MASTER EN SCIENCES INFORMATIQUES

La carte à micro-calculateur multi-émetteurs

Peeters, André

Award date:
1991

Awarding institution:
Universite de Namur

FACULTES
UNIVERSITAIRES
N.D. DE LA PAIX

NAMUR

INSTITUT D'INFORMATIQUE

La carte à micro-calculateur multi-émetteurs

Description et comparaison des cartes TB100 de BULL CP8 et MCOS de Gemplus

André Peeters

Promoteur :
Professeur J. RAMAEKERS

Mémoire présenté en vue
de l'obtention du titre
de Licencié et Maître
en Informatique

Année académique 1990-1991

Résumé

La carte à micro-calculateur est une carte à puce équipée d'un composant actif. Il en existe deux types : mono-émetteur et multi-émetteurs.

Les informations stockées dans la mémoire d'une carte mono-émetteur sont protégées par son système de sécurité unique. Ce type de carte ne permet donc pas de sécuriser des applications indépendantes.

La carte multi-émetteurs peut être émise en commun par plusieurs prestataires de services. Chacun réserve une partie de la mémoire qu'il peut gérer et protéger indépendamment des autres applications résidant dans la carte. Ce texte présente l'analyse et la comparaison des cartes multi-émetteurs TB100 de BULL CP8 et MCOS de Gemplus.

Abstract

The micro-calculator card is a chip card equipped with an active component. Two types can be distinguished : single purpose and multi-purposes cards.

The information gathered in the memory of a single-purpose card is protected by its unique security system. So, it is not possible to secure independent applications with this sort of card.

The multi-purposes card can be issued by several services administrators together. Each one reserves a part of the memory which he can manage and protect regardless of the other applications written down in the card. This text sets out to analyse and compare the multi-purposes cards TB100 of BULL CP8 and MCOS of GEMPLUS.

J'exprime toute ma gratitude à Monsieur le Professeur Ramaekers, promoteur de ce mémoire, et à son assistante Madame Cocirdan, pour l'attention qu'ils ont portée à la réalisation de ce travail.

Je voudrais rendre un dernier hommage à Monsieur Peter Schnabel, rapporteur externe de ce mémoire, qui nous a quittés au mois de janvier dernier.

Je remercie vivement Monsieur Michel Hazard, Responsable de l'unité carte chez BULL CP8, pour son accueil chaleureux au sein de son équipe.

L'élaboration du contenu de ce mémoire aurait été impossible sans l'appui efficace de mes collègues au cours de mon stage. Je remercie particulièrement Madame Corinne Perret et Monsieur Jean-Michel Desjardins pour les très nombreuses explications qu'ils m'ont données, ainsi que pour leur sympathie.

Enfin, j'exprime ma reconnaissance à Monsieur Jacques Thielen, conseiller scientifique, pour ses précieux conseils.

Namur, le 31 août 1991.

André Peeters

Table des matières

Chapitre 2 :Description de la carte TB100

et de ses modules de sécurité associés 7

Introduction

Depuis une dizaine d'années, la diffusion à grande échelle de la carte à bande magnétique dans diverses activités a apporté de nombreux avantages : une diminution du traitement de documents, un accroissement de la vitesse des opérations, une gestion automatisée par l'informatique, et surtout une amélioration de la sécurité.

Malgré cet apport de sécurité, la fraude reste importante (0,28 % du chiffre d'affaires des cartes bancaires en France, soit environ 600 millions de francs français par an [JOLY 1990]). De nombreux domaines sont touchés par ce problème et la carte à microcalculateur doit permettre d'y remédier. De par sa conception monolythique, le microcircuit garantit une protection physique optimale des données qu'il contient. Il offre une plus grande capacité que la bande magnétique, et son microprocesseur contrôle lui-même tous les accès à la mémoire de la carte. Il renferme en effet un véritable calculateur.

Les premières cartes à microcalculateur sont mono-émetteurs : l'ensemble des données de la carte est protégé par un système de sécurité unique, configuré par l'émetteur. Dans une telle carte, il n'est pas possible de faire cohabiter plusieurs applications indépendantes du point de vue de la sécurité logique. L'utilisateur est donc souvent en possession de plusieurs cartes destinées à des usages différents et doit connaître plusieurs codes secrets. Le coût total d'émission de ces nombreuses cartes est très élevé.

Plusieurs constructeurs ont maintenant développé des cartes multi-émetteurs permettant à plusieurs prestataires d'émettre en commun une carte dans laquelle chacun se réserve une partie de la mémoire qu'il peut gérer et protéger indépendamment des autres applications. Avec une telle carte, l'utilisateur peut par exemple réaliser des transactions bancaires, accéder à une piscine, payer son parking, téléphoner et transporter une partie de son dossier médical.

L'objectif de ce mémoire est d'analyser et de comparer deux cartes multi-émetteurs : d'une part la carte TB100 de BULL CP8 et d'autre part, la carte MCOS de GEMPLUS. Ce travail m'a été confié au cours du stage effectué chez BULL CP8 pendant les cinq premiers mois de cette année académique. Ce stage a débuté par une formation

technique sur l'utilisation de la carte multi-émetteurs de BULL CP8. J'ai ensuite disposé de documents internes à la société et des outils nécessaires à la manipulation des cartes. Nous avons également réussi à obtenir quelques cartes et données techniques de la firme Gemplus.

