

UNITED STATES PATENT AND TRADEMARK OFFICE

---

**BEFORE THE PATENT TRIAL AND APPEAL BOARD**

---

INTERNATIONAL BUSINESS MACHINES CORPORATION

Petitioner

v.

SECURITY FIRST INNOVATION, LLC

Patent Owner

---

Case No. IPR2025-01201

U.S. Patent No. 8,904,194

---

**PETITION FOR *INTER PARTES* REVIEW OF**  
**U.S. PATENT NO. 8,904,194**

CHALLENGING CLAIMS 1–20  
UNDER 35 U.S.C. § 312 AND 37 C.F.R. § 42.104

**TABLE OF CONTENTS**

|                                                                          | <b><u>Page</u></b> |
|--------------------------------------------------------------------------|--------------------|
| 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. Grounds For Challenge .....                                           | 4                  |
| VI. The '194 Patent.....                                                 | 4                  |
| A. Prosecution History .....                                             | 10                 |
| VII. Level Of Ordinary Skill In The Art.....                             | 11                 |
| VIII. Claim Construction.....                                            | 11                 |
| IX. Specific References And Grounds For Challenge.....                   | 12                 |
| A. Ground I: Dickinson And Hardjono Render Obvious Claims 1–<br>20. .... | 12                 |
| 1. Dickinson (EX1003) .....                                              | 12                 |
| 2. Hardjono (EX1004).....                                                | 18                 |
| 3. Dickinson in view of Hardjono (“Dickinson/Hardjono”).....             | 20                 |
| 4. Claim 1 .....                                                         | 34                 |

Petition for *Inter Partes* Review of U.S. Patent No. 8,904,194  
Claims 1-20

|     |                                                                                        |    |
|-----|----------------------------------------------------------------------------------------|----|
| 5.  | Claim 2 .....                                                                          | 49 |
| 6.  | Claim 3 .....                                                                          | 53 |
| 7.  | Claim 4 .....                                                                          | 54 |
| 8.  | Claim 5 .....                                                                          | 55 |
| 9.  | Claim 6 .....                                                                          | 56 |
| 10. | Claim 7 .....                                                                          | 56 |
| 11. | Claim 8 .....                                                                          | 63 |
| 12. | Claim 9 .....                                                                          | 64 |
| 13. | Claim 14 .....                                                                         | 67 |
| 14. | Claim 16 .....                                                                         | 73 |
| 15. | Claims 10–13, 15, and 17–20 .....                                                      | 73 |
| B.  | Ground II: Dickinson, Hardjono and Moulton Render Obvious<br>Claims 2, 8, And 15 ..... | 74 |
| 1.  | Moulton (EX1018) .....                                                                 | 74 |
| 2.  | Dickinson in View of Hardjono and Moulton .....                                        | 76 |
| 3.  | Claim 2 .....                                                                          | 79 |
| 4.  | Claim 8 .....                                                                          | 82 |
| 5.  | Claim 15 .....                                                                         | 83 |
| X.  | Conclusion .....                                                                       | 83 |

**TABLE OF AUTHORITIES**

**Page(s)**

**Cases**

*Keynetik, Inc. v. Samsung Elecs. Co.*,  
No. 2022-1127, 2023 WL 2003932 (Fed. Cir. Feb. 15, 2023)..... 25, 30, 33

*Samsung Elecs. Am., Inc. v. Prisia Eng’g Corp.*,  
948 F.3d 1342 (Fed. Cir. 2020) .....60

**Statutes**

35 U.S.C. § 102 ..... 3, 11, 12

35 U.S.C. § 103 .....3, 4

35 U.S.C. § 314(a) .....4

**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**

| <b>Exhibit No.</b> | <b>DESCRIPTION</b>                                                                                                                                                                                                               |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1001               | U.S. Patent No. 8,904,194 (“’194 Patent”)                                                                                                                                                                                        |
| 1002               | Declaration of Dr. Erez Zadok                                                                                                                                                                                                    |
| 1003               | PCT Patent Application Publication No. 01/022322 (“Dickinson”)                                                                                                                                                                   |
| 1004               | U.S. Patent No. 6,363,481 (“Hardjono”)                                                                                                                                                                                           |
| 1005               | Claudia Canali <i>et al.</i> , <i>Performance Comparison of Distributed Architectures for Content Adaptation and Delivery of Web Resources</i> , 25 IEEE Int’l Conf. on Distributed Comput. Sys. Workshops 331 (2005) (“Canali”) |
| 1006               | File History of U.S. Patent No. 8,904,194 (“’194 File History”)                                                                                                                                                                  |
| 1007               | File History of U.S. Patent Application Serial No. 11/258,839 (“’839 App. File History”)                                                                                                                                         |
| 1008               | Reserved                                                                                                                                                                                                                         |
| 1009               | U.S. Provisional Application Serial No. 60/622,146 (“Provisional Application No. 60/622,146”)                                                                                                                                    |
| 1010               | U.S. Provisional Application Serial No. 60/718,185 (“Provisional Application No. 60/718,185”)                                                                                                                                    |
| 1011               | Microsoft Computer Dictionary                                                                                                                                                                                                    |
| 1012               | Bruce Schneier, <i>Applied Cryptography</i> , 2nd ed., 1996, excerpts (“Schneier”)                                                                                                                                               |
| 1013               | Highlighted Comparison Between Dickinson and the ’194 Patent                                                                                                                                                                     |
| 1014               | Yitzhak Birk, “Random RAIDs with Selective Exploitation of Redundancy for High Performance Video Servers,” IEEE 1997 (“Birk”)                                                                                                    |
| 1015               | U.S. Patent Publication No. 2003/0016596A1 (“Chiquoine”)                                                                                                                                                                         |
| 1016               | Curriculum Vitae of Dr. Erez Zadok                                                                                                                                                                                               |
| 1017               | John Kubiawicz, et al., “OceanStore: An Architecture for Global-Scale Persistent Storage,” ACM 2000 (“Kubiawicz”)                                                                                                                |
| 1018               | U.S. Patent Publication No. 2001/0034795 (“Moulton”)                                                                                                                                                                             |
| 1019               | U.S. Patent Publication No. 2003/0046551 (“Brennan”)                                                                                                                                                                             |
| 1020               | U.S. Patent Publication No. US2002/0049655 (“Bennet”)                                                                                                                                                                            |

Petition for *Inter Partes* Review of U.S. Patent No. 8,904,194  
Claims 1-20

| <b>Exhibit No.</b> | <b>DESCRIPTION</b>                                                                                                                                                                                                                                                |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1021               | <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”) |

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

**LISTING OF CLAIMS**

**Claim 1**

[1Pre] A method for securely storing and retrieving data, the method comprising:

[1A-1] generating, using an electronic computing system that includes processing circuitry, a plurality of shares by performing a cryptographic operation on a data set and

[1A-2] distributing the data set in the plurality of shares such that the data set can be reconstructed using any subset of the shares that includes at least a minimum number less than all of shares;

[1B] storing the plurality of shares at a plurality of storage devices;

[1C] receiving, the electronic computing system, request to retrieve the data set;

[1D] identifying from the plurality of storage devices a set of fastest-responding storage devices necessary to retrieve the minimum number of shares, wherein the set of fastest-responding storage devices are identified based at least in part on the response time of the storage devices;

[1E] retrieving from the set of fastest-responding storage devices, the minimum number of shares;

[1F] reconstructing the data set using the minimum number of shares; and

[1G] sending the data set responsive to the request.

**Claim 2**

[2] The method of claim 1, wherein storing the shares comprises:

[2A] storing a first subset of the shares at a first subset of the plurality of storage devices that is physically located at a first data center; and

[2B] storing a second subset of the shares at a second subset of the plurality of storage devices that is physically located at a second data center, the first data center being geographically separated from the second data center.

**Claim 3**

[3] The method of claim 1, further comprising establishing secure connections between the electronic computing system and the plurality of storage devices.

**Claim 4**

[4] The method of claim 1, wherein the plurality of data blocks shares contain a substantially random distribution of the data set.

**Claim 5**

[5] The method of claim 1, wherein the data set can be reconstructed from shares received from at least two storage devices.

**Claim 6**

[6] The method of claim 1, wherein the shares are encrypted with a corresponding number of different keys.

**Claim 7**

**[7Pre]** An electronic computing system for securely storing and retrieving data, the electronic computing system comprising:

**[7A]** a processing unit; and

**[7B]** a system memory comprising instructions that, when executed by the processing unit, cause the processing unit to:

**[7B-1]** generate a plurality of shares by performing a cryptographic operation on a data set and distributing the data set in the plurality of shares such that the data set can be reconstructed using any subset of the shares that includes at least a minimum number less than all of the shares and such that the data set cannot be reconstructed using any subset of the shares that includes fewer than the minimum number of the shares;

**[7B-2]** store the plurality of shares at a plurality of storage devices;

**[7B-3]** receive, via the primary interface, a request to retrieve the data set;

**[7B-4]** identify, from the plurality of storage devices, a set of fastest-responding storage devices necessary to retrieve the minimum number of shares, wherein the set of fastest-responding storage devices are identified based at least in part on the response time of the storage devices, retrieve from the set of fastest-responding storage devices the minimum number of shares;

[7B-5] reconstruct the data set using exclusively the minimum number of shares; and

[7B-6] send the data set responsive to the request.

**Claim 8**

[8] The electronic computing system of claim 7, wherein the instructions cause the processing unit to store a first subset of the shares at a first subset of the plurality of storage devices that is physically located at a first data center and to store a second subset of the shares at a second subset of the plurality of storage devices that is physically located at a second data center, the first data center being geographically separated from the second data center.

**Claim 9**

[9] The electronic computing system of claim 7, wherein the instructions further cause the processing unit to generate the plurality of shares in response to receiving a request to store the data set.

**Claim 10**

[10] The electronic computing system of claim 7, wherein the instructions further cause the processing unit to establish secure connections between the electronic computing system and the plurality of storage devices.

**Claim 11**

[11] The electronic computing system of claim 7, wherein the plurality of shares contain a substantially random distribution of the data set.

**Claim 12**

[12] The electronic computing system of claim 7, wherein the data set can be reconstructed from shares received from at least two storage devices.

**Claim 13**

[13] The electronic computing system of claim 7, wherein the shares are encrypted with a corresponding number of different keys.

**Claim 14**

[14Pre] A non-transitory computer-readable storage medium comprising instructions that, when executed at an electronic computing device, cause the electronic computing device to:

[14A] receive a request to write a data set to a storage location;

[14B] generate a plurality of shares by performing a cryptographic operation on the data set and distributing the data set in the plurality of shares such that the data set can be reconstructed using any subset of the shares that includes at least a minimum number less than all of the shares and such that the data set cannot be reconstructed using any subset of the shares that includes fewer than the minimum number of the shares;

[14C] store the plurality of shares at a plurality of storage devices;

[14D] receive a request to retrieve the data set;

[14E] identify, from the plurality of storage devices, a set of fastest-responding storage devices necessary to retrieve the minimum number of shares, wherein the set of fastest-responding storage devices are identified based at least in part on the response time of the storage devices;

[14F] retrieve from the set of fastest-responding storage devices, the minimum number of shares;

[14G] reconstruct the data set using exclusively the minimum number of shares; and

[14H] send the data set responsive to the request.

**Claim 15**

[15] The non-transitory computer-readable storage medium of claim 14, wherein the instructions cause the processing unit to store a first subset of the shares at a first subset of the plurality of storage devices that is physically located at a first data center and to store a second subset of the shares at a second subset of the plurality of storage devices that is physically located at a second data center, the first data center being geographically separated from the second data center.

**Claim 16**

[16] The non-transitory computer-readable storage medium of claim 14, wherein the instructions further cause the processing unit to generate the plurality of shares in response to receiving the request to write the data set to the storage location.

**Claim 17**

[17] The non-transitory computer-readable storage medium of claim 14, wherein the instructions further cause the processing unit to establish secure connections between the electronic computing device and the plurality of storage devices.

**Claim 18**

[18] The non-transitory computer-readable storage medium of claim 14, wherein the plurality of shares contain a substantially random distribution of the data set.

**Claim 19**

[19] The non-transitory computer-readable storage medium of claim 14, wherein the data set can be reconstructed from shares received from at least two storage devices.

**Claim 20**

[20] The non-transitory computer-readable storage medium of claim 14, wherein the shares are encrypted with a corresponding number of different keys.

**I. Introduction**

Petitioner respectfully requests cancellation of all claims in U.S. Patent No. 8,904,194 (“’194 Patent”).

**II. Mandatory Notices**

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

Petitioner identifies itself as a real party-in-interest (“RPI”).

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

The ’194 Patent is currently asserted against Petitioner by Patent Owner in *Security First Innovations, LLC v. International Business Machines Corp.*, No. 1:25-cv-00514 (EDVA) (“Litigation”). Petitioner will file petitions for *inter partes* review of U.S. Patent Nos. 8,271,802, 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:

| <b>Lead Counsel</b>                                                                                                                                                         | <b>First Back-up Counsel</b>                                                                                                                                                             |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Taeg Sang Cho (Reg. No. 69,618)<br>Desmarais LLP<br>230 Park Ave<br>New York, NY 10169<br>Telephone: (212) 351-3400<br>Email: tcho@desmaraisllp.com                         | Kurt Fredrickson (Reg. No. 77,281)<br>Desmarais LLP<br>101 California Street, Suite 3000<br>San Francisco, CA 94111<br>Telephone: (415) 573-1900<br>Email: kfredrickson@desmaraisllp.com |
| <b>Back-up Counsel</b>                                                                                                                                                      | <b>Back-up Counsel</b>                                                                                                                                                                   |
| Lindsey Miller ( <i>pro hac vice</i><br>forthcoming)<br>Desmarais LLP<br>230 Park Ave<br>New York, NY 10169<br>Telephone: (212) 351-3400<br>Email: lmiller@desmaraisllp.com | Laura Avena (Reg. No. 78,920)<br>Desmarais LLP<br>230 Park Ave<br>New York, NY 10169<br>Telephone: (212) 351-3400<br>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

kfredrickson@desmaraisllp.com

lmiller@desmaraisllp.com

lavena@desmaraisllp.com

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 '194 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 1-20 ("Challenged Claims") of the '194 Patent.

#### A. Prior Art References

The '194 Patent was filed on May 10, 2012, claiming priority to U.S. Provisional Application Nos. 60/622,146, filed October 25, 2004; and 60/718,185, filed September 16, 2005. EX1001, Cover. Accordingly, this Petition applies pre-AIA provisions of 35 U.S.C. §§ 102 and 103.

The following references are pertinent to the grounds of unpatentability explained below:

| Pat. Pub. No. or Title                  | Publication/Priority Date | Prior Art Under At Least (35 U.S.C.) |
|-----------------------------------------|---------------------------|--------------------------------------|
| WO 2001/022322<br>("Dickinson")(EX1003) | published March 29, 2001  | 102(b)                               |

| Pat. Pub. No. or Title                | Publication/Priority Date  | Prior Art Under At Least (35 U.S.C.) |
|---------------------------------------|----------------------------|--------------------------------------|
| US 6,363,481<br>("Hardjono")(EX1004)  | issued March 26, 2002      | 102(b)                               |
| US2001/0034795<br>("Moulton")(EX1018) | published October 25, 2001 | 102(b)                               |

### B. Grounds For Challenge

Petitioner requests cancellation of claims 1-20 of the '194 Patent under 35 U.S.C. § 103 based on the following Grounds.

