

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Netskope, Inc.

Petitioner,

v.

Fortinet, Inc.

Patent Owner.

Case No. IPR2023-00458

Patent No. 9,280,678

**SUPPLEMENTAL DECLARATION OF DR. JOHN BLACK JR.
IN SUPPORT OF PATENT OWNER'S RESPONSE**

1. I have been retained on behalf of Fortinet, Inc. (“Patent Owner”) in connection with the above-captioned *inter partes* review (IPR). I have been retained to provide my opinions in support of Fortinet’s Patent Owner Preliminary Response. I am being compensated for my time at the rate of \$625 per hour. I have no interest in the outcome of this proceeding.

2. In preparing this Declaration, I have reviewed and am familiar with the Petition for IPR2023-00458, U.S. Patent No. 9,280,678 B2 (the “’678 patent”) and its file history, and all other materials cited and discussed in the Petition (including the declaration of Petitioner’s expert, Dr. Wenke Lee) and cited and discussed in this Declaration. I have also reviewed and am familiar with the Board’s institution decision and the transcript from Dr. Lee’s deposition, dated December 13, 2023. I previously submitted a declaration in connection with this IPR on June 20, 2023, which I incorporate herein in its entirety.

3. The statements made herein are based upon the cited documents as well as my own knowledge and opinion. This Declaration represents only the opinions I have formed to date. I may consider additional documents as they become available or other documents that are necessary to form my opinions. I reserve the right to revise, supplement, or amend my opinions based on new information and on my continuing analysis.

I. QUALIFICATIONS

4. My qualifications can be found in my Curriculum Vitae (*see* Ex. 2002), which includes my detailed employment background, professional experience, and list of technical publications and patents.

5. I am an Associate Professor of Computer Science at the University of Colorado, Boulder. I received a B.S. in Mathematics and Computer Science from the California State University at Hayward (now "California State University, East Bay") in 1988. I received an M.S. in Computer Science in 1997, and a Ph.D. in Computer Science in 2000, both from the University of California at Davis.

6. I have taught more than 60 classes in computer science, on subjects including data structures, algorithms, networking, operating systems, software engineering, security, cryptography, discrete mathematics, and quantum computing. I have authored or coauthored more than 20 publications, primarily on issues relating to computer security. I have been involved with computers for over 40 years in both commercial and academic capacities.

7. My earliest interest was in networks and security. My first memories in this regard were around 1975 when a group of friends and I learned about the telephone network and its security. A few years later, personal computers became available and I spent most of my free time studying, programming, and modifying them. I worked extensively with various networking products throughout the 1980s,

and developed an interest in cryptography soon thereafter. Although my undergraduate institution had no courses in cryptography or security in the 1980s, I decided to pursue self-study at the time, and opted to double major in Computer Science and Mathematics because cryptography is a blend of these two subject areas.

8. After earning my B.S. degree in 1988, I worked for six years at Ingres Corp as a software developer, writing and reviewing code written in C. My work primarily was directed at transaction logging, data type support, and security.

9. In 1995, I began my Ph.D. at UC Davis under cryptographer Phillip Rogaway. My area of focus was cryptography and security and my research involved encryption, authentication, hash functions, and network security.

10. After graduation I took a position as an Assistant Professor at the University of Nevada at Reno. In the Fall of 2001, I taught the networking class there, which included coverage of Ethernet, interior gateway protocols, exterior gateway protocols, ARP, DHCP/BOOTP, IP, UDP, TCP, HTTP, SMTP and other protocols. In 2001, a graduate student and I looked at the security of ARP and invented a new protocol "AuthARP" to add security to the protocol.

11. In 2002, I moved to the University of Colorado at Boulder where I am currently employed. In the Fall of 2002, I co-designed and co-taught a new course called "Foundations of Computer and Network Security," which included descriptions of security issues around both wired and wireless security challenges,

mostly for public-facing network services including the world-wide web. I have taught this class seven more times since then, including modern topics such as wireless networking, the Internet of Things, and so forth.

12. In my career at the University of Colorado, I have published several more papers in the area of cryptography and network security. I have taught more than 30 courses in network security and cryptography, and have graduated several PhD students in these areas.

13. I also have worked for a consulting company at times, writing software on contract basis. Although most projects are covered by NDAs, many involved networks and/or aspects of computer security.

14. In 2011, I began technical consulting for a local company called Cardinal Peak, which focuses primarily on video encoding and delivery systems. My work for Cardinal Peak has largely been directed to video encoding, transcoding, compression, encryption, and DRM, but also has included networking projects and embedded device work. For example, I designed the security system for the Pro1 smart thermostat, implemented the DRM for the Yonder Music App, worked on 802.1X code for smart dog collars, and helped design the cryptography used in Fitbit devices for wireless transfer of a Fitbit watch to a phone or laptop.

15. In 2016, I took a two-and-a-half year leave of absence from the University of Colorado to start a company named "SecureSet" in Denver, Colorado.

The objective of SecureSet is to take reasonably proficient technical people and turn them into computer and network security specialists via five months of intensive training. SecureSet was sold to WeWork in 2019 and continues to offer computer security classes today. In 2018 I returned to my position at the University where I remain employed to the present day.

II. LEGAL UNDERSTANDING

A. The Person of Ordinary Skill in the Art

16. I understand that a person of ordinary skill in the relevant art at the time of the invention (also referred to herein as “POSITA”) is presumed to be aware of all pertinent art, thinks along conventional wisdom in the art, and is a person of ordinary creativity—not an automaton.

17. I have been asked to consider the level of ordinary skill in the field that someone would have had at the time the claimed invention was made. In deciding the level of ordinary skill, I considered the following:

- the levels of education and experience of persons working in the field;
- the types of problems encountered in the field; and
- the sophistication of the technology.

18. A person of ordinary skill in the art relevant to the ’678 patent at the time of the invention would have had a Bachelor of Science degree in electrical

engineering and/or computer science, and two years of work or research experience in the fields of network and data security, or a Master's degree in electrical engineering and/or computer science and one year of work or research experience in related fields.

19. I have reviewed the declaration of Dr. Lee, including his opinions regarding the Person of Ordinary Skill in the Art. My description of the level of ordinary skill in the art is essentially the same as that of Dr. Lee. The opinions set forth in this Declaration would be the same under either my or Dr. Lee's proposal.

20. I am well-qualified to determine the level of ordinary skill in the art and am personally familiar with the technology of the '678 patent. I was a person of at least ordinary skill in the art at the time of the priority date of the '678 patent. Regardless if I do not explicitly state that my statements below are based on this timeframe, all of my statements are to be understood as a POSITA would have understood something as of the priority date of the '678 patent.

B. Legal Principles

21. I am not a lawyer and will not provide any legal opinions. Though I am not a lawyer, I have been advised that certain legal standards are to be applied by technical experts in forming opinions regarding the meaning and validity of patent claims.

1. Obviousness

22. I understand that to obtain a patent, a claimed invention must have, as of the priority date, been nonobvious in view of prior art in the field. I understand that an invention is obvious when the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art.

23. I understand that to prove that prior art, or a combination of prior art, renders a patent obvious, it is necessary to: (1) identify the particular references that singly, or in combination, make the patent obvious; (2) specifically identify which elements of the patent claim appear in each of the asserted references; and (3) explain how the prior art references could have been combined to create the inventions claimed in the asserted claim.

24. I understand that a patent composed of several elements is not proved obvious merely by demonstrating that each of its elements was, independently, known in the prior art, and that obviousness cannot be based on the hindsight combination of components selectively culled from the prior art to fit the parameters of the patented invention.

25. I also understand that a reference may be said to teach away when a person of ordinary skill, upon reading the reference, would be discouraged from

following the path set out in the reference, or would be led in a direction divergent from the path that was taken by the applicant. Even if a reference is not found to teach away, I understand its statements regarding preferences are relevant to a finding regarding whether a skilled artisan would be motivated to combine that reference with another reference.

2. *My Understanding of Claim Construction Law*

26. I have been informed that patent claims are construed from the viewpoint of a person of a POSITA and that patent claims generally should be understood consistent with their ordinary and customary meaning at the time of the invention. A review of the patent claim language, the patent specification, and its prosecution history are also necessary to determine the proper meaning and scope of the term at issue.

27. I have further been informed that in the specification and prosecution history the patentee may define a claim term in a way that differs from the ordinary and customary meaning. I understand that during prosecution of the patent before the U.S. Patent and Trademark Office, the Applicant may make representations or provide definitions of terms that may affect the scope of the patent claims. In particular, the Applicant may, during the course of prosecution, limit the scope of the claims to overcome prior art and/or disavow claim coverage by making clear and unambiguous statements to that effect.

28. I have been informed that a POSITA may, among other things, consider dictionaries, publications, other patents, and treatises (*i.e.*, “extrinsic evidence”). I understand that extrinsic evidence may generally not be relied on if it contradicts the meaning of claim language provided by the intrinsic evidence, such as express definitions given for terms in the specification.

III. OVERVIEW OF THE '678 PATENT

A. The '678 Patent Specification

29. The '678 patent is entitled “Secure Cloud Storage Distribution and Aggregation.” The patent provides a vendor independent secure cloud storage solution that introduces new systems and methods for “implementing a policy based framework for encrypting, storing, accessing, querying and managing data across one or more cloud platforms.” Ex. 1001, 3:23-28. Examples of cloud platforms include Dropbox, Google Drive, Amazon Web Services, etc. *Id.*, 6:9-14.

30. In some embodiments, the '678 patent provides a cloud storage distribution and aggregation service via a cloud storage gateway device situated between third-party cloud storage platforms and users. *Id.*, Abstract. Among other attributes, the service may provide an API that supports both file storage and search requests across stored files, along with the ability to remit content from stored files according to the search results. Moreover, the service may assign policies that can be assigned to users that determine access rights, storage attributes, encryption type,

etc. *Id.* And when a file storage request is received, the service may create encrypted data suitable for that user's policy, and then distribute the associated ciphertext among the third-party cloud storage providers based on storage requirements, again as defined by the user's policy. *Id.*, 2:10-30.

