PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G07F 7/08 — A1

(11) International Publication Number: **WO 00/08610**

(43) International Publication Date: 17 February 2000 (17.02.00)

(21) International Application Number: PCT/US99/17503

(22) International Filing Date: 2 August 1999 (02.08.99)

(30) Priority Data:
09/128,985 3 August 1998 (03.08.98) US

(71) Applicant: MICROSOFT CORPORATION [US/US]; One Microsoft Way, Redmond, WA 98052–6399 (US).

(72) Inventor: WATERS, Lester, L.; 954 Broadway Avenue East, Seattle, WA 98102 (US).

(74) Agents: LEE, Lewis, C. et al.; Suite 500, 421 West Riverside Avenue, Spokane, WA 99201 (US).

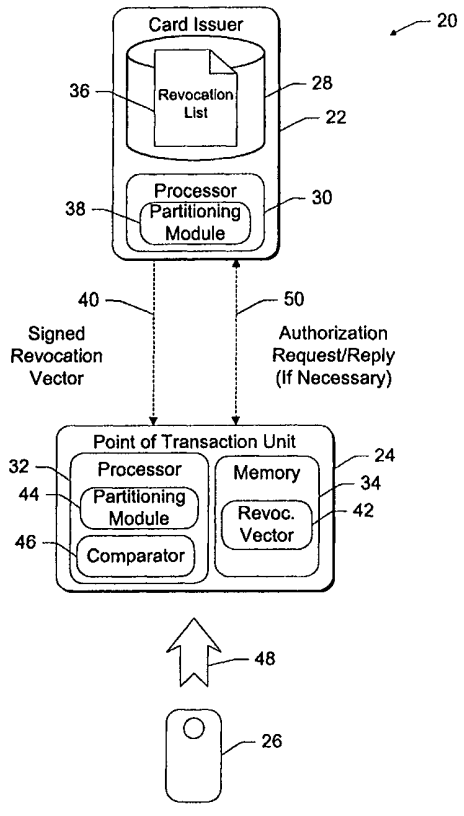
(81) Designated States: CA, JP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(54) Title: OFFLINE VERIFICATION OF INTEGRATED CIRCUIT CARD USING HASHED REVOCATION LIST

(57) Abstract

A system that performs offline verification of integrated circuit (IC) devices (such as smart cards, electronic wallets, PC cards and the like) includes an issuing unit and multiple point–of–transaction units. The issuing unit maintains a database that stores a recovation list of ID codes of bad IC devices that have been revoked. The issuing unit partitions the ID codes on the revocation list into multiple buckets and derives a revocation vector into the buckets. The issuing unit occasionally downloads the recovation vector to multiple point–of–transaction units, such as merchant computers, standalone kiosks, vending machines, and the like. During a transaction, a point–of–transaction unit verifies a tendered IC device in an offline procedure. The point–of–transaction unit takes the ID code of the tendered IC device, partitions it, and compares the result to the revocation vector to determine whether the ID code partitions into a non–empty bucket. If so, the ID code of the tendered IC device *might* be on the revocation list and the point–of–transaction unit initiates an online authentication process to further verify the IC device. Otherwise, if the ID code partitions to an empty bucket, the IC device is not on the revocation list and the transaction may proceed.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

OFFLINE VERIFICATION OF INTEGRATED CIRCUIT CARD USING HASHED REVOCATION LIST

TECHNICAL FIELD

This invention relates to integrated circuit (IC) cards, such as smart cards, and methods for verifying IC cards in offline transactions. This invention may also be extended to verifying other types of IC devices with limited memory and processing capabilities, such as smart diskettes, electronic wallets, PC cards, and the like. More particularly, the invention relates to methods for managing and processing revocation lists for compromised IC devices.

BACKGROUND OF THE INVENTION

Authentication systems are used for security purposes to verify the authenticity of one or more parties during a transaction. Traditionally, authentication systems have been manual, involving personal recognition or quick verification of a party via some form of additional identification. One very familiar authentication process occurs when purchasing an item with a personal check. The sales clerk will process the check only if he/she recognizes the person writing the check or if the person presents another piece of identification (e.g., a credit card or driver's license) to verify their authenticity as the specific person who is tendering the check.

Some authentication systems are electronic. A familiar electronic authentication system is a common credit card purchase. A card issuer issues a credit card to a consumer to enable the consumer to purchase items on credit. Credit cards that are primarily in use today consist of magnetic-stripe memory cards

that have a single magnetic stripe ("mag-stripe") on one side. The magnetic stripe contains information about the card issuer, the consumer, and his/her account.

During a purchase transaction, the consumer presents the credit card to a sales clerk, who authenticates the card before finalizing the transaction. The credit card authentication process is typically performed "online". The sales clerk swipes the card through a reader, which extracts the card data from the magnetic stripe and transmits the data over a network to the card issuer (or a third party contracted to handle authentication requests). The card issuer checks to ensure that the card is still valid (i.e., has not expired), has not been revoked as being lost or stolen, and the corresponding account is below the authorized credit limit. If the authentication is successful, the card issuer returns an approval and the sales clerk completes the transaction. With conventional telecommunications and computerized processes, the entire credit card authentication process is typically handled in an acceptable length of time, such as a few seconds.

Today, there is an increasing use of "smart cards" in place of, or in addition to, mag-stripe cards. A "smart card" is a card with a built-in processor that enables the card to modify, or even create, data in response to external stimuli. The processor is a single-wafer integrated circuit (IC) which is mounted on an otherwise plastic credit card. For this reason, smart cards are often referred to as one class of "integrated circuit (IC) cards".

As smart card technology becomes more pervasive, it paves the way for conducting a variety of new transactions, such as electronic money, which are not available with conventional mag-stripe cards. Smart cards also open up the arena for conducting certain new "offline" transactions, which do not involve validating a card with a central authority. These offline electronic transactions are typically performed without the human intervention, such as from a sales clerk.

As an example, smart cards can be configured as electronic wallets to hold electronic assets such as cash, coins, tokens, entertainment tickets, government entitlement provisions, and so on. "Electronic assets" are digital, typically random, binary strings that represent cash or other value, thereby replacing traditional asset forms (bills, coins, ticket paper, etc.). A major segment of commerce is found at the low end of the value scale. This commerce involves values equivalent to present-day cash, such as paper bills (i.e., $1, $5, $10, $20, $50, and $100 bills) and coins (i.e., nickels, dimes, quarters, half-dollars, and dollars). It is impractical to perform online transaction verification for each one of these low-end purchases. Imagine how difficult it would be to connect each vending machine to a central authority so that every soda pop purchase made with a smart card could be verified. The electronic network traffic alone would most likely overwhelm conventional systems. Online systems are simply too expensive or too slow for this low end of the market. Telecommunication costs and bandwidth are significantly impacted and in some countries, these telecommunication costs are prohibitive.

Unfortunately, fraud is more likely to germinate in offline electronic asset transactions because there is no validity check performed remotely by a central authority. A dishonest individual can present a lost, stolen, or otherwise compromised smart card to a merchant (or vending machine). Because the merchant (or vending machine) is operating offline, there is little opportunity for authenticating the card before committing the transaction.

Another problem is that electronic assets can be easily duplicated. Unlike paper dollars or coins, a string of bits that constitutes the electronic assets can be easily and rapidly replicated using computers. This presents a significant risk of fraud. Criminals can reproduce the bit string of an asset and pass off the forged or counterfeited electronic assets as real. To the recipient, the counterfeit bit string

offered by the criminal is identical to the expected asset bit string, rendering it difficult to detect whether the offered bit string is the original asset or a reproduced asset that has been used many times before. If successful, the criminals have the opportunity to multi-spend the same asset many times. This type of digital fraud is known as "double spending".

One proposed solution to these problems is to devise tamper-"proof" devices, which by their design, make it nearly impossible to modify or clone the devices to perform fraudulent transactions. Unfortunately, such designs are never truly tamper-"proof," rather just tamper-"resistant." In other words, if criminals were willing to invest the necessary capital, albeit large, they could reverse engineer the electronic devices to perform fraudulent tasks. The cost of breaking tamper-resistant devices varies dramatically with the technology and the evolution of technology over time.

