

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SAMSUNG ELECTRONICS CO., LTD., and
SAMSUNG ELECTRONICS, AMERICA, INC.,

Petitioners

v.

TRUESIGHT COMMUNICATIONS LLC,

Patent Owner

Case IPR2025-00123

U.S. Patent No. 8,745,749

Declaration Of Erez Zadok, Ph.D.
In Support Of Petition for *Inter Partes* Review of
U.S. Patent No. 8,745,749

Table of Contents

I.	ASSIGNMENT.....	1
II.	BACKGROUND AND QUALIFICATIONS.....	2
III.	MATERIALS CONSIDERED.....	12
IV.	LEVEL OF ORDINARY SKILL IN THE ART.....	15
V.	RELEVANT LEGAL STANDARDS.....	16
VI.	SUMMARY OF OPINIONS.....	20
VII.	TECHNOLOGY OVERVIEW.....	21
	A. General Computer Operations.....	21
	B. Storage Devices and Media.....	27
	1. SD Cards vs. Smart Cards.....	29
	C. Files and File Systems.....	36
	D. Data Security and Privacy.....	40
	1. Hash Functions and Their Uses.....	47
	E. Virtualization and its Uses in Storage and File Systems.....	51
VIII.	OVERVIEW OF THE '749 PATENT.....	57
	A. The '749 Specification.....	57
	B. Prosecution History.....	65
IX.	THE CLAIMS.....	66
X.	OVERVIEW OF THE PRIMARY PRIOR ART REFERENCES.....	72
	A. Maeda.....	73

B.	Hamann.....	80
C.	Applicant Admitted Prior Art (AAPA)	82
D.	Hirota.....	84
E.	Lu.....	87
F.	Matsukawa.....	88
XI.	CLAIM CONSTRUCTION	89
A.	“File System”	89
XII.	DETAILED EXPLANATION OF GROUNDS.....	92
A.	Ground 1: Maeda In View Hamann And the AAPA Renders Claims 1, 14-16, 18, and 22-24 Obvious	92
1.	Claim 1	92
a.	[1pre] “A method for generating a virtual secure digital (SD) card, the method comprising:”	92
b.	[1a] “reading a media key block and media identification for an SD card”	94
c.	[1b] “storing the media key block and the media identification”	98
d.	[1c] “creating a file system for secure data on a storage device for storage of secure data corresponding to a secure data area of the SD card; and”	101
e.	[1d] “creating a file system for user data on the storage device for storage of user data corresponding to a user data area of the SD card”	106
f.	[1e] “wherein a size of the file system for secure data is determinant on a size of the file system for user data, the determination based on a lookup table providing a relationship between the sizes.”	110

2.	Motivation To Combine Maeda, the AAPA, and Hamann	117
3.	Claim 14	125
a.	[14pre] “A method using a virtual secure digital (SD) card, the method comprising:”	125
b.	[14a] “determining if the virtual SD card is provisioned;”	126
c.	[14b] “provisioning the virtual SD card in response to the virtual SD card not being provisioned, the provisioning including:”	130
d.	[14c] “creating a file system for secure data on a storage device for storage of secure data corresponding to a secure data area of the SD card; and”	136
e.	[14d] “creating a file system for user data on the storage device for storage of user data corresponding to a user data area of the SD card,”	136
f.	[14e] “wherein a size of the file system for secure data is determinant on a size of the file system for user data, the determination based on a lookup table providing a relationship between the sizes;”	137
g.	[14f] “accessing the data stored in the secure area of the provisioned virtual SD card; and”	137
h.	[14g] “extracting the data from the secured area of the provisioned virtual SD card.”	139
4.	Claim 15: “The method of claim 14, further comprising performing an authentication using Authenticated Key Encryption (AKE) before accessing the data in the secure area of the provisioned virtual SD card.”	141
5.	Claim 16: “The method of claim 14 or 15, wherein data received from an application program through Authenticated Key Encryption (AKE) is first decrypted.”	144

6.	Claim 18	149
a.	[18pre] “A non-transitory computer readable medium configured to store instructions that, when executed, cause at least one processor to:”	149
b.	[18a] “read a media key block and media identification for an SD card;”	150
c.	[18b] “store the media key block and the media identification;”	150
d.	[18c] “create a file system for secure data on a storage device for storage of secure data corresponding to a secure data area of the SD card; and”	151
e.	[18d] “create a file system for user data on the storage device for storage of user data corresponding to a user data area of the SD card;”	151
f.	[18e] “wherein a size of the file system for secure data is determinant on a size of the file system for user data, the determination based on a lookup table providing a relationship between the sizes.”	151
7.	Claim 22	152
a.	[22pre] “A non-transitory computer readable medium configured to store instructions that, when executed, cause at least one processor to:”	152
b.	[22a] “determine if a virtual SD card is provisioned;”	152
c.	[22b] “provision the virtual SD card in response to the virtual SD card not being provisioned, wherein to provision the virtual SD card includes to:”	152
d.	[22c] “create a file system for secure data on a storage device for storage of secure data corresponding to a secure data area of the SD card; and”	153

e.	[22d] “create a file system for user data on the storage device for storage of user data corresponding to a user data area of the SD card,”	153
f.	[22e] “wherein a size of the file system for secure data is determinant on a size of the file system for user data, the determination based on a lookup table providing a relationship between the sizes,”	153
g.	[22f] “access the data stored in the secure area of the provisioned virtual SD card; and”	153
h.	[22g] “extract the data from the secured area of the provisioned virtual SD card.”	154
8.	Claim 23 “The computer readable medium of claim 22, further comprising: performing an authentication using Authenticated Key Encryption (AKE) before accessing the data in the secure area of the provisioned virtual SD card.”	154
9.	Claim 24 “The computer readable medium of claim 23, wherein data received from an application program through Authenticated Key Encryption (AKE) is first decrypted.”	154
B.	Ground 2: Maeda In View Hamann, the AAPA, and Hirota Renders Claims 2, 6-8, and 19 Obvious.....	154
1.	Claim 2: “The method of claim 1, wherein the file system for secure data is encrypted.”	154
2.	Motivation To Combine Maeda, Hamann, the AAPA, and Hirota.....	160
3.	Claim 6: “The method of claim 2, wherein the file system for secure data and the file system for user data are located on different physical partitions of the storage device.”	161
4.	Claim 7: “The method of claim 2, wherein the file system for secure data and the file system for user data are located on different logical partitions of the storage devices.”	165

5.	Claim 8: “The method of claim 2, wherein the file system for secure data and the file system for user data are within an existing file system.”	167
6.	Claim 19: “The computer readable medium of claim 18, wherein the file system for secure data is encrypted.”	168
C.	Ground 3: Maeda In View Hamann, the AAPA, Hirota and Lu Renders Claims 3, 4, 20, and 21 Obvious	168
1.	Claim 3: “The method of claim 2, wherein the file system is encrypted with an encryption key, the encryption key derived from computer system information and the encryption key not stored by the computer system.”	168
2.	Motivation To Combine Maeda, Hamann, the AAPA, Hirota, and Lu.....	173
3.	Claim 4: “The method of claim 2, wherein the file system is encrypted with an encryption key, the encryption key derived from a computer system and stored by the computer system.”	174
4.	Claim 5: “The method of claim 2, wherein the file system is encrypted with an encryption key, the encryption key derived from a computer system and stored by the computer system within a trusted host.”	175
5.	Claim 17: “The method of claim 14, wherein data received from an application program through Authenticated Key Encryption (AKE) is encrypted with another key before storing the data in the secure area of the provisioned virtual SD card.”	178
6.	Claim 20: “The computer readable medium of claim 19, wherein an encryption key is derived from a computer system information, the encryption key not stored by the computer system.”	182
7.	Claim 21: “The computer readable medium of any one of claims 18, 19, or 20, wherein the encryption key is derived	

	from a computer system and stored by the computer system.”	
		182
8.	Claim 25: “The computer readable medium of claim 23, wherein data received from an application program through Authenticated Key Encryption (AKE) is encrypted with another key before storing the data in the secure area of the provisioned virtual SD card.”	183
D.	Ground 4: Maeda In View Hamann, the AAPA And Matsukawa Renders Claim 13 Obvious	183
1.	Claim 13: “The method of any one of claims 1 through 5, wherein the media identification comprises random data not allocated from the SD card.”	183
2.	Motivation To Combine Maeda, Hamann, the AAPA, and Matsukawa	184
XIII.	SECONDARY CONSIDERATIONS	185
XIV.	CONCLUSION.....	185

I, Erez Zadok, Ph.D., declare that:

I. ASSIGNMENT

1. I have been retained by Samsung Electronics Co., Ltd., Samsung Electronics America, Inc. (together “Samsung or “Petitioners”) as an independent expert consultant in this proceeding before the United States Patent and Trademark Office (“PTO”).

2. My consulting company, Zadoks Consulting, LLC, is being compensated for my time at my standard consulting rate. I am also being reimbursed for expenses that I may incur during the course of this work.

3. My compensation is in no way contingent on the nature of my findings, the presentation of my findings in testimony, or the outcome of this or any other proceeding. I have no other interest in this proceeding.

4. I have been asked to consider whether certain references disclose or suggest the features recited in the claims of U.S. Patent No. 8,745,749 (“the ’749 patent”) (EX1001).¹ My opinions are set forth below.

¹ Where appropriate, I refer to exhibits I understand will be attached to the petition for *inter partes* review of the ’749 patent (the “Petition”).

II. BACKGROUND AND QUALIFICATIONS

5. I am a Professor in the Computer Science Department at Stony Brook University (part of the State University of New York (“SUNY”) system). I direct the File-systems and Storage Lab (FSL) at Stony Brook’s Computer Science Department. My research interests include file systems and storage systems, operating systems, transactional systems including database technologies, information technology and system administration, security/privacy and information assurance, networking, energy efficiency, performance and benchmarking, virtualization, cloud systems, compilers, applied machine learning, and software engineering.

6. I studied at a professional high school in Israel, focusing on electrical engineering (“EE”), and graduated in 1982. I spent one more year at the high school’s college division, receiving a special Certified Technician’s degree in EE. I then went on to serve in the Israeli Defense Forces for three years (1983–1986). I received my Bachelor of Science degree in computer science (“CS”) in 1991, my Master’s degree in CS in 1994, and my Ph.D. in CS in 2001—all from Columbia University in New York.

7. When I began my undergraduate studies at Columbia University, I also started working as a student assistant in the various campus-wide computer labs, eventually becoming an assistant to the head labs manager, who was managing all

public computer labs on campus. During that time, I also became more involved with research within the CS Department at Columbia University, conducting research on operating systems, file and storage systems, distributed and networked systems, security, and other topics. I also assisted the CS department's computer administrators in managing the department's computers, which included storage, IT, networking, and cyber-security related duties.

8. In 1991, I joined Columbia University's CS department as a full-time systems administrator, studying towards my MS degree part-time. My MS thesis topic is related to file system reliability, fault tolerance, replication, and failover in mobile networked storage systems using file virtualization. My main duties as a systems administrator involved installing, configuring, and managing many networked servers, proxies, and desktops running several operating systems, as well as network devices setup; this included many software and hardware upgrades, device upgrades, and BIOS firmware/chipset updates/upgrades. My duties also included ensuring reliable, secure, authenticated access to networked systems/storage and licensed software, as well as software updates, security and bug fixes. Examples of servers and their protocols included email (SMTP), file transfer (FTP), domain names (DNS), network file systems (NFS), network news systems (NNTP), and Web (HTTP).

9. In 1994, I left my systems administrator position to pursue my doctoral studies at Columbia University. My Ph.D. thesis topic was on versatile file system development using stackable (virtualized) file systems, with examples in the fields of security and encryption, efficiency, reliability, and failover. I continued to work part-time as a systems administrator at the CS department, and eventually I was asked to serve as manager to the entire information technology (“IT”) staff. From 1991 to 2001, I was also a member of the faculty-level Facilities Committee that oversaw all IT operations at the CS department.

10. As part of my Ph.D. studies at Columbia, I collaborated on projects to develop advanced AI-like techniques to detect previously unknown viruses (a.k.a. “zero-day malware”), using data mining and rule-based detection. This work led to several highly cited papers (over 1,600 citations for one of the papers alone), and two patents. I also became a Teaching Assistant (TA) for a first-ever Computer Security course given at Columbia University’s CS department with Dr. Matt Blaze as instructor.

11. From 1990 to 1998, I consulted for SOS Corporation and HydraWEB Technologies, as a systems administrator and programmer, managing data storage use and backup/restore duties, databases, web servers, as well as information assurance and cyber-security (*e.g.*, malware protection, software licensing). From 1994 to 2000, I led projects at HydraWEB Technologies, and then became the

Director of Software Development—overseeing the development of several products and appliances such as stateful firewalls and HTTP load-balancers, utilizing network-virtualization and high-availability techniques. From 2009 to 2019, I have consulted for Packet General Networks, a startup specializing in secure, virtualized, network storage and applications' data security in the cloud.

12. In 2001, I joined the faculty of Stony Brook University, a position I have held since that time. In 2002, I joined the Operations Committee, which oversees the IT operations of the CS department at Stony Brook University. From 2006 to 2010, I was the Director of IT Operations of the CS department. My day-to-day duties included setting policies regarding computing, hiring and training new staff, assisting any staff with topics of my specialty, defining requirements for new software/hardware, and purchasing. From 2010 to 2015, I had served as the Co-Chair to the Operations Committee. From 2016 to 2019, I oversaw the IT Operations as the Chair of the Operations Committee. A significant component of these duties included defining and helping implement policies for data management, to ensure the security of users and their data, and data reliability and availability, while minimizing the inconvenience and performance impact to users. I personally helped setup and maintain an initial virtual-host infrastructure in the department. Since late 2019, I've been a member of the department's Executive Committee that also oversees all IT operations.

13. In 2017, I became the department's Graduate Academic Adviser, advising all Master students (over 400 annually on average) and many other graduate students on an assortment of academic matters. In August 2024, I took over as the department's Graduate Program Director, overseeing the entire graduate CS program (700-800 students annually on average).

14. Since 2001, I have personally configured and managed my own research lab's network. This includes setting up and configuring multiple storage systems (*e.g.*, NFS, CIFS/SMB, NAS), virtual and physical environments, applications such as database (*e.g.*, MySQL, Postgresql), Web servers (*e.g.*, Apache), and mail servers; user access control (*e.g.*, NIS, LDAP), backups and restores, snapshot policies, and more. I've personally installed, configured, changed, replaced parts, and upgraded components in numerous devices including mobile devices, laptops, desktops, and servers, both physical and virtual.

15. Since 1995, I have taught courses on operating systems, storage and file systems, advanced systems programming in Unix/C, systems administration, data structures, data/software security, and more. My courses often use storage, file systems, distributed systems, and system/network security as key teaching principles and practical examples for assignments and projects. I have taught these concepts and techniques to my students, both to my direct advisees as well as in my courses. For example, in my graduate Operating Systems course, I often cover Linux's kernel

mechanisms to protect users, applications, and data files, virtual file systems, as well as distributed storage systems (*e.g.*, NFS). And in the System Administration undergraduate course, I covered many topics such as networking, storage, backups, and configuring complex applications such as mail, web, and database servers.

16. My research often investigates computer systems from many angles: security, efficiency, energy use, scalability, reliability, portability, survivability, usability, ease-of-use, versatility, flexibility, and more. My research gives special attention to balancing five often-conflicting aspects of computer systems: performance, reliability, energy use, security, and ease-of-use.

17. Since joining Stony Brook University in 2001, my group in the File-systems and Storage Lab (FSL) has developed many file systems and operating system extensions; examples include a highly-secure cryptographic file system, a portable copy-on-write (COW) versioning file system, a tracing file system useful to detect intrusions, a replaying file system useful for forensics, a snapshotting and sandboxing file system, a namespace unification file system (that uses stackable, virtualized, file-based COW), an anti-virus file system, an integrity-checking file system, a load balancing and replication/mirroring file system, network file system extensions for security and performance, distributed secure cloud-based storage systems, transactional key-value stores and file systems, OS-level embedded databases, a compiler to convert user-level C code to in-kernel efficient yet safe

code, GCC plugins, stackable file system templates, and a Web-based backup system. Many of these projects used one form of virtualization or another (storage, network, host, etc.). I continue to maintain and release newer versions of some of these file systems and software.

18. I have published over 120 refereed publications (in ACM, IEEE, USENIX, and more). To date, my publications have been cited more than 10,000 times (as per Google Scholar as of September 24, 2024). My papers cover a wide range of related technologies such as file systems, storage systems, transactional systems, security, clouds and virtualization, performance benchmarking and optimization, energy efficiency, system administration, web systems, and more. I also published a book titled “Linux NFS and Automounter Administration” (Sybex, 2001), covering systems administration topics related to network storage and data security.

19. Some of my research has led to public software releases that have been used worldwide. I have publicly maintained the Amd Berkeley Automounter in a package called “am-utils” since 1992; this software helps administrators manage the multitude of file system mounts on dozens of different Unix systems, especially helping to automate access to multiple NFS/NAS storage volumes. Since 1997, I have maintained and released several stackable (virtualized) file system software projects for Linux, FreeBSD, and/or Sun Solaris, in a package called FiST. One of

my stackable file system encryption projects, called Cryptfs, became the basis for IBM's public release of eCryptfs, now part of Linux. Packet General Networks, for whom I have provided consulting services between 2009 and 2019, licensed another encryption file system called Ncryptfs. Another popular file system released in 2003, called Unionfs, offers virtual namespace unification, transparent shadow copying (a.k.a. copy-on-write or COW), file system snapshotting (*e.g.*, useful for forensics and disaster recovery), and the ability to save disk space by sharing a read-only copy of data among several computers, among other features.

20. My research and teaching make extensive use of data security features. For example, each time I taught the graduate operating system course, the first homework assignment includes the creation of a new system call that performs new or added functionality, often for encrypting a file or verifying its integrity; many of my other assignments cover topics of user/process access control, anti-virus filtering, and more. Since 2001, over 1,000 graduate students were exposed to these principles directly through my teaching and research at Stony Brook University.

21. Moreover, in an undergraduate course titled "Advanced Systems Programming in Unix/C," I cover many topics of system security and vulnerabilities, such as the structure of UNIX processes, and memory segments such as the heap and stack. This course covers details of several hundred Linux system calls. Often, the first assignment for this course is to develop a tool to encrypt/decrypt files using

advanced ciphers, use digital signatures to certify the cipher keys used, and reliably recover files in case of failures. Since 2001, several hundred undergraduate students were exposed to these principles directly through my teaching and research at Stony Brook University.

22. In another undergraduate course, System Administration, I taught network configuration, security, and storage configuration and reliability. In a special topics course on Storage Systems, I covered many topics such as data deduplication, RAID, transactional storage, storage hardware including modern Flash based ones, virtual storage, backup/restore, snapshots and continuous data protection (CDP), NAS and SAN, and NFS.

23. Overall, in addition to the aforementioned experience, my technical experience relevant to the challenged patent(s) at the time of the alleged invention(s) included the following: I've used numerous storage devices, from volatile RAM and variants, to non-volatile E/PROMS, hard disks, floppies, CD-ROMs, Smart Cards, SD cards, PCMCIA cards, flash-based storage, and more. Many of these were removable media devices. I've designed, developed, released, and/or maintained dozens of file systems—both local, remote/networked, or for removable media. Many of my file systems had security uses: encryption, malware detection, integrity checking, and more. My own Ph.D. dissertation covered encrypted file systems, virtual file systems, and more. My own file system, Crypfts, was picked up by IBM

and made into a commercial version, now part of Linux; another variant, called Ncryptfs, was licensed by Stony Brook University to a startup. I've used and/or developed an assortment of virtualization technologies, especially in the storage and file system arena. I've been involved in several standardization efforts, most notably the Network File System version 4 (NFSv4) standard in the late 1990s; my own book covers NFSv4 in detail.

24. My research has been supported by many federal and state grants as well as industry awards, including an NSF CAREER award, two IBM Faculty awards, two NetApp Faculty awards, a Western Digital award, a Facebook award, several Dell-EMC awards, and several equipment gifts. I received the 2008 SUNY Chancellor's Excellence in Teaching award, and the 2022 SUNY Chancellor's Award for Excellence in Scholarship and Creative Activities (both awards can be given only once in a lifetime). In 2021 I was named an ACM Distinguished Member for "Outstanding Scientific Contributions to Computing."

25. My service record to the community includes serving as the co-chair for the USENIX Annual Technical Conference in 2020 (ATC'20); serving as the co-chair for USENIX File and Storage Technologies (FAST'15) in 2015 and on the FAST Conference Steering Committee from 2015 to 2023; serving on the ACM HotStorage Steering Committee since 2021; and serving as the co-chair in 2012 and on the Steering Committee of the ACM SYSTOR conference since 2012. I have

served as an Associate Editor to the ACM Transactions on Storage (TOS) journal from 2009 to 2022; in 2022, I was named the Editor-in-Chief for ACM's TOS journal.

26. I am a named inventor on four patents, two titled "Systems and Methods for Detection of New Malicious Executables" (U.S. Patent No. 7,487,544, issued February 3, 2009; and U.S. Patent No. 7,979,907, issued July 12, 2011); and two more titled "Multi-Tier Caching," (U.S. Patent No. 9,355,109, issued May 31, 2016; and U.S. Patent 9,959,279, issued May 1, 2018).

27. I have been disclosed as a testifying expert in 18 cases (including *inter partes* review (IPR) proceedings) in the past four years. I have been deposed 13 times and testified in trial twice.

III. MATERIALS CONSIDERED

28. The opinions contained in this declaration are based on the documents I reviewed, my professional judgment, as well as my education, experience, and knowledge regarding electrical engineering, computer science, video and audio and communications sciences and technologies.

29. In forming my opinions expressed in this declaration, I reviewed the following materials:

Exhibit No.	DESCRIPTION
1001	U.S. Patent 8,745,749 (“’749Pat”)
1003	Zadok Curriculum Vitae
1004	File History of U.S. Patent 8,745,749 (Application No. 13/078,323) (“’749FH”)
1005	U.S. Provisional Patent Application 61/413,926
1006	U.S. Patent 6,611,907 (“Maeda”)
1007	U.S. Patent Publication No. 2002/0080190A1 (“Hamann”)
1008	U.S. Patent 6,789,192 (“Hirota”)
1009	CN100468438C (“Lu”) (with certified translation)
1010	U.S. Patent Publication No. 2012/0272065 (“Matsukawa”)
1011	Intellectual Property (IP) Information from the SD Card Association (SD-3C LLC) https://www.sd-3c.com/IPInformation.aspx
1012	A Dictionary of Computing, 6th ed., Oxford University Press (2008), excerpts (“Oxford Dictionary”)
1013	Screen Shot of http://www.4centity.com/about.aspx (4C Entity - About Us)
1014	Microsoft Computer Dictionary, 3rd ed., 1997, excerpts (“Microsoft Computer Dictionary”)
1015	DOD Standard, Department of Defense Trusted Computer System Evaluation Criteria, December 26, 1985 (“Orange Book”)
1016	U.S. Publication No. 2005/0273858 (“Zadok Patent Publication”)

Exhibit No.	DESCRIPTION
1017	Smart Card Handbook by Rankl and Effing, 1999-2000, John Wiley & Sons, Ltd., excerpts (“Smart Card Handbook”)
1018	C. P. Wright, M. Martino, and E. Zadok. NCryptfs: A secure and convenient cryptographic file system. In <i>Proceedings of the Annual USENIX Technical Conference</i> , pages 197–210, San Antonio, TX, June 2003. USENIX Association. (“NCryptfs”).
1019	SD Association – Fact Sheet (https://www.sdcard.org/about-sda/fact-sheet/)

30. I have also reviewed any other materials I refer to in this declaration in support of my opinions.

31. All of the opinions contained in this declaration are based on the documents I reviewed and my knowledge, experience, and professional judgment. My opinions have also been guided by my appreciation of how a person of ordinary skill in the art would have understood the claims and the specification of the ’749 patent at the time of the alleged invention, which I have been asked to initially consider as no earlier than November 15, 2010 (the filing date of U.S. Provisional Patent Application 61/413,926, EX1005). My opinions reflect how one of ordinary skill in the art would have understood the ’749 patent, the prior art to the patent, and the state of the art at the time of the alleged invention.

32. Based on my experience and expertise, it is my opinion that certain references disclose and/or suggest all the features recited in claims 1-8, 13-17, and 18-25 (“challenged claims”) of the ’749 patent, as I discuss in detail below.

IV. LEVEL OF ORDINARY SKILL IN THE ART

33. I have been informed and understand that, in the context of an invalidity analysis, a person having ordinary skill in the art (“POSITA”) is a hypothetical person who looks to prior art at the time of the invention. I further understand that the factors that may be considered in determining the level of ordinary skill include: (1) the problems encountered in the art; (2) the prior art solutions to the problems encountered in the art; (3) the rapidity of innovation; (4) the sophistication of the technology; and (5) the education level of active workers in the field. I understand that these factors need not all be considered for the analysis and that one or more of these factors may control.

34. I was asked to provide my opinion on the level of one of ordinary skill in the art with respect to the alleged invention of the ’749 patent as of November 15, 2010. Based on my consideration of the factors above, in my opinion a POSITA would have had a bachelor’s degree computer science, computer engineering, or equivalent degree, and approximately two years of experience working in the computer science or engineering field. Additional experience might substitute for less education and vice versa. To that end, a POSITA in November 2010 would

have been familiar with the design and implementation of data storage and/or file systems, as well as storage security and privacy techniques.

35. As of November 15, 2010, I met, and in fact exceeded, the qualifications of a person of ordinary skill in the art. To be clear, all of my opinions in this declaration are from the perspective of one of ordinary skill in the art as I have defined it here during the relevant timeframe.

V. RELEVANT LEGAL STANDARDS

36. I am not an attorney and offer no legal opinions, but in the course of my work, I have had experience studying and analyzing patents and patent claims from the perspective of a person skilled in the art.

37. For the purposes of this declaration, I have been informed about certain aspects of the law that are relevant to forming my opinions. My understanding of the law is as follows.

38. Petitioners' counsel has informed me that a published nonprovisional application is entitled to claim priority to the filing date of its provisional application (as opposed to the filing date of the published non-provisional application) if (1) the provisional application discloses the portions of the published non-provisional application relied upon as prior art, and (2) at least one claim of the published non-provisional claim is supported by the provisional application, *i.e.*, the provisional

application must disclose the subject matter of the claim and enable a person of ordinary skill in the art to make and use the claim.

39. Petitioners' counsel has informed me that for the prior art to inherently disclose a claimed limitation, the prior art need not expressly disclose the limitation, so long as the claimed limitation necessarily flows from a disclosure in the prior art. I also understand that it is acceptable to examine evidence outside the prior art reference (extrinsic evidence) in determining whether a feature, while not expressly discussed in the reference, is necessarily present in that reference.

40. Petitioners' counsel has informed me that a patent claim can be considered to have been obvious to a person having ordinary skill in the art at the time the application was filed. I am informed that this means that, even if all of the requirements of a claim are not found in a single prior art reference, the claim is not patentable if the differences between the subject matter in the prior art and the subject matter in the claim would have been obvious to a person having ordinary skill in the art at the time of the invention.

41. I have been informed by Petitioners' counsel that a determination of whether a claim would have been obvious should be based upon several factors, including, among others:

- the level of ordinary skill in the art at the time the application was filed;
- the scope and content of the prior art; and

- what differences, if any, existed between the claimed invention and the prior art.

42. I have been informed by Petitioners' counsel that a single reference can render a patent claim obvious if any differences between that reference and the claims would have been obvious to a person having ordinary skill in the art. Alternatively, I understand that the teachings of two or more references may be combined in the same way as disclosed in the claims, if such a combination would have been obvious to one having ordinary skill in the art. In determining whether a combination based on either a single reference or multiple references would have been obvious, it is appropriate to consider, among other factors:

- whether the teachings of the prior art references disclose known concepts combined in familiar ways, and when combined, would yield predictable results;
- whether a person having ordinary skill in the art could implement a predictable variation, and would see the benefit of doing so;
- whether the claimed elements represent one of a limited number of known design choices, and would have a reasonable expectation of success by those skilled in the art;
- whether a person having ordinary skill would have recognized a reason to combine known elements in the manner described in the claim;

- whether there is some teaching or suggestion in the prior art to make the modification or combination of elements claimed in the patent; and
- whether the innovation applies a known technique that had been used to improve a similar device or method in a similar way.

43. I have been informed by Petitioners' counsel that one of ordinary skill in the art has ordinary creativity and is not an automaton. Petitioners' counsel has also informed me that in considering obviousness, obviousness may not be determined using the benefit of hindsight, including hindsight derived from the patent being considered.

44. I further understand that a hypothetical person of ordinary skill in the art is assumed to know and to have all relevant prior art in the field of endeavor covered by the patent at issue. I have been informed that there are two criteria for determining whether prior art is analogous and thus can be considered prior art: (1) whether the art is from the same field of endeavor, regardless of the problem addressed, and (2) if the reference is not within the field of the patentee's endeavor, whether the reference still is reasonably pertinent to the particular problem with which the patentee is involved. I have also been informed that the field of endeavor of a patent is not limited to the specific point of novelty, the narrowest possible conception of the field, or the particular focus within a given field. I have also been informed that a reference is reasonably pertinent if, even though it may be in a

different field from that of the patentee's endeavor, it is one which, because of the matter with which it deals, logically would have commended itself to a patentee's attention in considering their problem.

45. I further understand that, in making a determination as to whether the claimed invention would have been obvious to a person of ordinary skill, the Board may consider certain objective factors if they are present, such as: commercial success of products practicing the claimed invention; long-felt but unsolved need; teaching away; unexpected results; copying; and praise by others in the field. These factors are generally referred to as "secondary considerations" or "objective indicia" of nonobviousness. I understand, however, that for such objective evidence to be relevant to the obviousness of a claim, there must be a causal relationship (called a "nexus") between the claim and the evidence, and that this nexus must be based on a novel element of the claim rather than something in the prior art. I also understand that even when they are present, secondary considerations may be unable to overcome primary evidence of obviousness (such as motivation to combine with predictable results) that is sufficiently strong.

VI. SUMMARY OF OPINIONS

46. This Declaration explains the conclusions I have formed based on my analysis. These conclusions are summarized the table below:

Ground	Basis	Relied-On References	Claim(s)
1	Obviousness	Maeda, Hamann, and the Admitted Prior Art (“AAPA”)	1, 14-16, 18, 22-24
2	Obviousness	Maeda, Hamann, the AAPA, and Hirota	2, 6-8, 19
3	Obviousness	Maeda, Hamann, the AAPA, Hirota, and Lu	3-5, 20, 21, 25
4	Obviousness	Maeda, Hamann, the AAPA, and Matsukawa	13

VII. TECHNOLOGY OVERVIEW

47. In this section, I provide a brief overview of the background art as understood by a person of ordinary skill in the art as of November 15, 2010.

A. General Computer Operations

48. Computers are hardware and software devices that execute instructions. The hardware often includes a central processing unit (CPU) or other microprocessors, typically called “processor,” volatile memory (*e.g.*, RAM (random-access memory), or DRAM (dynamic random-access memory)), non-volatile (persistent) storage (*e.g.*, a hard disk or flash drive), and input/output (I/O) devices (*e.g.*, network interface card, mouse, keyboard, display). RAM, hard disks, and flash drives are considered, generally, “storage” or “memory” devices because they can store and hold bits of information (whether persistently or not). Computers can have multiple storage devices. Processors are coupled to main memory using a

memory bus and to peripherals such as hard disks and network interface cards using one or more I/O busses.

49. Memory and storage devices have considerable differences in their speeds, capacities, and costs. Non-volatile storage devices such as hard disks have the largest capacity, are least expensive per gigabyte, and are slowest (typically operating at millisecond speeds). Volatile DRAM devices are smaller, more expensive, and faster than hard disks (typically operating at microsecond speeds). CPUs' own memory caches and operations are the smallest, most expensive, and fastest (typically operating at nanosecond speeds). Overall, CPUs are about 1,000× faster than DRAM, which in turn is about 1,000× faster than hard disks.

50. Computer programs are often written in a high-level human-readable language (*e.g.*, C, C++, Java, Perl, PHP, Python), then translated using a compiler or script processor to machine instructions understood by the CPU (*e.g.*, Intel or AMD processor). Machine instructions—generally called “software”—are stored in files on persistent media (*e.g.*, HDD or ROM), loaded into DRAM, and then loaded into the CPU where they can be executed. All modern computers operate in this manner, regardless of their purpose: a small laptop, a server in a data center, a mainframe computer, a personal workstation, a proxy device, a gateway, a firewall, a router, an appliance, a virtual machine, a handheld or mobile device, smart card,

SD card, etc. Computer systems may run multiple software programs, from background servers and daemons to user applications.

51. Users (including administrators) can login to one or more computers (e.g., desktop, laptop, server) and execute one or more applications (e.g., word processing, web browsing, backup tool). Some computers offer external interfaces and APIs to communicate with certain devices and especially peripherals. For example, both a USB storage device as well as a printer use a well-defined “protocol” to allow users to communicate with them through the main computer.

52. A Local Area Network (LAN) is a typically a network at a local site where network latencies are short and physical distances between computers are small. A LAN and its computers are often protected from the rest of the internet using a firewall. EX1014 (Microsoft Computer Dictionary) at 276 (“**LAN** *n.* Acronym for **local area network**. A group of computers and other devices dispersed over a relatively limited area and connected by a communications link that enables any device to interact with any other on the network. LANs commonly include microcomputers and shared resources such as laser printers and large hard disks. The devices on a LAN are known as nodes, and the nodes are connected by cables through which messages are transmitted. *See also* baseband network, broadband network, bus network, collision detection, communications protocol, contention,

CSMA/CD, network, ring network, star network, token bus network, token passing, token ring network. *Compare* wide area network.”).

53. Another form of a local area network is one that operates inside a single computer. A “localhost” is “[t]he name that is used to represent the same computer on which a TCP/IP message originates. An IP packet sent to localhost has the IP address 127.0.0.1 and does not actually go out to the Internet. *See also* IP address, packet (definition 1), TCP/IP.” EX1014 (Microsoft Computer Dictionary) at 287. Often, when users/processes want to communicate with other users/processes, using TCP/IP communications on the same machine, they can use the localhost address: it is considered a trusted, high-speed internal network.

54. Conversely, a Wide-Area Network (WAN) refers to computers across the entire Internet, who are distant from each other, exhibit longer latencies, and afford little protection. EX1014 (Microsoft Computer Dictionary) at 507 (“**wide area network** *n.* A communications network that connects geographically separated areas. *Acronym:* WAN.”)

55. A firewall typically intercepts all traffic between computers in a LAN and the WAN, monitors and controls access, and can filter out undesired traffic. Often, LAN computers are far more trusted than WAN computers; therefore, a firewall can be said to separate the trusted network (LAN or localhost) from the untrusted network (WAN). EX1014 (Microsoft Computer Dictionary) at 197

(“**firewall** *n.* A security system intended to protect an organization’s network against external threats, such as hackers, coming from another network, such as the Internet. A firewall prevents computers in the organization’s network from communicating directly with computers external to the network and vice versa. Instead, all communication is routed through a proxy server outside of the organization’s network, and the proxy server decides whether it is safe to let a particular message or file pass through to the organization’s network.”).

56. A server is a computer or software that provides a service to other computers, called clients. There are many servers on the Internet, for example, Web servers and Mail servers. EX1014 (Microsoft Computer Dictionary) at 430 (“**server** *n.* 1. On a local area network (LAN), a computer running administrative software that controls access to the network and its resources, such as printers and disk drives, and provides resources to computers functioning as workstations on the network. 2. On the Internet or other network, a computer or program that responds to commands from a client. For example, a file server may contain an archive of data or program files; when a client submits a request for a file, the server transfers a copy of the file to the client. *See also* client/server architecture. *Compare* client (definition 3).”