Le premier chapitre est consacré à la présentation des différents types de cartes existant sur le marché.

Le second chapitre propose au lecteur une description détaillée de la carte TB100 de BULL CP8. Le rôle des différents types de modules de sécurité présents dans les terminaux et serveurs d'un système utilisant cette carte est également abordé dans ce chapitre.

La carte MCOS de Gemplus est l'objet du troisième chapitre. Sa description, aussi complète que possible, est basée sur les documents que nous avons pu obtenir et sur les essais effectués sur cette carte.

Le quatrième chapitre est consacré à la comparaison des deux cartes étudiées. Cette comparaison n'est pas relative à un besoin précis, elle tient compte de tous les aspects propres aux cartes à microcalculateur multi-émetteurs.

Chapitre 1 : Evolution des cartes

1.1. Historique de la carte

La carte à microcalculateur a un ancêtre : c'est dans les années 40 que la première carte de crédit fait son apparition aux Etats-Unis. Elle n'est introduite en Europe qu'en 1966 avec la Barclaycard en Angleterre. Depuis, de nombreuses banques l'ont adoptée, d'abord en Angleterre, puis ailleurs en Europe.

Aujourd'hui, on évalue le nombre total de cartes plastiques, toutes applications et toutes technologies confondues, à environ un milliard. Cet ensemble comprend environ 300 millions de cartes VISA et MASTERCARDS, et approximativement 500 millions de cartes non bancaires [Bright 1988].

Les premières cartes supportaient uniquement des informations embossées et la signature du porteur. La sécurité fut ensuite renforcée par des procédés de gravure, puis par l'adjonction d'une bande magnétique, et enfin par l'ajout d'une puce.

1.2. Typologie des cartes

Toutes les cartes sont constituées d'un support plastique conforme aux normes ISO 2894 et 3554. Ces normes spécifient entre autres le format de la carte : 85,60 mm x 53,98 mm x 0,76 mm, souvent appelé format "carte de crédit".

1.2.1. La carte embossée

Sur une telle carte, les informations sont simplement embossées dans le plastique et ne sont donc pas protégées en lecture. Sa capacité est très limitée et sa reproduction est aisée. Le porteur doit en général signer sa carte afin de permettre aux prestataires de services de l'authentifier.

Ce procédé est utilisé sur les cartes de crédit (en plus de la bande magnétique et/ou de la puce) sur lesquelles sont embossées les informations identifiant le porteur. C'est le cas en France où la plupart des commerçants disposent rarement d'un terminal ON-LINE et utilisent une presse pour imprimer les données embossées, le porteur s'authentifiant par sa signature.

1.2.2. La carte à piste magnétique

La présence d'une piste magnétique au dos de la carte renforce la protection de l'information (les données peuvent être chiffrées) et facilite le traitement informatisé. Les inconvénients de cette carte restent nombreux :

- sa faible capacité,
- son rôle passif,
- sa faible protection : il est facile de lire, d'écrire et donc d'altérer les informations, voire même de dupliquer la carte.

Il faut cependant apprécier son faible coût, avantage qui lui assure encore de nombreuses années d'existence.

1.2.3. La carte à microcircuit

La carte à microcircuit ou carte à puce est équipée d'un composant électronique dont la capacité est nettement supérieure aux supports précédents. On distingue trois types de cartes à microcircuit.

1.2.3.1. La carte à mémoire simple

La puce est ici une simple mémoire dont l'accès en lecture et en écriture est libre. Ses avantages par rapport à la bande magnétique sont d'une part une plus grande capacité et d'autre part une reproduction moins aisée. Cette carte reste cependant passive.

Un exemple d'utilisation est la télécarte qui contient en mémoire un certain nombre d'unités que le publiphone brûle au fur et à mesure de son utilisation.

1.2.3.2. La carte à logique câblée

L'ajout d'un circuit câblé permet de protéger l'accès à la mémoire par un ou plusieurs codes secrets. La sécurité de cette carte est donc renforcée mais ses fonctionnalités sont figées et pré-déterminées à la fabrication, ce qui impose une modification du composant et donc un surcoût pour chaque nouvelle application.

1.2.3.3. La carte à microcalculateur

L'accès à la mémoire est ici contrôlé par un microprocesseur. Ce système mémorise les événements qui surviennent au cours de la vie de la carte et en tient compte dans son comportement afin d'éviter toute fraude logique. La mémoire ROM du microcircuit contient un véritable système d'exploitation qui permet d'exécuter un ensemble d'instructions éventuellement chiffrées.

La sécurité physique de ce composant est garantie par sa structure monolythique. Le microprocesseur et ses mémoires sont en effet intégrés dans une seule puce, ce qui rend le composant tout à fait inviolable. Des bits témoins le protègent contre toute attaque par rayons ultra-violets.