31. In the example shown below in Figure 2, the '678 patent provides a system that operates as an intermediary "Gateway" [204], highlighted below, which enables "interactions between enterprise users [202a, 202b, 202c] and cloud service providers [206a, 206b, 206c] for storage, retrieval, and searching of data and content stored by the providers." *Id.*, 8:27-9:4.

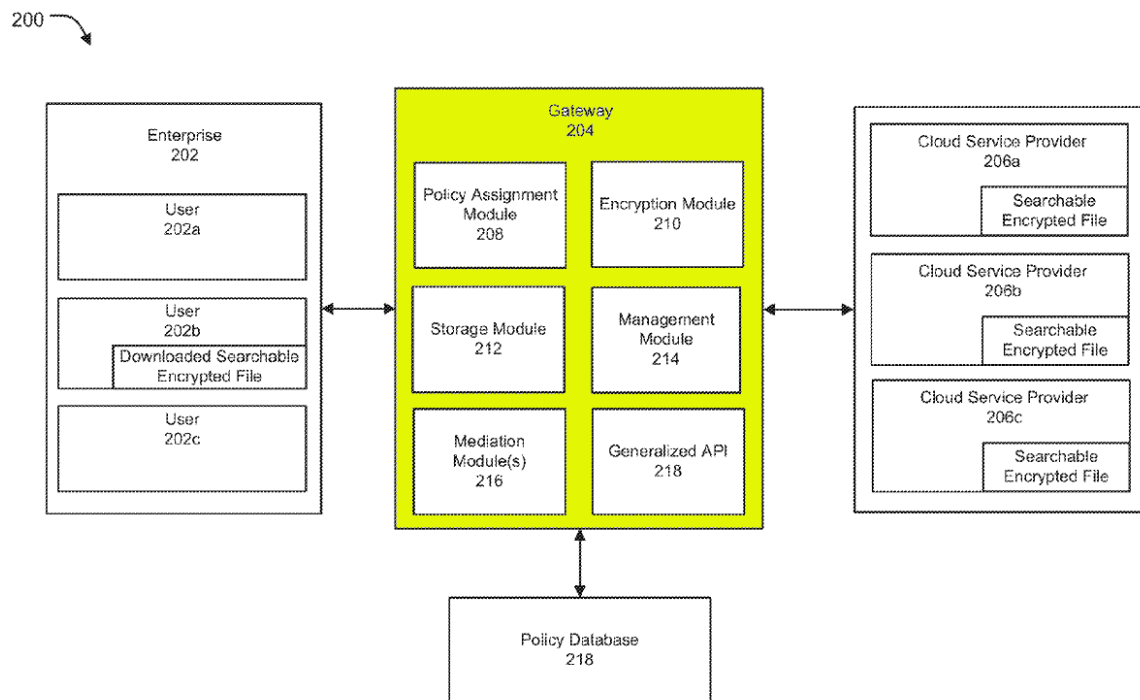


Fig. 2

32. In this example, users within an Enterprise 202 interact with Gateway 204 in order to access cloud storage via cloud storage providers 206. The policy assignment module 208 selects and assigns policies from the policy database 218 to one or more enterprise users 202. *Id.*, 9:10-25. And the encryption module 210 encrypts files that are stored across one or more cloud platforms based on the given user policy. *Id.*, 9:53-56.

33. While typical prior art cloud storage systems allowed for only “rigid and non-flexible” interactions between enterprises and individual cloud service providers, the ’678 patent’s Gateway system gives enterprises the ability to more easily “store data across multiple cloud platforms to ensure more security, redundancy, and reduction in potential threat of data compromise,” as well as the ability to implement global file storage policies that “allow storage of data across different service providers, allow different cloud storage access rights across users and computing devices, [and] allow searching of downloaded encrypted data across cloud service vendors.” *Id.*, 1:38-2:30. As the ’678 patent explains, the Gateway may assign users a policy that “not only defines the manner in which the users can access and process content stored on the cloud, but can also configure the mode in which the data is encrypted, stored, searched, and accessed to ensure secure and vendor independent cloud management.” *Id.*, 3:28-67.

34. The '678 patent also recognizes that storing files on cloud platforms in an “unencrypted format” may expose them to “network attack[s] that may lead to the [files] being compromised.” *Id.*, 1:52-54. For this reason, the Gateway can “encrypt one or more files to be uploaded/stored” using “cryptographic key information” defined by the assigned policy. *Id.*, 4:1-12. But because the '678 patent recognizes that conventional encryption techniques also “make[] the files ... unsearchable and hence unfriendly for user access” (*id.*, 1:54-57), the Gateway is able to produce a “searchable encrypted file” via a process that involves “dividing the file into [] chunks; creating namespaces for one or more of the [] chunks; and configuring the namespaces of the one or more chunks such that content of the file is encrypted in a manner that makes it searchable.” *Id.*, 19:62-20:3; *see also* Fig. 3A.

35. The '678 patent introduces a novel approach for encrypting files to be stored on a cloud storage service—and in such a way that the contents of the files can be searchable using said service despite the service having no access to the file’s encrypted contents.

36. In prior art systems, such as those described in the Shikfa and Chiueh references that the Petition relies on, searches could be performed on encrypted files by first extracting the keywords from the unencrypted file and storing them in an index (*e.g.* a mapping of keywords to file identifiers), and then encrypting both the

file and the index using known encryption techniques. *See, e.g.*, Ex. 1006, ¶29 (“The searchable encryption mechanism used to encrypt the indexes 24 may be any searchable encryption mechanism known in the art”); Ex. 1011, ¶40. The encrypted file could then be stored in a cloud storage service (*e.g.* Dropbox), and searched by querying the encrypted index running on a user device. However, this approach requires that queries be run through the encrypted index, which has several problems: (1) the user device hosting the index may not be powerful enough to perform large, complex queries, and (2) only the device that has the index can perform queries, which reduces portability and reliability. In other words, the encrypted file itself is not searchable—searches are performed through an index.

37. The ’678 patent’s novel approach forgoes the need for a separate index and allows for the use of the built-in searching functionalities of the cloud vendor. In the ’678 patent, “indices, also referred to as keywords hereinafter, which would typically be used as search terms in the form of search queries are stored as ***filenames*** of the chunks.” Ex. 1001, 12:7-9.

38. For example, as shown in Figure 4A, Container 1 (a folder in the cloud service) contains four file chunks (files in the cloud service), with “[t]he name of the folder ... the encrypted version of the original file name” (such as foo.txt) and “the names of the files may include one or more searchable indices,” such as the fourth file “uetywo” being the encrypted keyword “Ronn.” *Id.*, 12:58-13:35. Thus, when

a user wants to search for “Ronn,” the gateway device need merely encrypt the word “Ronn” to generate the cipher text “uetywo,” and then use the cloud storage service’s built-in search functionality to search for the file “uetywo.” Ex. 1001, 10:17-19 (“Access to such searchable encrypted files can allow the users to search through the content of the files within the context of the cloud service providers 206”).

39. The Rozier paper, which the specification incorporates by reference as “Appendix A” (see Ex. 1001, 11:31-36) explicitly explains that the system “makes use of the most basic functionality that almost all of the existing cloud providers provide,” *i.e.* “the file-name search functionality.” Ex. 1003, 40. This novel process provides multiple benefits over the prior art encrypted indices, such as (1) search being performed by the powerful hardware of the cloud service instead of the gateway device or a user device, (2) search not being limited to a device that hosts the index—any device can perform the query, as the keywords are encrypted using a public key. Ex. 1001, 11:63-67.

40. Beyond these improvements over the prior art, applying the searchable encryption process at the Gateway system provides other advantages over traditional prior art systems. Some prior art systems relied on the individual client devices to encrypt files before storing them on a cloud platform, but this approach can make it harder for enterprises to implement global policies that allow other authorized client devices and users in the enterprise to search for and access the encrypted files. In

comparison, the '678 patent's Gateway system provides enterprises with these capabilities. *See, e.g., id.*, 17:21-18:37. And still other prior art systems relied on the cloud service provider to perform file encryption and enforce access control policies, but this created the risk that the files could be “exposed to attack within the cloud, insider jobs and subpoena—all without the knowledge or consent of the data owner.” *Id.*, 1:59-62. In comparison, the '678 patent's Gateway may implement the patent's searchable encryption process “independent of the vendor/service provider of the cloud platforms.” *Id.*, 10:1-6.

41. This combination of features in the '678 patent provides a more secure, flexible, and user-friendly cloud storage solution than what existed in the prior art.

42. Another advantage the '678 patent provides, in some embodiments, is the ability for enterprises and/or their users to specify data storage diversity requirements via their policies (e.g., enterprises/users may specify a variety of third-party cloud storage providers in order to prevent vendor “lock in”), and to switch to other service providers, as desired, by changing policy. This feature not only provides flexibility and robustness, but also means that a user's data is distributed to a diverse set of storage entities, thereby increasing security. *Id.*, 1:38-51.

43. Moreover, another advantage the '678 patent provides, in some embodiments, is simultaneous use of searchable encryption so that large datasets can be searched by a user without the need to retrieve the full file, decrypt it locally, and

then search. In some cases, encrypted data stored via a third party cloud provider involves encryption keys controlled by the cloud provider, meaning that the provider could potentially decrypt and thereby obtain the user's data without the user's consent or knowledge. This means that insider attacks, outside breach, subpoenas and other avenues of attack would be possible. By taking the keys out of the control of the cloud provider, these risks are reduced. *Id.*, 1:52-63.

44. In addition, another advantage the '678 patent provides, in some embodiments, is the provision of a single API that provides a simple uniform abstraction, instead of requiring developers to learn each specific vendor API for each of the underlying cloud providers. *Id.*, 1:63-2:5.

45. Still other advantages are described and/or made evident elsewhere in the '678 patent's specification.

B. The '678 Patent Claims

46. I understand Petitioner challenges independent claims 1 and 16 and dependent claims 2-3, 6, 8-15, 17, and 21-27 of the '678 patent.

