—are determined based on “random numbers,” each bit of the **collection** (including the **transformed data**) is **randomly assigned** to one of the data portions, which are then assigned to the **data blocks**. EX1002, ¶311.

Such random assignments of each bit of the **collection** to data portions, and the subsequent assignment of the data portions to **data blocks** are illustrated below. Here, each block identifies one of 32 bits in the **collection**; blocks with the same color correspond to a data portion of an arbitrary length; and the data portions are added to **data blocks** in a round-robin manner per Tajima’s Figure 11. And as explained in Section IX.A.4.iv [1C], the **data blocks** are then processed into multiple **shreds** (not shown). EX1002, ¶312.



Because each bit of the **collection** (including the **transformed data**) is *randomly* associated with one of the data portions—which are then interleaved into multiple **data blocks** and subsequently into multiple **shreds** (“*plurality of shares*”)—the **shreds** have “*a substantially random distribution of the encrypted data set.*” EX1002, ¶313.

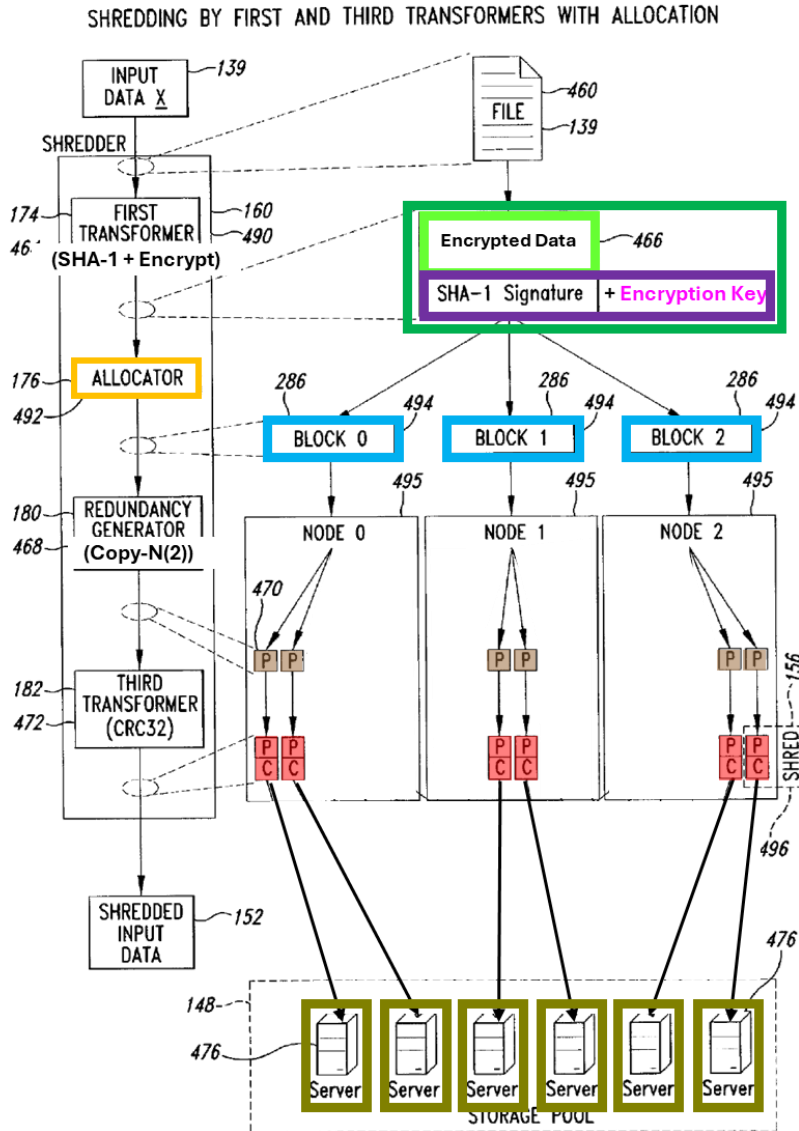
7. Claim 9

- i. [9] The method of claim 1, wherein the step of storing the **shares** in respective separate storage locations comprises storing the **shares** on at least two **separate storage devices**.

Torre discloses the additional limitation of [9]. EX1002, ¶¶315, 318.

As explained in Section IX.A.4.vii [1F], the **multiple shreds** are stored in their **respective storage devices** that may be provided in “*separate physical locations.*” And as shown below, Torre/ Tajima System stores the **shreds** (“*shares*”) on at least two **separate servers**. EX1003, [0135]. *See also* EX1003, [0052]. EX1002, ¶¶316-317.

Torre/Tajima System



8. Claim 10

Claim 10 is identical to claim 1 except for the preamble [10Pre]. Torre/Tajima System satisfies [10A]-[10G] for the reasons provided above for [1A]-[1G], respectively. EX1002, ¶¶319-321.

- i. **[10Pre] A non-transitory computer readable medium storing computer executable instructions that, when executed by at least one processor, cause a computer system to carry out a method for securing a data set, the method comprising the steps of**

Torre discloses [10Pre]. EX1002, ¶¶322, 326.

As explained in Section IX.A.4.i [1Pre], Torre discloses “*a method for securing a data set.*” EX1002, ¶323.

Torre states that its teachings “can be implemented ... by ... *software.*” EX1003, [0141]. Torre explains that its software is implemented as “code” (“*computer executable instructions*”) (EX1003, [0141]), and that the software “*when executed by at least one processor, cause[s] a computer system*” to perform the method. EX1003, [0141] (explaining that its teachings are implemented “as one or more *computer programs* running on *one or more computers* ..., as one or more *programs* running on *one or more processors*”); [0142]; [0064]. Torre also teaches that the software code is stored in “signal bearing media” such as “floppy disks, hard disk drives, CD ROMs, digital tape, and computer memory” (“[a] *non-transitory computer readable medium*”). EX1003, [0141]; [0064]. EX1002, ¶¶324-325.

9. Claim 19

Claim 19 is identical to claim 1 except for the preamble [19Pre] and limitations [19A]-[19B]. Torre/Tajima System satisfies [19C]-[19I] for the reasons provided above for [1A]-[1G]. EX1002, ¶¶327-329.

i. **[19Pre] A computer system for securing a data set, the system comprising:**

[19A] at least one processor;

Torre discloses [19Pre]-[19A]. EX1002, ¶¶330, 332, 334.

As explained in Section IX.A.4.i [1Pre], Torre discloses a method for “*securing a data set.*” EX1002, ¶330.

Torre teaches that its method is performed by a “*computer system*” having “*at least one processor.*” EX1003, [0141] (explaining its teachings are implemented as “computer programs running on *one or more computers* ... programs running on *one or more processors.*”); [0142]; [0064]. EX1002, ¶¶331, 333.

ii. **[19B] a non-transitory computer readable medium storing computer executable instructions that, when executed by the at least one processor, cause the computer system to carry out a method for securing a data set, the method comprising the steps of:**

Torre/Tajima System satisfies [19B] for the reasons provided above for [10Pre]. See Section IX.A.8.i [10Pre]. EX1002, ¶335.

10. Claims 11-12, 14, 18, 20-21, 23, 27

- Torre/Tajima System satisfies the additional limitation of claims 11 and 20 as provided in Section IX.A.5 [Claim 2];
- Torre/Tajima System-2 satisfies the additional limitation of claims 12, 14, 21, and 23 for the reasons provided in Section IX.A.6 [Claims 3, 5]; and

- Torre/Tajima System satisfies the additional limitation of claims 18 and 27 for the reasons provided in Section IX.A.7 [Claim 9].

EX1002, ¶337.

B. Ground II: Torre, Tajima, And Aldridge Render Obvious Claims 4, 13, 22.

1. Aldridge

Aldridge is directed to “storage and transfer of large datasets with high security across existing networks.” EX1005, [0013]. Aldridge explains that “[s]ecurity is achieved through two steps” (EX1005, [0013]), as illustrated in Figure 2 (EX1005, [0049]). First, “original data source 1 comprised of files and subdirectories 2-7” are “split (‘sliced’)” “into segments of fixed length 8-13.” EX1005, [0056]. Second, “[a]djacent segments may be then grouped in multiples of n (n=3 in the example in FIG. 2)” and “[t]he data in each group of n adjacent segments may then be ‘*scrambled*’ resulting in segments 14-19 of equal length.” EX1005, [0057]. EX1002, ¶¶341-342.

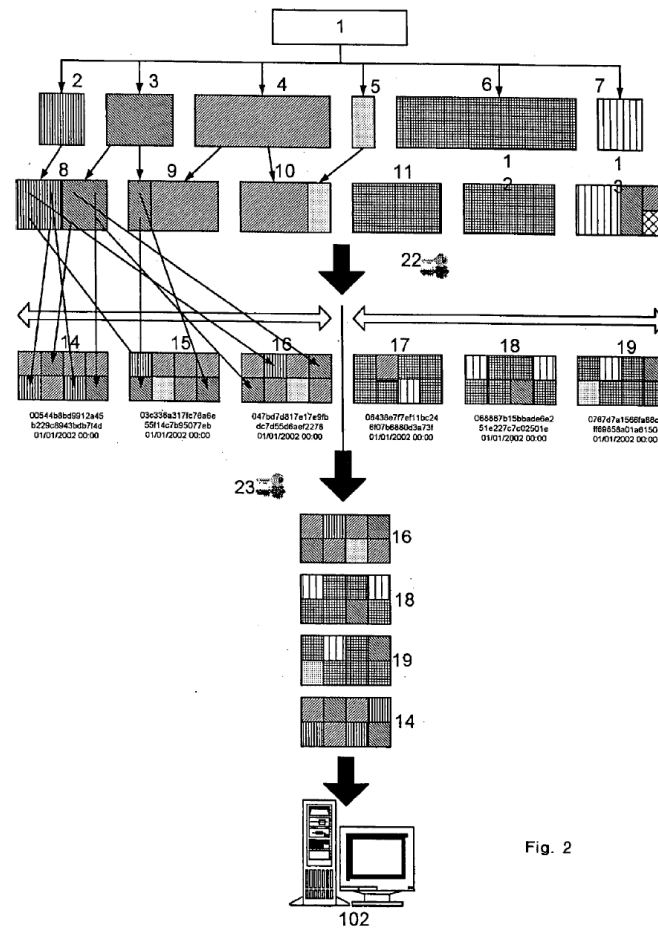


Fig. 2

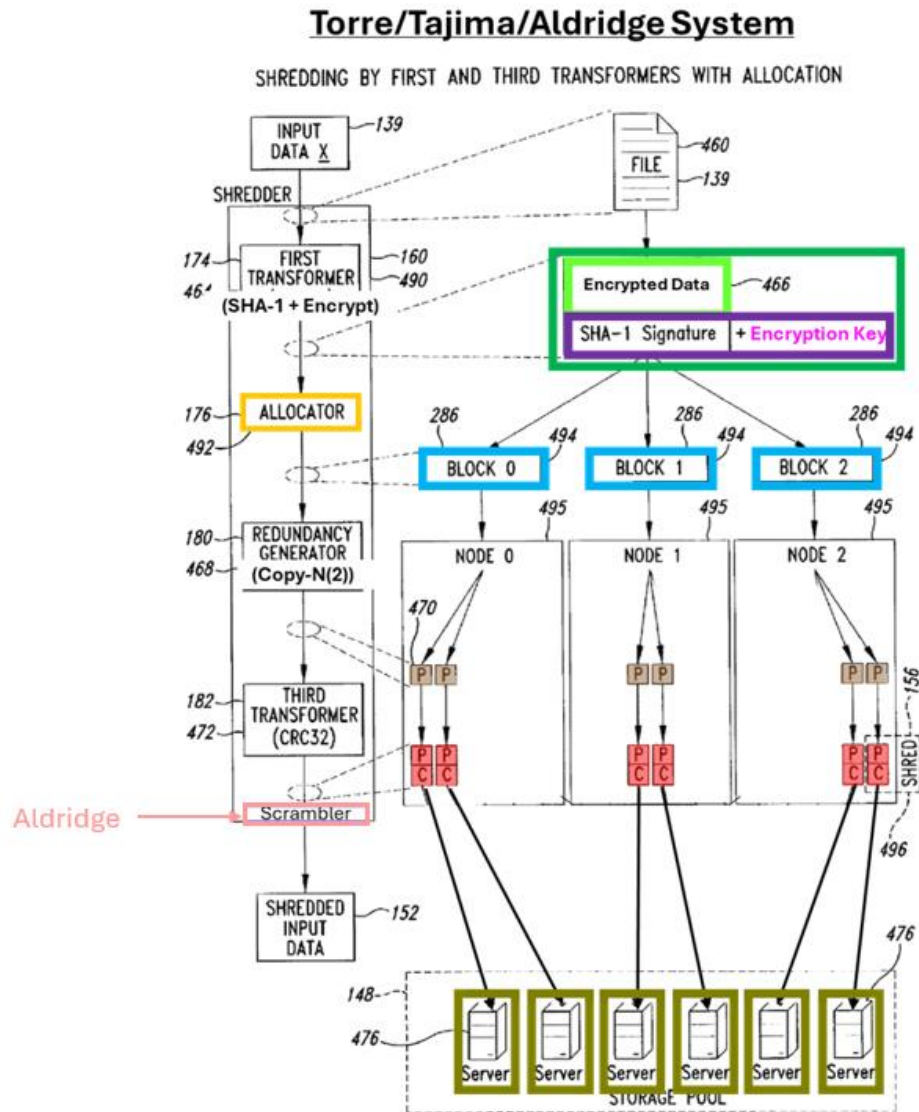
EX1005, Figure 2.

