PCT ORGANISATION MONDIALE DE LA PROPRIETE INTELLECTUELLE
Bureau international

DEMANDE INTERNATIONALE PUBLIEE EN VERTU DU TRAITE DE COOPERATION EN MATIERE DE BREVETS (PCT)

(51) Classification internationale des brevets 6 : G07F 7/10 — A1

(11) Numéro de publication internationale: **WO 99/53451**

(43) Date de publication internationale: 21 octobre 1999 (21.10.99)

(21) Numéro de la demande internationale: PCT/FR99/00837

(22) Date de dépôt international: 9 avril 1999 (09.04.99)

(30) Données relatives à la priorité:
98/04453 9 avril 1998 (09.04.98) FR

(71) Déposant *(pour tous les Etats désignés sauf US)*: INNOVATRON ELECTRONIQUE, SOCIETE ANONYME [FR/FR]; 1, rue Danton, F-75006 Paris (FR).

(72) Inventeurs; et

(75) Inventeurs/Déposants *(US seulement)*: DIDIER, Stéphane [FR/FR]; 113, rue de Meaux, F-75019 Paris (FR). GRIEU, François [FR/FR]; 8, rue de Rambouillet, F-75012 Paris (FR).

(74) Mandataire: DUPUIS-LATOUR, Dominique; Cabinet Bardehle, Pagenberg & Partner, 14, boulevard Malesherbes, F-75008 Paris (FR).

(81) Etats désignés: AE, AL, AU, BA, BB, BG, BR, CA, CN, CU, CZ, EE, GE, HR, HU, ID, IL, IN, IS, JP, KP, KR, LC, LK, LR, LT, LV, MG, MK, MN, MX, NO, NZ, PL, RO, SG, SI, SK, SL, TR, TT, UA, UG, US, UZ, VN, YU, ZA, brevet ARIPO (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), brevet eurasien (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), brevet européen (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), brevet OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Publiée

Avec rapport de recherche internationale.
Avant l'expiration du délai prévu pour la modification des revendications, sera republiée si des modifications sont reçues.

(54) **Title:** METHOD FOR INDIVISIBLY MODIFYING A PLURALITY OF SITES IN A MICROCIRCUIT CARD NON VOLATILE MEMORY, IN PARTICULAR A CONTACTLESS CARD

(54) **Titre:** PROCEDE POUR MODIFIER DE MANIERE INDIVISIBLE UNE PLURALITE D'EMPLACEMENTS DE LA MEMOIRE NON VOLATILE D'UNE CARTE A MICROCIRCUIT, NOTAMMENT UNE CARTE SANS CONTACT

(57) **Abstract**

The card is temporarily connected to a terminal while a transaction is being executed comprising the application by the terminal to the card of a plurality of modification commands each comprising at least an operation for posting, in the card memory, a respective information indicated by the command, the different data being thus posted mutually interdependent. Said method comprises the following steps executed by the card: a) on receiving from the terminal the corresponding respective modification commands, modifying the card memory content by provisional posting, in the card memory, each of said interdependent data without losing previous values corresponding to said data; then b) finalising said modifications, either by confirming all of them, or by denying them, such that for subsequent operations the commands executed at step a) are either all taken into account, or are all null and void.

(57) **Abrégé**

La carte est couplée temporairement à un terminal pendant l'exécution d'une transaction comportant l'application par le terminal à la carte d'une pluralité de commandes de modifications comprenant chacune au moins une opération d'inscription, dans la mémoire de la carte, d'une information respective désignée par la commande, les différentes informations ainsi inscrites étant mutuellement interdépendantes. Ce procédé comprend l'exécution par la carte des étapes suivantes: a) sur réception de commandes respectives correspondantes reçues du terminal, modifications du contenu de la mémoire de la carte par inscription provisoire, dans la mémoire de la carte, de chacune desdites informations interdépendantes sans perte de valeurs antérieures correspondant à ces informations; puis b) finalisation de ces modifications, soit en les confirmant toutes, soit en les infirmant toutes, de sorte que pour des opération ultérieures les commandes exécutées à l'étape a) soient soit toutes prises en compte, soit toutes sans effet.

UNIQUEMENT A TITRE D'INFORMATION

Codes utilisés pour identifier les Etats parties au PCT, sur les pages de couverture des brochures publiant des demandes internationales en vertu du PCT.

AL	Albanie	ES	Espagne	LS	Lesotho	SI	Slovénie
AM	Arménie	FI	Finlande	LT	Lituanie	SK	Slovaquie
AT	Autriche	FR	France	LU	Luxembourg	SN	Sénégal
AU	Australie	GA	Gabon	LV	Lettonie	SZ	Swaziland
AZ	Azerbaïdjan	GB	Royaume-Uni	MC	Monaco	TD	Tchad
BA	Bosnie-Herzégovine	GE	Géorgie	MD	République de Moldova	TG	Togo
BB	Barbade	GH	Ghana	MG	Madagascar	TJ	Tadjikistan
BE	Belgique	GN	Guinée	MK	Ex-République yougoslave de Macédoine	TM	Turkménistan
BF	Burkina Faso	GR	Grèce			TR	Turquie
BG	Bulgarie	HU	Hongrie	ML	Mali	TT	Trinité-et-Tobago
BJ	Bénin	IE	Irlande	MN	Mongolie	UA	Ukraine
BR	Brésil	IL	Israël	MR	Mauritanie	UG	Ouganda
BY	Bélarus	IS	Islande	MW	Malawi	US	Etats-Unis d'Amérique
CA	Canada	IT	Italie	MX	Mexique	UZ	Ouzbékistan
CF	République centrafricaine	JP	Japon	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Pays-Bas	YU	Yougoslavie
CH	Suisse	KG	Kirghizistan	NO	Norvège	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	République populaire démocratique de Corée	NZ	Nouvelle-Zélande		
CM	Cameroun			PL	Pologne		
CN	Chine	KR	République de Corée	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Roumanie		
CZ	République tchèque	LC	Sainte-Lucie	RU	Fédération de Russie		
DE	Allemagne	LI	Liechtenstein	SD	Soudan		
DK	Danemark	LK	Sri Lanka	SE	Suède		
EE	Estonie	LR	Libéria	SG	Singapour		

Procédé pour modifier de manière indivisible une pluralité d'emplacements de la mémoire non volatile d'une carte à microcircuit, notamment une carte sans contact

L'invention concerne les cartes à microcircuit, et plus particulièrement les cartes à microprocesseur, qui réalisent elles-mêmes diverses modifications de leur mémoire non volatile.

Lors de l'exécution d'une transaction, la mémoire est généralement modifiée, une ou plusieurs fois, et il est bien entendu nécessaire de s'assurer alors que toutes les modifications ont bien été correctement effectuées avant de pouvoir exploiter les informations nouvellement inscrites, les informations nouvellement inscrites devant être ignorées ou effacées en cas d'erreur ou de défaut d'intégrité de l'inscription.

Le US-A-4 877 945 décrit ainsi la manière de détecter une anomalie survenue au cours d'une séquence d'écriture de plusieurs informations afin d'empêcher la poursuite de la transaction sur des bases erronées.

Il est par ailleurs souhaitable, en cas d'anomalie, de pouvoir revenir au *statu quo ante*, c'est-à-dire qu'une transaction ultérieure devra être à même d'opérer sur les valeurs des informations qui étaient inscrites dans la carte avant l'exécution de la transaction incorrecte.

Le US-A-4 877 945 précité n'offre pas cet avantage, car les anciennes valeurs des informations auront, pour certaines, été perdues pendant l'exécution de la transaction incorrecte, de sorte qu'il ne sera pas possible de restaurer ces informations à leur état antérieur, du moins à partir des seules informations contenues dans cette carte.

Le WO-A-89/02140, quant à lui, décrit une telle manière d'opérer, mais qui n'est applicable qu'au cas de la modification d'une information unique ou de plusieurs modifications d'informations indépendamment les unes de autres.

Dans de nombreux cas, il est cependant nécessaire de modifier au cours de la même transaction plusieurs informations, et elles seront considérées "mutuellement interdépendantes" si elles nécessitent d'être traitées ensemble pour la bonne exécution de l'ensemble des modifications de l'ensemble des informations.