47. By way of example, Claim 1 recites:

1. A method comprising:

assigning to one or more users, by a gateway device, a policy for managing access to and processing a file to be stored on one or more cloud platforms, wherein the policy defines access rights of the one or more users;

encrypting, by the gateway device, using cryptographic key information defined by the policy, content of the file to produce a searchable encrypted file by:

dividing the file into a plurality of chunks;

creating namespaces for one or more of the plurality of chunks; and

configuring the namespaces of the one or more chunks such that content of the file is encrypted in a manner that makes it searchable;

storing, by the gateway device, the searchable encrypted file on the one or more cloud platforms based on the policy; and

managing access to the searchable encrypted file by the one or more users based on the policy.

48. Claim 1 of the '678 patent captures many of the central elements of the invention that I described above. Among other things, a gateway device assigns a policy for managing access to a file stored on one or more cloud platforms, the file is encrypted using a searchable encryption algorithm and stored in chunks with configurable namespaces on the one or more cloud platforms based on the policy, and access to the file is thereafter managed by the policy.

IV. OVERVIEW OF CERTAIN ART

49. The petition cites to a number of references, but the only references it relies on in challenging the independent claims are Cidon, Shikfa, Herrmann, Auradkar, Chiueh, and Chambers. The petition's first ground asserts that the combination of Cidon, Shikfa, and Herrmann renders obvious the independent claims (as well as some of the dependent claims). The petition's fourth ground asserts that the combination of Auradkar, Chiueh, and Chambers also renders obvious the independent claims (as well as some of the dependent claims). (Grounds

2, 3, and 5 of the petition address only dependent claims). Below, I briefly summarize Cidon, Shikfa, Herrmann, Auradkar, Chiueh, and Chambers.

A. Cidon (Ex. 1004)

50. Cidon is primarily directed to systems and methods for encrypting files stored on cloud-based storage services. Ex. 1004, ¶10. Cidon depicts its basic method for encrypting a file in Figure 15, shown below. *See also id.*, Abstract, ¶10. However, importantly, a POSITA would have understood that Cidon never discloses that any of its file encryption methods encrypt the content of a file in a manner that makes the content searchable.

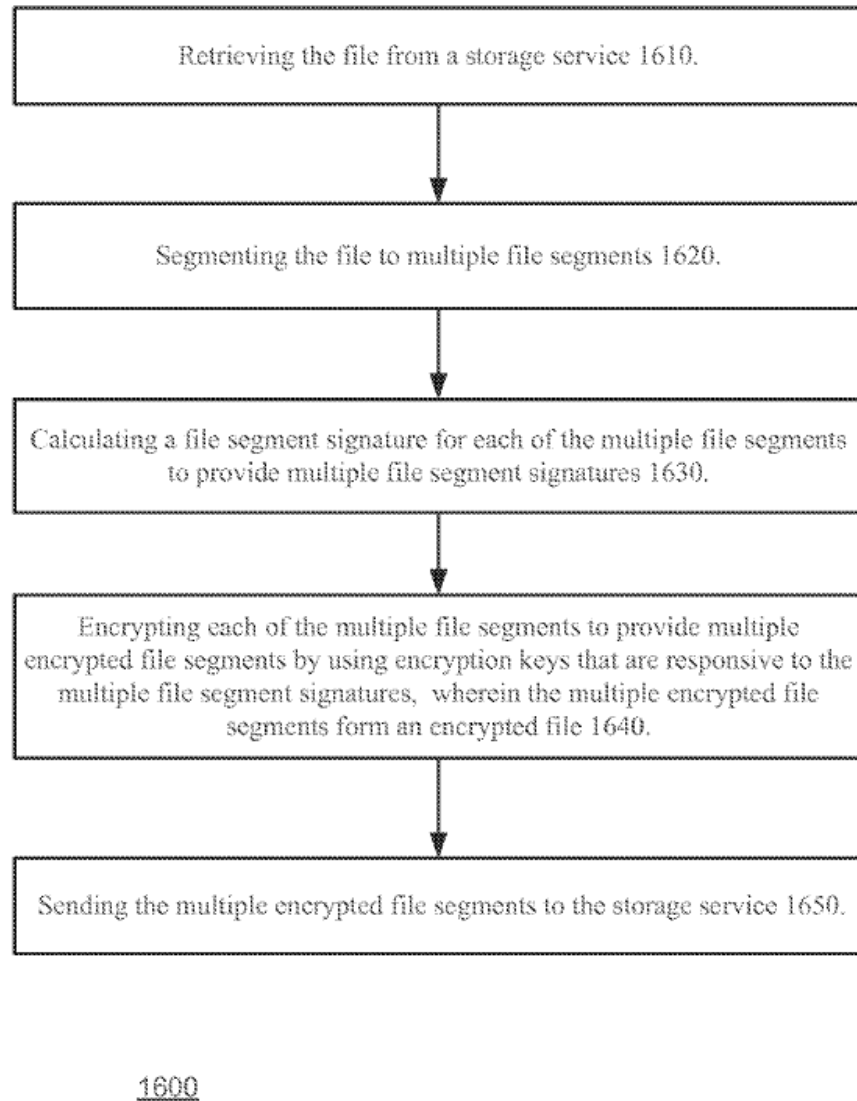


FIG. 15

51. Cidon teaches that its file encryption methods are implemented by a “management server,” which Cidon clearly explains is a “complete software system which is responsible for encrypting files in the cloud and enforcing secure access to these cloud files by providing the right decryption keys to the right users.” *Id.*, ¶102, Fig. 2. The management server includes an “IT Policy Enforcement Module” that

enforces user “access policies” controlling the ability for users to access secure files. *Id.*, ¶108. But, importantly, a POSITA would have understood that Cidon never discloses that these user “access policies” also define cryptographic key information or otherwise control the encryption and storage of files on cloud storage systems.

B. Shikfa (Ex. 1006)

52. Shikfa is primarily directed to a “system and method for enabling searchable encryption of encrypted documents stored by a client on one or more storage providers.” Ex. 1006, Abstract. Shikfa’s system includes a “broker server,” which “transfer[s] the encrypted documents between the client and the one or more storage providers[,] maintain[s] information indicating where the encrypted documents are transferred[,] [and] stores information for at least one encrypted index for the encrypted documents.” *Id.* But, importantly, Shikfa clearly teaches it is the client, not the broker server, that “encrypts the one or more documents” and “generate[s] [the] one or more encrypted indexes.” *Id.*, ¶29. Shikfa teaches that performing these functions at the client “ensures that the broker server 12 cannot access and/or discover the content of the client's data because the broker server 12 does not know the encryption used on the encrypted documents.” *Id.*, ¶48. Shikfa explains that this aspect of Shikfa’s system is important for “maintaining the confidentiality of data external to the client.” *Id.*; *see also* ¶50.

53. Notably, Shikfa discloses that the client may encrypt documents and generate encrypted indexes using any “encryption mechanism known in the art.” *Id.*, ¶29. But, importantly, a POSITA would have understood that Shikfa never discloses performing an encryption process that divides the documents into chunks and creates and configures namespaces for one or more of the chunks.

C. Herrmann (Ex. 1005)

54. Herrmann describes technologies for “anti-virus cooperative enforcement.” Ex. 1005, Abstract, ¶¶48-50. For example, Herrmann explains that “[b]efore ... a client computer[] is permitted to join a network or access protected resources, a check is made to evaluate the antivirus engine and virus definitions that are installed and in use on [the client computer].” *Id.*, ¶50.

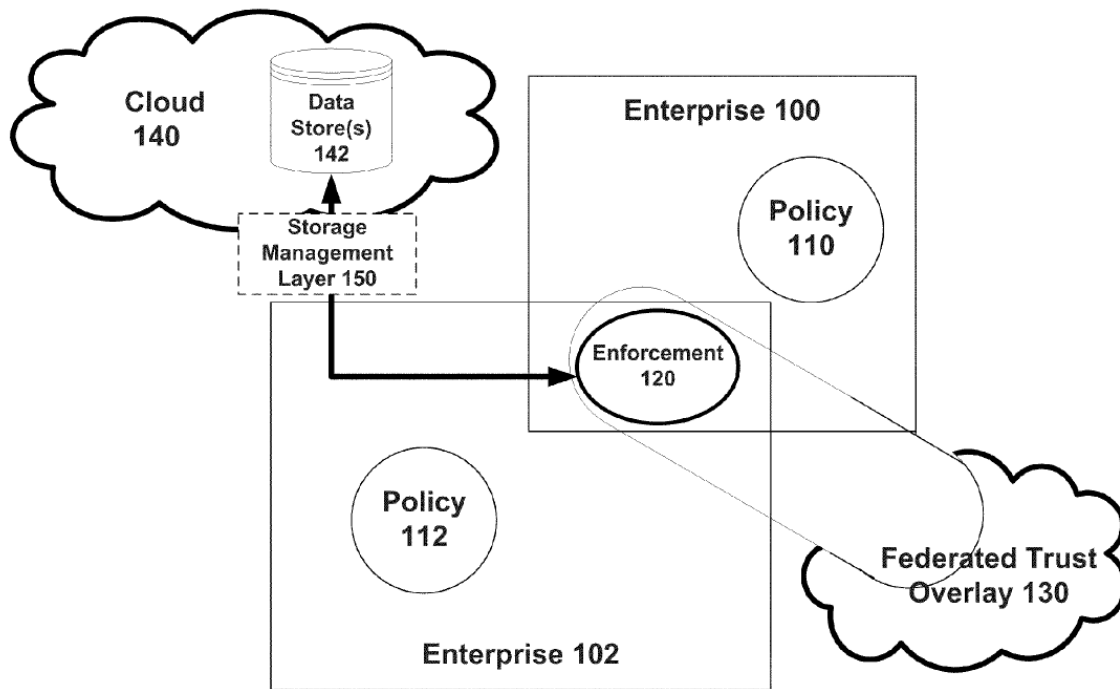
55. Herrmann further explains that “when a client computer requests a VPN connection to a gateway server, the system ... determines whether or not a client computer is using an approved anti-virus application.” *Id.*, ¶50. Herrmann teaches that the system makes this determination through a series of interactions between the client computer and what Herrmann describes as a separate “integrity server.” *Id.*, ¶29; *see also* Figs. 3-4. Only “[a]fter the integrity server 370 has evaluated compliance with anti-virus policies or rules, it notifies the gateway server about whether to permit or restrict access by this particular client device,” and “[t]he gateway server may then decide how much access to permit by this particular client

device (i.e., the gateway server 350 assigns an access policy to the connection/session between the gateway server 350 and the client computer 310) based upon the notification.” *Id.*, ¶77.