57. A client is typically an end-point computer or software, sometimes used by a user, that communicates across LAN and WAN networks; a client may contact servers to request some service. A user’s desktop, laptop, and handheld device are

examples of clients. (Note that computer can be both a client and a server.) EX1014 (Microsoft Computer Dictionary) at 92 (“**client** *n.* 1. In object-oriented programming, a member of a class (group) that uses the services of another class to which it is not related. *See also* inheritance (definition 1). 2. A process, such as a program or task, that requests a service provided by another program—for example, a word processor that calls on a sort routine built into another program. The client process uses the requested service without having to ‘know’ any working details about the other program or the service itself. *Compare* child (definition 1), descendant (definition 2). 3. On a local area network or the Internet, a computer that accesses shared network resources provided by another computer (called a server). *See also* client/server architecture, server.”).

58. A client/server architecture describes how client computers interact with server computers. EX1014 (Microsoft Computer Dictionary) at 92 (“**client/server architecture** *n.* An arrangement used on local area networks that makes use of distributed intelligence to treat both the server and the individual workstations as intelligent, programmable devices, thus exploiting the full computing power of each. This is done by splitting the processing of an application between two distinct components: a ‘front-end’ client and a ‘back-end’ server. The client component is a complete, stand-alone personal computer (not a ‘dumb’ terminal), and it offers the user its full range of power and features for running

applications. The server component can be a personal computer, a minicomputer, or a mainframe that provides the traditional strengths offered by minicomputers and mainframes in a time-sharing environment: data management, information sharing between clients, and sophisticated network administration and security features. The client and server machines work together to accomplish the processing of the application being used. Not only does this increase the processing power available over older architectures but it also uses that power more efficiently. The client portion of the application is typically optimized for user interaction, whereas the server portion provides the centralized, multiuser functionality. *See also* distributed intelligence.”).

B. Storage Devices and Media

59. As mentioned in §VII.A above, computers often include two types of storage or memory devices. The first type is volatile memory device, called Random Access Memory (“RAM”), that is often used to hold programs that are executing and their data. EX1014 (Microsoft Dictionary), p. 372 (“RAM”).

60. The second type is a non-volatile, or persistent storage device, such as a floppy disk or hard disk drive (“HDD”).

61. A floppy disk is a flexible, soft, magnetic, removable media. Floppy disks are examples of removable media, similarly to SD cards. A “floppy disk” is “[a] round piece of flexible plastic film coated with ferric oxide particles that can

hold a magnetic field. When placed inside a disk drive, the floppy disk rotates to bring different areas, or sectors, of the disk surface under the drive's read/write head, which can detect and alter the orientation of the particles' magnetic fields to represent binary 1s and 0s. A floppy disk 5.25 inches in diameter is encased in a flexible plastic jacket and has a large hole in the center, which fits around a spindle in the disk drive; such a disk can hold from a few hundred thousand to over one million bytes of data. A 3.5-inch disk encased in rigid plastic is also called a floppy disk or a microfloppy disk. In addition, 8-inch floppy disks were common in DEC and other minicomputer systems. *See also* 3.5-inch floppy disk, 5.25-inch floppy disk, microfloppy disk." EX1014 (Microsoft Dictionary), p. 201.

62. Compared to flexible "soft" media such as floppies, "hard disks" are self-contained devices. HDDs are used to store programs and data persistently. EX1014 (Microsoft Dictionary), p. 226 ("hard disk").

63. HDDs are electro-mechanical devices with moving parts. Computers can also have other types of non-volatile storage devices that have no moving parts, also called "solid-state devices" or "semi-conductor memories," such as flash memories. EX1014 (Microsoft Dictionary), p. 199 ("flash memory").

64. Some devices can be read or written, while others are read-only (e.g., ROM, CD-ROM). This helps to protect their contents so it can never be altered. "ROM" is "1. Acronym for **read-only memory**. A semiconductor circuit into which

code or data is permanently installed by the manufacturing process. The use of this technology is economically viable only if the chips are produced in large quantities; experimental designs or small volumes are best handled using PROM or EPROM.

2. Acronym for **read-only memory**. Any semiconductor circuit serving as a memory that contains instructions or data that can be read but not modified (whether placed there by manufacturing or by a programming process, as in PROM and EPROM). *See also* EEPROM, EPROM, PROM.” EX1014 (Microsoft Dictionary), p. 413.

65. Moreover, any single device (*e.g.*, a hard disk) can be logically broken down into several units called “partitions.” EX1014 (Microsoft Dictionary), p. 152 (“**disk partition** *n.* A logical compartment on a physical disk drive. A single disk might have two or more logical disk partitions, each of which would be referenced with a different disk drive name. Multiple partitions are divided into a primary (boot) partition and one or more extended partitions.”).

66. Above the storage media often lives a logical software layer called a “file system” discussed in §VII.C below.

1. SD Cards vs. Smart Cards

67. Smart Cards predate SD Cards but both share very similar features and are based on nearly identical technologies.

68. A “smart card” is, “1. ... [i]n computers and electronics, a circuit board with built-in logic or firmware that gives it some kind of independent decision-

making ability. 2. In banking and finance, a credit card that contains an integrated circuit that gives it a limited amount of ‘intelligence’ and memory.” EX1014 (Microsoft Computer Dictionary), p. 439.

69. Smart Cards’ main “feature is an integrated circuit embedded in the card, which has components for the transmission, storage and processing of data.” EX1017 (Smart Card Handbook), p. 18.

70. Like SD Cards, the “possible applications for the Smart Card are extremely diverse. They are also constantly being extended with the increasing arithmetic power and storage capacity of available integrated circuits.” EX1017 (Smart Card Handbook), p. 5.

71. Smart Cards have a diverse set of application areas, and wide range of storage capacities, as can be seen from Figure 1.1 below.

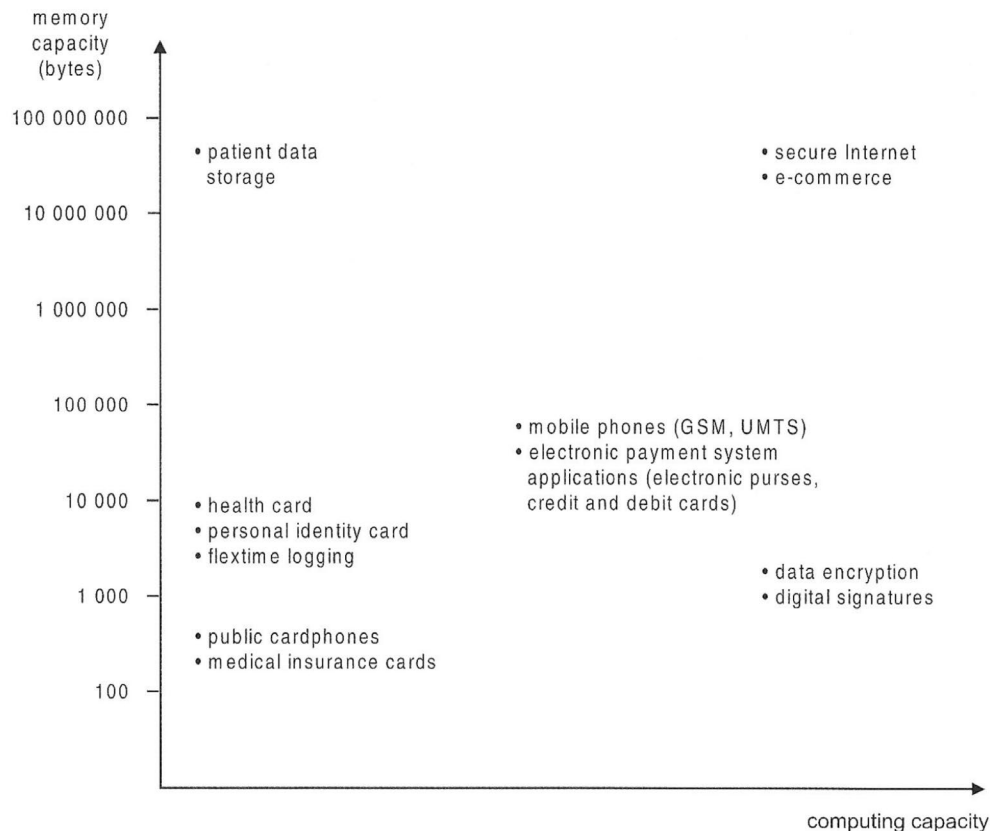


Figure 1.1 Typical Smart Card application areas, showing the required storage capacity and arithmetic processing capacity

EX1017 (Smart Card Handbook), Figure 1.1, p. 7.

72. These applications include “secure internet”, storing sensitive “patient data,” “electronic payment systems” including “electronic purses, credit and debit cards,” “data encryption,” and “digital signatures.” Clearly, all of these examples and more are in require secure storage, encryption, and strong authentication methods. This is identical purpose to the **Secure** Digital (SD) card.

73. Like SD Cards, Smart Cards have also been standardized by an international body. “The development of international standards for Smart Cards takes place under the aegis of ISO/IEC and/or the CEN. The German standards

association DIN is represented in all relevant committees.” EX1017 (Smart Card Handbook), p. 11.

74. “As can be seen, there are two technical committees that are concerned with the standardization of Smart Cards. The first of these is ISO TC68/SC6, which is responsible for the standardization of cards used for financial transactions, while the second is ISO/IEC JTC1/SC17, which is responsible for general applications.” EX1017 (Smart Card Handbook), p. 12.

75. SD Cards were designed as general-purpose portable storage devices, with security and privacy capabilities. Smart Cards, despite being older, are in some ways more versatile than SD Cards because Smart Cards can be used to store data securely (as a memory card), as well as act as microprocessor cards and support contactless, secure data access. *See, generally*, EX1017 (Smart Card Handbook), pp. 18-25

76. SD Cards come in a variety of capacities. So do Smart Cards, as can be seen in Figure 1.1 above. Moreover, “[t]he maximum storage capacity of a Smart Card is many times greater than that of a magnetic-strip card. Chips with over 32 kB of memory are currently available, and this figure will multiply with each new chip generation.” EX1017 (Smart Card Handbook), p. 18.

77. Like SD Cards, Smart Cards are first and foremost data storage devices using secure access. “In addition to containing mechanisms for identification and

authentication, Smart Cards are above all data stores. Here they have a decisive advantage with respect to other storage media, such as diskettes, in that access to the data can be coupled to certain conditions.” EX1017 (Smart Card Handbook), p. 175.

78. Both SD Cards and Smart Cards include a file system to organize files and folders hierarchically. “Nowadays, all newer types of Smart Cards have complete, hierarchical file management systems with symbolic, hardware-independent addressing.” EX1017 (Smart Card Handbook), p. 175.

79. “The structure of Smart Card file systems, which is specified in the ISO/IEC 7816-4 standard, is similar to the file structures of DOS or Unix systems.” EX1017 (Smart Card Handbook), p. 176.

80. “There are basically two categories of files for file Smart Cards. The first category is directory files, which are called ‘Dedicated Files’ (DFs). The second category consists of the files that hold the actual useful data, which are called ‘Elementary Files’ (EFs).” EX1017 (Smart Card Handbook), p. 177.

81. Both Smart Cards and SD Cards were designed to protect stored data by encrypting and authenticating access to it. “However, one of the most important advantages of Smart Cards consists in the fact that their stored data can be protected against unauthorized access and tampering. As access to data takes place only via a serial interface that is controlled by an operating system and security logic, it is possible to write confidential data to the card so that it can never be read from the

outside. The confidential data can be processed only internally by the chip's processing unit. In principle, the memory functions of writing, erasing and reading can be restricted and linked to specific conditions by both hardware and software. This allows the construction of numerous security mechanisms, which can also be tailored to the special demands of specific applications. This together with the ability to compute cryptographic algorithms allows Smart Cards to be used to implement convenient security modules that can be carried about at all times, for example in a briefcase." EX1017 (Smart Card Handbook), p. 18.

82. Both SD Cards and Smart Cards require authenticated access to secure and encrypt files, and use random numbers and generators in the process.

83. "The most commonly used method of user identification is the entry of a secret number, which is generally referred to by the acronym PIN (for personal identification number), or sometimes CHV (cardholder verification)." EX1017 (Smart Card Handbook), p. 384. "A biometric identification procedure is one that can unambiguously identify a person by means of unique, individual biological features." EX1017 (Smart Card Handbook), p. 390.

84. SD Cards include a unique media identifier that can be used to derive encryption keys. Smart Cards similarly include a "unique chip number." "When the semiconductor chips are manufactured, a unique chip number is written to this memory. This means that each chip is different and can be uniquely traced, and the

Smart Cards can later be unambiguously identified within the system. In addition, chip numbers can also be used for the derivation of keys, and they allow the generation of ‘blacklists’ that can be used to take suspect cards out of circulation.” EX1017 (Smart Card Handbook), p. 407.

85. “The security of a Smart Card application is based on the secret keys used for the cryptographic algorithm. In order to perform certain types of access or take certain actions with the card, the terminal must always first authenticate itself with the help of a secret key. ... The Smart Card authenticates the terminal by sending it a random number, which the terminal then encrypts and returns to the card. The Smart Card then performs the same encryption and compares the result with the value received from the terminal. If they match, the terminal has been authenticated and it receives a corresponding return code.” EX1017 (Smart Card Handbook), p. 431.

86. Smart Cards also use random numbers in the security scheme. “*Attack and defense: random number generators* The random numbers generated by the Smart Card are used in authentication to individualize a session, which means to make each session unique and different from all preceding and following sessions.” EX1017 (Smart Card Handbook), p. 436 (emphasis original).

87. Finally, SD Cards and Smart Cards were not the only standards for removable storage media. In the 1990s I used many different PCMCIA (or PC) cards for telecommunications, Internet access, Flash storage, and more.

88. The “PC Card” is “[a] trademark of the Personal Computer Memory Card International Association (PCMCIA) that is used to describe add-in cards that conform to the PCMCIA specification. A PC Card is a removable device, approximately the same size as a credit card, that is designed to plug into a PCMCIA slot. ... Type III cards accommodate devices that require more space, such as wireless communications devices and rotating storage media (such as hard disks). *See also* PCMCIA, PCMCIA slot.” EX1014 (Microsoft Dictionary), p. 357

89. Moreover, “PCMCIA” is an “[a]cronym for **P**ersonal **C**omputer **M**emory **C**ard **I**nternational **A**ssociation. A group of manufacturers and vendors formed to promote a common standard for PC Card-based peripherals and the slot designed to hold them, primarily on laptop, palmtop, and other portable computers, as well as for intelligent electronic devices. PCMCIA is also the name of the standard for PC Cards, first introduced in 1990 as release 1. *See also* PC Card, PCMCIA slot.” EX1014 (Microsoft Dictionary), p. 358.

C. Files and File Systems

90. A file is a collection of data and attributes, or meta-data, about the file, including the file’s name and extension. EX1014 (Microsoft Dictionary) at p. 211

(“**file** *n.* A complete, named collection of information, such as a program, a set of data used by a program, or a user-created document. A file is the basic unit of storage that enables a computer to distinguish one set of information from another. A file is the ‘glue’ that binds a conglomeration of instructions, numbers, words, or images into a coherent unit that a user can retrieve, change, delete, save, or send to an output device.”).

91. File attributes are also called file meta-data and include security attributes, the name (title), size of a file, timestamps, and more. EX1014 (Microsoft Dictionary) at p. 305 (“**metadata** or **meta data** *n.* 1. Data about data. For example, the title, subject, author, and size of a file constitute metadata about the file.”), p. 194 (“**file attribute** *n.* A restrictive label attached to a file that describes and regulates its use—for example, hidden, system, read-only, archive, and so forth. In MS-DOS, this information is stored as part of the file’s directory entry.”).

92. A file’s name, in particular, is an important attribute of a file and conveys a lot of useful information about the file’s meaning and purpose. EX1014 (Microsoft Dictionary) at p. 195 (“**filename** *n.* The set of letters, numbers, and allowable symbols assigned to a file to distinguish it from all other files in a particular directory on a disk. A file name is the label under which a computer user saves and requests a block of information. Both programs and data have file names and often extensions that further identify the type or purpose of the file. Naming

conventions, such as maximum length and allowable characters of a file name, vary from one operating system to another. *See also* directory, path (definition 5).”).

93. A file’s path or path-name conveys the location of a file in the name hierarchy and even which file server the file may reside on (*e.g.*, the local C:\ drive or a network drive like N:\). EX1014 (Microsoft Dictionary) at p. 356 (“**path** *n.* [...] 5. In file storage, the route followed by the operating system through the directories in finding, sorting, and retrieving files on a disk.”), *Id.* (“**pathname** *n.* In a hierarchical filing system, a listing of the directories or folders that lead from the current directory to a file. *Also called:* directory path.”).

94. Moreover, a file’s extension conveys vital information about the file’s type and hence its meaning and purpose. For example, a file ending with “xls” is known to be a Microsoft Excel spreadsheet, files with “doc” extensions are Microsoft Word files, and those ending with “html” are Web server hyper-text markup language files. EX1014 (Microsoft Dictionary) at p. 211 (“**file extension** *n.* *See* extension (definition 1).”), p. 212 (“**filename extension** *n.* *See* extension (definition 1).”), p. 186 (“**extension** *n.* 1. A set of characters added to a filename that serves to extend or clarify its meaning or to identify a file as a member of a category. An extension may be assigned by the user or by a program, as, for example, .com or .exe for executable programs that MS-DOS can load and run.”).

95. The (usually non-volatile) storage media that holds the actual bits of file data is often called the “secondary storage” or the “backing storage” (because it is in the “back end”); this is different than “backup storage” that’s often used to backup files for disaster recovery purposes. A “primary storage” is “Random access memory (RAM); the main general-purpose storage region to which the microprocessor has direct access. A computer’s other storage options, such as disks and tape, are called *secondary storage* or (sometimes) *backing storage*.” EX1014 (Microsoft Dictionary), pp. 378-379. A “secondary storage” is “Any data storage medium other than a computer’s random access memory (RAM)-typically tape or disk. *Compare* primary storage.” EX1014 (Microsoft Dictionary), p. 425.

96. Above the storage media often lives a logical software layer called a “file system.” File systems provide a convenient abstraction to users and applications of files, folders or directories, and naming. EX1014 (Microsoft Dictionary), p. 196 (“**file system** *n.* In an operating system, the overall structure in which files are named, stored, and organized. A file system consists of files, directories, and the information needed to locate and access these items. The term can also refer to the portion of an operating system that translates requests for file operations from an application program into low-level, sector-oriented tasks that can be understood by the drivers controlling the disk drives. *See also* driver.”).

97. The “FAT file system” is “[t]he system used by MS-DOS to organize and manage files. The FAT (file allocation table) is a data structure that MS-DOS creates on the disk when the disk is formatted. When MS-DOS stores a file on a formatted disk, the operating system places information about the stored file in the FAT so that MS-DOS can retrieve the file later when requested. The FAT is the only file system MS-DOS can use; OS/2, Windows NT, and Windows 95 operating systems can use the FAT file system¹ in addition to their own file systems (HPFS, NTFS, and VFAT, respectively). *See also* file allocation table, HPFS, NTFS, OS/2, VFAT, Windows 95, Windows NT.”). EX1014 (Microsoft Dictionary) at 189-190.

98. Finally, before any storage media or partition can be used to access files and folders, it must be formatted. To “format” is “2. To prepare a disk for use by organizing its storage space into a collection of data ‘compartments,’ each of which can be located by the operating system so that data can be sorted and retrieved. When a previously used disk is formatted, any preexisting information on it is lost.” EX1014 (Microsoft Dictionary), p. 204 (“Format” definition 2).

D. Data Security and Privacy

99. All modern file and storage systems include measures of security to protect their data from unauthorized access. EX1014 (Microsoft Dictionary), p. 195 (“**file protection** *n.* A process or device by which the existence and integrity of a file are maintained. Methods of file protection range from allowing read-only access

and assigning passwords to covering the write-protect notch on a disk and locking away floppy disks holding sensitive files.”).

100. The POSIX standard, first started in the late 1980s, codifies the access application program interfaces (APIs) that operating systems export to user applications. EX1014 (Microsoft Dictionary), p. 351 (“**POSIX** *n.* Acronym for **Portable Operating System Interface for UNIX**. An IEEE standard that defines a set of operating-system services. Programs that adhere to the POSIX standard can be easily ported from one system to another. POSIX was based on UNIX system services, but it was created in a way that allows it to be implemented by other operating systems.”).

101. A simple data protection scheme defines whether users can only read files, only write files, or have both read and write access. EX1014 (Microsoft Dictionary), p. 361 (“**permission** *n.* In a networked or multiuser computer environment, the ability of a particular user to access a particular resource by means of his or her user account. Permissions are granted by the system administrator or other authorized person; these permissions are stored in the system (often in a file called a *permissions log*) and are checked when a user attempts to access a resource.”).

102. In the basic POSIX model, for example, a file has three sets of permissions (read, write, or execute) for three classes of users (owner, group member, and everyone else).

103. More sophisticated schemes generalize these simpler permission models to allow for lists of users, lists of groups, as well as logical operations on lists (*e.g.*, negation, conjunction, and disjunction). These are called “access control lists,” or ACLs. EX1014 (Microsoft Dictionary), p. 13 (“**access control** *n.* The mechanisms for limiting access to certain items of information or to certain controls based on users’ identities and their membership in various predefined groups. Access control is typically used by system administrators for controlling user access to network resources, such as servers, directories, and files. *See also* access privileges, system administrator.”), p. 13 (“**access control list** *n.* A list associated with a file that contains information about which users or groups have permission to access or modify the file. *Acronym:* ACL.”). ACLs existed in Unix systems since at least the 1990s (and I personally used them).

104. Security has often been defined in terms of levels because it was more logical and convenient to use: each higher level progressively has greater security control than the level just below, and hence subsumes the latter. The U.S. Department of Defense defined a standard in 1985, famously called the “Orange Book” (because the book’s cover was colored orange), which includes four general

different levels of security (from A (most secure) to D (least secure)), each with one or more sub-levels of security. EX1014 (Microsoft Dictionary), p. 323 (“Orange Book” definition).

105. The Orange Book defines a “Security Level” as “The combination of a hierarchical classification and a set of non-hierarchical categories that represents the sensitivity of information.” and “sensitivity label” as “A piece of information that represents the security level of an object and that describes the sensitivity (e.g., classification) of the data in the object. Sensitivity labels are used by the TCB as the basis for mandatory access control decisions.” EX1015 (Orange Book), p. 111.

106. The Orange Book detailed how data of different security levels has to be stored in separate compartments or locations, for example level C1. EX1015 (Orange Book), p. 15 (“The Trusted Computing Base (TCB) of a class (C1) system nominally satisfies the discretionary security requirements by providing separation of users and data. It incorporates some form of credible controls capable of enforcing access limitations on an individual basis, i.e., [] allowing users to be able to protect project or private information and to keep other users from accidentally reading or destroying their data. [].”

107. A POSITA would have understood that separating information of different security levels into different compartments made it much easier to safeguard and control access to such information. This is no different than when

people store some sensitive information (*e.g.*, bills) in a desk drawer at home, more sensitive information in a home safe (*e.g.*, passports), and the most sensitive information in a bank safe (*e.g.*, estate planning documents, home title/deed).

108. Similarly, the U.S. National Security Agency (NSA) defined its own standard called the “Red Book”, again defining multiple security levels. EX1014 (Microsoft Dictionary), p. 400 (“Red Book” definition).

109. A POSITA would have understood that these security levels can be applied to any piece of information, including paper and electronic documents and data files. The U.S. military has long had similar security levels, for example top-secret, secret, and confidential—each level being higher than the next and subsuming the access rights of those below (*e.g.*, a person with “secret” clearance could also access “confidential” documents but not those marked “top-secret”).

110. Even without having such standards, Unix administrators knew how to leverage much simpler POSIX permission models to effectively create security levels and control users’ access via authentication and authorization methods (something I did myself in the 1990s).

111. For example, one could create two Unix groups: (1) a group named “sales-managers” that includes the identities of all the sales managers in an organization and (2) a second group named “sales-people” the includes the same managers but also all the salespersons in the organization. In other words, the sales-

people group includes everyone in the sales-managers group and more. A file whose access permissions are set to group sales-people will be accessible by all sales people as well as managers; but a file whose access permissions are set to group sales-managers will only be accessible to the sales managers, and hence regular salespersons will not have access to that file. In such a way, a two-level security hierarchy is created, with sales-managers being logically at a higher security level than sales-people. The same mechanism can be used to create an arbitrarily long set of security levels.

112. As discussed above, ACLs are more versatile than these simpler Unix permission models. Therefore, with ACLs, the construction of any number of data or file security levels becomes even easier.

113. Another common transformation, useful to protect user's privacy, is "encryption", or "[t]he process of encoding data to prevent unauthorized access, especially during transmission. Encryption is usually based on a key that is essential for decoding. The U.S. National Bureau of Standards created a complex encryption standard, Data Encryption Standard (DES), which provides almost unlimited ways to encrypt documents. *See also* DES." EX1014 (Microsoft Computer Dictionary) at 175.

114. There are many forms of verification that can be applied to data files. One example is to verify the integrity of data using cryptographic hashing algorithms

such as SHA. “SHA” is an “[a]cronym for **Secure Hash Algorithm**. A technique that computes a 160-bit condensed representation of a message or data file, called a *message digest*. The SHA is used by the sender and the receiver of a message in computing and verifying a digital signature, for security purposes. *See also* algorithm, digital signature.” EX1014 (Microsoft Computer Dictionary) at 432. The term “data integrity” represents “[t]he accuracy of data and its conformity to its expected value, especially after being transmitted or processed.” EX1014 (Microsoft Computer Dictionary) at 131.

115. Users can also verify the authenticity of files’ sources using digital signatures. A “digital signature” is “[a] personal authentication method based on encryption and secret authorization codes used for ‘signing’ electronic documents.” EX1014 (Microsoft Computer Dictionary) at 145.

116. Digital signatures often combine cryptographic techniques and hashing algorithms to enable bi-directional authentication and secure communications. “public key cryptography” or “public key encryption” is “[a]n asymmetric scheme that uses a pair of keys for encryption: the public key encrypts data, and a corresponding secret key decrypts it. For digital signatures, the process is reversed: the sender uses the secret key to create a unique electronic number that can be read by anyone possessing the corresponding public key, which verifies that the message

is truly from the sender. *See also* private key, public key.” EX1014 (Microsoft Computer Dictionary) at 388-389.

117. The ubiquitous Secure Socket Layer (SSL), used heavily in HTTPS communications, uses public-private keys and certificates to establish a secure session; thereafter, it generates temporary “session keys” that are used for that session, but are never stored anywhere (other than temporary RAM) and are discarded immediately upon the session's termination. SSL is “[a] proposed open standard developed by Netscape Communications for establishing a secure communications channel to prevent the interception of critical information, such as credit card numbers. The primary purpose of Secure Sockets Layer is to enable secure electronic financial transactions on the World Wide Web, although it is designed to work with other Internet services as well. This technology, which uses public key encryption, is incorporated into the Netscape Navigator Web browser and Netscape's commerce servers. *Acronym: SSL*). *See also* commerce server, open standard, public key encryption. *Compare* S-HTTP.” EX1014 (Microsoft Computer Dictionary) at 425-426, 447.

1. Hash Functions and Their Uses

118. Hashing algorithms are mathematical functions that take as input (typically) a long piece of data and produce a shorter output number that can (almost always) uniquely identify the input data. Depending on the specific context, hash

functions are sometimes called checksums, fingerprints, message digests, or digital signature functions.

119. Hash functions are one-way: it is typically considered cryptographically hard to produce the original input data from the hash itself (*e.g.*, invert it). *See* EX1014 (Microsoft Computer Dictionary) at pp. 228 (“**hash**² *vb.* To be mapped to a numerical value by a transformation known as a hashing function. Hashing is used to convert an identifier or key, meaningful to a user, into a value for the location of the corresponding data in a structure, such as a table”), 432 (“**SHA** *n.* Acronym for **Secure Hash Algorithm**. A technique that computes a 160-bit condensed representation of a message or data file, called a *message digest*.”), 122-123 (“**CRC** *n.* Acronym for **cyclical (or cyclic) redundancy check**.... CRC error checking uses a complex calculation to generate a number based on the data transmitted.”).

120. Hashing functions are said to produce a “probabilistically unique” output because there is a small chance that two different inputs may produce the same output hash value. The mathematical probability itself depends on the length of the hash value and the number of items being hashed: the smaller the hash and the more items are hashed, the higher this “collision” probability is. Weaker hash functions often use 32- to 64-bit hashes (*e.g.*, CRC32, MD4); stronger hashes, or

cryptographically strong ones, often produce hashes that are at least 128-160 bits long (*e.g.*, MD5, SHA1).²

121. Well-designed hash functions are supposed to exhibit the “avalanche effect”: small input changes should produce large output changes.³ If a single input bit changed (*e.g.*, flipped), the avalanche effect calls for half of the output hash bits to have changed on average. This means that it should be rare that few or most bits would have changed due to a single input bit change. This property helps ensure the (probabilistic) uniqueness of hash values, their uniformity across the full possible space, and the minimization of collisions.

122. Hash functions have been used for many years in various settings from security, cryptography, integrity, data deduplication, and others. For example, the Tripwire system I personally used in the early 1990’s hashes files and their meta-data and compares those hashes against the files later on; if a mismatch is detected, it means that the file was changed, which could indicate that an intruder broke into the system and installed malware such as Trojans.

² See <https://preshing.com/20110504/hash-collision-probabilities/>, last viewed September 28, 2024.

³ See https://en.wikipedia.org/wiki/Avalanche_effect, last viewed September 28, 2024.

123. In another example, vendors hash known malicious binaries and distribute those to customers. The hashes are small, can easily be distributed, and also do not identify the actual binary (because customers do not like exposing their own files even to their own security vendors). On the customer's computers, existing and incoming binaries can be compared: any matches found would indicate that the customer has or received malware. This malware can be cleaned (if possible), quarantined (*e.g.*, saved in a special folder), deleted, and/or reported.

124. An alternative to handle the zero-day malware is to detect the malicious program by its behavior rather than a hash of the file's contents or parts thereof. For example, a program that reads an entire computer's password file and immediately appears to transmit it over the network is more likely to be malware. (Such behavioral scanning or activity monitoring to detect previously unknown malware using its behavior was the subject of my own work in the late 1990's, which produced several highly cited and award-winning publications, as well as two patents, as I described in §II above.)

125. Hashes are useful beyond security applications. Storage systems and file systems can store numerous files. Many of those files, or parts thereof, can be duplicates (*e.g.*, due to users copying files repeatedly, backup procedures, and more). Storage systems therefore have employed data de-duplication techniques to remove these duplicates: this can save a lot of space and reduce costs. Because files' names

do not have to be unique and can change over time, deduplication systems often use a different identifier for the file's actual data—unique hash computed from the actual data (or parts thereof). In that way, the hash can uniquely identify the file's data regardless of the file's name or location.

E. Virtualization and its Uses in Storage and File Systems

126. Among computer scientists, there is an axiom stating that “you can solve almost any problem by adding another level of indirection.” Indeed, virtualization is a indirection technique that makes one thing seem like another; virtualization often presents a physical-world artifact as one that appears as an illusion or “fake”. For something to be “virtual”, it is “[o]f or pertaining to a device, service, or sensory input that is perceived to be what it is not in actuality, usually as more ‘real’ or concrete than it actually is.” EX1014 (Microsoft Computer Dictionary), p. 497.

127. Indeed, the Microsoft Computer Dictionary includes 35 examples and definitions of virtualization. EX1014 (Microsoft Computer Dictionary), pp. 497-500.

128. In one example, a “virtual machine” is “[s]oftware that mimics the performance of a hardware device, such as a program that allows applications written for an Intel processor to be run on a Motorola chip. *Acronym: VM.*” EX1014 (Microsoft Computer Dictionary), p. 498.

129. In another example, “virtual memory” is “[m]emory that appears to an application to be larger and more uniform than it is. Virtual memory may be partially simulated by secondary storage such as a hard disk. Applications access memory through virtual addresses, are translated (mapped) by special hardware and software onto physical addresses. *Acronym: VM. Also called disk memory. See also paging, segmentation.*” EX1014 (Microsoft Computer Dictionary), p. 498.

130. In yet another example, “virtual reality” is “[a] simulated 3-D environment that a user can experience and manipulate as if it were physical. The user sees the environment on display screens, possibly mounted in a special pair of goggles. Special input devices, such as gloves or suits fitted with motion sensors, detect the user’s actions. *Acronym: VR.*”. EX1014 (Microsoft Computer Dictionary), p. 499.

131. Virtualization, of course, had been applied to storage systems, disk devices, and drivers.

132. A “virtual device driver” is “[s]oftware in Windows 95 that manages a hardware or software system resource. If a resource retains information from one access to the next that affects the way it behaves when accessed (for example, a disk controller with its status information and buffers), a virtual device driver must exist for it.” EX1014 (Microsoft Computer Dictionary), p. 498.

133. A “virtual disk” or “RAM disk” is “[s]hort for random access memory disk. A simulated disk drive whose data is actually stored in RAM memory. A special program allows the operating system to read from and write to the simulated device as if it were a disk drive. RAM disks are extremely fast, but they require that system memory be given up for their use. Also, RAM disks usually use volatile memory, so the data stored on them disappears when power is turned off. Many portables offer RAM disks that use battery-backed CMOS RAM to avoid this problem. *See also* CMOS RAM. *Compare* disk cache.” EX1014 (Microsoft Computer Dictionary), pp. 395-396, 498.

134. Virtualization is sometimes said to be “logical” representation of physical reality. “Logical” is “[o]f or pertaining to a conceptual piece of equipment or frame of reference, regardless of how it may be realized physically. *Compare* physical.” EX1014 (Microsoft Computer Dictionary), p. 288 (“logical” definition 2).

135. A “logical drive” or “logical device” is “[a] device named by the logic of a software system, regardless of its physical relationship to the system. For example, a single floppy disk drive can simultaneously be, to the MS-DOS operating system, both logical drive A and drive B.” EX1014 (Microsoft Computer Dictionary), p. 288.

136. Moreover, one can virtualize files and file systems. This was the subject of much of my graduate-level research and became part of my Ph.D. Dissertation. See §II above and EX1003 (Zadok Curriculum Vitae).

137. The Virtual File System (VFS) is a layer of abstraction inside modern operating systems that abstracts different physical file systems into and presents a single, unified interface to users and applications. EX1016 (Zadok Patent Publication) at ¶[0052] (“The VFS 203 manages the different file systems that are mounted at any given time. The VFS 203 maintains data structures that describe the whole (e.g., virtual) file system and the real, mounted, file systems.”).

138. While normally the VFS resides immediately above the physical (e.g., disk based) file systems that live below it, one can create yet another layer of indirection that looks both like a VFS and a lower-level physical file system. This is called a “stackable file system.” EX1016 (Zadok Patent Publication) at ¶[0036] (“A stackable file system (stacking) is a layered software technology that can be used to wrap another file system, even another stackable file system. By wrapping other file systems, stackable file systems can monitor file system activity that comes from users before that activity is passed on to the actual file system that the stackable file system is wrapping. Accordingly, the network file system is a proxy implementing stacking.”).

139. Stackable file systems enable the easier construction of security layers, such as transparent encryption. “In another example, the stackable file system can transparently encrypt and decrypt data, e.g., from a file, passing through the stack. For example, a user issues a request via a system call to write to a data file; the stackable file system encrypts a data buffer and reissues the write request to the wrapped file system, wherein the stackable file system passes the encrypted data buffer to the wrapped file system. This provides transparent encryption without user intervention.” EX1016 (Zadok Patent Publication) at ¶[0038]; see also ¶¶[0161]-[0185].