Aldridge explains that n “is less than or equal to the total number of segments” (EX1005, [0057]), and that it is beneficial to reduce “the number of segments over which the data from a file is distributed” (*i.e.*, n) so that “if the files are to be held in obscured format on the recipient's device[,]. . . it reduces the time required to reconstruct a file from the scrambled data by reducing the amount of data that must be accessed.” EX1005, [0059]. A POSITA would have understood that n is beneficially small and can be 1, or at a minimum, would have found it obvious to

make $n=1$, to minimize “the number of segments over which the data from a file is distributed.” EX1005, [0059]. EX1002, ¶343.

**2. Torre/Tajima In Combination With Aldridge
 (“Torre/Tajima/Aldridge”)**

It would have been obvious to augment the Torre/Tajima System with Aldridge’s data scrambling technique to improve security. Specifically, it would have been obvious to implement Aldridge’s data scrambling technique to scramble individual **shreds** as shown below. EX1002, ¶¶344-346.



A POSITA would have been motivated to implement Aldridge’s data scrambling technique in Torre/Tajima System. EX1002, ¶347.

Aldridge explains that the scrambled data segment “is obscured in a way that makes it very difficult to reconstruct without using the correct protocols” and that improves “security.” EX1005, [0032]. See EX1005, [0005], [0013], [0031]. And Aldridge specifically teaches scrambling data within a portion of input data—like a

shred in Torre/Tajima System—to improve security. EX1005, [0056]-[0057], [0059]; Section IX.B.1 [Aldridge]. Accordingly, a POSITA would have been motivated to implement a known technique (Aldridge’s scrambling technique) to improve security of a known device (Torre/Tajima System) in the same way (scrambling data within a **shred** (EX1005, [0059])) to obtain predictable results of making it “very difficult to reconstruct” the input data. EX1002, ¶¶348-352.

A POSITA would have had a reasonable expectation of success in implementing Aldridge’s data scrambling technique in Torre/Tajima System. Aldridge teaches that “[t]here is much prior art which may be described as ‘scrambling’” and that “[t]his will be obvious to a person skilled in the art.” EX1005, [0055]. Indeed, “[s]crambling’ may involve any process which obscures data, including but not limited to encryption, byte shuffling, byte and/or bit rotation” (EX1005, [0057]), which would have merely entailed rearranging data units within each shred using either hardware or software. EX1002, ¶¶353-355.

3. Claim 4

- i. **[4] The method of claim 1, further comprising causing a plurality of data units in each of the **shares** to be rearranged relative to one another after the separating step.**

Torre/Tajima in combination with Aldridge renders obvious the additional limitation of [4]. As explained in Section IX.B.2 [Torre/Tajima/Aldridge], the

Torre/Tajima/Aldridge System causes bytes or bits (“data units”) in each of the **shreds** (“shares”) to be rearranged relative to one another, as shown in Aldridge’s Figure 2. EX1005, [0057] (“‘Scrambling’ may involve ... *byte* shuffling, *byte* and/or *bit* rotation.”). EX1002, ¶¶356, 358.

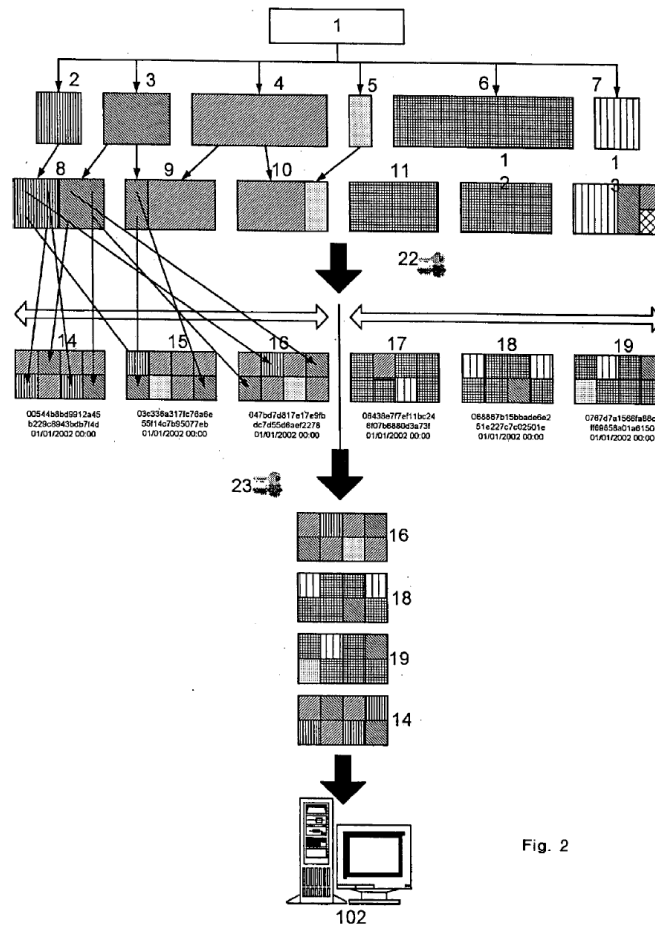
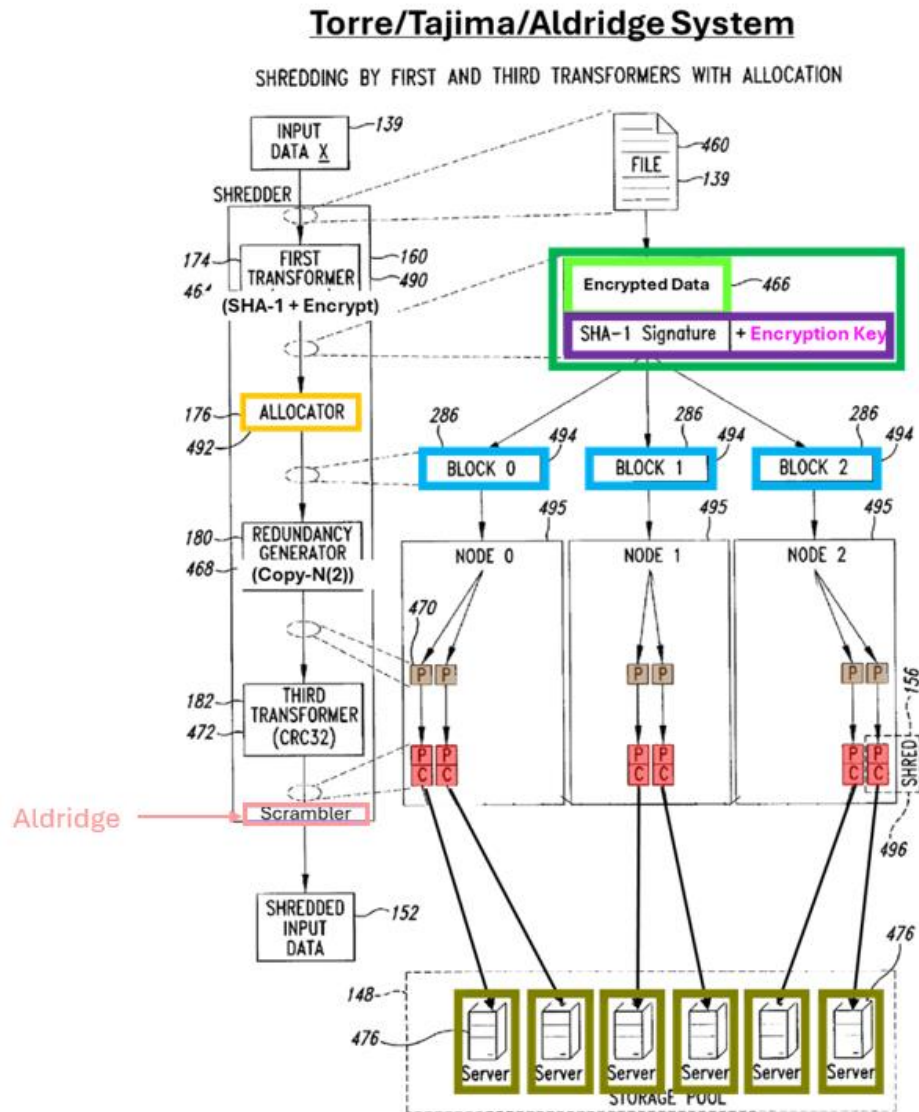


Fig. 2

EX1005, Figure 2.

In this combination, the scrambling step is performed after the separating step that forms the **shreds**, as discussed in Section IX.A.4.v [1D]. EX1002, ¶357.



4. Claims 13 and 22

The additional limitation in claims 13 and 22 is identical to the additional limitation in claim 4, except that claim 4 requires “causing a plurality of data units in *each of the shares* to be rearranged,” whereas claims 13 and 22 require “causing a plurality of data units *of the encrypted data set* to be rearranged.” As explained in Section IX.A.4.iv [1C], each **shred** (“share”) includes data units of the **encrypted**

data set 466. Accordingly, by causing data units in each of the shares to be rearranged as explained above in Section IX.B.3 [Claim 4], Torre/Tajima/Aldridge teaches causing a plurality of data units *of the encrypted data set* to be rearranged. EX1002, ¶¶359-362.

C. Ground III: Torre, Tajima, And Krawczyk Render Obvious Claims 6, 15, 24.

1. Krawczyk

Krawczyk is directed to a “secret sharing” technique that combines “an information dispersal scheme with a secure encryption scheme and a perfect (e.g. Shamir’s) secret sharing scheme.” EX1006, 8, 12. Krawczyk’s technique proceeds as below:

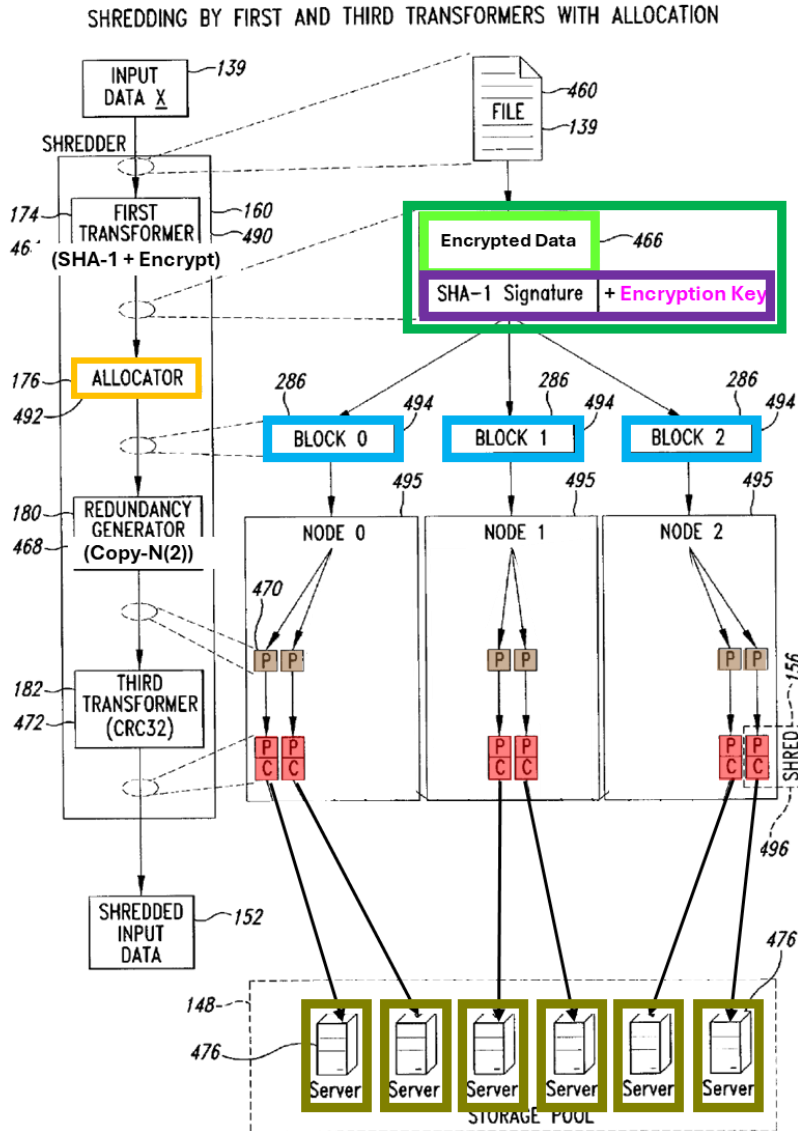
1. “Choose a random encryption key K [and] [e]ncrypt the secret S” using the encryption key K to form an encrypted secret E.
2. Use an information dispersal scheme to split the encrypted secret E into n fragments: E1, ... En.
3. Use “a perfect (n, m)-secret sharing scheme (e.g. Shamir’s)” to “generate n shares for the key K”: K1, ... Kn.
4. Send, to each of n participants, the corresponding share Si = (Ei, Ki).