1.2.4. Autres cartes

1.2.4.1. La carte laser

Créée en 1981, cette carte est schématiquement composée d'une couche réflective et d'une couche non réflective protégées de part et d'autre par une feuille plastique. L'écriture d'un bit consiste à brûler une partie de la surface réflective à l'aide d'un rayon laser. La lecture est réalisée à l'aide d'un émetteur et d'un récepteur de laser, un bit à '1' ne renvoyant pas le rayon au récepteur.

Une telle carte est dotée d'une grande capacité de mémorisation (2 Mb), son prix est relativement bas, mais elle reste passive. Elle est par exemple utilisée comme dossier portable, ou remplace les bandes perforées des machines outils [Bright 1988].

1.2.4.2. La carte à infra-rouge

Cette carte fonctionne selon le même principe que la carte laser. Elle est moins chère mais offre une moindre capacité.

1.3. Les constructeurs de cartes à microcalculateur

1.3.1. BULL CP8

En France, l'histoire de la carte à mémoire commence en 1974 lorsque l'inventeur Roland Moreno dépose son premier brevet portant sur l'apport de dispositifs de sécurité dans les objets portables à mémoire. Il rencontre en 1975 des ingénieurs du groupe Bull qui dispose déjà des techniques nécessaires au développement et à l'industrialisation de son invention. En 1977, BULL décide de se lancer dans la recherche et le développement de cette technologie avec la mise en place d'une équipe de chercheurs et d'ingénieurs qui seront les pionniers de BULL CP8 (Portable Computer of the '80s'). Michel Ugon (actuel directeur de la recherche et du développement de BULL CP8) va donner une nouvelle orientation à ces travaux en proposant d'utiliser non pas une simple mémoire, mais un véritable microcalculateur capable de traiter l'information contenue dans l'objet portable.

Depuis, 65 brevets sur la carte à microcalculateur ont été déposés par BULL CP8. En 1989, la société a réalisé un chiffre d'affaires de 251 millions de francs français.

1.3.2. GEMPLUS

La société Gemplus Card International a été créée en 1988 par une équipe de SGS-THOMSON Microelectronics. Elle emploie aujourd'hui 240 personnes et a réalisé un chiffre d'affaires de 220 millions de francs en 1990.

1.3.3. Autres constructeurs

NTT DATA, SCHLUMBERGER, IBM, PHILIPS.

Chapitre 2 : Description de la carte TB100 et de ses modules de sécurité associés

2.1. La carte TB100

BULL CP8 vend deux types de cartes multi-émetteurs : TB10 et TB100. Elles diffèrent principalement par leur type de mémoire programmable.
La carte TB100 est équipée d'un composant à mémoire EEPROM. Sa structure physique et logique ainsi que ses fonctionnalités sont décrites tout au long de ce chapitre.
La carte TB10 est munie d'un composant à mémoire EPROM. Sa structure physique et logique est similaire à celle de la carte TB100, mais ses fonctionnalités n'offrent pas les instructions d'effacement de données et de signature de message existant dans TB100.

2.1.1. Présentation physique

La carte se présente extérieurement comme une carte de crédit en plastique conforme aux normes ISO 2894 et 3554. Le composant est inséré dans la partie gauche du recto du support. Il est recouvert d'un bouton de contact assurant l'interface physique avec le monde extérieur.

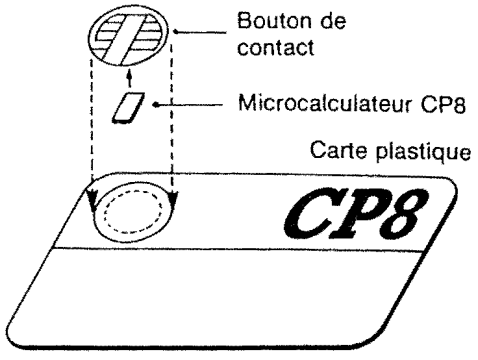
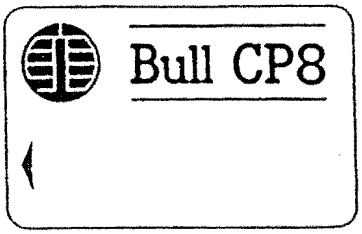


figure 1 : Présentation physique de la carte TB100

L'ensemble peut être implanté en position haute



ou basse:

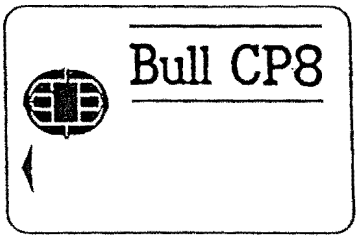


figure 2 : Les deux positions d'encartage

Le composant est un Microcalculateur Autoprogrammable Monolythique, ou MAM (en anglais Self Programmable One-chip Micro-computer, ou SPOM). Il s'agit d'un microcalculateur puisque le microcircuit contient un microprocesseur; il est autoprogrammable car il programme lui-même sa mémoire; et est monolythique étant donné qu'il regroupe sur un composant unique les éléments du schéma ci-dessous.

