EXHIBIT U

(19) Deutsches Patent- und Markenamt

(10) **DE 10 2018 001 565 A1** 2019.06.06

(12) # Offenlegungsschrift

(21) Aktenzeichen: **10 2018 001 565.4**
(22) Anmeldetag: **28.02.2018**
(43) Offenlegungstag: **06.06.2019**

(51) Int Cl.: ***G06F 21/77*** (2013.01)

(71) Anmelder:
Giesecke+Devrient Mobile Security GmbH, 81677 München, DE

(72) Erfinder:
Hartel, Karl Eglof, 80689 München, DE

(56) Ermittelter Stand der Technik:

US	2005 / 0 289 646	A1
US	2006 / 0 236 127	A1
US	2010 / 0 327 059	A1
EP	2 839 602	B1

Active. In: Microsoft Docs, 2. Mai 2017. URL: https://docs.microsoft.com/en-us/windows-hardware/customize/desktop/unattend/microsoft-windows-setup-diskconfiguration-disk-modifypartitions-modifypartition-active [abgerufen am 2. Januar 2019]

Host card emulation. In: Wikipedia, The Free Encyclopedia. Bearbeitungsstand: 9. September 2016. URL: https://en.wikipedia.org/w/index.php?title=Host_card_emulation&oldid=738567522 [abgerufen am 2. Januar 2019]

Hypervisor. In: Wikipedia, The Free Encyclopedia. Bearbeitungsstand: 25. Februar 2018. URL: https://en.wikipedia.org/w/index.php?title=Hypervisor&oldid=827645190 [abgerufen am 2. Januar 2019]

ISO/IEC 7816-3, Third Edition, 2006. URL: http://read.pudn.com/downloads132/doc/comm/563504/ISO-IEC%207816/ISO%2BIEC%207816-3-2006.pdf [abgerufen am 2. Januar 2019]

Partitionierung. In: WinBoard Wiki. Bearbeitungsstand: 27. Dezember 2010. URL: http://wiki.winboard.org/index.php?title=Partitionierung&oldid=19063 [abgerufen am 2. Januar 2019]

Virtualisierung (Informatik). In: Wikipedia, Die freie Enzyklopädie. Bearbeitungsstand: 30. Dezember 2017. URL: https://de.wikipedia.org/w/index.php?title=Virtualisierung_(Informatik)&oldid=172417281 [abgerufen am 2. Januar 2019]

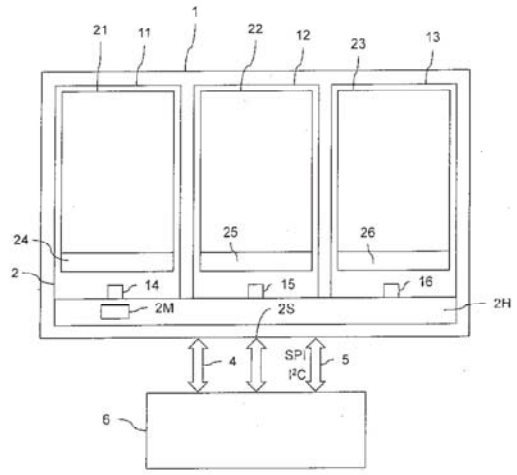
Mit Einverständnis des Anmelders offengelegte Anmeldung gemäß § 31 Abs. 2 Ziffer 1 PatG

Prüfungsantrag gemäß § 44 PatG ist gestellt.

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

(54) Bezeichnung: **Sicherheitselement und Verfahren zur Zugriffskontrolle auf ein Sicherheitselement**

(57) Zusammenfassung: Die Erfindung betrifft ein Sicherheitselement (1) mit einem Speicher (2), der zumindest eine Partition (11, 12, 13) umfasst, und mit einem Speicher-Controller (2H), der dazu eingerichtet ist, die zumindest eine Partition (11, 12, 13) zu steuern und in einer jeweiligen der zumindest einen Partition (11, 12, 13) eine Chipkarte (21, 22, 23) zu virtualisieren.



Beschreibung

[0001] Die Erfindung betrifft ein Sicherheitselement und ein Verfahren zur Zugriffskontrolle auf ein Sicherheitselement.