EX1006, 13. EX1002, ¶¶366-367.

**2. Torre/Tajima In Combination With Krawczyk
("Torre/Tajima/Krawczyk")**

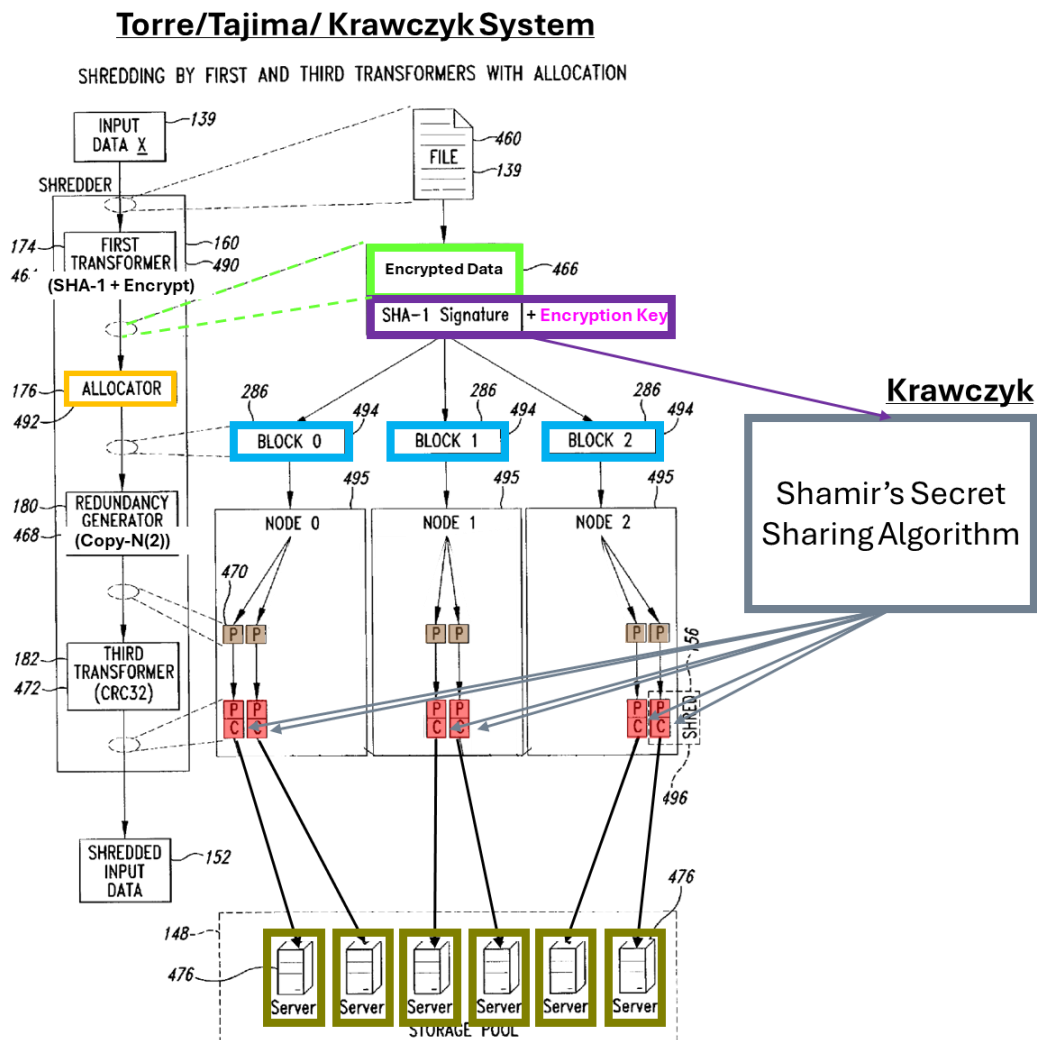
In Torre/Tajima System, when Torre's first transformer performs encryption, it provides the encryption key and SHA-1 signature as a tag, as shown below. *See* Section IX.A.3.i [Torre/Tajima]. And it further parses and splits the encrypted data and the tag to provide shreds that include data indicative of the encryption key and SHA-1 signature, as explained in Sections IX.A.4.iv [1C] and IX.A.4.vi [1E]. EX1002, ¶¶368-369.

Torre/Tajima System



Krawczyk teaches another data splitting method for providing the tag in the shreds: Krawczyk teaches using “a perfect (n, m)-secret sharing scheme (e.g. Shamir’s)” to process the tag separately from the **transformed data** and appending the outputs of the secret sharing scheme to the **shreds**. EX1006, 12. *See* Section IX.C.1 [Krawczyk]. A POSITA would have found it obvious to modify

Torre/Tajima System to separately process the tag (including the **encryption key**) using Shamir's secret sharing scheme as taught by Krawczyk, and append the processed tag (or, to be more precise, its portions) to the shares as shown below. This way, information indicative of the encryption key is provided to the **shreds**. EX1002, ¶¶370-371.



A POSITA would have been motivated to combine Torre/Tajima with Krawczyk to improve security. A POSITA would have understood that it is

important to keep the tag—particularly the **encryption key**—secure so that a third party cannot decrypt the encrypted data. Accordingly, a POSITA would have been motivated to improve the security of the tag stored in the **shreds**. And Krawczyk teaches that “a *perfect (n, m)-secret sharing scheme (e.g. Shamir’s)*” may be used to create “n shares for the key K” that can then be appended to n shares of encrypted data. EX1006, 13. *See also* EX1006, 13 (“this is essential in Shamir’s scheme to provide *perfect secrecy*”). Thus, a POSITA would have been motivated to use Shamir’s secret sharing scheme to further secure the tag. EX1002, ¶¶372-373.

Indeed, the combination would have merely amounted to using a known technique (Shamir’s algorithm) to improve a similar device (Torre/Tajima System for securely storing input data) in the same way (applying Shamir’s algorithm to the tag to create n shares and appending the n shares of tag to n shares of encrypted data as taught by Krawczyk) to yield predictable results (improving the security of the tag). EX1002, ¶374.

A POSITA would have had a reasonable expectation of success in combining Torre/Tajima with Krawczyk. **First**, Torre/Tajima and Krawczyk are both directed to the same technical problem of distributing information securely, and to substantially similar solutions that split encrypted information to shares and distributing the split encrypted information along with information on the encryption key. *See* EX1003, [0049]-[0052], [0061]-[0062]; EX1004, [0044], [0048], [0109];

EX1006, 13. Further, both Torre/Tajima and Krawczyk employ redundancy to recover the original secret information from less than all shares. EX1003, [0051]; EX1006, 12-13. Thus, the combination would have merely amounted to applying a known technique (applying Shamir's secret sharing scheme to the tag (including the encryption key) and appending the outcome of the secret sharing scheme to shares, as taught by Krawczyk) to a known device (Torre/Tajima System) ready for improvement (for improved security of the tag) to yield predictable results (improving the tag's security). EX1002, ¶¶375-378, 380.

Second, a POSITA would have understood that implementing Shamir's secret sharing scheme in Torre/Tajima System would have merely entailed writing a program code. That was well within the skill of a POSITA. EX1003, [0141]; *Keynetik*, 2023 WL 2003932, at *2. EX1002, ¶379.

3. Claim 6

- i. **[6] The method of claim 1, wherein the data indicative of the encryption key comprises data created using a Shamir algorithm.**

Torre/Tajima in view of Krawczyk teaches the additional limitation of [6]. Torre/Tajima/Krawczyk System uses “a perfect (n,m)-secret sharing scheme (e.g. ***Shamir's***)” to process the tag (including the encryption key) and appending the portions of the processed tag to the shares. EX1006, 13; Section IX.C.2.

Accordingly, the shares include “*the data indicative of the encryption key*” “*created using a Shamir algorithm.*” EX1006, 13. EX1002, ¶¶381-382.

4. Claims 15 and 24

In claims 15 and 24, the additional limitation is materially the same as the additional limitation of claim 6. Thus, Torre/Tajima/Krawczyk renders claims 15 and 24 obvious for the reasons provided for claim 6. EX1002, ¶383.

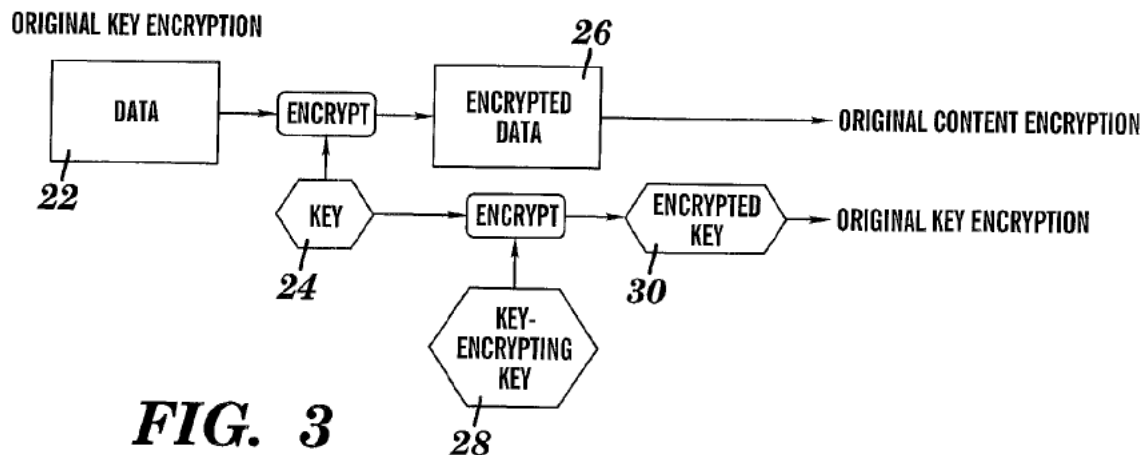
D. Ground IV: Torre, Tajima, And Foster Render Obvious Claims 7-8, 16-17, 25-26.

1. Foster

Foster discloses “[a] method and system for attaching a title key [an encryption key] to encrypted content.” EX1007, Abstract. “Specifically, a *title key* used to encrypt content is *itself encrypted*, and *attached to the encrypted content* so that both can be synchronously transmitted to a recipient.” EX1007, [0002]. For example, Figure 3 of Foster shows that:

content 22 is encrypted with title key 24 to yield encrypted content 26. Then, title key 24 is itself encrypted with key encrypting key 28. The encrypted title key 30 can then be transmitted along with the encrypted content 26 to a recipient.

EX1007, [0053]. EX1002, ¶387.

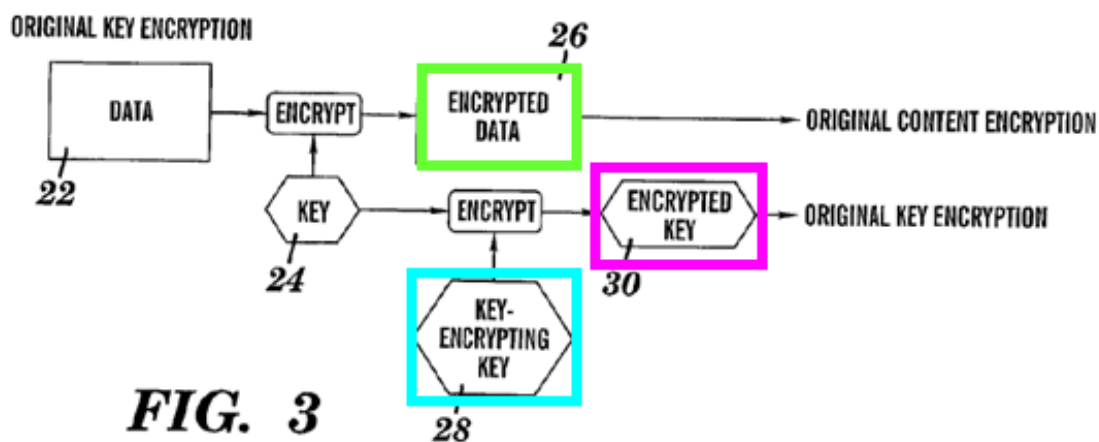


EX1007, Figure 3.

To recover the original data, the recipient decrypts the encrypted data by: (1) recovering the title key by decrypting the encrypted key 30 using the key encrypting key; and (2) decrypting the encrypted data using the title key. EX1007, [0055], [0061]. EX1002, ¶¶388-389.