140. Another example of stackable file systems’ usefulness is to transparently detect intrusions. See EX1016 (Zadok Patent Publication) at ¶¶[0051]-[0055]. “[A] file system IDS monitors messages for predetermined behavior indicative of a security event. For example, repeated attempts to delete a large number of files, attempts to access unauthorized files (‘snooping’), and attempts to copy large amounts of proprietary data, even by authorized users who deviate from a predetermined usage pattern.” EX1016 (Zadok Patent Publication) at ¶[0054].

141. “The file system IDS has full access to file data and operations. Therefore, file system operations of the wrapped file system can be accurately observed and filtered. Further, the file system IDS can check the validity of the data before permitting it to be written to a server or read by a user. Checking validity

includes checking integrity using checksum methods (e.g., SHA1 and MDS), as well as Virus Protection (e.g., inspecting file data for known and unknown patterns indicating that the file contains a virus, other malicious code, or any other undesirable patterns).” EX1016 (Zadok Patent Publication) at ¶[0055].

142. In yet another example, stackable file systems were created to detect and quarantine viruses in files. “Another example of an application of a stackable file system is an anti-virus stackable file system (AVFS). The AVFS can add virus detection to other file systems including Ext3, NFS, etc. AVFS supports forensic modes that can prevent a virus from reaching the disk, automatic versioning of potentially infected files to allow safe recovery, quarantining of known infected files, and isolation of infected files from user processes.” EX1016 (Zadok Patent Publication) at ¶¶[0072], [0071]-[0083].

143. Finally, I note that data encryption can take place in almost any layer in the storage stack, not only as in the “stackable file system” form just discussed. Indeed, well before 2010, various data and file encryption techniques were well known in the art (and some were subject of my own research). By 2003, a wide range of techniques were available for data/file encryption methods: for an entire MS-DOS partition using a device driver, as user-level NFS server, as a modified in-kernel NFS client, as a virtual loopback driver, as part of Microsoft NTFS (EFS), and as a modified Linux Ext2 disk-based file system. Moreover, these encrypting

file systems could use user-supplied keys, randomly generated ones, ones derived from fixed hardware IDs, and more. *See* EX1018 (NCryptfs), pp. 2-3.

VIII. OVERVIEW OF THE '749 PATENT

A. The '749 Specification

144. The '749 patent discloses a “system (and a method) ... for generating a virtual secure digital (SD) card.” '749 patent, Abstract. The specification states that “[s]ecure Digital (SD) cards **are in wide use for many storage applications**,” and that “[t]hey are most commonly used to store audio (MP3 players), pictures (digital cameras), and full motion video (camcorders and HD camcorders).” '749 patent, 1:17-20.⁴ The specification further notes that SD cards “provide[] an acceptable portable storage medium to transfer unprotected content between different devices, especially for the video capture devices mentioned above.” *Id.*, 1:20-23.

145. The '749 patent also notes that while “[c]onsumers are familiar with the SD card’s basic storage capabilities, ... they are mostly unaware of its security features (the S in Secure Digital).” *Id.*, 1:29-31. The reason for this, according to the patent, is that “there are very few applications that use this advanced functionality.” *Id.*, 1:31-32. The patent states that digital rights management (“DRM”) is an example of such an application. *Id.*, 1:32-35.

⁴ All **bold**/*italics*/underline have been added unless noted.

146. According to the '749 patent, "SD cards are suited to content loading, purchasing and transporting content, but not for creating a digital library." *Id.*, 1:43-45. The patent identifies several reasons for this, including that SD cards have insufficient storage size, lack labeling, and have high cost. *Id.*, 1:45-58 ("Accordingly, the size, cost, and the lack of labeling on the SD card result in an undesirable format for a digital library"). Moreover, the '749 patent recognizes that SD cards use an industry standard DRM known as "content protection for recordable media," or "CPRM," and that while CPRM specifies how DRM-protected content can be transferred to other CPRM-complaint devices, it does not provide any standardized method for transferring such content to non-CPRM complaint devices. *Id.*, 1:59-2:25.

147. The '749 patent discloses provisioning a virtual SD card by detecting an SD card, reading its media key block and media identification (discussed further next), and storing each on the virtual card. The system also creates a file system for secure data storage corresponding to the SD card's secure data area and creates a file system for user data for storage of user data in an SD card's user data area. *Id.*, 3:4-13. Fig. 2 discloses an exemplary system capable of reading and writing to SD cards:

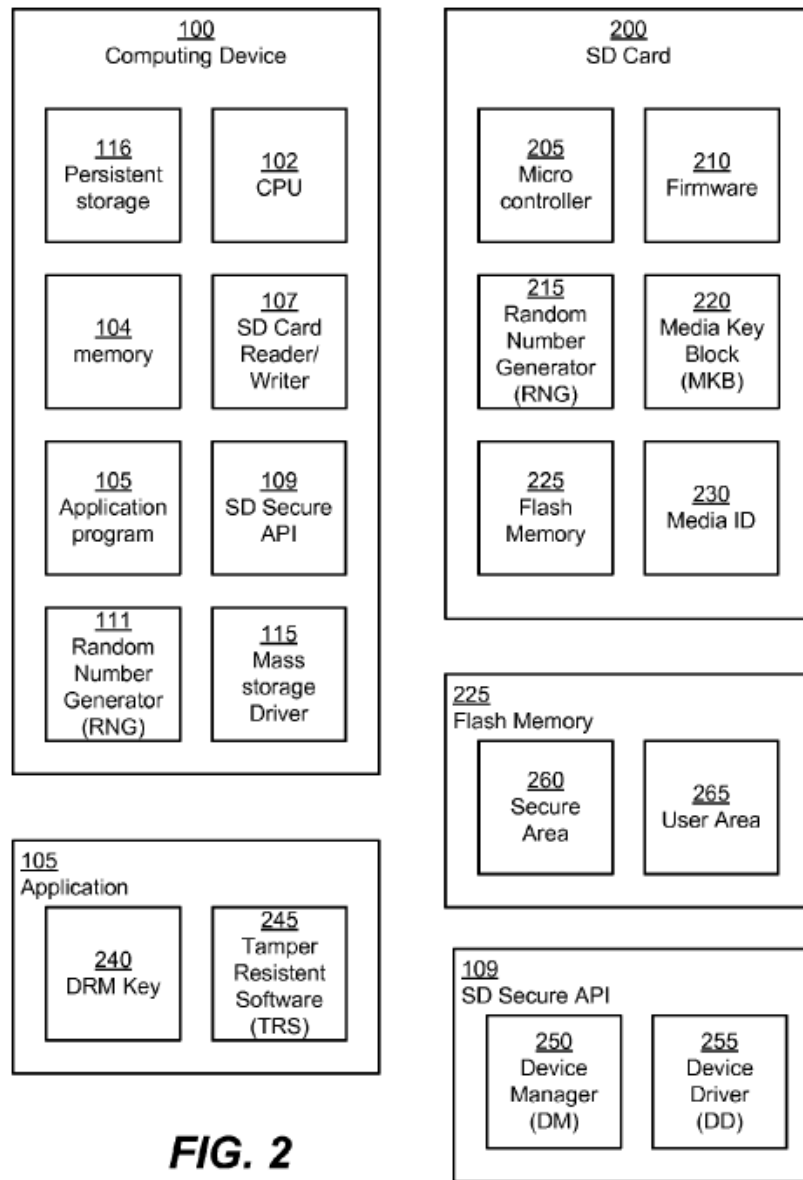


FIG. 2

148. As can be seen, the system includes a computing device 100 containing, *inter alia*, an SD card reader/writer 107.

149. Fig. 2 above also shows SD card 200, and is described in a section of the specification entitled “Typical SD Card Architecture and Functions.” *Id.*, 4:43. According to the ’749 patent, “SD card 200 must conform to the specifications

developed and enforced by **the SD ASSOCIATION.**” *Id.*, 4:45-46. POSITAs would have been aware of the SD Association as the membership organization that developed the specifications for SD cards. See also §VII.B.1 above.

150. I understand the discussion regarding “typical” SD card as the inventors admitting that an SD card “includes a microcontroller 205, firmware 210, a random number generator (RNG) 215, a media key block (MKB) 220, flash memory 225, and a media identifier (MID) 230.” ’749 patent, 4:56-60. Indeed, by the time of the priority date, POSITAs would have known the features the ’749 patent states were in “typical” SD cards were in fact contained within SD cards.

151. The ’749 patent further admits that in “typical” SD cards, “[d]ata is stored within the flash memory 225, which is partitioned into two areas: Secure Area 260 and User Area 265.” *Id.*, 4:60-62. The patent further states that in “typical” SD cards, “secured area 260 may be implemented as a physical or logical device within the SD card as chosen by the SD card’s original equipment manufacturer (OEM),” with secure area’s 260 size being “determined at design time **according to the SD card specifications.**” *Id.*, 4:63-5:1.

152. The ’749 patent further admits “that every SD card is provisioned with a secure area 260, even though it may never be used by any application,” and also admits that the user area 265 of a “typical” SD card “may be implemented in the

same manner as the secure area 260,” and “also may be either physical or logical partitions depending on implementation within the SD card.” *Id.*, 5:1-6.

153. The '749 patent also admits to several other features of “typical,” *i.e.*, prior art, SD cards. The patent admits that in such “typical” SD cards, DRM key 240 is stored in the computer system 100 “and used to complete an authenticated key exchange with the SD card 200.” *Id.*, 5:42-44. In particular, the '749 patent admits that in SD cards, “the authenticated key exchange is used to gain access the SD card secure area 260 in accordance with the Authenticated Key Exchange (AKE) used by Content Protection for Recordable Media (CPRM) specifications for use with SD cards.” *Id.*, 5:44-49. As of the priority date, POSITAs would have known that “CPRM” specifications were promulgated by the “4C Entity,” which “was formed in 1999 to address the interoperability challenge of sharing premium content among a wide range of digital devices.” <http://www.4centity.com/about.aspx> (EX1013, 4C Entity - About Us.)

154. The '749 patent also admits that “typical” SD cards have a random number generator (RNG) 215 that “can be implemented in either hardware or software,” and that a “random number generator 111 in the computer system 100 and the random number generator 215 in the SD card 200 are both used during AKE.” '749 patent, 5:66-6:3. Finally, the '749 patent admits that “typical” SD cards

have a “media key block (MKB) 220 [that] is structured in accordance with the CPRM specification for SD Cards.” *Id.*, 6:3-4.

155. The architecture of the '749 patent's virtual SD card is shown in Fig. 3:

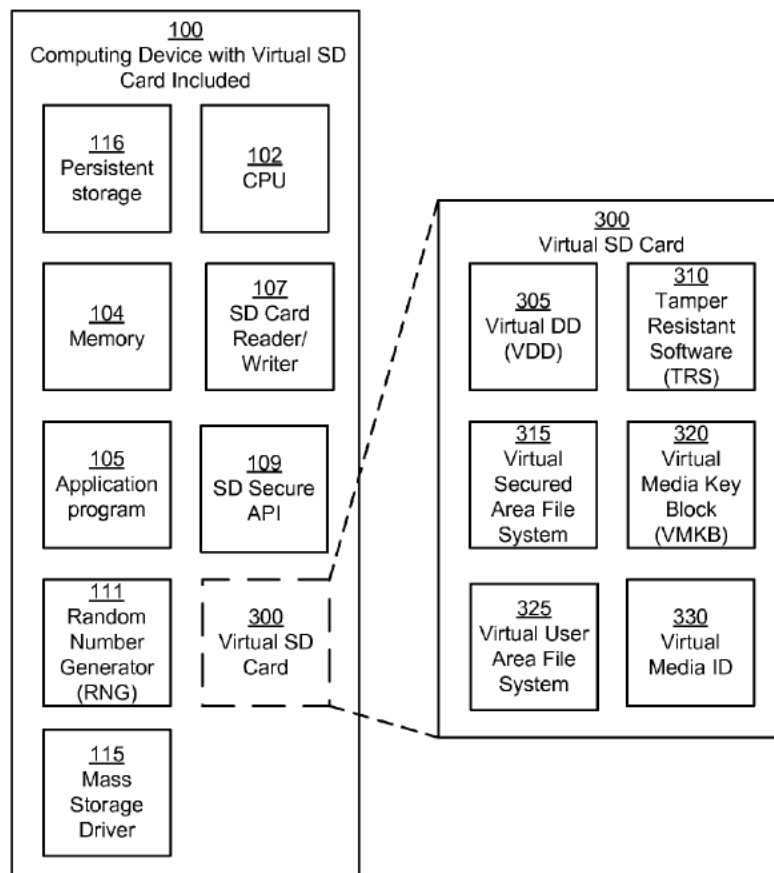


FIG. 3

156. According to the '749 patent, its “virtual SD card 300 implements in software the functionality previously implemented by SD card 200 in hardware.” '749 patent, 6:6-8. Virtual SD card 300 “includes a virtual device driver (DD) 305, tamper resistant software 310, a virtual secure area filed [sic: file] system

315, a virtual media key block 320, virtual user area file system 325, [and] a virtual media identification (ID) 330.” *Id.*, 6:37-40. Virtual secure area file system 315 and virtual user area file system 325 can either “be implemented as file systems within an encrypted file” or “as physical or logical partitions on the same or separate devices, with or without encryption depending on the overall system security requirements.” *Id.*, 6:50-56.

157. The ’749 patent states that “secured area file system 315 is configured as a storage area for secure data.” *Id.*, 6:64-65. To protect the data in that secured storage area, the data is encrypted using an encryption key. This encryption key “is derived from the computing device’s unique information (e.g., processor serial number, bios serial number, device serial number), and hidden information effectively binding the file systems to the computer system 100” in which the virtual SD card is operating. *Id.*, 7:2-6.

158. The method for provisioning the virtual SD card is shown in Fig. 4:

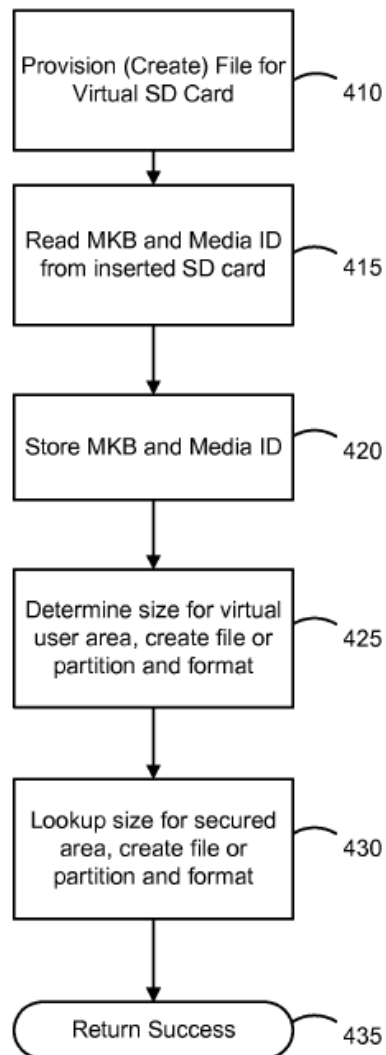


FIG. 4

See also '749 patent, 7:45-58.

159. According to the '749 patent, “[t]he process determines 425 the physical or logical devices to use, size for a virtual user area 325 on the virtual SD card 300 and creates a file or partition on the storage 116 that is accordingly formatted for the virtual user area 325. A size of the secured area file system 315 is

determined at provisioning time according to the SD card specifications. The size of the virtual secured area file system 315 is a table lookup from **SD ASSOCIATION specifications** based on the size of the virtual user area file system 325. The process creates a file or partition on the storage 116 that is accordingly formatted for the virtual secured area 315. Once completed the process is returned as successful 435. With the virtual SD card 300 provisioned, it is now ready for use.” ’749 patent, 7:59-8:4.

B. Prosecution History

160. The ’749 patent was filed on April 1, 2011, and claimed the benefit of provisional application 61/413,926, which was filed on November 15, 2010. EX1004 (’749 File History). For purposes of this petition only, I have been asked to assume the priority date is November 15, 2010. The original application contained 27 claims. On July 1, 2013, the Examiner issued an office action. In addition to certain rejections under 35 U.S.C. §§ 101 and 112, the Office Action rejected claims 1, 10, 15-18 19, and 24-27 as obvious under 35 U.S.C. § 103, and indicated that claims 2-9 and 11-14 would be allowable if rewritten in independent form. EX1004 (XXX-NAME) at 74-88. In the Examiner’s “statement of reasons for the indication of allowable subject matter,” the Examiner indicated the following:

The technique of memory allocation as set forth in claim
2 wherein a lookup table is utilized to allocate memory

shared between user data and secure data in a virtual SD card appears to be novel and unobvious.

Id. at 83.

161. The application's claim 2 recited that "a size of the file system for secure data is determinant on a size of the file system for user data, the determination based on a lookup table providing a relationship between the sizes." *Id.* at 174.

162. On October 1, 2013, patentee responded to the Office Action by amending the application's claim 2 to include the limitations of the application's claim 1, and amending independent application claims 15, 19 and 24 to include the application claim 2's limitations. *Id.* at 45-54. *Id.* at 174. On January 24, 2014, a Notice of Allowance was issued.

IX. THE CLAIMS

163. The tables below include identifiers for the '749 patent claims and limitations, which are referenced throughout this Declaration.

Reference	Claim Limitation
Claim 1	
[1pre]	1. A method for generating a virtual secure digital (SD) card, the method comprising:
[1a]	reading a media key block and media identification for an SD card;
[1b]	storing the media key block and the media identification;

Reference	Claim Limitation
Claim 1	
[1c)]	creating a file system for secure data on a storage device for storage of secure data corresponding to a secure data area of the SD card; and
[1d]	creating a file system for user data on the storage device for storage of user data corresponding to a user data area of the SD card,
[1e]	wherein a size of the file system for secure data is determinant on a size of the file system for user data, the determination based on a lookup table providing a relationship between the sizes.

Claim 2	
[2]	2. The method of claim 1, wherein the file system for secure data is encrypted.

Claim 3	
[3]	3. The method of claim 2, wherein the file system is encrypted with an encryption key, the encryption key derived from computer system information and the encryption key not stored by the computer system.

Claim 4	
[4]	4. The method of claim 2, wherein the file system is encrypted with an encryption key, the encryption key derived from a computer system and stored by the computer system.

Claim 5	
[5]	5. The method of claim 2, wherein the file system is encrypted with an encryption key, the encryption key derived from a computer system and stored by the computer system within a trusted host.

Claim 6	
[6]	6. The method of claim 2, wherein the file system for secure data and the file system for user data are located on different physical partitions of the storage device.

Claim 7	
[7]	7. The method of claim 2, wherein the file system for secure data and the file system for user data are located on different logical partitions of the storage devices.

Claim 8	
[8]	8. The method of claim 2, wherein the file system for secure data and the file system for user data are within an existing file system.

Claim 13	
[13]	13. The method of any one of claims 1 through 5, wherein the media identification comprises random data not allocated from the SD card.

Claim 14	
[14pre]	14. A method using a virtual secure digital (SD) card, the method comprising:
[14a]	determining if the virtual SD card is provisioned;

Claim 14	
[14b]	provisioning the virtual SD card in response to the virtual SD card not being provisioned, the provisioning including:
[14c]	creating a file system for secure data on a storage device for storage of secure data corresponding to a secure data area of the SD card; and
[14d]	creating a file system for user data on the storage device for storage of user data corresponding to a user data area of the SD card,
[14e]	wherein a size of the file system for secure data is determinant on a size of the file system for user data, the determination based on a lookup table providing a relationship between the sizes;
[14f]	accessing the data stored in the secure area of the provisioned virtual SD card; and
[14g]	extracting the data from the secured area of the provisioned virtual SD card.

Claim 15	
[15]	15. The method of claim 14, further comprising performing an authentication using Authenticated Key Encryption (AKE) before accessing the data in the secure area of the provisioned virtual SD card.

Claim 16	
[16]	16. The method of claim 14 or 15, wherein data received from an application program through Authenticated Key Encryption (AKE) is first decrypted.

Claim 17	
[17]	17. The method of claim 14, wherein data received from an application program through Authenticated Key Encryption (AKE) is encrypted with another key before storing the data in the secure area of the provisioned virtual SD card.

Claim 18	
[18pre]	18. A non-transitory computer readable medium configured to store instructions that, when executed, cause at least one processor to:
[18a]	read a media key block and media identification for an SD card;
[18b]	store the media key block and the media identification;
[18c]	create a file system for secure data on a storage device for storage of secure data corresponding to a secure data area of the SD card; and
[18d]	create a file system for user data on the storage device for storage of user data corresponding to a user data area of the SD card,
[18e]	wherein a size of the file system for secure data is determinant on a size of the file system for user data, the determination based on a lookup table providing a relationship between the sizes.

Claim 19	
[19]	19. The computer readable medium of claim 18, wherein the file system for secure data is encrypted.

Claim 20	
[20]	20. The computer readable medium of claim 19, wherein an encryption key is derived from a computer system information, the encryption key not stored by the computer system.

Claim 21	
[21]	21. The computer readable medium of any one of claims 18, 19, or 20, wherein the encryption key is derived from a computer system and stored by the computer system.

Claim 22	
[22pre]	22. A non-transitory computer readable medium configured to store instructions that, when executed, cause at least one processor to:
[22a]	determine if a virtual SD card is provisioned;
[22b]	provision the virtual SD card in response to the virtual SD card not being provisioned, wherein to provision the virtual SD card includes to:
[22c]	create a file system for secure data on a storage device for storage of secure data corresponding to a secure data area of the SD card; and
[22d]	create a file system for user data on the storage device for storage of user data corresponding to a user data area of the SD card,
[22e]	wherein a size of the file system for secure data is determinant on a size of the file system for user data, the determination based on a lookup table providing a relationship between the sizes;
[22f]	access the data stored in the secure area of the provisioned virtual SD card; and
[22g]	extract the data from the secured area of the provisioned virtual SD card.

Claim 23	
[23]	23. The computer readable medium of claim 22, further comprising: performing an authentication using Authenticated Key Encryption (AKE) before accessing the data in the secure area of the provisioned virtual SD card.

Claim 24	
[24]	24. The computer readable medium of claim 23, wherein data received from an application program through Authenticated Key Encryption (AKE) is first decrypted.

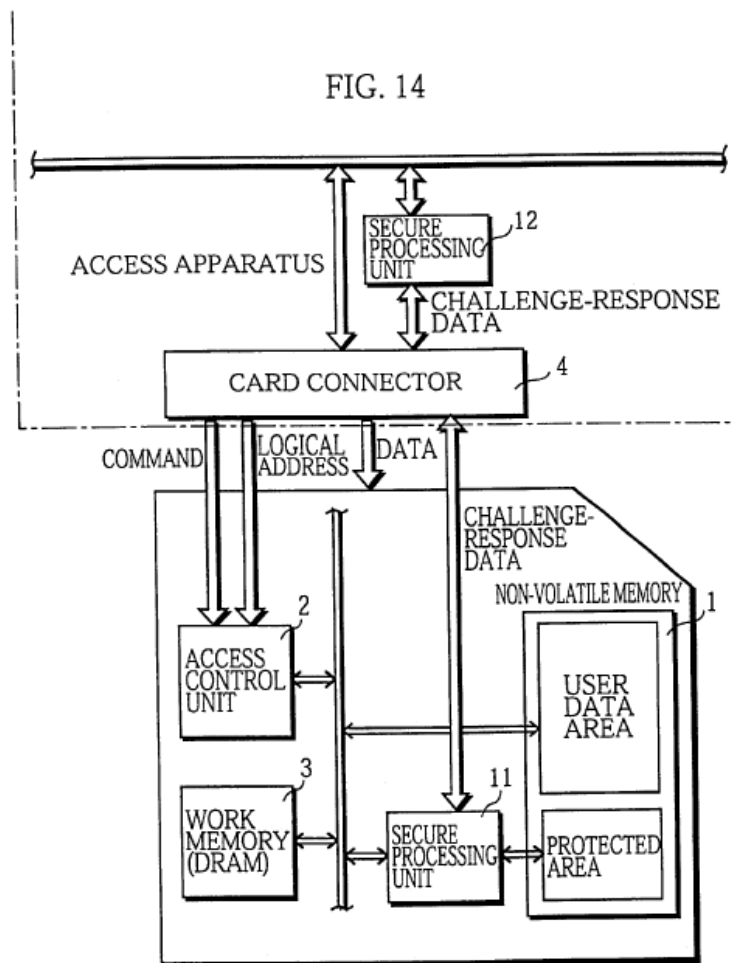
Claim 25	
[25]	25. The computer readable medium of claim 23, wherein data received from an application program through Authenticated Key Encryption (AKE) is encrypted with another key before storing the data in the secure area of the provisioned virtual SD card.

X. OVERVIEW OF THE PRIMARY PRIOR ART REFERENCES

164. I first note that the prior art references analyzed in the grounds below are analogous to the '749 patent because they are within the same field of endeavor and/or reasonably pertinent to one or more problems addressed by the '749 patent. Like the '749 patent, Hamann is directed to virtualizing of memory cards. Like the '749 patent, each of Maeda, the Applicant Admitted Prior Art (AAPA), Hirota, and Matsukawa are directed to SD cards. Finally, like the '749 patent, Lu describes generating encryption keys.

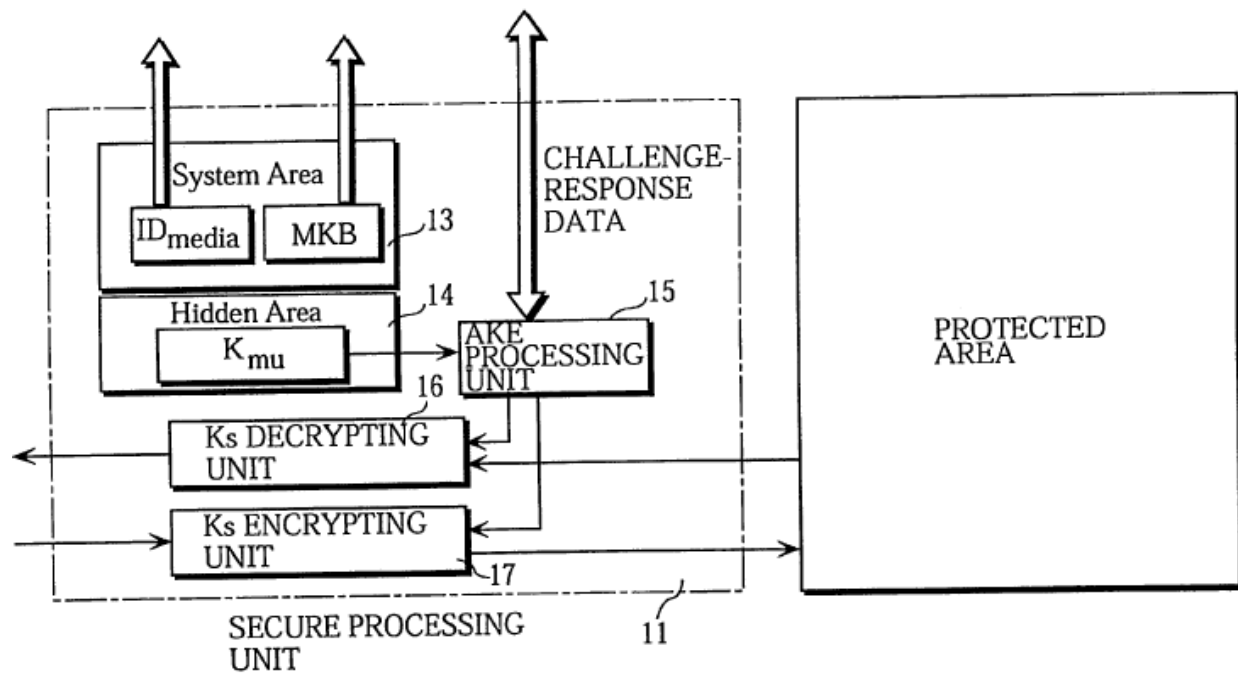
A. Maeda

165. POSITAs would have recognized that Maeda (EX1006) discloses the general architecture of SD Cards.⁵ A simplified illustration of this architecture is shown in Fig. 14:



⁵ Indeed, I note that *Maeda* has been designated as an “essential patent” by the SD Card Association. See EX1011, Intellectual Property (IP) Information from the SD Card Association (SD-3C LLC) (“<https://www.sd-3c.com/IPInformation.aspx>”).

166. As seen in its Fig. 14, Maeda discloses that SD cards have a non-volatile memory storage area where a “user can gain access (the user data area), and an area that stores data securely (the protected area).” EX1006, Maeda, 13:30-33. As is also seen, this SD card also has a secure processing unit 11, the details of which are shown in its Fig. 15:



EX1006, Maeda, Fig. 15.

167. As seen in Fig. 15, SD cards have a “system area 13,” in which Maeda discloses that SD cards read and store a media key block and media ID from an “access apparatus 200.” *Id.*, 13:49-54 (“The system area 13 is a read-only area storing a media key block (MKB) and a media-ID.”).

168. Maeda further discloses that before accessing the secure area of an SD card, AKE processing unit 15 performs “authentication and key exchange,” or “AKE.” *Id.*, 13:62-14:3 (“The AKE (authentication and key exchange) processing unit 15 performs mutual authentication between the access apparatus 200 and the semiconductor memory card 100 using the challenge-response method, verifies the authenticity of the opposing device, and if the opposing device is invalid, stops processing. If the opposing device is valid, however, an encryption key (session key Ks) is shared by the access apparatus 200 and the semiconductor memory card 100.”).

169. Maeda also discloses that data stored within the secure data file system is encrypted. *Id.* at 14:57-62 (“When encrypted data to be written into the protected area is output from the access apparatus 200 connected to the semiconductor memory card 100, the Ks decrypting unit 16 uses the session key Ks to decrypt this data which has already been encrypted by session key Ks before being output.”); *see also id.*, 13:62-67, 14:3-31.

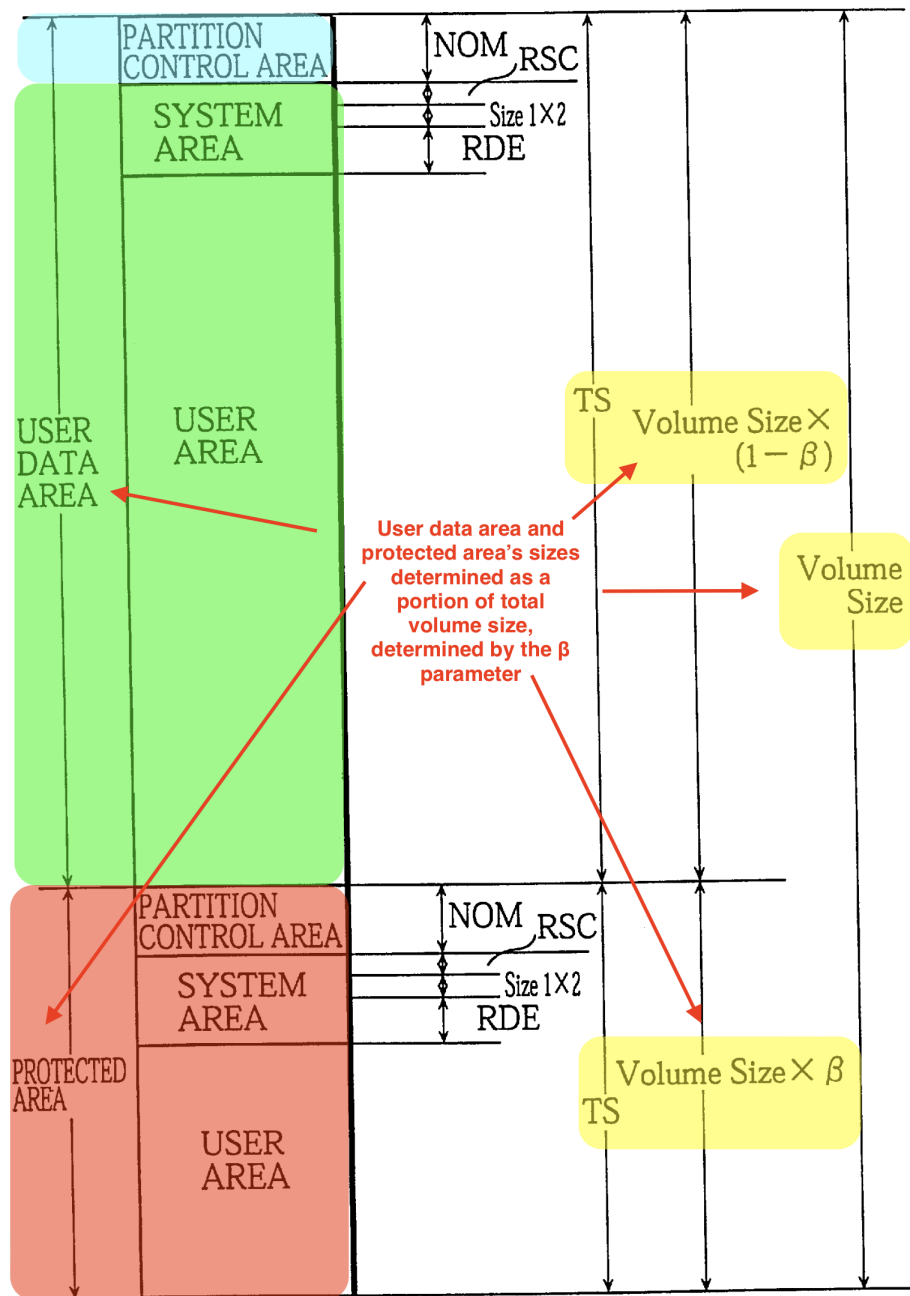
170. Maeda further discloses that the size of the file system for secure data and the file system for user data are calculated such that they each have a size that is relative to the other.

171. Maeda discloses that its user and protected areas are configurable. Maeda states: “[n]ext, the format of the volume area in the second embodiment is

explained, with reference to the detailed drawing of the volume area including the protected area shown in FIG. 16. In the drawing, if the total size of the volume area is 'volume size', the user data area is an area 'volume size $\times$ (1- β)' extending from the start of the volume area, and an area 'volume size $\times\beta$ ' following the user data area is allocated to the protected area. Here, β is a value for which the minimum setting is 0.01. To take one example, if the volume size is 64 MB and β is 0.01, the protected area is set at 640 KB. The format of the user data area and the protected area is realized when the file system operation unit 9 executes the initialization procedure shown in FIG. 9." EX1006, Maeda, 15:12-21. A POSITA would have understood that the sizes of the protected and user areas are configurable, and that one example ratio is that the protected area's size is 0.01 (*i.e.*, 1%) of the user's area size.