[0002] Gegenwärtig existiert eine Vielzahl an unterschiedlichen Chipkarten und Sicherheitselementen, die von unterschiedlichen Herausgebern ausgegeben und mit unterschiedlichen Betriebssystemen betrieben werden. In mobilen Endgeräten (z.B. Smartphones, Tablet PCs, SmartWatches etc.) werden vielfach eingebettete Sicherheitselemente (sog. embedded Secure Elements, SEs) verbaut. Bei solchen mobilen Endgeräten kann parallel zu den eingebetteten Sicherheitselementen darüber hinaus eine andere Chipkarte, z.B. eine µSD-Karte mit Sicherheitselement vorgesehen sein.

[0003] Um eine größere Anzahl an unterschiedlichen Sicherheitselementen oder Chipkarten zu vermeiden, besteht der Wunsch, diese auf einer gemeinsamen Hardware zu integrieren. Dabei besteht die Herausforderung darin, die für die unterschiedlichen Sicherheitselemente und Chipkarten entwickelten Betriebssysteme und Applikationen mit einem gemeinsamen Betriebssystem nutzen zu können. Eine Konsolidierung auf Basis eines gemeinsamen Betriebssystems erfordert jedoch einen großen Entwicklungsaufwand und ist im Hinblick auf unterschiedliche Herausgeber unflexibel.

[0004] Es ist Aufgabe der Erfindung, ein Sicherheitselement sowie ein Verfahren zur Zugriffskontrolle auf ein Sicherheitselement anzugeben, das die oben beschriebenen Nachteile nicht aufweist.

[0005] Diese Aufgabe wird gelöst durch ein Sicherheitselement gemäß den Merkmalen des Anspruchs 1 und ein Verfahren gemäß den Merkmalen des Anspruchs 14. Vorteilhafte Ausgestaltungen ergeben sich aus den abhängigen Ansprüchen.

[0006] Zur Lösung der Aufgabe wird ein Sicherheitselement vorgeschlagen, das einen Speicher und einen Speicher-Controller umfasst. Der Speicher umfasst zumindest eine Partition. Der Speicher-Controller ist dazu eingerichtet, die zumindest eine Partition zu steuern, und in einer jeweiligen der zumindest einen Partition eine Chipkarte zu virtualisieren.

[0007] Bei einem derartigen Sicherheitselement ist es möglich, jede virtualisierte Chipkarte mit ihrem eigenen Betriebssystem zu versehen. Damit kann die Differenzierung, die gegenwärtig zwischen unterschiedlichen Herausgebern und unterschiedlichen Funktionen, wie z.B. Telekommunikation, Zahlungsverkehr und Transit, existiert, beibehalten werden.

[0008] Insbesondere ermöglicht es ein derartiges Sicherheitselement auch, kundenspezifische Erweiterungen eines für eine jeweilige Chipkarte verwendeten Betriebssystems sowie ggf. erwünschte Sonderlösungen zu nutzen. Die Verwendung eines erfindungsgemäßen Sicherheitselements ermöglicht es jedem Herausgeber, eine virtuelle Chipkarte für seine bisher bekannte, individuelle Umgebung für Applikationen zur Verfügung zu stellen.

[0009] Darüber hinaus wird eine Zertifizierung vereinfacht. Aufgrund einer klaren Abgrenzung zwischen unterschiedlichen virtualisierten Chipkarten ist lediglich der Speicher-Controller zusätzlich zu zertifizieren. Die Zertifizierung der Chipkarte(n) entspricht dem bislang verwendeten Verfahren und kann ggf. von einer nicht-virtualisierten Chipkarte übernommen werden. Chipkarten, die auf anderen Partitionen genutzt werden, sind von der Zertifizierung einer jeweiligen Chipkarte nicht betroffen.

[0010] Der Speicher-Controller ist zweckmäßigerweise ein Hypervisor, insbesondere ein Typ-1-Hypervisor. Hypervisor werden auch Virtual-Machine-Monitore genannt und bezeichnen eine Klasse von Systemen, die als abstrahierende Schicht zwischen tatsächlich vorhandener Hardware und weiteren zu installierenden Betriebssystemen dienen. Die tatsächlich vorhandene Hardware wird bei der vorliegenden Erfindung durch das Sicherheitselement repräsentiert. Hypervisoren erlauben es, eine virtuelle Umgebung mit Hardwareressourcen (wie CPU, Speicher, verfügbare Peripherie) zu definieren, die unabhängig von der tatsächlich vorhandenen Hardware als Basis für die Installation von (Gast)Betriebssystemen, hier: den virtualisierten Chipkarten, dient.