2. Torre/Tajima In Combination With Foster (“Torre/Tajima/Foster”)

It would have been obvious to combine Torre/Tajima with Foster to encrypt Torre/Tajima’s encryption key using a “key encryption key” before storing the encryption key. EX1007, [0053], Figure 3. Specifically, it would have been obvious to implement the first transformer using, in part, Foster’s Figure 3 (below) to: (1) encrypt the encryption key using a **key encrypting key**, and (2) store the **encrypted encryption key** instead of the encryption key itself. EX1007, [0053]. EX1002, ¶390.

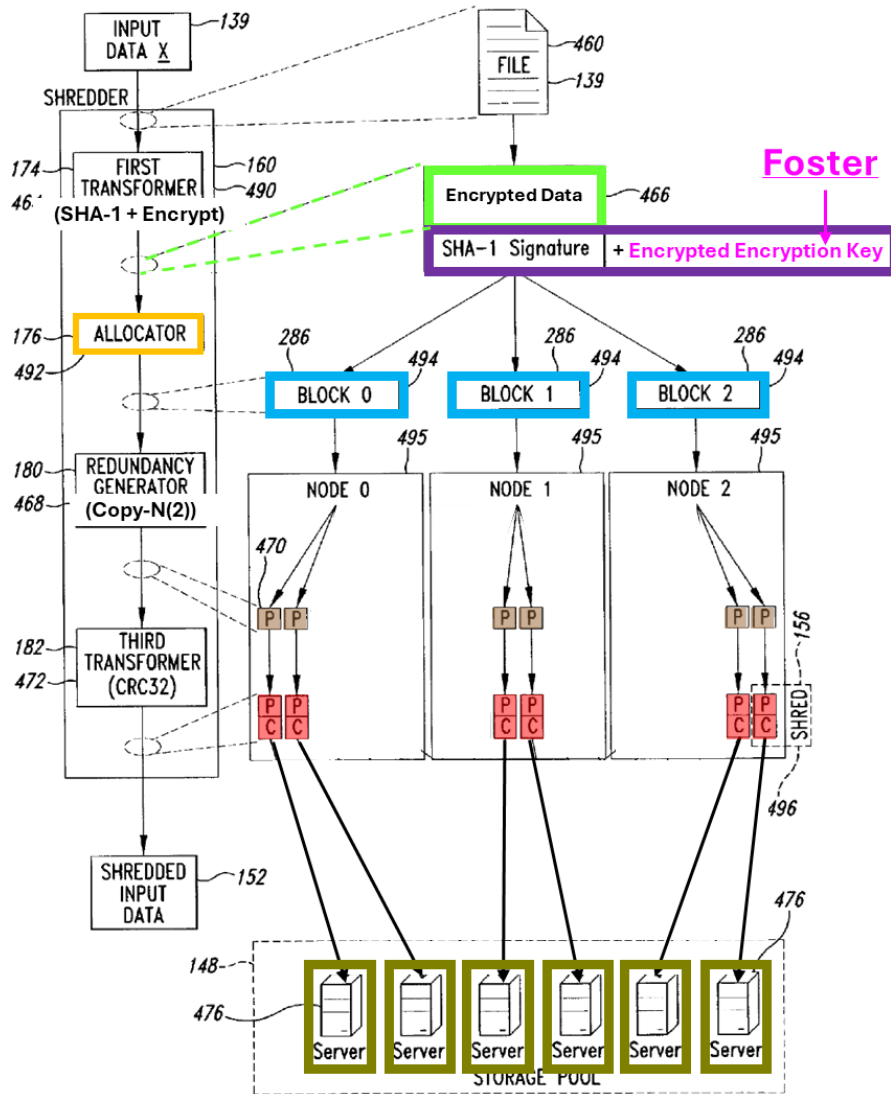


EX1007, Figure 3.

The resulting combination is illustrated below. EX1002, ¶391.

Torre/Tajima/Foster System

SHREDDING BY FIRST AND THIRD TRANSFORMERS WITH ALLOCATION



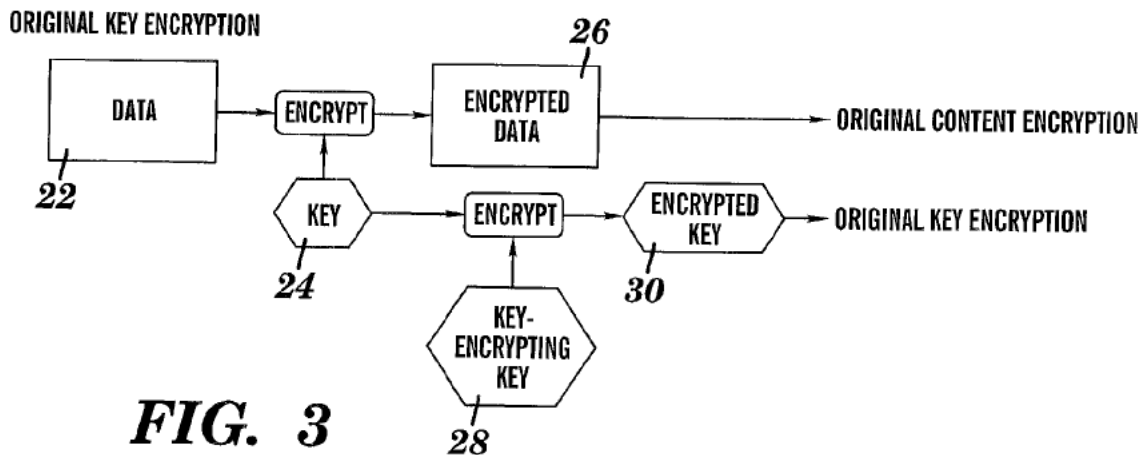
In this combination, the deshredder can obtain the input data by reversing the operation performed by the shredder described above, including decrypting the encrypted encryption key using the **key-encrypting key** 28. See EX1003, [0078], [0123]; EX1007, [0055], [0061]. The deshredder—which operates like a recipient

in Foster—can cache the **key-encrypting key** 28 in memory, as taught by Foster. *See* EX1007, [0058]. EX1002, ¶¶392-393.

A POSITA would have been motivated to combine Torre/Tajima with Foster to improve the security of the encryption key. A POSITA would have understood that it is important to keep the encryption key secure so that a third party cannot decrypt the encrypted data. Accordingly, a POSITA would have been motivated to encrypt the encryption key before storing it, as in Foster. EX1007, [0053]. The combination would have merely amounted to applying a known technique (Foster’s key encryption scheme) to a known device (Torre/Tajima System) ready for improvement (for improved security of the stored encryption key) to yield predictable results (improving the stored encryption key’s security). EX1002, ¶¶394-397.

A POSITA would have had a reasonable expectation of success in combining Torre/Tajima System with Foster. **First**, Torre/Tajima and Foster are both directed to the same technical problem of distributing secret information securely, and to substantially similar solutions that split encrypted information to shares/packets and distributing the split encrypted information along with information on the encryption key. *See* EX1003, [0049]-[0052], [0061]-[0062]; EX1004, [0044], [0048], [0109]; EX1007, [0053]. And the modification would have merely entailed inserting an

additional encryption step to encrypt the encryption key before placing it in the tag, as in Foster's Figure 3. EX1002, ¶¶398-400, 402.



EX1007, Figure 3.

Second, a POSITA would have understood that implementing Foster's key encryption scheme in Torre/Tajima System would have merely entailed writing a program code. That was well within the skill of a POSITA. EX1003, [0141]. *Keynetik*, 2023 WL 2003932, at *2. EX1002, ¶401.

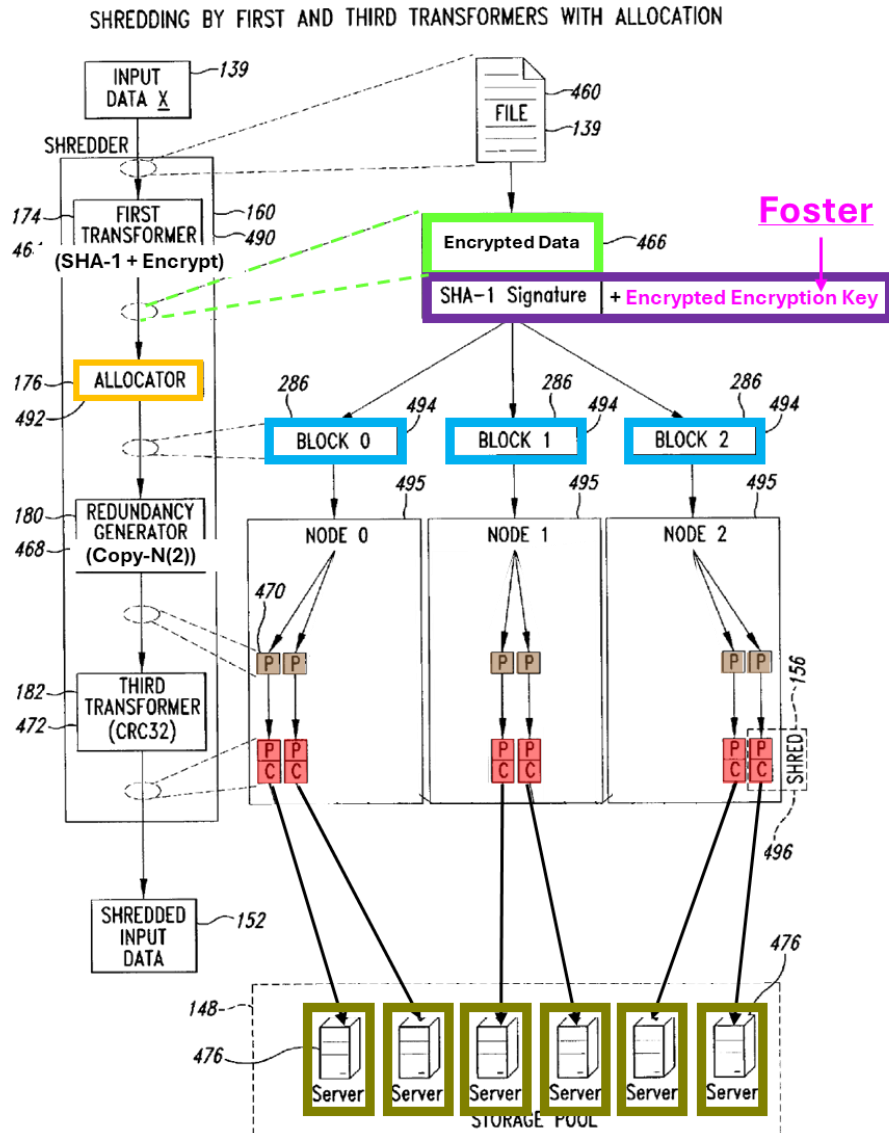
3. Claim 7

- i. [7A] The method of claim 1, wherein the step of including data indicative of the encryption key comprises: encrypting the encryption key with a second key; and
- ii. [7B] including in the plurality of shares data indicative of the encrypted encryption key.

Torre/Tajima/Foster teaches the additional limitations of [7A]-[7B]. In Torre/Tajima/Foster System, the first transformer *encrypts the encryption key* using

a **key encrypting key**; and provides the **encrypted key** as part of the tag. EX1007, [0053]. See Section IX.D.2 [Torre/Tajima/Foster]. EX1002, ¶¶403, 405.

Torre/Tajima/Foster System



Then it parses and splits the **encrypted data** and **tag**—including the **encrypted encryption key**—into **shreds**, as explained in Section IX.A.4.vi [1E]. Thus, Torre/Tajima/Foster includes in the **shreds** (“*plurality of shares*”) portions of the

encrypted encryption key (“data indicative of the encrypted encryption key”).

EX1002, ¶404.

4. Claim 8

- i. [8] The method of claim 7 further comprising the method step of storing the second key outside of the plurality of shares, wherein the data set is restorable by accessing less than all, but at least a threshold number of, the plurality of shares, and the second key.

Torre/Tajima/Foster teaches the additional limitation of [8]. EX1002, ¶¶406, 411.

First, Torre/Tajima/Foster teaches “storing the second key outside of the plurality of shares.” In Torre/Tajima/Foster, the deshredder uses the key-encrypting key to decrypt the encrypted encryption key. EX1007, [0055], [0061]. That key-encrypting key is *cached* in memory. EX1007, [0058] (“[A] receiver may *cache* ... the corresponding key encrypting keys[] in memory.”). Accordingly, Torre/Tajima/Foster stores the key-encrypting key (“second key”) outside of the shreds (“plurality of shares”). EX1002, ¶407.

Second, Torre/Tajima/Foster teaches “the data set is restorable by accessing less than all, but at least a threshold number of, the plurality of shares” as explained in Section IX.A.4.viii [1G]. EX1002, ¶408.