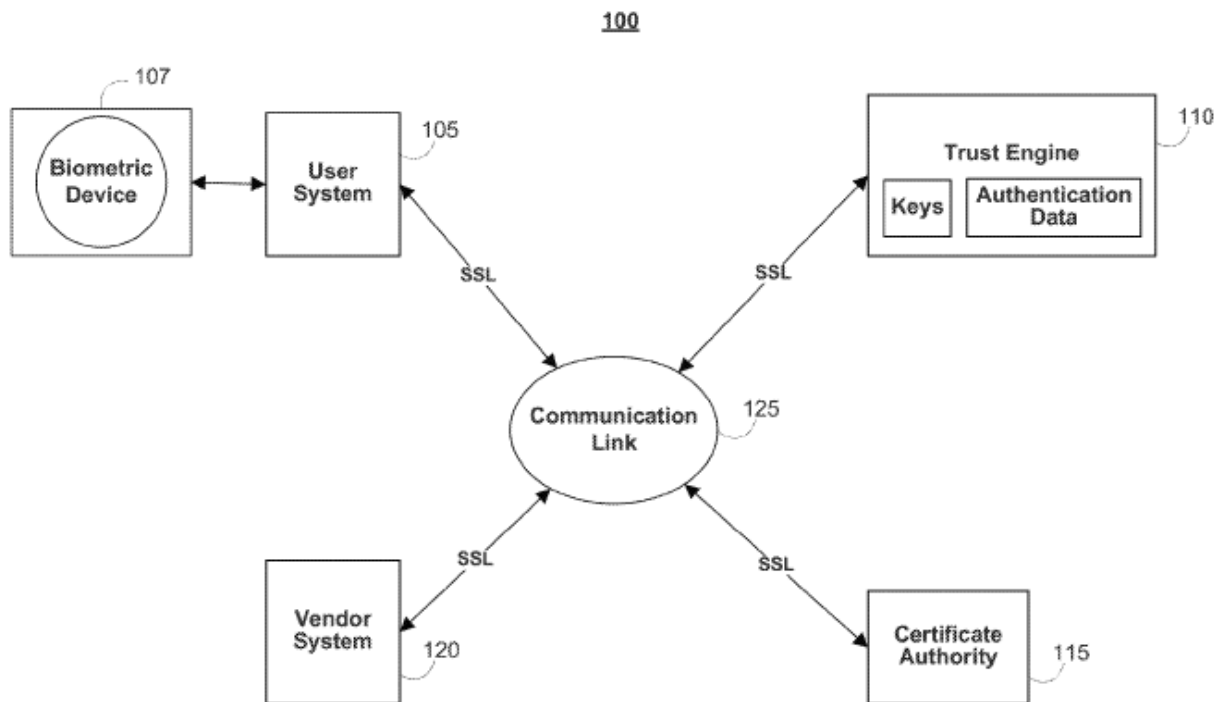
| Grounds | Claims   | Prior Art References         |
|---------|----------|------------------------------|
| I       | 1-20     | Dickinson, Hardjono          |
| II      | 2, 8, 15 | Dickinson, Hardjono, Moulton |

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

### VI. The '194 Patent

Figure 1 of the '194 Patent illustrates a "cryptographic system 100." EX1001, 8:42-47. Among other things, the cryptographic system includes a "user system 105" and a "vendor system 120" linked to a "trust engine 110" via a "communication link 125." EX1001, 8:42-47, FIG. 1. The trust engine 110 provides "complete cryptographic functionality," such as encryption and decryption. EX1001, 10:4-17. The trust engine also contains a "depository 210," which "comprises one or more data storage facilities, such as, for example, a directory server, a database server, or

the like.” EX1001, 12:1–24, 13:20–34, FIG. 2. The trust engine 110 also includes “authentication engine 215” and “cryptographic engine 220,” which “employ their respective data splitting modules to divide sensitive data into undecipherable portions, and then transmit one or more undecipherable portions of the sensitive data to a particular data storage facility” located in the depository 210. EX1001, 18:32–46.



**FIG. 1**

EX1001, FIG. 1.

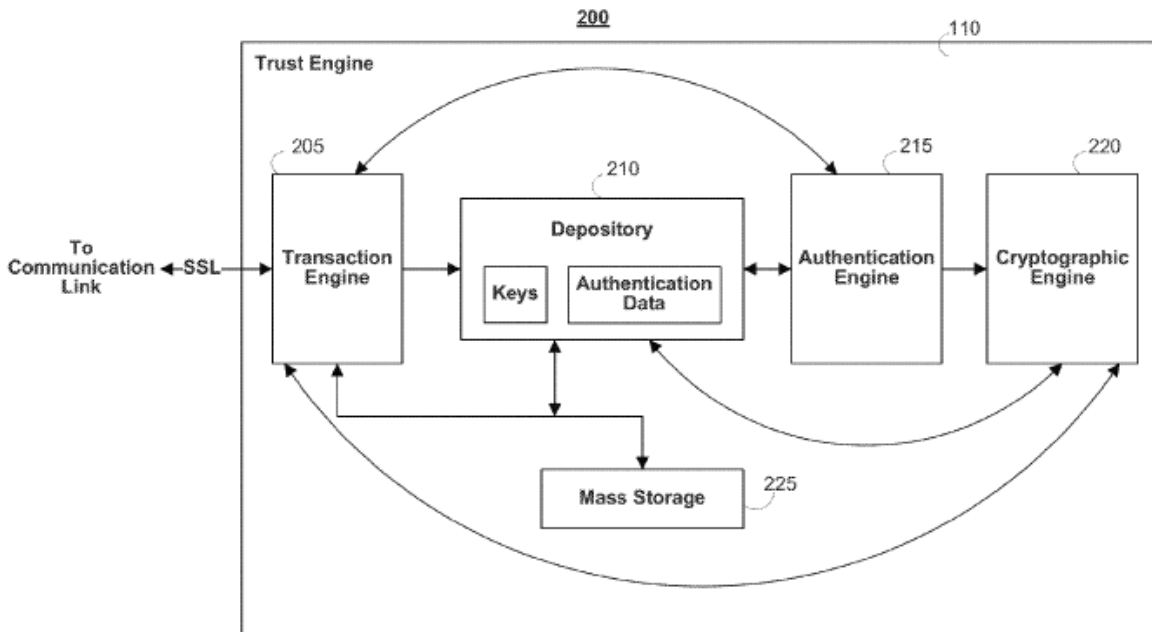


FIG. 2

EX1001, FIG. 2. EX1002, ¶¶148–149.

The “depository system ... advantageously comprises multiple data storage facilities.” EX1001, 18:8–31. Figure 7 of the ’194 Patent illustrates the “depository 210” of the trust engine 110 that secures a data set by generating shares from the data set and distributing the shares in data storage facilities D1–D4. EX1001, 18:47–19:5.

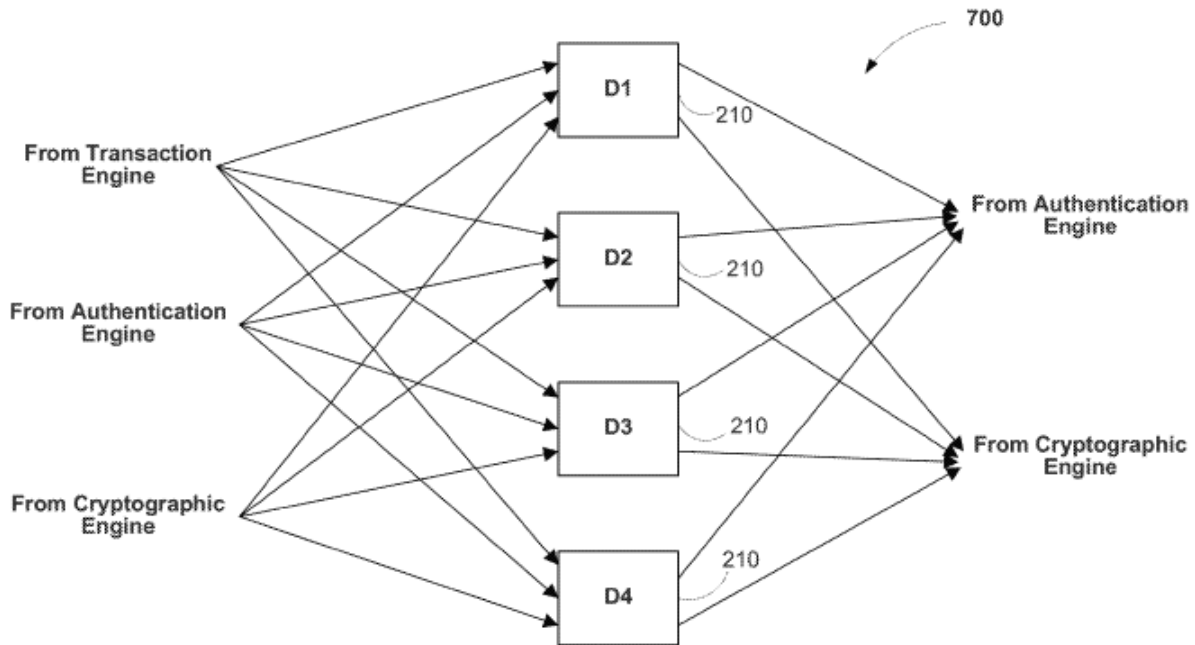


FIG. 7

EX1001, FIG. 7. EX1002, ¶150.

Figure 7's system can perform a method for securely storing and retrieving data, as described in Figure 8. EX1001, 19:37–67. For example, the system receives a request to write or store a data set, EX1001, 18:8–31, and the system performs a cryptographic operation on the data set to create multiple data shares, EX1001, 19:16–36. Then, the system distributes the data shares so that the data set can be reconstructed using any subset of the shares that includes at least a minimum number less than all the shares. EX1001, 20:1–9, 20:38–60. For example, the system creates two random numbers, values, strings, or set of bits “A” and “C.” EX1001, 19:16–67. The system also combines A and C with the sensitive data “S” to make new

numbers “B” and “D,” respectively, such as through an exclusive/or (XOR) operation:  $B = A \text{ XOR } S$ ,  $D = C \text{ XOR } S$ . EX1001, 19:37–67. Then, the system pairs A, B, C, and D “such that none of the pairings contain sufficient data, by themselves, to reorganize and decipher the original sensitive data S.” EX1001, 19:37–67. For example, the data may be paired and stored as AC, AD, BC, and BD and such that any two provide one of (A and B) and one of (C and D). EX1001, 20:38–60. Each of those pairs are stored at the four data storage facilities D1 through D4, exemplified in Figure 7. EX1001, 19:37–20:9.

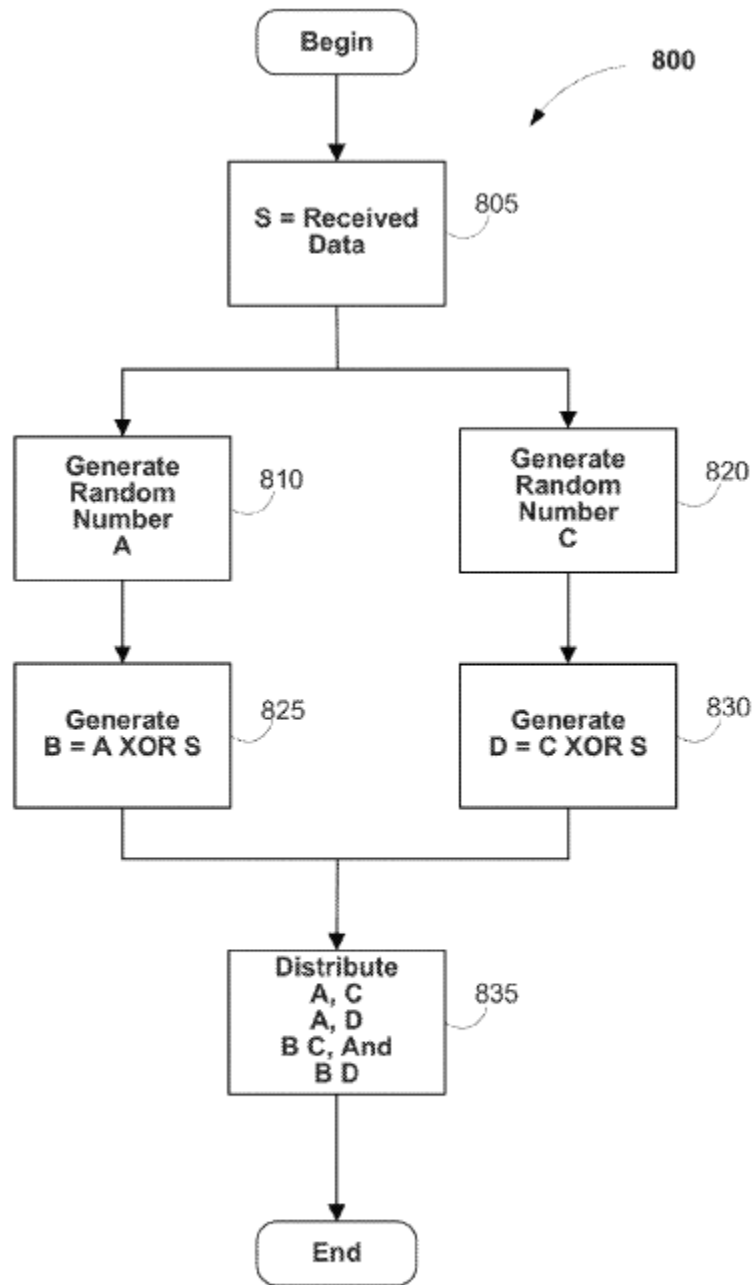


FIG. 8

EX1001, FIG. 8. EX1002, ¶¶151–152.

When a user wants to recover the original data set, the system and method in the '194 Patent retrieve the data shares from a set of “fastest-responding” storage devices that contain a minimum number of shares necessary to recreate the data set. Specifically, the system identifies a set of fastest-responding storage devices needed to retrieve the minimum number of shares. EX1001, 18:8–31. Then, the system retrieves the minimum number of shares from the identified storage devices. EX1001, 18:8–31, 20:1–9, 20:38–60. To identify the “fastest-responding” devices, the system “may broadcast requests to particular data storage facilities based on a wide number of criteria, such as, for example, response time, server loads, maintenance schedules, or the like.” EX1001, 18:8–31. Subsequently, the system combines the retrieved shares to reconstruct the original data set. EX1001, 20:38–60. EX1002, ¶153.

#### **A. Prosecution History**

The '194 Patent originated from U.S. Patent Application No. 13/468,562, filed May 10, 2012; which is a continuation of U.S. Patent Application No. 11/258,839, filed October 25, 2005; which, in turn, claims priority to U.S. Patent Provisional Application Nos. 60/622,146 (filed October 25, 2004) and 60/718,185 (filed September 16, 2005). EX1006, 3; EX1007, 240. EX1002, ¶154.

The '194 Patent's claims issued after three rounds of Office Actions and three amendments. Initially, the Examiner issued a Non-Final Rejection rejecting claims

1–20 under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent Application Publication No. 2010/0162003 (“Dodgson”). EX1006, 771–84. In response, Applicant amended the independent claims to recite “identifying, from the plurality of storage devices a set of fastest-responding storage devices necessary to retrieve the minimum number of shares, wherein the set of fastest-responding storage devices are identified based at least in part on the response time of the storage devices.” EX1006, 793–99. Subsequently, the Examiner allowed the amended claims. EX1006, 820–26. EX1002, ¶¶155–157.

## **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 ’194 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, ¶158.

## **VIII. Claim Construction**

Petitioner submits that no terms require construction for the purposes of this IPR.<sup>1</sup> EX1002, ¶161.

---

<sup>1</sup> Petitioner reserves all rights to raise claim construction arguments in other forums.

## **IX. Specific References And Grounds For Challenge**

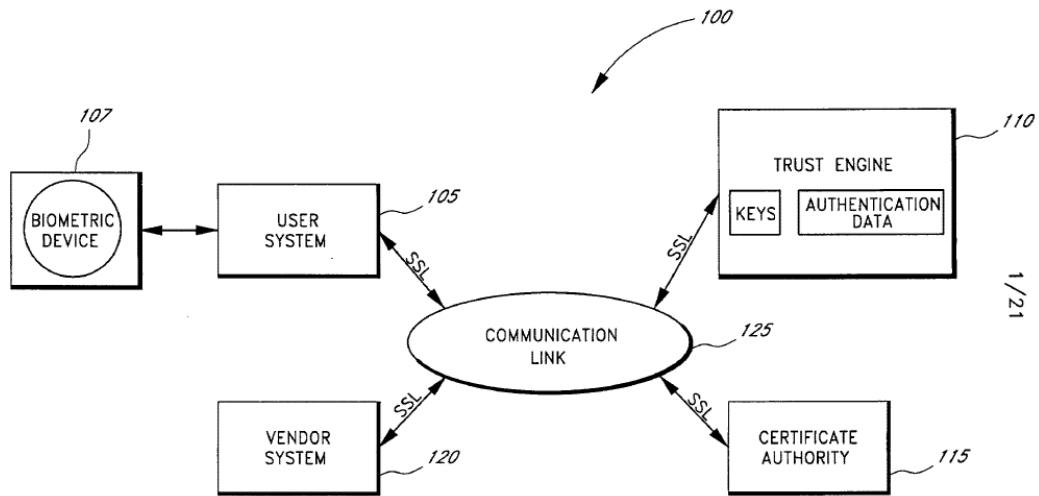
### **A. Ground I: Dickinson And Hardjono Render Obvious Claims 1–20.**

#### **1. Dickinson (EX1003)**

Dickinson includes substantially identical disclosures to the '194 Patent. In fact, Dickinson's Figures 1–20 and the associated descriptions are substantially identical to the '194 Patent's Figures 1–20 and the associated descriptions. *See* EX1013 (providing a highlighted version of the '194 Patent's specification where the highlighted portion appears in Dickinson). Because the '194 Patent's Figures 1–20 and the associated descriptions provide written description support for most of the '194 Patent's claim limitations, Dickinson, too, provides invalidating disclosures for most of the '194 Patent's claim limitations. EX1002, ¶165.