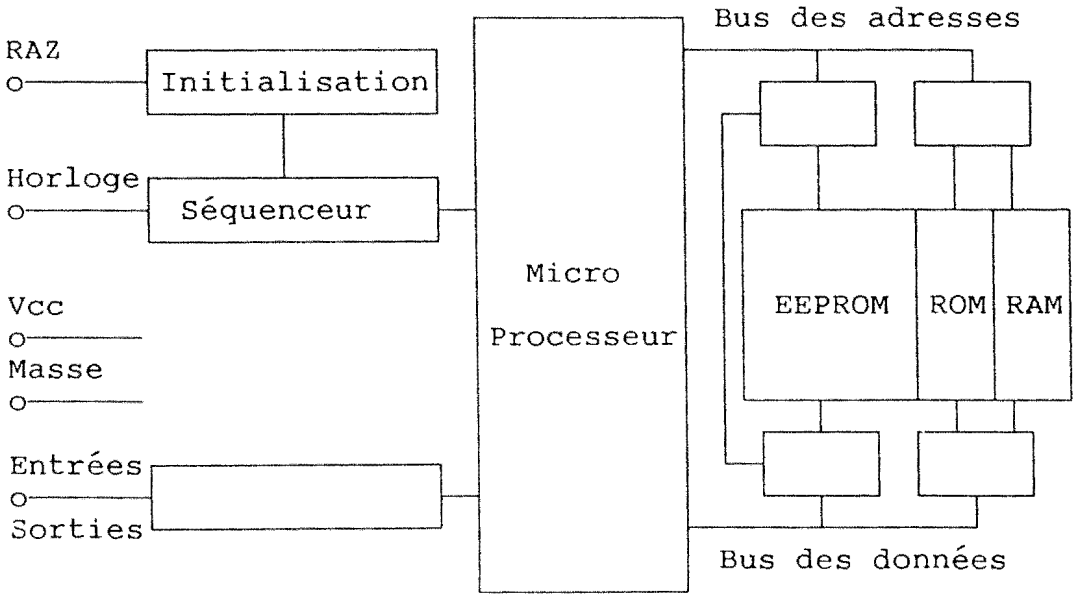


figure 3 : Composants du MAM d'une carte TB100

Le microprocesseur (MOTOROLA 6805 8 bits) exécute le programme (souvent appelé masque) mémorisé en ROM, il contrôle tous les accès à la carte.
La mémoire ROM (6 Koctets) contient le programme inscrit par 'masquage' lors de la fabrication du composant. Elle est inaccessible de l'extérieur, ce qui la rend non altérable et non duplicable. Ce programme remplit les fonctions suivantes :

- la gestion des entrées/sorties,
- la gestion de l'EEPROM,
- l'exécution de l'algorithme DES (utilisé par les instructions chiffrées),
- l'exécution des instructions d'accès au composant par le monde extérieur.

La mémoire RAM (128 octets) est la zone de travail du microprocesseur. Elle contient des informations temporaires.
La mémoire EEPROM (3 Koctets) est décrite dans le paragraphe suivant.

Cinq contacts relient le composant au monde extérieur:

- Vcc : tension d'alimentation de 5 V.
- RAZ (Remise à Zéro) : une tension appliquée sur ce contact provoque l'initialisation du composant. Cette opération est nécessaire avant chaque nouvelle session avec la carte.
- l'horloge : ce contact reçoit une base de temps de 3,57 MHz permettant de synchroniser le microprocesseur.
- la masse
- la ligne d'entrée/sortie : ligne série à 9600 bits/s.

Notons que la plupart des autres types de composants utilisés pour les cartes à microcalculateur sont munis d'un contact supplémentaire : Vpp (tension de programmation). Une tension y est appliquée lors de chaque écriture ou effacement.
Ce contact est supprimé sur le composant de la carte TB100, il est remplacé par un système interne de capacité permettant de fournir la tension requise.

2.1.2. Organisation de la mémoire EEPROM

L'organisation de la mémoire EEPROM reflète l'aspect multi-émetteur de la carte TB100. Elle offre en effet une structure multi-répertoires organisée selon une hiérarchie arborescente à trois niveaux : les niveaux carte, application et service.
Cette structure est comparable à l'organisation des fichiers d'un système d'exploitation dans lequel il serait possible de créer un répertoire racine et deux niveaux de sous-répertoires.