While credit card companies would like to eliminate all avenues of fraud, they are willing to tolerate a small percentage in tradeoff to keep overall customer satisfaction high. Specific requirements include the ability to conduct transactions quickly, and to ensure that a legitimate customer card is never falsely rejected (i.e., it is never mistaken as a lost or stolen card). Of course, the credit card companies would like to minimize or eliminate false acceptances (i.e., mistaking a lost/stolen card as legitimate) as well to reduce fraud.

In an effort to meet these goals, a card issuer typically maintains a revocation list of cards that are reported as being lost, stolen, or otherwise compromised (for instance, a card that is reprogrammed to double spend assets). The revocation list is used during online verification to determine whether a card has been revoked. For offline verification, however, use of a revocation list is often unworkable. In the early days of Visa and MasterCard credit cards, the card issuers mailed paper

booklets with lists of revoked cards to the merchants. During a credit card transaction, a sales clerk would leaf through the booklet to determine whether the tendered card was listed. If listed, the card was deemed revoked and the transaction was halted. If the card did not appear on the list, the sales clerk accepted payment using the card. The exposure to the merchant was the time between accepting bad cards and receiving updated revocation lists. To the customer, waiting for a paper check was often inconvenient.

It would therefore be desirable to provide an electronic system that enabled efficient offline verification of cards using computer processing, without resorting to manual look-up booklets.

By virtue of the resident on-chip processor, smart cards are self-validating and can perform offline verification. Ideally, it would be convenient to download the revocation list to the smart card processor and ask it to determine whether the card is still valid and has not been revoked. Unfortunately, the shear size of a revocation list makes this operation prohibitive. For example, a list of just 1,000 bad credit card numbers might take 16KB (around 6-7KB compressed), which consumes a significant portion or exceeds the smart card memory. Unfortunately, in reality, there are far more bad credit cards than this. Furthermore, smart card processors are not all that powerful. With limited processors and memory capacities, it is infeasible to process a full revocation list in the smart card processor.

It would therefore be desirable to provide an electronic process for offline verification of smart cards (and other IC devices, such as electronic wallets, PC cards, smart diskettes, and so on) that enables quick transactions, while minimizing fraud without falsely rejecting a legitimate card.

SUMMARY OF THE INVENTION

This invention concerns a system and method for performing offline verification of integrated circuit (IC) devices, such as smart cards, electronic wallets, PC cards, and the like. The IC device has a processor and a memory to hold an identification (ID) code.

An issuing unit maintains a database that stores a revocation list of ID codes of bad IC devices that have been revoked. The issuing unit partitions the ID codes on the revocation list into multiple partitions or "buckets". The issuing unit might further partition each bucket into multiple sub-lists. The partitioning process is selected to provide a relatively even but somewhat sparse distribution of ID codes across a number of buckets, wherein it is likely that some of the buckets will be empty (i.e., contain no ID codes), while other buckets will be non-empty (i.e., contain at least one ID code).

Once the revocation list is partitioned, the issuing unit derives a revocation vector into the buckets. The revocation vector represents a distribution of the ID codes among the buckets. The revocation vector might be, for example, a bit array with one bit per bucket, wherein the bit is set to one binary value if the bucket is not empty and to the other binary value if the bucket is empty.

The issuing unit occasionally downloads the revocation vector to multiple point-of-transaction units, such as merchant computers, standalone kiosks, vending machines, teller machines, and the like. During a transaction, a point-of-transaction unit verifies a tendered IC device in an offline procedure. The point-of-transaction unit takes the ID code of the tendered IC device, processes it using the same partitioning process, and compares the result to the revocation vector to determine whether the ID code would partition into an empty bucket or a non-empty bucket. If the ID code partitions into a non-empty bucket, the ID code of the tendered IC

device *might* be on the revocation list because this bucket contains at least one ID code of a revoked IC device.

At this point, there are several options for the point-of-transaction unit. One option is to simply deny the transaction, without further processing. If the revocation vector contains sub-vector information to the sub-lists, another option is for the point-of-transaction unit to perform a secondary determination as to whether the ID code of the tendered IC device partitions into a non-empty sub-list. A third option for those units that have access to online verification is to initiate an online verification procedure.

The partitioning process and number of buckets are selected so that it is rare for a device ID code to partition to a non-empty bucket, and more rare for it to partition to a non-empty sub-list. In this manner, the point-of-transaction unit can verify most IC devices without resorting to online verification for each transaction.

It is noted that the same verification process can be performed on the IC device itself. That is, the revocation vector is passed into the IC device during authentication, and the device processor compares it to the partitioned ID code to determine whether the ID code partitions into an empty bucket or a non-empty bucket. The small revocation vector requires substantially less memory and processing capabilities than a full revocation list. Accordingly, the process provides an effective means for reducing telecommunication costs and bandwidth requirements for card verifications.

In another implementation, the IC device may wish to authenticate the point-of-transaction unit. A digitally signed small revocation vector is passed to the IC device along with the credentials for the point-of-transaction unit (such as a digitally signed certificate containing the ID code of the unit). The IC device, in turn, authenticates the digital signature on the revocation vector as well as the

credentials presented. If the ID code of the point-of-transaction unit hashes to a non-empty bucket, the IC device may request the unit to engage in an online verification sequence. In this verification sequence, the point-of-transaction unit would obtain a validation from the certification authority. The validation would be digitally signed and contain (at minimum) the ID code of the point-of-transaction unit and challenge data from the IC device.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of an electronic offline verification system.

Fig. 2 is a block diagram of an IC card.

Fig. 3 illustrates a process for converting a revocation list of revoked ID codes into a condensed revocation vector indicative of the revocation list.

Fig. 4 is a flow diagram showing steps in a method for creating a revocation vector.

Fig. 5 is a flow diagram showing steps in a method for offline verification of an IC card.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Fig. 1 generally shows a verification system 20 having an issuing unit 22, a point-of-transaction unit 24, and an integrated circuit (IC) device 26. In this most basic model, the issuing unit 22 represents both the entity that issued the IC device 26 (e.g., credit card company, bank, etc.), as well as the entity that performs online verifications of the devices, although these tasks can be separated among two different entities. The issuing unit 22 represents a computer system having a database 28 and a processing unit 30. The computer system might be implemented

as a PC-based server (such as one configured to run an NT operating system from Microsoft Corporation), a workstation, a minicomputer, or a mainframe computer.

The point-of-transaction unit 24 is representative of computerized devices that are distributed locally for interaction with consumers. The point-of-transaction unit 24 may take the form of a general-purpose computer, an ATM (automated teller machine), a kiosk, a vending machine, and the like. Regardless of the form, the point-of-transaction unit 24 has basic electronic computing components including a processing unit 32 and a memory 34.

The IC device 26 is illustrated as a smart card. In addition, the IC device might be embodied as an electronic wallet, personal digital assistant, smart diskette (i.e., an IC-based device having a form factor and memory drive interface to enable insertion into a floppy disk drive), smart card, PC card (formerly PCMCIA card), and the like. Generally, the IC device 26 is characterized as an electronic device with limited processing capabilities and memory wherein large size number crunching, such as processing an entire revocation list, would prove impractical. However, aspects of this invention may be utilized with IC devices that do not meet this limitation, as well as to verification of non-computerized items, such as conventional credit cards. For purposes of continuing discussion and within the context of the illustrated implementation, the terms "IC device" and "smart card" will be used interchangeably to reference the smart card 26.

The issuing unit 22 stores a revocation list 36 on the database 28. The revocation list contains all of the identification (ID) codes that are associated with smart cards that have been revoked. The ID codes can take on one of many different types. For instance, the ID code might be a specific card number that is assigned to the card, or an account number that identifies the holder's account, or other information that is unique to the IC device. When a smart card is detected as

being lost, stolen, or otherwise compromised in any manner, the issuing unit 22 places the card's ID code onto the revocation list.