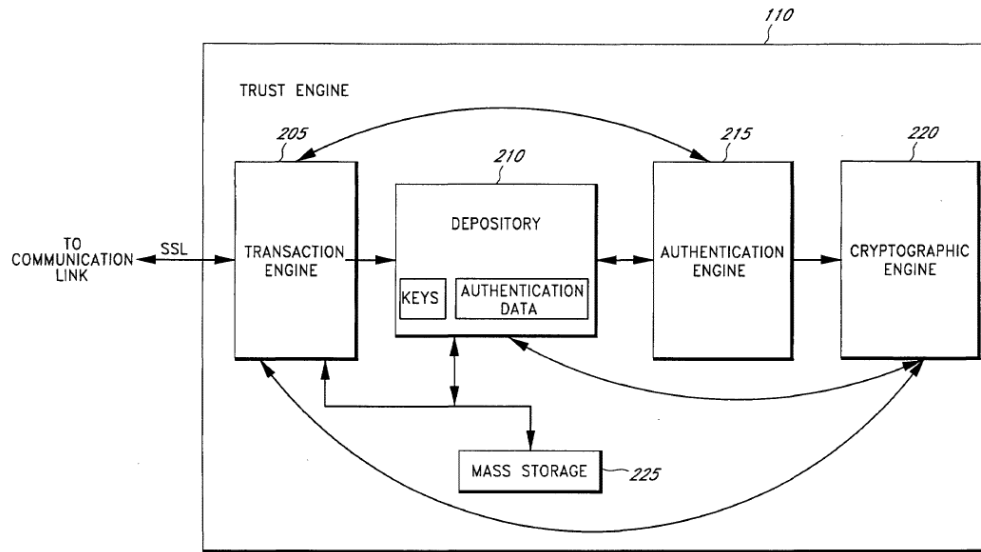
Figure 1 of Dickinson illustrates a “cryptographic system 100.” EX1003, 9:22–24. Among other things, the cryptographic system includes a “user system 105” and a “vendor system 120” linked to a “trust engine 110” via a “communication link 125.” EX1003, 9:22–24, FIG. 1. The trust engine 110 provides “complete cryptographic functionality,” such as encryption and decryption. EX1003, 11:3–9. The trust engine also contains a “depository 210,” which “comprises one or more data storage facilities, such as, for example, a directory server, a database server, or the like.” EX1003, 13:5–17, 14:17–24, FIG. 2. The trust engine 110 also includes “authentication engine 215” and “cryptographic engine 220,” which “employ their

respective data splitting modules to divide sensitive data into undecipherable portions, and then transmit one or more undecipherable portions of the sensitive data to a particular data storage facility” located in the depository 210. EX1003, 13:5–17, 19:27–34.



**FIG. 1**

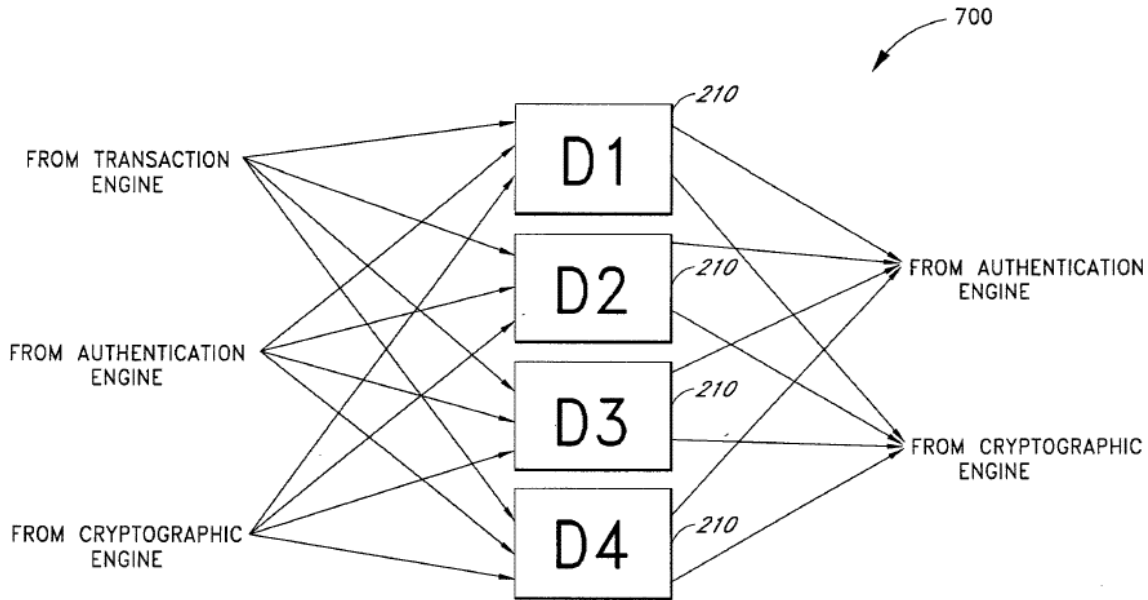
EX1003, FIG. 1.



**FIG. 2**

EX1003, FIG. 2. EX1002, ¶¶166–168.

The “depository system ... advantageously comprises multiple data storage facilities.” EX1003, 19:16–26. Figure 7 of Dickinson illustrates the “depository 210” of the trust engine 110 that secures a data set by generating shares from the data set and distributing the shares in data storage facilities D1–D4. EX1003, 20:1–14.



**FIG. 7**

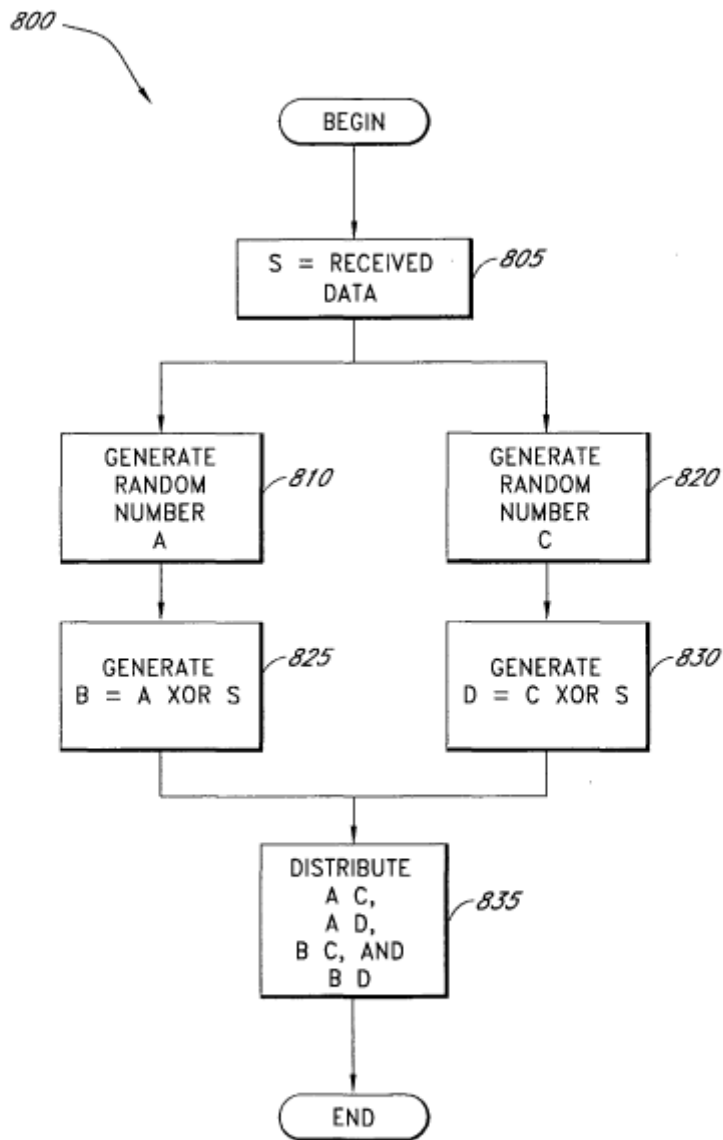
EX1003, FIG. 7. EX1002, ¶169.

Figure 7's system can perform a method for securely storing and retrieving data, as described in Figure 8. EX1003, 20:30–21:9. For example, the system receives a request to write or store a data set, EX1003, 19:16–34, and the system performs a cryptographic operation on the data set to create multiple data shares, EX1003, 20:20–21:9. EX1002, ¶¶170–171.

Then, the system distributes the data shares so that the data set can be reconstructed using any subset of the shares that includes at least a minimum number less than all the shares. EX1003, 21:10–14, 21:21–32. For example, the system creates two random numbers, values, strings, or set of bits “A” and “C.” EX1003,

20:20–21:9. The system also combines A and C with the sensitive data “S” to make new numbers “B” and “D,” respectively, such as through an exclusive/or (XOR) operation:  $B = A \text{ XOR } S$ ,  $D = C \text{ XOR } S$ . EX1003, 20:30–21:9. Then, the system pairs A, B, C, and D “such that none of the pairings contain sufficient data, by themselves, to reorganize and decipher the original sensitive data S.” EX1003, 20:30–21:9. For example, the data may be paired and stored as AC, AD, BC, and BD and such that any two provide one of (A and B) and one of (C and D). EX1003, 21:21–32. Each of those pairs are stored at the four data storage facilities D1 through D4, exemplified in Figure 7. EX1003, 20:30–21:14.

8/21



**FIG. 8**

EX1003, FIG. 8. EX1002, ¶¶171–172.

When a user wants to recover the original data set, Dickinson’s system and method retrieve the data shares from a set of “fastest-responding” storage devices

that contain a minimum number of shares necessary to recreate the data set. Specifically, the system identifies a set of fastest-responding storage devices needed to retrieve the minimum number of shares. EX1003, 19:16–26. Then, the system retrieves the minimum number of shares from the identified storage devices. EX1003, 19:16–26, 21:10–14, 21:21–32. To identify the “fastest-responding” devices, the system “may broadcast requests to particular data storage facilities based on a wide number of criteria, such as, for example, response time, server loads, maintenance schedules, or the like.” *See* EX1003, 19:16–26. EX1002, ¶173.

Subsequently, the system combines the retrieved shares to reconstruct the original data set. EX1003, 21:21–32. EX1002, ¶174.

## **2. Hardjono (EX1004)**

Hardjono discloses a system and method that securely stores and retrieves data. EX1004, Abstract, 7:65–8:2. Also similarly to the '194 Patent, Hardjono discloses splitting the data into pieces and storing those pieces on multiple storage devices. *See* EX1004, 3:28–39, 5:49–6:14, 6:53–67. And like the '194 Patent, only some data pieces are required to reassemble the original data. *See* EX1004, 3:28–39, 5:49–6:14, 6:53–67. EX1002, ¶¶175–177.

Hardjono discloses a “threshold scheme” to split the data into shares that are stored on different databases, and only some shares are required to recreate the original data. Hardjono applies a “threshold scheme” to the data to create a plurality

of data shares, EX1004, 3:28–38, which are then “distributed to multiple separate databases,” EX1004, 3:40–50. EX1002, ¶¶176, 178–179.

When a user wants the original data, the user uses Hardjono’s network/communication interface to communicate their request. EX1004, 8:18–25. Hardjono discloses a network/communication interface that connects Hardjono’s system to various outside networks, such as a LAN or the Internet. EX1004 8:18–25. A user can thus use the LAN or the Internet to communicate with the system via the network/communication interface, allowing the user to request the stored data. *See* EX1004, 8:18–25. EX1002, ¶180.

When a user requests the data, Hardjono’s system identifies which distributed databases have the fastest response time. Hardjono discloses “identify[ing] databases where  $k$  shares of the block are stored,” doing so “based on determining which of the distributed databases are currently, most easily, or most quickly accessible.” EX1004, 7:57–64. The “most quickly accessible” database refers to the storage devices with the fastest-response time. EX1002, ¶181.

After retrieving the shares from the databases, Hardjono recombines the pieces and returns the original data to the requester. The original data is re-created using at least the  $k$  data pieces. EX1004, 6:53–67. Hardjono discloses that, “[u]pon subsequent receipt of a request for the data,” “[t]he re-constructed data is then

returned to the requester.” EX1003, 3:40–50; *see also* EX1003, 7:65–8:2. EX1002, ¶182.

### **3. Dickinson in view of Hardjono (“Dickinson/Hardjono”)**

Both Dickinson and Hardjono teach segmenting data across multiple storage devices and reconstructing it by reversing that process (*see* Sections IX.A.1 [Dickinson] and IX.A.2 [Hardjono]). Although Dickinson does not explicitly disclose every claimed limitation, those missing details would have been obvious in light of Hardjono. A POSITA would therefore have been motivated to integrate Hardjono’s teachings into Dickinson’s system, with a reasonable expectation of success, as explained below. EX1002, ¶183.

For clarity, the combined teachings of Dickinson and Hardjono—and the motivations for that combination—are organized into three topics: (i) employing the fastest-responding storage devices (Section IX.A.3.i); (ii) transmitting reconstructed data in response to a request (Section IX.A.3.ii); and (iii) encrypting each data segment with a separate key (Section IX.A.3.iii). EX1002, ¶184.

#### **i. Fastest Responding Storage Devices in View of Hardjono**

Both Dickinson and Hardjono disclose identifying storage devices based on characteristics of those devices. Dickinson explains that its “transaction engine 205 may broadcast requests to particular *data storage facilities* based on a wide number

of criteria, such as, for example, *response time*, server loads, maintenance schedules, or the like.” EX1003, 19:24–26. Likewise, Hardjono discloses “*identify[ing] databases* where k shares of the block are stored” including “*based on* determining which of the distributed *databases* are currently, most easily, or *most quickly accessible*.” EX1004, 7:57–64. EX1002, ¶185.

It would have been obvious to combine Dickinson’s system with Hardjono’s teachings such that Dickinson’s trust engine identifies, from among its storage facilities D1–D4, “where k shares of the block are stored,” “based on [] which of the distributed databases are currently, most easily, or *most quickly accessible*.” EX1004, 7:57–64. EX1002, ¶185.

**a. Reasons to Combine Dickinson with Hardjono’s Teaching of Fastest Responding Storage Devices**

A POSITA would have been motivated to implement Hardjono’s method for selecting the fastest-responding storage devices in Dickinson’s architecture. EX1002, ¶186.

*First*, both Dickinson and Hardjono address a substantially similar technical issue and provide a substantially similar solution. For example, Dickinson explains that “the transaction engine 205 may broadcast requests to particular *data storage facilities* based on a wide number of criteria, such as, for example, *response time*, server loads, maintenance schedules, or the like.” EX1003, 19:24–26. Similarly,

Hardjono teaches “*identify[ing] databases* where  $k$  shares of the block are stored” including “*based on* determining which of the distributed *databases* are currently, most easily, or *most quickly accessible*.” EX1004, 7:57–64. Accordingly, a POSITA would have understood that the teachings of Hardjono—using the fastest-responding storage devices to reconstruct the original data—would have been readily applicable to Dickinson. EX1002, ¶187.

*Second*, there would have been a particular motivation to implement Hardjono’s teachings in Dickinson’s trust engine to improve user experience. It was well known that data storage systems with a faster response time—like the databases identified using Hardjono’s technique “based on [] which of the distributed databases are currently, most easily, or *most quickly accessible*” (EX1004, 7:57–64)—would improve the user experience by reducing the amount of wait time for the user. Accordingly, a POSITA would have been motivated to implement Dickinson’s trust engine using Hardjono’s technique to reduce the data retrieval response time. EX1002, ¶188.

*Third*, a POSITA would have been motivated to implement Dickinson’s trust engine using Hardjono’s technique to identify and use the “most quickly accessible” storage devices because a POSITA would have understood that such technique improves the efficiency of Dickinson’s networked system. It was well known that the “most quickly accessible” nodes tend to be geographically closer. *See* EX1005,

5 (“This [response time] performance gain is due to the placement of the adaption servers close to the clients[.]”). Accordingly, by using Hardjono’s technique to identify and use the “most quickly accessible” storage devices in Dickinson’s network system, a POSITA would have understood that Dickinson’s network system can shorten network paths and reduce overall packet-travel distance. By requiring packets to travel less distance, the system as a whole would have been more efficient. EX1002, ¶189.

**Fourth**, a POSITA would have understood that using Hardjono’s technique to identify and use the “most quickly accessible” storage devices in Dickinson’s network system would have improved the network load distribution. For example, a POSITA would have understood that selecting the fastest responding servers “contribut[es] to evenly distribute network load[.]” EX1005, 6. The system as a whole would have been able to serve a larger number of requests more quickly and more efficiently, with less failures and improved user experience, by evenly distributing the network load. Thus, a POSITA would have been motivated to identify and use the fastest-responding servers, as taught by Hardjono, in Dickinson’s system for improved system efficiency. EX1002, ¶190.

**Fifth**, a POSITA would have recognized that this combination simply involves applying a known technique (Hardjono’s method of identifying databases that are currently, most easily, or most quickly accessible) to a known system

(Dickinson’s storage system) suitable for enhancement to yield the predictable result of improving the speed of data retrieval. EX1002, ¶191.

A POSITA would have had a reasonable expectation of success in combining Hardjono’s technique for identifying the most quickly accessible databases with Dickinson’s system. EX1002, ¶192.

**First**, both Dickinson and Hardjono describe methods for selecting databases or shares of data based on performance-related characteristics that enhance the efficiency of the data reconstruction process. *Compare* EX1003, 19:24–26; *with* EX1004, 7:57–64. A POSITA would have had a reasonable expectation of success in combining the teachings of Dickinson and Hardjono as described above: it would simply make explicit that one of the “criteria” referenced in Dickinson (EX1003, 19:24–26) is the ability to identify database locations that are “most quickly accessible” (EX1004, 7:57–64). EX1002, ¶¶193–194.

**Second**, a POSITA would have understood that the combination would have simply involved adding Hardjono’s identification step after Dickinson’s step in which Dickinson’s system receives a request to retrieve the data. A POSITA would have readily understood that implementing this additional step would predictably enhance the speed of data recombination. EX1004, 7:57–64. EX1002, ¶195.

**Third**, a POSITA would have understood that implementing Dickinson’s system using Hardjono’s teachings would have involved simply writing program

code—Dickinson teaches that its trust engine modules are implemented in software. EX1003, 18:20–21 (“The authentication engine 215 also includes the data splitting module 520,” which “advantageously comprises a *software*, hardware, or combination module[.]”), 18:23–24, 19:3–5. That was well within the skill of a POSITA. *Keynetik, Inc. v. Samsung Elecs. Co.*, 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, ¶196.