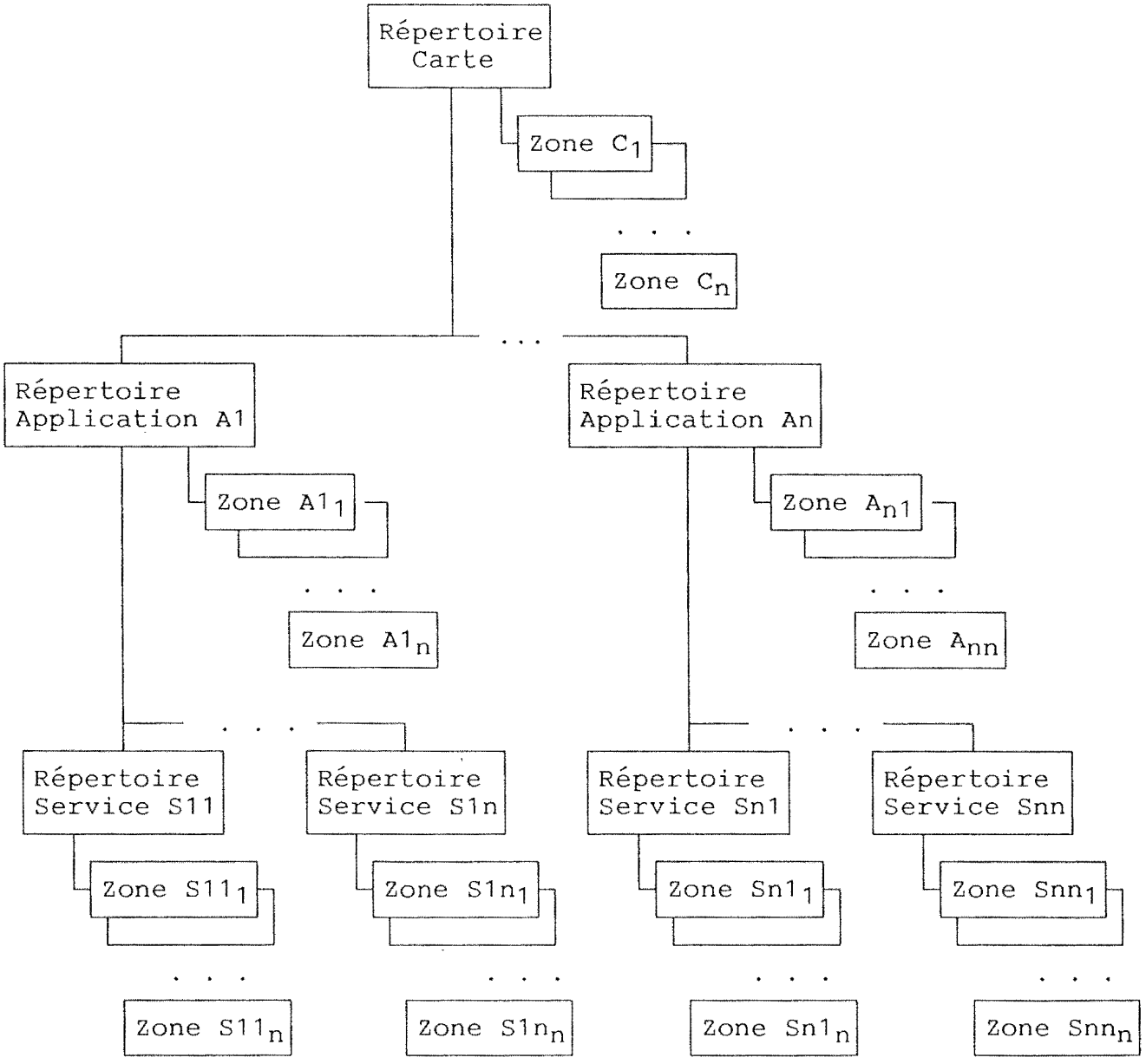


figure 4 : Organisation de l'EEPROM de la carte TB100

Le répertoire carte (ou racine) est créé et configuré par l'émetteur de la carte. Il est obligatoirement pourvu d'un système de sécurité qui contrôle l'accès aux informations

mémorisées dans les différentes zones de ce répertoire et la création des zones ou des répertoires de niveau directement inférieur.

L'émetteur de la carte attribue des sous-ensembles de la carte à des gestionnaires (émetteurs) de niveau application. Ces derniers peuvent à leur tour en céder une partie à des gestionnaires (émetteurs) de niveau service.
Les répertoires applications et services ainsi créés peuvent être dotés de leur propre système de sécurité, ils sont alors autonomes. Si ce n'est pas le cas, ils sont protégés par le système de sécurité du répertoire auquel ils appartiennent.

Chaque répertoire, quel que soit son niveau, peut contenir quatre types de zones (fichiers). Les deux premiers contiennent des données de contrôle exploitées par le microcalculateur. Les deux derniers contiennent des données de transaction exploitées par l'extérieur de la carte.

1) Les zones secrètes contiennent chacune une clé ou un code secret. Leur écriture est protégée. Leur accès en lecture et en effacement est impossible.
   Un répertoire créé avec son propre système de sécurité contient autant de zones secrètes que de clés et de codes.
2) Une zone d'accès est créée dans chaque répertoire doté de son propre système de sécurité. Elle conserve une trace des présentations (réussies ou non) des clés et des codes secrets appartenant au répertoire concerné afin d'empêcher un éventuel fraudeur de les rechercher exhaustivement. Une zone d'accès est accessible en lecture et interdite en écriture.
3) Les zones publiques contiennent des informations générales, libres d'accès en lecture. L'écriture dans une telle zone est protégée, l'effacement y est interdit.
   On y trouve par exemple des renseignements concernant le porteur de la carte ou le gestionnaire du répertoire.
   Chaque répertoire peut contenir un nombre variable de zones publiques.
4) Les zones de transactions contiennent des informations qui évoluent au cours de la vie de la carte. Leur accès en lecture, en écriture et en effacement est contrôlé par le système de sécurité.