Third, Torre/Tajima/Foster teaches accessing and using key-encrypting key (“second key”) to restore the input data (“data set”). In Torre/Tajima/Foster, the

deshredder recovers the input data from **shreds** by reversing the operations performed by the shredder. *See* Section IX.D.2 [Torre/Tajima/Foster]; EX1003, [0078], [0123]. For instance, the deshredder first recovers the **encrypted data** and the tag (including the **encrypted encryption key**) from the shreds. Section IX.D.2 [Torre/Tajima/Foster]; EX1003, [0136]. And the deshredder recovers the input data (“*data set*”) by: (1) accessing the **key-encrypting key** 28 (“*second key*”) stored locally at the deshredder, (2) decrypting the **encrypted encryption key** using the **key-encrypting key** 28 (“*second key*”) to recover the key 24; and (3) decrypting the **encrypted data** using the key 24. EX1007, [0055], [0058], [0061]. EX1002, ¶¶409-410.

5. Claims 16-17, 25-26

In claims 16 and 25, the additional limitation is materially the same as the additional limitation of claim 7; in claims 17 and 26, the additional limitation is materially the same as the additional limitation of claim 8. Thus, Torre/Tajima/Foster renders claims 16-17, 25-26 obvious for the reasons provided above for claims 7 and 8. EX1002, ¶¶412-413.

E. Ground V: Orsini And Torre Render Obvious Claims 1-5, 9-14, 18-23, 27.

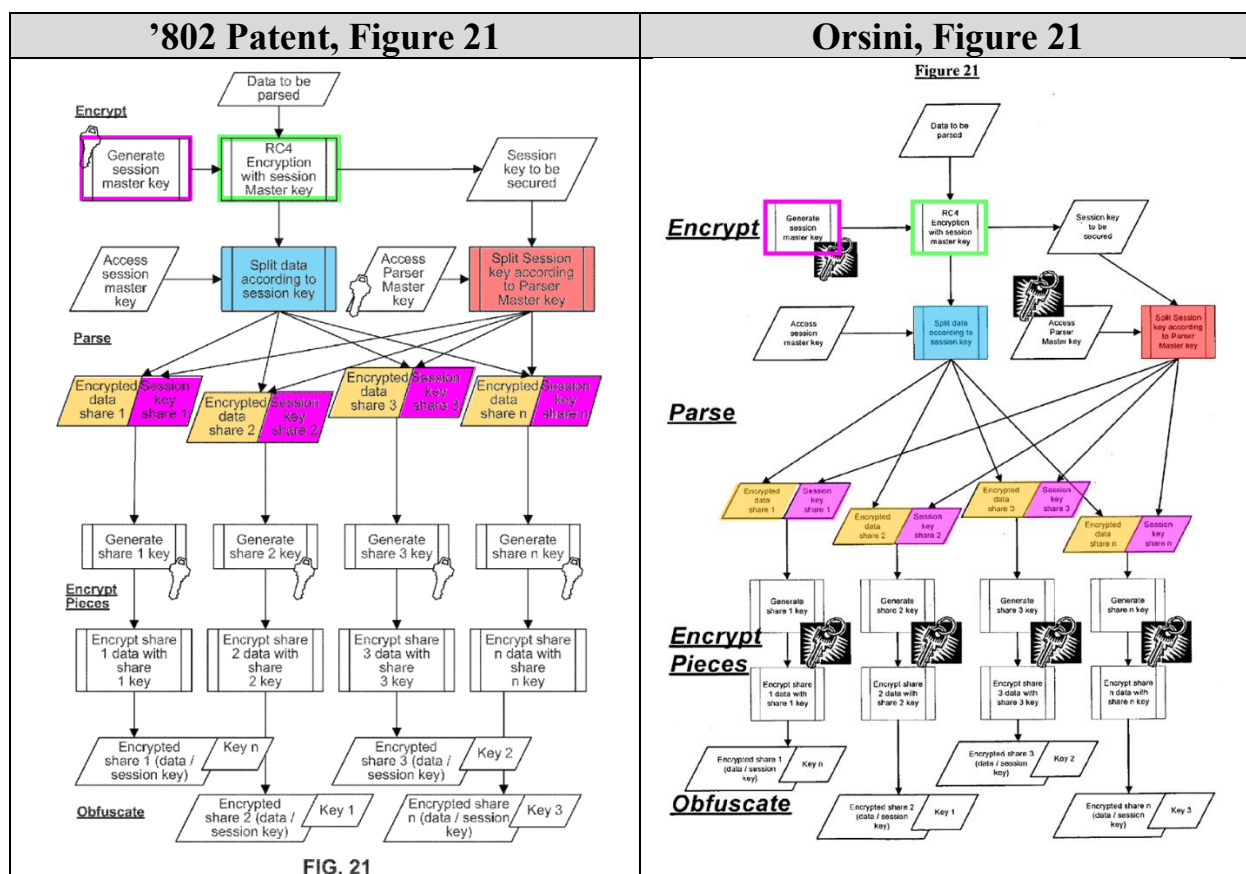
1. Orsini

Orsini is a patent publication naming, as inventors, three of the four inventors of the ’802 Patent, and an additional person not named in the ’802 Patent. Further,

Orsini was published before the earliest possible priority date of the '802 Patent. Thus, Orsini is prior art under at least 35 U.S.C. § 102(a).

Orsini includes substantially identical disclosures to the '802 Patent. In fact, Orsini's Figures 1-25 and the associated descriptions are substantially identical to the '802 Patent's Figures 1-25 and the associated descriptions. EX1012 (showing redlines between Orsini and the '802 Patent's specification). Because '802 Patent's Figures 1-25 and the associated descriptions provide written description support for most of the limitations in the '802 Patent's claims, Orsini, too, provides invalidating disclosures for most of the limitations in the '802 Patent's claims. EX1002, ¶¶417-418.

For example, just like Figure 21 of the '802 Patent, Figure 21 of Orsini illustrates a secure data parser relevant to the '802 Patent's claims, "show[ing] the steps of the process performed by the parser software suite on data to store the session master key with the parsed data." EX1008, [0294]. *See* Section VI; EX1008, [0295]-[0298], [0346]-[0357], [0363]-[0367]. EX1002, ¶¶419-421.



Orsini also teaches that the process of parsing data can be performed using “[a] cryptographic split (cryptosplit)” (EX1008, [0281]), which may “provide sufficient redundancy in the shares data such that only a subset of the shares are needed to reassemble or restore the data” (EX1008, [0283]). *See also* EX1008, [0284], [0288]-[0293]. EX1002, ¶422.

Orsini’s Figure 21 implements such redundant cryptosplit process. Orsini explains that “[t]he *parser software suite* of the present invention ... perform[s] a *cryptographic split* or parsing of the encrypted file into ... four or more shares” (EX1008, [0292]) and Orsini further states that “[a]n example of one embodiment

of the *parser software suite* of the present invention and an example of how it may be utilized is *shown in FIG. 21*” (EX1008, [0293]). To the extent a POSITA would have understood that the ’802 Patent’s claims are supported by the combination of embodiments described therein, a POSITA would have also understood that the redundant cryptosplit process may be used with Figure 21 to provide redundancy. EX1008, [0293], [0424]. EX1002, ¶423.

At a minimum, it would have been obvious to use a redundant cryptosplit process in Figure 21 “such that only a subset of the shares are needed to reassemble or restore the data to its original or useable form.” EX1008, [0283]. It was well known that providing such redundancy is beneficial in case one of the storage devices fails. EX1003, [0111]. Thus, a POSITA would have been motivated to incorporate the redundant cryptosplit into Figure 21 to provide a fault-tolerant storage system. A POSITA would have had a reasonable expectation of success because the combination would have merely amounted to applying a known technique (redundant cryptosplit process) to a known device (storage system of Figure 21) ready for improvement (to guard against failure of certain storage devices) to yield predictable results of improving fault tolerance so “that only a subset of the shares are needed to reassemble or restore the data to its original or useable form.” EX1008, [0283]. *See* EX1008, [0424]. EX1002, ¶¶424-425.

2. Orsini In Combination With Torre (“Orsini/Torre”)

Orsini does not expressly teach a mechanism to verify that the restored data set is identical to the original data set. But such verification mechanism was well known. EX1002, ¶426.

For example, Torre discloses “[a] system and method for shredding and deshredding ... to *store and retrieve data* while emphasizing considerations such as [] *integrity*.” EX1003, [0049]. See Section IX.A.1 [Torre]. Figure 34 (below) illustrates such shredding system that first processes input data using “a SHA-1 signature generation function to generate a signature” and appends the input data with the resulting SHA-1 signature to form a pre-shred 466. EX1003, [0131]. The shredder system then processes the pre-shred 466 to parse and split the pre-shred 466 into redundant shreds. EX1003, [0131]-[0132]. EX1002, ¶427.

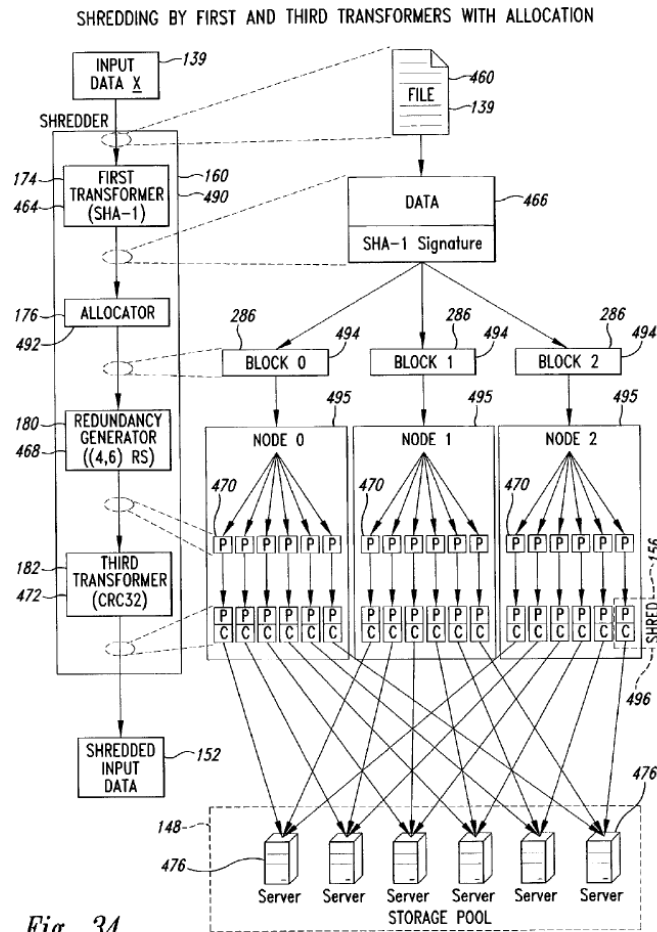


Fig. 34

EX1003, Figure 34.

As shown in Figure 35, during reconstruction, the deshredder “performs an SHA-1 signature verification function to verify that the data of the pre-shred 466 is uncorrupted.” EX1003, [0134]. EX1002, ¶428.

DESHREDDING BY FIRST AND THIRD INVERSE TRANSFORMERS WITH DEALLOCATOR

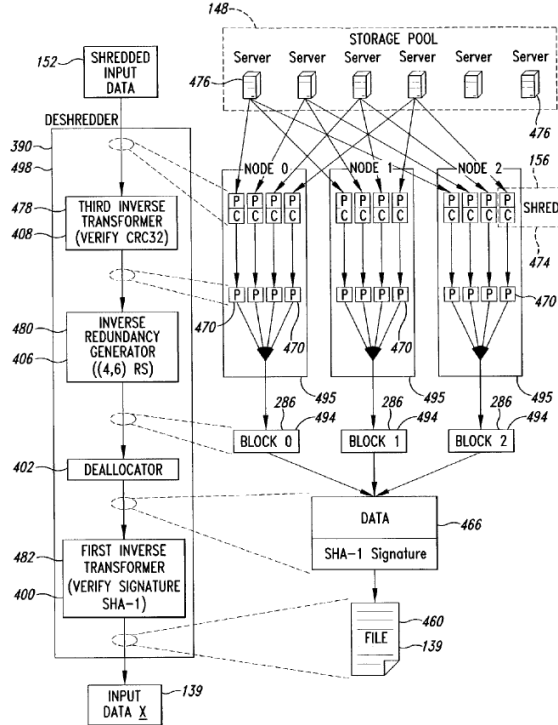


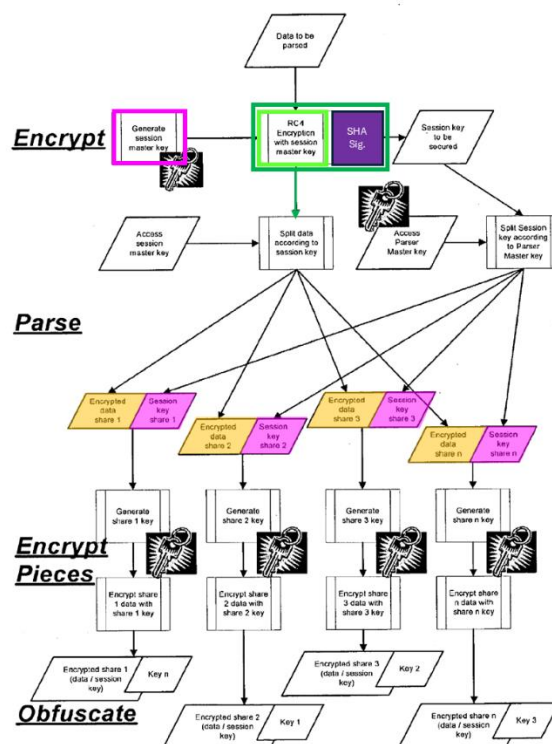
Fig. 35

EX1003, Figure 35.