Typically, smart cards (or the assets stored thereon) are designed to expire after a preset time interval. Overtime all smart cards and assets stored thereon will time out, and must be re-validated with the issuing unit. Accordingly, the revocation list 36 only carries the ID codes for revoked, but still valid, cards which have not yet expired.

General Authorization Process

The issuing unit 22 has a partitioning module 38 to partition the ID codes on the revocation list 36 into multiple buckets. The partitioning process is selected to provide a relatively even but somewhat sparse distribution of ID codes across a number of buckets, wherein it is likely that some of the buckets will be empty (i.e., contain no card ID codes), while other buckets will be non-empty (i.e., contain at least one card ID code). The partitioning process and number of buckets are selected so that many or most of the buckets are empty. If desired, the partitioning module 38 might further partition individual buckets into sub-lists, if a more finite breakdown of revoked ID codes is warranted.

The partitioning module 38 derives a revocation vector that represents the distribution of ID codes within the buckets. As one example, the revocation vector might be a bit array of binary "1"s and "0"s, where each bit indicates whether a corresponding bucket is empty or non-empty. One particular technique for deriving a revocation vector is described below with reference to Figs. 3-5. If the partitioning process includes sub-lists, the revocation vector also includes sub-vectors to individual sub-lists.

The issuing unit 22 digitally signs the revocation vector with a signature of the issuer. The issuing unit 22 occasionally supplies the signed revocation vector to the point-of-transaction unit 24 via a supply path 40. The frequency of delivering the revocation vector is adjustable and may vary according to different classes of units. For example, the issuing unit 22 might download the revocation vector more frequently (e.g., once per day) to a merchant, and less frequently (e.g., once per week) to a vending machine.

The supply path 40 is representative of different delivery mechanisms. As one example, the supply path 40 can be embodied as an electronic communication channel, such as a direct local connection or a remote connection over a communication network, such as a wire-based network (e.g., the Internet, telephone, cable TV, etc.) or a wireless network (e.g., cellular phone, paging network, satellite, etc.). As another example, the supply path 40 may be a non-electronic delivery mechanism, such as mailing a floppy diskette or other memory device with the revocation vector stored thereon to the point-of-transaction unit 24.

The issuing unit may further encrypt the signed revocation vector, if it is sent over an unsecured network connection. The reader is assumed to be familiar with cryptography, including such functions as public/private key encryption and decryption, digital signing, and signature authentication. For a basic introduction of cryptography, the reader is directed to a text written by Bruce Schneier and entitled "Applied Cryptography: Protocols, Algorithms, and Source Code in C," published by John Wiley & Sons with copyright 1994 (with a second edition in 1996), which is hereby incorporated by reference.

The point-of-transaction unit 24 receives the revocation vector 42 and authenticates the signature attached to the revocation vector 42 as belonging to the issuer. If the signature is valid, the point-of-transaction unit 24 stores the

revocation vector 42 in memory 34 for future offline verification of IC cards 26. The point-of-transaction unit 24 has a client-side partitioning module 44 that contains the same partitioning algorithm used in the issuer-based partitioning module 38. The point-of-transaction unit 24 also has a comparator 46. The partitioning module 38 and comparator 46 are implemented in software, and together form one embodiment of verification code used to verify the IC device in an offline procedure.

It is noted that as an alternative, the issuing unit 22 might simply download a signed revocation list 36 to the point-of-transaction unit 24. In this case, the point-of-transaction unit 24 is configured to convert the revocation list to a revocation vector using the same partitioning process described above as being performed at the issuing unit. However, this alternative consumes more storage and processing resources at the point-of-transaction unit 24, which may not be desirable in all cases. For instance, a vending machine may not be equipped to store and process an entire revocation list.

During a transaction, a consumer tenders the IC card 26, which is interfaced electronically with the point-of-transaction unit 24 via an electronic interface 48. As an example, the IC card 26 might be inserted into a card reader at the point-of-transaction unit. Alternatively, the consumer may be remotely coupled to the point-of-transaction unit 24 via a network connection, in which case a card reader is located at the consumer end to read the IC card 26 and transmit the desired contents over the network to the point-of-transaction unit.

According to one implementation, the IC card 26 outputs its card ID code to the point-of-transaction unit 24. The partitioning module 44 passes the ID code through the partitioning algorithm and the comparator 46 compares the result with the revocation vector 42 to determine whether the card ID code partitions into an

empty or non-empty bucket. If it partitions into an empty bucket, the IC card 26 is not on the revocation list 36 (as of the latest revocation vector) and the point-of-transaction unit 24 can complete the transaction. Conversely, if the ID code partitions into a non-empty bucket, the IC card 26 *might* be on the revocation list 36 because it partitioned to a bucket that contains at least one bad ID code. However, there is no absolute determination that the ID code is bad, only a possibility. The partitioning process and number of buckets are selected so that it is rare for a device ID code to partition to a non-empty bucket.

Assuming the IC card 26 does map to a non-empty bucket, the point-of-transaction unit 24 has several options. One option is to simply deny the transaction, without further processing. As another option, the point-of-transaction unit can perform a secondary determination as to whether the ID code of the tendered IC partitions into a non-empty sub-list, assuming the revocation vector from the issuing unit contains sub-vector information to bucket sub-lists. The point-of-transaction unit processes the ID code according to a secondary partitioning process to analyze whether the ID code partitions to an empty or a non-empty sub-list. Mapping to an empty sub-list indicates that the ID code is not the bad ID code contained in the bucket, and hence the IC card is okay.

A third option is to initiate an online verification procedure, if available. In this example, the point-of-transaction unit 24 can connect to the issuing unit 22 via a bi-directional communications channel 50 to perform an online verification. The point-of-transaction unit 24 sends an authorization request over the channel 50 to the issuing unit, and awaits a reply. The issuing unit 22 performs a conclusive search to determine whether the ID code is on the revocation list 36. This online verification process is conducted according to conventional techniques. The channel 50 may be a proprietary network (e.g., VisaNet) or a public network (e.g.,

Internet). The online verification process might further assign a new ID code to the IC card. That is, a cryptographic exchange takes place to give the IC card a new ID code so that the card does not collide in future revocation checks. The IC card is assigned a new ID code that falls into an empty bucket.

The offline verification procedure is advantageous because the point-of-transaction unit can quickly and accurately discern that the IC card is not on a revocation list, without resorting to an online verification process. Transaction speed is improved because the point-of-transaction unit handles only the revocation vector, as opposed to the entire revocation list. Furthermore, the point-of-transaction unit only turns to online verification in the event that the IC card is found to *possibly* be on the revocation list, which is rare occurrence. As a result, transaction and communications costs are reduced.

IC Card-Based Authorization

According to another implementation, the offline verification process might be performed on the IC card itself, rather than the point-of-transaction unit 24. In this implementation, the point-of-transaction unit 24 can be constructed without the partitioning module 44 or comparator 46, as these components are moved to the IC card 26.

Fig. 2 shows an IC card 26 embodied as a smart card. The IC card 26 has a reader interface 60 for coupling to a card reader, a CPU or processor 62, a volatile rewritable RAM (Random Access Memory) 64, a ROM (Read Only Memory) 66, and an persistent reader/write memory such as EEPROM (Electrically Erasable Programmable ROM) 68. A multi-bit bus (not shown) connects the components.

A cryptography module 70 is stored in ROM 66 and executes on the processor 62 to perform certain cryptographic functions, including encryption,

decryption, signing, and verification. A partitioning module 72 and a comparator 74 are also stored in ROM 66 for execution the processor 62. These software components perform essentially the same function as partitioning module 44 and comparator 46 in the point-of-transaction unit 24 (Fig. 1).