Le nombre de zones de transactions créées dans un répertoire est variable.
La zone de transactions d'un répertoire réservé pour une application bancaire permet par exemple de mémoriser les informations relatives à chaque transaction financière effectuée par le porteur.

Le nombre de zones ou de sous-répertoires (répertoires de niveau directement inférieur) appartenant à un répertoire n'est limité que par la taille de celui-ci.

Nous parlerons de domaine pour désigner un répertoire ou une zone. Il s'agit en d'autres termes d'un noeud de l'arborescence. Chaque domaine est constitué d'un descripteur et d'un corps.

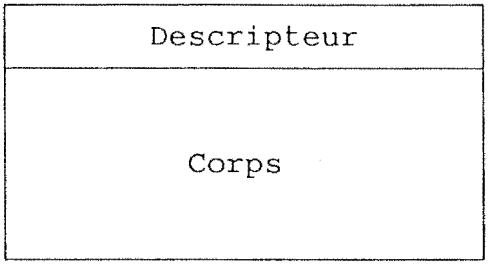


figure 5 : Mapping d'un domaine de la carte TB100

Le descripteur contient les informations spécifiant le mode de gestion et de sécurité du domaine.
Le corps d'un répertoire contient des zones et éventuellement des répertoires de niveau inférieur. Le corps d'une zone renferme des données propres au type de zone concerné.
L'adressage des données contenues dans le corps d'un domaine est relatif au début du corps de ce domaine: l'adresse du premier octet suivant le descripteur vaut donc $'0000'_H$.

L'ensemble des domaines est écrit dans la mémoire utilisateur sous forme de blocs imbriqués au fur et à mesure de leurs créations. La figure ci-après représente le mapping d'une EEPROM.

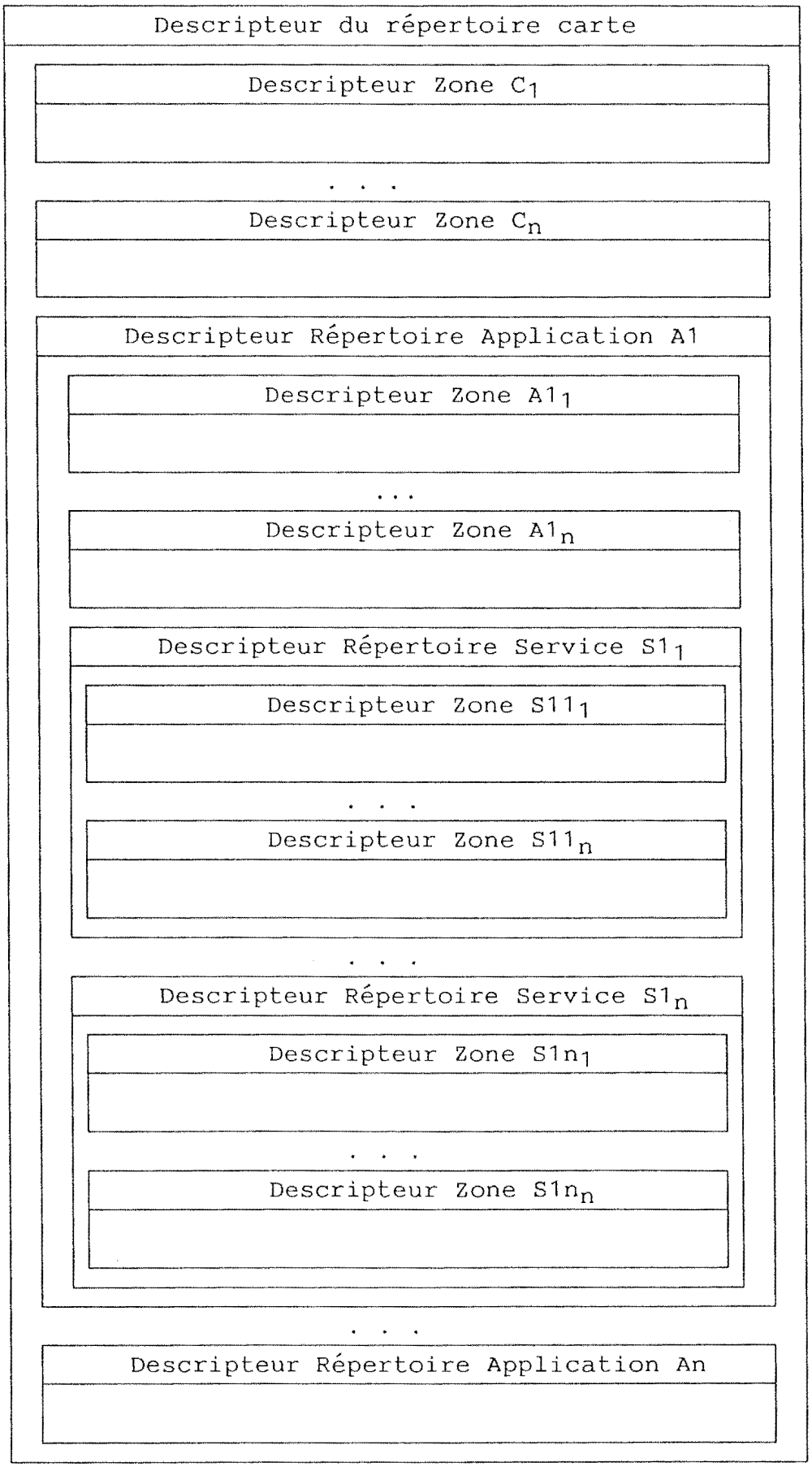


figure 6 : Mapping de l'EEPROM de la carte TB100

2.1.3. La diversification des clés

Objectif:

Le système de sécurité de chaque répertoire autonome est composé de différents types de clés secrètes. Si toutes les cartes diffusées par un émetteur contiennent des clés identiques, la sécurité globale du système peut être affectée de deux manières :

- il est possible de réutiliser les messages échangés avec l'une de ces cartes pour simuler un dialogue avec une autre carte du système. Ce type de fraude peut néanmoins être évité si les messages sont chiffrés avec un nombre aléatoire.
- la divulgation des clés de l'une des cartes permet de connaître les clés de l'ensemble des cartes diffusées.

Pour éviter ce type de fraude, il suffit d'affecter à chaque carte un jeu de clés unique dans le système. Dans un tel système, les clés sont dites diversifiées.

Principe de la diversification:

Afin de ne pas devoir gérer et conserver des listes importantes de clés uniques, et par mesure de sécurité, la valeur de chaque clé diversifiée résulte d'un calcul. Une clé diversifiée est fonction d'une clé de base (clé maître), unique pour le système considéré et connue uniquement par l'émetteur du répertoire concerné, et d'une information caractéristique de la carte. Cette information caractéristique est appelée le diversifiant. Il s'agit par exemple du numéro de série de la carte.
Le calcul de la clé diversifiée est exprimé par la formule suivante, où F est l'algorithme de diversification :

Clé Diversifiée := F(Clé de Base , Diversifiant)

Utilisation des clés diversifiées :

Lors d'une transaction mettant en oeuvre une telle clé, le monde extérieur doit recalculer la clé diversifiée. Ceci est possible après avoir extrait le diversifiant de la carte et à condition de connaître la clé de base.

2.1.4. Le cycle de vie d'une carte

Nous avons cité les principaux éléments contenus dans l'EEPROM d'une carte TB100. Avant de les détailler, voyons d'abord le cycle de vie de la carte.

a) La fabrication du composant

Lors de la fabrication du composant, le code exécutable (programme) est implanté dans la ROM par procédé de 'masquage'. Tous les composants issus de cette étape sont identiques.
Plusieurs informations sont ensuite écrites dans l'EEPROM sous pointe :

- la clé de fabrication diversifiée,
- le numéro et la version de la clé de fabrication maître,
- le numéro de diversification,
- l'identifiant de l'encarteur. Ce numéro repère le responsable de l'étape suivante : l'encartage.

La clé de fabrication est diversifiée à partir d'une clé de fabrication maître identifiée par son numéro et sa version, le diversifiant étant le numéro de diversification.

b) L'encartage

Cette étape consiste à assembler le composant et le bouton de contact sur le support plastique. Aucune donnée n'est inscrite dans la carte pendant cette phase.

c) Les tests et la pré-personnalisation

Le composant implanté sur la carte est d'abord testé, puis, à condition d'avoir présenté la clé de fabrication correctement, les informations suivantes sont écrites dans l'EEPROM :

- la clé de personnalisation diversifiée,
- le numéro de série de la carte,
- le numéro et la version de la clé de personnalisation maître,
- l'identifiant du personnalisateur. Ce numéro repère le responsable de l'étape suivante : la personnalisation.

La clé de personnalisation est diversifiée à partir de la clé de personnalisation maître identifiée par son numéro et sa version, le diversifiant étant le numéro de série de la carte.

Le verrou de fabrication est ensuite inscrit pour marquer la fin de la phase de pré-personnalisation. La carte est alors en phase de personnalisation. Elle est sous contrôle de la clé de personnalisation, la clé de fabrication n'est plus opérationnelle.

d) Le personnalisation

Cette étape consiste à créer le répertoire carte et son système de sécurité minimal.
Les données minimales à inscrire pendant cette phase sont :

- le descripteur du répertoire carte,
- la clé de l'émetteur de niveau carte,
- le code confidentiel PIN (Personnal Identification Number),
- le descripteur de la zone d'accès de niveau carte.

La clé de l'émetteur de niveau carte et le PIN sont chacun écrits dans des zones secrètes créées au niveau carte.
Les zones et les répertoires de niveau inférieur connus à ce stade de la vie de la carte peuvent également être créés lors de la personnalisation.
Le verrou d'utilisation est ensuite écrit afin de marquer la fin de la personnalisation. La carte est alors en phase d'utilisation. La clé de personnalisation n'est plus opérationnelle. Toute action dans la carte est alors contrôlée par le système de sécurité du répertoire carte ou par les systèmes de contrôle des autres répertoires éventuellement créés.

e) La phase d'utilisation

La carte est enfin délivrée au porteur qui peut l'utiliser sur tout le réseau de terminaux disponibles pour les applications concernées.
Pendant cette phase, de nouveaux répertoires et de nouvelles zones peuvent être créés sous le contrôle des systèmes de sécurité des différents répertoires qui les accueillent.

f) La fin de vie de la carte

Une carte meurt

- lorsque la mémoire EEPROM est saturée ou

- lorsque le répertoire carte est invalidé par une instruction spécifique.