It would have been obvious to augment Orsini as taught by Torre to: (1) compute a **SHA-1 signature** of input data, (2) encrypt the input data; and (3) append the SHA-1 signature to the encrypted data. EX1002, ¶¶429-430.

Orsini / Torre System

Figure 21



A POSITA would have been motivated to combine Orsini and Torre. Both Orsini and Torre are directed to a distributed storage system that stores data in multiple storage devices in portions/shares, and later reconstructs the input data from the portions/shares. See EX1003, [0049], [0051]-[0052], [0061]-[0062]; EX1008, Abstract, [0278]-[0280], [0291]. As Torre explains, when a deshredder reconstructs the original data from shares/shreds, it is beneficial to verify that the reconstructed data is identical to the original data, and one way of doing so is by using a digital signature of the original data. EX1003, [0086], [0088], [0134]. Accordingly, a POSITA would have been motivated to incorporate, into Orsini, Torre's teaching of

computing a digital signature of input data and appending it to the encrypted data, as shown above, so that Orsini's system can use the digital signature to verify the integrity of the reconstructed data. EX1003, [0086], [0088]. The combination would have merely amounted to applying a known technique (Torre's digital signature generation scheme) to a known device or method (Orsini's system and method) for ready for improvement (improving the integrity of reconstructed data) to yield predictable results (using the digital signature to verify the integrity of the reconstructed data.) EX1002, ¶¶431-435

A POSITA would have had a reasonable expectation of success. The combination would have entailed simply modifying Orsini's teachings to append a digital signature of the input data to the encrypted data, as taught by Torre, and to execute remaining steps as taught by Orsini. Appending a digital signature to the encrypted data would have merely involved writing a program code, which was well within the skill of a POSITA. EX1003, [0141]; *Keynetik*, 2023 WL 2003932, at *2. And the combination would have yielded predictable results of providing digital signature information in shares that can later be used to verify the integrity of reconstructed data—as taught by Torre. EX1002, ¶¶436-438.

Accordingly, the combination would have merely amounted to combining prior art elements (Orsini's system and Torre's digital signature generation system) according to known methods (writing code to implement Torre's digital signature

generation process in Orsini) to yield predictable results as explained. EX1002, ¶¶439-440.

3. Claim 1

i. [1Pre]

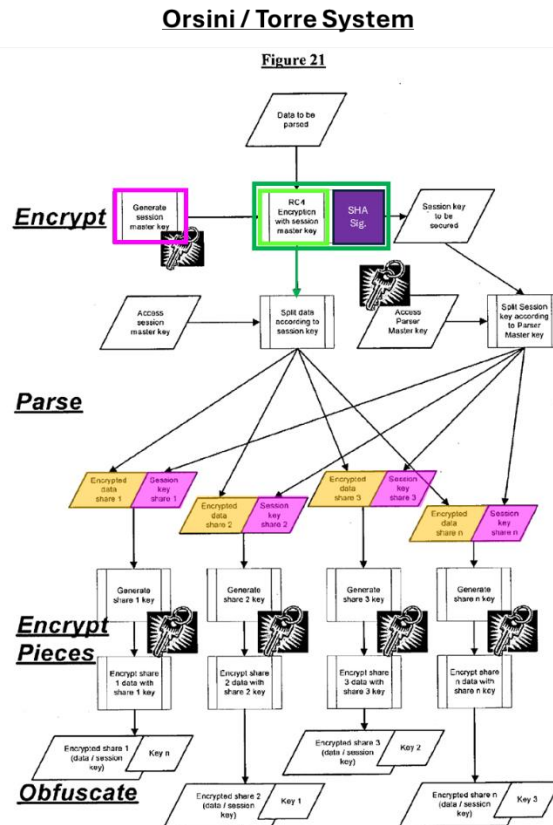
Orsini discloses [1Pre]. Orsini states “[t]he present invention provides a method and system for *securing sensitive data* from unauthorized access or use.” EX1008, Abstract, [0059]. *See also* EX1008, [0278]. And Orsini’s teachings are implemented in a “parser *software* suite” that programs a computer system. *See* EX1008, [0278], [0293]-[0294], [0409]-[0410], [0422]; Figure 21. EX1002, ¶¶441-443.

ii. [1A]

Orsini in combination with Torre teaches [1A]. Torre discloses [1A] for the reasons provided in Section IX.A.4.ii [Ground I, 1A], and it would have been obvious to incorporate Torre’s digital signature scheme into Orsini to create a SHA-1 signature (“*integrity information*”) using Orsini’s “data to be parsed,” as described in Section IX.E.2 [Orsini/Torre]. The data to be parsed—like the input data in Figure 21—is a “*data set*.” *See* EX1008, [0278] (“the present invention comprises a complete system for performing the data securing methods of the present invention on any *data set*.”); Claim 1(a); Claim 9. EX1002, ¶¶444-446

iii. [1B]

Orsini discloses [1B]. Orsini discloses “[g]enerating a **session master key** [“*encryption key*”] and **encrypt[ing]** the data [*data set*] using RS1 stream cipher” to generate an **encrypted data set**, as shown below. EX1008, [0295], Figure 21 (“Encryption with **session master key**”). See also EX1008, [0296] (“resulting **encrypted data**”); Claim 1(a) (“encrypting a data set to provide an **encrypted data set**”). EX1002, ¶¶447-449.



iv. [1C]

Orsini discloses or renders obvious [1C]. Orsini discloses “[g]enerating a **session master key**” (“*data splitting information*”) and “[s]eparating the resulting

encrypted data into four shares or portions of parsed data according to the pattern of the session master key.” EX1008, [0295]-[0296], Figure 21. “The resulting four shares of data will contain encrypted portions of the original data and portions of the session master key.” EX1008, [0298]. For example, to distribute the data “into one of the 4 shares,” Orsini teaches a cryptosplit technique that generates a sequence of random numbers (e.g., session master key) “each with a value between 1 and 4 corresponding to the four share,” and assigns each data unit to the share corresponding to the random number. EX1008, [0282]. See also EX1008, [0284]-[0290]. Since the “pattern of the session master key” is used to determine into which of the four shares a unit of data of the encrypted data into four shares will be placed, Orsini discloses [1C]. EX1008, [0295], [0298]. EX1002, ¶¶450-454, 457.

To the extent Patent Owner argues Figure 21 of Orsini does not teach the “data splitting information” because Ground V maps the session master key to both “encryption key” and “data splitting information,” Petitioner disagrees because a POSITA would have understood claim 1 to encompass embodiments of the ’802 Patent. *Sequoia Tech., LLC v. Dell, Inc.*, 66 F.4th 1317, 1327 (Fed. Cir. 2023) (“[A] claim construction excluding a preferred embodiment is rarely, if ever correct.”) (cleaned up). Indeed, the ’802 Patent states that the session master key—just like the one in Orsini—is both “used to encrypt the data” and is also a “splitting key.” EX1001, 60:57-58, 74:17-21. EX1002, ¶455.

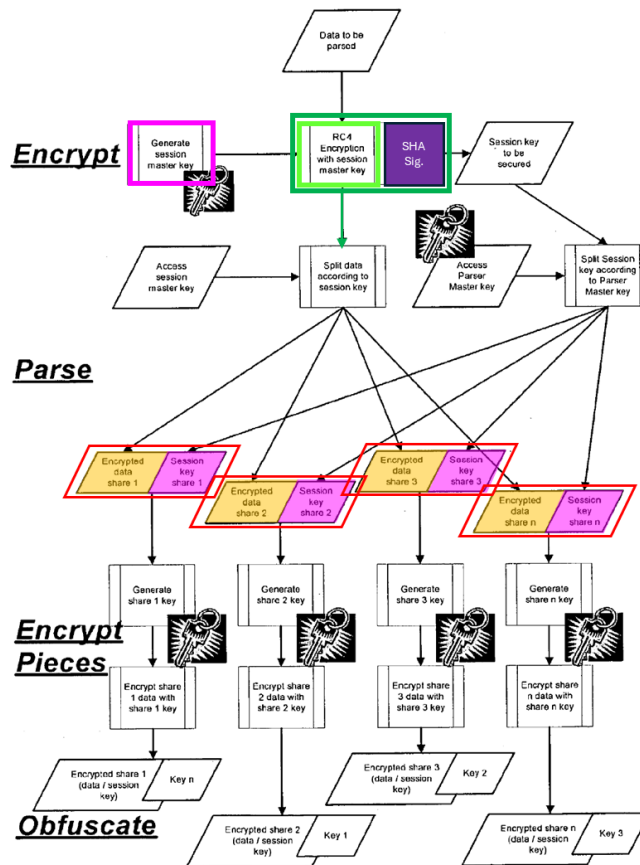
At a minimum, it would have been obvious to generate a sequence of random numbers—separate from the encryption key—for splitting the encrypted data because that would have merely entailed using two separate keys to perform the exact operations described in Figure 21 of Orsini. EX1008, [0282]. *See* EX1008, [0284]-[0290]. It would have been apparent to a POSITA that using two separate keys instead of one improves security because an intruder would need to hack both keys to recover the original data. *See also* EX1008, [0424] (“other combinations, admissions, substitutions and modifications will be apparent to the skilled artisan in view of the disclosure herein.”). EX1002, ¶456.

v. [1D]

Orsini discloses or renders obvious [1D]. Orsini discloses “[s]eparating the resulting encrypted data into four shares or portions of parsed data according to the pattern of the session master key.” EX1008, [0282], [0284]-[0290], [0296], Figure 21. To the extent “data splitting formation” must be distinct from the encryption key, using a separate key for data splitting would have been obvious. *See* Section IX.E.3.iv [1C]. EX1002, ¶¶458-462.

Orsini / Torre System

Figure 21



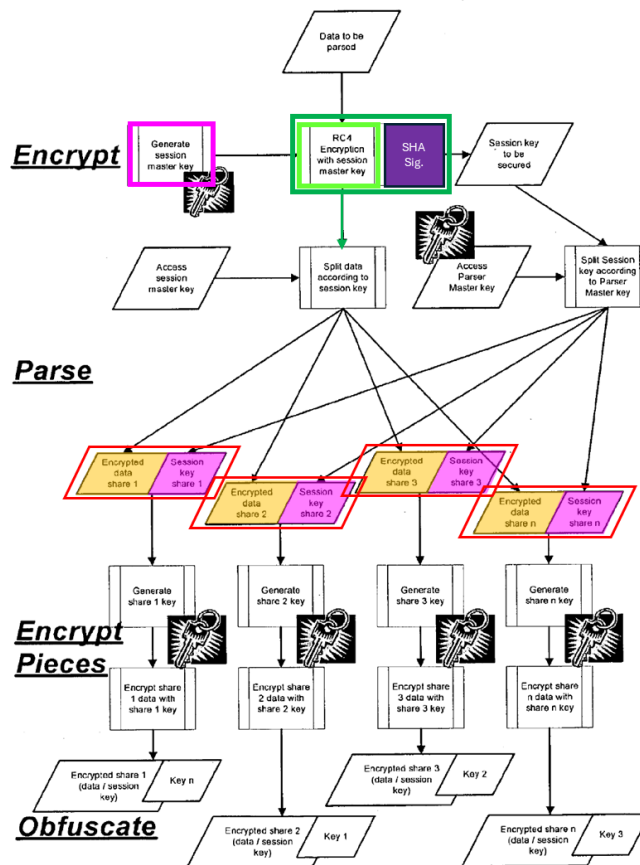
vi. [1E]

Orsini in view of Torre teaches [1E]. As shown below, Orsini/Torre teaches separating the collection of the **encrypted data**, **encryption key**, and **SHA-1 digital signature** into **shares**. EX1008, [0296]-[0298] (“The resulting **four shares of data** will contain **encrypted portions of the original data** and portions of the **session master key**.”); EX1003, [0131] (“generate a pre-shred 466 containing the received data and a tag of the **SHA-1 signature**.”); Section IX.E.3.v [1D]. Thus, the **shares** include

data indicative of (a) the encryption key and (b) the SHA-1 digital signature (“integrity information”). EX1002, ¶¶463-465.