172. Maeda's Fig. 16 (annotated below) further illustrates this, where the β parameter (in yellow) controls the ratios of the user data area (green) and protected area (red):

FIG. 16



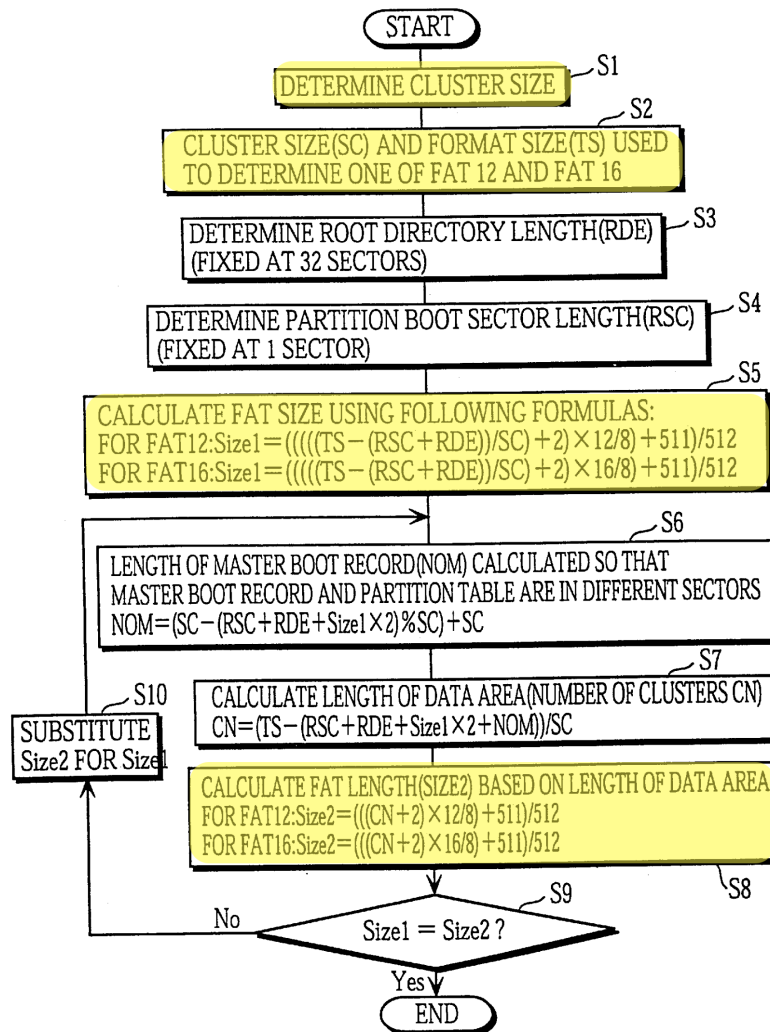
173. A POSITA would have understood the above figure to mean that since the size ratios are configurable and parameterized, there could easily be multiple

possible ratios for different sizes, especially as SD cards and Smart cards came in many different sizes. *See also* §VII.B.1 above.

174. Furthermore, Maeda, using a typical FAT-based file systems, also uses a typical partition table which determines the number and positions of the partitions. See “partition control area” annotated in blue in Fig. 16 above. A POSITA would have understood that FAT file systems include partitions that reside on the same device.

175. Maeda further discloses that the data size area, and hence the protected area’s size, also depend on which file system is being used to format the volume/partition. Maeda discloses that the volume can be partitioned using FAT12 or FAT16. A POSITA would know that FAT12 has a maximum size of 16MB (using 4KB size clusters) or 32MB (using 8KB size clusters); FAT16 has a maximum size of 4GB. See https://en.wikipedia.org/wiki/File_Allocation_Table (last viewed September 30, 2024). Therefore, there are three possible FAT file system sizes depending on the type and cluster size. These can be seen in at least the annotated steps of Maeda’s Fig. 13 (annotated below):

FIG. 13



176. As Fig. 13 shows above, steps S5 and S8 includes at least one if-then-else decision based on the FAT type; and many steps also depend on the cluster size (SC). A POSITA would have found it obvious and understood that the decisions based on the FAT type (and cluster size) could easily be determined also with a lookup table, especially if additional FAT types were desired (e.g., 8-bit FAT, FAT16B, or FAT32, all of which were in wide use well before 2010). Lastly, such

a table could easily accommodate additional determinations based on different desired β values.

177. *See also id.* at 15:12-21, 17:1-11, Figs. 16, 17, and Claim 7.

178. Finally, I note that, Matsushita, the assignee of Maeda, was a founding member of the SD association. See https://en.wikipedia.org/wiki/SD_Association (last viewed September 28, 2024.).

B. Hamann

179. Hamann describes a system for the “secure back-up and usage of secure copies of smart card data objects, especially in the case when the smart card is lost or damaged or data objects stored on the smart card are not accessible or destroyed.” EX1007, Hamann, ¶[0001]. Hamann’s method “provid[es] a virtual smart card (VSC) having the same defined logical file structure and the same content of data objects as its assigned real smart card....” *Id.*, ¶[0014]. Hamann describes a smart card as “any programmed data carrier used in any portable device, like mobile phone, digital personal assistant etc., to securely hold subscriber specific information” (*id.*, ¶[0002]) and a virtual smart card as “a software implemented version of a real smart card providing the equivalent functionality of a real smart card” (*id.*, ¶[0015]).

180. To generate its virtual smart card, Hamann uses “a VSC control program handling the creation, the security and the read/write process of the VSC.” *Id.*, ¶[0015]. The result is a “VSC having a logical file structure comprising a public

area, a private area, a secure key area, password area, and an [sic] unique identifier area.” *Id.*, ¶[0016]. Hamann’s file system for its virtual smart card is shown in Fig. 1 (annotated below):

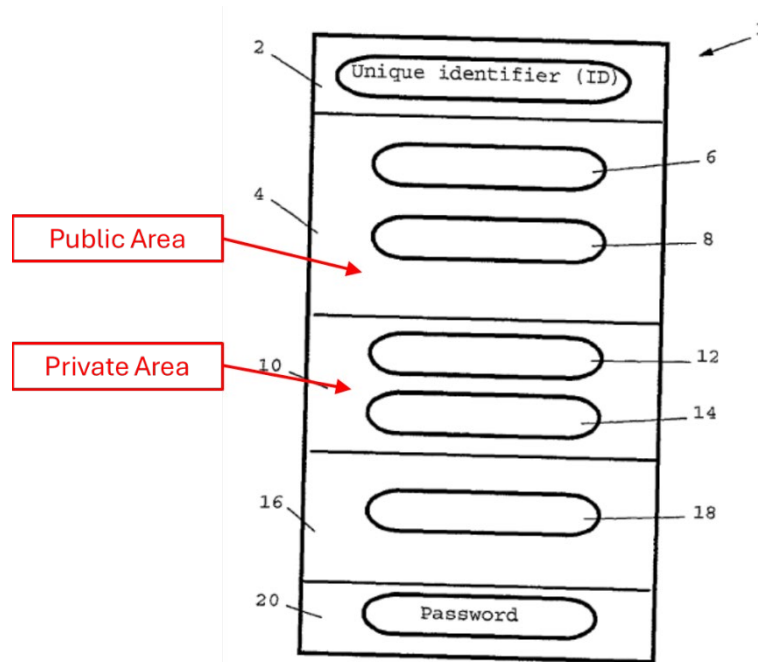


FIG. 1

181. As can be seen, the file system for Hamann’s virtual smart card includes a public area and a private area. The public area stores “public data objects having no access conditions,” while the private area stores “private objects” that have been encrypted. *Id.*, ¶¶[0025]-[0027] (“[0025] The logical file structure of the VSC (1) is preferably defined by the following data areas: [0026] (a public area (4) in which public data objects having no access conditions are placed, e.g. Certificate (6) and address (8) [0027] (a private area (10) in which private objects being encrypted are

placed; private objects may only be accessed providing a password (10), e.g. account no (12) and key information (14)').

C. Applicant Admitted Prior Art (AAPA)

182. The '749 patent (EX1001) admits that SD cards "are in wide use for many storage applications." '749 patent, 1:17-18. The '749 patent further admits that SD cards "must conform to the specifications developed and enforced by the SD ASSOCIATION." *Id.*, 4:45-47. Even putting aside the '749 patent's many admissions regarding the structure, function and operation of SD cards, POSITAs reviewing the patent would have known that the SD Association existed before the '749 patent's priority date, as did its SD card specifications. *See* EX1019 (<https://www.sdcard.org/about-sda/fact-sheet/>, last accessed on October 24, 2024).

183. As I discuss in §VIII.A above, the '749 patent describes what it admits is the "**Typical** SD Card Architecture and Functions" ('749 patent, 4:43-6:4) and the fact that SD cards "**must** conform to the specifications developed and enforced by the SD ASSOCIATION" ('749 patent, 1:17-18). Indeed, the fact that the '749 patent states that SD cards "**must** conform" to the SD Card Association's specifications further demonstrates that POSITAs would have understood that the '749 patent's description of various features of SD cards were already in the prior art.

184. The '749 patent provides extensive details regarding “typical” SD cards, as I describe in §VIII.A above. The '749 patent contains the following additional admission:

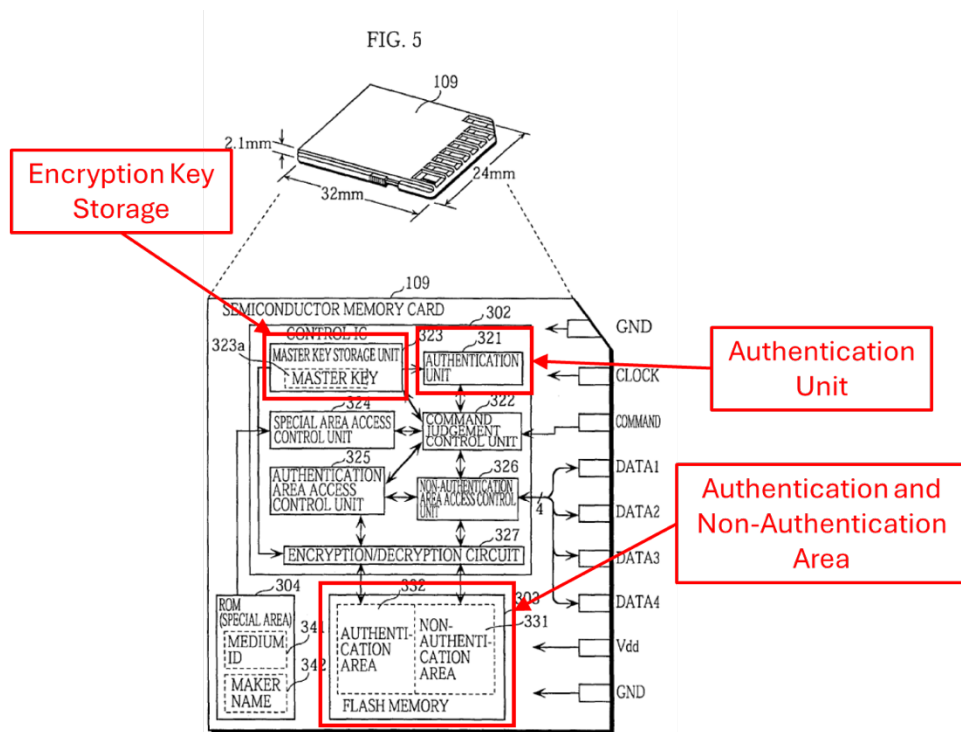
A size of the secured area file system 315 is determined at provisioning time according to the SD card specifications. The size of the virtual secured area file system 315 is a table lookup from SD ASSOCIATION specifications based on the size of the virtual user area file system 325. The process creates a file or partition on the storage 116 that is accordingly formatted for the virtual secured area 315.

'749 patent, 7:62-8:2.

185. I note that to a POSITA, having a “specification” defined by an “association” strongly suggests that such a specification was standardized by all member organizations of the association. Indeed, the SD Association was founded in 2000 by SanDisk, Panasonic (Matsushita), and Toshiba. See https://en.wikipedia.org/wiki/SD_Association (last viewed September 28, 2024.). Over my years of experience, I have observed and participated in several standardization processes; I noticed that organizations, who are often competitors, will form an associations in order to standardize some technology, mostly when they believe it would be mutually beneficial.

D. Hirota

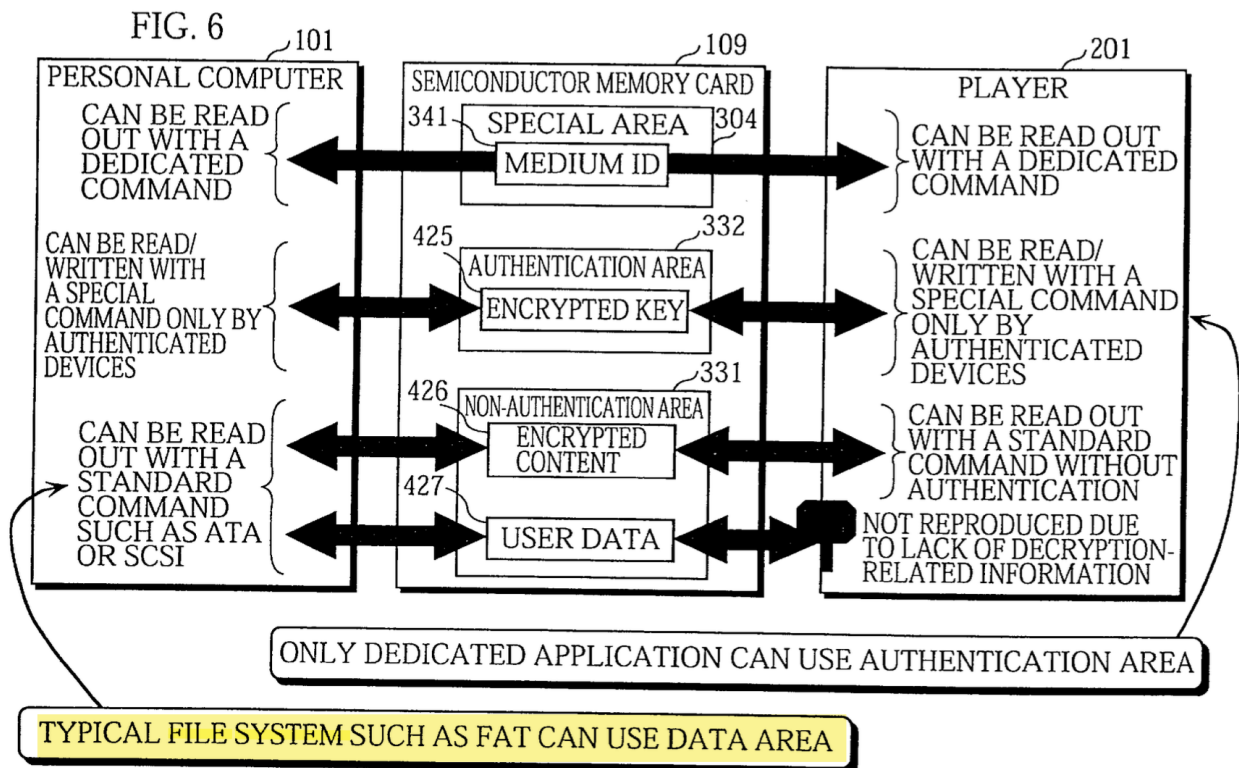
186. Hirota (EX1008) discloses “a semiconductor memory card that can be used as a storage medium for storing digital contents and as a storage medium for storing general-purpose computer data (not an object of copyright protection).” EX1008, Hirota at 2:3-7.⁶ Similar to Maeda, Hirota discloses that its semiconductor memory card includes flash memory 303 having “an authentication area and a non-authentication area,” which is shown in Fig. 5 (annotated below):



⁶ Note that like *Maeda*, *Hirota* has also been designated as an “essential patent” by the SD Card Association. See <https://www.sd-3c.com/IPInformation.aspx>, EX1011.

187. See also EX1008, Hirota, 10:27-36 (“The authentication area 332 can be accessed only by the apparatuses that have been authenticated as proper apparatuses. The non-authentication area 331 can be accessed by any apparatuses whether they are authenticated or not. In the present embodiment, the authentication area 332 is used for storing important data related to copyright protection, and the non-authentication area 331 is used as an auxiliary storage apparatus in a typical computer system.”).

188. Hirota’s Fig. 6 (annotated below) provides additional information regarding its storage architecture:



189. As can be seen, Hirota's non-authentication area 331 includes both encrypted content 426 and user data 427. *See also* EX1008, Hirota, 12:47-51. Hirota also discloses that the encryption key 425 for encrypted content 426 is stored in authentication area 332. *Id.*, 12:51-54 ("The authentication area 332 stores an encryption key 425 which is a secret key used for decrypting the encrypted content 426 stored in the non-authentication area 331.").

190. Hirota discloses using file systems such as FAT. Hirota also suggests that a "typical file systems such as FAT can use data are." *Id.*, Fig. 6 (annotated above). EX1008, Hirota, 21:41-47 ("FIG. 14C is a flowchart showing the procedure of the PC 102 or the player 201 for deleting blocks beforehand using the not-deleted block list command and the delete command in the above-stated states. It is presumed here that the flash memory 303 contains a table such as **FAT (File Allocation Table)** which shows the use state of the logical blocks, as shown in FIG. 14D."), 9:23-28 ("The PC 102 can perform the following: (1) use the memory card 109 as an auxiliary storage apparatus having an independent **file system (e.g., ISO9293) as hard disks have by executing the file management software program** stored in the hard disk 120").

191. Next, I note that while Hirota does not use the term "SD Card", a POSITA would have clearly known that Hirota discloses one for at least three reasons.

192. First, Matsushita, the assignee of Hirota, was a founding member of the SD association. See https://en.wikipedia.org/wiki/SD_Association (last viewed September 28, 2024.).

193. Second, Fig. 5 clearly illustrates an SD card with its right dimensions, pinning, wedge-shape design, and write-protect switch.

194. Third, Hirota discloses the exact dimensions and specifications that became the standard SD card. *Id.*, 10:13-22 (“The memory card 109 contains a rewritable nonvolatile memory to which data can be written repeatedly. The rewritable nonvolatile memory ... is driven by power supply voltage of 3.3V and a clock signal supplied from external sources. The memory card 109 is a 2.1 mm-thick, 24 mm-wide, and 32 mm-deep rectangular parallelopiped. The memory card 109 is provided with a write-protect switch on its side, and is electrically connected to an external apparatus via a 9-pin connection terminal formed at an end of the memory card 109.”). Compare with https://en.wikipedia.org/wiki/SD_card (last view September 27, 2024).

E. Lu

195. Lu (EX1009) discloses “encryption and decryption methods for implementing hardware and software binding.” EX1009, Lu at 1. Lu discloses that to effect this binding, an encryption key is derived from the computer system. See *id.* at Abstract; see also *id.* at 5 (“The present invention includes two parts:

encryption and decryption. First, a hardware chip stores a unique serial number (HW_UID) that represents the identity of the chip. A length of the serial number is determined by the manufacturer, but it must not be repeated. The following describes in detail how the unique serial number is used to implement the binding of software and hardware from two aspects: encryption and decryption.”).

F. Matsukawa

196. Matsukawa (EX1010) discloses an authentication method for use in memory devices, including SD cards. EX1010, Matsukawa, ¶¶[0003], [0007]. Matsukawa states that SD cards and host computers “generate the same session key for each authentication process between the host computer and the recording medium (SD card).” *Id.*, ¶[0007]. To do so, Matsukawa discloses a “media ID generating module 130 [having] the function of generating the media ID in such a manner as not to duplicate a media ID generated in the past.” *Id.*, ¶[0051].

197. I note that during prosecution, the Examiner rejected claim 13, asserting that Matsukawa discloses claim 13’s subject matter. I further note that patentee never asserted that the Examiner’s rejection of claim 13 under Matsukawa was incorrect; as seen from the above, I do not believe such an argument would have succeeded since Matsukawa discloses the recited subject matter. In particular, as discussed in §VIII.B above, patentee overcame the prior art rejections by amending

all the independent claims to recite that the “size of the file system for secure data” is determined “based on a lookup table....”

XI. CLAIM CONSTRUCTION

198. Here, I have given all the claim terms of the challenged claims their plain and ordinary meaning, as would have been understood by a person having ordinary skill in the art at the time of the alleged invention, having taken into consideration the language of the claims, the specification, and the prosecution history of record. Below, I explain how POSITAs would have understood the claim term “file system.”

199. For purposes of this declaration, I’ve been asked to assume that the time of the alleged invention is January 11, 2010 (the filing date of U.S. Provisional Patent Application 61/294,070).

A. “File System”

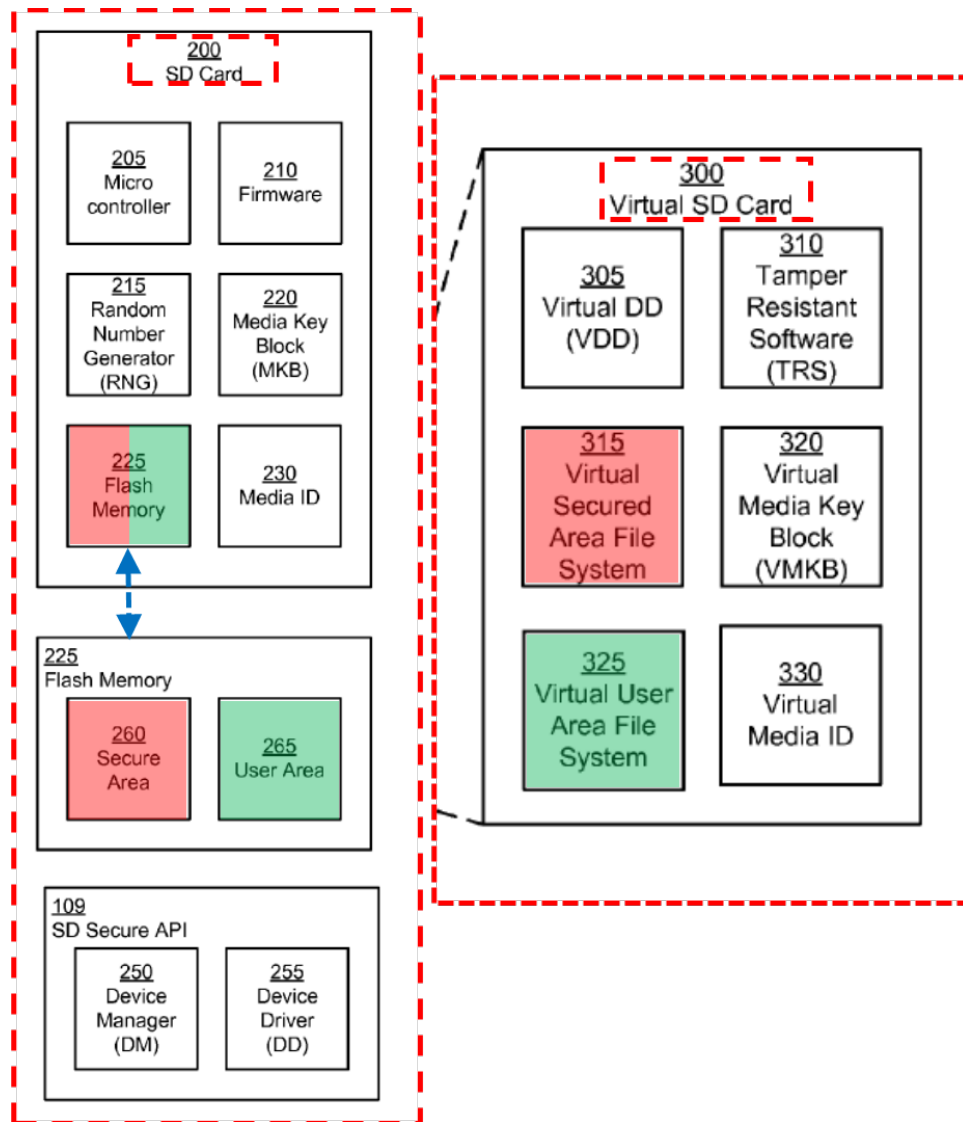
200. The independent claims of the ’749 patent recite “a file system for secure data on a storage device for storage of secure data corresponding to a secure data area of the SD card” and “a file system for user data on the storage device for storage of user data corresponding to a user data area of the SD card.” The patent does not explicitly define what it means by the term “file system.” Normally, the term “file system” means a “[t]he part of a computer’s backing store where files are held.” EX1012 (“Oxford Dictionary”), p. 196.

201. Note that according to the same dictionary, a “backing store” is “the memory on which information is held for reference but not for direct execution ... The term may be used either in an absolute sense, in which case it usually refers to a disk, or it may be used in a relative sense to refer to the device next down in the memory hierarchy” and “file” is “[a] collection of information, referred to by *file name; for example, a user-created document, program data, or the program itself.” See EX1012 (“Oxford Dictionary”), pp. 34, 194.

202. Certain claims, *e.g.*, challenged claims 2-3 and 19 recite that “the file system for secure data is encrypted.” Moreover, when referring to “file systems,” the ’749 patent explains that “the system creates a file system for secure data storage corresponding to the SD card’s secure data area and creates a file system for user data for storage of user data corresponding to the SD card’s user data area.” ’749 patent, 3:8-11. The ’749 patent goes on to state that “[t]he secured area file system 315 is configured **as a storage area for secure data**” (*id.*, 6:64-65) and that the “user area file system file system 325 is a **storage area for unsecured data**.” (*id.*, 7:26-27).

203. Based on this context provided by the ’749 patent, POSITAs would have understood that when the ’749 patent refers to a “file system” for either secure data or user data, it is referring to a storage area that stores either secure data or user data), which is consistent with the ordinary meaning I provided above. EX1012

(“Oxford Dictionary”) at 194. That this is the case is seen by comparing how the ’749 patent depicts a physical SD card to how it depicts a virtual SD card. In particular, the right-hand side of Fig. 2 shows the various portions of a “typical” SD card 200. The right-hand side of Fig. 3 shows the various portions of virtual SD card 300. Below, each are placed side-by-side and are color-coded to show correspondence between them:



204. As can be seen, a typical SD card 200 has a “secure area 260,” highlighted in red, and a “user area 265,” highlighted in green. The virtual SD card 300 is essentially the same, having a “virtual secured area file system 315,” highlighted in red, and a “virtual user area file system 325,” highlighted in green. This demonstrates that POSITAs would have understood that in the context of the ’749 patent, the ordinary meaning for the term “file system” is “a storage area.” *See also* §VII.C above.

XII. DETAILED EXPLANATION OF GROUNDS

A. Ground 1: Maeda In View Hamann And the AAPA Renders Claims 1, 14-16, 18, and 22-24 Obvious

1. Claim 1

a. [1pre] “A method for generating a virtual secure digital (SD) card, the method comprising:”

205. To the extent the preamble is limiting, Maeda-Hamann-AAPA discloses/suggests this feature. As I will discuss below, Maeda discloses the architectural features of secure digital “SD” cards. I note that by the priority date, the architecture of SD cards was well-known, as SD cards had been available for purchase by the public for years, a fact that the ’749 patent admits. ’749 patent, 1:17-18 (“Secure Digital (SD) cards are in wide use for many storage applications.”). Indeed, according to the ’749 patent, SD cards were required to conform to industry standards promulgated by the SD Association, the standards-setting organization

that developed the devices. *Id.*, 4:45-46 (“The SD card 200 must conform to the specifications developed and enforced by the SD ASSOCIATION.”). Thus, POSITAs would have been familiar with SD cards, their architecture, and their uses. Indeed, Maeda (as well as the AAPA, and Hirota) demonstrates that SD card architecture was well-known to POSITAs long before the priority date.

206. Hamann discloses “a method and system for secure back-up and usage of secure copies of smart card data objects.” EX1007, Hamann, ¶[0001]. To do so, Hamann “provid[es] a **virtual smart card (VSC) having the same defined logical file structure and the same content of data objects as its assigned real smart card...**” *Id.*, ¶[0014]. Hamann describes a smart card as “**any programmed data carrier used in any portable device**, like mobile phone, digital personal assistant etc., to securely hold subscriber specific information.” *Id.*, ¶[0002].

207. While Hamann does not explicitly disclose that the smart card being virtualized is an SD card, POSITAs would have understood that SD cards fall within Hamann’s smart card definition, as SD cards are data carriers that are portable and can be used in portable devices. A POSITA would not have found meaningful differences between a smart card and an SD card in the context of the ’749 patent. See §VII.B.1 above.

208. Hamann discloses that its virtual smart card is “a **software implemented version of a real smart card** providing the **equivalent functionality**

of a real smart card.” *Id.*, ¶[0015]. Hamann further discloses that its virtual smart card “has **the same defined logical file structure** and the same content of data objects as its assigned real smart card....” *Id.*, ¶¶[0014].

209. Because Maeda and the AAPA describe an SD card, and Hamann discloses virtualizing similar memory cards, Maeda-Hamann-AAPA discloses the preamble. POSITAs would have been motivated to combine Maeda, the AAPA, and Hamann for the reasons I discuss in §XII.A.2 below.

210. In sum, in my opinion, the Maeda-Hamann-AAPA combination discloses and/or suggests a method for generating a virtual secure digital (SD) card [*e.g.*, Maeda-AAPA’s SD card virtualized by Hamann’s smart card].

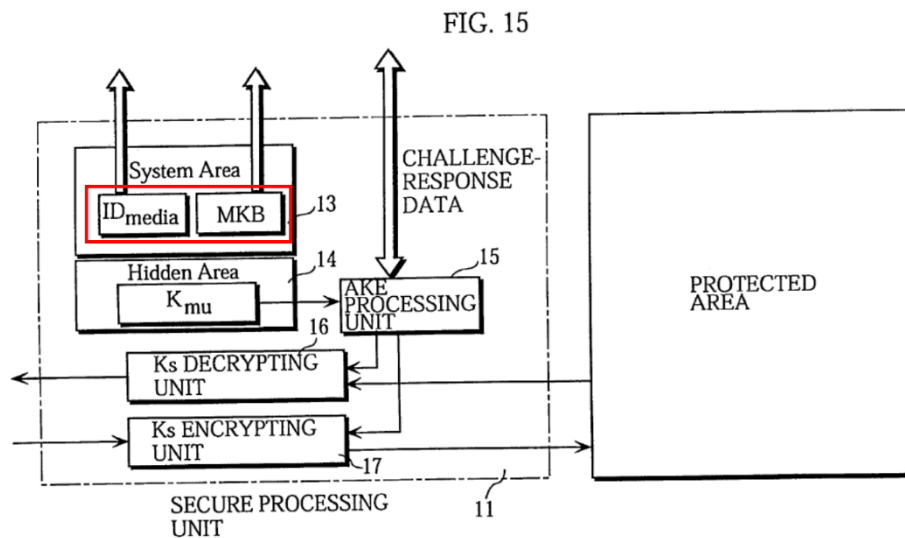
b. [1a] “reading a media key block and media identification for an SD card”

211. As an initial matter, I note that the claim recites reading the information “*for* an SD card.” The claim does not say “*from* an SD card.” Therefore, a POSITA would have understood that the media key block and the media identification can be read from any location, including but not limited to the SD card itself.

212. The Maeda-Hamann-AAPA combination discloses/suggests this feature. This element requires reading the media key block and media identification “for an SD card,” which POSITAs would understand to include reading these items from a physical (not virtual) SD card, although they could be read from elsewhere.

The '749 patent admits that prior art SD cards stored both the media key block (“MKB”) and media identification (*see* '749 patent, 4:57-60). Note that POSITAs would have understood that an SD card’s “media identification” is a number, a portion of which is unique to a particular SD card, and that SD cards use the “media key block” to encode and decode data stored therein.

213. Regardless of the patent’s admissions that SD card media IDs and media key blocks were prior art, Maeda discloses this feature. *See* Maeda’s Fig. 15 (annotated below):

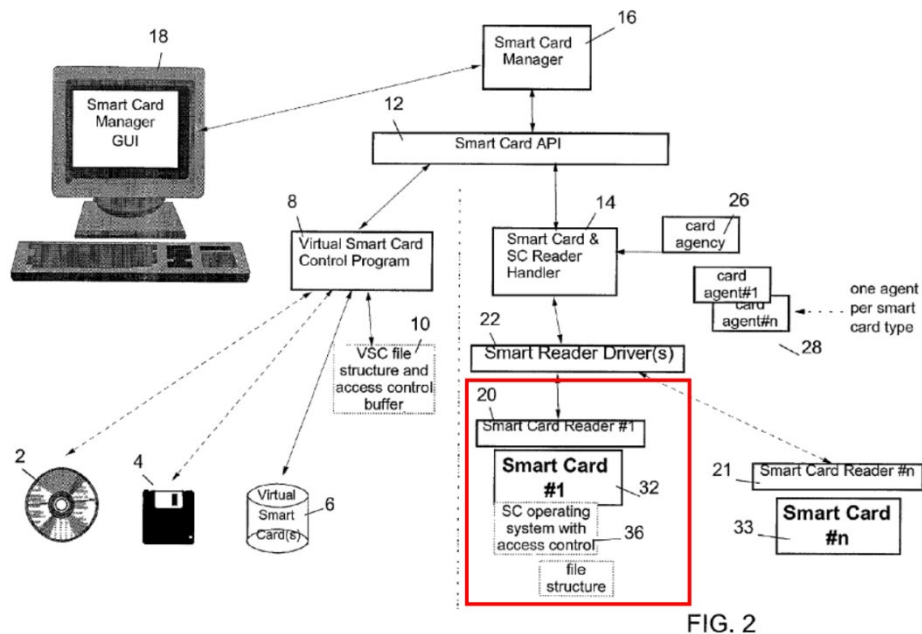


EX1006, Maeda, Fig. 15.

214. Consistent with SD card standards from before the priority date, Maeda discloses that “[t]he system area 13 is a read-only area storing a media key block (MKB) and a media-ID,” and that “[t]he MKB and media-ID stored in this area cannot be rewritten.” EX1006, Maeda, 13:49-51.

215. Maeda also discloses that “semiconductor memory card 100 is connected to an access apparatus 200, and the MKB and media-ID is read by that access apparatus 200.” *Id.*, 13:52-54. Thus, POSITAs would have recognized that Maeda discloses this element.

216. To create its virtual smart card, Hamann discloses that the physical smart card must be read by a reader, as shown in Hamann’s Fig. 2:



EX1007, Hamann, Fig. 2. *See also* EX1007, Hamann, ¶[0043].

217. To create a virtual smart card using Hamann, the “smart card is inserted into a smart card reader and the smart card manager is started.” *Id.*, ¶[0048]. After entering a password, Hamann’s “smart card manager displays all public and private data objects stored on the smart card.” *Id.*, ¶[0049]. POSITAs would have understood that when Hamann virtualizes an SD card like that described in Maeda

(and the AAPA), the SD card's media key block and media identification would be read, just as claimed.

218. Whether virtual or physical, a POSITA would have understood that important meta-data such as the MKB and media identification must be read first, then processed, before individual files and folders can be accessed; that is how media-based file systems operate. *See also* §VII.B above.

219. Moreover, the smart cards that Hamann virtualizes are similar to the SD cards described in Maeda and the AAPA. Hamann discloses that its virtual smart card has “the same defined logical file structure and the same content of data objects as its assigned real smart card...” *Id.*, ¶[0014]. Hamann further discloses that its VSC has “public area, a private area, a secure key area, password area, and an [sic] unique identifier area.” *Id.*, ¶¶ [0014], [0025]-[0030]. POSITAs would have recognized that this is similar to SD cards, which have a user data area and a protected area that stores data securely. EX1006, Maeda, 13:30-33. *See also* §VII.B.1 above.

220. Hamann also discloses that its VSC has a “unique identifier (ID),” which Hamann states “should be specified using the serial number of the real smart [sic: card] to be assigned to the new VSC.” EX1007, Hamann, ¶[0050]. POSITAs would have recognized that the “serial number of the real smart [card]” is analogous to an SD card's media identifier (*e.g.*, both being unique numbers), and would have

further understood that for Hamann to assign a unique identifier using that serial number, it would have had to read it, just as claimed.

221. Thus, when the smart card being virtualized using Hamann is an SD card like that in Maeda (and the AAPA), the Maeda-Hamann combination would read the MKB and media-ID stored on Maeda's (and the AAPA's) SD card.

222. In sum, in my opinion, the Maeda-Hamann-AAPA combination discloses and/or suggests reading [*e.g.*, Maeda-AAPA's access apparatus 200] a media key block [*e.g.*, Maeda's MKB] and media identification [*e.g.*, Maeda-AAPA's ID media, Hamann's unique identifier (ID)] for an SD card.

c. [1b] “storing the media key block and the media identification”

223. The Maeda-Hamann-AAPA combination discloses/suggests this feature. As I discussed in §XII.A.1.b above Maeda discloses that SD cards have media key blocks and a media-IDs. POSITAs would have understood that when using Hamann to create a virtual SD card like those in Maeda (and the AAPA), Hamann's VSC control program will store the virtual SD card on a storage medium, such as a hard disk. EX1007, Hamann, ¶[0040]. See §VII.B above.