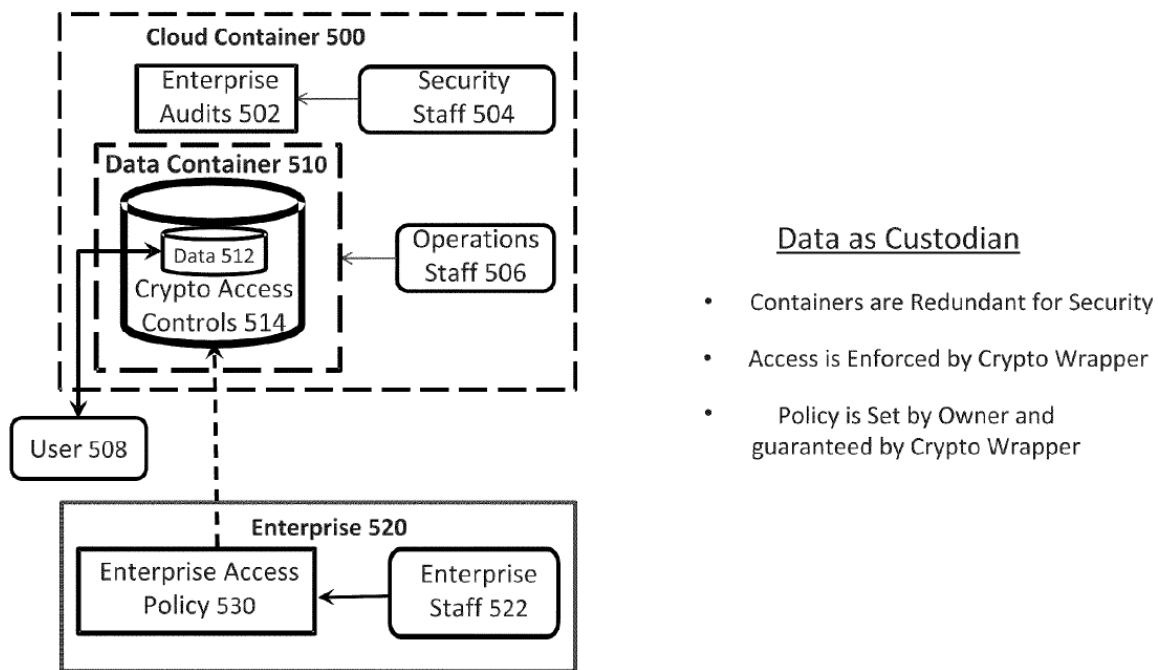
56. Importantly, a POSITA would have understood that Herrmann never discloses using its anti-virus cooperative enforcement system in connection with encrypting, storing, or controlling access to files stored on the cloud. In fact, Herrmann does not reference cloud storage at all, and it appears to be working in a completely different setting than the ’678 patent.

D. Auradkar (Ex. 1009)

57. Auradkar aims to provide a comprehensive “trustworthy platform” in which “data services, including mathematical transformation techniques, such as searchable encryption, deassembling/reassembling or distribution techniques, for data, are provided in a way that distributes trust across multiple entities ... and decouples data protection requirements ... from the containers that hold the data (e.g., file systems, databases, etc.) enabling the data to act as its own custodian.” Ex. 1009, ¶¶9-10. According to Auradkar, Figure 1 below depicts “a general environment for providing one or more embodiments of secure, private and selectively accessible network data services as described herein.” *Id.*, ¶110.

**FIG. 1**

58. Auradkar teaches that Figure 5 below illustrates more specific “aspects of data acting as its own custodian,” wherein “an enterprise 520 has enterprise staff 522 that can are subject to enterprise access policy 530 ... [that] can affect how data 512 stored in a data container 510 of a cloud container 500 can be accessed, manipulated, retrieved, searched, etc.” *Id.*, ¶¶116-117. But, importantly, a POSITA would have understood Auradkar never discloses that these “access polic[ies]” governing “users seek[ing] access to data” also define cryptographic key information or otherwise control the encryption and storage of this data on cloud storage systems.

**FIG. 5****E. Chiueh (Ex. 1011)**

59. Chiueh is primarily directed to a “document processing method and ... system” that is intended to “improve information security of remote storage documents, and facilitate a remote server to deal with various data processing requirements,” including keyword searching. Ex. 1011, ¶7. Chiueh’s system includes a “client [] connected to [a] remote server through a communication network.” *Id.*, ¶9, Fig. 1. Chiueh clearly teaches that before storing a document, the client performs a series of steps to create “encrypted pages” and a “significant word set (SWS),” which “the client transmits ... to [the] remote server for storage.” *Id.*, ¶¶8-9, Fig. 2.

60. Importantly, Chiueh (like Shikfa) clearly teaches that its document processing method is performed by the client, and that Chiueh's method should not be performed by the remote server, so that "the remote server does not have the keys [and it can] not decrypt the document pages and the significant word set." *Id.*, ¶10. Chiueh teaches that this is important because "the remote server is probably not reliable," and if "the remote server ha[d] the decryption capability, the client could not prevent an unfaithful query processing of the remote server," and "the document stored in the remote server could be peeked/leaked." *Id.*, ¶5.

F. Chambers (Ex. 1010)

61. Chambers is primarily directed to "[t]echniques for managing access control policies" that control access to computer networks. Chambers, Abstract. More specifically, Chambers describes "[t]echniques for automatically assigning access control policies (ACPs) based on device types of network client devices (NCDs)." Ex. 1010, ¶21. Chambers explains that "[w]hen a network device first connects to the network, it is examined by the system, identified by type, and assigned an access policy according to the specified rules for that device type." *Id.* This network access policy specifies network parameters, such as "bandwidth limits and traffic shaping rules, VLAN assignment, firewall rules, whether a captive portal should be applied to that device, etc." *Id.*, ¶¶32-40.

62. Chambers teaches that these network access policies are assigned and enforced by “network access devices (NADs) ... managed by a management server.” *Id.*, ¶24; Fig. 1. The NADs “may operate as a gateway device, an access point (AP), a network switch, or a combination thereof.” *Id.* Chambers further explains that these network access policies are initially “downloaded from management server 101 to [the] network access device” under the management server’s control, and then “[w]hen a client device ... is associated with [the] network access device 102, [the NAD’s] access control module 116 controls [network] access of [the] client device 108 by enforcing [the] ACPs.” *Id.*, ¶28, Fig. 6.

63. Importantly, a POSITA would have understood that Chambers (like Herrmann) never discloses using its network access control policy management techniques in connection with encrypting, storing, or controlling access to files on cloud storage systems. In fact Chambers, like Herrmann, does not reference cloud storage at all.

V. CLAIM CONSTRUCTION

64. In my opinion, no claim terms presently require construction. I reserve the right to proffer a construction for a claim term at a later point if I believe it pertinent to any particular dispute.

65. I have interpreted all claim terms as they would have been understood by a POSITA at the time of the priority date of the ’678 patent.

66. However, I do want to provide comment as to one particular term, “namespaces,” which Dr. Lee provides several descriptions and definitions for in his report. I note that these definitions of a namespace are inconsistent with he and the Petition’s theory for how the prior art discloses a namespace for a purported file chunk.

67. In my opinion, and I believe in Dr. Lee’s opinion, a “namespace” has a plain meaning to a POSITA: a set of names available within a particular context. Dr. Lee provides several descriptions and definitions that encompass this plain meaning, which I agree with. For example, Dr. Lee frequently cites to the Dictionary of Computer and Internet Terms, which defines namespace as “the set of names available for naming things.” Ex. 1014; Ex. 1002, ¶¶98, 101, 217. Dr. Lee’s definition encapsulate the idea that a namespace defines the set of allowable names for the objects to which the namespace is applied. Dr. Lee provides an example where “a namespace in a flat structure can simply be a number chosen from within a range (e.g., 0 to $2^{128}-1$).” Ex. 1002, ¶42. Thus, in this example, the namespace defines that a name must be a number and must be within a certain range. If an object is given the name “1234,” that is a valid name within the namespace—although a POSITA would not consider the name “1234” to itself be a namespace.

68. Dr. Lee also states that “[a] namespace ensures that every object within the **context** of the namespace can be identified uniquely by its name(s).” *Id.*, ¶42.

For example, Dr. Lee testified that a directory named “documents” is a namespace for the files within that directory—within that “documents” namespace, files can be identified by simply referring to the file name, such as “foo.” Ex. 2004, 38:18-23. Other contexts, such as other directories, may have a different file also named “foo,” but they belong to a different namespace, a different context. *Id.*, 35:19-36:8; 39:11-40:1.

69. Again, I agree with the above description of a namespace provided by Dr. Lee. I think, from Dr. Lee’s definition, it is quite clear that one should not conflate a namespace with the name of an object (e.g. an object identifier). Instead, to the extent a namespace is applied to object identifiers, it will define ***the set of allowable identifiers*** within a particular context. Thus, in the preceding example, a file has an object identifier (*i.e.* filename) “foo” within the “documents” (*i.e.* the directory it belongs) namespace. Ex. 2004, 38:18-23; Ex. 1002, ¶42. That same file may have a different identifier in a different namespace context—for example, in the root context, that file may have the identifier “/home/wenke/documents/foo,” *i.e.* the identifier contains the full path and filename. Ex. 2004, 37:15-38:8.