Orsini / Torre System

Figure 21



vii. [1F]

Orsini discloses or renders obvious [1F]. Orsini states that in Figure 21, “the secured data shares [are stored] in a data depository.” EX1008, [0297]. For example, Orsini teaches “[e]ncrypting each share” and storing each encrypted share in various “locations.” EX1008, [0299]. Orsini teaches that the shares can be stored in “different physical and/or logical locations.” EX1008, [0292]. Indeed, Orsini

teaches that “the depository system 700 comprises multiple ***geographically separated independent data storage systems.***” EX1008, [0111]. *See also* EX1008, [0111] (“By ***distributing the sensitive data into distinct and independent storage facilities*** D1 through D4.”), [0114], [0175], [0379], Claim 1(d). EX1002, ¶¶466-469.

viii. [1G]

Orsini discloses or renders obvious [1G]. The process of parsing the data into multiple shares can be performed using “[a] cryptographic split (cryptosplit)” (EX1008, [0281]), which “provide[s] sufficient ***redundancy*** in the shares such that ***only a subset of the shares are needed*** to reassemble or restore the data to its original or useable form” (EX1008, [0283]). *See also* EX1008, [0284], [0288]-[0293]. As explained in Section IX.E.1 [Orsini], Figure 21 performs such redundant cryptosplit process, or at a minimum, it would have been obvious to use the redundant cryptosplit process. EX1002, ¶¶470-473.

4. Claims 2-3, 5

Orsini discloses the additional limitations of Claims 2, 3, and 5. “A cryptographic split (cryptosplit) partitions the data into N number of shares.”

EX1008, [0281].⁴ “The ***partitioning*** can be on any size unit of data ... as well as any pattern or combination of data unit sizes whether ***predetermined*** [*deterministic* as required by claim 2] or ***randomly generated*** [*substantially random* as required by claim 3].” EX1008, [0281]. “The units are then ***distributed*** (either ***randomly*** [*substantially randomly distribution* as required by claim 5] or by a predetermined set of values) into the *N shares*.” EX1008, [0281]. *See also* EX1008, [0116]. EX1002, ¶¶474-478.

5. Claim 4

Orsini discloses the additional limitation of Claim 4. Orsini states “***the data units in each share could be shuffled*** utilizing a different algorithm. This data unit shuffling may be performed as the original data is split into the data units, or after the data units are placed into the shares, or after the share is full, for example.” EX1008, [0286]. *See also* EX1008, [0282]-[0285]. EX1002, ¶¶479-482.

6. Claim 9

Orsini discloses the additional limitation of Claim 9. As explained in Section IX.E.3.vii [1F], Orsini teaches storing the shares on “***geographically separated independent data storage systems***,” such as “distinct and independent storage

⁴ The “data” in the context of Figure 21 is an “encrypted data set.” *See* EX1008, Figure 21, [0296]. EX1002, ¶475.

facilities ***D1 through D4***.” EX1008, [0111]; [0379] (“***one or more*** data depositories or storage devices”). EX1002, ¶¶483-485.

7. Claim 10

Claim 10 is identical to claim 1 except for the preamble [10Pre]. Orsini/Torre teaches [10A]-[10G] for the reasons provided above for [1A]-[1G]. EX1002, ¶¶486-488.

i. [10Pre]

Orsini discloses or renders obvious [10Pre]. As explained in Section IX.E.3.i [1Pre], Orsini discloses “*a method for securing a data set*.” EX1002, ¶¶489-490, 493.

Orsini’s Figure 21 describes a “parser ***software*** suite” that implements the method described in Figure 21. EX1008, [0292]-[0293]. Orsini explains that the “method may be incorporated into the operations of ***any computer***.” EX1008, [0013]. *See also* EX1008, [0119], [0379]. A POSITA would have understood, or at least would have found it obvious, that a computer includes ***at least one processor***, and that the parser ***software*** suite of Figure 21 would be carried in a non-transitory computer readable medium storing computer executable instructions executed by the at least one processor. EX1003, [0141]-[0142]; [0064]. EX1002, ¶¶491-492.

8. Claim 19

Claim 19 is identical to claim 1 except for the preamble [19Pre] and limitations [19A]-[19B]. Orsini/Torre teaches [19C]-[19I] for the reasons provided above for [1A]-[1G]. Orsini also discloses or renders obvious [19Pre]-[19B] for the reasons provided above for [10Pre]. EX1002, ¶¶494-497.

9. Claims 11-14, 18, 20-23, 27

- Orsini/Torre teaches the additional limitation of claims 11 and 20; 12 and 21; and 14 and 23 for the reasons provided in Section IX.E.4 [Claims 2, 3, 5] for claims 2, 3, and 5, respectively;
- Orsini/Torre teaches the additional limitation of claims 13 and 22 for the reasons provided in Section IX.E.5 [Claim 4];
- Orsini/Torre teaches the additional limitation of claims 18 and 27 for the reasons provided in Section IX.E.6 [Claim 9].

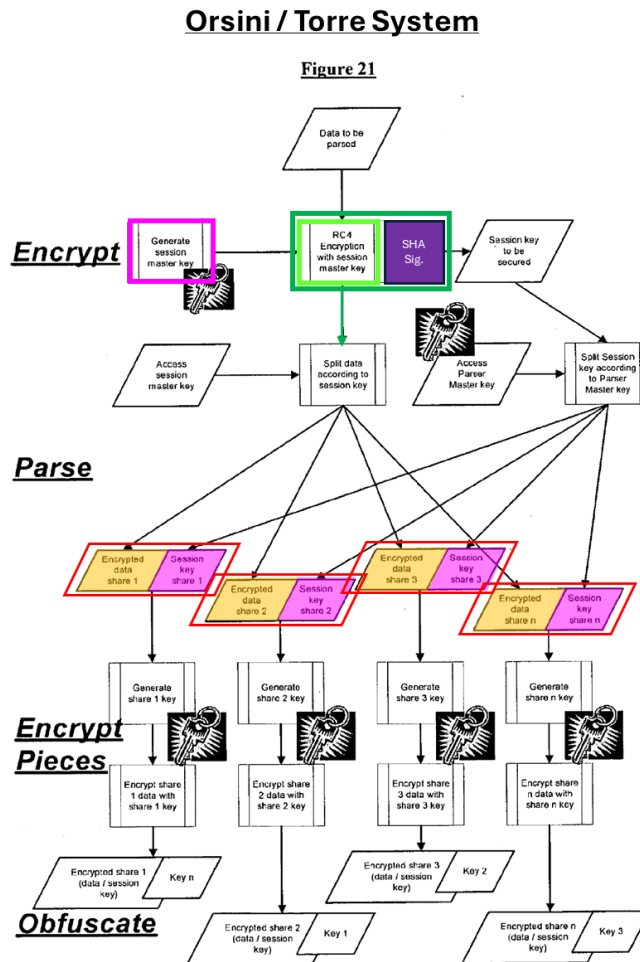
EX1002, ¶498.

F. Ground VI: Orsini, Torre, And Krawczyk Render Obvious Claims 6, 15, 24.

1. Claim 6

Orsini/Torre in combination with Krawczyk teaches the additional limitation of Claim 6. Orsini teaches including the data indicative of the encryption key in the shares by splitting the encryption key using a parser master key. EX1008, [0297]. A POSITA would have understood that the key-based splitting scheme is not highly

secure because pieces of the encryption key may be directly stored. EX1002, ¶¶502-503, 515-516.



EX1008, Figure 21.

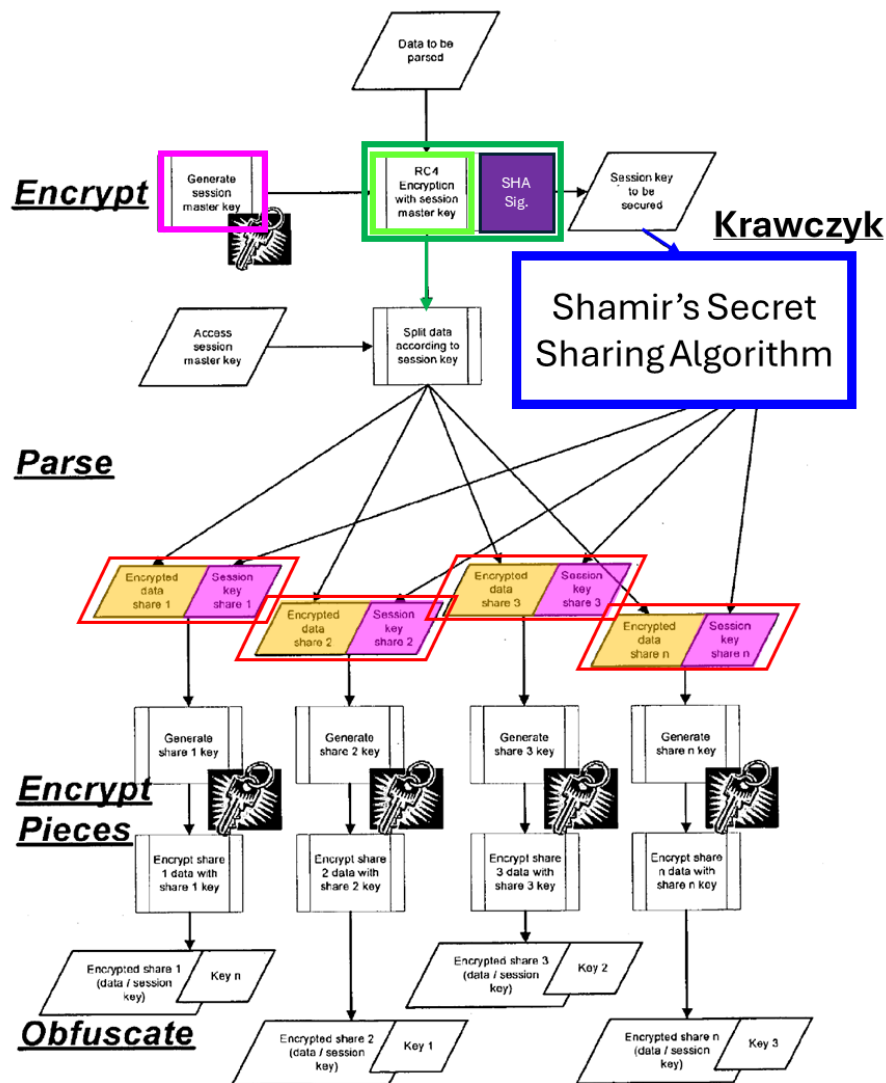
Krawczyk teaches another method for providing the encryption key in the shares, by using “a perfect (n, m)-secret sharing scheme (*e.g. Shamir’s*).” EX1006, 13. Section IX.C.1 [Krawczyk]. EX1002, ¶504.

A POSITA would have found it obvious to process Orsini’s encryption key using Shamir’s secret sharing scheme and append the processed portions of the

encryption key to the shares as shown below. The processed portions of the encryption key appended to the shares form “*the data indicative of the encryption key ... created using a Shamir algorithm.*” EX1002, ¶505.

Orsini / Torre/Krawczyk System

Figure 21



EX1008, Figure 21.

A POSITA would have been motivated to combine Orsini/Torre with Krawczyk as provided above to improve security. A POSITA would have understood that it is important to keep the encryption key secure so that a third party cannot decrypt the encrypted data. Accordingly, a POSITA would have been motivated to improve the security of the encryption key stored in the shares. And Krawczyk teaches that “a *perfect (n, m)-secret sharing scheme (e.g. Shamir’s)*” may be used to create “n shares for the key K” that can then be appended to n shares of encrypted data. EX1006, 13. The combination would have merely amounted to using a known technique (Shamir’s algorithm) to improve a similar device (Orsini/Torre System) in the same way (applying Shamir’s algorithm to the encryption key to create n shares and appending the n shares of tag to n shares of encrypted data as taught by Krawczyk) to yield predictable results (improving the security of the encryption key). EX1002, ¶¶506-508.

A POSITA would have had a reasonable expectation of success in combining Orsini/Torre System with Krawczyk. **First**, Orsini/Torre and Krawczyk are both directed to the same technical problem of distributing secret information securely, and to substantially similar solutions. EX1003, [0049]-[0052], [0061]-[0062]; EX1006, 13; EX1008, Abstract, [0278]-[0280], [0291]. Further, both Orsini/Torre and Krawczyk employ redundancy to recover the original secret information from

less than all shares. EX1003, [0051]; EX1006, 12-13; EX1008, [0283]-[0290]. EX1002, ¶¶509-512.

Second, a POSITA would have understood that implementing Shamir’s secret sharing scheme in Orsini/Torre would have merely entailed writing a program code. That was well within the skill of a POSITA. EX1003, [0141]. *Keynetik*, 2023 WL 2003932, at *2. EX1002, ¶¶513-514.

2. Claims 15 and 24

In claims 15 and 24, the additional limitation is materially the same as the additional limitation of claim 6. Thus, Orsini/Torre/Krawczyk renders claims 15 and 24 obvious for the reasons provided in Section IX.F.1 [Claim 6]. EX1002, ¶517.