224. POSITAs would have further understood that the SD card's media key block and media identification would be stored on that media since Hamann discloses that the virtual smart card it creates has “the same defined logical file

structure and the same content of data objects as its assigned real smart card....”
EX1007, Hamann, ¶[0014].

225. Moreover, as I discussed in §XII.A.1.b above, Hamann discloses that its virtual smart card has a “unique identifier (ID)” that uses “the serial number of the real smart [sic: card] to be assigned to the new VSC.” EX1007, Hamann, ¶[0050]. As I discussed, POSITAs would have understood that the “serial number of the real smart [card]” is analogous to an SD card’s media identifier (*e.g.*, both being unique numbers). Therefore, POSITAs would have understood that Hamann suggests storing Maeda’s (and the AAPA’s) media identifier.

226. Hamann’s Fig. 1 (annotated below) shows the file structure of its virtual smart card, and it shows that its unique ID (*i.e.*, the claimed “media identification”) is stored as part of its virtual smart card:

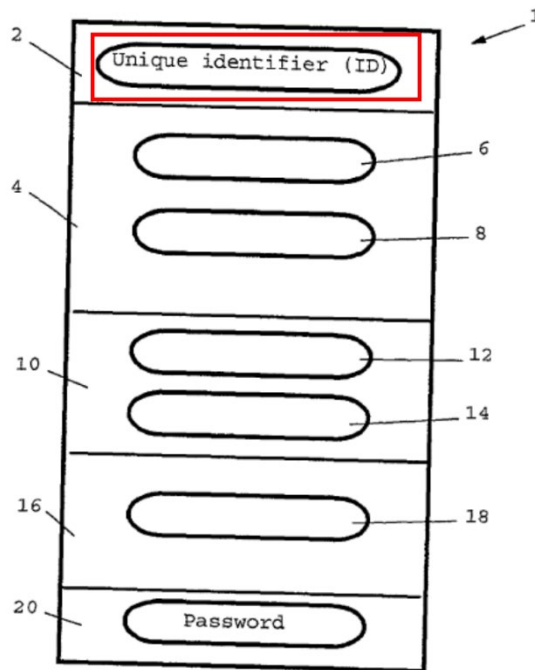


FIG. 1

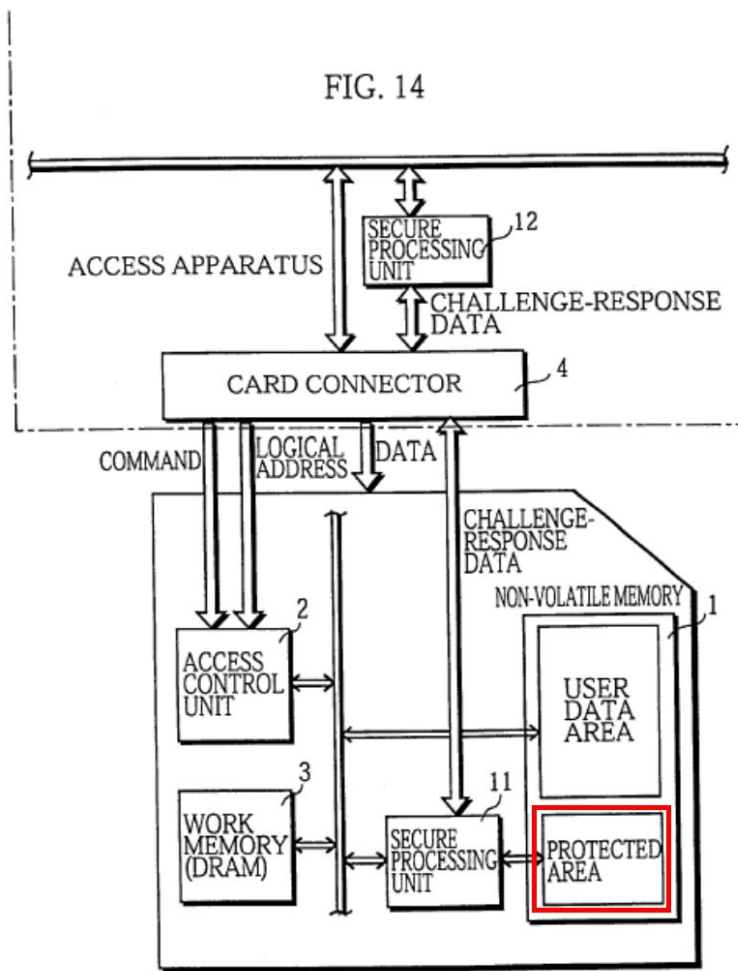
EX1007, Hamann, Fig. 1 (annotated above).

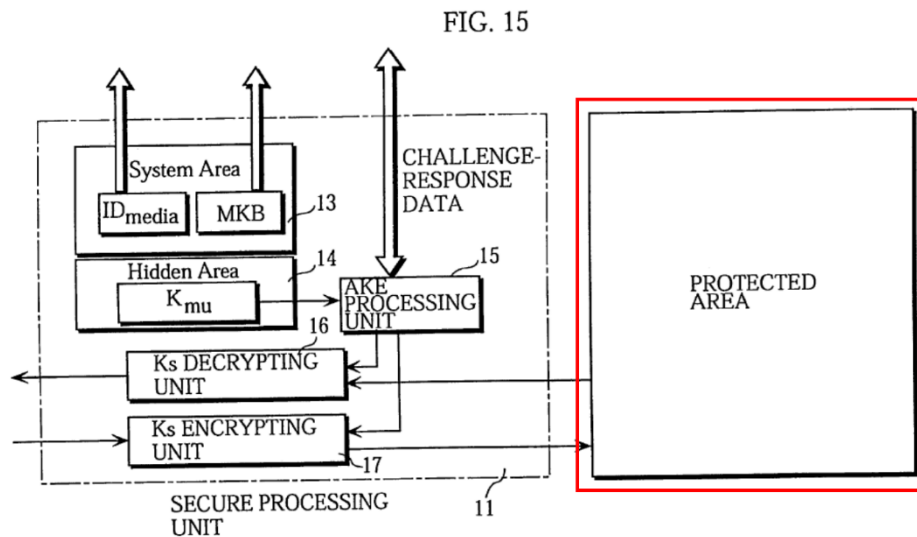
227. Thus, when the smart card being virtualized using Hamann is an SD card like that in Maeda (and the AAPA), POSITAs would have understood that the Maeda-Hamann combination would store Maeda's (and the AAPA's) media key block and the media identification as part of its virtual SD card.

228. In sum, in my opinion, the Maeda-Hamann-AAPA combination discloses and/or suggests storing [e.g., Hamann's VSC using the same structure/content as a real smart card] the media key block and the media identification.

- d. [1c] “creating a file system for secure data on a storage device for storage of secure data corresponding to a secure data area of the SD card; and”

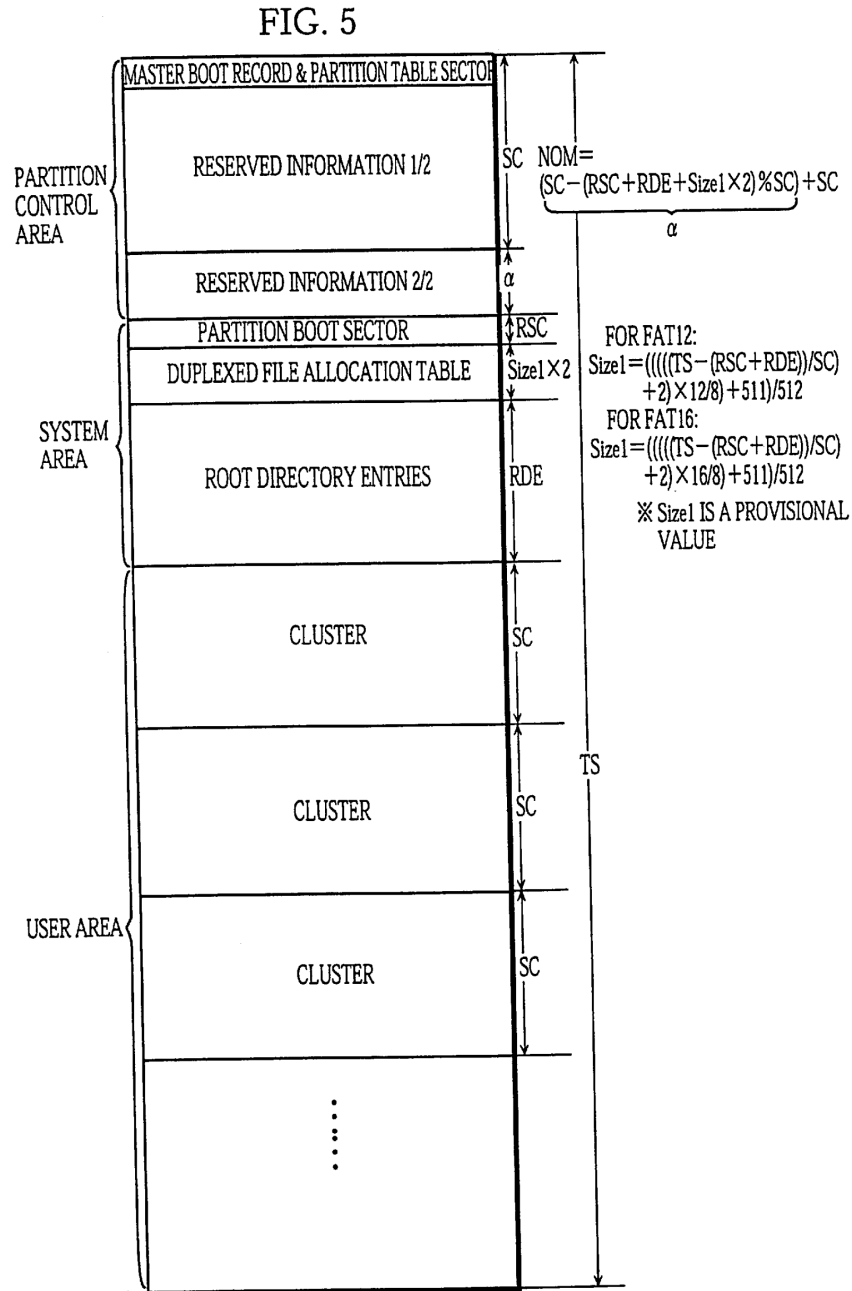
229. The Maeda-Hamann-AAPA combination discloses/suggests this feature. Maeda discloses that SD cards have what it refers to as a “protected area,” shown, *e.g.*, in Figs. 14-15 (annotated below):





230. Maeda discloses that its SD cards have a non-volatile memory storage area where a “user can gain access (the user data area), and **an area that stores data securely (the protected area).**” EX1006, Maeda, 13:30-33. *See also, id.*, 2:61-3:3, 13:36-41. Note that Maeda discloses that SD cards use an FAT file system. *See e.g., id.*, Figs 1C, 5 (below), 12-13, 10:51-67, 11:11-25 (“Once cluster size has been determined, at step S2, the **file system operation unit** 9 uses cluster size SC and the total sector size TS to **determine whether a FAT12 or a FAT16 file system should be used.** Once one of a FAT12 and FAT16 file system has been determined, at step S3 the file **system operation unit** 9 determines the length RDE of the root directory entry (in these embodiments the RDE is fixed at 32 sectors), and following this, in step S4, determines the length RSC of the partition boot sector (in these embodiments the RSC is fixed at 1 sector). Once the RDE and the RSC have been

obtained, the file system operation unit 9 uses the equation 11 to calculate the data length of the FAT, at step S5”).



231. Likewise, the AAPA includes the fact that “typical,” *i.e.*, prior art, SD cards included a “file system for secure data on a storage.” ’749 patent, 4:60-62

(“Data is stored within the flash memory 225, which is partitioned into two areas: **Secure Area 260** and User Area 265.”). Indeed, the ’749 patent admits “that every SD card is provisioned with a secure area 260, even though it may never be used by any application.” *Id.*, 5:1-3.

232. As I discussed in §XII.A.1.a above, Hamann discloses that its “virtual smart card (VSC) [has] the same defined logical file structure and the same content of data objects as its assigned real smart card.” EX1007, Hamann, ¶[0014]. Thus, POSITAs would have understood that when Hamann’s teachings are used to virtualize an SD card as in Maeda (and the AAPA), the resulting virtual SD card would have the “same defined logical file structure and the same content of data objects as its assigned real smart card.” This means that the resulting virtual SD card would have a “file system,” *i.e.*, a “storage area” of a virtual SD card, “for secure data on a storage device for storage of secure data corresponding to a secure data area of the SD card,” just as claimed.

233. Note also that Hamann creates a virtual smart card having private area 10, as seen in its Fig. 1 (annotated below):

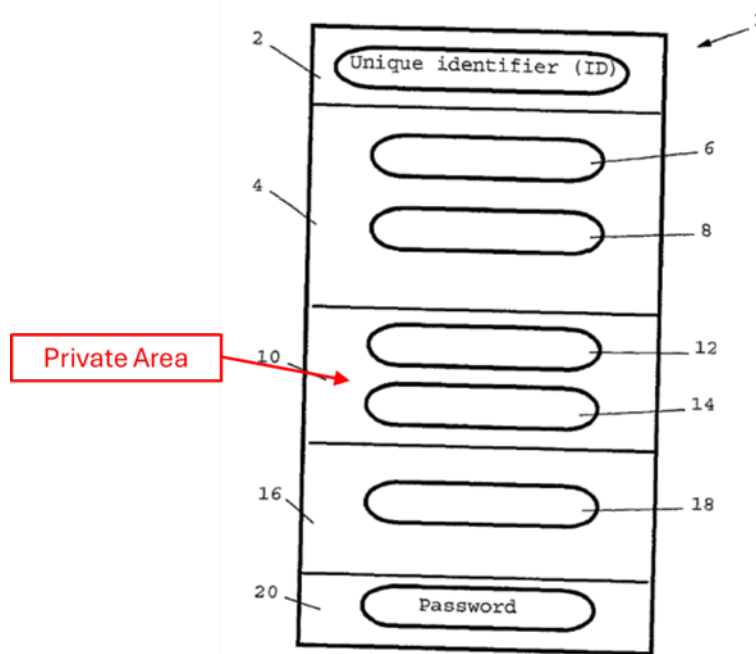


FIG. 1

234. Hamann describes the private area in Fig. 1. See EX1007, Hamann, ¶¶[0025]-[0027] (“[0025] The logical file structure of the VSC (1) is preferably defined by the following data areas: ... [0027] (a private area (10) in which private objects being encrypted are placed; private objects may only be accessed providing a password (10), e.g. account no (12) and key information (14)”).

235. Hamann discloses “a VSC control program handling the creation, the security and the read/write process of the VSC,” which generates a “**VSC having a logical file structure** comprising a public area, **a private area**, a secure key area, password area, and an [sic] unique identifier area.” *Id.*, ¶¶[0015]-[0016]. Hamann’s

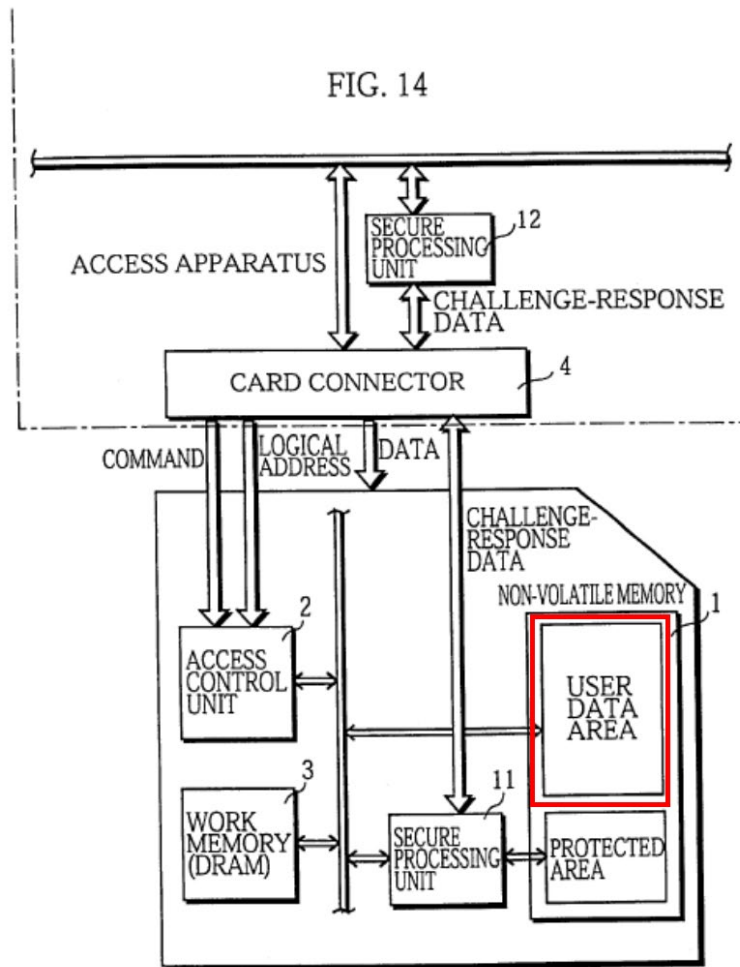
private area stores “private objects” that are “being encrypted.” *Id.*, ¶¶[0016], [0027].

236. POSITAs would have understood that Hamann’s “private area,” which has a “logical file structure,” corresponds to the claimed “file system for secure data on a storage device for storage of secure data corresponding to a secure data area of the SD card,” especially when combined with Maeda (and the AAPA).

237. In sum, in my opinion, the Maeda-Hamann-AAPA combination discloses and/or suggests creating a file system [*e.g.*, Maeda-AAPA protected area FAT/FAT12/FAT16 file system(s), Hamann’s private area having the same logical file structure as a real smart card] for secure data [*e.g.*, Maeda-AAPA’s protected area, Hamann’s private area 10] on a storage device for storage of secure data corresponding to a secure data area of the SD card.

e. [1d] “creating a file system for user data on the storage device for storage of user data corresponding to a user data area of the SD card”

238. The Maeda-Hamann-AAPA combination discloses/suggests this feature. Maeda discloses that SD cards have what it refers to as a “user data area,” shown, *e.g.*, in Fig. 14 (annotated below):



239. Maeda discloses that its SD cards have a non-volatile memory storage area where a **“user can gain access (the user data area)**, and an area that stores data securely (the protected area).” EX1006, Maeda, 13:30-33. *See also, id.*, 2:61-3:3, 13:36-41. Maeda further discloses that its user data area is “for recording user data.” Maeda, 1:60-61. As I previously discussed, Maeda discloses that SD cards use an FAT file system. *See e.g.*, Figs 1C, 5, 12-13, 10:51-67, 11:11-25. *See also* §X.A above.

240. Likewise, the AAPA includes the fact that “typical,” *i.e.*, prior art, SD cards included a “file system for secure data on a storage device.” *See* ’749 patent, 4:60-62 (“Data is stored within the flash memory 225, which is partitioned into two areas: Secure Area 260 and **User Area 265**.”).

241. Thus, POSITAs would have realized that prior art SD cards like those described in Maeda (and the AAPA) included “a file system for user data on the storage device for storage of user data,” just as claimed.

242. As I discussed in §XII.A.1.a above, Hamann discloses that its “virtual smart card (VSC) [has] the same defined logical file structure and the same content of data objects as its assigned real smart card.” EX1007, Hamann, ¶[0014]. Thus, POSITAs would have understood that when Hamann’s teachings are used to virtualize an SD card as in Maeda (and the AAPA), the resulting virtual SD card would have “same defined logical file structure and the same content of data objects as its assigned real smart card.” This means that the resulting virtual SD card would have a “file system,” *i.e.*, a “storage area” of a virtual SD card, “for user data on the storage device for storage of user data corresponding to a user data area of the SD card,” just as claimed.

243. I note that Hamann creates a virtual smart card having public area 4, as seen in Fig. 1 (annotated below):

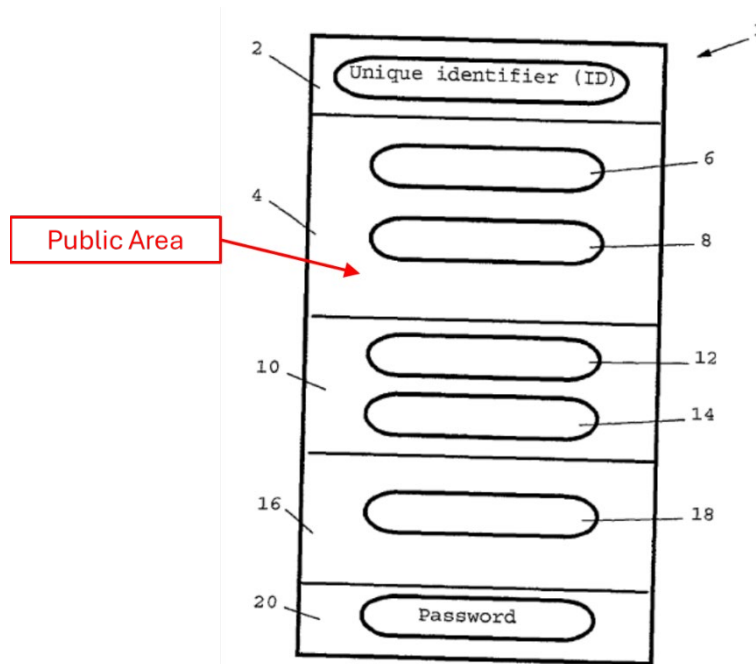


FIG. 1

244. Hamann describes the public area in Fig. 1. See EX1007, Hamann, ¶¶[0025]-[0026] (“[0025] The logical file structure of the VSC (1) is preferably defined by the following data areas: [0026] (a public area (4) in which public data objects having no access conditions are placed, e.g. Certificate (6) and address (8))”).

245. Hamann discloses “a VSC control program handling the creation, the security and the read/write process of the VSC,” which generates a “**VSC having a logical file structure** comprising **a public area**, a private area, a secure key area, password area, and an [sic] unique identifier area.” *Id.*, ¶¶[0015]-[0016]. Hamann discloses that “[t]he data objects contained in the public area hav[e] no access restrictions....” EX1007, Hamann, ¶¶[0016], [0026], [0027].

246. POSITAs would have understood that Hamann’s “public area,” which has a “logical file structure,” corresponds to the claimed “file system for user data on the storage device for storage of user data corresponding to a user data area of the SD card,” especially when combined with Maeda (and the AAPA).

247. In sum, in my opinion, the Maeda-Hamann-AAPA combination discloses and/or suggests creating a file system [*e.g.*, Maeda-AAPA’s user data area FAT/FAT12/FAT16 file system(s), Hamann’s public area having the same logical file structure as a real smart card] for user data [*e.g.*, Maeda-AAPA’s user area, Hamann’s public area 4] on the storage device for storage of user data corresponding to a user data area of the SD card

- f. **[1e] “wherein a size of the file system for secure data is determinant on a size of the file system for user data, the determination based on a lookup table providing a relationship between the sizes.”**

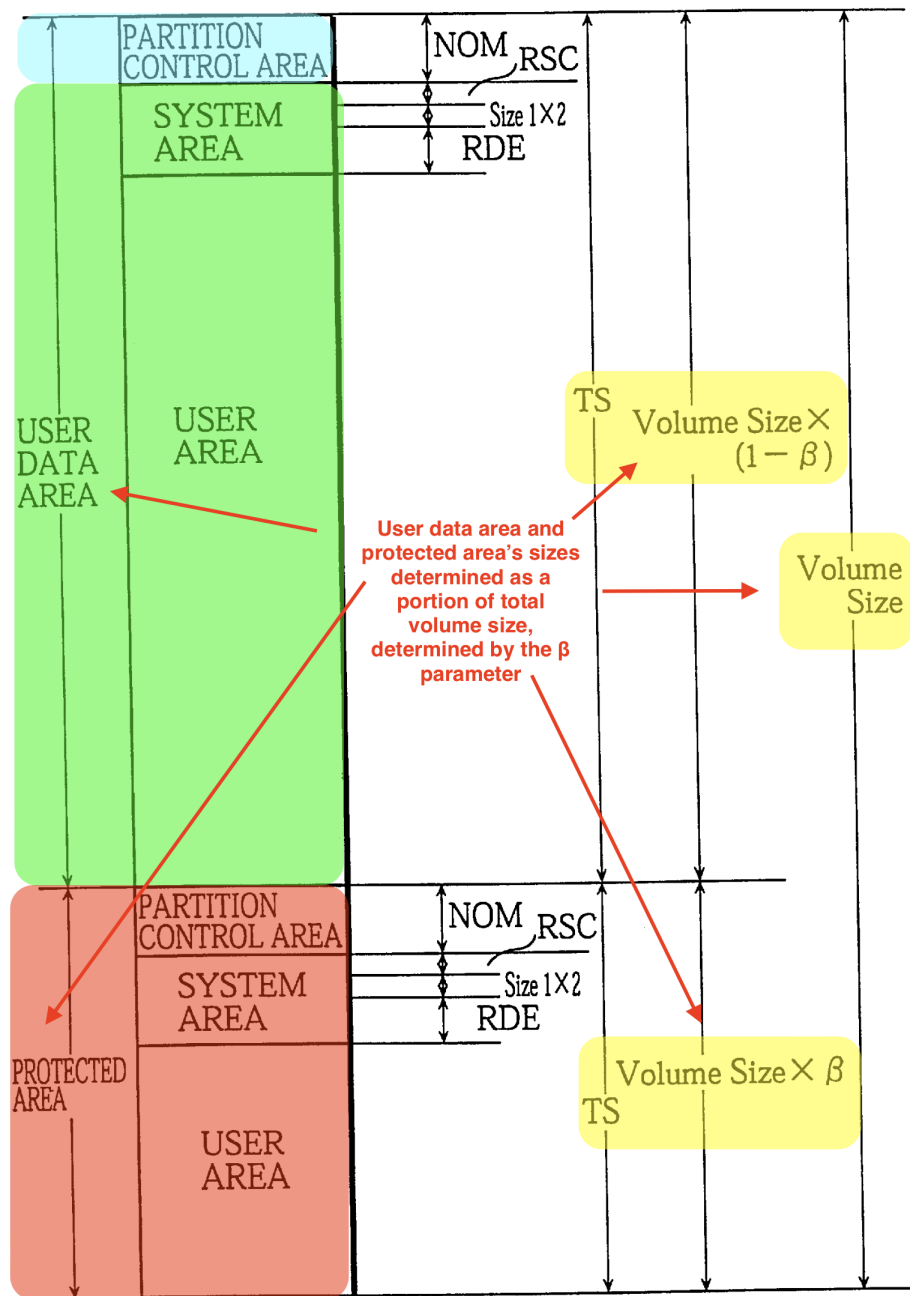
248. The Maeda-Hamann-AAPA combination discloses/suggests this feature. Maeda discloses that the size of the file system for secure data and the file system for user data are calculated such that they each have a size that is relative to the other. EX1006, Maeda, at 15:12-21. *See also id.*, 17:1-11, Figs. 16, 17, Claim 7. Thus, Maeda recognized that the size of the size of its protected area (*i.e.*, “file system for secure data”) was determinant on the size of its user data area (*i.e.*, “file system for user data”), and POSITAs would have recognized that Maeda was

suggesting that the size of its protected area was related to the size of its user data area.

249. Maeda discloses that the size of its user and protected areas are configurable. Maeda states: “[n]ext, the format of the volume area in the second embodiment is explained, with reference to the detailed drawing of the volume area including the protected area shown in FIG. 16. In the drawing, if the total size of the volume area is ‘volume size’, the user data area is an area ‘volume size $\times$ (1- β)’ extending from the start of the volume area, and an area ‘volume size $\times\beta$ ’ following the user data area is allocated to the protected area. Here, β is a value for which the minimum setting is 0.01. To take one example, if the volume size is 64 MB and β is 0.01, the protected area is set at 640 KB. The format of the user data area and the protected area is realized when the file system operation unit 9 executes the initialization procedure shown in FIG. 9.” EX1006, Maeda, 15:12-21. A POSITA would have understood that the sizes of the protected and user areas are configurable, and that one example ratio is that the protected area’s size is 0.01 (*i.e.*, 1%) of the user’s area size.

250. Maeda’s Fig. 16 (annotated below) further illustrates this, where the β parameter (in yellow) controls the ratios of the user data area (green) and protected area (red):

FIG. 16

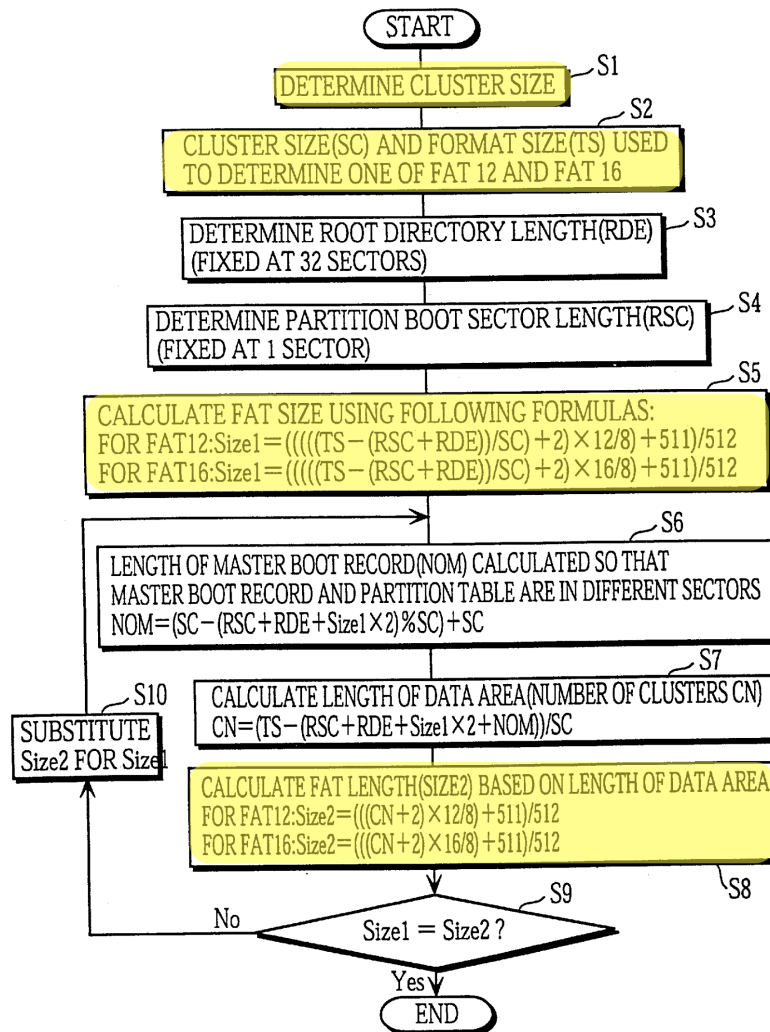


251. A POSITA would have understood the above figure to mean that since the size ratios are configurable and parameterized, there could easily be multiple

possible ratios for different sizes, especially as SD cards and Smart cards came in many different sizes. *See also* §VII.B.1 above.

252. Maeda further discloses that the data size area, and hence the protected area's size, also depend on which file system is being used to format the volume/partition. Maeda discloses that the volume can be partitioned using FAT12 or FAT16. A POSITA would know that FAT12 has a maximum size of 16MB (using 4KB size clusters) or 32MB (using 8KB size clusters); FAT16 has a maximum size of 4GB. See https://en.wikipedia.org/wiki/File_Allocation_Table (last viewed September 30, 2024). Therefore, there are three possible FAT file system sizes depending on the type and cluster size. These can be seen in at least the annotated steps of Maeda's Fig. 13 (annotated below):

FIG. 13



253. As Fig. 13 shows above, steps S5 and S8 includes at least one if-then-else decision based on the FAT type; and many steps also depend on the cluster size (SC). A POSITA would have found it obvious and understood that the decisions based on the FAT type (and cluster size) could easily be determined also with a lookup table, especially if additional FAT types were desired (e.g., 8-bit FAT, FAT16B, or FAT32, all of which were in wide use well before 2010). Lastly, such

a table could easily accommodate additional determinations based on different desired β values.

254. Despite these suggestions, Maeda does not directly disclose that the size of protected area relative to the size of its user data is determined “based on a lookup table providing a relationship between the sizes,” as claimed. However, the ’749 patent admits that this was in the prior art. In particular, the ’749 patent states that “secure area’s 260 size is determined at design time according to the SD card specifications.” ’749 patent, 4:66-5:1. The ’749 patent admits that in SD cards, the size of the file system for secure data is “determinant on a size of the file system for user data,” and that this “determination [is] based on a lookup table providing a relationship between the sizes:”

A size of the secured area file system 315 is determined at provisioning time according to the SD card specifications. The size of the virtual secured area file system 315 is a table lookup from SD ASSOCIATION specifications based on the size of the virtual user area file system 325. The process creates a file or partition on the storage 116 that is accordingly formatted for the virtual secured area 315.

’749 patent, 7:62-8:2.

255. Thus, the feature recited in element [1e] is an AAPA. POSITAs would have been motivated to use this feature from the SD card standard that is the AAPA for the reasons discussed in §XII.A.2 below.

256. I note that the only reason the independent claims were allowed was because patentee amended them to include this very element, after the examiner suggested that this feature “appears to be novel and unobvious” and that the Applicant could rewrite the (rejected) claims to include those features. *See* §VIII.B above; EX1004 (Prosecution History), at 74-88, 83. I have not been able to find any evidence in the ’749 patent’s prosecution history that the Examiner knew about patentee’s admissions regarding this feature or that this feature was standardized by the SD Association. Given these admissions, it appears to me that the Examiner incorrectly thought this feature was novel and not in the prior art.

257. In sum, in my opinion, the Maeda-Hamann-AAPA combination discloses and/or suggests wherein a size of the file system for secure data is determinant on a size of the file system for user data, the determination based on a lookup table providing a relationship between the sizes [*e.g.*, AAPA disclosing the two sizes are determined by a table lookup, which the AAPA further admits is defined and enforced by the SD Association; Maeda-AAPA’s suggestions of different sizes depending on FAT type and size-ratio β parameter].

2. Motivation To Combine Maeda, the AAPA, and Hamann

258. POSITAs would have been motivated to combine Maeda, the AAPA, and Hamann for at least eight reasons described next.

259. **First**, both Maeda and the AAPA disclose the same technology, an SD card, with its protected area, user area, etc.

260. POSITAs would have recognized that Maeda discloses SD cards. EX1006, Maeda, at 2:61-3:3 (“The type of semiconductor memory card known as a secure digital (SD) memory card has made great strides toward resolving the aforementioned problems of reduction in processing time and nonvolatile memory longevity for the following reasons. In an SD memory card, an area known as the protected area, which cannot be used by the general user, is provided. This protected area is used to store secrete [sic] information, such as an encryption key used for encrypting data, billing information used to bill a user when a copyrighted material is reproduced, and the like.”).

261. Likewise, Applicant admitted that SD cards were well known, in typical use, and standardized. ’749 patent, 4:43-46 (“**Typical** SD Card Architecture and Functions The SD card 200, may be a full size, mini SD, or micro SD physical format. The **SD card 200 must conform to the specifications developed and enforced by the SD ASSOCIATION.**”). POSITAs would have been motivated to use features that were contained in industry standards (and indeed would have been

essentially forced to do so), and thus would have been motivated to use the lookup table feature, which the AAPA admits is required by SD card standards.

262. **Second**, both Maeda and the AAPA disclose/suggest that the sizes of the user and protected areas are related to each other.