The EEPROM 68 is partitioned into a public storage 76 and a private storage 78. The public storage 76 contains non-confidential user information 80, such as cardholder name and expiration. This information is distributed freely by the IC card 26, without any special security protocol or the need for the user to enter a personal passcode. The private storage 78 maintains information of which the user wishes to control distribution. The processor 62 only retrieves information from the private storage upon authorization by the user as indicated when the user enters a personal passcode.

In this example, the private storage 78 of EEPROM 68 stores the card ID code 82, although this code may be moved to the public storage 76. The private storage 78 further holds cryptography keys 84 for encryption/decryption and signing, electronic assets 86, and any non-cryptographic programs 88 that the user might wish to load onto the IC card.

During a transaction, the IC card 26 is interfaced electronically with the point-of-transaction unit 24 via the electronic interface 48 (Fig. 1). If warranted, the consumer may be asked to enter a passcode to verify the consumer and to enable the processor 62 to access the private storage 78. The point-of-transaction unit 24 passes the revocation vector 42 into the IC card 26, where it is temporarily stored in RAM 64. The cryptography module 70 verifies the issuer's signature on the revocation vector 42 before further processing. If the signature checks out, the ID code 82 is passed through the card-based partitioning module 72, with the result being compared by comparator 74 to the revocation vector in RAM 64 to determine

whether the card ID code 82 partitions into an empty or non-empty bucket. If it partitions into an empty bucket, the IC card 26 is not on the revocation list 36 (as of the latest revocation vector) and the IC card 26 allows the transaction to continue. Conversely, if the ID code partitions into a non-empty bucket, the IC card 26 passes out a warning to the point-of-transaction unit 24 indicating that the card *might* be on the revocation list 36.

Revocation Vector Creation

Figs. 3 and 4 show one implementation of the offline verification system that employs a hash partitioning process to condense a revocation list 36 to a revocation vector 42. Fig. 3 is described in conjunction with steps in the Fig. 4 flow diagram, which are implemented at the issuing unit 22 (or point-of-transaction unit).

In Fig. 3, the revocation list 36 is illustrated with a list of ID codes for corresponding IC cards that have been revoked. The ID codes are formatted as conventional 16-digit credit card numbers. The IC codes on revocation list 36 is passed through the partitioning module 38 (Fig. 1), which employs a first hashing function (Hash 1). The partitioning module 38 creates a hash or partition table having a number of entries that reference corresponding "partitions" or "buckets" 90. The hash function partitions the ID codes on revocation list 36 into buckets 90(1) to 90(n) (step 100 in Fig. 4). Many different kinds of hashing functions can be used. The hash function is selected to provide an even distribution of ID codes among the partitions. The number of buckets is selected with the tradeoff of minimizing the size of the revocation vector, while providing enough buckets that many will remain empty for more rapid processing at the point-of-transaction unit.

If desired, individual buckets may be hashed a second time into multiple sub-lists according to second hashing function (Hash 2) (step 102 in Fig. 4). In Fig. 3,

the ID codes in non-empty buckets 90(2) and 90(n) are hashed into sets of sub-lists 92(2) and 92(n), respectively.

A bucket bit array 94 is derived from the distribution of ID codes in the buckets (step 104 in Fig. 4). In this example, there is one bit in the bucket bit array 94 for each bucket 90. The bit is set to a first binary value, say binary "1", if the corresponding bucket contains at least one ID code (i.e., a non-empty bucket) and to a second binary value, say binary "0", if the corresponding bucket is empty. Here, buckets 90(2) and 90(n) are not empty, and hence the corresponding bit is set.

Sub-list bit arrays 96 are derived from the distribution of ID codes in the sub-lists 92 (step 106 in Fig. 4). There is one bit in each sub-list bit array 96 for each sub-list in the corresponding set. The bit is set to a first binary value, say binary "1", if the corresponding sub-list contains at least one ID code (i.e., a non-empty sub-list) and to a second binary value, say binary "0", if the corresponding bucket is empty.

The bucket bit array 94 and sub-list bit array 96 are concatenated and signed, as represented by the signature 98, to form the signed revocation vector 42 (step 108 in Fig. 4). As an alternative, each bit array may be individually signed. The signed revocation vector is then downloaded to the point-of-transaction unit 24 (step 110 in Fig.4).

Offline Authentication

Fig. 5 shows exemplary steps in a method for offline verification of an IC card 26. These steps can be performed at either the point-of-transaction unit 24 or within the IC card 26. For discussion purposes, the process will be described as being performed by the transaction unit 24. At step 120 in Fig. 5, the point-of-transaction unit 24 authenticates the signature on the revocation vector to ensure

that the vector was signed by the issuer and not subsequently altered. If the authentication process fails, the point-of-transaction unit 24 denies any future transaction until a valid vector is received (step 122 in Fig. 5).

If signature authentication is successful, the partitioning module 44 at the point-of-transaction unit 24 hashes the card ID code using the first hashing function (i.e., Hash 1). The hash result is used to compute a bit index in the bucket bit array 94 (step 124 in Fig. 5). The bit index is used to index into the revocation vector 42, and more particularly, the bucket bit array 94 of the revocation vector (step 126 in Fig. 5). The partitioning module 44 then analyzes whether the indexed bit in the bucket bit array is set, thereby resulting in a collision between the hashed ID code and the bucket bit array (step 128 in Fig. 5).

If the bit is not set, indicating that the bucket is empty and contains no ID codes of revoked IC cards, the transaction is immediately approved (step 130 in Fig. 5). On the other hand, if the bit is set, indicating that the bucket contains at least one revoked ID code, the verification process is continued. At step 132, the partitioning module 44 obtains the sub-list bit array, which is associated with the subject bucket, from the revocation vector 42. At step 134, the partitioning module optionally authenticates the signature on the sub-lists (if individually signed).

The partitioning module 44 hashes the card ID code using the second hashing function (i.e., Hash 2) to yield a bit index into the sub-list bit array (step 136 in Fig. 5). The partitioning module 44 indexes into the sub-list bit array (step 138 in Fig. 5) and analyzes whether the indexed bit in the sub-list bit array is set (step 140 in Fig. 5). If the bit is not set, indicating that the sub-list is empty and contains no ID codes of revoked IC cards, the transaction is approved (step 142 in Fig. 5). If the bit is set, indicating that the bucket contains at least one card ID code, there remains a possibility that the ID card might be revoked. Accordingly,

the point-of-transaction unit 24 can initiate an online verification process to determine whether the IC card has indeed been revoked (step 144).

Verification of the Point-of-Transaction Unit

In another implementation, the roles of the IC device 26 and the point-of-transaction unit 24 are reversed so that the IC device authenticates the point-of-transaction unit. The IC device 26 receives a digitally signed small revocation vector from a trusted third party (i.e., issuer, certifying authority, etc.) along with the credentials for the transaction unit 24 (such as a digitally signed certificate containing the ID code of the unit). The IC device authenticates the digital signature on the revocation vector as well as the credentials as belonging to the trusted third party.

The partitioning module 72 at the IC device 26 hashes the unit's ID code using a hashing function. If the ID code hashes to an empty bucket, the IC device 26 continues with the transaction. On the other hand, if the ID code of the point-of-transaction unit hashes to a non-empty bucket, the IC device has several options, including terminating the session or requiring the unit to engage in an online verification sequence prior to continuing the transaction.

In an online verification sequence, the IC device 26 creates a unique challenge value and requests that the transaction unit 24 submit the challenge value during the online verification session. The point-of-transaction unit obtains a validation from a certifying authority (perhaps the same entity as the trusted third party that issued the revocation vector, although it can be a different entity). The validation is digitally signed by the certifying authority and contains (at a minimum) the ID code of the point-of-transaction unit and challenge value from the IC device. The transaction unit passes the validation to the IC device. The IC device verifies

the signature and the challenge value to ensure that the validation came from the certifying authority and that the transaction unit truly engaged in an online verification session with the certifying authority. If the verification is successful, the IC device repeats the process of hashing and verifying the new ID code received from the certifying authority.