G. Ground VII: Orsini, Torre, And Foster Render Obvious Claims 7-8, 16-17, 25-26.

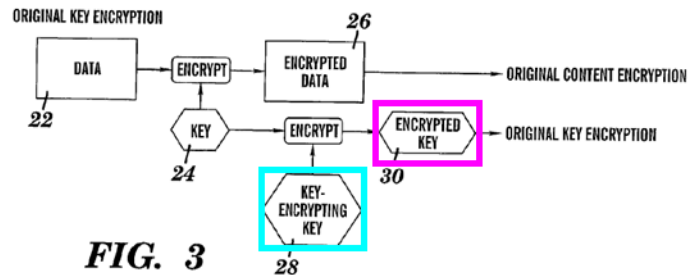
1. Orsini/Torre In Combination With Foster

As explained in Section IX.F.1 [Claim 6], a POSITA would have understood that the key-based splitting scheme used in Orsini is not highly secure. EX1002, ¶521.

It would have been obvious to combine Orsini/Torre with Foster to encrypt Orsini/Torre’s encryption key using a “key encryption key” before storing it to improve security. EX1007, [0053], Figure 3. Specifically, it would have been obvious to incorporate into Orsini/Torre Foster’s key encryption method in Figure 3 (below) to: (1) encrypt the encryption key using a **key encrypting key**; and (2) store

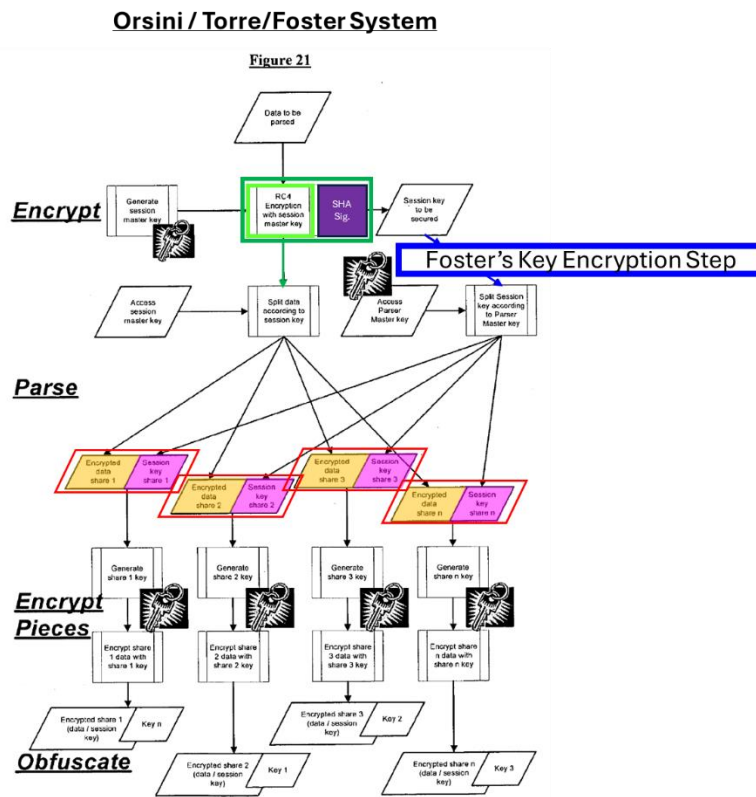
the **encrypted encryption key** instead of the encryption key itself. EX1007, [0053].

EX1002, ¶522.



EX1007, Figure 3.

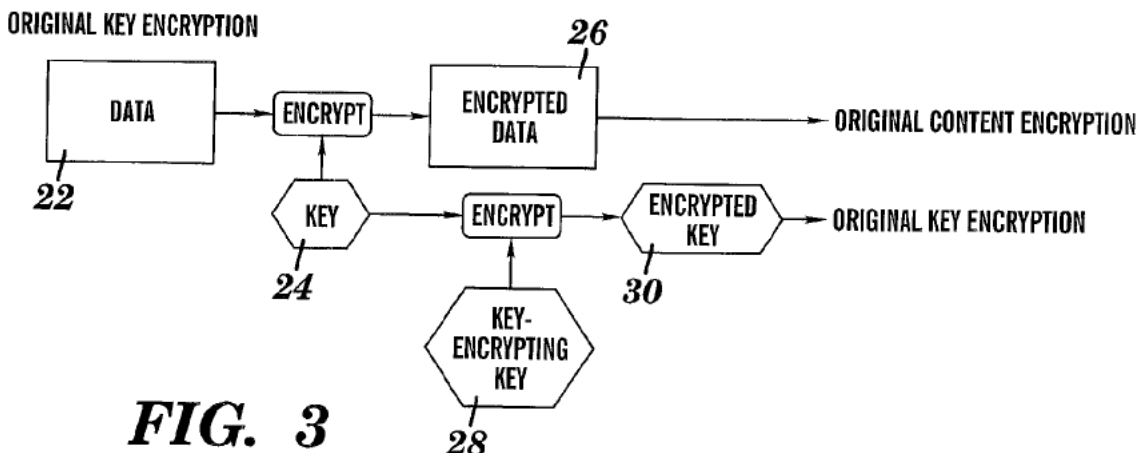
The resulting combination is illustrated below. EX1002, ¶523.



EX1008, Figure 21.

A POSITA would have been motivated to combine Orsini/Torre with Foster to improve the security of the encryption key. A POSITA would have understood that it is important to keep the encryption key secure so that a third party cannot decrypt the encrypted data. Accordingly, a POSITA would have been motivated to encrypt the encryption key before storing it, as in Foster. EX1007, [0053]. The combination would have merely amounted to applying a known technique (Foster's key encryption scheme) to a known device (Orsini/Torre System) ready for improvement (for improved security of the stored encryption key) to yield predictable results (improving the stored encryption key's security). EX1002, ¶¶524-526.

A POSITA would have had a reasonable expectation of success in combining Orsini/Torre system with Foster. **First**, Orsini/Torre and Foster are both directed to the same technical problem of distributing secret information securely, and to substantially similar solutions that split encrypted information to shares/packets and distributing the split encrypted information along with information on the encryption key. EX1003, [0049]-[0052], [0061]-[0062]; EX1007, [0053]; EX1008, Abstract, [0278]-[0280], [0291]. And the modification would have merely entailed inserting an additional encryption step to encrypt the encryption key before placing it in the tag, as in Foster's Figure 3. EX1002, ¶¶527-529, 531.



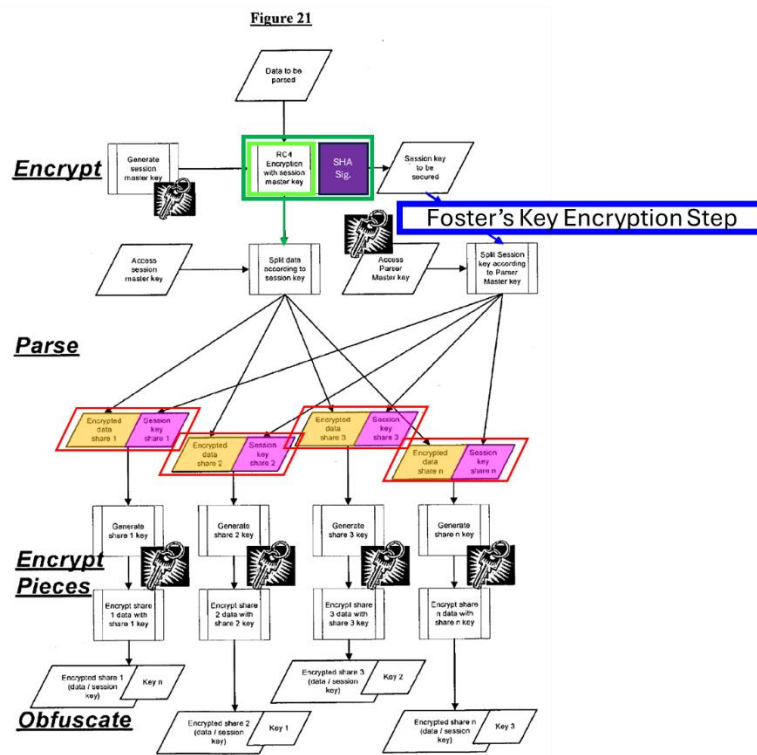
EX1007, Figure 3.

Second, a POSITA would have understood that implementing Foster's key encryption scheme in Orsini/Torre System would have merely entailed writing a program code. That was well within the skill of a POSITA. EX1003, [0141]. *Keynetik*, 2023 WL 2003932, at *2. EX1002, ¶530.

2. Claim 7

Orsini/Torre in view of Foster teaches the additional limitations of Claim 7. In Orsini/Torre/Foster System, the system *encrypts the encryption key* using a *key encrypting key*; and provides the *encrypted key* as part of the tag. EX1007, [0053]. See Section IX.G.1 [Orsini/Torre/Foster]. EX1002, ¶¶532, 534.

Orsini / Torre/Foster System



EX1008, Figure 21.

Then it parses and splits the **encrypted data** and the **encrypted encryption key** into **shreds**, as explained in Section IX.E.3.vi [1E]. Thus, Orsini/Torre/Foster includes in the **shreds** (“*plurality of shares*”) portions of the **encrypted encryption key** (“*data indicative of the encrypted encryption key*”). EX1002, ¶533.

3. Claim 8

Orsini/Torre/Foster teaches the additional limitation of Claim 8. EX1002, ¶¶535, 539.

First, Orsini/Torre/Foster teaches “storing the **second key** outside of the **plurality of shares**.” In Orsini/Torre/Foster, the data recovery process uses the **key-**

encrypting key to decrypt the encrypted encryption key. EX1007, [0055], [0061]. That key-encrypting key is cached in memory. EX1007, [0058]. Accordingly, Orsini/Torre/Foster stores the key-encrypting key (“second key”) outside of the shreds (“plurality of shares”). EX1002, ¶536.

Second, Orsini/Torre/Foster teaches “the data set is restorable by accessing less than all, but at least a threshold number of, the plurality of shares” as explained in Section IX.E.3.viii [1G]. EX1002, ¶537.

Third, Orsini/Torre/Foster teaches accessing and using key-encrypting key (“second key”) to restore the input data (“data set”). Orsini teaches that “[t]o restore the original data format, the steps are reversed.” EX1008, [0300]. Thus, in Orsini/Torre/Foster, the key encrypting key is used to decrypt the encrypted encryption key, and the encryption key is then used to decrypt the encrypted data, thereby restoring the original data set. EX1007, [0055], [0058], [0061]. EX1002, ¶538.

4. Claims 16-17, 25-26

In claims 16 and 25, the additional limitation is materially the same as the additional limitation of claim 7; in claims 17 and 26, the additional limitation is materially the same as the additional limitation of claim 8. Thus, Orsini/Torre/Foster renders claims 16-17, 25-26 obvious for the reasons provided above for Ground VII, Claims 7 and 8. EX1002, ¶540.

X. Conclusion

Petitioner respectfully requests that the Board institute *inter partes* review, hold the challenged claims unpatentable, and cancel the challenged claims.

Dated: July 9, 2025

Respectfully submitted,

/Taeg Sang Cho/

Taeg Sang Cho (Reg. No. 69,618)

DESMARAIS LLP

230 Park Ave

New York, NY 10169

212-808-1060 (telephone)

212-351-3401 (facsimile)

Counsel for Petitioner

International Business Machines Corporation

CERTIFICATION UNDER 37 C.F.R. § 42.24(D)

I hereby certify that this Petition for *Inter Partes* Review of U.S. Patent No. 8,271,802 has, excluding the portions exempted under 37 C.F.R. § 42.24(a), 13,938 words as counted by the word-processing system used to prepare this document, in compliance with 37 C.F.R. § 42.24(d).

Dated: July 9, 2025

Respectfully submitted,

/Taeg Sang Cho/

Taeg Sang Cho (Reg. No. 69,618)

DESMARAIS LLP

230 Park Ave

New York, NY 10169

212-808-1060 (telephone)

212-351-3401 (facsimile)

Counsel for Petitioner

International Business Machines Corporation

CERTIFICATE OF SERVICE

Under 37 C.F.R. §§ 42.6(e) and 42.105, the undersigned certifies that on July 9, 2025, complete copies of the foregoing and any accompanying exhibits were caused to be served by sending them via Federal Express Priority Overnight shipping, which is at least as fast and reliable as U.S. Priority Mail Express, to the correspondence address of record for U.S. Patent No. 8,271,802 as indicated in Patent Center:

Security First Innovations, LLC
c/o Farjami & Farjami LLP
26522 La Alameda Ave., Suite 360
Mission Viejo, CA

Dated: July 9, 2025

Respectfully submitted,

/Taeg Sang Cho/
Taeg Sang Cho (Reg. No. 69,618)
DESMARAIS LLP
230 Park Ave
New York, NY 10169
212-808-1060 (telephone)
212-351-3401 (facsimile)

Counsel for Petitioner
International Business Machines Corporation