[0011] Hypervisoren erlauben den simultanen Betrieb mehrerer Gastsysteme auf dem Sicherheitselement, das das Hostsystem darstellt. Der Hypervisor verwaltet die Ressourcenzuteilung für die einzelnen Gastsysteme, d.h. die auf jeweiligen Partitionen virtualisierten Chipkarten. Der Hypervisor verteilt die Hardware-Ressourcen derart, dass für jedes einzelne Gast-Betriebssystem alle Ressourcen bei Bedarf verfügbar sind, so, als ob ein Betriebssystem vorhanden wäre. Dies kann durch Hardware-Emulation, Hardware-Virtualisierung oder Paravirtualisierung erfolgen. Den einzelnen Gastsystemen wird dabei jeweils ein eigener kompletter Rechner mit allen Hardwareelementen (Prozessor, Arbeitsspeicher und dergleichen) vorgespielt.

[0012] Die tatsächlich vorhandene Hardwareumgebung des Sicherheitselements wird als Hostsystem bezeichnet.

[0013] Ein Typ-1-Hypervisor (native oder bare metal) setzt direkt auf der Hardware des Sicherheitselements auf und benötigt keine vorherige Betriebssys-

tem-Installation. Voraussetzung für dieses Vorgehen ist, dass die Hardware des Sicherheitselements vom Typ-1-Hypervisor durch entsprechende Treiber unterstützt wird.

[0014] Eine weitere zweckmäßige Ausgestaltung sieht vor, dass das Sicherheitselement eine oder mehrere Schnittstellen für eine Kommunikation mit einem externen Gerät umfasst, wobei die eine oder mehreren Schnittstellen durch den Speicher-Controller verwaltet werden. Diese Ausgestaltung berücksichtigt den Umstand, dass das Sicherheitselement - unabhängig von der Anzahl virtualisierter Chipkarten - typischerweise nur eine Schnittstelle aufweist, die dahingehend modifiziert wird, dass die in den Partitionen virtualisierten Chipkarten einzeln angesprochen werden können. Dabei kann eine oder mehrere der nachfolgend beschriebenen Varianten zum Einsatz kommen.

[0015] Gemäß einer ersten Variante ist der Speicher-Controller zur Verwaltung der einen oder mehreren Schnittstellen dazu eingerichtet, dass eine Adressinformation, z.B. ein Node Address Byte (NAD), einer Nachricht in einem von zumindest einer der Chipkarten verwendeten Protokoll auszuwerten, um in Abhängigkeit des Inhalts der Adressinformation (z.B. des Node Address Bytes) eine zugeordnete Chipkarte in eine der Chipkarte zugeordneten Partition anzusprechen. Beispielsweise definiert das Chipkartenprotokoll T = 1 das Node Address Byte, das derzeit nicht verwendet wird. Die Verwendung dieses Node Address Bytes erlaubt es, die verschiedenen virtualisierten Chipkarten mit unterschiedlichen Ziel-Adressen (Destination Node Address) anzusprechen.

[0016] In einer zweiten Variante umfassen die eine oder mehreren Schnittstellen eine USB-Schnittstelle (USB = Universal Serial Bus), wobei der Speicher-Controller dazu eingerichtet ist, einen USB-Hub zu simulieren, an den eine jeweilige Chipkarte in einer der Chipkarte zugeordneten Partition als USB-Smartcard Device angeschlossen ist. Mit anderen Worten simuliert der Speicher-Controller einen USB-Hub, an den die einzelnen virtualisierten Chipkarten als eigene Geräte angeschlossen sind, die jeweils ihrerseits die USB-Smartcard Device Class implementieren.

[0017] Eine dritte Variante sieht vor, dass die eine oder mehreren Schnittstellen eine proprietäre Schnittstelle, insbesondere I2C oder SPC, umfassen, wobei der Speicher-Controller dazu eingerichtet ist, aus einem empfangenen Kommando-Datenblock (z.B. einer APDU) eine Adresse einer der Chipkarten auszulesen und den Kommando-/Datenblock an die adressierte Chipkarte in der der Chipkarte zugeordneten Partition weiterzuleiten. Als die Schnittstelle kann dabei jede proprietäre Schnittstelle, welche bislang auch nicht für Chipkarten zum Einsatz kommen braucht, implementiert werden.

[0018] Gemäß einer zweckmäßigen Ausgestaltung ist der Speicher-Controller dazu eingerichtet, eine von dem Sicherheitselement von einem externen Gerät empfangene Nachricht einer bestimmten Partition der zumindest eine Partition zuzuordnen und die Nachricht und an die in der bestimmten Partition enthaltene Chipkarte weiterzuleiten.