Although the invention has been described in language specific to structural features and/or methodological steps, it is to be understood that the invention defined in the appended claims is not necessarily limited to the specific features or steps described. Rather, the specific features and steps are disclosed as preferred forms of implementing the claimed invention.

CLAIMS

1. A system comprising:

an integrated circuit (IC) device having an associated device identification code stored thereon;

an issuing unit having a database that stores a revocation list of identification codes for associated IC devices that have been revoked, the issuing unit having a processor to partition the identification codes on the revocation list into multiple buckets, the processor deriving a revocation vector that represents a distribution of the identification codes within the buckets; and

a point-of-transaction unit to verify an IC device during an offline verification procedure, the point-of-transaction unit occasionally receiving the revocation vector from the issuing unit for use in performing the offline verification, wherein one of the IC device or the point-of-transaction unit determines whether the identification code of the IC device partitions into a non-empty bucket that is indicated by the revocation vector as containing a device identification code of a revoked IC device.

2. A system as recited in claim 0, wherein the IC device comprises a smart card.

3. A system as recited in claim 0, wherein the issuing unit partitions the identification codes using a hash partitioning process.

4. A system as recited in claim 0, wherein the revocation vector includes a shared secret that is shared between the issuing unit and the point-of-transaction unit.

5. A system as recited in claim 0, wherein the issuing unit digitally signs the revocation vector.

6. A system as recited in claim 0, wherein the point-of-transaction unit performs an online verification process in the event that the identification code of the IC device partitions into the non-empty bucket.

7. A system as recited in claim 6, wherein the IC device is assigned a new identification code that partitions into an empty bucket as a result of the online verification process.

8. A system as recited in claim 0, wherein:

the issuing unit further partitions the identification codes in individual ones of the buckets into sub-lists and the revocation vector includes sub-vectors for the sub-lists; and

said one of the IC device or the point-of-transaction unit further determines whether the identification code of the IC device partitions into a sub-list that is indicated by a revocation sub-vector as containing a device identification code of a revoked IC device.

9. A system comprising:

a transaction unit having an associated identification code stored thereon;

an issuing unit having a database that stores a revocation list of identification codes for associated transaction units that have been revoked, the issuing unit having a processor to partition the identification codes on the revocation list into multiple buckets, the processor deriving a revocation vector that represents a distribution of the identification codes within the buckets; and

an integrated circuit (IC) device being configured to verify the transaction unit during an offline verification procedure, the IC device occasionally receiving the revocation vector from the issuing unit for use in performing the offline verification, wherein one of the IC device determines whether the identification code of the transaction unit partitions into a non-empty bucket that is indicated by the revocation vector as containing an identification code of a revoked transaction unit.

10. A system as recited in claim 9, wherein the issuing unit partitions the identification codes using a hash partitioning process.

11. A system as recited in claim 9, wherein the issuing unit digitally signs the revocation vector.

12. A system as recited in claim 9, wherein the revocation vector includes a shared secret that is shared between the issuing unit and the IC device.

13. A system as recited in claim 9, wherein the issuing unit also provides credentials of the transaction unit to the IC device, the issuing unit digitally signing the credentials.

14. A system as recited in claim 9, wherein the IC device requests the transaction unit to perform an online verification process in the event that the identification code of the transaction unit partitions into the non-empty bucket.

15. A system as recited in claim 14, wherein the transaction unit is assigned a new identification code that partitions into an empty bucket as a result of the online verification process.

16. A system as recited in claim 14, wherein the IC device creates a challenge value and the transaction unit includes the challenge value in the online verification process, the challenge value being included in a response returned as a result of the online verification process.

17. A system as recited in claim 9, wherein:

the issuing unit further partitions the identification codes in individual ones of the buckets into sub-lists and the revocation vector includes sub-vectors for the sub-lists; and

the IC device further determines whether the identification code of the transaction unit partitions into a sub-list that is indicated by a revocation sub-vector as containing an identification code of a revoked transaction unit.

18. A system for performing offline verification of an integrated circuit (IC) device, the IC device having an associated device identification code, the identification code being placed on a revocation list when the associated IC device is revoked, the system comprising:

a partitioning module to partition the identification codes on the revocation list into multiple buckets and derive a revocation vector that represents a distribution of the identification codes within the buckets; and

a verification module to determine whether a particular identification code of a particular IC device partitions into a non-empty bucket that is indicated by the revocation vector as containing at least one identification code of a revoked IC device.

19. A system as recited in claim 18, wherein the partitioning module partitions the identification codes using a hash partitioning process.

20. A system as recited in claim 18, wherein the partitioning module digitally signs the revocation vector and the verification module authenticates the signed revocation vector.

21. A system as recited in claim 18, wherein the verification module performs an online verification process in the event that the particular identification code of the particular IC device partitions into the non-empty bucket.

22. A system as recited in claim 21, wherein the IC device is assigned a new identification code that partitions into an empty bucket as a result of the online verification process.

23. A system as recited in claim 18, wherein:

the partitioning module further partitions the identification codes in individual ones of the buckets into sub-lists and the revocation vector includes sub-vectors for the sub-lists; and

the verification module further determines, in the event that the particular identification code of the particular IC device partitions into the non-empty bucket, whether the particular identification code of the particular IC device partitions into a non-empty sub-list of the non-empty bucket that is indicated by a revocation sub-vector as containing a device identification code of a revoked IC device.

24. A system as recited in claim 18, wherein the verification module is implemented in the particular IC device.

25. A system as recited in claim 18, wherein the verification module is implemented in a point-of-transaction unit that operably interfaces with the particular IC device during verification.

26. A system as recited in claim 18, wherein the partitioning module and the verification module are implemented in a point-of-transaction unit that operably interfaces with the particular IC device during verification.

27. A computer program embodied on a computer-readable medium for offline verification of an identification code, wherein identification codes are placed on a revocation list when revoked, the program comprising:

code means for hashing the revocation list into multiple partitions whereby non-empty partitions contain at least one identification code and empty partitions contain no identification codes;

code means for deriving a revocation vector into the partitions;

code means for hashing a particular identification code that is being verified, said hashing code means producing a hash value; and

code means for comparing the hash value of the particular identification code with the revocation vector to determine whether the particular identification code hashes to a non-empty partition, indicating a possibility that the particular identification code might be on the revocation list.

28. A computer program as recited in claim 27, further comprising code means for digitally signing the revocation vector.

29. A computer program as recited in claim 27, further comprising code means for digitally signing the revocation vector and code means for authenticating the signed revocation vector.

30. A computer program as recited in claim 27, further comprising code means for initiating an online verification process in the event that the particular identification code hashes to the non-empty bucket.

31. A computer program as recited in claim 27, further comprising:

code means for secondarily hashing the identification codes within individual ones of the partitions into sub-lists whereby non-empty sub-lists contain at least one identification code from the corresponding partition and empty sub-lists contain no identification codes;

code means for deriving revocation sub-vectors for the sub-lists; and

code means for determining whether the particular identification code hashes into a non-empty sub-list.

32. An integrated circuit (IC) device configured to perform offline verification, the IC device being supplied a revocation vector that is derived from hashing a revocation list of revoked IC devices, the IC device comprising:

a processor;

a memory to hold a device identification code and the revocation vector; and

verification code stored in the memory and executed on the processor to hash the identification code and compare the hashed identification code with the revocation vector for possible collision, wherein a collision indicates a possibility that the particular IC device might be on the revocation list.

33. An IC device as recited in claim 32, wherein the verification code performs a secondary offline verification process in an event that the hashed identification code collides with the revocation vector.

34. An IC device as recited in claim 32, wherein the verification code denies a transaction with the IC device in an event that the hashed identification code collides with the revocation vector.

35. An IC device as recited in claim 32, wherein the revocation vector has a digital signature of an entity attached thereto, further comprising:

signature authentication code stored in memory and executed on the processor to authenticate the digital signature of the revocation vector.