Le risque de transaction imparfaite ou inachevée portant sur une pluralité d'informations interdépendantes est particulièrement élevé avec les cartes du type "sans contact", où les limites du volume dans lequel la carte peut fonctionner correctement autour du terminal ne sont pas perceptibles. Il existe dans ce cas un risque non négligeable de rupture inattendue de la communication entre carte et terminal, dû à la sortie de la carte du rayon d'action du terminal avant la fin du traitement, ou du fait d'une perturbation passagère, par exemple le passage d'une masse métallique à proximité.

Un exemple (bien entendu non limitatif) est l'utilisation d'une telle carte dans une transaction de télébillétique, c'est-à-dire pour l'accès à un réseau de transport public, la carte jouant le double rôle de titre de transport et de porte-monnaie électronique.

Pour pallier les difficultés précitées, et rendre "indivisibles" une pluralité d'écritures ou autres modifications de données interdépendantes, plusieurs solutions ont été proposées.

Dans l'exemple d'application indiqué plus haut, les systèmes connus commencent par débiter le porte-monnaie, puis inscrivent les droits de transport acquis par l'usager. Si l'usager retire sa carte entre les deux opérations, il est invité à présenter la carte à nouveau et l'écriture des droits de transport est reprise. En revanche, s'il part sans représenter sa carte, il aura été lésé. Il est bien évidemment impossible de procéder dans l'ordre inverse car l'usager aurait alors intérêt à retirer sa carte avant que le porte-monnaie ne soit débité.

Cette solution implique que le terminal soit spécialement configuré pour permettre, en cas d'interruption, l'activation d'un traitement d'exception gérant la reprise de la transaction (réinsertion de la carte sur demande du terminal). Outre la complexification du logiciel du terminal, cette solution n'est pas totalement satisfaisante dans la mesure où, comme on l'a indiqué, l'usager se trouve néanmoins lésé en cas de non-reprise de la transaction.

Une autre solution consiste à utiliser des informations croisées, en conservant dans le terminal des informations sur l'état du porte-monnaie de la carte, et réciproquement. Mais cette solution n'est pas non plus satisfaisante car, outre sa complexité, elle augmente le volume de

données échangées entre carte et terminal et ralentit donc l'exécution de la transaction. Elle est en outre difficilement applicable à un nombre important d'écritures à rendre indivisibles (trois et plus).

L'un des buts de l'invention est de proposer un procédé permettant d'effectuer une pluralité de modifications de la mémoire de la carte de manière indivisible.

Un autre but de l'invention est de proposer un tel procédé qui puisse être entièrement géré par la carte. Ce procédé pourra donc être mis en œuvre sans modification des terminaux et sans qu'il y ait lieu de prévoir des traitements d'exception par ces terminaux, en utilisant la syntaxe des ordres existants et donc avec une grande souplesse dans le choix des commandes.

Le procédé de l'invention est du type dans lequel la carte est couplée temporairement à un terminal pendant l'exécution d'une transaction comportant l'application par le terminal à la carte d'une pluralité de commandes de modifications comprenant chacune au moins une opération d'inscription, dans la mémoire de la carte, d'une information respective désignée par la commande, les différentes informations ainsi inscrites étant mutuellement interdépendantes.

De façon caractéristique de l'invention, ce procédé comprend l'exécution, par la carte, des étapes suivantes : a) sur réception de commandes respectives correspondantes reçues du terminal, modifications du contenu de la mémoire de la carte par inscription provisoire, dans la mémoire de la carte, de chacune desdites informations interdépendantes sans perte de valeurs antérieures correspondant à ces informations ; puis b) finalisation de ces modifications, soit en les confirmant toutes, soit en les infirmant toutes, de sorte que pour des opérations ultérieures les commandes exécutées à l'étape a) soient soit toutes prises en compte, soit toutes sans effet.

Le principe de base de l'invention consiste ainsi à grouper la pluralité de modifications à réaliser de manière indivisible au sein d'une même étape a) et, après avoir exécuté ces modifications, à valider globalement ces modifications par la carte. Si la validation est effective, à la prochaine opération effectuée par la carte (au cours de la même transaction ou au cours d'une transaction ultérieure), son contenu accessi-

ble reflétera nécessairement les modifications opérées.

Inversement, toute interruption du fonctionnement de la carte intervenant au cours de l'étape a) annulera l'ensemble des modifications effectuées, et les données de la mémoire non volatile resteront dans leur état antérieur à l'étape a).

Dans un mode de réalisation particulier, en cas de confirmation à l'étape b), on inscrit dans la mémoire de la carte un témoin confirmatif de bonne exécution et, lorsque la carte reçoit ultérieurement une commande impliquant la lecture et/ou la modification de l'une au moins des informations inscrites à l'étape a) ou de la valeur y correspondant, la carte examine préalablement l'état du témoin et, si celui-ci n'a pas été inscrit, la carte ignore ou annule les inscriptions provisoires antérieurement opérées à l'étape a) et exécute la commande sur la base desdites valeurs antérieures correspondant aux informations. Lorsque la carte examine l'état du témoin, si celui-ci a été inscrit la carte peut alors exécuter des opérations de recopie des écritures provisoires opérées à l'étape a).

Très avantageusement, la carte est apte à fonctionner selon deux modes, à savoir un mode en session, dans lequel les inscriptions sont opérées par exécution des étapes a) et b), et un mode hors session, dans lequel l'opération des inscriptions n'est pas confirmée à l'ensemble des étapes a) et b).

L'ouverture de session peut être implicite, par exemple à la remise à zéro (reset) de la carte ou suite à une commande à double action d'exécution d'une opération prédéterminée et interprétée comme un ordre d'ouverture de session.

Par exemple, quand une inscription normalement certifiée n'est pas accompagnée d'un certificat, la carte ouvre automatiquement une session qui traite l'inscription dans cette session.

De la même façon, la fermeture de session peut être implicite, suite à une commande à double action d'exécution d'une opération prédéterminée et interprétée comme un ordre de fermeture de session.

Par exemple, une opération de débit du porte-monnaie ferme la session, ce qui de plus évite de devoir différer la communication du certificat résultant et permet de confondre les certificats de session avec

ceux de transaction du porte-monnaie.

Très avantageusement, le procédé comprend une fonction d'authentification combinée à la fonction de finalisation de l'étape b), forçant l'infirmation à l'étape b) dans le cas où l'authentification échoue.

Dans une première mise en œuvre, cette authentification est opérée par la carte qui authentifie le terminal et/ou les données échangées entre terminal et carte, la carte contrôlant un certificat cryptographique produit par le terminal et transmis à la carte et ne confirmant les modifications à l'étape b) que si ce certificat est reconnu correct.

Dans le cas d'un mode avec session, on peut prévoir que, lorsque la carte reçoit du terminal des commandes de modification du contenu de la mémoire incluant la vérification d'un certificat cryptographique, cette vérification est opérée si la commande est reçue hors session, et ne l'est pas si la commande est reçue en session.

En d'autres termes, celles des commandes exécutées par la carte b) à l'étape b) qui normalement (c'est-à-dire hors session) vérifieraient un certificat cryptographique, ne comprennent plus cette vérification quand elles sont exécutées dans le cadre d'une session, le "certificat de session authentifiant le terminal" réalisant une fonction équivalente.

Dans une seconde mise en œuvre, l'authentification est opérée par le terminal qui authentifie la carte et/ou les données échangées entre terminal et carte, la carte produisant et transmettant au terminal un certificat cryptographique de manière conditionnelle, si et seulement si les modifications ont été confirmées à l'étape b).

Dans le cas d'un mode avec session, on peut prévoir que, lorsque la carte reçoit du terminal à l'étape b) des commandes de modification du contenu de la mémoire incluant la production d'un certificat cryptographique, cette production est opérée si la commande est reçue hors session, et ne l'est pas si la commande est reçue en session.

En d'autres termes, celles des commandes exécutées par la carte à l'étape b) qui normalement (c'est-à-dire hors session) produiraient un certificat cryptographique, ne comprennent plus cette production quand elles sont exécutées dans le cadre d'une session, le "certificat de session authentifiant le terminal" réalisant une fonction équivalente.