263. Maeda discusses the need to configure the sizes of the user and protected areas. EX1006, Maeda, at 3:3-13 (“The amount of data which needs to be kept secret varies according to the type of application program used, so that the size of the protected area has to change according to the type of application program. **If the size of the protected area is changed, this means that the layout of the volume area also changes in response** to the type of application program. If the layout of the volume area changes in this way, a layout in which clusters straddle the boundaries between erasable blocks is often generated, so achievement of the above-mentioned object is particularly desirable.”).

264. Similarly, the AAPA admits that the sizes of the user and protected areas are related and must conform to (and “enforced” by) the SD Association’s specifications. ’749 patent, 7:62-67 (“A size of the secured area file system 315 is determined at provisioning time according to the SD card specifications. The size of the virtual secured area file system 315 is a table lookup from SD ASSOCIATION specifications based on the size of the virtual user area file system 325.”).

265. Thus, Maeda and the AAPA both suggest that the sizes of the user and protected areas are related and configurable. A POSITA would have most naturally understood that a simple, fixed size relationship may not suffice because SD cards come in different sizes. Instead, a POSITA would expect there to be multiple such fixed size associations, possible one ratio for each size of SD card. A POSITA would have been familiar with simple computer data structures such as tables and arrays. Just as two-column spreadsheets are in widespread use, tables can easily relate numerical sizes of user areas to protected areas. Moreover, POSITAs would have been motivated to substitute features like those in Maeda for features that are required by industry standards. Thus, POSITAs would have been motivated to use the SD Associations lookup table feature from the AAPA because it was mandated by industry standards.

266. To the extent it is determined that Maeda does not disclose the “table lookup” element, it would have been obvious to a POSITA to associate the two sizes with a simple table, for the same reasons discussed just above. In fact, a person with even less experience than a POSITA would have recognized and be able to implement such a technique. This introductory material on basic data structures is often taught (as I personally have) to first year Computer Science undergraduate students (*i.e.*, those with far less experience than a POSITA as I defined in §IV above).

267. **Third**, Matsushita, the assignee of Maeda, was a founding member of the SD association. See https://en.wikipedia.org/wiki/SD_Association (last viewed September 28, 2024.). A POSITA would have been looking for SD card information from those who worked the most on it (*e.g.*, members of an association, vendors, their patents, and other documents and specifications).

268. **Fourth**, because Maeda and the AAPA disclose the same underlying SD card technology, combining them would have been trivial and would not have required extensive experimentation or impose undue burden. For the same reasons, the combination would have yielded predictable results. Lastly, a POSITA would have had a high expectation of success because the AAPA does nothing more than describe more details of an already standardized SD card technology.

269. Next, I focus on the motivations to combine Maeda and Hamann and the AAPA.

270. **First**, SD cards (Maeda and the AAPA) and smart cards (Hamann) are very similar technologies. Both are semiconductor, solid state storage devices, with a small form factor. Both include a microprocessor, volatile and non-volatile memory, and input/output methods. Both support a file system to store various files. Both include strong mechanisms to secure the data (*e.g.*, authentication, encryption, and more). Both support different device sizes and hence file system sizes. And both were standardized by their respective standards-body organizations. See details

of smart cards in comparison to SD cards in §VII.B.1 above. A POSITA familiar with smart cards would have readily recognized the great similarities to SD cards, and vice versa.

271. Indeed, POSITAs would have expected success in making the Maeda-Hamann combination, as the architectures in each are fairly similar. For example, Hamann’s “smart card” and Maeda’s SD card have similar data formats (*e.g.*, protected area, user area) and encryption. See EX1007, Hamann, ¶[0016] (“VSC having a logical file structure comprising a public area, a private area, a secure key area, password area, and an [sic] unique identifier area.”); *see also* EX1006, Maeda, at 6:66-7:2, 13:30-33 (describing that its semiconductor memory card includes a user data area, a protected area, as well as a partition control area and a the system area”). Thus, virtualizing Maeda’s SD card would have been well within a POSITA’s abilities, especially in view of Hamann and Maeda’s detailed teachings, as I detail next.

272. **Second**, Maeda and Hamann describe one key motivation being the protection of data through backups. Almost every user has at one point or another lost data and tried to recover it from backups (or, in hindsight, wish they had backed up their important data ahead of time).

273. Maeda discloses an embodiment where “the extended attribute is stored in at least one of the protected area and a backup area. ... The backup area is

managed separately from the normal partition in the user area. ... New extended attributes that cannot be defined in a FAT file system can be set for each file by storing such attributes in the protected area or the backup area.” EX1006, Maeda, at 26:39-51, FIG. 32.

274. Similarly, Hamann’s main purpose is for “[b]ack-up and usage of secure copies of smart card data objects” (EX1007, Hamann, Title). Indeed, Hamann states that “[i]t is therefore object of the present invention to provide an improved system and method allowing easy and secure back-up of the content of a smart card.” EX1007, Hamann, ¶[0008].

275. Any storage technology can be used to back up data, from mere floppy diskettes, to hard disks, and of course SD cards and smart cards. See §VII.B above. A POSITA using Maeda and interested in data backups would have sought out other references that disclose backup methods and would have found Hamann—especially as Hamann aims to make backups of important data easier to achieve (through virtualization as discussed next).

276. A POSITA would have also recognized that when users employ technologies to secure data (*e.g.*, SD cards, smart cards), they are most likely going to store critical information that must be protected (*e.g.*, encrypted, authenticated). Therefore, it is exactly that kind of critical information that must also be backed up, to protect against the loss, damage, or corruption of physical storage devices.

277. A POSITA would have required little to no experimentation because they are merely applying the same technique—storing and backing up data files securely—to different yet standardized devices. At most, a POSITA would have had to update their code from using an SD Card standard file-access API (*e.g.*, read/write methods) to using a different yet standard API to access files on smart cards. See §VII.B.1 above. This is well within a POSITA’s abilities.

278. **Third**, more than a decade before the ’749 patent was filed, many forms of virtualization were already well known. Virtualization improves the versatility and flexibility of computer systems, allows better resource utilization and sharing, and much more. Well before 2010, virtualization of many resources had become a commonplace technique, as I detail in §VII.E above.

279. Hamann teaches that virtualizing memory cards allows such memory cards to be backed up more easily. EX1007, Hamann, ¶[0002]. POSITAs would have recognized that virtualizing SD Cards like those described in Maeda (and the AAPA) would provide this benefit to users, as Hamann suggests. For example, Maeda teaches that users may store content such as music and images on the memory card disclosed therein. EX1006, Maeda, at 25:64-26:3. POSITAs would have recognized that the ability to backup such content (whether DRM protected content or a user’s unprotected content) would have been beneficial to users, as it would

prevent loss of their content if the user lost their card, or they experience an SD card hardware failure.

280. Indeed, Hamann suggests this, stating that “[a]n essential advantage of this invention is that **backed-up smart card data objects in the VSC** allows the user to continue working with the [sic] most of the applications **if the real smart card lost or damaged.**” EX1007, Hamann, ¶[0019]. Given Hamann’s explicit teachings, POSITAs would have understood the benefits provided by backing up SD cards using virtualization like those described in the combination of Maeda-AAPA using Hamann’s teachings.

281. **Fourth**, POSITAs would have been motivated to combine the AAPA with Maeda and Hamann because the use of lookup tables to vary conditions based on known variables was well-known, and thus POSITAs would have understood that using a table to select the size of the secure area based on the size of the user data was a simple, efficient, and cost-effective method.

282. POSITAs would have been further motivated to make this combination because, as the ’749 patent admits, using a lookup table for determining the size of the protected area was required by the SD card specification. POSITAs seeking to virtualize an SD-card based on the Maeda-Hamann combination would have naturally looked to the AAPA to determine the relationship between the total size of an SD card and the protected area and user area. Indeed, the fact that this feature is

required by the industry standard (charged with enforcing it) would have motivated POSITAs to use this feature. Implementing tables and similar data structure would have been well within a POSITA's expertise and would not have required undue burden.

283. *In sum*, a POSITA would have been motivated to combine Maeda, the AAPA, and Hamann because the storage technologies were fairly similar and standardized, were used for similar purposes, virtualization was well known and widely used, and table lookups are nothing more than simple data structures.

3. Claim 14

a. [14pre] “A method using a virtual secure digital (SD) card, the method comprising:”

284. To the extent the preamble is limiting, the Maeda-Hamann-AAPA combination discloses/suggests this feature. As I discussed in §XII.A.1.a, Hamann discloses virtualizing smart cards, and when combined with Maeda (and the AAPA) results in a virtual SD card.

285. Hamann also discloses using its virtual smart card. EX1007, Hamann, ¶[0015] (“The VSC is a software implemented version of a real smart card providing the equivalent functionality of a real Smart card. **The VSC is created and used** by a VSC control program handling the creation, the Security and the read/write process of the VSC.”); *see also id.*, ¶[0053] (“In a case of los[s] of a[] data object on the real

smart card either **the VSC can be used directly as temporary ‘smart card’** of the previously saved objects or may be transferred back to the real smart card using the same steps as described before in the opposite direction, i.e. copy the data objects from the VSC and paste them to the real smart card.”).

286. Thus, the Maeda-Hamann-AAPA combination discloses the preamble.

287. In sum, in my opinion, the Maeda-Hamann-AAPA combination discloses and/or suggests a method using a virtual secure digital (SD) card [*e.g.*, Maeda-AAPA’s SD card virtualized by Hamann’s smart card].

b. [14a] “determining if the virtual SD card is provisioned;”

288. The Maeda-Hamann-AAPA combination discloses/suggests this feature. As I discussed in §XII.A.1.b above, Hamann discloses that its system has one or more smart card readers. *See e.g.*, EX1007, Hamann, ¶[0043] and Fig. 2. As shown in Fig. 3A (annotated below), Hamann discloses that when a smart card is inserted into a reader, the VSC program’s graphical user interface (“GUI”) displays the card readers as well as the identity of any inserted smart cards:

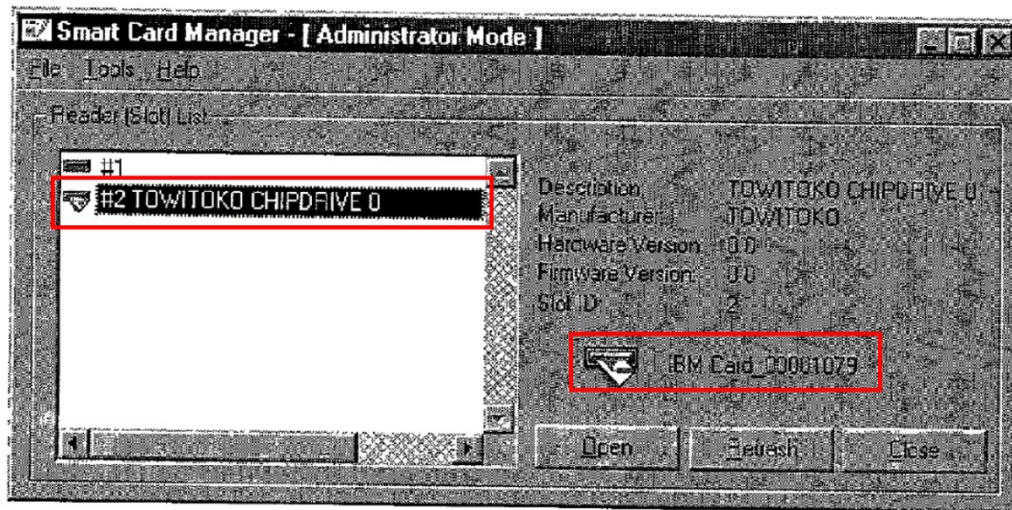


FIG.3A

289. In this example, Hamann’s GUI shows that card reader #2 has a smart card named “IBM Card_00001079” inserted therein. Indeed, Hamann discloses that “[t]he GUI of the smart card manager **displays all available smart card readers and VSCs.**” EX1007, Hamann, ¶[0049].

290. As shown in Fig. 3K (annotated below), once provisioned (*see* §XII.A.3.c below), this same GUI will display all available card readers and also the provisioned virtual smart card:

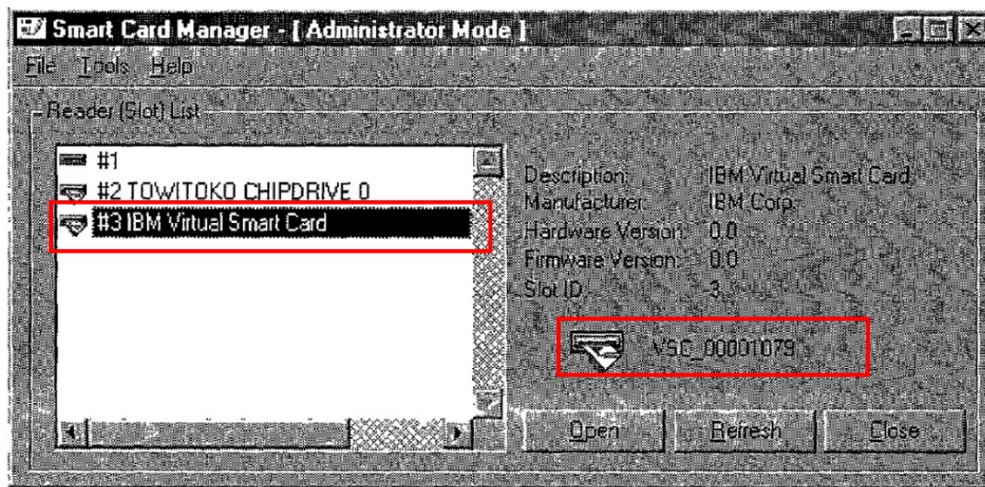


FIG. 3K

291. In this example, Hamann’s GUI shows that card reader #3 is actually a virtual smart card named “VSC_00001079.” EX1007, Hamann, ¶[0051].

292. Given these teachings, POSITAs would have understood that Hamann’s GUI identifies all available smart cards, including both physical smart cards that have been inserted into one of Hamann’s card readers (*e.g.*, “IBM Card_00001079”), and any virtual smart cards that have been created (*e.g.*, “VSC_00001079”). EX1007, Hamann, ¶[0049]. Thus, when a user inserts a smart card into Hamann’s system, Hamann’s GUI would inform the user whether a virtual version of that smart card had been set up on its system.

293. A POSITA would have understood that both virtual as well as physical smart cards would be recognized and displayed because the very nature of

virtualization is to ensure that virtual resources appear just like physical ones. *See* §VII.E above.

294. The '749 patent refers to a “provisioning process for creating the virtual SD card....” ’749 patent, 7:46-49. Given this, POSITAs would have understood that “determining if the virtual SD card is provisioned,” means determining whether a virtual SD card has been “created.” I note that other steps recited in claim 14 use the term “creating.” *See* elements [14c]-[14d] in §§XII.A.3.d-XII.A.3.e below. To the extent “provision/provisioned/provisioning” must mean something other than “create/created/creating,” POSITAs would have understood that the term commonly also means “set up/setting up,” which Hamann discloses.

295. As I discuss in §XII.A.1.a above, Hamann does not explicitly disclose virtualizing SD cards. However, as I also discussed there, SD cards fall within the scope of Hamann’s smart card definition. *See* §VII.B.1 above. Moreover, Hamann suggests and thus would have motivated POSITAs to virtualize SD cards like those described in Maeda and the AAPA, as I discussed in §XII.A.2 above.

296. In sum, in my opinion, the Maeda-Hamann-AAPA combination discloses and/or suggests determining if the virtual SD card is provisioned [*e.g.*, Hamann’s creating virtual smart cards (VCSs), shown and administered on its “VSC Manager” GUI].

c. [14b] “provisioning the virtual SD card in response to the virtual SD card not being provisioned, the provisioning including:”

297. The Maeda-Hamann-AAPA combination discloses/suggests this feature. As I discussed in §XII.A.3.b above, Hamann’s GUI informs a user as to whether a virtual smart card has been provisioned. *See* EX1007, Hamann, Fig. 3A, ¶[0049]. Hamann’s “Smart Card Manager” provisions the unprovisioned virtual smart card using the following process. After inserting a smart card into a reader, all the objects on that card are displayed, with any private objects being displayed only if a password is provided. EX1007, Hamann, ¶[0049] and Figs. 3B-3E.

298. To start the process of provisioning the virtual smart card, the user selects the “Copy All” command from the GUI, as shown in Fig. 3F (annotated below):

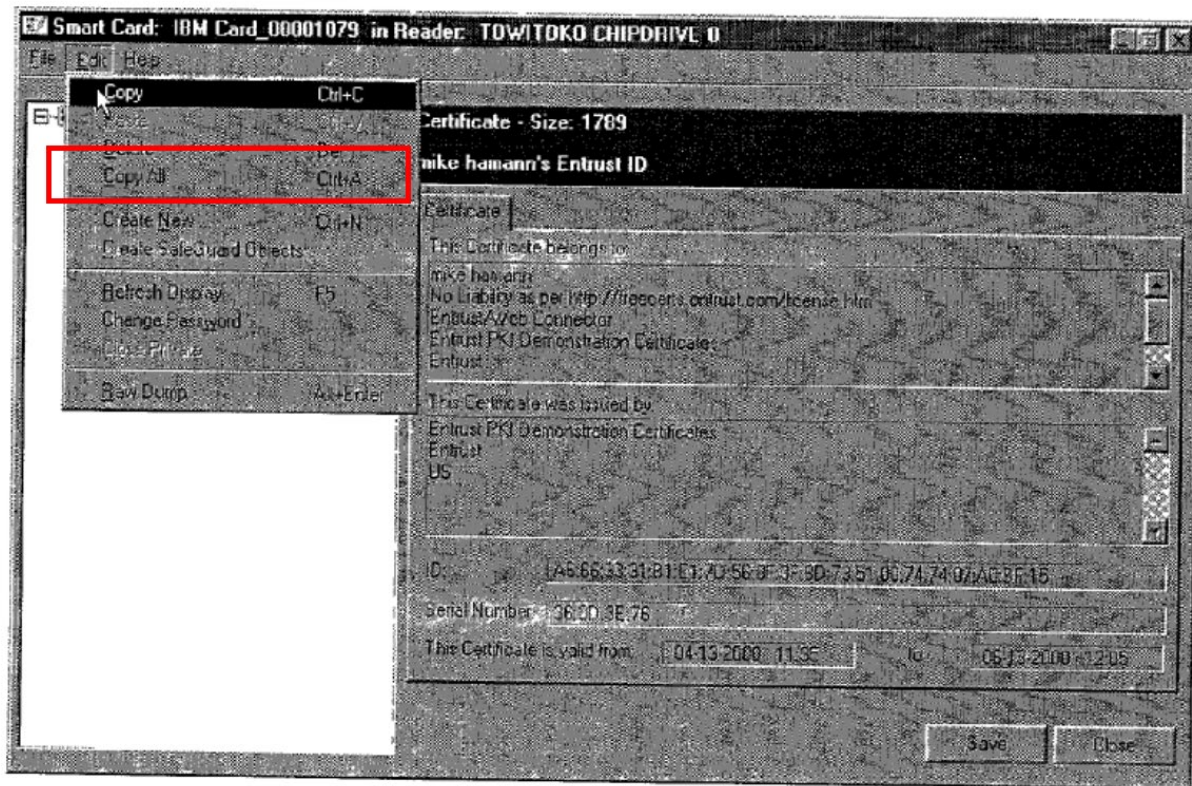


FIG. 3F

299. Hamann describes this step as follows. Hamann, ¶[0049] (“By selecting the ‘Copy command from the Edit menu of the smart card manager GUI’ (see FIG. 3F) the smart card manager stores the selected objected in an intermediate buffer. Furthermore, the smart card manager GUI offers via the Edit menu the possibility to copy all objects stored in the smart card (see FIG. 3F).”).

300. Next, using the “Tools” menu in the GUI, a user invokes the “VSC Manager,” as shown in Fig. 3G (annotated below):

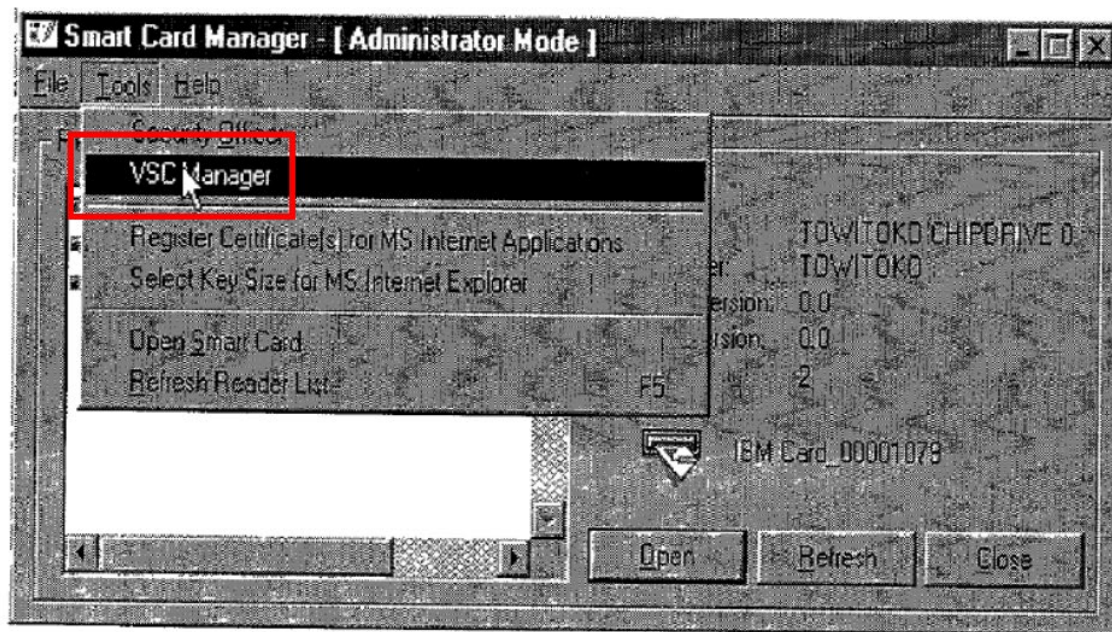


FIG. 3G

301. Hamann describes this step as follows. Hamann, ¶[0050] (“The virtual smart card control program may be started from the ‘Tools’ menu as shown in FIG. 3G.”). This “opens a menu having a button for creating a new VSC,” as shown in Fig. 3H (annotated below):

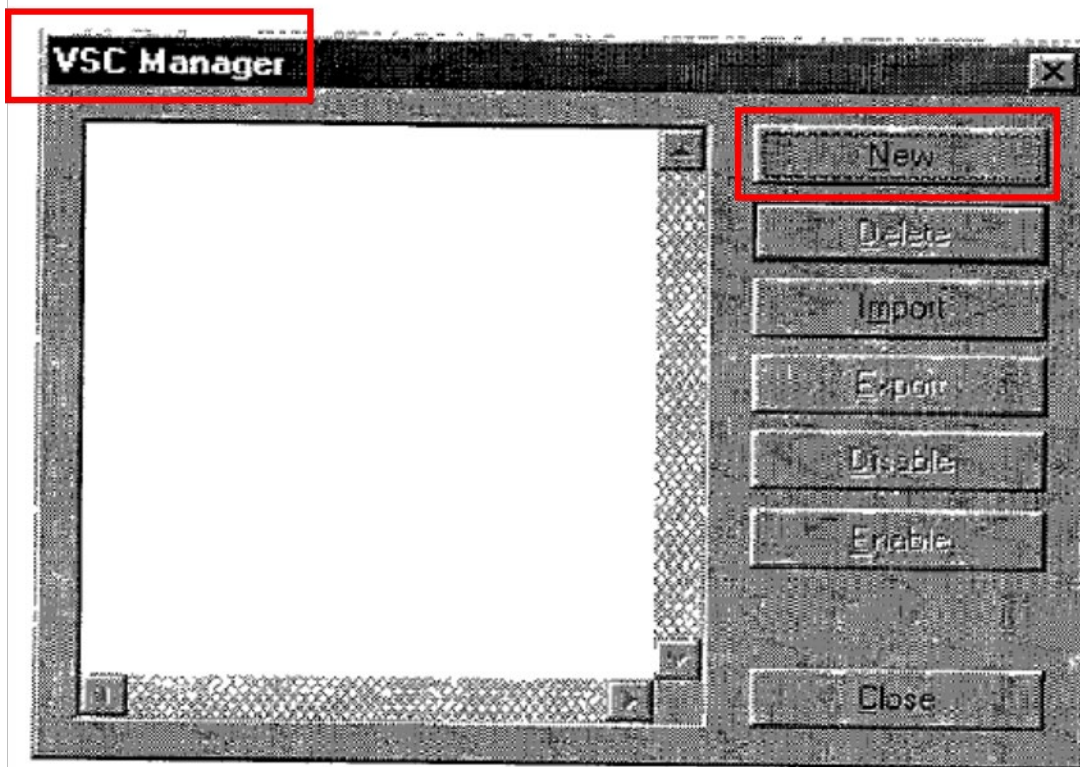


FIG. 3H

302. Hamann discloses that “[a] new VSC can be created by pressing the button ‘New.’” EX1007, Hamann, ¶[0050]. As I discussed in §§XII.A.1.b-XII.A.1.c above, Hamann discloses creating an ID for the virtual smart card (analogous to the media ID of an SD card) that is “specified using the serial number of the real smart to be assigned to the new VSC.” *Id.* This is shown in Fig. 3I (annotated below):

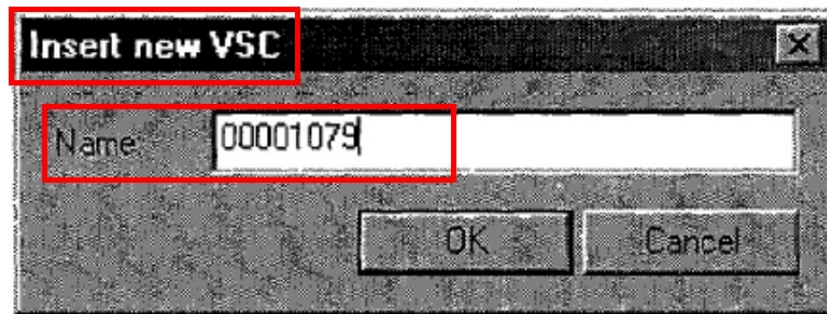


FIG. 3I

303. Once the virtual smart card has been created with the ID based on the real smart card, the user clicks on the “Close” button shown in Fig. 3J (annotated below):

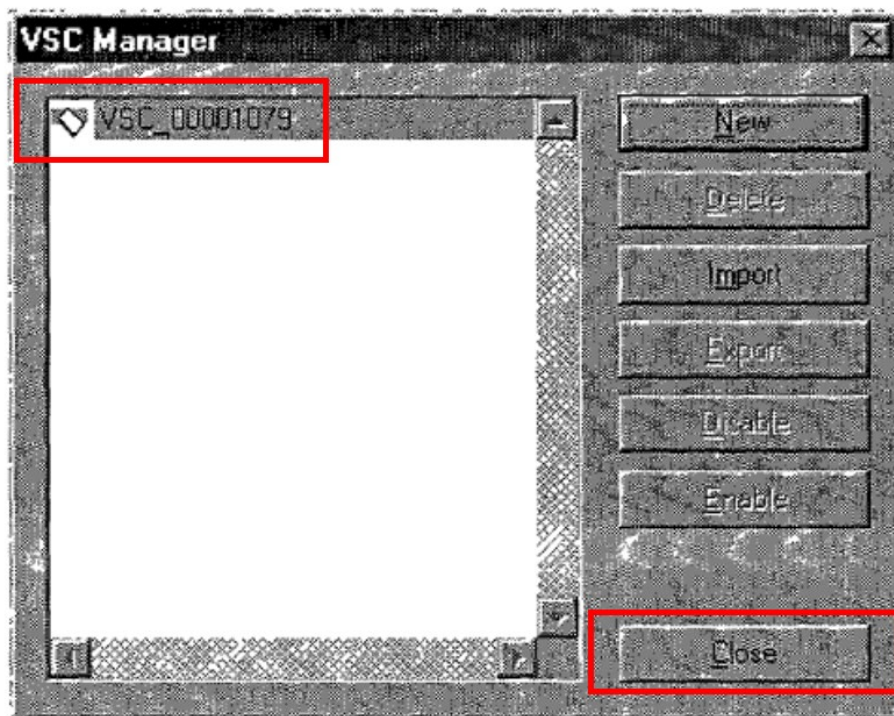


FIG. 3J

EX1007, Hamann, ¶[0050].

304. As I discussed in §XII.A.3.b, Hamann’s VSC manager then “displays the created VSC in the ‘Reader List’ as ‘IBM Virtual Smart Card’ and the smart [card] label ‘VSC 00001079’”:

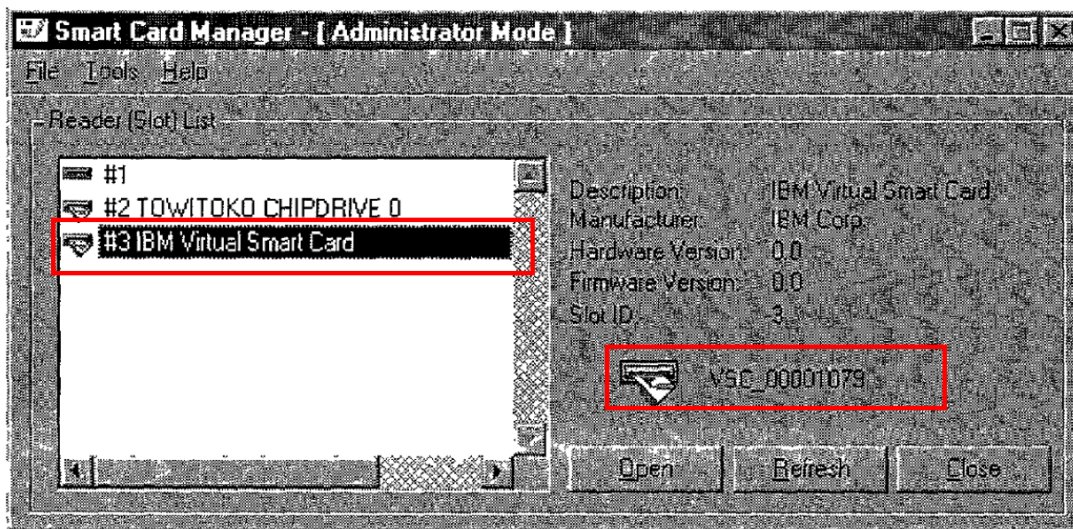


FIG. 3K

EX1007, Hamann, ¶[0051].

305. Thus, POSITAs would have understood that Hamann discloses provisioning a virtual smart card in response to the virtual smart card not being provisioned.

306. As I discussed in §XII.A.1.a above, Hamann does not explicitly disclose virtualizing SD cards. However, as I also discussed there, SD cards fall within the scope of Hamann’s smart card definition. *See also* §VII.B.1 above.

Moreover, Hamann would have motivated POSITAs to virtualize SD cards like those described in Maeda and the AAPA, as I discussed in §XII.A.2 above.

307. In sum, in my opinion, the Maeda-Hamann-AAPA combination discloses and/or suggests provisioning the virtual SD card in response to the virtual SD card not being provisioned [*e.g.*, Hamann’s creating a new VSC with a unique ID, via the “VSC Manager” GUI].

- d. **[14c] “creating a file system for secure data on a storage device for storage of secure data corresponding to a secure data area of the SD card; and”**

308. In my opinion, the Maeda-Hamann-AAPA combination discloses/suggests this limitation for the same reasons as I discuss for claim limitation 1c. *See* §XII.A.1.d above.

- e. **[14d] “creating a file system for user data on the storage device for storage of user data corresponding to a user data area of the SD card,”**

309. In my opinion, the Maeda-Hamann-AAPA combination discloses/suggests this limitation for the same reasons as I discuss for claim limitation 1d. *See* §XII.A.1.e above.

- f. [14e] “wherein a size of the file system for secure data is determinant on a size of the file system for user data, the determination based on a lookup table providing a relationship between the sizes;”

310. In my opinion, the Maeda-Hamann-AAPA combination discloses/suggests this limitation for the same reasons as I discuss for claim limitation 1e. *See* §XII.A.1.f above.

- g. [14f] “accessing the data stored in the secure area of the provisioned virtual SD card; and”

311. The Maeda-Hamann-AAPA combination discloses/suggests this feature. POSITAs would have understood that data can be accessed from the secure area of physical SD cards, as disclosed in Maeda. *See e.g.*, EX1006, Maeda, 13:42-45 (“The following is an explanation of the secure processing unit 11 provided in order to perform secure reading and writing to and from the protected area in the nonvolatile memory 1.”).

312. Likewise, the AAPA also admits that data can be accessed from the secure area of physical SD cards. ’749 patent, 4:43-63 (“Typical SD Card Architecture and Functions ... Both secure data and unsecure (user) data are stored in a persistent manner so that it can be retrieve at a later time.”).

313. Hamann discloses that virtual smart cards are “a software implemented version of a real smart card providing the equivalent functionality of a real smart card.” EX1007, Hamann, ¶[0015]. Hamann further teaches that its virtual smart

cards can be read from and written to, as it discloses that “[t]he VSC is created and used by a VSC control program handling the creation, the security **and the read/write process of the VSC.**” *Id. See also id.*, ¶[0017] (“All data objects can be **stored and retrieved on/from the virtual smart card’s public and private area** via the virtual smart card control program using the communication component.”).

314. Finally, as I discussed in §XII.A.1.d above, Hamann’s virtual smart card has a “private area” (*see* EX1007, Hamann, ¶¶[0016], [0027]) corresponding to the “file system for secure data on a storage device for storage of secure data corresponding to a secure data area of the SD card,” just as claimed.

315. Given this, POSITAs would have understood that the data stored in the Maeda-Hamann-AAPA virtual SD card that was provisioned could be accessed. Indeed, POSITAs would have recognized that one purpose of virtualizing a physical SD card, as suggested by Hamann, would be to allow accessing data stored in the virtual secure area. *Id. See also* §VII.E above.

316. In sum, in my opinion, the Maeda-Hamann-AAPA combination discloses and/or suggests accessing the data stored in the secure area [*e.g.*, Maeda-AAPA’s securely reading/writing the protected area, Hamann reading/writing the VSC’s private area] of the provisioned virtual SD card [*e.g.*, Maeda-AAPA’s SD card virtualized by Hamann’s created smart card].

h. [14g] “extracting the data from the secured area of the provisioned virtual SD card.”

317. The Maeda-Hamann-AAPA combination discloses/suggests this feature. According to the '749 patent, “extracting” data refers to decrypting data stored therein to return data from the secured area of the virtual SD card. *See* '749 patent, 8:32-38 (“If a **read operation, the process fetches and decrypts 560 data from the virtual secure area file system 315** using the key derived from the computing device unique information. The process encrypts 565 the data with the AKE key for transport to the application program 105. **The process returns (or retrieves or extracts) data from the virtual secure area 315** to the application program 105.”).

318. Maeda discloses that when reading data from its protected area (*i.e.*, the physical SD card’s secured area), the data is decrypted, *i.e.*, “extracted.” *See e.g.*, EX1006, Maeda, 15:2-8 (“**Reading and writing of data to and from the protected area is performed following decryption** performed by the Ks decrypting unit 16 and encryption performed by the Ks encrypting unit 17, so the protected area can only be accessed when the semiconductor memory card 100 and a connected access apparatus 200 have performed AKE processing successfully.”).

319. As I discussed in §XII.A.3.a above, Hamann discloses that its virtual smart cards have functionality that is equivalent to real smart cards, that its virtual

smart cards have a “private area” and that the data stored in this private area can be accessed. *See* EX1007, Hamann, ¶¶[0015]-[0016], [0027].