[0019] Es ist weiterhin zweckmäßig, wenn der Speicher-Controller dazu eingerichtet ist, der zumindest einen Partition eine jeweils zugeordnete Schnittstelle bereitzustellen, die der Schnittstelle der in der betreffenden Partition virtualisierten Chipkarte entspricht. Dadurch wird auf einfache Weise eine Adressierung der unterschiedlichen Chipkarten in dem Sicherheitselement ermöglicht.

[0020] Eine weitere zweckmäßige Ausgestaltung sieht vor, dass die Partitionen in dem Speicher so voneinander getrennt sind, dass eine der Chipkarten auf einer der Partitionen keine Aktionen auf einer der anderen/ übrigen Partitionen ausführen kann. Diese Abschottung kann beispielsweise mit Hilfe einer Speicherverwaltungseinheit (Memory Managing Unit (MMU)) realisiert werden, welche Bestandteil des Speicher-Controllers sein kann oder als separate Komponente auf dem Sicherheitselement vorgesehen sein kann.

[0021] Eine weitere Ausgestaltung sieht vor, dass in einer jeweiligen Partition ein vollständiges Betriebssystem mit einem Chipkartenprofil geladen oder gespeichert ist. Unter einem Profil wird insbesondere eine Sammlung von Attributen verstanden, die Betriebsweisen bzw. Operationen spezifizieren, welche ein Nutzer auf der betreffenden Partition ausführen darf. Dies betrifft insbesondere Leserechte, Schreibrechte, die Ausführung von Applikationen oder Programmen und dergleichen.

[0022] Zweckmäßigerweise ist der Speicher-Controller dazu eingerichtet, für jede der zumindest einen Partition eine Tabelle zu verwalten, welche Statusinformationen über die jeweilige Partition umfasst. Dabei übernimmt der Speicher-Controller die Aufgabe eines Partitions-Managers, welcher sicherstellt, dass zu einem gegebenen Zeitpunkt immer nur eine Partition und damit ein Profil aktiv ist.

[0023] Es ist zweckmäßig, wenn der Speicher-Controller dazu eingerichtet ist, beim Empfang eines Umschaltbefehls eines Nutzers oder eines Subskriptions-Managers eine Umschaltung von einer gerade aktiven Partition auf eine andere, nicht aktive Partition durchzuführen. Eine derartige Ausgestaltung des Speicher-Controllers ermöglicht es, eine Umschaltung zwischen verschiedenen Partitionen nutzerab-

hängig oder in Abhängigkeit des Erhalts eines Befehls eines Subskriptions-Managers durchzuführen. Auf diese Weise kann mit Hilfe des Speicher-Controllers sichergestellt werden, dass zu einem gegebenen Zeitpunkt niemals mehr als eine Partition und damit ein Profil aktiv ist.

[0024] Die Erfindung schafft weiter ein Verfahren zur Zugriffskontrolle auf ein Sicherheitselement, welches gemäß der oben beschriebenen Art ausgebildet ist. Das Verfahren umfasst die Schritte des Bereitstellens eines Speicher-Controllers, um den Zugriff auf zumindest eine Partition eines Speichers des Sicherheitselements zu steuern und den Schritt des Virtualisierens einer Chipkarte in einer jeweiligen der zumindest einen Partition.

[0025] Das erfindungsgemäße Verfahren weist die gleichen Vorteile auf, wie sie vorstehend in Verbindung mit einem Sicherheitselement beschrieben wurden.

[0026] Die Erfindung wird nachfolgend näher anhand eines Ausführungsbeispiels in der Zeichnung erläutert. Es zeigen:

Fig. 1 eine schematische Darstellung eines erfindungsgemäßen Sicherheitselements; und

Fig. 2 das erfindungsgemäße Sicherheitselement mit einer Mehrzahl von Partitionen, in denen jeweils Profile und ein Betriebssystem einer Chipkarte illustriert sind.

[0027] **Fig. 1** zeigt eine schematische Darstellung eines erfindungsgemäßen Sicherheitselements **1** gemäß der Erfindung. Das Sicherheitselement **1** umfasst einen Speicher **2**. Der Speicher **2** umfasst im Ausführungsbeispiel drei Partitionen **11**, **12**, **13**. Die Anzahl der Partitionen **11**, **12**, **13** kann in der Praxis größer oder kleiner als drei gewählt sein. Die Partitionen **11**,**12**, **13** werden von einem Speicher-Controller **2H** bereitgestellt und verwaltet. Der Speicher-Controller **2H** ist z.B. ein sog. Hypervisor, der es erlaubt, eine virtuelle Umgebung im Hinblick auf Hardwareressourcen (insbesondere CPU, Speicher und verfügbare Peripherie) zu definieren, die unabhängig von der tatsächlich vorhandenen Hardware als Basis für die Installation von Softwarekomponenten in den Partitionen **11**, **12**, **13** dient.