On peut par ailleurs prévoir que, lorsque la carte reçoit du terminal

des commandes de modification du contenu de la mémoire incluant la production d'une pluralité de certificats cryptographiques, ces certificats sont mémorisés à cette étape b), puis transmis ensemble au terminal si et seulement si les modifications ont été confirmées à l'étape b).

En d'autres termes, on prévoit de différer la communication par la carte des certificats cryptographiques produits normalement par les ordres de l'étape b). En particulier, si une commande d'écriture certifiée produit un certain certificat d'écriture, il est souhaitable que celui-ci ne sorte de la carte qu'une fois l'écriture effectuée irrévocablement.

Dans une forme de mise en œuvre particulière, au moins certaines des commandes susceptibles d'être exécutées à l'étape b) comprennent un éventuel attribut d'inhibition et, si la carte exécute une telle commande en session à une étape b), les modifications opérées par cette commande prennent effet indépendamment du résultat de l'étape b).

En d'autres termes, l'attribut définit si la commande est effectuée en session (c'est-à-dire sera annulée si la session n'est pas fermée) ou hors session (c'est-à-dire effective immédiatement, comme si elle était effectuée hors session, même si elle est chronologiquement en session).

Très avantageusement, le procédé prévoit en outre, après l'étape b) et en cas de confirmation des modifications, la séquence d'étapes suivante : d) exécution par le terminal d'une action suite à la confirmation par la carte ; e) en cas de bonne exécution de ladite action par le terminal, inscription dans la carte d'une information de ratification ultérieurement accessible en lecture.

Une telle "ratification" de la session indique à la carte que le terminal a effectivement pu prendre les décisions (par exemple l'ouverture d'un portillon dans le cas d'une application d'accès à un réseau de transport en commun) suite à l'exécution de la session.

On notera que cette ratification est gérée par la carte sans nécessité d'une écriture supplémentaire (la recopie des écritures provisoires étant une opération qui, de toute façon, doit être tôt ou tard effectuée). En outre cette recopie n'est opérée côté carte qu'à condition que l'action est bien exécutée côté terminal, c'est-à-dire uniquement en cas de cohérence de l'ensemble de la transaction.

L'ensemble des opérations étant gérée par la carte, on peut avanta-

geusement prévoir que la commande d'inscription de l'étape c) est une commande implicite, toute commande reçue par la carte après l'étape b) étant interprétée comme un ordre d'inscription dans la carte d'une information de ratification.

D'autres caractéristiques et avantages ressortiront de la description ci-dessous de deux exemples de mise en œuvre de l'invention.

Dans ces exemples, comme d'ailleurs dans le reste du texte, le mot "désigner", ici entendu au sens de "déterminer un parmi plusieurs", vise l'action consistant à caractériser une information particulière parmi les différentes informations contenues dans la carte.

Cette désignation peut être implicite, parce que la commande vise par elle-même une information particulière ; par exemple, la commande "débiter le porte-monnaie d'un montant x" désigne l'emplacement mémoire contenant la valeur de l'information "solde du porte-monnaie".

Elle peut être également explicite, comme par exemple dans l'exemple I ci-dessous, où il est prévu des commandes d'écriture avec une adresse ou un identifiant de secteur, indexés par un indice i.

Exemple I

On se propose de réaliser une carte stockant 100 valeurs de huit octets chacune, et supportant les ordres :

- Lecture d'une valeur v de 8 octets, désignée par son indice i de 1 à 100.
- Écriture d'une valeur v de 8 octets, désignée par son indice i de 1 à 100.
- Ouverture de session.
- Fermeture de session.

La carte doit permettre jusqu'à trois écritures dans une même session. Par convention, on utilisera des lettres majuscules pour désigner les valeurs en mémoire non volatile (EEPROM par exemple) et des lettres minuscules pour désigner les valeurs en mémoire volatile (RAM, dont le contenu est perdu à la mise hors tension).

Une zone de mémoire non volatile est affectée au stockage principal

des données de la carte (écritures définitives) :

- V[i], i de 1 à 100 : 100 x 8 octets

Une autre zone de mémoire non volatile est affectée au mécanisme de session, et comprend :

- T[k], j de 1 à 3 : 3 x 8 octets contenant les valeurs écrites pendant la session (écritures provisoires).
- I[k], j de 1 à 3 : 3 x 1 octet contenant les indices des valeurs écrites pendant la session.
- C : 1 octet de comptage qui sera écrit en fin de session.

C code le nombre d'écritures effectuées dans la session ; un mécanisme de redondance approprié (associant par exemple le complément de cette valeur) permet d'assurer que l'on sait détecter le cas où la valeur stockée dans cet octet de comptage est incertaine.

Le déroulement des opérations est le suivant.

Étape 0 : à un moment compris entre la mise sous tension de la carte et la première commande réalisée, C est examiné. S'il est à une valeur certaine de 1 à 3, alors pour k de 1 à C on copie la valeur T[k] à l'indice I[k] du tableau V[i]. Puis C est mis à 0, et une variable interne j à -1 (pour indiquer qu'une session n'est pas ouverte).

Étape 1 : à la lecture on examine si j>0; si oui, on compare l'indice i demandé avec les valeurs I[k] pour k de j à 1 en décroissant. En cas d'identité, on retourne T[k]. Dans tous les autres cas, on retourne V[i].

Étape 2 : à l'ouverture de session, on initialise j = 0 (à noter que si une session est ouverte, elle est annulée)

Étape 3 : à chaque écriture, si j =-1 (session non ouverte), on écrit la valeur v communiquée en T[0], l'indice i communiqué en I[0], puis on écrit C=1, puis on écrit v en V[i], puis on écrit C=0; si 0≤j<3 (écriture en session), on augmente j de 1, on écrit v en T[j], on écrit i en I[j]; si j=3 on refuse l'opération (dépassement de la limite des écritures en session).

Étape 4 : à la fermeture de session, si j>0, on écrit j en C, puis pour j de 1 à C on copie la valeur T[j] à l'indice I[j] du tableau V[]. Puis C est mis à 0, et j à -1.

On montre qu'à tout moment on peut couper l'alimentation de la carte et que les valeurs lues seront correctes, c'est-à-dire pour chaque indice i la dernière valeur écrite hors session ou écrite dans une session close (l'écriture est achevée ou la session est close au moment où une valeur non nulle est écrite dans C).

La cryptographie s'ajoute en empêchant certaines opérations si un certificat cryptographique fourni à la carte est incorrect, et/ou en faisant produire à la carte des certificats cryptographiques à l'issue de certaines opérations.

Les certificats cryptographiques utilisés sont basés sur une cryptographie de type connu. Par exemple, le "certificat de session authentifiant la carte" (respectivement, le terminal) est obtenu en appliquant côté carte et terminal l'algorithme *Secure Hash Algorithm* (SHA) aux données fournies par la carte (resp. le terminal) et à un nombre aléatoire fourni par le terminal (resp. la carte) à l'ouverture de la session ; le *Message Authentication Code* (MAC) résultant est signé par la carte (resp. le terminal) par l'algorithme de signature *Digital Signature Algorithm* (DSA) avec une clé secrète contenue dans la carte (resp. le terminal) ; le terminal (resp. la carte) vérifie cette signature avec une clé publique. Un algorithme de cryptographie symétrique tel que *Data Encryption Standard* (DES) peut aussi être utilisée pour la production du MAC et/ou l'élaboration des signatures.

Selon une option de l'invention, l'étape de production du MAC est commune aux deux sens d'authentification, et porte sur l'ensemble des données de la session. Et dans le cas d'une cryptographie symétrique, le certificat authentifiant la carte et celui authentifiant le terminal sont obtenus par une seule étape de chiffrement du MAC, les certificats respectifs de la carte et du terminal s'en déduisant par une opération élémentaire telle qu'extraction de certains bits prédéterminés.

Exemple II

Dans cet exemple les données de la mémoire sont organisées en secteurs comportant chacun quatre champs :

1. données ;
2. identifiant (clé d'accès permettant de sélectionner un secteur) ;
3. pertinence : permet de déterminer le secteur pertinent si deux secteurs ont le même identifiant ;
4. contrôle : permet de vérifier l'intégrité des trois champs précédents (par exemple un contrôle de type parité).