36. A point-of-transaction unit for performing offline verification of a portable integrated circuit (IC) device, the point-of-transaction unit being supplied with a device identification code from the IC device and a revocation vector that is derived from hashing a revocation list of revoked IC devices, the point-of-transaction unit comprising:

a processor;

a memory to hold the identification code and the revocation vector; and

verification code stored in the memory and executed on the processor to hash the identification code and compare the hashed identification code with the revocation vector for possible collision, wherein a collision indicates a possibility that the particular IC device might be on the revocation list.

37. A point-of-transaction unit as recited in claim 36, wherein the verification code performs a secondary offline verification process in an event that the hashed identification code collides with the revocation vector.

38. A point-of-transaction unit as recited in claim 36, wherein the verification code denies initiates an online verification process in an event that the hashed identification code collides with the revocation vector.

39. A point-of-transaction unit as recited in claim 36, wherein the verification code denies a transaction with the IC device in an event that the hashed identification code collides with the revocation vector.

40. A point-of-transaction unit as recited in claim 36, wherein the revocation vector has a digital signature of an entity attached thereto, further comprising:

signature authentication code stored in memory and executed on the processor to authenticate the digital signature of the revocation vector.

41. A method for offline verification of an integrated circuit (IC) device, comprising the following steps:

partitioning a revocation list of revoked device identification codes, which are associated with revoked IC devices, into multiple buckets whereby non-empty buckets contain at least one identification code and empty buckets contain no identification codes;

deriving a revocation vector into the buckets; and

determining, during offline verification of a particular IC device having a particular identification code, whether the particular identification code partitions into a non-empty bucket that is indicated by the revocation vector as containing a device identification code of a revoked IC device.

42. A method as recited in claim 41, further comprising the step of digitally signing the revocation vector with a signature.

43. A method as recited in claim 42, further comprising the step of authenticating the signature on the revocation vector during the offline verification.

44. A method as recited in claim 42, further comprising the step of initiating an online verification process in the event that the particular identification code partitions into the non-empty bucket.

45. A method as recited in claim 44, further comprising the step of assigning a new identification code to the IC device as a result of the online verification, the new identification code being selected to partition into an empty bucket.

46. A method as recited in claim 42, further comprising the following steps:

partitioning the identification codes in individual ones of the buckets into multiple sub-lists;

deriving revocation sub-vectors for the sub-lists; and

determining, in the event that the particular identification code partitions into the non-empty bucket, whether the particular identification code partitions into a sub-list of the non-empty bucket that is indicated by a revocation sub-vector as containing a device identification code of a revoked IC device.

47. Computer-readable media comprising computer-readable instructions for performing the steps in the method as recited in claim 42.

48. A method for offline verification of an integrated circuit (IC) device, comprising the following steps:

at an issuing authority:

hashing a revocation list of revoked identification codes, which are associated with revoked IC devices, into multiple buckets whereby non-empty buckets contain at least one identification code and empty buckets contain no identification codes;

deriving a revocation vector into the buckets;

occasionally downloading the revocation vector to a point of verification;

at the point of verification:

hashing a particular identification code, which is associated with a particular IC device that is being verified, to produce a bucket index; and

indexing the revocation vector using the bucket index to determine whether the particular identification code hashes to a non-empty bucket, indicating a possibility that the particular IC device might be on the revocation list.

49. A method as recited in claim 48, further comprising the step of digitally signing the revocation vector with a signature of the issuing authority.

50. A method as recited in claim 49, further comprising the step of authenticating the issuing authority's signature on the revocation vector at the point of verification.

51. A method as recited in claim 48, further comprising the step of initiating an online verification process in the event that the particular identification code hashes into the non-empty bucket.

52. A method as recited in claim 48, further comprising the following steps:

hashing the identification codes in individual ones of the buckets into multiple sub-lists;

deriving revocation sub-vectors for the sub-lists; and

determining, in the event that the particular identification code hashes into the non-empty bucket, whether the particular identification code hashes into a sub-list of the non-empty bucket that is indicated by a revocation sub-vector as containing a device identification code of a revoked IC device.

53. Computer-readable media distributed at the issuing authority and the point of verification comprising computer-readable instructions for performing the steps in the method as recited in claim 48.

54. A method for offline verification of an identification code, wherein revoked identification codes are placed on a revocation list, the revocation list being hash partitioned into multiple buckets and a revocation vector to the buckets being derived that is capable of distinguishing empty buckets from non-empty buckets, the method comprising the following steps:

hashing an identification code to produce a hash value; and

comparing the hash value with the revocation vector to determine whether the identification code maps to a non-empty bucket, indicating a possibility that the identification code might be on the revocation list.

55. A method as recited in claim 54, further comprising the step of performing a secondary offline verification process in an event that the identification code maps to the non-empty bucket.

56. A method as recited in claim 54, further comprising the step of initiating an online verification process in an event that the identification code maps to the non-empty bucket.

57. A method as recited in claim 56, further comprising the step of assigning a new identification code to the IC device as a result of the online verification, the new identification code being selected to partition into an empty bucket.

58. A method as recited in claim 54, further comprising the step of denying a transaction in an event that the identification code maps to the non-empty bucket.

59. A computer-readable medium comprising computer-readable instructions for performing the steps in the method as recited in claim 54.

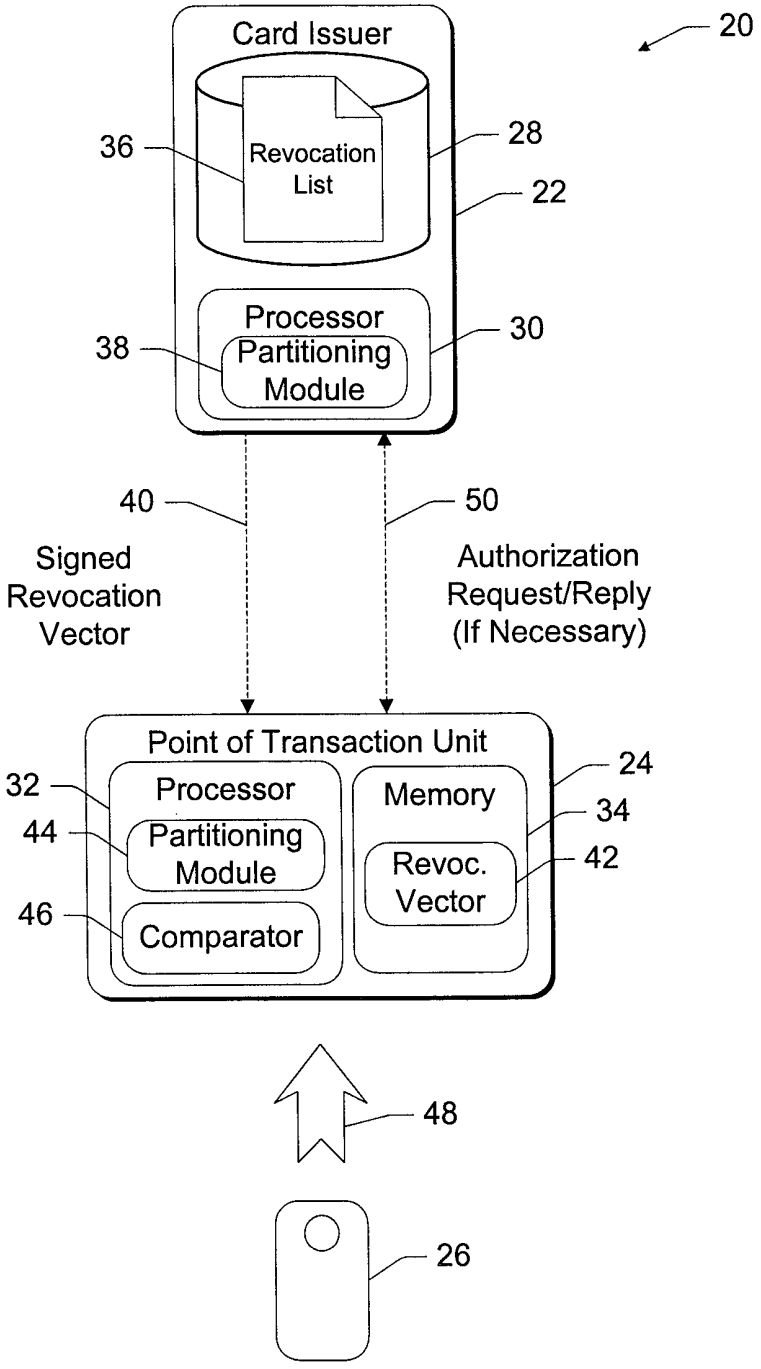

Card Issuer
20
36
Revocation List
28
22
Processor
38
Partitioning Module
30
40
50
Signed Revocation Vector
Authorization Request/Reply (If Necessary)
Point of Transaction Unit
24
32
Processor
Memory
34
44
Partitioning Module
Revoc. Vector
42
46
Comparator
48
26