**ii. Sending Assembled Data to the Requester in Response to a Request in View of Hardjono**

Both Dickinson and Hardjono describe that, once a data-retrieval request is received, the original data is reconstructed by substantively reversing the data splitting process. *See* Sections IX.A.1 [Dickinson], IX.A.2 [Hardjono]. EX1002, ¶198.

Dickinson teaches that vendor system 120 “forwards ... the authentication request to the trust engine 110,” which “includes the transaction engine 205” that “receives the [vendor request] ... and generates a request for the user’s enrollment authentication data to be assembled from the data storage facilities D1 through D4.” EX1003, 28:6–14. Dickinson further notes that “the *sensitive data S* includes enrollment authentication data.” EX1003, 13:5–17, 21:33–22:2. EX1002, ¶199.

Hardjono similarly explains that “[u]pon subsequent receipt of a *request for the data*,” “[t]he *re-constructed data* is then *returned* to the requester.” EX1004, 3:40–50. Hardjono also discloses that the server “re-creates the block from the *k shares*, step 430, and *returns* the *re-created data block* to the *requester*, step 435.” EX1004, 7:65–8:2. EX1002, ¶200.

It would have been obvious to combine these teachings from Dickinson and Hardjono so that, Dickinson’s trust engine assembles the stored segments and returns the reconstructed data “[u]pon subsequent receipt of a request for the data” to the requester, as taught by Hardjono. EX1004, 3:40-50. EX1002, ¶201.

**a. Reasons to Combine Dickinson with Hardjono’s Teaching of Sending Assembled Data to the Requester in Response to a Request**

A POSITA would have been motivated to combine Dickinson with Hardjono’s teaching of transmitting the assembled data in response to a request for at least the following two reasons. EX1002, ¶202.

*First*, Dickinson’s trust engine already reconstructs the data set upon receiving a retrieval request, and failing to forward that reconstructed data to the requester would leave the workflow incomplete and thereby use trust engine’s resources inefficiently. And Hardjono explicitly discloses that “[u]pon subsequent receipt of a request for the data,” the server “re-creates the block from the *k shares* ... and returns the re-created data block to the requester.” EX1004, 3:40–50; 7:65–

8:2. A POSITA would have recognized that this return step is both fundamental to any data-storage-and-retrieval system and an obvious addition to Dickinson's process. EX1002, ¶¶203–204.

**Second**, integrating Hardjono's return step into Dickinson's architecture would have been a straightforward application of a known technique to a substantially similar system. Particularly, Dickinson's data storage architecture is similar to Hardjono's and therefore applying Hardjono's step of returning the assembled data would be apparent to a POSITA. The purpose and effect of the combination remain the same: to enable the system to provide information in response to a request, thereby rendering Dickinson's data storage system fully operational in the same way. EX1002, ¶206.

Moreover, a POSITA would have had a reasonable expectation of successfully integrating Hardjono's technique of sending assembled data to the requester within Dickinson's system for several reasons. **First**, both Dickinson and Hardjono are directed to secure data storage and retrieval. Dickinson describes a cryptographic system in which one or more secure servers or a trust engine store cryptographic keys and user authentication data. EX1003, 9:7–12. Dickinson describes how authentication data is compared against enrollment data to verify users. EX1003, 10:20–32; 38:16–21. Hardjono likewise describes secure network servers. EX1004, Abstract; 7:65–8:2. EX1002, ¶¶207–212.

**Second**, both Dickinson and Hardjono describe partitioning data into discrete portions and storing those portions across multiple storage devices. Dickinson’s data splitting module “advantageously comprises a software, hardware, or combination module having the ability to mathematically operate on various data so as to substantially randomize and split the data into portions,” (EX1003, 18:21–22), and its authentication engine likewise “separates the authentication data associated with each individual authentication instance in STEP 1605” (EX1003, 38:16–21). Similarly, Hardjono teaches “creating a plurality of ‘shares’ of the data” and distributing those shares among multiple databases to increase security and fault tolerance. EX1004, 3:28–39; 5:49–6:14; 6:53–67. EX1002, ¶213.

**Third**, both Dickinson and Hardjono teach that only a subset of the distributed data portions is required to reconstruct the original data. Dickinson discloses that “only data from two of the multiple data storage facilities, D1 through D4, are needed to decipher and reassemble the sensitive data.” EX1003, 20:1–14. Likewise, Hardjono describes a “threshold scheme [that] generates multiple [n] ‘shares’” of a document but requires only a subset (k) of those shares (with  $2 \leq k \leq n$ ) to re-create the original block, thereby providing “increased fault tolerance.” EX1004, 3:28–39; 5:49–6:14; 6:53–67. EX1002, ¶214.

**Fourth**, both Dickinson and Hardjono address substantially the same technical challenge—secure, distributed data storage—and provide similar

functional and structural solutions. Therefore, a POSITA would have readily seen that Hardjono's return of reassembled data fits naturally into Dickinson's storage system. Dickinson teaches a cryptographic system in which one or more secure servers or a trust engine store cryptographic keys and user authentication data. EX1003, 9:7–12. Dickinson further describes that trust engine comprises “secure servers for accessing and storing sensitive information” (EX1003, 10:20–32), “employs a data-splitting module 520 to randomize and split enrollment authentication data into portions” (EX1003, 18:20–19:2, 20:1–14), and uses an authentication engine that separates and compares current authentication inputs against “the appropriate subset of enrollment data” (EX1003, 38:16–21). EX1002, ¶¶215–219.

Similarly, Hardjono discloses servers that are used for storing and securing data and keys. Particularly, Hardjono teaches generating  $n$  shares from a data block, distributing those shares to multiple databases, and requires only some of them to reconstruct the original block—thereby enhancing security and fault tolerance. EX1004, Abstract, 3:28–39, 5:49–6:14, 6:53–67. Upon a data request, the server retrieves the shares, re-creates the block, and returns the re-created data to the requester. EX1004, 7:65–8:2. A POSITA would have understood that combining Hardjono's explicit return-of-assembled-data step with Dickinson's system—in which, upon receipt of a data-retrieval request, the assembled segments are

transmitted back to the requester—is an obvious modification requiring no more than routine implementation. EX1002, ¶220.

*Fifth*, incorporating Hardjono’s return-of-assembled-data step into Dickinson’s system would have required only routine software development, since Dickinson’s trust engine components are all implemented as software. EX1003, 18:20–21, 18:23–24, 19:3–5. A person of ordinary skill could readily write the necessary code to invoke Hardjono’s taught function of sending the reconstructed data to the requester. *See Keynetik*, 2023 WL 2003932, at \*2. EX1002, ¶221.

Such a modification would have been straightforward and would not have demanded extensive experimentation or imposed an undue burden, because both Dickinson and Hardjono already disclose secure, software-based distributed storage architectures. EX1002, ¶222.

For the same reasons, the results of this integration would have been predictable to a POSITA, because returning the assembled data upon request is a well-established practice in distributed storage systems. EX1002, ¶223.

*Finally*, a POSITA would have had a reasonable expectation of success in making this change, since Hardjono’s teaching merely provides implementation detail for functionality—namely, transmitting assembled data—that directly aligns with the objectives and design of Dickinson’s trust engine. EX1002, ¶224.

**iii. Encrypting Data Portions with Separate Keys in View of Hardjono**

Dickinson describes a trust engine that enforces data usage policies by restricting access to individual data segments and encrypting those segments. EX1003, 3:13–25. Hardjono teaches encrypting each piece of data independently—using the same or different keys. EX1004, 1:30–41. Hardjono further teaches that encrypting each data piece with independent keys enhances security, because an attacker would need to compromise multiple keys to gain access to the entire dataset. *Id.* Hardjono additionally teaches that encryption may be applied at one or more stages of the document-storage process using any conventional scheme, including the RSA encryption algorithm, which employs one or more keys to secure the data. EX1004, 6:24–33. EX1002, ¶¶225–228.

It would have been obvious to combine Hardjono’s teachings with Dickinson’s trust engine so that, rather than using a single key for all data shares, each segment is encrypted with its own key. Hardjono discloses that encrypting each piece with separate keys enhances overall security by requiring compromise of multiple keys to access the full dataset. EX1002, ¶¶229–230.

**a. Reasons to Combine Dickinson with Hardjono’s Teaching of Encrypting Data Portions with Separate Keys**

A POSITA would have been motivated to modify Dickinson’s trust engine so that each data segment is encrypted with its own key in view of Hardjono’s teachings, for at least two reasons. *First*, Hardjono explicitly teaches that using “different encryption keys” for individual pieces “provides an additional level of security because possibly multiple keys must be compromised in order to access the entire data.” EX1004, 1:30–41. A POSITA would have been motivated to apply this technique to each portion of a dataset in Dickinson’s system because this would directly enhance overall security by limiting the impact of a single key compromise. EX1002, ¶¶231–232.

*Second*, a POSITA would have recognized that integrating Hardjono’s per-segment keying simply involves adopting a known method—namely, encrypting each fragment with a different key—within Dickinson’s storage system. Because this approach targets functionality already present in Dickinson’s trust engine, its implementation would have been routine and would predictably improve confidentiality without imposing undue effort. EX1002, ¶233.

A POSITA would have had a reasonable expectation of successfully implementing Hardjono’s per-fragment, multi-key encryption within Dickinson’s system for at least three reasons. EX1002, ¶234.

**First**, both references already disclose the use of multiple encryption keys. Dickinson teaches “associating a user with one or more keys drawn from a plurality of private cryptographic keys stored on a secure server” (EX1003, 3:13–25), and Hardjono describes encrypting each piece separately using same or different keys (EX1004, 1:30–41). Because Dickinson already uses multiple keys, applying Hardjono’s approach of encrypting individual data shares with separate keys would have been a simple modification that would enhance security of Dickinson’s system. EX1002, ¶235.

**Second**, implementing Hardjono’s technique in Dickinson’s system would have merely required performing the encryption step after the data set is split. This combination of the two references would predictably enhance confidentiality by ensuring that an attacker must compromise multiple keys to recover the entire dataset. EX1004, 1:30–41. EX1002, ¶236.

**Third**, Dickinson’s trust engine modules are implemented in software. A POSITA could readily write the necessary code to generate and apply separate keys to each fragment, a routine programming task well within ordinary skill. *See Keynetik*, 2023 WL 2003932, at \*2. EX1002, ¶237.

Therefore, modifying Dickinson to encrypt each data share with a different encryption key would have been simple, requiring no more than routine software

development without undue experimentation, yielding predictable security benefits, and aligning directly with the goals of Dickinson’s system. EX1002, ¶238.

**4. Claim 1**

- i. [1Pre] A method for securely storing and retrieving data, the method comprising:**

Dickinson discloses [1Pre]. EX1002, ¶¶240, 242.

Dickinson discloses “[a] *method for securely storing and retrieving data.*”<sup>2</sup>

For example, Dickinson discloses “provid[ing] a cryptographic system where one or more *secure* servers, or a trust engine, *stores cryptographic keys and user authentication data*” (EX1003, 9:7–12) where “the trust engine 110 comprises one or more *secure* servers for accessing and *storing sensitive information*” (EX1003, 10:20–32). *See also* EX1003, 38:13 (“assembling the *enrollment data* 410 *retrieved* from the depository 210.”). EX1002, ¶241.

- ii. [1A] generating, using an electronic computing system that includes processing circuitry, a plurality of shares by performing a cryptographic operation on a data set and distributing the data set in the plurality of shares such that the data set can be reconstructed using any subset of the shares that includes at least a minimum number less than all of shares;**

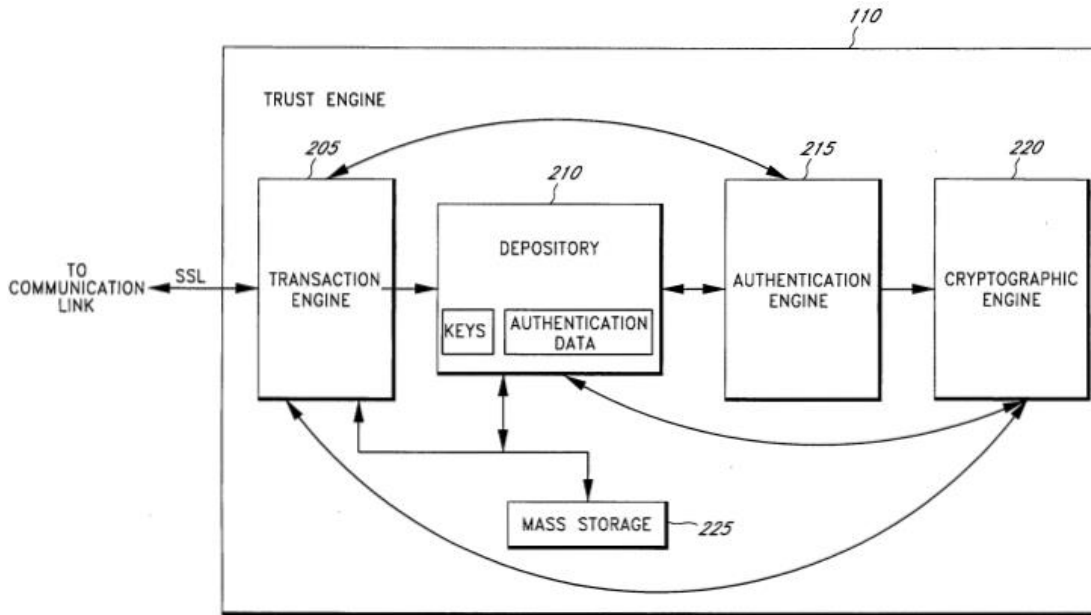
Dickinson discloses [1A]. EX1002, ¶¶243, 255, 260.

---

<sup>2</sup> In this Petition, the quoted claim language is identified using *unbolded italics* and emphasized, quoted language from the reference is identified using *bolded italics*.

- a. [1A-1] generating, using an electronic computing system that includes processing circuitry, a plurality of shares by performing a cryptographic operation on a data set and

*First*, Dickinson discloses “using an electronic computing system that includes processing circuitry[.]” For example, Dickinson discloses a “*trust engine 110*,” which includes a transaction engine 205, a depository 210, an authentication engine 215, a cryptographic engine 220 and a mass storage 225 and “provides the user with complete cryptographic functionality.” EX1003, 11:5–6; *see also* EX1003, 13:5–7 (describing FIGURE 2 as illustrating “a block diagram of the trust engine 110 of FIGURE 1”). Dickinson explains that these components can communicate in various ways that would have included *processing circuitry*, including by transmitting XML documents to IP addresses (EX1003, 13:10–17) and routing data using conventional HTTP routing techniques like URLs or URIs (EX1003, 13:29–30). Indeed, Dickinson teaches that the modules inside the trust engine 110 are implemented in “*software, hardware, or combination.*” EX1003, 18:20–28. A POSITA would have understood that the trust engine is an example of an electronic computing system.



**FIG. 2**

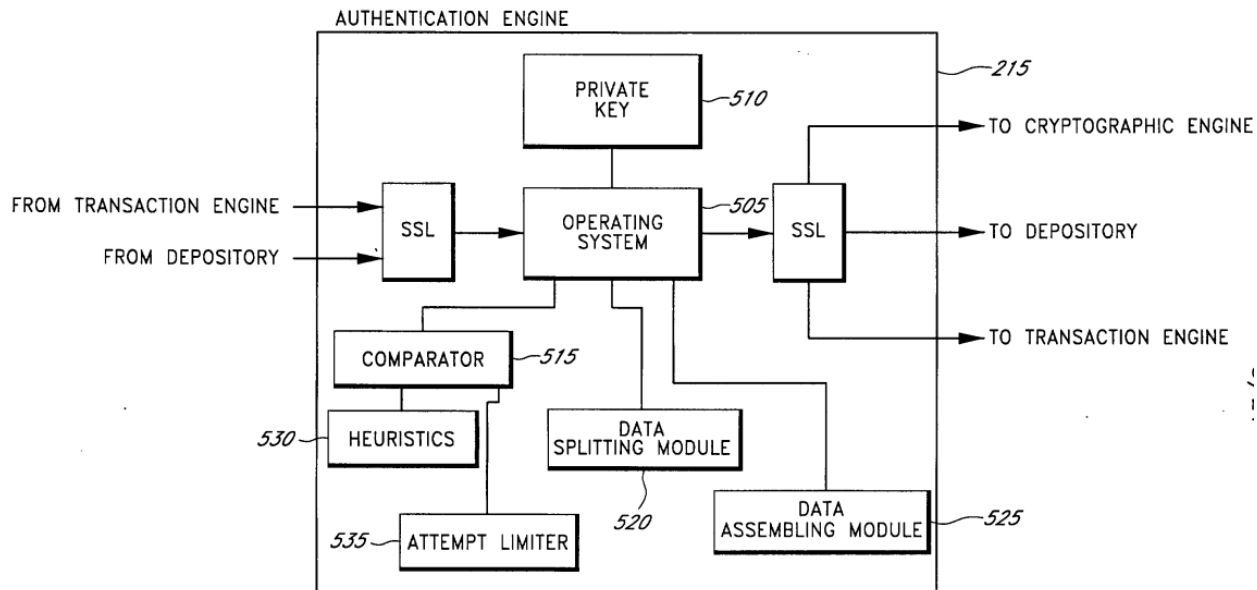
EX1003, FIG. 2. EX1002, ¶¶244–246.