70. I further note that the Challenged Claims recite “creating” and “configuring” namespaces for file chunks as part of an encryption process. In other words, the Challenged Claims do not merely recite assigning an identifier to those file chunks, but require that the system, as part of the process of encrypting the file

chunks, create and configure the namespace—*i.e.* the “set of available names” within a particular context. Ex. 1002, ¶¶41-42.

71. The '678 patent provides an example of creating and configuring namespaces as part of an encryption process. For example, the '678 patent teaches that “each chunk can be encrypted based on public and private/secret keys defined by the encryption module 210,” where keys are managed based on user-assigned policies. Ex. 1001, 9:40-41, 11:63-65. These policies therefore create a namespace that defines the naming convention for the file chunks (*e.g.* encryption based on a particular key). The '678 patent further explains that file chunks are stored in folders within the cloud storage service, and “[t]he name of the folder may be the encrypted version of the original file name.” Ex. 1001, 13:2-6. Hence, naming the folder with the encrypted filename and storing the file chunks in that folder creates a namespace scope for the file chunks.

72. And with respect to configuring a namespace for file chunks, the '678 patent describes that encrypted keywords “are stored as filenames of the chunks.” Ex. 1001, 12:9; *see also* cl. 6 (“configuring the namespaces comprises storing content of the file as filenames”). Thus, in this example, the namespace of the file chunks is configured to include the set of specific encrypted keywords as the set of available names and configured to be used as the filename of the file chunk stored

within the cloud storage service. This allows the file (as split into file chunks) to be encrypted in a manner that makes it searchable.

73. In view of this description from the '678 patent, as well as Dr. Lee's own descriptions and definitions of a namespace, I do not agree with Petitioner's contention that the prior art discloses "creating" and "configuring namespaces" as part of an encryption process. As I explain in fuller detail below, the prior art at best discloses that identifiers are assigned to file chunks. The assignment of identifiers, even if within a valid namespace, is not itself the creating or configuring of a namespace. Using another example from Dr. Lee, one could create and configure a namespace that defines the set of allowable names as a number between 0 to $2^{128}-1$. However, assigning an object a valid number within that namespace is not itself creating and configuring the namespace—instead, the namespace was created in advance.

VI. THE PETITION FAILS TO SHOW THAT CIDON, SHIKFA, AND HERRMANN RENDER CLAIMS 1-3, 8-14, AND 16-27 OBVIOUS (GROUND 1)

A. The Petition Fails to Show the References Render Obvious "Creating Namespaces" or "Configuring The Namespaces ... Such That Content of the File is Encrypted in a Manner that Makes it Searchable"

74. Independent claims 1 and 16 both require "creating namespaces for one or more of the plurality of chunks" and then "configuring the namespaces of the one or more chunks such that content of the file is encrypted in a manner that makes it

searchable.” Ex. 1001, claim 1[b][ii] and 16[D]. A file encrypted in this manner is referred to as a “searchable encrypted file.” *E.g., id.*, at 20:4-5, 21:5.

75. In my opinion, Petitioner and Dr. Lee’s arguments that Cidon in view of Shikfa render obvious these limitations fail for several reasons: (1) creating an identifier to assign to a file chunk does not create a namespace, (2) Shikfa’s use of identifiers in an encrypted index is not configuring a namespace, (3) Shikfa’s use of identifiers in an encrypted index has no bearing on the manner in which a file is encrypted, and (4) a POSITA would not be motivated to combine Cidon and Shikfa because Shikfa requires the encryption to be performed on the user device.

1. *Creating an Identifier is Not Creating a Namespace*

76. Petitioner and Dr. Lee assert that Cidon discloses the “creating namespaces” limitation because Cidon “calculat[es] (and **creat[es]**) a ‘unique’ file segment identifier (or file segment signature) for each of the file segments (chunks) based on the content of a file segment using, e.g., SHA2 signature computation.” Pet. 36-37. Petitioner argues that segment identifier “constitute[] ‘namespaces’ because these identifiers were ‘the set of names available for naming’ corresponding file segments consistent with the plain and ordinary meaning of this term.” Pet. 39-40. In other words, it is Petitioner’s position that when Cidon calculates a SHA2 signature to be assigned to a file segment, it has created a namespace for the file chunk.

77. In my opinion, this argument is not consistent with Dr. Lee’s own definition of a namespace, which I agreed with as I explained above in Section V. Under Dr. Lee’s own definition of namespace as “the set of names available for naming,” the creation of a namespace should involve the **creation of the set of available names**. An identifier alone is not the “set of names available for naming” because an identifier is merely a single item within that set—at best, it is a name **within** the namespace. Petitioner does not point to anything in Cidon that states the set of available names—*i.e.* the universe of valid SHA2 signatures—is created or defined during the encryption process. Instead, a POSITA would understand that the set of available names, to the extent there is one, is simply defined by the Cidon reference in advance: SHA2 signatures. Thus, the system does not **create** the namespace during the encryption process because it was created in advance.

78. An example of the mistake Petitioner makes is as follows. The namespace for a URL defines the set of available names for a URL. It requires that the URL contain a protocol, a hostname and a filename. For example, “https://www.google.com/home.html” is a URL where the protocol is “https,” the hostname is “www.google.com” and the filename is “home.html.” However, there was no particular computer or encryption process that **created** this namespace—instead, the namespace was created by the IETF (and other standards bodies) and promulgated by standards documents. When the URL

“https://www.google.com/home.html” is generated to be assigned to an object, the creation of the URL did not create the namespace itself, which already existed (and indeed, had to exist in order to create the URL).

79. Similarly, the namespace for the URL could also be defined as the hostname, *i.e.* “www.google.com,” with the local name being “home.html.” In this example, the namespace “www.google.com” was created when Google registered that domain name and assigned to its web servers. Thus, generating the URL “https://www.google.com/home.html” is not itself creating the namespace, as the namespace has already been created.

80. Thus, and for additional reasons I described above in Section V, an identifier is not a namespace. A namespace defines the set of allowable names within a particular context. Standing alone, an identifier is simply a single item within the set of allowed names—they do not the define the set of allowable names.

2. *Shikfa’s Use of File Identifiers In an Index Does Not Meet “Configuring the Namespaces”*

81. Shikfa teaches the use of an encrypted index that maps extracted keywords to a file pointer (or file identifier). In my opinion, for the reasons explained below, this does not meet the limitation “configuring the namespaces.”

82. Shikfa discloses “generat[ing] one or more indexes 24 for one or more documents that are to be encrypted,” where “[e]ach index 24 includes one or more keywords for the encrypted documents 14 and pointers [or identifiers] to the

encrypted documents 14 containing the keywords.” Ex. 1006, ¶29. Petitioner contends that “Shikfa’s pointers/identifiers were namespaces because they allowed identification of encrypted documents” and that “[t]hese identifiers ... were configured to allow the encrypted documents to be searched” because “they mapped search keywords and encrypted documents.” Pet. 42-43.

83. But as I discussed in the previous section and also above in Section V, an identifier is not a namespace. A namespace defines the set of names within a particular context. Standing alone, an identifier is simply a single item within the set of allowed names. It does not define the set of allowable names.

84. Moreover, in Petitioner’s Cidon-Shikfa combination, the “set of allowed names”—*i.e.* the alleged namespace—is the universe of SHA2 signatures that are used as identifiers for file chunks, as disclosed in Cidon. See Pet. 38-40. However, this namespace is never “configured” by Shikfa’s encryption process. Instead, in the proposed Cidon-Shikfa combination, the SHA2 identifiers are ***used*** (not configured) within Shikfa’s indexes as pointers to file chunks. No configuration of the alleged namespace is performed, because the namespace—*i.e.* SHA2 signatures—was defined in advance.

3. *Shikfa’s Use of File Identifiers In an Index Has No Bearing On the Manner in Which a File is Encrypted*

85. The independent claims also require that the configuration of the namespace is done “such that content of the file is encrypted in a manner that makes

it searchable.” But as I explain below, Shikfa’s use of file identifiers in its encrypted index has no bearing on the manner in which the file itself is encrypted.

86. Shikfa is quite clear that the index is created **before** a document is encrypted. Ex. 1006, ¶29. Also, Shikfa expressly discloses that its documents are encrypted “using any desired encryption mechanism known in the art.” *Id.* Thus, the use of file identifiers simply has no bearing on the manner in which a file is encrypted—which occurs **after** the file identifier has been assigned to the file chunk and which can use any encryption mechanism. Dr. Lee confirmed at his deposition that the creation of the index is “independent” and “orthogonal” to the encryption of the document. Ex. 2004, 136:13-137:7. Thus, even if the identifiers are used in an index for searching an encrypted file, those indexes are completely independent of the file itself. The manner in which the file is encrypted is, as Dr. Lee put it, “orthogonal” to the use of identifiers in Shikfa’s index.

87. In my opinion, this distinction is important. As discussed above, whereas the prior art describes the use of external encrypted indexes, the ’678 Patent describes a novel process where an encrypted index is embedded into the encrypted file by using the encrypted keywords as filenames for the file’s chunks. As a result, queries can now be performed using a cloud storage service’s built-in searching mechanism. Ex. 1001, 10:17-19; Ex. 1003, 40. The encrypted file itself becomes searchable. Indeed, the ’678 Patent explains that the users can “download[] the

searchable encrypted files ..., which would then allow the user 202b to search metadata and/or content associated with the searchable encrypted files even in offline mode when cloud service providers 206 are inaccessible via a network connection.” Ex. 1001, 10:23-28. Such portability is not possible using Shikfa’s teachings, because the encrypted files must be accompanied by the index in order to be searched—the encrypted files themselves are not searchable.

4. *A POSITA Would Not Have Been Motivated to Combine Cidon And Hermann With Shikfa*

88. In my opinion, a POSITA would not be motivated to combine Cidon and Hermann with Shikfa. Petitioner’s theory is that Shikfa’s identifiers would have been implemented in Hermann’s gateway server, which Petitioner reads the claimed “gateway device” on. Pet. at 44-45.