[0028] Hierdurch erlaubt der Speicher-Controller **2H** den Betrieb mehrerer in den Partitionen **11**, **12**, **13** virtualisierter Chipkarten **21**, **22**, **23**. Bei den virtualisierten Chipkarten **21**, **22**, **23** kann es sich beispielsweise um verschiedene Chipkarten, z.B. eine für Telekommunikation, eine für Zahlungsverkehr und eine für Transit etc., handeln. Jede der virtualisierten Chipkarten **21**, **22**, **23** verfügt hierbei über ein spezifisches Betriebssystem **24**, **25**, **26** zur Ausführung jeweils spezifischer Funktionen und Applikationen. Die Steuerung der Partitionen und der darauf installierten, virtualisierten Chipkarten **21**, **22**, **23** erfolgt mittels des Speicher-Controllers **2H**.

[0029] Der Speicher-Controller **2H** stellt den einzelnen Partitionen **21**, **22**, **23** eine Schnittstelle **14**, **15**, **16** bereit, welche sich nicht von den jeweiligen Schnittstellen der betreffende Chipkarte **21**, **22**, **23** unterscheidet. Die Partitionen sind z.B. über eine Speicherverwaltungseinheit **2M** (Memory Management Unit) so gegeneinander abgeschottet, dass keine Chipkarte **21**, **22**, **23** (d.h. keine von der Chipkarte ausgeführte Applikation) die Existenz einer anderen Chipkarte auf den anderen Partitionen realisiert und darüber hinaus auch keinerlei Aktivität außerhalb seiner Partition **11**, **12**, **13** ausführen kann.

[0030] Der Speicher-Controller stellt darüber hinaus gemeinsame Funktionen für die virtualisierten Betriebssysteme **24**, **25**, **26** der Chipkarten **21**, **22**, **23** zur Verfügung, wie z.B. hardwarenahe, angriffsresistente Kryptographie- bzw. Verschlüsselungsroutinen.

[0031] Das Sicherheitselement **1** weist eine oder mehrere Schnittstellen **2S** für eine Kommunikation mit einem externen Gerät **6** auf. Die eine oder die mehreren Schnittstellen **2S** werden hierbei durch den Speicher-Controller **2H** verwaltet. Unabhängig von der Anzahl der vorgesehenen Schnittstellen **2S** ist es erforderlich, eine jeweilige der Schnittstellen derart zu erweitern, dass die virtualisierten Chipkarten **21**, **22**, **23** in ihren jeweils zugeordneten Partitionen **11**, **12**, **13** einzeln von einem externen Gerät **6** adressiert werden können.

[0032] Hierzu sind verschiedene Varianten möglich: Beispielsweise kann in dem bekannten Chipkartenprotokoll T = 1 ein sog. Node Address Byte (NAD) verwendet werden, das derzeit im Chipkartenprotokoll ungenutzt ist (Kommunikation **3**). Die Verwendung des Node Address Byte erlaubt es, die verschiedenen, virtualisierten Chipkarten **21**, **22**, **23** mit unterschiedlichen Adressen zu adressieren. Die Adressen werden als Destination Node Address bezeichnet.

[0033] Alternativ kann der Speicher-Controller **2H**, sofern die Schnittstelle(n) **2S** als USB-Schnittstelle ausgebildet ist, einen USB-Hub simulieren (Kommunikation **4**). Dabei ist die Simulation derart, dass an dem USB-Hub die einzelnen virtualisierten Chipkarten **21**, **22**,**23** als eigene Geräte angeschlossen erscheinen. Diese implementieren ihrerseits die USB Smartcard Device Class, wodurch die einzelnen Chipkarten **21**, **22**, **23** eindeutig adressierbar sind.

[0034] Darüber hinaus besteht die Möglichkeit, dass das Sicherheitselement **1** eine eigene Schnittstelle, die bislang nicht für Chipkarten verwendet wird, implementiert (Kommunikation **5**). Eine solche proprietäre Schnittstelle könnte beispielsweise eine **SPI** oder

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.