320. Hamann also discloses that data stored in its virtual smart card’s private area is encrypted, and that it can be “**accessed [by] providing** a password (10), e.g. account no (12) **and key information** (14).” *Id.*, ¶[0027]. Thus, POSITAs would have understood that Hamann also discloses decrypting, *i.e.*, “extracting,” data from its private area.

321. POSITAs thus would have understood that when virtualizing an SD card like that described in Maeda using Hamann’s virtualization teachings, the data stored in Maeda’s protected area (which corresponds to the claimed “secure area”), would be decrypted, *i.e.*, “extracted.”

322. POSITAs would have been motivated to combine these references for the reasons discussed in §XII.A.2 above. Thus, the Maeda-Hamann-AAPA combination renders claim 14 obvious.

323. In sum, in my opinion, the Maeda-Hamann-AAPA combination discloses and/or suggests extracting the data from the secured area [*e.g.*, Maeda-AAPA’s reading/writing data from secure area after decrypting it, Hamann accessing the VSC’s data and decrypting it using password and key information] of the provisioned virtual SD card.

4. Claim 15: “The method of claim 14, further comprising performing an authentication using Authenticated Key Encryption (AKE) before accessing the data in the secure area of the provisioned virtual SD card.”

324. As an initial matter, the ’749 patent’s written description explains AKE as “Authenticated Key Exchange”. ’749 patent, 5:42-55. In the rest of the written description, the ’749 patent’s describes exchanging keys using some form of authentication. The written description also discloses encrypting files using keys. But nowhere does the written description explains what “authenticated key **encryption**” is or suggests that is what “AKE” stand for. Conversely, the term “Authenticated Key **Encryption**” appears six times in the claims, and also abbreviated “AKE”. Therefore, the ’749 patent at least appears to use the abbreviation “AKE” inconsistently and possibly to mean two different things.

325. A POSITA would have known of techniques such as “authenticated key **exchange**” (*e.g.*, using public/private key pairs, Diffie-Hellman’s key-exchange protocol) and “encryption” (*e.g.*, using ciphers such as AES). See https://en.wikipedia.org/wiki/Authenticated_Key_Exchange (last viewed September 29, 2024). A POSITA would have also been familiar with “Authenticated Encryption” (AE) techniques. See https://en.wikipedia.org/wiki/Authenticated_encryption (last viewed September 29,

2024). However, “Authenticated Key **Encryption**,” as claimed does not appear to be a known technique—and if its, then it is not described in the ’749’s specification.

326. Therefore, it is not clear what the ’749 patent means when it claims “Authenticated Key **Encryption**.” Therefore, I will assume it meant for the claimed AKE to mean “authenticated key exchange.” To the extent Patent Owner argues that the AKE terms in the claims mean something other than what is in the specification, then I reserve the right to respond to such arguments.

327. Nevertheless, the Maeda-Hamann-AAPA combination discloses/suggests this claim. Maeda discloses that “the protected area **can only be accessed when the semiconductor memory card 100** and a connected access apparatus 200 **have performed AKE processing successfully.**”) EX1006, Maeda, 15:2-8. POSITAs would have therefore understood that Maeda’s physical SD cards perform the claimed “authentication using Authenticated Key Encryption (AKE) before accessing the data in the secure area” of physical SD cards.

328. I also note that Maeda states that “AKE” stands for “Authentication and Key Exchange.” EX1006, Maeda, 13:62-63. As I discussed just above, while claim 15 of the ’749 patent refers to “AKE” as “Authenticated Key *Encryption*,” POSITAs would have understood that Maeda’s terminology and the “AKE” described in the ’749 patent specification (*see e.g.*, ’749 patent, 5:44-49 referring to “Authenticated Key *Exchange*”) is correct (unless patentee really did mean

authenticated key encryption, in which case POSITAs would have no idea what the claim means since the specification never explains what this is). The '749 patent also states that AKE is performed in accordance with the Content Protection for Recordable Media (CPRM) specifications ('749 patent, 5:44-49), which POSITAs would have known refers to the same "AKE" process as in Maeda.

329. As I previously discussed, Hamann discloses that its virtual smart cards are "a software implemented version of a real smart card providing **the equivalent functionality of a real smart card.**" EX1007, Hamann, ¶[0015]. Thus, POSITAs would have recognized that when applying Hamann's teachings to Maeda, the resulting virtual SD card would have the "equivalent functionality" as a physical SD card, meaning that a virtual SD card would function in essentially the same. *See also* §VII.E above.

330. Indeed, POSITAs would have understood that for a virtual SD card to operate in the same manner with the same level of security as a physical SD card, the virtual version would have to perform authentication in the same manner as the real version. As Maeda discloses, SD cards perform AKE before accessing the data in their secure area, which is just as is claimed. Thus, claim 15 is obvious.

331. In sum, in my opinion, the Maeda-Hamann-AAPA combination discloses and/or suggests the method of claim 14, further comprising performing an authentication using Authenticated Key Encryption (AKE) [*e.g.*, Maeda-AAPA

using AKE] before accessing the data in the secure area of the provisioned virtual SD card.

5. Claim 16: “The method of claim 14 or 15, wherein data received from an application program through Authenticated Key Encryption (AKE) is first decrypted.”

332. As an initial matter, I assume this limitation meant to claim “Authenticated Key **Exchange** (AKE),” as discussed in §XII.A.4 above.

333. The Maeda-Hamann-AAPA combination discloses/suggests this claim. The '749 patent discloses that application programs are located on the computer system in which the virtual SD card is implemented. *See e.g.*, '749 patent, Fig. 3. Because this claim refers to “data received from an application program,” POSITAs would have understood that this claim is directed to a “write” operation, which corresponds to at least part of step 555 shown in Fig. 5. '749 patent, 8:22-28.

334. Indeed, the '749 patent discloses that when writing data to the virtual SD card's secure area, its “process decrypts 555 data with the key derived during AKE, reads from storage, decrypts the with encryption key, updates data, encrypts with the encryption key, and writes data to the virtual secure area file system 315 of the virtual SD card 300.” '749 patent (EX1001), 8:22-27.

335. This can also be seen in the portion of Fig. 5 of the '749 patent reproduced below:

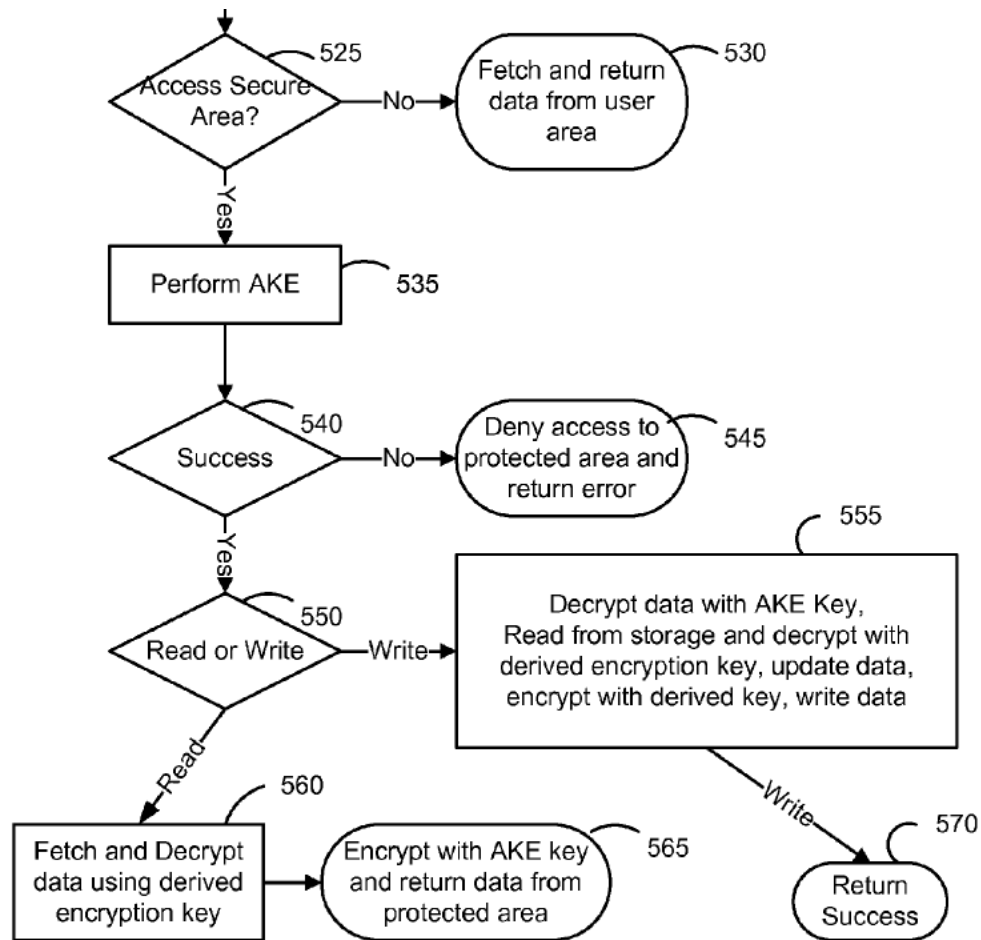


FIG. 5

336. As can be seen in Fig. 5, when accessing the virtual SD card's secure area 315 (step 525), AKE is performed (step 535). If the AKE key exchange is successful (step 540), and data is to be written to the virtual secure area 315, the process advances to step 555. As seen in step 555, the data to be written is decrypted with the AKE key, and after updating data, the data is then encrypted using a key derived from computer system information. A POSITA would have understood the sub-steps listed in step 555 to mean that the latter encryption is performed with

another key because that key is derived anew; it would make little sense to derive a new key and then ignore it by using an older key.

337. In my opinion, POSITAs would have understood that decrypting data “with the key derived during AKE” is recited in claim 16, and is rendered obvious by the Maeda-Hamann-AAPA combination.

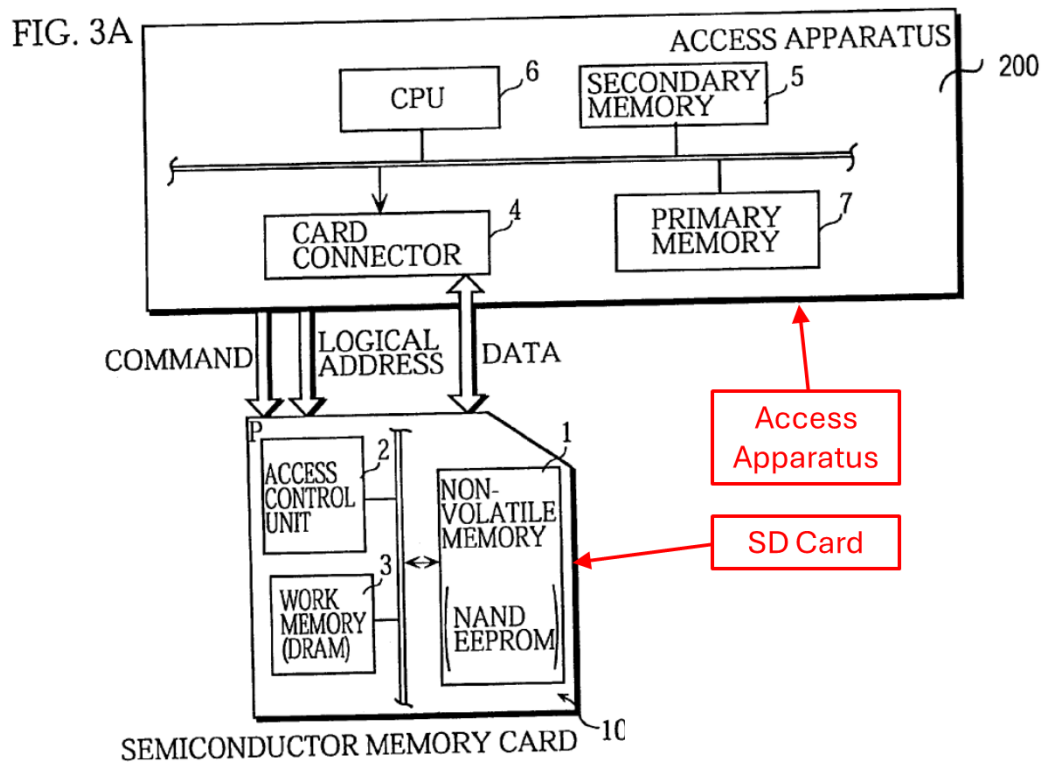
338. Note that POSITAs would have understood that claim 17 is directed to the second part of this process, “encrypt[ing] with the encryption key, and writ[ing] data to the virtual secure area file system ...,” which I discuss in §XII.C.5 below.

339. For this process, the ’749 patent discloses “that there are two encryption keys.” ’749 patent (EX1001), 8:27-28. The first key “is for physical storage,” while “[t]he second key is calculated during AKE.” *Id.*, 8:28-29. Because claim 16 recites that “data received from an application program through Authenticated Key Encryption (AKE) is first decrypted,” POSITAs would have understood that it is directed to the second key.

340. Maeda discloses that “[w]hen **encrypted data to be written into the protected area** is output from the access apparatus 200 connected to the semiconductor memory card 100, **the Ks decrypting unit 16 uses the session key Ks to decrypt this data** which has already been encrypted by session key Ks before being output.” EX1006, Maeda, 14:57-62. Maeda also discloses that “data **obtained by this decryption** is written into the protected area, as the original data.” *Id.*, 62-

63. Note that Maeda's "access apparatus" is the device the SD card is inserted into.

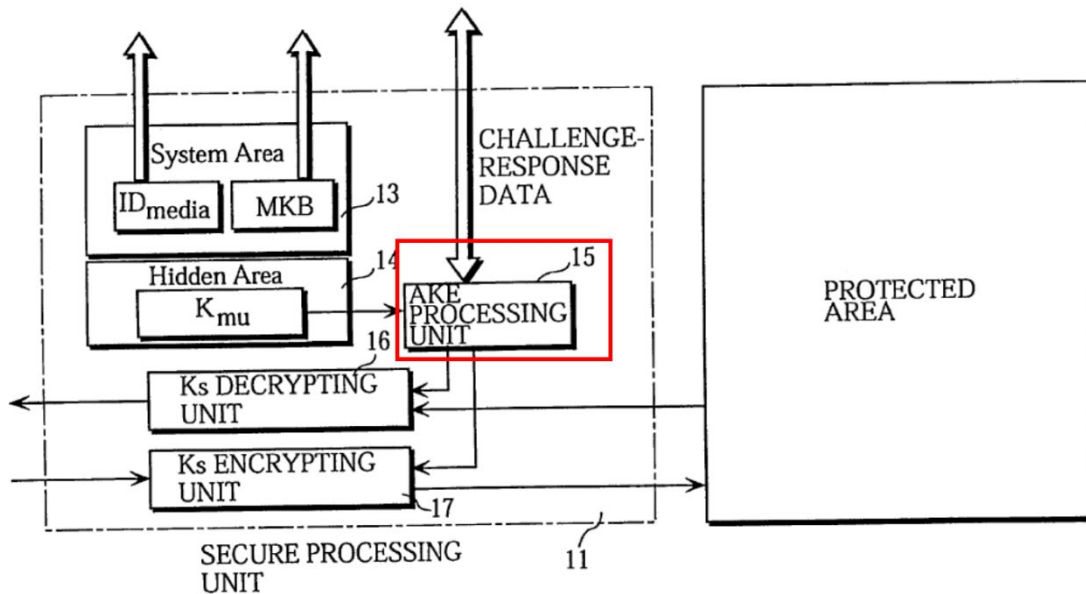
See e.g., Fig. 3A (annotated below):



341. Examples of such "access apparatus 200" include personal computers. See EX1006, Maeda, 6:25-28. POSITAs would have understood that personal computers would have application programs running thereon that would write data to the connected SD card. See also §VII.A above.

342. Maeda also discloses "AKE processing unit 15," shown in Fig. 15 (annotated below):

FIG. 15



343. As shown, Maeda's AKE processing unit 15 is coupled to Ks decrypting and encrypting units 16 and 17, respectively. Maeda discloses that for data to be successfully written to the protected area, it must have gone through an AKE process. EX1006, Maeda, 15:2-8 ("Reading and writing of data to and from the protected area is performed following decryption performed by the Ks decrypting unit 16 and encryption performed by the Ks encrypting unit 17, so **the protected area can only be accessed when the semiconductor memory card 100 and a connected access apparatus 200 have performed AKE processing successfully.**")

344. Thus, Maeda discloses that in physical SD cards, "data received from an application program through Authenticated Key Encryption (AKE) is first decrypted," just as claimed.

345. As I discussed in §XII.A.3.a above, Hamann discloses that its virtual smart card is “a **software implemented version of a real smart card** providing the **equivalent functionality of a real smart card**.” *Id.*, ¶[0015]. Thus, POSITAs would have understood that when applying Hamann’s teachings regarding virtualization of smart cards to Maeda, the resulting virtual SD card would have functioned in the same manner as Maeda’s physical SD card. Thus, claim 16 is obvious. *See also* §VII.E above.

346. In sum, in my opinion, the Maeda-Hamann-AAPA combination discloses and/or suggests the method of claim 14 or 15, wherein data received from an application program through Authenticated Key Encryption (AKE) [*e.g.*, Maeda-AAPA using AKE] is first decrypted.

6. Claim 18

- a. [18pre] “A non-transitory computer readable medium configured to store instructions that, when executed, cause at least one processor to:”

347. In my opinion, the Maeda-Hamann-AAPA combination discloses/suggests this limitation for the same reasons as I discuss for claim limitation 1pre. *See* §XII.A.1.a above.

348. Note that Hamann discloses that it virtualizes smart cards using “virtual smart card control program 18,” which Hamann discloses is stored on a computer usable medium. EX1007, Hamann, claim 8. POSITAs would have understood that

a such a program would comprise instructions and that such instructions would be stored in a non-transitory medium and executed by a processor. *See* §§VII.A and VII.B.1 above. POSITAs would have been motivated to combine Maeda, Hamann, and the AAPA for the reasons discussed in §XII.A.2 above.

349. In sum, in my opinion, the Maeda-Hamann-AAPA combination discloses and/or suggests a non-transitory computer readable medium [*e.g.*, Hamann running software that resides in a non-transitory medium, memory, and a processor] configured to store instructions.

b. [18a] “read a media key block and media identification for an SD card;”

350. In my opinion, the Maeda-Hamann-AAPA combination discloses/suggests this limitation for the same reasons as I discuss for claim limitation 1a. *See* §XII.A.1.b above.

c. [18b] “store the media key block and the media identification;”

351. In my opinion, the Maeda-Hamann-AAPA combination discloses/suggests this limitation for the same reasons as I discuss for claim limitation 1b. *See* §XII.A.1.c above.

- d. **[18c] “create a file system for secure data on a storage device for storage of secure data corresponding to a secure data area of the SD card; and”**

352. In my opinion, the Maeda-Hamann-AAPA combination discloses/suggests this limitation for the same reasons as I discuss for claim limitation 1c. *See* §XII.A.1.d above.

- e. **[18d] “create a file system for user data on the storage device for storage of user data corresponding to a user data area of the SD card,”**

353. In my opinion, the Maeda-Hamann-AAPA combination discloses/suggests this limitation for the same reasons as I discuss for claim limitation 1d. *See* §XII.A.1.e above

- f. **[18e] “wherein a size of the file system for secure data is determinant on a size of the file system for user data, the determination based on a lookup table providing a relationship between the sizes.”**

354. In my opinion, the Maeda-Hamann-AAPA combination discloses/suggests this limitation for the same reasons as I discuss for claim limitation 1e. *See* §XII.A.1.f above.

7. **Claim 22**

- a. **[22pre] “A non-transitory computer readable medium configured to store instructions that, when executed, cause at least one processor to:”**

355. In my opinion, the Maeda-Hamann-AAPA combination discloses/suggests this limitation for the same reasons as I discuss for claim limitation 14pre. *See* §XII.A.3.a above.

356. POSITAs would have understood that Hamann discloses the “non-transitory computer readable medium” features of this claim. *See* §XII.A.6.a above. For motivation to combine, *see* §XII.A.2 above.

- b. **[22a] “determine if a virtual SD card is provisioned;”**

357. In my opinion, the Maeda-Hamann-AAPA combination discloses/suggests this limitation for the same reasons as I discuss for claim limitation 14a. *See* §XII.A.3.b above.

- c. **[22b] “provision the virtual SD card in response to the virtual SD card not being provisioned, wherein to provision the virtual SD card includes to:”**

358. In my opinion, the Maeda-Hamann-AAPA combination discloses/suggests this limitation for the same reasons as I discuss for claim limitation 14b. *See* §XII.A.3.c above.

- d. **[22c] “create a file system for secure data on a storage device for storage of secure data corresponding to a secure data area of the SD card; and”**

359. In my opinion, the Maeda-Hamann-AAPA combination discloses/suggests this limitation for the same reasons as I discuss for claim limitation 14c. *See* §XII.A.3.d above.

- e. **[22d] “create a file system for user data on the storage device for storage of user data corresponding to a user data area of the SD card,”**

360. In my opinion, the Maeda-Hamann-AAPA combination discloses/suggests this limitation for the same reasons as I discuss for claim limitation 14d. *See* §XII.A.3.e above.

- f. **[22e] “wherein a size of the file system for secure data is determinant on a size of the file system for user data, the determination based on a lookup table providing a relationship between the sizes;”**

361. In my opinion, the Maeda-Hamann-AAPA combination discloses/suggests this limitation for the same reasons as I discuss for claim limitation 14e. *See* §XII.A.3.f above.

- g. **[22f] “access the data stored in the secure area of the provisioned virtual SD card; and”**

362. In my opinion, the Maeda-Hamann-AAPA combination discloses/suggests this limitation for the same reasons as I discuss for claim limitation 14f. *See* §XII.A.3.g above.

h. [22g] “extract the data from the secured area of the provisioned virtual SD card.”

363. In my opinion, the Maeda-Hamann-AAPA combination discloses/suggests this limitation for the same reasons as I discuss for claim limitation 14g. *See* §XII.A.3.h above.

8. Claim 23 “The computer readable medium of claim 22, further comprising: performing an authentication using Authenticated Key Encryption (AKE) before accessing the data in the secure area of the provisioned virtual SD card.”

364. In my opinion, the Maeda-Hamann-AAPA combination discloses/suggests this limitation for the same reasons as I discuss for claim 15. *See* §XII.A.4 above

9. Claim 24 “The computer readable medium of claim 23, wherein data received from an application program through Authenticated Key Encryption (AKE) is first decrypted.”

365. In my opinion, the Maeda-Hamann-AAPA combination discloses/suggests this limitation for the same reasons as I discuss for claim 16. *See* §XII.A.5 above.

B. Ground 2: Maeda In View Hamann, the AAPA, and Hirota Renders Claims 2, 6-8, and 19 Obvious

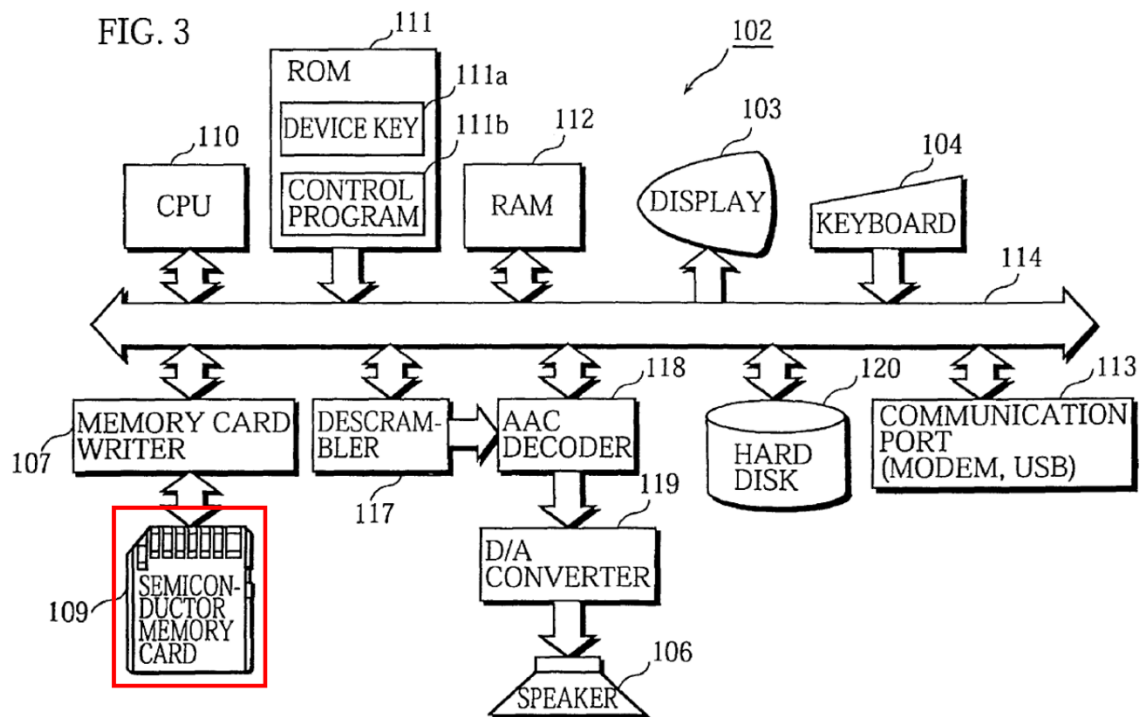
1. Claim 2: “The method of claim 1, wherein the file system for secure data is encrypted.”

366. As an initial matter, I note that encrypted file systems were well known in the art (and were also a subject of my own research). *See* §§VII.C-VII.E above.

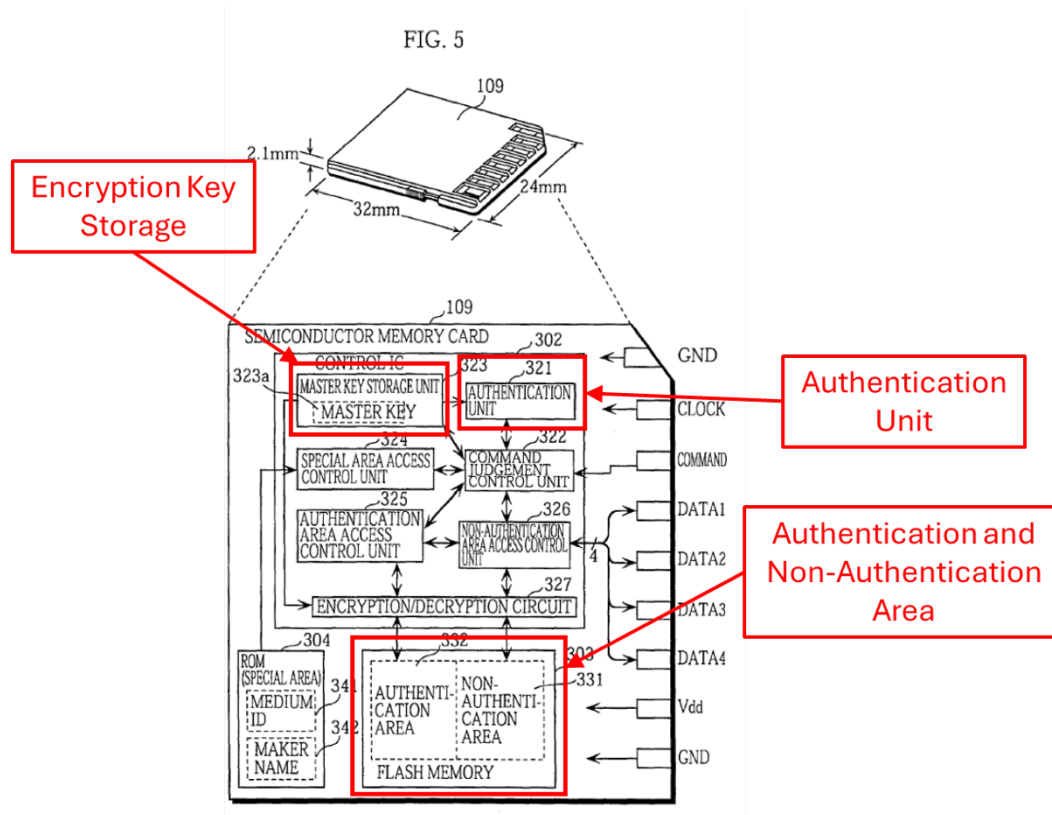
367. The Maeda-Hamann-AAPA-Hirota combination discloses/suggests this claim. As I discussed in §XII.A.1.d above, Maeda and the AAPA disclose that physical SD cards have a “protected area” that, when virtualized using Hamann’s teachings, results in a “file system,” *i.e.*, a “storage area” of a virtual SD card, “for secure data on a storage device for storage of secure data corresponding to a secure data area of the SD card,” just as claimed.

368. Claim 2 requires that the “file system for secure data” be encrypted. Maeda discloses that encrypted data is “to be written into the protected area....” (EX1006, Maeda, 14:57-63) but does not explicitly state that its protected area is encrypted.

369. Hirota, like Maeda, discloses a semiconductor memory card that can be inserted into a computer, as shown in Fig. 3 (annotated below):

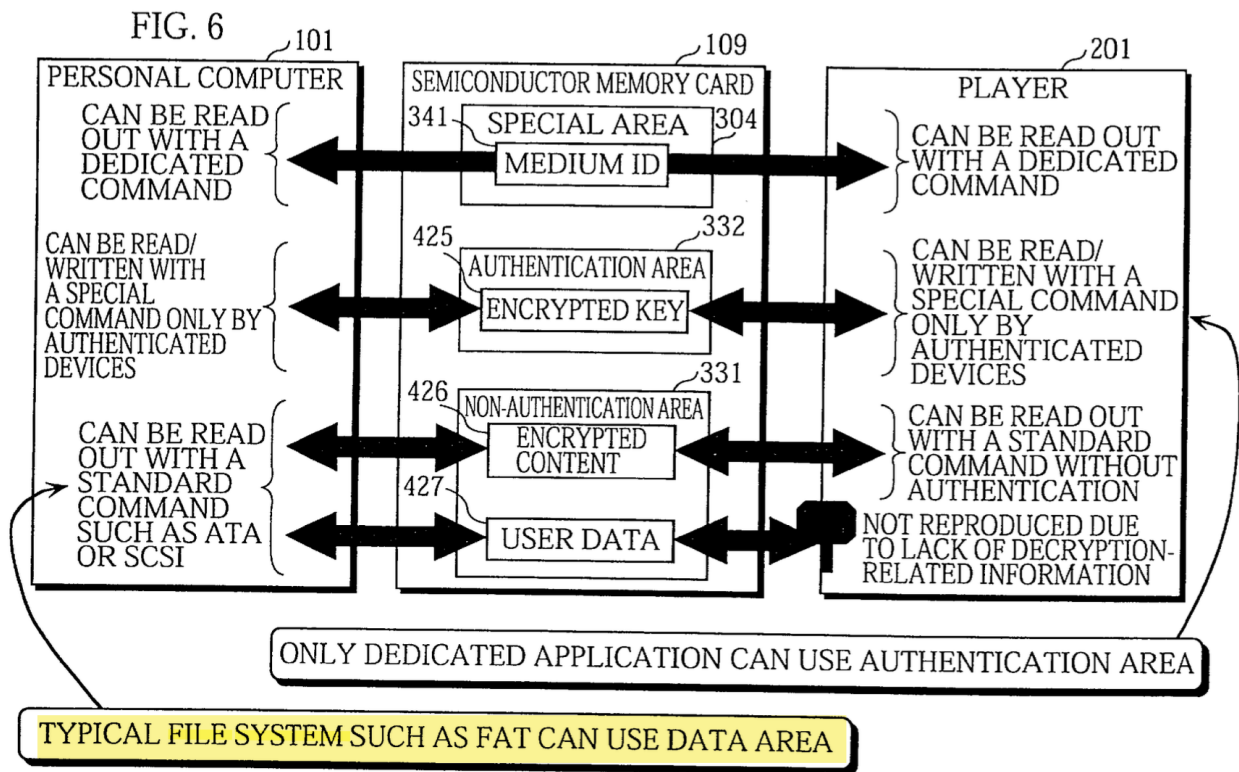


370. Hirota’s semiconductor memory card “can be used as a storage medium for storing digital contents and as a storage medium for storing general-purpose computer data (not an object of copyright protection).” EX1008, Hirota at 2:3-7. Like Maeda, Hirota discloses that its semiconductor memory card includes flash memory 303 having “an authentication area and a non-authentication area,” which is shown in Fig. 5 (annotated below):



371. Hirota explains how these two areas are used. EX1008, Hirota, 10:27-36 (“The authentication area 332 can be accessed only by the apparatuses that have been authenticated as proper apparatuses. The non-authentication area 331 can be accessed by any apparatuses whether they are authenticated or not. In the present embodiment, the authentication area 332 is used for storing important data related to copyright protection, and the non-authentication area 331 is used as an auxiliary storage apparatus in a typical computer system.”).

372. Hirota’s Fig. 6 (annotated below) provides additional information regarding its storage architecture:



373. Hirota discloses using file systems such as FAT. Hirota also suggests that a “typical file systems such as FAT can use data area.” *Id.*, Fig. 6 (annotated above). EX1008, Hirota, 21:41-47 (“FIG. 14C is a flowchart showing the procedure of the PC 102 or the player 201 for deleting blocks beforehand using the not-deleted block list command and the delete command in the above-stated states. It is presumed here that the flash memory 303 contains a table such as **FAT (File Allocation Table)** which shows the use state of the logical blocks, as shown in FIG. 14D.”), 9:23-28 (“The PC 102 can perform the following: (1) use the memory card 109 as an auxiliary storage apparatus having an independent **file system (e.g.,**

ISO9293) as hard disks have by executing the file management software program stored in the hard disk 120”).

374. As can be seen, Hirota’s “**authentication area 332 stores an encryption key 425** which is a secret key used for decrypting the encrypted content 426 stored in the non-authentication area 331.” EX1008, Hirota, 12:51-54. Thus, Hirota’s “authentication area” is similar to Maeda’s “protected area,” since Maeda teaches that its “protected area is used to store secret[] information, **such as an encryption key used for encrypting data**” EX1006, Maeda, 2:67-3:2.

375. POSITAs thus would have understood that Hirota’s authentication area and Maeda’s protected area (as well as the AAPA’s secure area) are directed to essentially the same structure.

376. Hirota also discloses that its authentication area is encrypted. *See* EX1008, Hirota, 2:41-45 (“... the instruction to access the **authentication area has been encrypted** ... such a semiconductor memory card has a reliable function to **protect an unauthorized access of the authentication area**.”); 11:58-12:4 (“The encryption/decryption circuit 327 is a circuit which **performs encryption** and decryption **using the master key 323a** stored in the master key storage unit 323.... The encryption/decryption circuit 327 **encrypts data before writing the data to the flash memory 303**.... These encryption and decryption are performed to prevent unauthorized acts such as an act of disassembling the memory card 109, analyzing

the contents of the flash memory 303 directly, and **stealing the password from the authentication area 332.**”).

377. Thus, POSITAs would have understood that the when the SD cards described in the Maeda-AAPA-Hirota combination are virtualized using Hamann’s teachings, the resulting file system (*i.e.*, “storage area” of a virtual SD card) for secure data would be encrypted, just as claimed.

378. In sum, in my opinion, the Maeda-Hamann-AAPA-Hirota combination discloses and/or suggests the method of claim 1, wherein the file system [*e.g.*, Maeda’s storage area, Hirota’s file systems such as FAT] for secure data is encrypted.

2. Motivation To Combine Maeda, Hamann, the AAPA, and Hirota

379. I described the motivation to combine Maeda, Hamann, and the AAPA in §XII.A.2 above.

380. POSITAs would have been further motivated to combine Hirota with the Maeda-Hamann-AAPA combination. POSITAs would have recognized that encrypting the file system for secure data created when virtualizing the Maeda-AAPA SD card would have increased the security of the information stored therein, particularly in view of Hirota’s teachings that encryption keys are stored therein.