89. I’ll initially note that Shikfa already discloses its own “gateway device,” namely the broker server, which acts as an intermediary between the client and cloud storage providers. See Ex. 1006, ¶28 (“The broker server 12 acts as an intermediary between the client 16 and the cloud storage provider 18...”); ¶¶42-43. Thus, even if a POSITA were to be motivated to implement a gateway device in Cidon’s system, there is no reason he/she would have incorporated Hermann’s gateway server into the modified system (because Shikfa already discloses one).

90. Also, Shikfa is explicit that the client, not the broker server, performs the encryption of the document. See Ex. 1006, ¶¶30, 36. Dr. Lee confirmed that

Shikfa is “pretty clear that the client performs the encryption.” Ex. 2004, 148:22-25. Dr. Lee also confirmed that “the broker doesn’t know the encryption algorithm being used and the key, so they cannot decrypt ... the file content.” *Id.*, 151:21-23.

91. Shikfa explains why it is important that the client performs encryption. In Shikfa’s system, isolating the broker service from the client ensures that only the client can access the content of the client’s data. *Id.*, ¶48 (“The broker system 10 also ensures that the broker server 12 cannot access and/or discover the content of the client’s data because the broker server 12 does not know the encryption used on the encrypted documents 14. Thus, the broker system 10 provides searchable encryption while maintaining the confidentiality of data external to the client 16.”). Dr. Lee confirmed the importance of keeping the encryption on the user device, testifying that if a gateway device (such as a router) has the capability of encrypting data, that makes the data susceptible to a “man in the middle” attack. Ex. 2004, 80:10-82:5. I agree with Dr. Lee that removing encryption from the device imposes this risk.

92. In view of the above, it is my opinion that a POSITA seeking to modify Cidon, which describes performing encryption by a “management server,” would not have been motivated to look to Shikfa, which emphasizes the importance of using the user device to encrypt files (because of the lack of trust of other devices).

93. It is also my opinion that Shikfa teaches away from the claimed invention of a gateway device that performs encryption. I understand that a reference can teach away when a POSITA, after reading the reference, would be discouraged from following the path set out in the reference, or would be led in a direction divergent from the path that was taken in the claim.

94. I think there is ample evidence that a POSITA would be led away from the claimed invention because Shikfa explains why the lack of trust in external devices would necessitate performing encryption on a user's device. Shikfa's teachings are further corroborated by Dr. Lee's testimony, as I already discussed.

C. Cidon Does Not Disclose or Render Obvious a Single, User-Assigned “Policy for Managing Access to and Processing a File To Be Stored on Cloud Platforms”

95. In my opinion, Cidon fails to disclose a single, user-assigned policy that both manages access to a file and manages processing a file.

96. Claim 1 recites ***assigning to one or more users*** “a policy for managing access to and processing a file to be stored on one or more cloud platforms.” Ex. 1001, claim 1[a][i]. Similarly, claim 16 recites implementing a policy ***for a user*** where “the policy defines and manages access to and processing of a file to be uploaded on a cloud.” *Id.*, claim 16[c][i][A]. Both independent claims therefore require a ***user***-assigned policy that (1) manages access to one or more cloud platforms (hereinafter, the “access management” limitation) and (2) processes a file

to be stored on the cloud platforms (hereinafter, the “processing” limitation). Petitioner has failed to show that Cidon discloses such a policy for the reasons below.

1. *Cidon’s Encryption Policy is Assigned to a File, Not a User*

97. Cidon does not disclose that its encryption policy is assigned to a user. It is clear that it is instead assigned to a file, and as confirmed by Dr. Lee, it does not matter which user owns that file.

98. Petitioner relies exclusively on an “encryption policy” disclosed in Cidon’s paragraphs [0224] and [0227]. Pet. 25 (“Cidon’s policy also ‘dictate[d]’ whether certain files were to be encrypted (policy for processing a file).”).

99. But Cidon’s encryption policy is assigned to a “***specific file***,” not a user. See Ex. 1004, ¶[0227] (“[A] ***specific file*** was found to be under the encryption policy as defined by the file ancestry tree. This means that the original parent is encrypted and the policy dictates that all descendent files must be encrypted too.”). The context of Cidon’s disclosure also makes clear that an encryption policy, as part of placement policies, is “***defined on documents***” as a result of a file’s ancestry tree, not assigned to users. *Id.*, ¶[0207] (“Policies can be ***defined on documents*** which are ancestors of other documents. For example inheritance rules can be defined which imply that a policy that is defined on a document ... can be extended to its sub-tree.”).

100. As Dr. Lee explains it, “ancestry information for a file includes which file it was copied from,” thus if a parent file is encrypted, the encryption policy will

require all files that were copied from or modified from that file to be encrypted, too. Ex. 2004, 43:8-20; 61:22-62:11.

101. Assigning an encryption policy to a file is different from assigning an encryption policy to a user because, while a user-assigned policy would allow subjecting a file to different encryption requirements depending on the users, a file-assigned policy would require that certain files be encrypted regardless of who the users are. For example, with user-assigned encryption policies, users who typically create highly sensitive documents could have a policy where their documents are encrypted, which would not be possible with file-assigned encryption policies. Indeed, Cidon is clear that encryption policies assigned to a document are not assigned to a user, because “files in a single ancestry tree may belong to ... different users.” Ex. 1004, ¶[0195]; Ex. 2004, 48:14-17 (“B is a child of A regardless who owns B or who owns A”). Dr. Lee confirmed that Cidon does “not require [parent and child files] belong to the same user.” *Id.*.

102. It is also my opinion that a user-assigned policy would not be obvious in view of teachings that describe file-assigned policies. Again, the concepts are fundamentally different. And I note that neither the Petition nor Dr. Lee never even make such an argument, or in fact, acknowledge the fact that Cidon describes its encryption policy as assigned to files rather than users.

103. Consistent with the claim language, the specification of the ‘678 patent affirms that a policy must be assigned to a user, not a file. For example, the specification describes that “each client 102 is assigned a policy, which is configured to define the rights of the client 102,” and such rights “control the manner in which the respective client 102 encrypts, stores, access, and manages files ...” Ex. 1001, 7:61-8:4. The ‘678 patent also explains that “a policy can be implemented for a client 102 or a group of clients 102, also interchangeably referred to as users 102 hereinafter, based on their role and responsibility in the organization,” and “[t]he selected policy can then be used for defining the manner in which the client 102 interacts with multiple cloud storage providers ...” *Id.*, 8:4-14; *see also id.*, 2:17-21 (“A file storage policy is assigned by the cloud storage gateway device to each user. The assigned file storage policy defines access rights, storage diversity requirements and a type of encryption to be applied to files ***for the corresponding user.***”).

104. Thus, because the encryption policy follows the file (*i.e.* children files must be encrypted if parent files are encrypted) and not the user (*i.e.* the preceding policy applies regardless of which user owns the file), it cannot be a policy that is user-assigned.

2. *Cidon’s “Encryption Policy” Does Not Govern “Processing of a File To Be Uploaded on a Cloud”*

105. Cidon’s encryption policy also does not meet the “processing” limitation because it does not govern the “processing of a file ***to be uploaded*** on a

cloud.” Instead, Cidon’s encryption policy is with respect to files that have already been uploaded.

106. Specifically, Cidon explains that encryption of a file is performed by a “management server” which “encrypts files **on the user Dropbox accounts**,” *i.e.* files already uploaded to the cloud. Ex. 1004, ¶223. Moreover, Cidon explains that encryption occurs after “a specific file was found to be under the encryption policy as defined by the file ancestry tree.” *Id.*, ¶227. Dr. Lee confirmed that encryption does not occur until after “the user uploads an unencrypted file to [a Dropbox] secured folder.” Ex. 2004, 87:4-7.

107. Thus, Cidon is clear that its encryption policy applies to files that have already been uploaded. It therefore does not meet the claimed policy that governs “processing of a file **to be uploaded** on a cloud.”

3. Cidon’s “Encryption Policy” Does Not Govern Access Management

108. As I noted above, the claims require that there be at least one policy that can perform the functions of managing access and managing processing of a file. See claim 1 (“a policy for managing access to **and** processing of a file to be stored on one or more cloud platforms”); claim 16 (“the policy defines and manages access to **and** processing of a file to be uploaded on a cloud”). As such, there has to be at least one policy, assigned to a user (not a file), that governs **both** access management and processing.

109. There is no such policy in Cidon. For the access management limitation, Petitioner relies on Cidon's paragraphs [0102] and [0108], which mention enforcement of an "access policy" as one of the functions performed by a management server. Pet. at 25. But in Cidon's disclosures, an access policy is distinct from an encryption policy. Unlike an encryption policy which is defined by a file's ancestry, access policies are defined by an administrator; an encryption policy is specific to a file, whereas an access policy is assigned for a user or users; and an encryption policy dictates whether a file must be encrypted, whereas an access policy determines if a user can be granted access to a file. Cidon at ¶¶[0108], [0227]. Because Petitioner failed to identify a policy in Cidon that governs (1) managing access to cloud platforms **and** (2) processing of a file to be stored on the cloud platforms, Ground 1 should be rejected.

110. I understand that the Board rejected a similar argument in its Institution Decision, stating that "a policy" need not be limited to a single policy because "[a]s a general rule, the word 'a' or 'an' in a patent claim carry the meaning of 'one or more.'" Paper 10, at 17. I do not disagree that "a" or "an" means "one or more."

111. However, I do not agree that the plain meaning of the claims would allow for multiple policies that where one manages access, another other manages processing of a file, and yet another defines cryptographic key information. Instead, there could certainly be one or more policies, but at least one of the policies should

be able to perform these recited functions. It is my opinion that a POSITA would construe the terms in that way. This is because the claim is clear that there is “a policy” that “defines and manages access to **and** processing of a file,” and that “the policy” (which I understand has an antecedent basis to “a policy”) also defines “cryptographic key information.”