*Second*, “generating ... a *plurality of shares* by performing a *cryptographic operation* on a *data set*.” For example, Dickinson states that “the authentication engine 215 and the *cryptographic* engine 220,” which are part of the *trust engine* 110, “may advantageously employ their respective data splitting modules to *divide sensitive data into undecipherable portions*, and then transmit one or more *undecipherable portions* of the sensitive data to a particular data storage facility.”

EX1003, 19:31–34; *see also* EX1003, 18:20–28, 20:20–21:9. EX1002, ¶247.

Dickinson describes various cryptographic operations for performing such splitting operation. For example, Dickinson Figure 5 illustrates that “[t]he data splitting module 520 advantageously comprises a software, hardware, or

combination module having the ability to **mathematically operate on various data** so as to ... **split the data** into **portions**.” EX1003, 18:20–28.



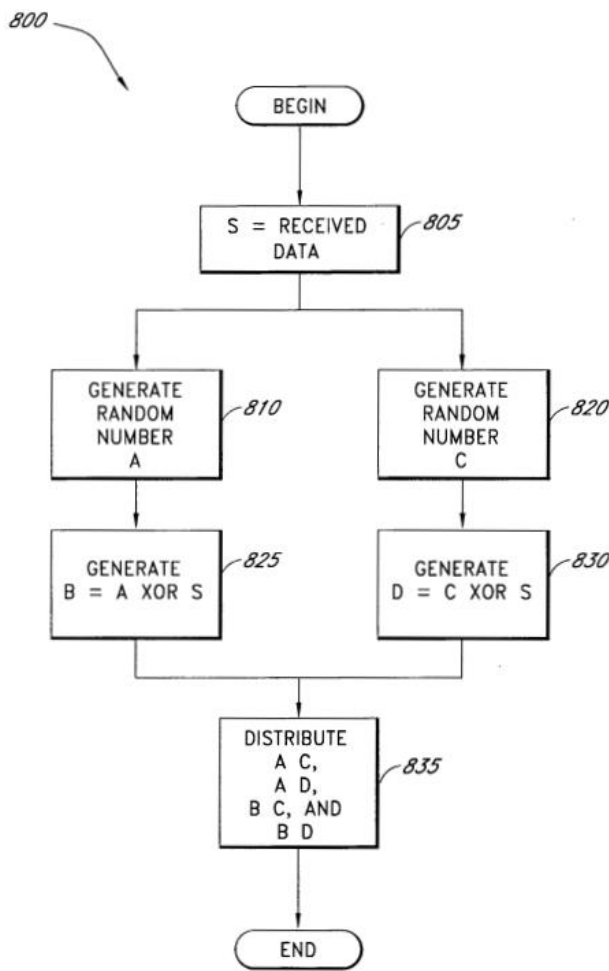
**FIG. 5**

EX1003, FIG. 5. EX1002, ¶248.

In another example, Dickinson explains that “the authentication engine 215 and the *cryptographic* engine 220 may advantageously employ their respective data splitting modules [520 and 610, respectively] to **divide sensitive data** into **undecipherable portions**.” EX1003, 19:27–34. Dickinson Figure 8 “illustrates a flowchart of a data splitting process 800 performed by the data splitting module” and shows that “**sensitive data ‘S’** is received by the data splitting module of ... the cryptographic engine 220” and “the data splitting module then generates **a**

*substantially random number value, or string or set of bits, 'A'.*” EX1003, 20:20–29, FIG. 8. EX1002, ¶249.

Furthermore, “the trust engine 110, may advantageously perform ... only some or all of the *cryptographic functions, such as data encryption and decryption*” as it generates the plurality of shares. EX1003, 16:6–14.



**FIG. 8**

EX1003, FIG. 8. EX1002, ¶250.

Dickinson also teaches that the “*sensitive data*”—to which the operations are performed—is a “*data set*.” Dickinson explains that in one example, “sensitive data S includes enrollment *authentication data*.” EX1003, 21:34–35; *see also* EX1003, 20:20–29. It was well known that the enrollment *authentication data* includes a set of information (e.g., username, password). *See* EX1019, [0012] (“The first web site authenticates the user using one authentication method, for example the *username and password*.”); EX1020, [0005] (“The connections shown provide a path for user client 12 to provide the user’s FI [Financial Institution] username and password to portal server 16, a path for portal server 16 to read and write user *authentication data* (such as *username, password, associate FI, etc.*)”). Dickinson also explains that the “communications ... to the data storage facilities D1 through D4 may include the transmission of sensitive data *to be stored*.” EX1003, 19:29–31; *see also* EX1003, 20:4–6 (“*By distributing the sensitive data into distinct and independent storage facilities D1 through D4, ...*”). Because Dickinson’s “sensitive data” such as “authentication data” includes a “set” of data items to be stored, Dickinson’s “sensitive data” meets the claimed “*data set*” limitation. EX1002, ¶¶251–253.

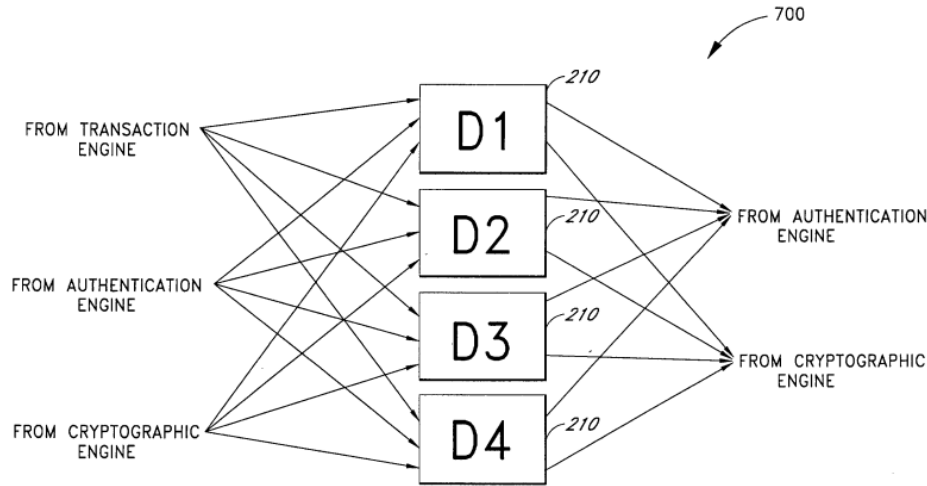
Indeed, in Patent Owner’s litigation against Google, Patent Owner broadly construed “*data set*” to mean “a collection of information for storage.” EX1021, 8. The “*sensitive data*” in Dickinson satisfies Patent Owner’s construction of “*data*

*set*” because as explained above, the sensitive data, such as “enrollment authentication data” is a collection of information that is stored by Dickinson’s system. EX1002, ¶254.

**b. [1A-2] distributing the data set in the plurality of shares such that the data set can be reconstructed using any subset of the shares that includes at least a minimum number less than all of shares;**

*First*, Dickinson discloses “*distributing the data set in the plurality of shares.*”

For example, Dickinson states that “[b]y *distributing* the *sensitive data* into distinct and independent storage facilities D1 through D4, some or all of which may be advantageously geographically separated, the depository system 700 provides redundancy along with additional security measures.” EX1003, 20:1–14. Dickinson discloses that “the data splitting process 800 advantageously places *portions* of the *sensitive data* in each of the four data storage facilities D1 through D4.” EX1003, 21:10–14.



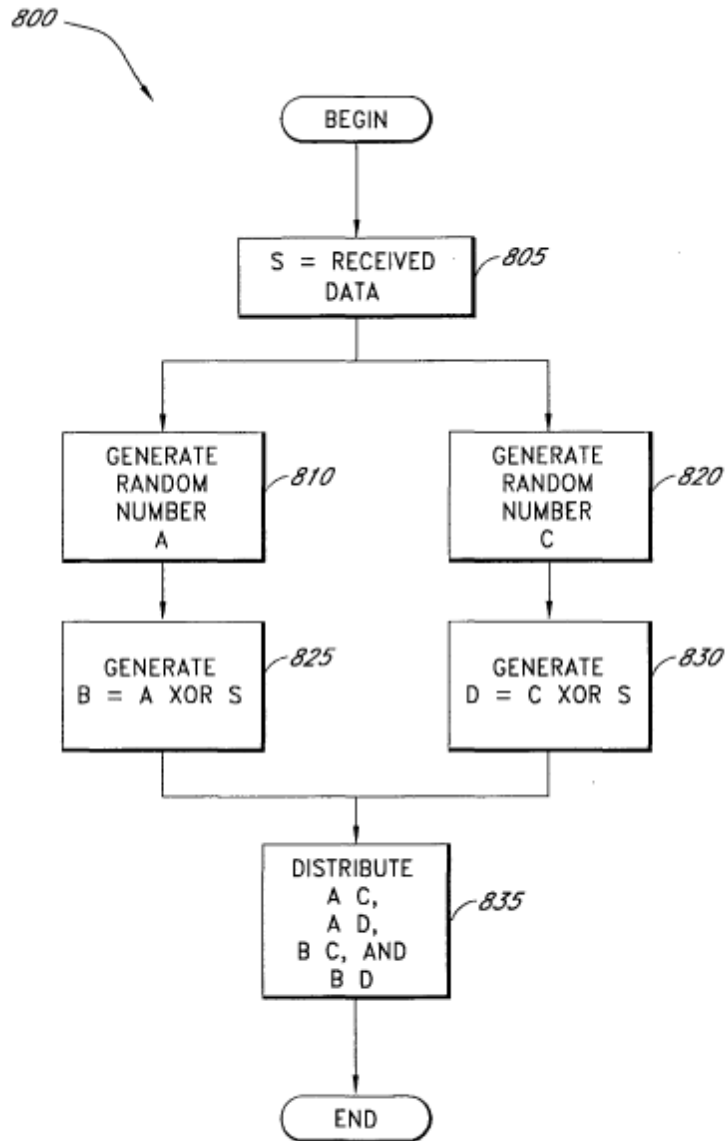
**FIG. 7**

EX1003, FIG. 7. EX1002, ¶256.

*Second*, Dickinson discloses that “the data set can be reconstructed using any subset of the shares that includes at least a minimum number less than all of shares.”

For example, Dickinson discloses that “the data splitting module then generates a substantially random number, value, or string or set of bits, ‘A,’” and “another statistically random number ‘C.’” EX1003, 20:20–21:9. Dickinson also discloses that “[t]he data splitting module then combines the numbers A and C with the *sensitive data S* such that new numbers ‘B’ and ‘D’ are generated.” EX1003, 20:30–21:9; *see also* EX1003, 20:33–34. And “the random numbers A and C and the numbers B and D are paired such that none of the pairings contain sufficient data, by themselves, to reorganize and decipher the original *sensitive data S*,” and “the numbers may be paired as follows: AC, AD, BC, and BD,” so that “the pairings of

AC, AD, BC, and BD, were distributed such that **any two** provide one of A and B, or, C and D.” EX1003, 20:30–21:9, 21:21–32; *see also* EX1003, FIG. 8. Finally, Dickinson discloses that “only data from **two** of the multiple data storage facilities, D1 through D4, are **needed** to decipher and reassemble the **sensitive data**.” EX1003, 20:1–14. That is, there are 4 total shares of data, and any **two** (less than four) shares are needed to reassemble the data.



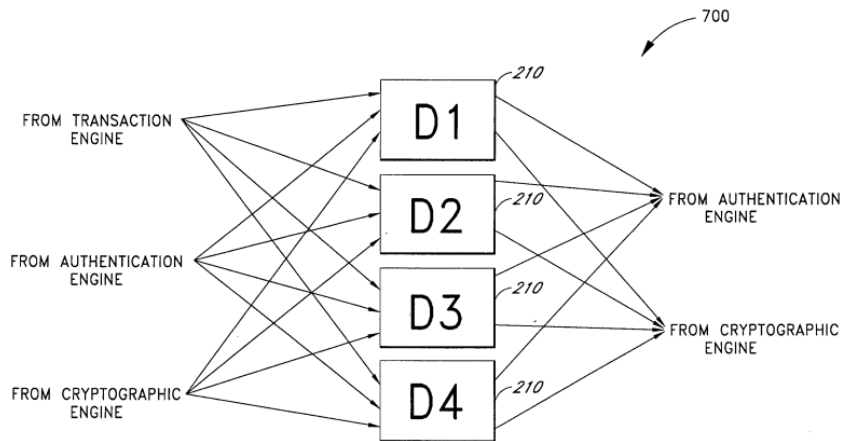
**FIG. 8**

EX1003, FIG. 8. EX1002, ¶¶257–259.

- iii. [1B] storing the **plurality of shares** at a plurality of storage devices;

Dickinson discloses [1B]. EX1002, ¶¶261, 265.

For example, Dickinson discloses that “the data splitting process 800 advantageously *places portions* of the sensitive data in each of the four *data storage facilities* D1 through D4.” EX1003, 21:10–14.



**FIG. 7**

EX1003, FIG 7. EX1002, ¶262.

Indeed, Dickinson states that “the authentication engine 215 and the cryptographic engine 220 may advantageously employ their respective data splitting modules to ... then transmit one or more undecipherable *portions* of the sensitive data *to a particular data storage facility*.” EX1003, 19:27–34. EX1002, ¶263.

Dickinson also teaches that “each *data storage facility, D1 through D4*, comprises a *separate and independent storage system*, such as, for example, a directory server.” EX1003, 20:1–14. EX1002, ¶264.

- iv. [1C] receiving, at the **electronic computing system**, request to retrieve the data set;

Dickinson discloses [1C]. EX1002, ¶¶266, 268.

For example, Dickinson discloses that “the vendor system 120 ... forwards ... the authentication *request* to the *trust engine 110*[.]” EX1003, 28:6–14. As discussed in IX.A.4.ii.a above [Ground I, 1A-1], Dickinson teaches that the modules inside the trust engine 110 are implemented in “software, hardware, or combination,” (EX1003, 18:20–28), and thus, the trust engine is an electronic computing system. Dickinson also discloses that “the *trust engine 110* includes a transaction engine 205.” EX1003, 13:5–17. Dickinson also teaches that “transaction engine 205 receives the [*vendor request*] ... and generates a *request* for the user’s enrollment authentication data to be assembled from the *data storage facilities D1 through D4*.” EX1003, 28:6–14. Dickinson discloses that “the *sensitive data S* includes enrollment authentication data.” EX1003, 21:33–22:2. EX1002, ¶¶244–246, 267.

- v. **[1D] identifying from the plurality of storage devices a set of fastest-responding storage devices necessary to retrieve the minimum number of shares, wherein the set of fastest-responding storage devices are identified based at least in part on the response time of the storage devices;**

Dickinson alone, or Dickinson in view of Hardjono, teaches [1D]. EX1002, ¶¶269, 276–277.

*First*, Dickinson discloses [1D]. Dickinson discloses that “the transaction engine 205 may broadcast requests to particular *data storage facilities* based on a

wide number of criteria, such as, for example, *response time*, server loads, maintenance schedules, or the like.” EX1003, 19:24–26. EX1002, ¶271.

This disclosure in Dickinson is the same disclosure as the supposed written description support for this limitation in the ’194 Patent. *See* EX1013 (highlighted version of the ’194 Patent’s specification showing where the highlighted portion appears in Dickinson), 18:27–31. A POSITA would have understood that to “broadcast requests to particular data storage facilities,” the transaction engine 205 would have first identified the “particular data storage facilities” from all data storage facilities. Furthermore, a POSITA would have also understood that identifying the “particular data storage facilities” “based on ... response time” would have included identifying the “*set of fastest-responding storage devices.*” EX1002, ¶272.

*Second*, alternatively, Dickinson in view of Hardjono also teaches [1D]. *See* Section IX.A.3 [Dickinson/Hardjono]. As explained in Section IX.A.3, Hardjono discloses “*identify[ing] databases* where *k shares* of the block are stored” including “*based on* determining which of the distributed *databases* are currently, most easily, or *most quickly accessible.*” EX1004, 7:57–64. A POSITA would have understood that the “most quickly accessible” database to refer to the response time of the storage devices. EX1002, ¶¶ 273, 275.

It would have been obvious to combine Dickinson with Hardjono such that Dickinson's trust engine identifies, from its storage facilities D1-D4, "*the set of fastest-responding storage devices ... based at least in part on the response time of the storage devices*" as discussed in Section IX.A.3 [Dickinson/Hardjono]. EX1002, ¶274.

**vi. [1E] retrieving from the set of fastest-responding storage devices, the minimum number of shares;**

Dickinson, or alternatively, Dickinson in view of Hardjono, teaches [1E]. EX1002, ¶¶278, 285–286.

*First*, Dickinson discloses [1E]. As explained in Section IX.A.4.ii.b [Ground I, 1A-2], Dickinson discloses "*receiv[ing]* data *portions* from at least the first *two* of the *data storage facilities D1 through D4*[" EX1003, 21:21–32. Dickinson discloses "the data assembling module 525 may *assemble* the *sensitive data S*, when, for example, it receives data *portions* from at least the first *two* of the *data storage facilities D1 through D4* to respond to an assemble request by the trust engine 110." EX1003, 21:21–32. EX1002, ¶¶279-280.