Un secteur sera désigné par son identifiant, notion qui se substitue à celle d'adresse. La procédure d'écriture d'un secteur a comme paramètre un identifiant et des données à associer à cet identifiant. La procédure de lecture d'un secteur a comme paramètre un identifiant, et retourne les données associées à cet identifiant lors de la dernière écriture effectuée avec ce même identifiant (ou une indication appropriée si cet identifiant n'a jamais été utilisé). En d'autres termes, on réalise un accès de type associatif au lieu d'un accès indexé.

Lors de la procédure de lecture d'un secteur, la carte recherche les secteurs dont l'identifiant a la valeur demandée, et qui (sur la base du champ de contrôle) sont intègres. Au cas où plusieurs secteurs répondent à ces deux critères, elle en retient un sur la base du champ de pertinence.

Lors d'une écriture de secteur, la carte écrit, dans un secteur disponible, les champs données et identifiant demandés, le champ pertinence tel que ce secteur sera, pour la procédure de lecture, le plus pertinent des secteurs intègres possédant cet identifiant, et le champ contrôle en accord avec les trois champs précédents (en d'autres termes, l'écriture est gérée de manière que la lecture ultérieure puisse être correctement opérée).

Avantageusement, la procédure d'écriture se poursuit par l'effacement du secteur rendu non pertinent par l'écriture du nouveau secteur, créant ainsi un nouveau secteur disponible.

On prévoit avantageusement un système (complémentaire) de type *garbage collection*, c'est-à-dire de récupération des secteurs inutiles, qu'ils soient non intègres ou non pertinents.

On prévoit avantageusement un système qui répartit l'usure résultant de l'écriture en évitant d'utiliser toujours les mêmes secteurs, par

exemple en choisissant aléatoirement un secteur parmi les secteurs disponibles.

Une variante généralement avantageuse de la procédure de recherche de secteur consiste à profiter de cette étape de recherche pour effacer les secteurs dont il est déterminé qu'ils sont non intègres, et/ou ceux qui ne sont pas les plus pertinents, recréant ainsi des secteurs libres (cela perd du temps lors de cette lecture, en faveur de la vitesse des lectures et écritures ultérieures). Avantageusement, avant l'effacement d'un secteur dont on a déterminé qu'il est intègre mais non pertinent, on écrira à nouveau le secteur pertinent, dont l'écriture peut être imparfaite.

La taille utile de la mémoire est égale au nombre de secteurs disponibles, moins un secteur qui doit rester effacé. Tous les secteurs (y compris celui effacé) sont répartis dynamiquement dans la mémoire.

Si les données doivent être structurées en fichiers, par exemple selon la norme ISO/IEC 7816-4, l'identifiant de secteur se décompose en deux sous-champs, un identifiant de fichier et un identifiant du secteur dans ce fichier.

On va donner ci-dessous une implémentation (non limitative) des opérations de lecture/écriture utilisant cette structuration particulière en secteurs :

— Le champ de contrôle contient, codé en binaire, le nombre de bits à zéro dans les trois autres champs ; on montre que si un problème tel qu'une écriture ou un effacement interrompu modifie un nombre quelconque de bits du secteur tous dans le même sens, le contrôle de la valeur du champ de contrôle permet toujours la détection du problème.
— Le champ pertinence est un entier de 0 à 3, codé sur 2 bits.
— La procédure de lecture parcourt séquentiellement tous les secteurs jusqu'à trouver un premier secteur possédant l'identifiant recherché, et intègre. Si cette recherche ne trouve aucun secteur, on termine la procédure avec un compte-rendu "secteur non trouvé". Si on trouve un tel premier secteur, on mémorise sa position, ses données, et sa pertinence p. La recherche se poursuit. Si l'on dé-

tecte un second secteur possédant l'identifiant recherché, et intègre, on teste si sa pertinence q est le reste de la division entière de p+1 par 3 ; si oui, on écrit à nouveau le second secteur, on efface le premier et on retourne les données du second ; sinon, on écrit à nouveau le premier secteur, on efface le second et on retourne les données du premier. Si un second secteur n'est pas trouvé et si la pertinence du premier secteur est p=3, on efface ce secteur et on donne le compte-rendu "secteur non trouvé" ; dans les autres cas, on retourne les données du premier secteur trouvé.

— La procédure d'écriture commence comme la procédure de lecture ci-dessus. Si l'on a trouvé le secteur que retournerait la procédure de lecture pour l'identifiant fourni, on mémorise la position de ce secteur et sa pertinence p (qui vaut 0, 1 ou 2) ; si on ne l'a pas trouvé, on sélectionne un secteur libre (par la procédure ci-après) et on écrit dans ce secteur les champs identifiant, données, pertinence p=3 et contrôle, et l'on mémorise la position et la pertinence de ce secteur. Dans les deux cas, on poursuit en sélectionnant un secteur libre (par la procédure ci-après). On écrit dans ce secteur les champs identifiant, données, pertinence q (calculée comme le reste de la division entière de p+1 par 3) et contrôle. Puis on efface le secteur mémorisé.

— Pour la recherche de secteur libre, on initialise à zéro le nombre n de secteurs libres trouvés. On examine séquentiellement les secteurs. Pour chaque secteur, s'il est non vierge et non intègre, on l'efface et il devient vierge (contribuant ainsi à la *garbage collection* mentionnée plus haut) ; s'il est intègre et si sa pertinence est p=3, on l'efface (idem) ; s'il est intègre et si sa pertinence n'est pas p=3, alors on recherche dans la zone non encore parcourue un autre secteur intègre de même identifiant, et si l'on en trouve un on efface celui qui n'est pas pertinent, en procédant comme pour la lecture ; si à l'issue de ce processus le secteur est vierge, on incrémente le nombre n de secteurs libres trouvés, et l'on effectue le tirage aléatoire d'un entier de 0 à n-1 ; si cet entier est 0, on mémorise la position du secteur vierge. Quand tous les secteurs ont été parcourus, tous les secteurs non vierges sont intègres, il n'existe pas deux

secteurs de même identifiant, on connaît le nombre n de secteurs vierges, et l'on a mémorisé l'un d'eux choisi aléatoirement de manière équiprobable. Si aucun secteur libre n'est trouvé, la procédure d'écriture est interrompue.

On va maintenant indiquer la manière dont la carte peut gérer des sessions de modifications indivisibles avec une telle structuration particulière en secteurs.

Pour stocker les modifications indivisibles, la carte dispose dans la mémoire non volatile de N secteurs effacés (N correspondant au nombre de modifications indivisibles que l'on pourra effectuer au cours d'une même session). De plus, elle gère une zone de la mémoire non volatile (hors secteurs) dédiée à la gestion de session et appelée "descripteur de session".

Cet exemple d'implémentation ne comprend aucune authentification propre à la session.

On définit un descripteur de session, composé de 3 champs :

— Liste des références des secteurs indivisibles (LRSA).
— Valeur de contrôle de création de la liste des références des secteurs indivisibles (VCC).
— Valeur de contrôle de prise en compte de la liste des références des secteurs indivisibles (VCPC), qui permettra de savoir si l'on a ou non fermé une session).

Étape 0 : initialisation : avant le premier accès aux données depuis la dernière interruption de fonctionnement de la carte, par exemple au *reset* (remise à zéro), la carte doit faire en sorte que le descripteur de session soit effacé. Il y a plusieurs cas à considérer, selon l'état du descripteur de session :

— Il est totalement effacé : la carte le laisse en l'état.
— Il n'est pas totalement effacé, et la VCPC est correcte : la carte recherche et efface (si nécessaire) tous les secteurs rendus obsolètes par ceux écrits (parmi ceux référencés dans la liste), puis efface le descripteur de session.
— Il n'est pas totalement effacé, la VCPC est effacée ou incorrecte et la VCC est correcte : la carte efface les secteurs indiqués dans

la LRSA, puis efface le descripteur de session.

— Il n'est pas totalement effacé, la VCPC est effacée ou incorrecte et la VCC est effacée ou incorrecte : la carte efface le descripteur de session.

Étape 1 : ouverture de session : la carte recherche N secteurs effacés, puis note la liste de leur référence et sa VCC dans le descripteur de session (supposé effacé).