112. The specification of the '678 Patent also clearly states that there is at least a single policy which performs these recited functions. For example, the specification describes that “each client 102 is assigned **a** policy, which is configured to define the rights of the client,” which “control the manner in which the respective client 102 encrypts, stores, accesses, and manages files.” Ex. 1001, 7:65-8:2; see *also id.*, 3:45-67.

D. Cidon Does Not Disclose or Render Obvious “Storing, By The Gateway Device” (Claim 1) or “a Storage Module” For Storing (Claim 16) “The Searchable Encrypted File on The One or More Cloud Platforms Based on the Policy.”

113. Claim 1 requires “storing, by the gateway device, the searchable encrypted file on the one or more cloud platforms based on the policy.” Ex. 1001, claim 1[c]. The term “**the** policy” has antecedent basis to “a policy” recited in limitation 1[a][i]. Similarly, claim 16 requires “a storage module, which, when executed by the one or more processors, stores the searchable encrypted file on the one or more cloud platforms based on **the** policy” (*id.*, claim 1[c][iii]), where the policy refers to the earlier-recited “a policy.” Petitioner asserts that the “storing” /

“storage module” limitation is met because Cidon’s policies included “[p]lacement [of files] in specific directories or shares.” Pet. at 48-49. While Cidon’s placement policies, standing alone, may meet the “storing” / “storage module” limitation, the placement policies are not the same “policy” that Petitioner pointed to for the earlier-recited “a policy.” As discussed above, for the “access management” function of limitation 1[a][i], Petitioner relies on Cidon’s paragraphs [0102] and [0108], which disclose an “access policy” defined for the users. Pet. at 25. The “placement policies” referred to in [0208] and [0209], however, are policies “defined on documents.” Cidon at [0206]-[0207]. In other words, Cidon’s placement policies apply as a result of file-specific properties such as “similarities and ancestry trees” and are not assigned to the users.

114. Moreover, as I discussed above, these placement policies are not the same as the access management policies. Cidon does not disclose a single policy that performs both recited functions. Moreover, as the Board noted in its Institution Decision (Decision at 25), Cidon enumerates the different functions performed by a management server in paragraph [0102], but these functions are distinct from the “placement policies” described in paragraphs [0206]-[0228]. See Cidon at [0102] (describing that a management server may be arranged to (1) provide access to file repositories, (2) store file repository information, (3) analyze and encrypt files, (4) store and retrieve encryption keys, (5) provide access policy configuration for secure

file access, (6) enforce a secure file access policy, and (7) report file access use and devices used for access.”). Cidon does not disclose that a management server performs these functions based on the placement policies. Indeed, unlike the placement policies, Cidon is explicit that access policies are assigned by IT departments for different users (see Cidon at [0108]), further undermining that Cidon discloses a policy that governs **both** access management and storage of an encrypted file on the cloud.

E. Cidon Does Not Disclose or Render Obvious Storing a File “Across Two Or More Cloud Platforms” As Recited in Dependent Claims 11 and 25

115. Dependent Claims 11 and 25 require that the encrypted file “is stored across two more cloud platforms.” Petitioner fails to establish that Cidon disclosure of “archiving” and “backup” meets this limitation because archiving/backup is merely copying of a file to another location, as opposed to storing a file **across** multiple locations.

116. Consistent with the specification, in my opinion, the plain meaning of storing “across” multiple platforms is that a single file has different parts stored on different platforms. For example, the ’678 Patent explains that “Clients may also store such content across two or more stores 114 such that the content **is divided into multiple chunks and each chunk is stored on a different cloud store** 114.” Ex.

1001, 7:43-45. This clearly teaches that a file is stored across services when its constituent parts are stored on different services.

117. Petitioner relies on Cidon's disclosure of "archiving and backup to other storage services." Ex. 1004, ¶174. As an initial matter, Cidon's disclosure of "archiving and backup" is with respect to "DNA extraction," not encrypted documents. *Id.*

118. But in any case, it is my opinion that archiving and backup of a file does not constitute splitting up a file into chunks and storing chunks on different platforms. Instead, archiving and backup would be understood to mean merely storing a **copy** of the file to a different store. Dr. Lee confirmed this, stating that "[b]ackup normally means that I ... put a **copy** of ... your file in some kind of backup." Ex. 2004, 107:24-108:1. By storing a **copy**, this allows recovery of a lost or corrupt file. *Id.* Dr. Lee also confirmed that archiving and backup are synonymous. *Id.*, 108:5-23. Thus, archiving/backup is materially different than storing a file across multiple services.

F. Cidon Does Not Disclose or Render Obvious a Policy That Defines "One or More Authorized Locations to Which The File May be Stored" as Claimed by Dependent Claim 14

119. Claim 14 requires that "the policy" of claim 1 define, among other things, "one or more authorized locations to which the file may be stored." Ex. 1001, claim 14. Petitioner asserts that "[a] POSITA would have found it obvious that the

file locations would be part of Cidon’s security policy that could also define (and manage) such location information.” Pet. at 64. Petitioner does not identify what “Cidon’s security policy” refers to. Petitioner relies on Cidon’s paragraph [0364]. That paragraph, however, simply discloses moving an encrypted file to another location, and Cidon does not disclose that such relocation is done based on any “policy.” Cidon at ¶364 (“The modifying action (2130) may include ... generating the modified file to comprise access information to another location in which the modified file is to be stored, and storing the modified file at the other location . . .”).

120. Further, as discussed above for claim 1, Petitioner does not identify any policy based on which the claimed access management and processing of a file to be stored are performed. Petitioner’s argument for claim 14 fails for the additional reason that Petitioner does not point to any policy based on which the functions recited in claim 1 *and* claim 14 are performed.

VII. GROUNDS 2 AND 3 FAIL FOR THE SAME REASONS THAT GROUND 1 FAILS

121. Grounds 2 and 3 of the petition challenge dependent claims that depend from claims 1 or 16. Thus, they rely on the analysis of Ground 1. In my opinion, Grounds 2 and 4 fail for the same reasons that I discussed above in connection with Ground 1.

VIII. THE PETITION FAILS TO SHOW THAT AURADKAR, CHIUEH, AND CHAMBERS RENDER CLAIMS 1-3, 8-19, AND 21-27 OBVIOUS (GROUND 4)

A. Auradkar In View of Chiueh Does Not Render Obvious “Creating Namespaces” or “Configuring The Namespaces ... Such That Content of The File is Encrypted in a Manner That Makes it Searchable”

122. In my opinion, Petitioner’s arguments Auradkar in view of Chiueh render obvious these limitations fail for at least the following reasons: (1) creating an identifier to assign to a file chunk does not create a namespace, (2) Chiueh’s use of identifiers in an encrypted index is not configuring a namespace, (3) Chiueh’s use of identifiers in an encrypted index has no bearing on the manner in which a file is encrypted, and (4) a POSITA would not be motivated to combine Auradkar’s server-based encryption with Chiueh’s device-based encryption.

1. *Creating an Identifier is Not Creating a Namespace*

123. As I discussed with the Cidon-Shikfa combination, the Auradkar-Chiueh combination likewise fails because creating an identifier to assign to a file chunk does not meet the claimed “creating namespaces” because an identifier is not a namespace.

124. Petitioner asserts that Auradkar discloses the “creating namespaces” limitation because Auradkar discloses “creating tags for payloads (e.g., payload 1200, XML fragment 1902 (chunks)), such as tag 1200 that included the filename of the payload (FName: fox.tx) and XML tags 1900, and encrypting them.” Pet. 94-

95. Petitioner argues that “[a] POSITA would have found it obvious that these tags constituted namespaces for the payloads (e.g., file 1202), because, consistent with the plain and ordinary meaning of a namespace, they were a ‘set of names available for naming things’ such as files or segments.” Pet. 96.

125. Petitioner also provides an alternative theory using Chiueh, arguing that “Chiueh’s system also created and assigned a document identification or ID (namespace) to each of the encrypted pages,” the document ID being a “combination of a filename and a page offset in the document.” Pet. 96-97.

126. Under Petitioner’s own definition of namespace as “the set of names available for naming things,” the creation of a namespace must therefore involve the **creation of the set of available names**. A filename is not the “set of names available for naming things” because, at best, an identifier like a filename is merely a single item within the namespace. Dr. Lee even agrees with that characterization, repeatedly calling the filename a “local name” that is “**within** the namespace,” whereas it is the directory containing the file that is the namespace (since it provides the context of the filename). Ex. 1002, ¶42; Ex. 2004, 37:8-38:12.

127. Likewise, a group of filenames is also not the “set of names available for naming” because the group of filenames is still merely a group of items within a larger set of available names—the creation of the identifiers within the namespace is not a creation of the namespace itself.

128. Petitioner does not point to anything in Auradkar or Chiueh that states the set of available names—*i.e.* the universe of valid filenames (or filenames and page offsets)—is created or defined during the encryption process. Instead, it is quite clear that the set of available names—valid identifiers—is simply defined by the Auradkar and Chiueh references in advance. Thus, the system does not **create** the namespace during the encryption process because it was created in advance.

129. Moreover, as I discussed above in Section V, *supra*, an identifier is not a namespace. A namespace defines the set of allowable names within a particular context. Standing alone, an identifier is, at best, simply a single name within the set of allowed names. For example, the Petition relies on the file name “fox.tx” as meeting the claimed namespace. Pet. 95As Dr. Lee explained it, different files that are both named “fox.tx” can exist within different namespaces, such as /home/wei or /home/wenke. Ex. 2004, 35:10-36:8.

2. *Chiueh’s Mapping of Keywords to Document Identifiers In an Index Is Not “Configuring The Namespaces”*

130. Chiueh’s teaching of an encrypted index that maps keywords document identifiers does not meet the limitation “configuring the namespaces” because identifiers are not namespaces.

131. Chiueh discloses generating an “SWS” (*i.e.* an index) that maps keywords extracted from an unencrypted document to a document identifier that “may include a file name of the document and a page offset in the document.” Ex.