And as explained in Section IX.A.4.v [Ground I, 1D], Dickinson discloses that "the transaction engine 205 may broadcast requests to particular *data storage facilities* based on a wide number of criteria, such as, for example, *response time*,

server loads, maintenance schedules, or the like.” EX1003, 19:24–26; *see also* EX1001, 18:27–31 (identical text). EX1002, ¶281.

*Second*, Dickinson in view of Hardjono also teaches [1E]. EX1002, ¶¶282–283.

As explained above, Dickinson discloses “*receiv[ing]* data *portions* from at least the first *two* of the *data storage facilities D1 through D4*[.]” EX1003, 21:21–32. Hardjono also discloses “identify[ing] databases where k shares of the block are stored” which “may be based on determining which of the *distributed databases* are currently, most easily, or *most quickly accessible*.” EX1004, 7:57–64. A POSITA would have understood that the “most quickly accessible” database to refer to the response time of the storage devices. Hardjono also discloses that “[o]nce the databases where k shares are located are identified, server 110 *retrieves the k shares* for the block of data step 425[.]” EX1004, 7:65–8:2. It would have been obvious for a POSITA to take Dickinson’s teachings of choosing *data storage facilities* depending on their *response time* and combine that teaching with Hardjono’s choosing of *distributed databases* depending on which is *most quickly accessible* to determine the “*fastest-responding storage devices*.” EX1002, ¶284.

- vii. [1F] reconstructing the data set using the minimum number of *shares*; and

Dickinson discloses [1F]. EX1002, ¶¶287, 289.

As explained in Section IX.A.4.vi [Ground I, 1E], Dickinson describes “*receiv[ing]* data *portions* from at least the first *two* of the *data storage facilities D1 through D4*[.]” EX1003, 21:21–32. Then, Dickinson discloses “the data assembling module 525 may *assemble* the *sensitive data S*, when, for example, it receives data *portions* from at least the first *two* of the *data storage facilities D1 through D4* to respond to an assemble request by the trust engine 110.” EX1003, 21:21–32. EX1002, ¶288.

**viii. [1G] sending the data set responsive to the request.**

Dickinson in view of Hardjono teaches [1G]. EX1002, ¶¶290, 292.

As explained in Section IX.A.2 [Hardjono], Hardjono discloses that, “[u]pon subsequent receipt of a *request for the data*,” “[t]he *re-constructed data* is then *returned* to the requester.” EX1004, 3:40–50; *see also* EX1004, 7:65–8:2 (“Server 110 then re-creates the block from the *k shares*, step 430, and *returns* the *re-created data block* to the *requester*, step 435.”). EX1002, ¶291.

It would have been obvious to combine Dickinson with Hardjono so that Dickinson’s trust engine returns the assembled data set to the requester, as taught by Hardjono as discussed in Section IX.A.3 [Dickinson/Hardjono]. EX1002, ¶290.

**5. Claim 2**

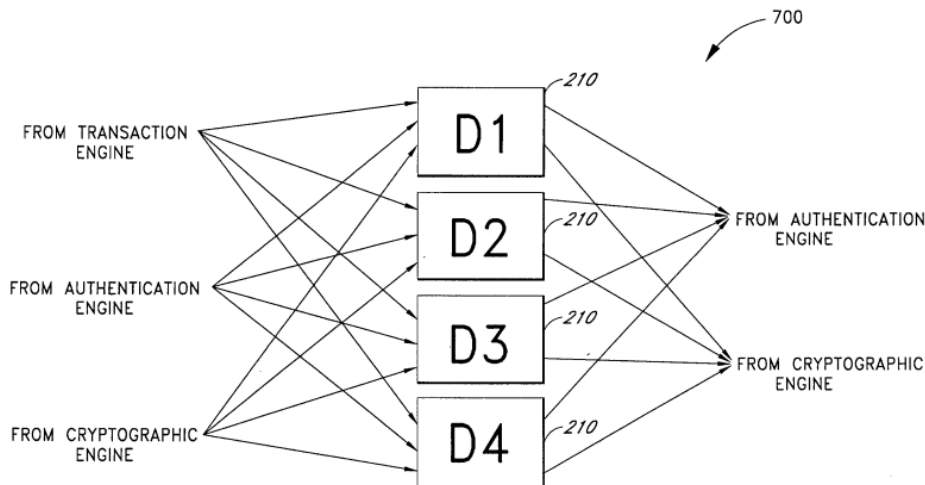
**i. [2] The method of claim 1, wherein storing the shares comprises:**

*See* claim 1. EX1002, ¶293.

- i. [2A] storing a first subset of the shares at a first subset of the plurality of storage devices that is physically located at a first data center; and

Dickinson discloses [2A]. EX1002, ¶¶294, 302.

Dickinson discloses that “the depository system 700” depicted in FIGURE 7 “comprises multiple data storage facilities, for example, data storage facilities D1, D2, D3, and D4,” which “may advantageously comprise some or all of the elements disclosed with reference to the depository 210 of FIGURE 4,” and can “comprise[] multiple *geographically separated independent data storage systems.*” EX1003, 19:17–20, 20:1–14.



**FIG. 7**

EX1003, FIG. 7. EX1002, ¶295.

As explained above in Section IX.A.4.iii [Ground I, 1B], Dickinson discloses that “the data splitting process 800 advantageously places *portions of the sensitive data in each of the four data storage facilities D1 through D4.*” EX1003, 21:10–

14. For example, Dickinson discloses that “each of [AC, AD, BC, and BD] is distributed to one of the depositories D1 through D4.” EX1003, 20:30–21:9. Dickinson describes these “*data storage facilities*” as comprising, for example, “a directory server, a database server, or the like,” “one or more lightweight directory access protocol (LDAP) servers,” and/or “distinct and physically separated data storage facilities, as disclosed further with reference to FIGURE 7.” EX1003, 14:17–19, 16:25–17:5, 19:16–26. EX1002, ¶¶296–297.

This disclosure in Dickinson is the same disclosure as the supposed written description support for this limitation in the ’194 Patent. *See* EX1013 (providing a highlighted version of the ’194 Patent’s specification where the highlighted portion appears verbatim in Dickinson), 13:20–23, 15:28–54, 18:27–31. EX1002, ¶298.

A POSITA would have understood that each of D1 through D4 may be a geographically separated data center, each containing multiple data storage devices because it was well known that data storage devices are located in different data centers. EX1018 (Moulton) at [0036] (“Internetwork 201 enables the interconnection of a heterogeneous set of computing devices and mechanisms ranging from a supercomputer or data center 301 to a hand-held or pen-based device 306.”); Claim 19 (“The method of claim 14 wherein the selected storage nodes comprise at least two storage nodes and the at least two storage nodes are located in *different data centers*.”). EX1002, ¶299.

Thus, a POSITA would have understood that, for example, Dickinson’s “AC” (“*a first subset of the shares*”) would be stored at Dickinson’s one or more geographically-separate data facilities D1-D4 (“*a first subset of the plurality of storage devices located at a first data center*”). EX1003, 14:17–19, 16:25–17:5, 19:16–26, 20:30–21:9, 20:1–14. EX1002, ¶¶300–301.

- ii. **[2B] storing a second subset of the shares at a second subset of the plurality of storage devices that is physically located at a second data center, the first data center being geographically separated from the second data center.**

Dickinson discloses [2B]. EX1002, ¶¶303, 308.

*First*, Dickinson discloses “*storing a second subset of the shares at a second subset of the plurality of storage devices that is physically located at a second data center.*” As explained in Section IX.A.5.i [Ground I, 2A], Dickinson discloses that the depository system comprises multiple data storage facilities D1, D2, D3, and D4, and can comprise multiple geographically separated independent data storage systems. EX1003, 19:17–20, 20:1–14, FIG. 7. And as explained in Section IX.A.4.iii [Ground I, 1B], Dickinson discloses that each of AC, AD, BC, and BD is distributed to one of the depositories D1 through D4. EX1003, 14:17–19, 16:25–17:5, 19:16–26, 20:30–21:9. Thus, a POSITA would have understood that, for example, Dickinson’s “BC” (“*a second subset of the shares*”) would be stored at another one of Dickinson’s geographically-separate data facilities D1-D4 (“*a*

*second subset of the plurality of storage devices located at a second data center*”).

EX1003, 16:25–17:5, 19:16–26, 20:30–21:9. EX1002, ¶¶304–06.

*Second*, Dickinson discloses that “*the first data center [is] geographically separated from the second data center.*” As explained above in this section, Dickinson teaches “distributing the sensitive data into distinct and *independent storage facilities D1 through D4*, some or all of which may be advantageously *geographically separated.*” EX1003, 20:1–14. EX1002, ¶307.

## 6. Claim 3

- i. **[3] The method of claim 1, further comprising establishing secure connections between the electronic computing system and the plurality of storage devices.**

Dickinson teaches the additional limitation of [3]. EX1002, ¶¶309, 312.

As discussed in IX.A.4.ii.a above, Dickinson teaches that the modules inside the trust engine 110 are implemented in “software, hardware, or combination” (EX1003, 18:20–28), and thus, the trust engine is an electronic computing system. Dickinson also teaches that the “*the data storage facilities D1 through D4* communicate with the *transaction engine 205, the authentication engine 215, and the cryptographic engine 220*, preferably through conventional *SSL communication* links transferring, for example, XML documents.” EX1003, 19:16–26. Dickinson also discloses that “the communications to the authentication engine *comprise secure communications*, such as, for example, *SSL technology.*”

EX1003, 15:13–19. Because the *SSL communications* are between the data storage facilities D1 through D4 and components of the *trust engine 110*, a POSITA would understand that Dickinson teaches “*establishing secure connections between the electronic computing system and the plurality of storage devices.*” EX1002, ¶¶244–246, 310–311.

7. **Claim 4**

- i. [4] The method of claim 1, wherein the **plurality of data blocks share**<sup>3</sup> contain a substantially random distribution of the data set.

Dickinson discloses the additional limitation of [4]. EX1002, ¶¶313, 318.

Dickinson discloses “the depository 210 may advantageously comprises a *plurality of secure data storage facilities*” “configured such that a compromise of the security in one individual *data storage facility* will not compromise ... the authentication data stored therein.” EX1003, 14:30–15:3. And “the authentication data are *mathematically operated* on so as to statistically and *substantially randomize* the *data stored in each data storage facility.*” EX1003, 14:30–15:3. Dickinson also discloses that “the *data stored in each data storage facility* is

---

<sup>3</sup> The term “*plurality of data blocks shares*” lacks antecedent basis. For purposes of this IPR only, Petitioner interprets “*plurality of data blocks shares*” to mean “*plurality of shares.*” EX1002, ¶314.

*randomized* and undecipherable.” EX1003, 20:1–14; *see also* EX1003, 18:20–28 (“The data splitting module 520 advantageously comprises a software, hardware, or combination module having the ability to *mathematically operate* on various data so as to *substantially randomize* and split the data into *portions*.”); EX1003, 21:3–6 (“[T]he numbers may be paired as follows: AC, AD, BC, and BD,” and “each of the foregoing pairings is *randomly distributed* to one of the *depositories D1 through D4*.”). EX1002, ¶¶315–17.

**8. Claim 5**

- i. [5] The method of claim 1, wherein the data set can be reconstructed from shares received from at least two storage devices**

Dickinson discloses the additional limitation of [5]. EX1002, ¶¶319, 321.

Dickinson teaches “*wherein the data set can be reconstructed from shares received from at least two storage devices*.” As explained in Section IX.A.4.vi [Ground I, 1E], Dickinson describes “receiv[ing] data *portions* from at least the *first two* of the *data storage facilities D1 through D4*[.]” EX1003, 21:21–32. EX1002, ¶320.

**9. Claim 6**

- i. **[6] The method of claim 1, wherein the shares are encrypted with a corresponding number of different keys.**

Dickinson in view of Hardjono teaches the additional limitation of [6].  
EX1002, ¶¶322, 324.

As explained in Section IX.A.2 [Hardjono], Hardjono teaches “*encrypt[ing]* each *piece* separately using ... *different encryption keys*.” EX1004, 1:30–41. Hardjono discloses that “[t]his solution provides an additional level of security because possibly *multiple keys* must be compromised in order to access the entire data.” EX1004, 1:30–41. Hardjono also teaches using that “[a]ny of a wide variety of conventional encryption processes can be used by the present invention, ... includ[ing] the RSA encryption scheme which *employs one or more encryption keys* to encrypt the data.” EX1004, 6:24–33. EX1002, ¶323.

It would have been obvious to combine Dickinson with Hardjono such that Dickinson’s data portions are encrypted using their own keys as discussed in Section IX.A.3 [Dickinson/Hardjono]. EX1002, ¶322.

**10. Claim 7**

Claim 7 only varies from claim 1 for the preamble [7Pre]; and limitations [7A], [7B], [7B-1], [7B-3], and [7B-5]. Accordingly, the Dickson/Hardjono

System discloses [7B-2], [7B-4], and [7B-6] for the same reasons provided above for [1B], [1D], [1E], and [1G]. EX1002, ¶¶325–326.

| Claim 1     | Claim 7 |
|-------------|---------|
| [1B]        | [7B-2]  |
| [1D] + [1E] | [7B-4]  |
| [1G]        | [7B-6]  |

The Dickinson/Hardjono System also discloses [7Pre], [7A], [7B], [7B-1], [7B-3], and [7B-5] for the reasons provided below. EX1002, ¶327.

**i. [7Pre] An electronic computing system for securely storing and retrieving data, the electronic computing system comprising:**

Dickinson discloses [7Pre]. EX1002, ¶¶328, 330.

As explained in Section IX.A.4.i [Ground I, 1Pre], Dickinson discloses a method for “*securely storing and retrieving data.*” Also as explained in Section IX.A.4.ii, Dickinson discloses “*an electronic computing system*” for the method for “*securely storing and retrieving data.*” EX1002, ¶329.

**ii. [7A] a processing unit; and**

Dickinson discloses [7A]. EX1002, ¶¶331, 333.

As explained in Section IX.A.4.v [Ground I, 1Pre], Dickinson’s trust engine modules are implemented in software, hardware, or a combination. *See, e.g.*, EX1003, 18:20–21, 18:23–24, 19:3–5. A POSITA would have understood that those implementations include a *processing unit* because they process data to perform the

storage method. *See* Sections IX.A.10.iv-IX.A.10.ix [Ground I, 7B-1 – 7B-6].  
EX1002, ¶332.

**iii. [7B] a system memory comprising instructions that, when executed by the processing unit, cause the processing unit to:**

Dickinson discloses [7B]. EX1002, ¶¶334, 337.

Dickinson discloses that trust engine includes a “depository 210 [that] comprises a single *logical memory structure* indexing authentication data and cryptographic key data to a unique user ID.” EX1003, 16:29–31. In addition, as explained in Section IX.A.4.v [Ground I, 1D], Dickinson’s trust engine modules are implemented in software, hardware, or a combination. *See, e.g.*, EX1003, 18:20–21, 18:23–24, 19:3–5. A POSITA would have understood that those implementations include a system memory comprising instructions that, when executed by the processing unit, cause the processing unit to perform the steps of [7B-1] to [7B-6] because they process data to perform the storage method. *See* EX1004, 8:63–65; Sections IX.A.10.iv-IX.A.10.ix [Ground I, 7B-1 – 7B-6]. EX1002, ¶¶334–336.

- iv. [7B-1] generate a plurality of shares by performing a cryptographic operation on a data set and distributing the data set in the plurality of shares such that the data set can be reconstructed using any subset of the shares that includes at least a minimum number less than all of the shares and such that the data set cannot be reconstructed using any subset of the shares that includes fewer than the minimum number of the shares;

Dickinson discloses [7B-1]. EX1002, ¶¶338, 341.

Dickinson discloses “generate a plurality of shares by performing a *cryptographic operation* on a data set and distributing the data set in *the plurality of shares* such that the data set can be reconstructed using any subset of *the shares* that includes at least a minimum number less than all of the shares” for the reasons provided in Section IX.A.4.ii [Ground I, 1A]. EX1002, ¶339.

Dickinson also teaches “and such that the data set cannot be reconstructed using any subset of *the shares* that includes fewer than the minimum number of *the shares*” for the reasons provided in Section IX.A.4.ii [Ground I, 1A]. A POSITA would have understood that reconstructing using the “*minimum number of shares*,” by definition, that the data set cannot be reconstructed by any number of *pieces* less than the *minimum number*. EX1002, ¶340.

- v. **[7B-2] store the plurality of shares at a plurality of storage devices;**

Dickinson discloses [7B-2] for the reasons provided in Section IX.A.4.iii [Ground I, 1B]. EX1002, ¶342.

- vi. **[7B-3] receive, via the primary interface<sup>4</sup>, a request to retrieve the data set;**

Dickinson discloses [7B-3]. EX1002, ¶¶343, 348.

*First*, Dickinson discloses “receive ... a request to retrieve the data set” for the reasons in Section IX.A.4.iv [Ground I, 1C]. For example, Dickinson discloses that “the vendor system 120 ... forwards ... the authentication *request* to the *trust engine 110*[.]” EX1003, 28:6–14. Dickinson also discloses that “the *trust engine 110* includes a transaction engine 205.” EX1003, 13:5–17. Dickinson also teaches that “transaction engine 205 receives the [*vendor request*] ... and generates a *request* for the user’s enrollment authentication data to be assembled from the *data storage facilities D1 through D4*.” EX1003, 28:6–14. Dickinson discloses that “the

---

<sup>4</sup> The term “primary interface” is indefinite. But the prior art discloses at least the embodiment provided in the ’194 Patent, and thus, the indefiniteness of “primary interface” does not hinder *inter partes* review. See *Samsung Elecs. Am., Inc. v. Prisua Eng’g Corp.*, 948 F.3d 1342, 1355 (Fed. Cir. 2020).