Fig. 1

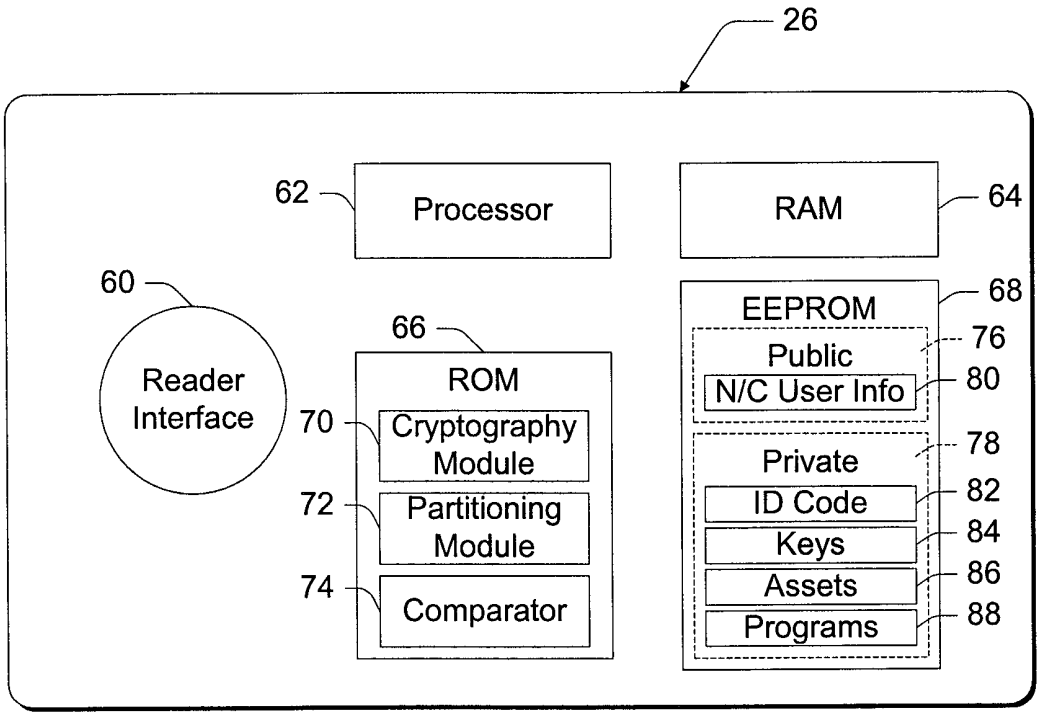
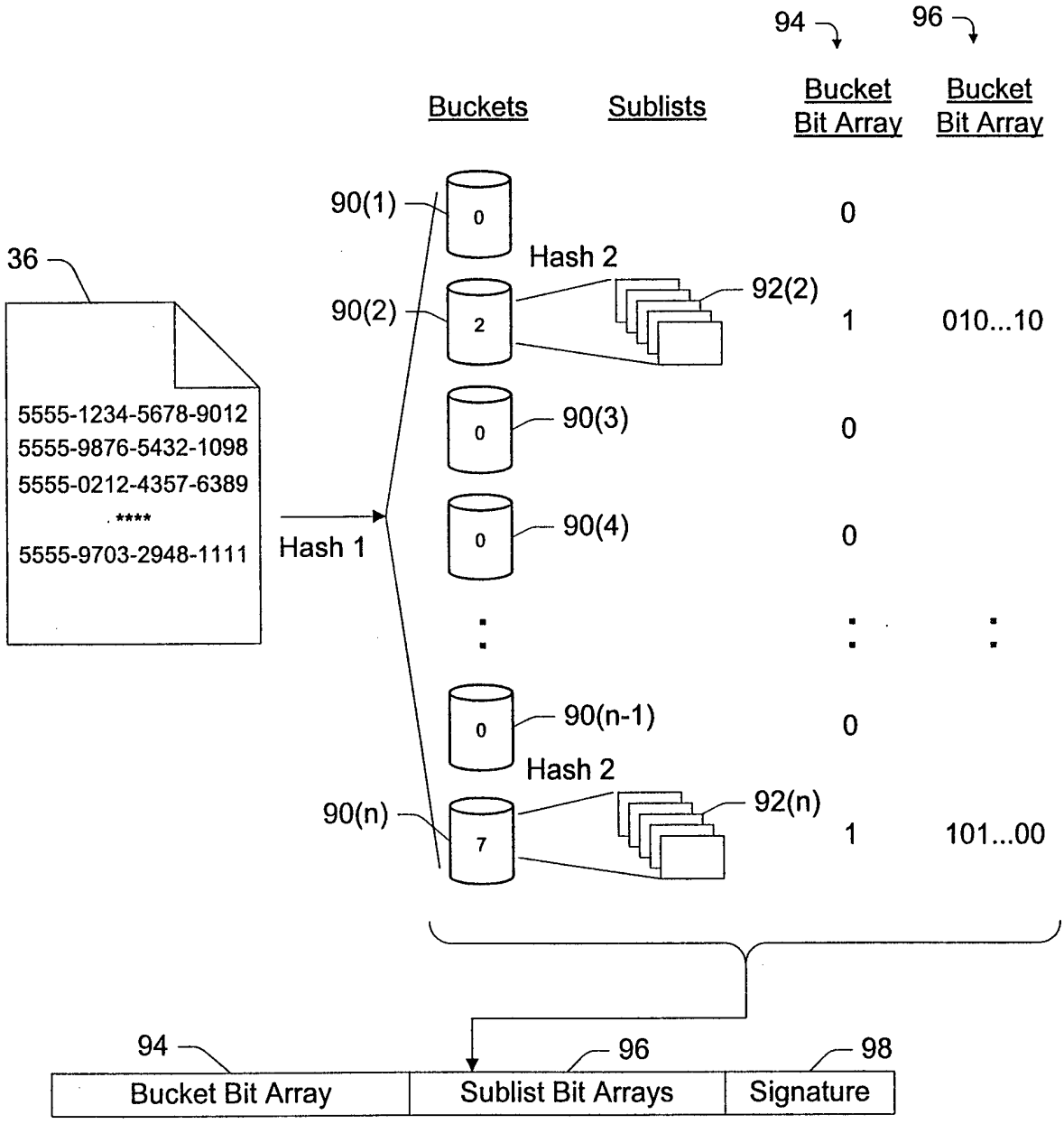

26
62
Processor
RAM
64
60
Reader Interface
66
ROM
70
Cryptography Module
72
Partitioning Module
74
Comparator
EEPROM
68
Public
76
N/C User Info
80
Private
78
ID Code
82
Keys
84
Assets
86
Programs
88