Étape 2 : en cours de session : la carte reçoit des commandes. Lorsque l'une d'elle provoque une ou plusieurs modifications indivisibles, les secteurs utilisés pour noter ces modifications sont ceux notés dans la LRSA, à concurrence de N secteurs modifiés.

Étape 3 : fermeture de session : pour fermer la session, la carte écrit la VCPC, qui assure que la LRSA et sa VCC ont été pris en compte. Ensuite, elle recherche et efface tous les secteurs rendus obsolètes par ceux écrits (parmi ceux référencés dans la liste). Enfin, elle efface le descripteur de session.

Si, en outre, la carte gère la ratification, la gestion des sessions comporte les modifications ci-après.

Étape 0 : initialisation : dans ce lui des cas où le descripteur de session n'est pas totalement effacé et la VCPC est correcte, la carte recherche et efface (si nécessaire) tous les secteurs rendus obsolètes par ceux écrits (parmi ceux référencés dans la liste), mais elle n'efface pas le descripteur de session.

Étape 1 : ouverture de session : la carte note en mémoire volatile qu'une session est ouverte. Si le descripteur de session n'est pas vierge, la carte signale que la session précédente n'a pas été ratifiée et peut même, en analysant la LRSA, indiquer quelles sont les données non ratifiées. Quoiqu'il arrive, elle ne modifie pas le descripteur de session.

Étape 2 : en cours de session : lors de la première commande avec modifications indivisibles, la carte efface le descripteur de session si nécessaire, recherche N secteurs effacés, puis écrit la LRSA et sa VCC.

Étape 3 : fermeture de session : la carte note en mémoire volatile

qu'aucune session n'est ouverte. Quoiqu'il arrive, elle n'efface pas le descripteur de session.

REVENDICATIONS

1. Un procédé pour modifier le contenu de la mémoire non volatile d'une carte à microcircuit, notamment d'une carte sans contact,

procédé dans lequel la carte est couplée temporairement à un terminal pendant l'exécution d'une transaction, notamment d'une transaction de télébillétique, comportant l'application par le terminal à la carte d'une pluralité de commandes de modifications comprenant chacune au moins une opération d'inscription, dans la mémoire de la carte, d'une information respective désignée par la commande, les différentes informations ainsi inscrites étant mutuellement interdépendantes,

procédé caractérisé en ce qu'il comprend l'exécution, par la carte, des étapes suivantes :

a) sur réception de commandes respectives correspondantes reçues du terminal, modifications du contenu de la mémoire de la carte par inscription provisoire, dans la mémoire de la carte, de chacune desdites informations interdépendantes sans perte de valeurs antérieures correspondant à ces informations ; puis

b) finalisation de ces modifications, soit en les confirmant toutes, soit en les infirmant toutes, de sorte que pour des opérations ultérieures les commandes exécutées à l'étape a) soient soit toutes prises en compte, soit toutes sans effet.

2. Le procédé de la revendication 1, dans lequel :

— en cas de confirmation à l'étape b), on inscrit dans la mémoire de la carte un témoin confirmatif de bonne exécution, et

— lorsque la carte reçoit ultérieurement une commande impliquant la lecture et/ou la modification de l'une au moins des informations inscrites à l'étape a) ou de la valeur y correspondant, la carte examine préalablement l'état du témoin et, si celui-ci n'a pas été inscrit, la carte ignore ou annule les inscriptions provisoires antérieurement opérées à l'étape a) et exécute la commande sur la base desdites valeurs antérieures correspondant aux informations.

3. Le procédé de la revendication 2, dans lequel, lorsque la carte

examine l'état du témoin, si celui-ci a été inscrit la carte exécute des opérations de recopie des écritures provisoires opérées à l'étape a).

4. Le procédé de l'une des revendications 1 et 2, dans lequel la carte est apte à fonctionner selon deux modes, à savoir :

— un mode en session, dans lequel les inscriptions sont opérées par exécution des étapes a) et b), et
— un mode hors session, dans lequel l'opération des inscriptions n'est pas confirmée à l'ensemble des étapes a) et b).

5. Le procédé de l'une des revendications 1 à 4, comprenant une fonction d'authentification combinée à la fonction de finalisation de l'étape b), forçant l'infirmation à l'étape b) dans le cas où l'authentification échoue.

6. Le procédé de la revendication 5, dans lequel ladite authentification est opérée par la carte qui authentifie le terminal et/ou les données échangées entre terminal et carte, la carte contrôlant un certificat cryptographique produit par le terminal et transmis à la carte et ne confirmant les modifications à l'étape b) que si ce certificat est reconnu correct.

7. Le procédé des revendications 4 et 6 prises en combinaison, dans lequel, lorsque la carte reçoit du terminal des commandes de modification du contenu de la mémoire incluant la vérification d'un certificat cryptographique, cette vérification est opérée si la commande est reçue hors session, et ne l'est pas si la commande est reçue en session.

8. Le procédé de la revendication 5, dans lequel ladite authentification est opérée par le terminal qui authentifie la carte et/ou les données échangées entre terminal et carte, la carte produisant et transmettant au terminal un certificat cryptographique de manière conditionnelle, si et seulement si les modifications ont été confirmées à l'étape b).

9. Le procédé des revendications 4 et 8 prises en combinaison, dans

lequel, lorsque la carte reçoit du terminal des commandes de modification du contenu de la mémoire incluant la production d'un certificat cryptographique, cette production est opérée si la commande est reçue hors session, et ne l'est pas si la commande est reçue en session.

10. Le procédé de l'une des revendications 1 et 2, dans lequel, lorsque la carte reçoit du terminal à l'étape b) des commandes de modification du contenu de la mémoire incluant la production d'une pluralité de certificats cryptographiques, ces certificats sont mémorisés à cette étape b), puis transmis ensemble au terminal si et seulement si les modifications ont été confirmées à l'étape b).

11. Le procédé des revendications 1 et 4 prises en combinaison, dans lequel au moins certaines des commandes susceptibles d'être exécutées à l'étape b) comprennent un éventuel attribut d'inhibition, et dans lequel, si la carte exécute une telle commande en session à une étape b), les modifications opérées par cette commande prennent effet indépendamment du résultat de l'étape b).

12. Le procédé de l'une des revendications 1 et 2, dans lequel il est en outre prévu, après l'étape b) et en cas de confirmation des modifications, la séquence d'étapes suivante :

d) exécution par le terminal d'une action suite à la confirmation par la carte ;

e) en cas de bonne exécution de ladite action par le terminal, inscription dans la carte d'une information de ratification ultérieurement accessible en lecture.

13. Le procédé de la revendication 12, dans lequel la commande d'inscription de l'étape e) est une commande implicite, toute commande reçue par la carte après l'étape b) étant interprétée comme un ordre d'inscription dans la carte d'une information de ratification.

[19]中华人民共和国国家知识产权局

[51]Int. Cl[7]
G07F 7/10

[12]发明专利申请公开说明书

[21]申请号 99804911.5

[43]公开日 2001年5月23日

[11]公开号 CN 1296601A

[22]申请日 1999.4.9 [21]申请号 99804911.5
[30]优先权
[32]1998.4.9 [33]FR [31]98/04453
[86]国际申请 PCT/FR99/00837 1999.4.9
[87]国际公布 WO99/53451 法 1999.10.21
[85]进入国家阶段日期 2000.10.9
[71]申请人 法商·英诺瓦特隆电子公司
地址 法国巴黎
[72]发明人 史蒂芬·狄戴尔
法兰柯伊斯·格雷优
[74]专利代理机构 上海专利商标事务所
代理人 吴蓉军

权利要求书2页 说明书9页 附图页数0页

[54]发明名称 以不可分割方式修改特别是非接触卡的微电路卡的非易失性存储器的多个位置的方法

[57]摘要

卡片在执行交易之时暂时耦合至终端机，包括终端机外加多个修改指令至卡片，各指令包含至少一记录作业在卡片存储器，指令规定的个别信息项目，通过该方式写入的各项信息彼此有交互关是。该方法包含由卡片执行下列步骤。a）当接收来自终端机的个别指令时，经由临时将有交互关是的各项信息记录在卡片存储器上，而临时修改卡片存储器但未丧失该项的对应先前数值；及然后b）经由全部确认或全部抛弃而结束修改，因此在随后作业中，步骤a）执行的指令已经列入考虑，否则全部不受影响。