*sensitive data S* includes enrollment authentication data.” EX1003, 21:33–22:2. EX1002, ¶344.

*Second*, Dickinson discloses receiving the request “*via a primary interface*” to the same extent the ’194 Patent provides written description for the primary interface. As an initial matter, the term “primary interface” lacks written description because the only “interface” described in the ’194 Patent is “any suitable communications interface 2508 (e.g., USB, serial, parallel, Bluetooth, IR, IEEE 1394, ethernet, or any other suitable communications interface) in order to access the original data.” EX1001, 64:51–65:8. To the extent the “*primary interface*” corresponds to “any suitable communications interface,” it is taught by Dickinson. EX1002, ¶¶345–46.

Specifically, Dickinson discloses that “the *data storage facilities D1 through D4* communicate with the transaction engine 205, the authentication engine 215, and the cryptographic engine 220, preferably through conventional SSL *communication links*[.]” EX1003, 19:16–26. Dickinson also discloses, in Figure 1, that “the cryptographic system 100 includes a user system 105, a *trust engine 110*, a certificate authority 115, and a *vendor system 120*, communicating through a *communication link 125*.” EX1003, 9:23–24. Accordingly, the trust engine 110 receives the request from the vendor system 120 via a communication link 125. EX1002, ¶347.

- vii. [7B-4] **identify, from the plurality of storage devices, a set of fastest-responding storage devices necessary to retrieve the minimum number of shares, wherein the set of fastest-responding storage devices are identified based at least in part on the response time of the storage devices, retrieve from the set of fastest-responding storage devices the minimum number of shares;**

Dickinson discloses [7B-4]. EX1002, ¶¶349, 353.

Dickinson discloses “*identify, from the plurality of storage devices, a set of fastest-responding storage devices necessary to retrieve the minimum number of shares, wherein the set of fastest-responding storage devices are identified based at least in part on the response time of the storage devices*” for the reasons provided in Section IX.A.4.v [Ground I, 1D]. EX1002, ¶¶350-351.

Dickinson discloses “*retrieve from the set of fastest-responding storage devices the minimum number of shares*” for the reasons provided in Section IX.A.4.vi [Ground I, 1E]. EX1002, ¶352.

- viii. [7B-5] **reconstruct the data set using exclusively the minimum number of shares; and**

Dickinson discloses [7B-5]. EX1002, ¶¶354, 356.

As explained in Section IX.A.4.vi [Ground I, 1E], Dickinson describes “*receiv[ing] data portions from at least the first two of the data storage facilities D1 through D4[.]*” EX1003, 21:21–32. Then, Dickinson discloses “the data assembling module 525 may *assemble* the *sensitive data S*, when, for example, it

receives data *portions* from at least the first *two* of the *data storage facilities D1 through D4* to respond to an assemble request by the trust engine 110.” EX1003, 21:21–32. Because Dickinson discloses “*assembl[ing]* the *sensitive data S* from ... the first *two* of the *data storage facilities D1 through D4*,” Dickinson discloses assembling the data from *exclusively two* of the data *portions*. EX1002, ¶355.

**ix. [7B-6] send the data set responsive to the request.**

Dickinson/Hardjono teaches [7B-6] for the reasons provided in Section IX.A.4.viii [Ground I, 1G]. EX1002, ¶357.

**11. Claim 8**

**i. [8Pre] The electronic computing system of claim 7, wherein the instructions cause the processing unit to**

Dickinson discloses [8Pre]. *See* Section IX.A.10 [Ground I, 7]. EX1002, ¶358.

In particular, “*wherein the instructions cause the processing unit to*”— is satisfied because, as explained in Section IX.A.4.v [Ground I, 1D], Dickinson’s trust engine modules are implemented in software, hardware, or a combination. *See, e.g.*, EX1003, 18:20–21, 18:23–24, 19:3–5. A POSITA would have understood that those implementations include a system memory comprising instructions that, when executed by the processing unit, cause the processing unit to perform [8A] and [8B], as disclosed further below, because they process data to perform the storage method. Using instructions to implement the software was well known. *See* EX1004, 8:63–

65 (“These software routines comprise a *plurality or series of instructions* to be executed by a processor, such as processor 502 of FIG. 5.”). EX1002, ¶¶359–60.

- ii. **[8A] store a first subset of the shares at a first subset of the plurality of storage devices that is physically located at a first data center and**

Dickinson teaches [8A] for the reasons provided in Section IX.A.5.i [Ground I, 2A]. EX1002, ¶361.

- iii. **[8B] to store a second subset of the shares at a second subset of the plurality of storage devices that is physically located at a second data center, the first data center being geographically separated from the second data center.**

Dickinson teaches [8B] for the reasons provided in Section IX.A.5.ii [Ground I, 2B]. EX1002, ¶362.

## 12. Claim 9

- i. **[9] The electronic computing system of claim 7, wherein the instructions further cause the processing unit to generate the plurality of shares in response to receiving a request to store the data set.**

Dickinson discloses the additional limitation of [9]. EX1002, ¶¶363, 369.

*First*, Dickinson discloses “*receiving a request to store the data set.*”

Dickinson discloses that “the enrollment process 900 begins at STEP 905 when a user *desires to enroll* with the trust engine 110 of the cryptographic system 100.”

EX1003, 22:7–17. Dickinson teaches that the *enrollment process* includes “*transmit[ting] enrollment authentication data.*” EX1003, FIG. 9A. Dickinson

also discloses “*transmit[ing] the enrollment* data, for example, through and [sic] XML document, to the *trust engine 110*, and in particular, to the transaction engine 205.” EX1003, 22:15–17. Dickinson discloses that, “[i]n STEP 925, the authentication engine 215 *forwards* each *portion* of the randomized numbers to one of the data storage facilities D1 through D4.” EX1003, 22:7–17. As explained in Section IX.A.4.iii [Ground I, 1B], the *portions* are *stored* at the *data storage facilities D1 through D4*. EX1003, 21:10–14. A POSITA would have understood that the *enrollment process*, which leads to *storing* the data set as *pieces* in the *data storage facilities D1 through D4* is a “*request to store the data set*”: a POSITA would understand that transmitting the enrollment data set to be stored would be the same as a request for that enrollment data to be stored. EX1002, ¶¶364–366.

*Second*, Dickinson discloses “*generat[ing] the plurality of shares* in response to receiving a *request to store the data set*.” Dickinson discloses that, “[i]n STEP 920,” “the authentication engine 215 employs the data splitting module to *mathematically operate* on the enrollment authentication data so as to *split* the data into at least *two independently undecipherable, randomized, numbers*.” EX1003, 23:7–17. That is, Dickinson discloses *storing* “each *portion of the randomized numbers*” without requiring an additional subprocess or command.

9/21

900

| ENROLLMENT DATA FLOW               |                              |      |                                                                                                                                                      |
|------------------------------------|------------------------------|------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| SEND                               | RECEIVE                      | SSL  | ACTION                                                                                                                                               |
| 905<br>USER                        | TRANSACTION ENGINE (TE)      | 1/2  | TRANSMIT ENROLLMENT AUTHENTICATION DATA (B) AND THE USER ID (UID) ENCRYPTED WITH THE PUBLIC KEY OF THE AUTHENTICATION ENGINE (AE) AS (PUB_AE(UID,B)) |
| 915<br>TE                          | AE                           | FULL | FORWARD TRANSMISSION                                                                                                                                 |
| 920                                |                              |      | AE DECRYPTS AND SPLITS FORWARDED DATA                                                                                                                |
| 925<br>AE                          | THE Xth DEPOSITORY (DX)      | FULL | STORE RESPECTIVE PORTION OF DATA                                                                                                                     |
| WHEN DIGITAL CERTIFICATE REQUESTED |                              |      |                                                                                                                                                      |
| 930<br>AE                          | CRYPTOGRAPHIC ENGINE (CE)    | FULL | REQUEST KEY GENERATION                                                                                                                               |
| 935                                |                              |      | CE GENERATES AND SPLITS KEY                                                                                                                          |
| 945<br>CE                          | TE                           | FULL | TRANSMIT REQUEST FOR DIGITAL CERTIFICATE                                                                                                             |
| 950<br>TE                          | CERTIFICATION AUTHORITY (CA) | 1/2  | TRANSMIT REQUEST                                                                                                                                     |
| 955<br>CA                          | TE                           | 1/2  | TRANSMIT DIGITAL CERTIFICATE                                                                                                                         |
| 960<br>TE                          | USER                         | 1/2  | TRANSMIT DIGITAL CERTIFICATE                                                                                                                         |
| TE                                 | MS                           | FULL | STORE DIGITAL CERTIFICATE                                                                                                                            |
| 965<br>TE                          | DX                           | FULL | STORE RESPECTIVE PORTION OF KEY                                                                                                                      |

**FIG. 9A**

EX1003, FIG. 9A. EX1002, ¶367.

*Third*, Dickinson discloses “*wherein the instructions further cause the processing unit to generate the plurality of shares.*” As explained in Section IX.A.4.v [Ground I, 1D], Dickinson’s trust engine modules are implemented in software, hardware, or a combination. *See, e.g.*, EX1003, 18:20–21, 18:23–24, 19:3–5. A POSITA would have understood that those implementations include a system memory comprising instructions that, when executed by the processing unit, “*cause the processing unit to generate the plurality of shares.*” EX1002, ¶368.

**13. Claim 14**

Claim 14 only varies from claim 7 for the preamble [14Pre] and limitation [14B]. The Dickinson/Hardjono System discloses [14B]–[14H] for the same reasons provided above for [7B-1]–[7B-6]. EX1002, ¶¶370–371.

| <b>Claim 7</b> | <b>Claim 14</b> |
|----------------|-----------------|
| [7B-1]         | [14B]           |
| [7B-2]         | [14C]           |
| [7B-3]         | [14D]           |
| [7B-4]         | [14E] + [14F]   |
| [7B-5]         | [14G]           |
| [7B-6]         | [14H]           |

The Dickinson/Hardjono System also discloses [14Pre] and [14A] for the reasons provided below. EX1002, ¶372.

- i. **[14Pre] A non-transitory computer-readable storage medium comprising instructions that, when executed at an electronic computing device, cause the electronic computing device to:**

Dickinson discloses [14Pre]. EX1002, ¶¶373, 378.

*First*, as explained in Section IX.A.10.i [Ground I, 7Pre], Dickinson discloses an “*electronic computing system*,” which is equivalent to an “*electronic computing device*.” EX1002, ¶374.

*Second*, Dickinson discloses a “*non-transitory computer-readable storage medium comprising instructions that ... cause the electronic computing device to*” perform the steps of [14A] through [14H]. Dickinson discloses that trust engine includes a “depository 210 [that] comprises a single *logical memory structure* indexing authentication data and cryptographic key data to a unique user ID.” EX1003, 16:29–31. In addition, as explained in Section IX.A.4.v [Ground I, 1D], Dickinson’s trust engine modules are implemented in software, hardware, or a combination. *See, e.g.*, EX1003, 18:20–21, 18:23–24, 19:3–5. A POSITA would have understood that those implementations include non-transitory computer-readable storage medium comprising instructions that ... cause the electronic computing device to perform the steps of [14A] through [14H] because they process data to perform the storage method. *See* Secs. IX.A.13.ii, IX.A.13.iii, IX.A.13.iv,

IX.A.13.v, IX.A.13.vi, IX.A.13.vii, IX.A.13.viii, IX.A.13.ix [Ground I, 14A-H].  
EX1002, ¶¶375–377.

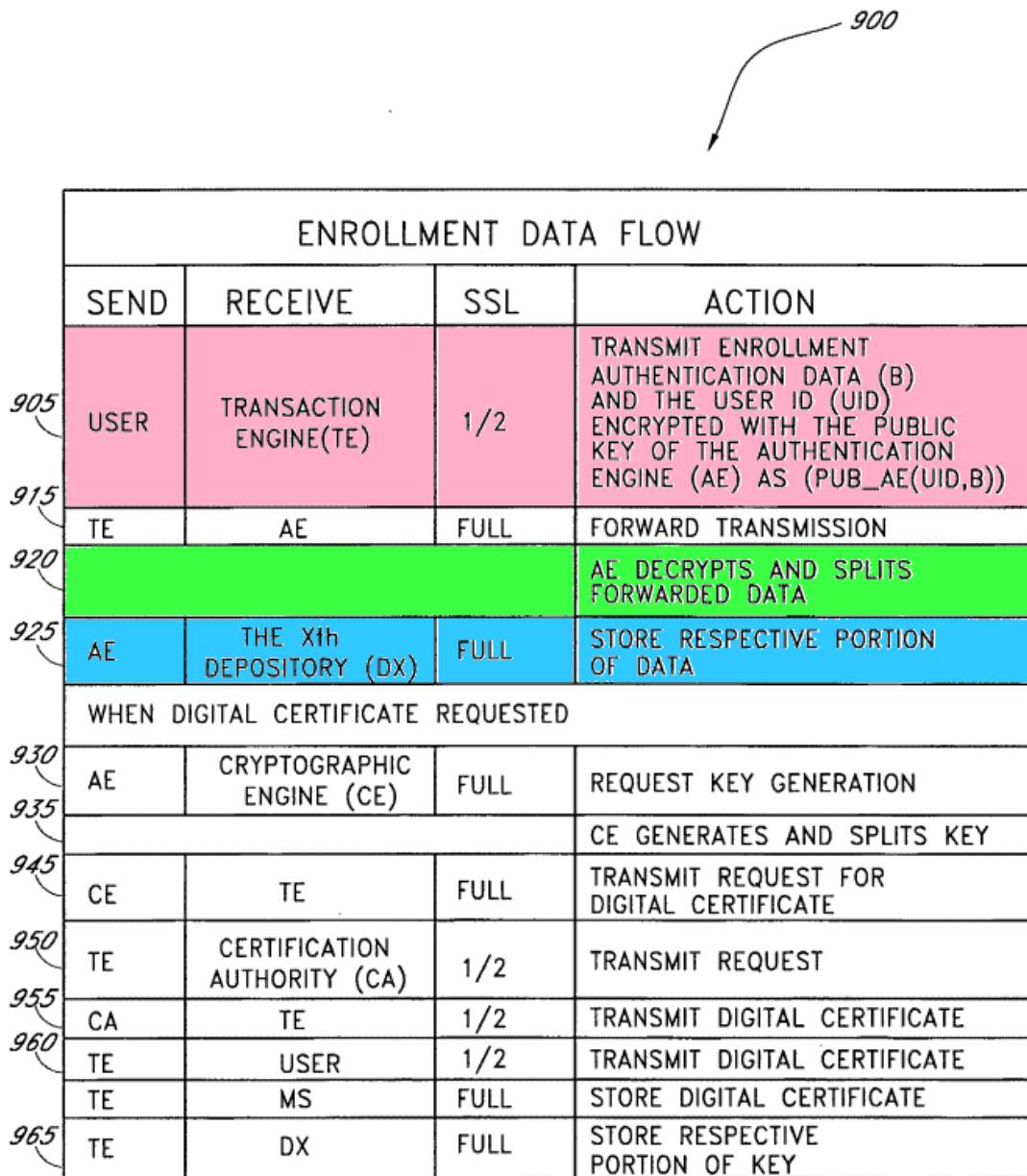
**ii. [14A] receive a request to write a data set to a storage location;**

Dickinson discloses [14A]. EX1002, ¶¶379, 384.

*First*, Dickinson discloses “*receive a request to [store] a data set to a storage location.*” Dickinson discloses that “the enrollment process 900 begins at STEP 905 when a user *desires to enroll* with the trust engine 110 of the cryptographic system 100.” EX1003, 22:7–17. Dickinson teaches that the *enrollment process* includes “*transmit[ting] enrollment authentication data.*” EX1003, FIG. 9A. Dickinson also discloses “*transmit[ting] the enrollment* data, for example, through and [sic] XML document, to the *trust engine 110*, and in particular, to the transaction engine 205.” EX1003, 22:15–16. Dickinson discloses that, “[i]n STEP 925, the authentication engine 215 *forwards* each *portion* of the randomized numbers to one of the data storage facilities D1 through D4.” EX1003, 22:7–17. As explained in Section IX.A.4.iii [Ground I, 1B], the *portions* are *stored* at the *data storage facilities D1 through D4*. EX1003, 21:10–14. A POSITA would have understood that the *enrollment process*, which leads to *storing* the data set as *pieces* in the *data storage facilities D1 through D4* is a “*request to [store] a data set to a storage*

location”: a POSITA would understand that transmitting the enrollment data set to be stored would be the same as a request for that enrollment data to be stored.

9/21



**FIG. 9A**

EX1003, FIG. 9A. EX1002, ¶¶380–382.

*Second*, Dickinson discloses “*receiving a request to write the data set.*” A POSITA would understand that “distributing the sensitive data into distinct and independent *storage facilities D1 through D4*” where it is “*stored*” in Dickinson would include “*writing*” that data to those storage locations, as claimed, because storing a data set requires writing that data set in either short- or long-term memory. EX1002, ¶383.