1011, ¶¶21-23, 40. Petitioner contends that “[a] POSITA would have found it obvious that Chiueh’s system configured document IDs (namespaces) of the encrypted pages (chunks) because it mapped those IDs (indicated in the index information) to keywords appearing in the corresponding pages in a global search index (GSI), such that the page content in encrypted form was searchable.” Pet. 105-06.

132. But as I discussed above in Section V, *supra*, an identifier is not a namespace. A namespace defines the set of names in a particular context. Standing alone, an identifier is simply a single item or name within the set of allowed names, not the namespace itself.

133. Moreover, in Petitioner’s Auradkar-Chiueh combination, the “set of allowed names”—*i.e.* the alleged namespace—is the universe of filenames and page offsets, as disclosed in Chiueh. However, this namespace is never “configured” by Chiueh’s system. Instead, in the proposed Auradkar-Chiueh combination, the document identifiers (filenames) are **used** (not configured) within Chiueh’s indexes as index information in order to locate a document page. No configuration of the alleged namespace is performed, because the alleged namespace—*i.e.* the universe of valid filenames and page offsets—was defined in advance by the reference.

3. *Chiueh's Mapping of Keywords to Identifiers In an Index Has No Bearing On the Manner in Which a File is Encrypted*

134. In addition, Chiueh's use of document identifiers within an encrypted index also does not involve a configuration "such that content of the file is encrypted in a manner that makes it searchable."

135. Chiueh's use of document identifiers within an index has no bearing on or relationship to the manner in which a file is encrypted. Chiueh is quite clear that its index is created **before** a document is encrypted. Ex. 1011, ¶21 ("the client 110 removes a part of words from the document pages that **are still not encrypted**"). Moreover, Dr. Lee confirmed at his deposition that the creation of the index is "orthogonal" to the encryption of the document. Ex. 2004, 184:24-185:8. Thus, even the identifiers are used in an index for searching an encrypted file, those indexes are completely independent of the file itself. The manner in which the file is encrypted is, as Dr. Lee put it, "orthogonal" to the use of document identifiers in Chiueh's index.

136. In my opinion, this distinction is important. As I discussed above, whereas the prior art describes the use of external encrypted indexes, the '678 Patent describes a novel process where an encrypted index is embedded into the encrypted file by using the encrypted keywords as filenames for the file's chunks. As a result, queries can now be performed using a cloud storage service's built-in searching mechanism. Ex. 1001, 10:17-19; Ex. 1003, 40. The encrypted file itself becomes

searchable. Indeed, the '678 Patent explains that the users can “download[] the searchable encrypted files ..., which would then allow the user 202b to search metadata and/or content associated with the searchable encrypted files even in offline mode when cloud service providers 206 are inaccessible via a network connection.” Ex. 1001, 10:23-28. Such portability is not possible using Chiueh’s teachings, because the encrypted document pages must be accompanied by the index in order to be searched—the encrypted document pages themselves are not searchable.

4. *A POSITA Would Not Have Been Motivated to Combine Auradkar and Chambers and Chiueh*

137. Petitioner failed to demonstrate a motivation to combine Auradkar and Chambers with Chiueh. Petitioner asserts that Chiueh’s encryption techniques would be incorporated into the Auradkar and Chambers combination. However, Chiueh is explicit that the user client device performs the encryption of the document pages and the keywords. See Ex. 1004, ¶¶20-24.

138. I note that the Board has rejected a similar argument because “Chiueh does not counsel against encrypting document pages and word sets at an enterprise system of Auradkar or at an intermediate gateway of the Auradkar-Chambers combination.” Paper 10, 44. In my opinion, this is not true. Chiueh explains why it is important the client device performs encryption (and decryption) instead of the server: because “the remote server is probably not reliable” as “the document stored

in the remote server could be peeked/leaked” if the remote server is given decryption capabilities. Ex. 1004, ¶5.

139. In Chiueh’s system, isolating the remote server from the client ensures that only the client can access the content of the client’s data. As explained by Chiueh, when a user wishes to query an encrypted index, the remote server only receives an “encrypted keyword,” searches the encrypted index using the encrypted keyword, and “transmits the targeted encrypted page.” Ex. 1004, ¶25. Thus, Chiueh’s system where the user device encrypts the document is contrary to the claim requirement that encryption of the file be performed “by the gateway device.” Dr. Lee likewise confirmed the importance of keeping the encryption on the user device, testifying that if a gateway device (such as a router) has the capability of encrypting data, that makes the data susceptible to a “man in the middle” attack. Ex. 2004, 80:10-82:5.

140. Thus, a POSITA seeking to modify Auradkar in view of Chambers, which Petitioner asserts performs encryption on a remote gateway device, would not have been motivated to look to Chiueh, which emphasizes the importance of using the user device to encrypt files (because of the lack of trust of other devices).

141. Also, it is my opinion that Chiueh teaches away from the claimed invention of a gateway device that performs encryption. I think it is clear that a POSITA would be led away from the claimed invention because Chiueh explains

why the lack of trust in external devices would necessitate performing encryption on a user's device. Chiueh's teachings are further corroborated by Dr. Lee's testimony of dangerous man-in-the-middle attacks if a gateway device is able to encrypt and decrypt transmissions.

B. Petitioner Fails to Show Auradkar Discloses “Cryptographic Key Information Defined by the Policy”

142. Claims 1 and 16 recite “encrypting ... using cryptographic key information defined by the policy.” Petitioner has failed to show any policy that defines cryptographic key information.

143. The '678 Patent describes that “a selected policy can be used to define encryption keys, decryption keys, and encryption type, among other attributes for carrying out the encryption of data.” Ex. 1001, 4:4-7. A POSITA would therefore understand that “cryptographic key information” at least includes defining the encryption and/or decryption keys and encryption type.

144. Petitioner argues that Auradkar discloses this limitation because it discloses that “a policy could be created that requires that (sensitive) medical data must be backed up once a day using AES 256 encryption to a secure server in a trusted data center.” Ex. 1009, ¶96. However, there are several problems with Petitioner's argument, in my opinion.

145. Auradkar states that this policy is with respect to a “paragraph's classification,” *i.e.* the secrecy classification of a paragraph within a document. *Id.*,

¶95. But Auradkar does not state that such a policy is the same Enterprise Access Policy that Petitioner relies on for the antecedent basis of “policy.” *See* Pet. 80-81. The antecedent basis for “the policy” that defines cryptographic key information is “a policy for managing access to and processing a file to be stored.” There is no assertion, nor any teaching in Auradkar they are the same policies, which is required as I stated above.

C. Petitioner Fails to Show the References Render Obvious A Policy for “Processing Of a File To Be Uploaded on a Cloud”

146. Auradkar’s policy also does not meet the “processing” limitation of claims 1 and 16 because it does not govern the “processing of a file *to be uploaded* on a cloud.” Instead, Auradkar’s policy is with respect to files that have already been uploaded to a cloud service.

147. Specifically, Auradkar explains that its policy “affect[s] how data 512 *stored in a data container 510 of a cloud container 500* [could] be accessed, manipulated, retrieved, searched.” Ex. 1009, ¶117. Indeed, Auradkar is explicit that access to the data is controlled by “crypto access controls” that reside within the cloud container—these controls are “guided by, but separated from, enterprise access policy 530” and “protect the data 512 from unwarranted access by users 508.” *Id.*; Fig. 5. As a result, Auradkar’s policy only governs data that has already been uploaded—it cannot affect data that is “*to be uploaded* on a cloud” because the crypto access controls that implement the policy only reside on the cloud server.

IX. GROUND 5 FAILS FOR THE SAME REASONS THAT GROUND 4 FAILS

148. Ground 5 of the petition challenges dependent claims that depend from claims 1 or 16. Thus, it relies on the analysis of Ground 4. In my opinion, Ground 5 fails for the same reasons that I discussed above in connection with Ground 4.

X. GROUND 5 ALSO FAILS BECAUSE INOUE DOES NOT DISCLOSE “STORING CONTENT OF THE FILES AS FILENAMES”

149. Ground 5 also fails because Inoue does not disclose or render obvious claim 6’s “wherein the step of configuring the namespaces comprises storing content of the file as filenames.” Claim 6 depends on claim 1’s “configuring the namespaces of the one or more chunks such that content of the file is encrypted in a manner that makes it searchable.” Specifically, Inoue does not teach that a filename stores *content of the file* nor does Inoue teach that the content of the file becomes *searchable*.

150. Inoue is a system where a user can label an image by providing a voice input with keywords. Specifically, Inoue teaches a method where an image file is given a filename using *one* keyword extracted from a *voice input*. Ex. 1012, ¶101 (“In the voice recognition processing, one or more keywords (words) can be extracted *from the input voice data* using a word spotting voice recognition technology.”); ¶104 (“[T]he voice information setting section 707 ... sets *one* of the keywords as the title (a filename) of the image data.”). The filename does not store

content of the file (*i.e.* the image) because the keyword comes from a voice input, not the image itself. As a result, because the keyword does not come from the content of the file (and indeed, could be any arbitrary word provided by the speaker), using Inoue's method does not result in making the ***content of the file*** searchable.

151. Thus, in my opinion, Petitioner's assertion that "[a] POSITA would have been motivated to incorporate Inoue's functionality of assigning a keyword extracted from file content ... into the Auradkar-Chiueh-Chambers system" is based on a false predicate. Inoue teaches assigning to a filename a keyword extracted from a ***voice input***, not from the file itself.

XI. CONCLUSION

152. In signing this declaration, I recognize that the declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also recognize that I may be subject to cross-examination in the case and that cross-examination will take place within the United States. If cross-examination is required of me, I will appear for cross-examination within the United States during the time allotted for cross-examination.

153. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on the information and belief are believed to be true; and further that these statements were made with the knowledge that willful

false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Executed: December 22, 2023

A handwritten signature in black ink, appearing to read 'J. Black Jr.', written over a horizontal line.

John Black Jr., Ph.D.