ISSN1008-4274

00.10.16

权 利 要 求 书

1．一种修改微电路卡片的非易失性存储器内容的方法，特别为非接触性卡片，所述方法中，卡片在执行交易时暂时耦合至一终端机，特别为远端购票交易，交易包括终端机对卡片外加多个修改指令，各自包含至少一记录作业于卡片存储器记录所述指令标示的个别数据项，通过这种方式记录的个别数据项是彼此互不相干，其特征在于，

所述方法包含卡片执行下列步骤：

a）当接收来自终端机的对应个别指令时，经由临时于卡片存储器记录各所述独立信息项修改卡片存储器内容，而未丧失先前对应该等项的数值：及然后

b）结束修改，包括全部都确认或全部都被抛弃，而随后作业，在步骤 a）执行的指令全部都列入考虑，否则全部都无效。

2．如权利要求 1 所述的方法，其特征在于：

*在步骤 b)的确认情况下确认适当执行的标记记录在卡片存储器；及

*当卡片随后接收到指令要求在步骤 a）写入的至少一数据项或其对应数值待被读取及/或修改，则卡片开始检验标记状态，若尚未记录，则卡片忽略或取消先前在步骤 a）所作的临时记录且基于对应数据项的先前值执行指令。

3．如权利要求 2 所述的方法，其特征在于，当卡片检验标记状态时，若标记已经记录，则卡片执行拷贝步骤 a）所作临时写入的作业。

4．如权利要求 1 或 2 所述的万法，其特征在于，所述卡片适合以双模式作业，即：

*对话进行中模式，其中记录是通过执行步骤 a）及 b）进行；及

*结束对话模式，其中全部步骤 a ）及 b）所作的记录皆未获得确认。

5．如权利要求 1 至 4 项中任一所述的方法，其特征在于，包含一认证功能结合结束化步骤 b）的功能，在认证失败时迫使步骤 b）被抛弃。

6 如权利要求 5 所述的方法，其特征在于，所述认证是由卡片执行，其认证终端机及/或终端机与卡片间互换的数据，卡片检验终端机产生并传输至卡片的加密认可，且唯有在认可被辨识为正确时才确认步骤 b）的修改。

7，如权利要求 4 及 6 所述的方法，其特征在于，当卡片接收到来自终端机的指令要求修改存储器内容且包括加密认可证明；若指令是在结束对话后接收，则执行证明；若指令是在对话进行中接收，则未执行。

8. 如权利要求 5 所述的方法，其特征在于，所述认证是由终端机执行，其认证卡片及/或终端机与卡片间互换的数据，卡片是以有条件方式产生并传输加密认可至终端机，若且唯若所述修改已经在步骤 b）获得确认。

9. 如权利要求 4 及 8 所述的方法，其特征在于，当卡片接收到来自终端机的指令要求修改存储器内容且包括加密认可证明；若指令是在结束对话后接收，则执行证明；若指令是在对话进行中接收，则未执行。

10. 如权利要求 4 或 2 所述的方法，其特征在于，当卡片在步骤 b）接收到来自终端机的指令要求修改存储器内容且包括产生多个加密认可时，此等认可是存储在步骤 b)，然后共同传输至终端机，若且唯若所述修改已经在步骤 b)获得确认。

11. 如权利要求 1 及 4 所述的方法，其特征在于，至少部分可在步骤 b）执行的指令包括选择抑制属性，及其中若卡片在步骤 b）于对话进行中执行此种指令，则所述指令执行的修改是与步骤 b）的结果无关。

12. 如权利要求 1 或 2 所述的方法，其特征在于，进一步规定在步骤 5）之后且在修改已经确认后，执行下列步骤顺序：

d）终端机执行卡片确认后的动作；及

e）若所述动作由终端机适当执行，实证信息记录在卡片上适合供随后通过读取存取。

13. 如权利要求 12 所述的方法，其特征在于，所述步骤 e）的记录指令为内部指令，任何在步骤 b）之后卡片接收到的指令都被解译为命令记录实证数据在卡片上。

00.10.16

说　明　书

以不可分割方式修改特别是非接触卡的微电路卡的非易失性存储器的多个位置的方法

本发明关于微电路卡，特别是关于对本身的非易失性存储器执行多种修改的微处理卡。

执行交易时，存储器通常修改一次或多次，当然在利用新记录信息前必须确定全部修改都正确，当有错误或记录讹误时，新记录的信息必须被忽略或消除。

US-A-4 877 945 叙述在多个数据项写入过程中如何检测异常以防交易基于错误情况持续下去。

在异常情况下，也希望可恢复原状，换言之，随后交易是对卡片执行不正确交易前已经记录的信息值进行。

前述 US-A-4 877 945 无法提供该优点，原因为某些案例中，当执行不正确交易时老旧信息值已经丧失，故无法将信息回复至稍早状态，至少单纯基于卡片所合信息无法复原。

WO-A-89/02140 叙述一种作业方式，但仅应用于单项信息已被修改或多项以彼此独立方式修改的情况。

但许多案例中，在单一笔交易期间需要修改多个数据项，而此等多个数据项处理时须视为“彼此有交互关联”，从而确保多个数据项全部都作妥善修改。

使用”无接触”型卡片在此种情况下，卡片环绕终端机可正确作业的空间边界无法察觉时，多个关联数据项的交易不完美或不完全的风险特高。此种情况下，卡片在终端机间意外通信中断的风险无法忽略，可能由于卡片在处理结束前已经移离超过终端机的范围，或因暂时性的干扰例如有一块金属通过旁边。

一范例（当然为非限制性）为此种卡片用于遥控验票交易，例如在大众运输网的入口，此时卡片扮演两种角色：旅行车票；以及电子钱包角色。

曾经提出若干解决办法来缓和前述困难且作多个写入或其他修改至彼此相关的“无法划分的”数据项。

前文举例说明的特殊用途中已知系统始于由电子钱包扣帐，然后记录使用者取得旅行权利。若使用者在两次操作间抽出卡片，则要求使用者再度出示该卡片且重新开始写入旅行权利。但若使用者走开而未再度出示卡片，则使用者可能出

差错。显然无法以相反顶序进行，后来使用者尝试在钱包扣帐前抽取卡片是不可能。

该解决之道暗示终端机特别配置成在交易中断的情况下激发例外处理来重新开始交易(要求将卡片重新插入终端机)。除了终端机的软件特别复杂外，如所述该种解决之道并非全然满意，若未重新开始交易使用者仍出差错。

另一种解决之道在于数据交叉，终端机保留有关卡片的电子钱包状态信息，反之亦然。但该解决之道仍令人满意，原因为除了复杂以外也增加卡片与终端机间交换数据的容量，因而减慢交易的执行。当有多笔（三笔或三笔以上）写入无法划分时也难以应用。

本发明的一个目的是提议一种致能以无法划分方式对卡片存储器作多次修改的方法。

本发明的另一目的是提出可完全由卡片执行的方法。如此可执行该方法而未修改终端机且无需对终端机提供另外处理，该方法使用原有指令的语法，因此可选用的指令上极有弹性。

本发明方法属于一种卡片在执行交易之时暂时耦合至终端机，交易包括终端机对卡片外加多个修改指令，各指令包含至少在卡片存储器上记录由指令指定的个别数据项，通过这种方式记录的个别数据项彼此有交互关联。以本发明的特征方式，该方法包括由卡片执行下列步骤：a）当接收来自终端机的个别指令时，经由临时将有交互关系的各项信息记录在卡片存储器而临时修改卡片存储器但未丧失该项的对应先前数值；及然后 b)经由全部确认或全部抛弃而结束修改，因此在随后作业中，步骤 a）执行的指令已经列入考虑，否则全部不受影响。

如此本发明原理包含将待进行的多个修改以无法分割方式集合于单一步骤 a)，然后在修改已经执行后于卡片中整体确认此等修改。若确认成功，则其次由该卡片执行的操作(无论在相同交易期间或在随后交易期间)，存取的内容必然反映出已经作的修改。