Fig. 2



Revocation Vector 42

Fig. 3

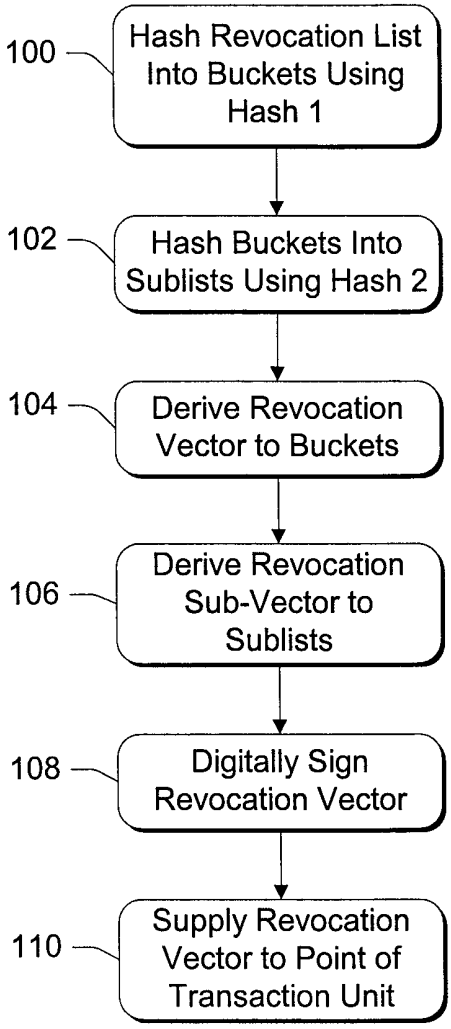

100
Hash Revocation List Into Buckets Using Hash 1
102
Hash Buckets Into Sublists Using Hash 2
104
Derive Revocation Vector to Buckets
106
Derive Revocation Sub-Vector to Sublists
108
Digitally Sign Revocation Vector
110
Supply Revocation Vector to Point of Transaction Unit

Fig. 4

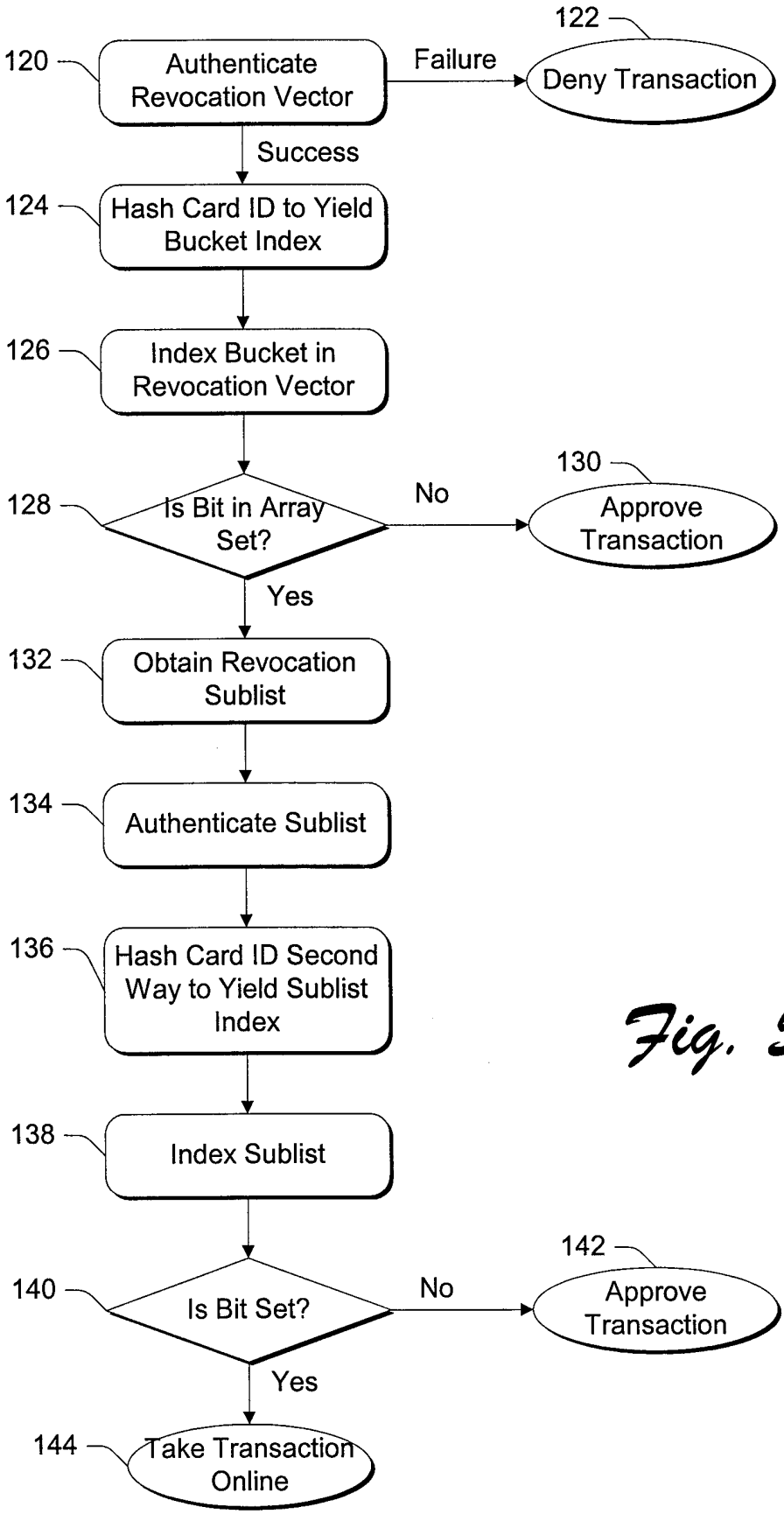

120
Authenticate Revocation Vector
Failure
122
Deny Transaction
Success
124
Hash Card ID to Yield Bucket Index
126
Index Bucket in Revocation Vector
128
Is Bit in Array Set?
No
130
Approve Transaction
Yes
132
Obtain Revocation Sublist
134
Authenticate Sublist
136
Hash Card ID Second Way to Yield Sublist Index
138
Index Sublist
140
Is Bit Set?
No
142
Approve Transaction
Yes
144
Take Transaction Online

Fig. 5

INTERNATIONAL SEARCH REPORT

Int. ional Application No
PCT/US 99/17503

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G07F7/08

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G07F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 349 413 A (SCHLUMBERGER IND SA) 3 January 1990 (1990-01-03) column 1, line 58 -column 2, line 57 column 3, line 47 -column 5, line 15 figures — -/—	1-3,6, 18,19, 21, 25-27, 30,36, 41,48, 51,53, 54,56,59

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
23 November 1999	30/11/1999

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Bocage, S

2

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Int ional Application No
PCT/US 99/17503

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 378 349 A (VISA INT SERVICE ASS) 18 July 1990 (1990-07-18) page 3, line 19 -page 5, line 11 page 6, line 10 -page 9, line 5 figure 1	1,3,6, 18,19, 21, 25-27, 30,36, 37,41, 48,51, 53,54, 56,59
A	US 5 396 624 A (CAMPBELL JR CARL M) 7 March 1995 (1995-03-07) column 2, line 64 -column 3, line 66 column 7, line 55 -column 9, line 53 column 11, line 61 -column 12, line 18 claim 1; figures 3,4,6	1,3,6, 18,19, 21, 25-27, 30,36, 37,41, 48,51, 53,54, 56,59

2

INTERNATIONAL SEARCH REPORT

Information on patent family members

Int tional Application No
PCT/US 99/17503

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0349413 A	03-01-1990	FR 2633411 A	29-12-1989
		DE 68909126 D	21-10-1993
		DE 68909126 T	13-01-1994
		US 5103079 A	07-04-1992
EP 0378349 A	18-07-1990	US 4908521 A	13-03-1990
		AT 110868 T	15-09-1994
		AU 613574 B	01-08-1991
		AU 4708389 A	19-07-1990
		CA 2007234 A,C	10-07-1990
		DE 69011877 D	06-10-1994
		DE 69011877 T	20-04-1995
		ES 2063911 T	16-01-1995
		JP 2226362 A	07-09-1990
		JP 2714869 B	16-02-1998
		NO 300944 B	18-08-1997
US 5396624 A	07-03-1995	NONE	