- iii. **[14B] generate a plurality of shares by performing a cryptographic operation on the data set and distributing the data set in the plurality of shares such that the data set can be reconstructed using any subset of the shares that includes at least a minimum number less than all of the shares and such that the data set cannot be reconstructed using any subset of the shares that includes fewer than the minimum number of the shares;**

Dickinson discloses [14B] for the reasons provided in Section IX.A.10.iv [Ground I, 7B-1]. EX1002, ¶385.

- iv. **[14C] store the plurality of shares at a plurality of storage devices;**

Dickinson discloses [14C] for the reasons provided in Section IX.A.10.v [Ground I, 7B-2]. EX1002, ¶386.

- v. **[14D] receive a request to retrieve the data set;**

Dickinson discloses [14D] for the reasons provided in Section IX.A.10.vi [Ground I, 7B-3]. EX1002, ¶387.

- vi. [14E] identify, from the plurality of storage devices, a set of fastest-responding storage devices necessary to retrieve the minimum number of shares, wherein the set of fastest-responding storage devices are identified based at least in part on the response time of the storage devices;**

Dickinson discloses [14E] for the reasons provided in Section IX.A.10.vii

[Ground I, 7B-4]. EX1002, ¶388.

- vii. [14F] retrieve from the set of fastest-responding storage devices, the minimum number of shares;**

Dickinson discloses [14F] for the reasons provided in Section IX.A.10.vii

[Ground I, 7B-4]. EX1002, ¶389.

- viii. [14G] reconstruct the data set using exclusively the minimum number of shares; and**

Dickinson discloses [14G] for the reasons provided in Section IX.A.10.viii

[Ground I, 7B-5]. EX1002, ¶390.

- ix. [14H] send the data set responsive to the request.**

Dickinson discloses [14H] for the reasons provided in Section IX.A.10.ix

[Ground I, 7B-6]. EX1002, ¶391.

**14. Claim 16**

- i. [16] The non-transitory computer-readable storage medium of claim 14, wherein the instructions further cause the processing unit to generate the plurality of shares in response to receiving the request to write the data set to the storage location.**

Dickinson/Hardjono System discloses the additional limitation of claim 16 for the reasons provided in Section IX.A.12.i [Ground I, Claim 9]. EX1002, ¶392.

In addition, a POSITA would have understood that “*writ[ing] a data set to storage*” is equivalent to “[*storing*] a data set to storage.” A POSITA would have understood that “[*storing*] a data set” is equivalent to “*writ[ing] a data set*,” as storing a data set requires writing that data set in either short- or long-term memory. EX1002, ¶393.

**15. Claims 10–13, 15, and 17–20**

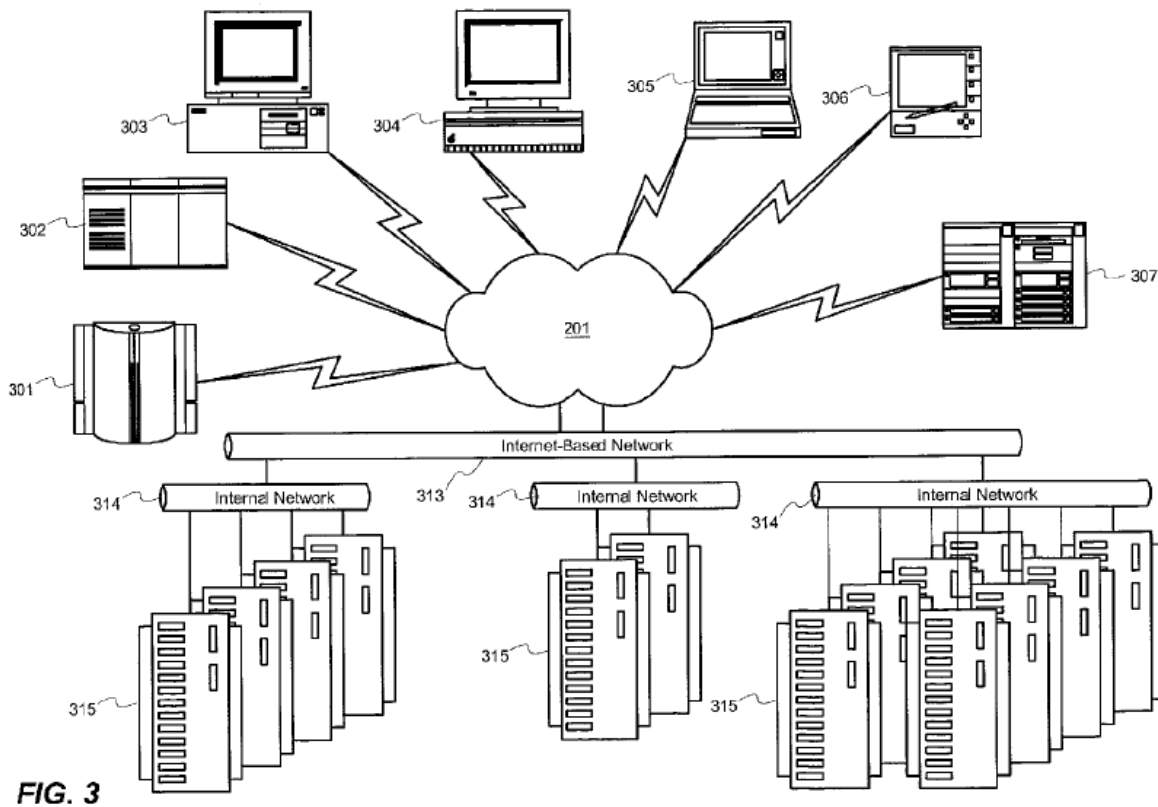
- Dickinson/Hardjono System satisfies the additional limitation of claims 10 and 17 as provided in Section IX.A.6.i [Ground I, 3];
- Dickinson/Hardjono System satisfies the additional limitation of claims 11 and 18 as provided in Section IX.A.7.i [Ground I, 4];
- Dickinson/Hardjono System satisfies the additional limitation of claims 12 and 19 as provided in Section IX.A.8.i [Ground I, 5];
- Dickinson/Hardjono System satisfies the additional limitation of claims 13 and 20 as provided in Section IX.A.9.i [Ground I, 6]; and
- Dickinson/Hardjono System satisfies the additional limitation of claim 15 as provided in Section IX.A.11 [Ground I, 8].

EX1002, ¶¶394–398.

**B. Ground II: Dickinson, Hardjono and Moulton Render Obvious  
Claims 2, 8, And 15.**

**1. Moulton (EX1018)**

Moulton is directed to a “distributed network storage” system focused on “intelligent management of globally distributed network storage.” EX1018, Title, [0002]. Moulton’s system comprises multiple “storage nodes,” each situated at a physical location with one or more defined “contexts.” EX1018, FIG. 3, [0017]. These storage nodes are interconnected via a communications network, and operate over public channels such as the Internet. EX1018, [0023], [0024]. For example, “[i]nternetnetwork 201 connects devices from data centers to handheld or pen-based devices,” as illustrated in FIG. 3. EX1018, [0036].



EX1018, Fig. 3. EX1002, ¶¶399–401.

Moulton teaches a mechanism for selecting one or more storage nodes whose associated contexts satisfy the desired criteria and executing storage tasks on those selected nodes. EX1018, [0027], Claim 14. Additionally, the selected storage nodes may be “located in *different data centers*,” providing geographic distribution of stored data. EX1018, Claim 19 (“The method of claim 14 wherein the selected storage nodes comprise at least two storage nodes and the at least two storage nodes are located in *different data centers*.”). EX1002, ¶¶402–403.

## 2. Dickinson in View of Hardjono and Moulton

As already explained above in Section IX.A.5 [Ground I, 2], Dickinson teaches that the first and second subsets of shares are stored at two geographically separated storage facilities, each of which may correspond to different data centers. EX1002, ¶404.

To the extent Patent Owner contends that Dickinson’s “geographically separated data storage facilities” do not expressly teach data centers, that would have been obvious in view of Moulton. EX1002, ¶405.

As explained above in Section IX.B.1 [Moulton], Moulton specifically discloses a distributed storage system with storage nodes in different data centers. EX1018 (Moulton) at [0036] (“Internetwork 201 enables the interconnection of a heterogeneous set of computing devices and mechanisms ranging from a supercomputer or data center 301 to a hand-held or pen-based device 306.”); Claim 19 (“The method of claim 14 wherein the selected storage nodes comprise at least two storage nodes and the at least two storage nodes are located in *different data centers.*”). EX1002, ¶405.

Accordingly, it would have been obvious to modify the Dickinson/Hardjono combination with Moulton’s teaching, such that the geographically separated storage facilities of Dickinson/Hardjono are implemented in different data centers (EX1018, [0036], Claim 19). EX1002, ¶406.

**i. Reasons to Combine Dickinson with Hardjono and Moulton**

A POSITA would have been motivated to combine Dickinson/Hardjono with Moulton. EX1002, ¶407.

*First*, Moulton teaches a “globally distributed network storage” architecture that “readily scale[s] to a virtually unlimited number of [storage] nodes” and places those nodes in contexts including data centers. EX1018, [0002], [0029], [0036], Claim 19. To scale the Dickinson/Hardjono’s storage system to a larger capacity, a POSITA would have been motivated to locate its storage facilities in multiple data centers as disclosed by Moulton. EX1002, ¶¶408–409.

*Second*, Dickinson already describes its depository system 700 as comprising multiple, geographically separated data storage facilities (D1–D4) that “may advantageously comprise ... multiple *geographically separated independent data storage systems*.” EX1003, 19:17–20, 20:1–14. Moulton’s teaching that such independent nodes may reside in different data centers (EX1018, Claim 19) would have provided a motivation to a POSITA to implement Dickinson’s separated facilities as “different data centers.” EX1002, ¶¶410–411.

*Third*, this combination of Dickinson/Hardjono and Moulton merely applies a known technique—Moulton’s use of storage nodes in different data centers—to an existing, compatible storage architecture. Because Dickinson/Hardjono already

describe a distributed storage framework, implementing Moulton's teaching would have been routine and would have predictably improved the system's capacity and scalability. EX1002, ¶412.

A POSITA would have had a reasonable expectation of success in incorporating Moulton's data-center teachings into the Dickinson/Hardjono architecture for at least the following reasons. EX1002, ¶413.

*First*, both the Dickinson/Hardjono System and Moulton are directed to a networked, distributed storage system. For example, Dickinson/Hardjono describe geographically separated storage systems communicating over conventional network channels (EX1003, 20:2–6, 19:23–24; EX1004, 1:8-9, 3:58–67), and Moulton similarly teaches networked distributed storage systems (EX1018, [0002], [0036]). EX1002, ¶¶414–416.

A POSITA would therefore have understood that Moulton's storage nodes in data centers would have been readily accessible by the computing devices in the Dickinson/Hardjono system—and would have operated like other storage devices accessible via a communications network, e.g., to store and provide data shares generated by the Dickinson/Hardjono system. Accordingly, a POSITA would have reasonably expected that the combination would produce the predictable result of using and operating storage devices in data centers. EX1002, ¶417.

**Second**, implementing Moulton’s teaching would have entailed nothing more than substituting each abstract “data storage facility” in Dickinson/Hardjono with a “storage node in a data center” as disclosed by Moulton. EX1018, Claim 19. A POSITA would have reasonably expected that such combination would result in computing devices within the Dickinson/Hardjono system accessing and using Moulton’s storage nodes in data centers just like other network-accessible storage devices, to store and retrieve data shares generated by the system. EX1002, ¶418.

A POSITA would have reasonably expected that such combination would result in computing devices within the Dickinson/Hardjono system accessing and using storage devices in data centers like other network-accessible storage devices, to store and retrieve data shares generated by the system. EX1002, ¶¶419–420.

### 3. Claim 2

- i. **[2Pre] The method of claim 1, wherein storing the shares comprises:**

Dickinson in view of Hardjono and Moulton teaches [2Pre]. *See* Section IX.A.4 [Ground I, Claim 1]. EX1002, ¶421.

- ii. **[2A] storing a first subset of the shares at a first subset of the plurality of storage devices that is physically located at a first data center; and**

The Dickinson/Hardjono/Moulton combination teaches “*storing a first subset of the shares at a first subset of the plurality of storage devices*” because

Dickinson/Hardjono teaches this portion of the limitation as explained in Section IX.A.5.i [Ground I, 2A]. EX1002, ¶¶422, 424.

Furthermore, the Dickinson/Hardjono/Moulton combination teaches that the “*first subset of the plurality of storage devices*” “*is physically located at a first data center*” because Moulton discloses that storage nodes may be physically located at “different data centers.” EX1018, [0036] (“Internetwork 201 enables the interconnection of a heterogeneous set of computing devices and mechanisms ranging from a supercomputer or *data center* 301 to a hand-held or pen-based device 306.”); Claim 14 (“selecting one or more of the plurality of storage nodes having an associated context satisfying the desired criteria; and executing the storage task in the one or more selected storage nodes.”); Claim 19 (“The method of claim 14 wherein the selected storage nodes comprise at least two storage nodes and the at least two storage nodes are *located in different data centers*.”). EX1002, ¶423.

- iii. **[2B] storing a second subset of the shares at a second subset of the plurality of storage devices that is physically located at a second data center, the first data center being geographically separated from the second data center.**

The Dickinson/Hardjono/Moulton combination teaches “*storing a second subset of the shares at a second subset of the plurality of storage devices*” because Dickinson/Hardjono teaches this portion of the limitation as explained in Section IX.A.5.ii [Ground I, Claim 2B]. EX1002, ¶¶425, 429.

Additionally, the Dickinson/Hardjono/Moulton combination teaches that the “*second subset of the plurality of storage devices*” “*is physically located at a second data center*” because Moulton discloses that storage nodes are physically located at “different data centers.” EX1018 (Moulton) at [0036] (“Internetwork 201 enables the interconnection of a heterogeneous set of computing devices and mechanisms ranging from a supercomputer or *data center* 301 to a hand-held or pen-based device 306.”); Claim 19 (“The method of claim 14 wherein the selected storage nodes comprise at least two storage nodes and the at least two storage nodes are *located in different data centers*.”). EX1002, ¶¶426, 428.

Lastly, as discussed in Section IX.A.5.ii, [Ground I, Claim 2B], Dickinson discloses that “*the first data center [is] geographically separated from the second data center.*” For instance, Dickinson teaches “distributing the sensitive data into distinct and *independent storage facilities D1 through D4*, some or all of which may be advantageously *geographically separated.*” EX1003, 19:17–20, 20:1–14. And Moulton further teaches that such “geographically separated” storage facilities are data centers. EX1018, [0036] (“Internetwork 201 enables the interconnection of a heterogeneous set of computing devices and mechanisms ranging from a supercomputer or *data center* 301 to a hand-held or pen-based device 306.”); Claim 19 (“The method of claim 14 wherein the selected storage nodes comprise at least

two storage nodes and the at least two storage nodes are located in *different data centers.*”). EX1002, ¶¶427–428.

**4. Claim 8**

**i. [8Pre]: The electronic computing system of claim 7, wherein the instructions cause the processing unit to**

Dickinson in view of Hardjono and Moulton teaches [8Pre]. *See* Section IX.A.10 [Ground I, Claim 7]. EX1002, ¶430.

Dickinson explains that its trust engine’s core modules—the data splitting module 520, the data assembling module 525, and the cryptographic handling module 625—“advantageously comprise a software, hardware, or combination module,” and that “the cryptographic handling module 625 may comprise software modules or programs, hardware, or both.” EX1003, 18:20–21, 18:23–24, 19:3–5. In other words, the instructions for performing the steps recited in [8A]–[8B] are stored in system memory and realized by software and/or hardware components within the processing unit. In addition, implementing the software via instructions was well known. *See, e.g.*, EX1004, 8:63–65 (“These software routines comprise a *plurality or series of instructions* to be executed by a processor, such as processor 502 of FIG. 5.”). EX1002, ¶¶431–432.

A POSITA would have understood that this architecture involves a processing unit executing stored instructions to carry out the first two elements of claim [8],

because those elements involve data processing required to implement the storage method. EX1002, ¶433.

- ii. **[8A]: to store a first subset of the shares at a first subset of the plurality of storage devices that is physically located at a first data center and**

Dickinson in view of Hardjono and Moulton teaches [8A] for the reasons provided in Section IX.B.3.ii [Ground II, 2A]. EX1002, ¶434.

- iii. **[8B]: to store a second subset of the shares at a second subset of the plurality of storage devices that is physically located at a second data center, the first data center being geographically separated from the second data center.**

Dickinson in view of Hardjono and Moulton teaches [8B] for the reasons provided in Section IX.B.3.iii [Ground II, 2B]. EX1002, ¶435.

## 5. Claim 15

Dickinson in view of Hardjono and Moulton renders obvious [15] for the reasons provided in Section IX.B.4 [Ground II, Claim 8]. EX1002, ¶436.

## X. Conclusion

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

Petition for *Inter Partes* Review of U.S. Patent No. 8,904,194  
Claims 1-20

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. 8,904,194 has, excluding the portions exempted under 37 C.F.R. § 42.24(a), 13,733 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. 8,904,194 as indicated in Patent Center:

Security First Innovations, LLC  
c/o Farjami & Farjami LLP  
26522 La Alameda Ave., Suite 360  
Mission Viejo, CA 92691

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*