381. This is reinforced by Hamann, which teaches that its virtualized private area is also encrypted. *See* EX1007, Hamann, ¶[0027]. Thus, adding Hirota's encryption would be seen by POSITAs as not only providing a highly desirable feature, but also nothing more than combining known techniques for enhancing security in a predictable manner, with a reasonable expectation of success.

382. Indeed, well before 2010, data and file encryption techniques were well known in the art (and some were subject of my own research). By 2003, a wide range of techniques were available for data/file encryption methods as I detail in §VII.E above. A POSITA would have sought to protect important data via encryption. A POSITA would have had a lot of options (both open-source and commercial) to choose from to implement file or file system encryption. Given the widespread availability of file-encryption techniques, a POSITA would have had a strong expectation of success in adding Hirota to the Maeda-Hamann-AAPA combination, with no undue burden.

3. Claim 6: “The method of claim 2, wherein the file system for secure data and the file system for user data are located on different physical partitions of the storage device.”

383. As an initial matter, I note that well before 2010, computers already supported multiple storage devices, and each device supported multiple partitions. *See* §§VII.A-VII.B above. Formatting different devices and/or partitions for different uses was routine and even encouraged. For example, a POSITA would

have been motivated to place highly secure data in one partition and less sensitive data in another. This made it easier to apply different retention, backup, and reliability policies—and it also helped compartmentalize data of different security levels. *See* §VII.D above.

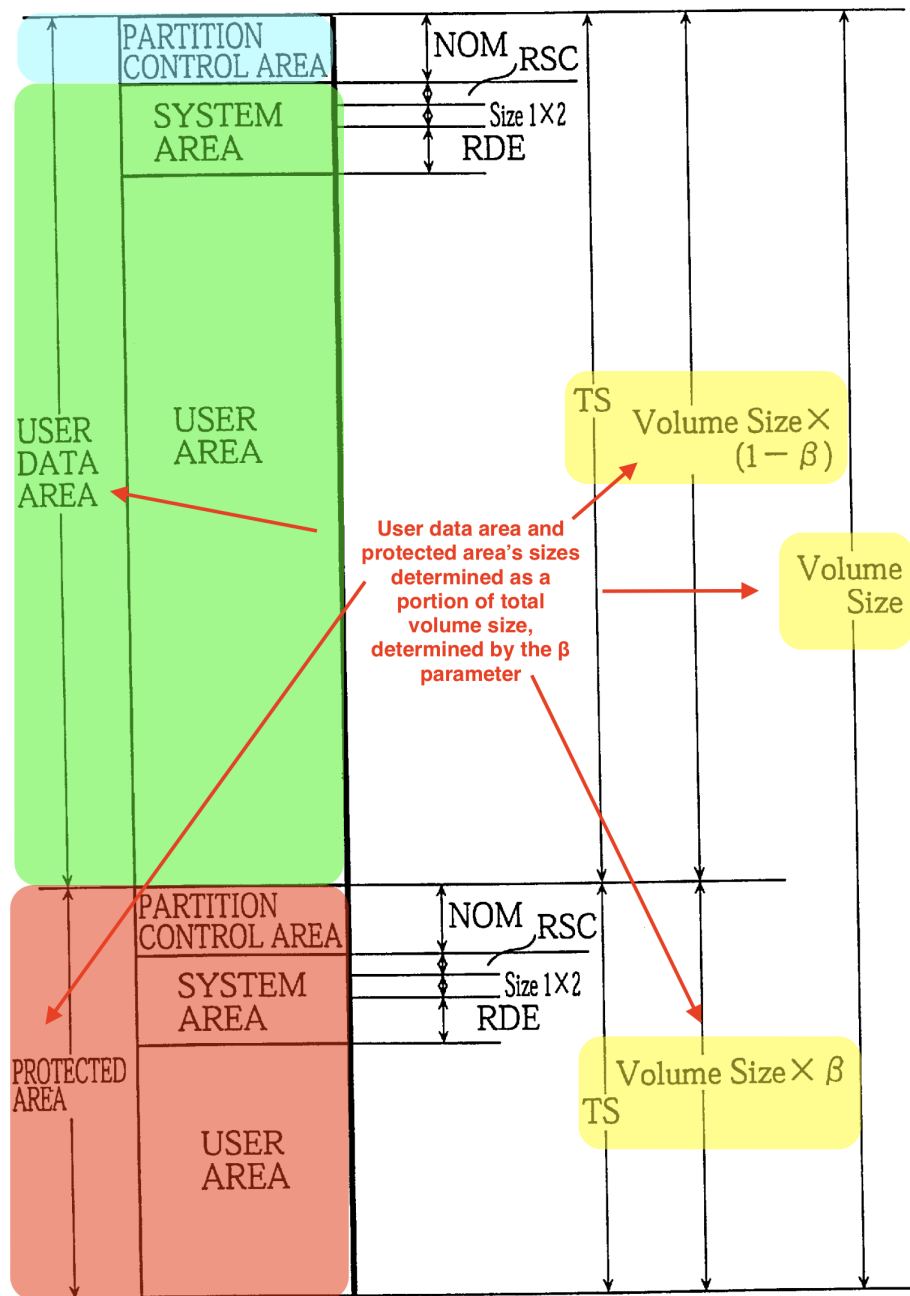
384. Nevertheless, the Maeda-Hamann-AAPA-Hirota combination discloses/suggests this feature. As I discuss in §XII.C.1 below, Hamann teaches its “virtual smart card (VSC) [has] the same defined logical file structure ... as its assigned real smart card.” EX1007, Hamann, ¶[0014]. POSITAs would have understood this file structure includes the hardware structure of the VSC’s architectural features, such as the layout of sectors and erasable blocks described in Maeda’s SD card. *See* EX1006, Maeda, 1:44-55, Figs. 1A, 1B. Thus, POSITAs would have recognized that Hamann was suggesting that the secure data and user data of its VSC reside on different physical partitions.

385. Despite this suggestion, Hamann does not disclose “the file system for secure data and the file system for user data are located on different physical partitions of the storage device.” However, the ’749 patent states that in “typical” SD cards, “[t]he **secured area 260 may be implemented as a physical or logical device** within the SD card as chosen by the SD card’s original equipment manufacturer (OEM)” (*id.* at 4:63-66) and “[t]he **user (or unsecured) area 265 also**

may be either physical or logical partitions depending on implementation within the SD card” (*id.* at 5:4-6). Thus, the feature recited in claim 6 is an AAPA.

386. Maeda’s Fig. 16 (annotated below) illustrates using partitions. Maeda, discloses using known FAT-based file systems, which use a partition table to determine the number and positions of the partitions. See “partition control area” annotated in blue in Fig. 16 below. A POSITA would have understood that FAT file systems include partitions that reside on the same device. *See also* §X.A above.

FIG. 16



387. Accordingly, POSITAs would have understood that when Hamann's teachings are used to virtualize an SD card as in the Maeda-AAPA-Hirota

combination, the resulting virtual SD card would have the same file structure as the original SD card, including the same physical partitions.

388. In sum, in my opinion, the Maeda-Hamann-AAPA-Hirota combination discloses and/or suggests the method of claim 2, wherein the file system for secure data and the file system for user data are located on different physical partitions of the storage device [*e.g.*, Hamann’s VSC using the same structure/content as a real smart card, such as logical/virtual partitions; Maeda-AAPA disclosing separate physical FAT partitions].

4. Claim 7: “The method of claim 2, wherein the file system for secure data and the file system for user data are located on different logical partitions of the storage devices.”

389. As an initial matter, I described in §VII.E above how the terms “logical” and “virtual” are often used interchangeably in this context. Moreover, just as physical device can be partitioned into physical partitions, so can a virtual (or logical) device be partitioned into virtual (or logical) partitions; from the perspective of the user or application, there is little meaningful operational difference. *See* §VII.D above.

390. The Maeda-Hamann-AAPA-Hirota combination discloses/suggests this feature. As I discuss in §XII.C.1 below, Hamann teaches its “virtual smart card (VSC) [has] the same defined logical file structure ... as its assigned real smart card.” EX1007, Hamann, ¶[0014]. POSITAs would have understood this file structure

includes the logical structure of the VSC, such as the file system layout (*e.g.*, FAT file system) of Maeda's SD card. *See* EX1006, Maeda, 1:55-2:2, Figs. 1C, 1D. Thus, POSITAs would have recognized that Hamann was suggesting that the secure data and user data of its VSC to be on different logical partitions.

391. Despite this suggestion, Hamann does not disclose “the file system for secure data and the file system for user data are located on different logical partitions of the storage device.” However, the '749 patent states that in “typical” SD cards, “[t]he **secured area 260 may be implemented as a physical or logical device** within the SD card as chosen by the SD card's original equipment manufacturer (OEM)” (*id.* at 4:63-66) and “[t]he **user (or unsecured) area 265 also may be either physical or logical partitions** depending on implementation within the SD card” (*id.* at 5:4-6). Thus, the feature recited in claim 7 is an AAPA.

392. Accordingly, POSITAs would have understood that when Hamann's teachings are used to virtualize an SD card as in the Maeda-AAPA-Hirota combination, the resulting virtual SD card would have the same file structure as the original SD card, including the same logical partitions.

393. In sum, in my opinion, the Maeda-Hamann-AAPA-Hirota combination discloses and/or suggests the method of claim 2, wherein the file system for secure data and the file system for user data are located on different logical partitions of the

storage devices [e.g., Hamann's VSC using the same structure/content as a real smart card, such as logical/virtual partitions].

5. Claim 8: "The method of claim 2, wherein the file system for secure data and the file system for user data are within an existing file system."

394. As an initial matter, I described in §VII.E above how file systems can be virtualized and even layered on top of each other. Thus, placing one file system inside another was well known in the art.

395. The Maeda-Hamann-AAPA-Hirota combination discloses/suggests this feature. The '749 patent discloses that its virtual SD card can be configured in either internal or external storage of a computer. '749 patent, 6:18-23. POSITAs would therefore have understood that the "existing file system" can be the file system of a computer on which the virtual smart card operates. As I discussed in §XII.A.3.a above, Hamann discloses that its virtual smart card "is a **software implemented** version of a real smart card providing the equivalent functionality of a real smart card," that is "used by a VSC control program...." EX1007, Hamann, ¶[0015].

396. POSITAs would therefore have understood that the Maeda-Hamann-AAPA-Hirota virtual SD card would have been implemented on the computer running Hamann's VSC control program. Thus, POSITAs would have also understood that the Maeda-Hamann-AAPA-Hirota virtual SD card's file systems for

secure data and user data would be “within an existing file system,” *i.e.*, an existing storage area on the computer running Hamann’s VSC control program. *Id.*

397. In sum, in my opinion, the Maeda-Hamann-AAPA-Hirota combination discloses and/or suggests the method of claim 2, wherein the file system for secure data and the file system for user data are within an existing file system [*e.g.*, Hamann’s virtualization software providing the equivalent functionality of a real smart card].

6. Claim 19: “The computer readable medium of claim 18, wherein the file system for secure data is encrypted.”

398. In my opinion, the Maeda-Hamann-AAPA-Hirota combination discloses/suggests this limitation for the same reasons as I discuss for claim 2. *See* §XII.B.1 above.

C. Ground 3: Maeda In View Hamann, the AAPA, Hirota and Lu Renders Claims 3, 4, 20, and 21 Obvious

1. Claim 3: “The method of claim 2, wherein the file system is encrypted with an encryption key, the encryption key derived from computer system information and the encryption key not stored by the computer system.”

399. As an initial matter, I described in §VII.E above that there are many forms of encrypting file system, who can take user-supplied encryption keys, randomly generate ones, construct them from fixed hardware information, and more. Thus, deriving a (cipher) key from computer system information was well known in

the art. I note also that this claim states that the encryption key is “not stored by the computer system,” suggesting that the computer system may have generated and used the key, but does not store it persistently anywhere. Using temporary, ephemeral, or regenerable encryption keys was well known in art. *See* §VII.D above.

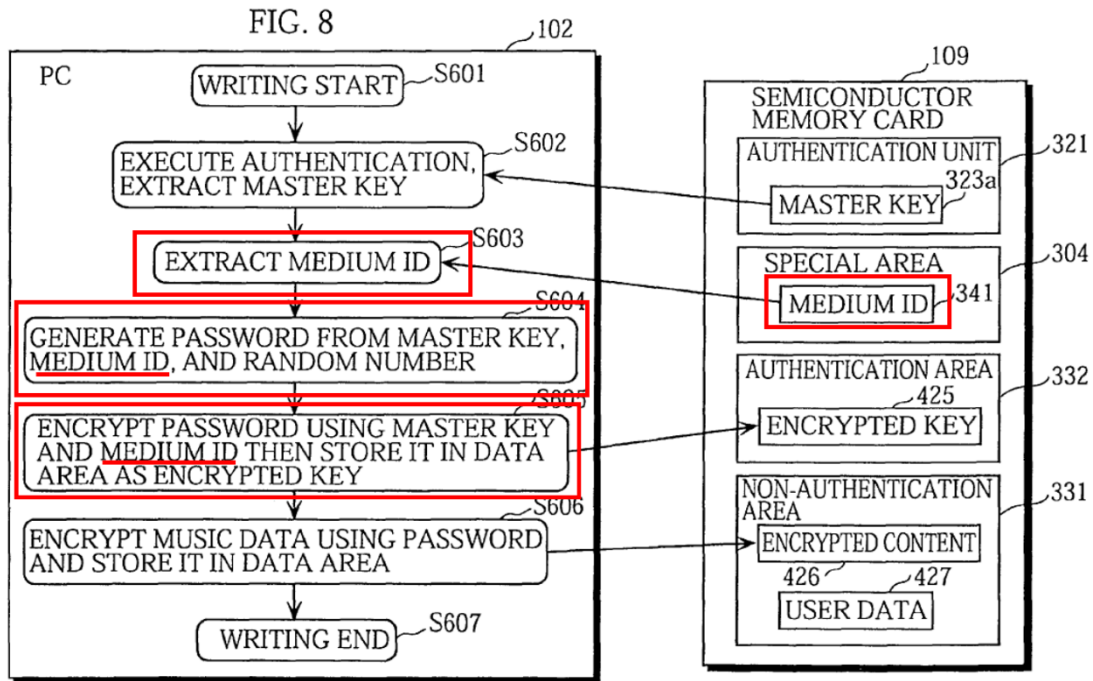
400. Moreover, whether keys are stored or not represents a trade-off of security vs. convenience. If the cipher key is not stored anywhere, then it must be supplied by a user (*e.g.*, from his/her own memory) before files can be decrypted. Alternatively, if the cipher key is stored in a different (*e.g.*, remote) location, then the user or system must retrieve the key first, which represents a delay in processing. This is less convenient, slower, but more secure. Conversely, if the cipher keys are, say, automatically stored on the computer system, they could then be automatically and quickly retrieved each time the user needs to decrypt files. This is more convenient and faster, but riskier: if the cipher key is compromised or stolen, a hacker could decrypt files and steal their contents.

401. The Maeda-Hamann-AAPA-Hirota-Lu combination discloses/suggests this claim. Claim 3 first requires that the recited “file system” be encrypted with an encryption key. POSITAs would have understood that encryption is almost always performed with an encryption key. *See* §VII.D above.

402. Regardless, Hirota discloses that its “authentication area,” *i.e.*, the claimed “file system for secure data,” is encrypted with an encryption key. *See e.g.*,

EX1008, Hirota, 2:41-45 (“... the instruction to access the **authentication area has been encrypted**... such a semiconductor memory card has a reliable function to **protect an unauthorized access of the authentication area.**”); 11:58-12:4 (“The encryption/decryption circuit 327 is a circuit which **performs encryption** and decryption **using the master key 323a** stored in the master key storage unit 323 ... The encryption/decryption circuit 327 **encrypts data before writing the data to the flash memory 303.**... These encryption and decryption are performed to prevent unauthorized acts such as an act of disassembling the memory card 109, analyzing the contents of the flash memory 303 directly, and **stealing the password from the authentication area 332.**”). *See also* EX1008, Hirota, 23:31-57.

403. Claim 3 also requires that the encryption key be “derived from computer system information.” According to the ’749 patent, such “computer system information” can be, “e.g., processor serial number, bios serial number, device serial number.” ’749 patent, 7:2-6. Hirota suggests using device data for deriving encryption keys, as it discloses that the encrypted key 425 stored in its authentication area uses a key that is derived, in part, from the “medium ID” of the SD card itself. (I note that the claim does not appear to require that the key be derived exclusively from computer system information.) This is shown in Fig. 8 (annotated below):



404. Hirota discloses that the “medium ID” is an “identifier of the memory card 109.” EX1008, Hirota, 10:41-42. To store encrypted key 425 in the authentication area, Hirota teaches that the card’s medium ID 341 is extracted by the PC in which the card is inserted, that a password is generated using, inter alia, the card’s medium ID, and encrypting this password, which then “is written to the authentication area 332 as the encryption key 425 (S605).” EX1008, Hirota, 14:58-15:3. A POSITA would have understood Hirota to disclose a process whereby the actual key is derived from the medium ID, including additional steps.

405. Still, I note that the encryption key in Hirota is not the encryption key used to encrypt the authentication area (*i.e.*, the claimed “file system for secure data”). Nevertheless, POSITAs would have understood these teachings to be a

suggestion to use device data to derive encryption keys. POSITAs would have understood that instead of using a data from an SD card to derive encryption keys, computer system information could be used to derive encryption keys, as described in Lu, which discloses associating encryption with a serial number for software/hardware binding and therefore discloses an encryption key derived from a computer system. *See* EX1009, Lu, p. 1 (describing an “encryption and decryption methods for implementing hardware and software binding.”⁷). To do so, Lu discloses a chip storing a unique serial number HW_UID “that represents the identity of the chip.” EX1009, Lu, p. 5. Lu uses a HASH to transform the HW_UID to a SW_UID, which in turn is used to create the encryption key. *Id. See also* §§VII.D-VII.E above. Thus, POSITAs would have understood that Lu discloses deriving an encryption key using a serial number of the hardware system.

406. Thus, when virtualizing an SD card like that in the combination of Maeda-AAPA-Hirota using Hamann, POSITAs would have understood that whenever the encryption key was needed to access the authenticated area, it could be derived from a hardware feature in the computer system using *Lu*’s teachings. POSITAs would have recognized that this means that the encryption key need not be saved by the computer system on which the virtual SD card is stored since it could

⁷ All *Lu* quotations are from the certified translation, EX1009.

be derived using Lu whenever it was needed. Thus, in the combination of Maeda-Hamann-AAPA-Hirota-Lu, the encryption key derived from the computer system need not be “stored by the computer system,” just as claimed.

407. In sum, in my opinion, the Maeda-Hamann-AAPA-Hirota-Lu combination discloses and/or suggests the method of claim 3, wherein the file system is encrypted with an encryption key, the encryption key derived from computer system information [*e.g.*, Lu’s serial number of the hardware system] and the encryption key not stored by the computer system since it can be derived using Lu’s teachings when needed.

2. Motivation To Combine Maeda, Hamann, the AAPA, Hirota, and Lu

408. I described the motivation to combine Maeda, Hamann, the AAPA, and Hirota in §XII.B.2 above.

409. POSITAs would have been motivated to combine Lu with the Maeda-Hamann-AAPA-Hirota combination because Lu would add additional security features to the system and allow tying the security to the hardware platform. *Id.* at 1. POSITAs would have understood that the when the SD cards described in the Maeda-AAPA-Hirota combination are virtualized using Hamann’s teachings, binding the virtual SD card to a host computer (*e.g.*, by using a hardware serial number) would ensure that secure data on the virtual SD card cannot be transferred

between computers, as only the host computer would have access to the secure area of the virtual SD card. Indeed, POSITAs would have recognized that tying the secure area to the particular computer would make the backed-up data even more secure than it otherwise would be, and prevent circumvention of copyrights because encrypted data could only be decrypted on the same authorized computer.

410. A POSITA would have understood that changing the source of information where keys are derived from is a simple change that requires little experimentation, can be accomplished without undue burden, and a POSITA would have expected success in making such a small change. Thus, claim 3 is obvious.

3. Claim 4: “The method of claim 2, wherein the file system is encrypted with an encryption key, the encryption key derived from a computer system and stored by the computer system.”

411. The only difference I see between claim 4 and claim 3 is that claim 4 requires that encryption key is stored by the computer system, rather than “not stored by the computer system,” as in claim 3. The Maeda-Hamann-AAPA-Hirota-Lu combination discloses this claim. As I discussed in §XII.C.1 above, the Maeda-Hamann-AAPA-Hirota-Lu combination discloses that “the file system is encrypted with an encryption key, the encryption key derived from a computer system.”

412. Hirota discloses that its master key 323a can be stored “by the computer system,” just as claimed. EX1008, Hirota, 16:1-2 (“... master key 323a may be

embedded in ... the PC 102 beforehand.”). When Hirota’s SD card is virtualized using Hamann’s teachings, POSITAs would have understood that the encryption key can be stored by the computer system. As such, only the computer used to create the virtual SD card would have the master key and therefore, access to the authentication area.

413. Similarly, Hamann discloses that encryption keys can be stored in a “secret key area 16” of its virtual smart card, which a POSITA would understand is necessarily stored by the computer system used to create the virtual smart card. *See* EX1007, Hamann, ¶[0028] and Fig. 1. Thus, the Maeda-Hamann-AAPA-Hirota-Lu combination also discloses this claim.

414. In sum, in my opinion, the Maeda-Hamann-AAPA-Hirota-Lu combination discloses and/or suggests the method of claim 4, wherein the file system is encrypted with an encryption key, the encryption key derived from a computer system and stored by the computer system [*e.g.*, Hamann storing encryption keys in secret key area 16].

4. Claim 5: “The method of claim 2, wherein the file system is encrypted with an encryption key, the encryption key derived from a computer system and stored by the computer system within a trusted host.”

415. As an initial matter, the ’749 patent does not describe what it means by a “trusted host.” A sole sentence states that, “[i]n another embodiment, the

encryption key may be stored in a trusted host further enhancing security protection against data access if storage is transferred.” EX1001, 7:9-12. Absent details, a POSITA would have understood that, of course, any computer that stores encryption keys should be a trusted one; otherwise, if encryption keys are not safely guarded, then the entire security scheme may be compromised (*e.g.*, what’s the point of storing one’s valuables in a bank safe if the bank cannot be trusted). *See also* §VII.D above.

416. I note that the only difference between claims 5 and 4 is that claim 5 requires that the claimed “encryption key” be stored “within a trusted host” of the computer system. The Maeda, Hamann, AAPA, Hirota, and Lu combination discloses this claim. Hamann discloses that its computer system uses a VSC control program and further discloses that its computer system uses logic and cryptographic routines for authentication. EX1007, Hamann, ¶[0040]. POSITAs would have understood that this means that the computer is “authenticated” and therefore, a “trusted host.” As such, any key stored “by the computer system” would necessarily mean “within a trusted host.” To follow my bank safe example above, banks require strong authentication (*e.g.*, government issued IDs as well as a matching safe key) before they would allow anyone access to a safe in the bank; with such authentication, the bank and its safes are (more) trusted.

417. Moreover, as discussed in §XII.C.3 above, Hirota also discloses that encryption keys can be stored “by the computer system.” Hirota teaches that only authenticated devices can get access to the authentication area since master key 323a is used by Hirota’s encryption/decryption circuit 327 to control access to the authentication area. EX1008, Hirota, 11:58-12:4.

418. When Hirota’s SD card is virtualized using Hamann’s teachings, and the encryption key is derived from the computer system using Lu’s teachings (as discussed in §XII.C.3 above), then the virtual SD card is bound to the computer (to prevent other computers from accessing the SD card). POSITAs would have understood that if master key 323a is stored on the computer system, as Hirota teaches is an option, it must be stored in a trusted host manner (*e.g.*, an authenticated device). This way, only the authenticated computer (the one used to create the virtual SD card) would have access to the master key and therefore, access to the authentication area, while other computers would not be able to access the authentication area of the virtual SD card. *Id.*

419. In sum, in my opinion, the Maeda-Hamann-AAPA-Hirota-Lu combination discloses and/or suggests the method of claim 5, wherein the file system is encrypted with an encryption key, the encryption key derived from a computer system and stored by the computer system [*e.g.*, Hamann storing encryption keys in

secret key area 16] within a trusted host [Hamann's computer system being authenticated as in Hirota].

5. **Claim 17: "The method of claim 14, wherein data received from an application program through Authenticated Key Encryption (AKE) is encrypted with another key before storing the data in the secure area of the provisioned virtual SD card."**

420. In my opinion, Maeda-Hamann-AAPA-Hirota-Lu discloses/suggests this claim. The '749 patent discloses that when writing data to the virtual SD card's secure area, its "process decrypts 555 data with the key derived during AKE, reads from storage, decrypts the with encryption key, updates data, encrypts with the encryption key, and writes data to the virtual secure area file system 315 of the virtual SD card 300." '749 patent (EX1001), 8:22-27.

421. This can also be seen in the portion of Fig. 5 of the '749 patent reproduced below:

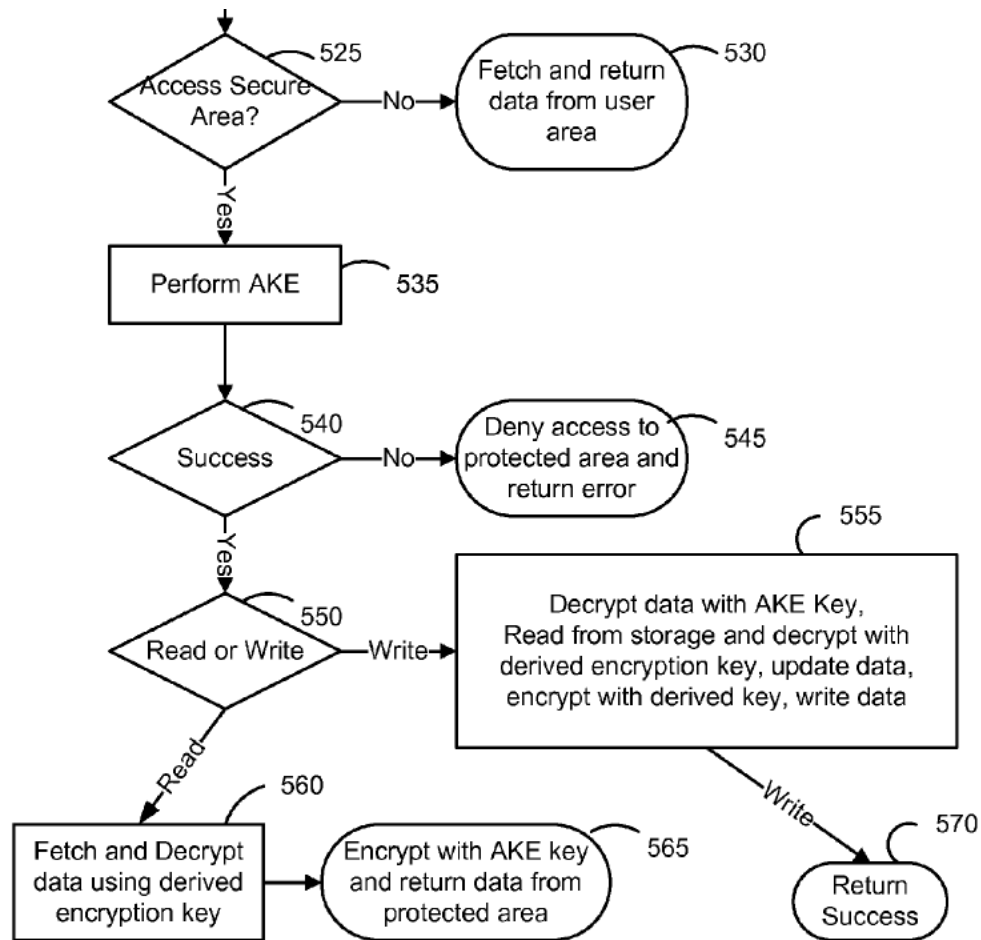


FIG. 5

422. As can be seen, when accessing the virtual SD card's secure area 315 (step 525), AKE is performed (step 535). If the AKE key exchange is successful (step 540), and data is to be written to the virtual secure area 315, the process advances to step 555. As seen is step 555, the data to be written is decrypted with the AKE key, and after updating data, the data is then encrypted using a key derived from computer system information. Finally, the data is written to the virtual SD card's secure area 315.

423. In my opinion, POSITAs would have understood that decrypting data “with the key derived during AKE” is recited in claim 16, and is rendered obvious by the Maeda-Hamann-AAPA combination, as discussed in §XII.A.5 above. In my opinion, POSITAs would have understood that claim 17 is directed to the second part of this process, “encrypt[ing] with the encryption key, and writ[ing] data to the virtual secure area file system”

424. For this process, the ’749 patent discloses “that there are two encryption keys.” ’749 patent (EX1001), 8:27-28. The first key “is for physical storage,” while “[t]he second key is calculated during AKE.” *Id.*, 8:28-29. Because claim 17 recites that “data received from an application program through Authenticated Key Encryption (AKE) is encrypted with another key before storing the data in the secure area of the provisioned virtual SD card,” POSITAs would have understood that it is directed to the first key.

425. POSITAs would have understood that the ’749 patent discloses that one possible “encryption key for storage” is an encryption key that is derived from unique information regarding the computer on which the virtual SD card is stored. In particular, the ’749 patent discloses that secure area file system 315 is encrypted using an encryption key “derived from the computing device’s unique information....” ’749 patent (EX1001), 6:64-7:6. Indeed, the ’749 patent discloses that during a read operation, “the process fetches and decrypts 560 data from the

virtual secure area file system 315 using the key derived from the computing device unique information.” *Id.*, 8:32-35. POSITAs would have understood that if data read from the secure area is decrypted using the derived key, it would have been encrypted using that same derived key. *See also* §VII.D above.

426. As further discussed in §XII.C.1 above regarding claim 3, the Maeda-Hamann-AAPA-Hirota-Lu combination discloses/suggests deriving an encryption key for encrypting data stored in the claimed “file system for secure data (*e.g.*, Hamann’s private area/Hirota’s authentication area).” Thus, claim 17 is rendered obvious for the same reasons as discussed regarding claim 3.

427. In sum, in my opinion, the Maeda-Hamann-AAPA-Hirota-Lu combination discloses and/or suggests the method of claim 17, wherein data received from an application program through Authenticated Key Encryption (AKE) is encrypted with another key [*e.g.*, Lu’s serial number of the hardware system] before storing the data in the secure area of the provisioned virtual SD card [*e.g.*, Hirota’s authentication area/Hamann’s private area].

6. **Claim 20: “The computer readable medium of claim 19, wherein an encryption key is derived from a computer system information, the encryption key not stored by the computer system.”**

428. In my opinion, the Maeda-Hamann-AAPA-Hirota-Lu combination discloses/suggests this limitation for the same reasons as I discuss for claim 3. *See* §XII.C.1 above.

7. **Claim 21: “The computer readable medium of any one of claims 18, 19, or 20, wherein the encryption key is derived from a computer system and stored by the computer system.”**

429. In my opinion, the Maeda-Hamann-AAPA-Hirota-Lu combination discloses/suggests this limitation for the same reasons as I discuss for claim 4. *See* §XII.C.3 above.

430. I note that claim 21 is a multiple dependent claim, depending from claims 18, 19, or 20. I note that certain of these dependencies have ambiguities. For example, claim 21 recites “the encryption key,” but claim 18 contains no such recitation, making it unclear what the encryption key is. Additionally, while claim 21 states that the encryption key is “stored by the computer system,” claim 20 recites that the encryption key is “not stored by the computer system.” Thus, claim 21’s dependency on claim 20 is nonsensical. Regardless, claim 21, when dependent on claim 19, is rendered obvious by the Maeda-Hamann-AAPA-Hirota-Lu combination.

8. **Claim 25: “The computer readable medium of claim 23, wherein data received from an application program through Authenticated Key Encryption (AKE) is encrypted with another key before storing the data in the secure area of the provisioned virtual SD card.”**

431. *See* §XII.C.3 above.

D. Ground 4: Maeda In View Hamann, the AAPA And Matsukawa Renders Claim 13 Obvious

1. **Claim 13: “The method of any one of claims 1 through 5, wherein the media identification comprises random data not allocated from the SD card.”**

432. The Maeda-Hamann-AAPA-Matsukawa combination discloses/suggests this claim. As I discussed in §X.F above, Matsukawa discloses an authentication method for use in memory devices, including SD cards. EX1010, Matsukawa, ¶¶[0003], [0007].

433. Matsukawa states that SD cards and host computers “generate the same session key for each authentication process between the host computer and the recording medium (SD card).” *Id.*, ¶[0007]. To do so, Matsukawa discloses a “media ID generating module 130 [having] the function of generating the media ID in such a manner as not to duplicate a media ID generated in the past” by, for example, “collating with past media IDs after random number generation.” *Id.*, ¶[0051].

434. Matsukawa's media ID generating module 130 is located in Matsukawa's key generation center unit 100, which is separate from recording medium 300. *Id.*, ¶[0039]. Matsukawa states that an SD card is "an example of the recording medium." *Id.*, ¶[0007]. Thus, POSITAs would have recognized that Matsukawa discloses that a media identification is "not allocated from the SD card," and therefore, the Maeda-Hamann-AAPA- Matsukawa combination renders claim 13 obvious.

435. In sum, in my opinion, the Maeda-Hamann-AAPA-Matsukawa combination discloses and/or suggests the method of any one of claims 1 through 5, wherein the media identification comprises random data [*e.g.*, Matsukawa using a random number generator] not allocated from the SD card.

2. Motivation To Combine Maeda, Hamann, the AAPA, and Matsukawa

436. POSITAs would have been motivated to combine Matsukawa with the Maeda-Hamann-AAPA combination. First, POSITAs would have recognized that Matsukawa, like Maeda and the AAPA, is directed to SD cards. *See* EX1010, Matsukawa, ¶[0007].

437. POSITAs would have recognized that using a random media identification not allocated from an SD card, by adding Matsukawa's random media identification using a media ID generating module 130, would require nothing more

than combining known techniques for enhancing security in a predictable manner, with a reasonable expectation of success. Moreover, POSITAs would have recognized that a media identification that includes random data would make the virtual SD card more secure than it otherwise would be.

438. Indeed, using random number generators is a long-established technique in cryptography, because random numbers' high entropy makes them less susceptible to guessing. Conversely, if, say, a vendor assigned sequential ID numbers to each device, it would be much easier to guess them.

439. Thus, claim 13 is obvious.

XIII. SECONDARY CONSIDERATIONS

440. I have seen no evidence in either the '749 patent's prosecution history or elsewhere supporting any secondary considerations arguments, or evidence of nexus of such alleged evidence to the challenged claims. To the extent patent owner or its expert asserts the existence of any secondary considerations in its responses, I reserve the right to address any such evidence.

XIV. CONCLUSION

441. The findings and opinions set forth in this declaration are based on my work and examinations to date.

442. I may continue my examinations. I may also receive additional documentation and other factual evidence over the course of this IPR that will allow

me to supplement and/or refine my opinions. I reserve the right to add to, alter, or delete my opinions and my declaration upon discovery of any additional information. I reserve the right to make such changes as may be deemed necessary.

443. In signing this declaration, I recognize that the declaration will be filed as evidence in an IPR before the PTAB. 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.

444. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on 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 are punishable by fine or imprisonment, or both, under §1001 of Title 18 of the United States Code.

Date: November 1, 2024

A handwritten signature in black ink, appearing to read "Erez Zadok", is positioned above a horizontal line.

Erez Zadok, Ph.D.