相反地，在步骤 a）期间卡片进行操作有任何中断则将取消全部执行的修改，非易失性存储器中的数据保持在步骤 a）之前的状态。

特殊实务中，在步骤 b）的确认情况下，在卡片存储器上记录一标记证实已经适当执行；当卡片随后接收一指令要求步骤 a）写入的至少一数据项或其对应值被读取及／或修改时，卡片开始检视标记状态，若尚未记录标记，则卡片忽略或取消先前在步骤 a）所作的临时记录，并基于对应该数据项之先前数值执行该

等指令。若卡片检视标记状态时发现已经记录，则卡片执行拷贝在步骤 a）执行的临时写入作业。

最佳卡片适合以双模式作业，即对话进行中模式，其中经由执行步骤 a）及 b）作记录；及结束对话模式，其中步骤 a）及 b）都未确认作记录。

开启一段对话例如可以卡片复置为零表示或可于一指令具有两种动作之后，一者执行预定作业而同时被解译为开启一段对话。

例如当籍一证明无法完成通常核准记录时，卡片自动开启一段于此对话进行中处理记录。

以相同方式，结束对话例如可于一指令可执行两种动作之后：执行预定作业且同时被解译为关闭一段对话的指令。

例如电子钱包扣帐作业结束对话，因而避免须与所得的认可通信，如此使对话认可与电子钱包交易认可变成无法分割。

更佳该方法包含认证功能组合结束步骤 b)功能，在认证失败的情况下强迫步骤 b)被抛弃。

第一实务中，认证的执行方式是通过卡片进行其证实终端机的真实性及／或终端机与卡片间交换数据的真实性，卡片检查由终端机产生加密认可并传输至卡片，唯有在认可被确认为正确时才证实在步骤 b)的修改。

在对话模式过程中可作临时修改，故当卡片接收到来自终端机的指令要求修改存储器内容且包括证实密码认可时，证明该指令是否接收到结束对话，若指令是在对话进行中接收，则无需如此执行。

换言之，由卡片在步骤 b)执行的指令以及通常（亦即结束对话）的指令证实加密认可，当在对话进行中执行时，则不再包括此种确认，以“对话认可终端机的真实性”来执行相当的功能。

第二实务中，认证真实性是由终端机进行，其认证卡片的真实性及/或终端机与卡片间交换的数据的真实性，若且唯若在步骤 b）确认修改，卡片才以有条件方式产生并传输加密的认可至终端机。

在对话模式中，可作临时修改，故卡片接收来自终端机的指令用于修改存储器内容且包括产生加密认可，若指令是在结束对话后接收，则进行该作业，若指令是在对话进行中接收，则无需进行该项作业。

换言之，由卡片在步骤 b）执行的指令以及正常（亦即停止结束对话）产生加密认可的指令时，当是在对话进行中执行则不再产生此种认可，而以“执行认

可证实终端机真实性”来执行相当的功能。

可作临时修改，故当卡片在步骤 b）接收到来自终端机的指令要求修改存储器内容且包括产生多个加密认可时，该等认可存储在步骤 b），且最后共同传输至终端机，若且唯若修改已经在步骤b）获得证实。

换言之，规定由卡片展期通信通常由步骤 b)的命令产生的加密认可。特别若核可的写入指令产生某种写入认可，则希望唯有在写入已经不可变更的执行之后认可才离开卡片。

特定实务中，至少部分可能在步骤 b）执行的指令包括选择性抑制属性，及若卡片于步骤 b)在对话进行中执行此种指令，则该指令执行的修改是在步骤 b)的结果独立无关地执行。

换言之，该属性定义指令是在对话进行中执行(即若对话未完成则将被取消)，或在结束对话时执行(即，即刻有效仿佛已经完成对话，即使就时间上而言仍然处在对话进行中亦如此)。

最佳本发明在步骤 b）之后且在确认修改情况下进一步提供下列步骤顺序：d)终端机遵照卡片的确认执行某种动作；及 e）若该动作由终端机妥当执行，则实证信息记录在卡片方便随后通过读取存取。

此种“实证”对话通知卡片在执行对话之后终端机确实可采行决策（例如应用于大众运输网的进出口开启栅门)。

观察到此种实证是由卡片处理而无需额外写入（临时写入的拷贝则为迟早需要执行的作业)。此外，此种卡片在拷贝端执行的情况唯有当动作已经在终端机端适当执行的情况，换言之，唯有整个交易符合一致。

全部操作皆由卡片执行的情况下较佳规定步骤 b)的记录指令为内部指令，卡片在步骤 b）之后接收到的任何指令被解译为命令记录实证信息在卡片上。

其他特点及优点由后文说明本发明的二实施例显然易明。

此等实例中且确实于全文中，”指定”一词用以表示”特定多个中之一者”，且是有关于卡片所含的多个项目当中特征化一项特定信息的动作。

此种指定可能为内部者，由于指令本身规定一项特定信息，例如指令“由电子钱包中扣帐 x 量”指示存储器位在含有“电子钱包结馀”数据项数值。

指定也可为明确者，如下实例 I 所示，规定写入指令有一位址或扇区识别记号，指令是通过指标 I 加索引。

实例 I

提供一卡片其可存储 100 个 8-位元组债，且可执行下列命令：

*读取在 1 至 100 范围由指标 i 规定的 8-位元组值 v；

*写入在 1 至 100 范围由指标 i 规定的 8-位元组值 v；

*开启一段对话；

*结束一段对话。

卡片在一次对话进行中至多允许三次写入。习惯上使用大写字母来标示非易失性存储器（例如 EEPROM）的值而小写字母用于标示易失性存储器（RAM，其内容当未供电时即消失）的值。

非易失性存储器区段分配作业存储器的主要数据存储区（确定写入）；

*V [i] 对 1 至 100 范围的 i：100 x 8 位元组。

另一非易失性存储器区段分配给对话机制且包含：

*T[k]用于 1 至 3 的 j：3x8 位元组

含有一对话过程写入的值（临时写入）；

*I[K]用于 1 至 3 范围的 j：3x1 位元组

含有对话过程写入值的指标；及

*C：于对话结束时写入的计数位元组。

C 编码于对话过程中执行的写入次数；适当冗馀机制（例如结合该值的补数）可检测于计数位元组存储值不确定的案例。

作业如下进行。

步骤 0：在卡片供电与执行第一指令间的瞬间检验 C。若为 1 至 3 范围的确定值，则对 k 等于 1 至 C 而言，于指标 I[k]的值 T[k]由表 V[i]拷贝。随后 C 复置至零及内部变量 j 设定为-1(指示尚未开启对话)。

步骤 1：读取中，进行测试了解 j 是否大于 0：若是，则要求的指标 i 对 k 由 j 至 1 以递减顶序与 I[k]比较。若有匹配则送返 T[k]。所有其他案例则送返 V[i]。

步骤 2：在开启一段对话时，j 初始化为 0(若对话已经开启则取消)。

步骤 3：在各次写入时若 j=-1(对话未开启)，通信值 V 是于 T[0]写入，通信指称 i 是于 I[0]写入，写入 C=0 随后 v 于 V [i] 写入及写入 C=0：若 0≤j<3（于对话进行中写入），则 j 递增 1，v 于 T[j]写入，及 i 于 I[j]写入；若 j=3 则操作被拒绝（已经超过一次对话进行中的写入上限）。

步骤 4：结束对话进行中，若 j>0，则 j 于 C 写入，然后对 j 为 1 至 C，拷

贝于表 V[]于指称 I[j]的值 T[j]。然后 C 设定为 0 及 j 设定为-1。

显然卡片的电源可能于任一瞬间被中断，读取数值需正确，亦即对各指标 i 而言最末写入值非于对话进行中或写入已经结束的对话（在一个非零值写入 C 时，写入已经完成或对话已经结束）。

增加加密以防当供给卡片的加密认可不正确时出现某种作业，及/或在某个作业结束时对卡片产生加密认可。

使用的加密认可是基于已知的密码类型。例如“对话认可证实卡片的真实性”（或终端机）的获得方式是当开启对话时在卡片端及在终端机端对卡片（或终端机）供给的数据及/或对终端机（或卡片）供给的乱数应用安全杂散演算法（SHA）；由其中所得信息确认码(MAC)由卡片（或终端机）使用卡片（或终端机）所含密钥籍数位签章演算法（DSA）核章；终端机（或卡片）使用公钥证实签章。对称加密演算法例如数据加密标准也可用于产生信息确认码(MAC)及/或产生签章。

本发明中产生信息确认码的步骤为双向认证所共通，且载于全部对话的数据。当使用对称加密时，籍由认证卡片的认可及认证终端机的认可是由信息确认码编译密码的单一步骤获得，卡片及终端机的认可是藉单元操作例如撷取某个预定住元导出。

实例 II

本实例中，存储器的数据被组织作为扇区，各个扇区包含四栏位：

1. 数据；

2. 识别者（可选定某个扇区的存取钥）；

3. 合适性：若二节段具有相同识别者用于决定何扇区为适合；及

4. 检验：饶实前三个栏位皆未讹误（例如执行同位型检验）。

一扇区以其识别者标示，以此通知置换位置通知。扇区写入过程具有一识别者作为参数随同该识别者相关数据。读取扇区程序有个识别者作为参数，其送返末次使用该识别者（或若识别者未曾使用则为适当指示）执行写入时关联该识别者的数据。换言之，是执行关联型存取而非索引型存取。

读取一扇区过程中，卡片搜寻扇区其带有含要求值的识别者且非讹误（通过检验栏位决定）。当多个扇区可满足此两项标准时，基于适合性栏位保有一特定扇区。

当写入一扇区时，卡片将下列写于适当扇区：要求的数据；识别者；适当栏位故在读取过程中，此扇区为最适合的具有此种识别者的非讹误扇区；及一检验

栏位匹配前三栏位（换言之，写入的处理方式为随后可适当进行读取）。

较佳写入过程后，接着去除已经通过写入新扇区而变成不适合的扇区，如此可利用一新扇区。

较佳也提供（额外）资源回收型系统，亦即由一系统回收可能由于讹误或由于不适合而已经无用的扇区。

较佳提供一系统其可延迟因写入造成的磨耗，确保不会经常使用同一扇区，例如由可利用的多个扇区中随机运用一扇区。

搜寻一扇区程序的概略优异变化方法包括利用搜寻步骤来消除已经发现讹误的扇区及＊或并非最适合的扇区藉此重新产生自由扇区（放特定读取过程耗费时间对随后读取与写入速率有利）。较佳于消除一个已经发现为非讹误但非适合的扇区前，再度写入适合扇区原因为可能是写入不当所致。

存储器的工作量等于可利用的扇区数目减一个必须保持被消除的扇区。全部扇区（包括被消除扇区）是动态分布在存储器内部。

若数据待于档案结构化，例如应用 ISO / IEC 7816－4 标准结构化，则扇区识别者分成两栏位：档案识别者及档案内部的扇区识别者。

使用此种特定扇区的结构的写入/读取作业的非限制性实务列举如下：

后文说明使用此种特定扇区结构（非限制性）执行读写作业的说明：

*检验栏位以二元码含有其他三栏位的零位元数目；显然若有问题，例如中断写入或消除修改扇区内全部相同方向的位元数目，则检查检验栏位的数值可经常测知已经发生问题。

*适合栏位是以二位元编码的 0 至 3 的整数。

*读取程序循序议取全部扇区至找到第一扇区具有寻找的识别者且非讹误为止。若未找到任何扇区，则程序结束且报告“未找到扇区”。若找到第一扇区，其位置连同其数据及其适合性 P 存储。持续搜寻。若找到第二扇区具有寻找的识别者且非讹误，则测试其适合性 q 是否为 p+1 除以 3 整除的余数；若是，则改写第二扇区，消除第一扇区，及来自第二扇区的数据被送返；否则改写第一扇区，第一扇区被消除，送返来自第一扇区的数据。若未找到第二扇区且若第一扇区的适合性为 p=3，则此扇区被消除且报告“未找到扇区”；否则被送返的数据是来自找到的第一扇区。

*写入程序类似前述读取程序般开始。若找到先前存储的扇区其已经由读取程序通过特定识别者送返，则此扇区位置连同其适合性 p（等于 0、1 或 2）保留；

若未找到此种扇区，则选择一自由扇区（使用下述程序选择）及识别者、数据、适合性 p=3 及检验栏位写至该扇区，保留该扇区的位置及适合性。两种情况下经由选择一自由扇区（使用下述程序）继续进行。识别者、数据、适合性 P（计算为 P+1 除以 3 整除的余数），及检验栏位写入此扇区。随后若有任何先前存储的扇区则消除它。

为了寻找一自由扇区，找到自由扇区数目 n 初始化为零。循序检验扇区。对非空白且讹误的扇区则消除它使其变空白（如此促成前述资源回收）；若扇区非讹误及若适合性为 p=3 则消除它（也送到资源回收）；若该扇区非讹误及若其适合性非 p=3，则尚未扫描区段被搜寻是否有另一非讹误扇区具有相同的识别者：若找到一者则非适合属区被消除，如同读取程序般进行：若此过程结束时扇区为空白，则找到自由扇区数目 n 递增，随机整数是以 0 至 n−1 的范围取出；若整数为 0，则存储空白扇区位置。当全部扇区皆已经扫描时，全部非空白扇区会非讹误，并无任何两个扇区具有相同的识别者，空白扇区数目 n 为已知，及其中一者已经存储作为以相等机率方式的随机选择。若未找到自由扇区，则写入过程中断。

后文说明卡片使们此种特定扇区结构处理无法分割的修改对话的方式。

为了存储无法分割的修改，卡片于非易失性存储器有 n 个可利用的已经被消除的扇区（此处 n 对应于单一对话进行中可能需要作的无法分割的修改数目）。此外，卡片处理专用于处理一对话的非易失性存储器区段（不含于扇区），被称作“对话描述者”。

此种实务并无对话的特定认证。

对话描述者是以三栏位定义：

*无法分割扇区的参考表单(LRSA)；

*形成无法分割扇区参考表单的检验值(VCC)；及

*考虑无法分割扇区参考表单的检验值(VCPC)，用于发现对话是否结束。

步骤 0：初始化：自从最近中断卡片作业例如复置开始初次存取数据前，卡片必须确定对话描述者已经被消除。某些案例依据对话描述者的状态需列入考虑。

*完全消除：卡片保持未改变；

*未完全消除，及 VCPC 为正确：卡片搜寻及消除（若有所需）全部已经被写入而变化无用的扇区（来自参考表单），然后消除对话描述者；

*未完全消除，VCPC 被消除或不正确，及 VCC 完全正确：卡片消除 LRSA 所给扇区然后消除对话描述者；或

*未完全消除，VCPC 被消除或不正确，及 VCC 被消除或不正确：卡片消除对话描述者。

步骤 1：开启对话：卡片寻找 n 个被消除的扇区，然后将参考表单记录在对话描述者(假定被消除）的 VCC。

步骤 2：对话进行中：卡片接受指令。当其中一指令产生一或多个无法分割的修改时，用于记录此等修改的扇区是记录在 LRSA 至多高达全部 n 个修改扇区。

步骤 3：结束对话：为了结束对话，卡片写入 VCPC，其确保 LRSA 及其 VCC 已经列入考虑。随后搜寻并消除全部已经被写入而变化无用的扇区（来自参考表单）。随后消除对话描述者。

此外，若为处理实证的卡片，则对话处理包括下列修改：

步骤 0：初始化：在对话描述者未完全消除及 VCPC 为正确的情况下，卡片寻找并消除（若有所需）全都已经被写人而变无用的属区（来自参考表单），但未消除对话描述。

步骤 1：开启对话：卡片于易失性存储器记录对话被开启。若对话描述者非空白，则卡片指示前一对话尚未被实证，经由分析 LRSA 甚至指示哪个数据项尚未被实证。总而言之未修改对话描述者。

步骤 2：对话进行中：在第一指令带有无法分割的修改时，若有所需卡片消除对话描述者，搜寻 n 个被消除的扇区，然后写入 LRSA 及其 VCC。

步骤 3：结束对话：卡片在易失注存储器记录无开启的对话。无论如何不消除对话描述者。