2.1.5. Le système de sécurité d'un répertoire

Le système de sécurité d'un répertoire contrôle l'accès aux données mémorisées dans ses zones, et la création de domaines de niveau inférieur. Il est composé

- des clés et codes mémorisés dans ses zones secrètes,
- de sa zone d'accès,
- des paramètres écrits dans son descripteur et dans les descripteurs de ses zones de transactions.

Le répertoire carte est obligatoirement muni d'un système de sécurité.

Un répertoire application créé sans système de sécurité est protégé par celui du niveau carte.

Un répertoire service créé sans système de sécurité est protégé par celui du répertoire application auquel il appartient si ce dernier est autonome ou, dans le cas contraire, par celui du répertoire carte.

Nous verrons qu'il existe une instruction de sélection de domaine qui permet de sélectionner un noeud de l'arborescence de la carte.

Le répertoire actif est

- soit le dernier répertoire sélectionné s'il est autonome,
- soit le répertoire dont le système de sécurité protège le dernier répertoire sélectionné si ce dernier n'est pas autonome.

Les clés et les codes d'un répertoire :

Le système de sécurité de chaque répertoire autonome contient les clés et codes secrets suivants :

- une clé d'émetteur primaire IK (Issuer Key),
- un éventuel code porteur additionnel AID (Alternate Identification),
- de 0 à 4 clés d'émetteurs secondaires SK (Secondary Key, 16 versions possibles pour chacune),
- de 0 à 4 clés d'authentification AK (Authentification Key, 16 versions possibles pour chacune),

- de 0 à 4 clés d'effacement EK (Erase Key, 16 versions possibles pour chacune),
- 0, une ou plusieurs zones secrètes MAC (Message Authentification Code). Une telle zone contient les informations nécessaires à la génération des clés mises en oeuvre dans le calcul de signature de message.

Ces clés et codes secrets peuvent être exploités lorsque le répertoire auquel ils appartiennent est actif.
Le résultat de la présentation de ces clés et codes est mémorisé dans la zone d'accès du répertoire actif. Les autorisations qu'ils délivrent ne sont valables que dans le répertoire auquel ils sont rattachés.

Le système de sécurité du répertoire carte contient en plus des clés et codes cités ci-dessus :

- le code porteur PIN. Ce code peut être présenté même si le répertoire actif n'est pas le répertoire carte. Le résultat de sa présentation est dans tous les cas mémorisé dans la zone d'accès du répertoire carte. Une authentification correcte du porteur par son PIN reste valable pendant toute une session, quels que soient les répertoires sélectionnés.
- la clé de fabrication (MK),
- la clé de personnalisation (PK)

La zone d'accès

Le rôle de cette zone a été cité au paragraphe 2.1.2., il est détaillé au paragraphe 2.1.10.

Les paramètres des descripteurs de répertoires et de zones de transactions:

Les paramètres d'un descripteur de répertoire indiquent les conditions à remplir pour créer des répertoires de niveau inférieur et des zones de transactions dans ce répertoire.
Les paramètres des descripteurs des zones de transactions indiquent les conditions à remplir pour pouvoir lire, écrire et effacer des données dans ces zones.
Ces conditions exigent la présentation d'une clé, d'un code, ou d'une combinaison de clés et/ou de codes.
Lors d'un accès à une zone ou lors de la création d'un domaine,

le système de sécurité contrôle si ces conditions sont satisfaites et, le cas échéant, autorise l'action demandée.

2.1.6. Rôle des différents codes et clés

Les deux premières clés sont enregistrées à des adresses fixes de la mémoire EEPROM.

La clé de fabrication (MK)

Cette clé assure la sécurité du composant entre le fabricant et l'encarteur. Ce dernier doit la présenter correctement pour effectuer la pré-personnalisation de la carte.

La clé de personnalisation (PK)

Cette clé sécurise la carte entre l'encarteur et l'émetteur de la carte et permet de réaliser des écritures sécurisées en phase de personnalisation.

Les clés et codes suivants sont chacun contenus dans une zone secrète. Le lecteur trouvera ci-dessous leurs différents rôles. La description des instructions sécuritaires montrera comment ils sont mis en oeuvre.

Les codes PIN et AID

Les codes secrets PIN et AID permettent d'authentifier le porteur de la carte.

Le PIN est unique. Le code secret de quatre chiffres à présenter lors de l'utilisation d'un distributeur automatique de billets en est un exemple.

Chaque répertoire, quel que soit son niveau, peut contenir un code AID. Un tel code est en général une donnée mémorisée sans difficulté par le porteur telle que sa date de naissance.

La clé de l'émetteur primaire (IK)

L'émetteur primaire est l'acteur qui décide de lancer une application utilisant des cartes. Il a le pouvoir le plus haut dans le répertoire qui lui a été réservé dans les cartes. Il s'agit par exemple de la maison mère d'une chaîne de magasins.

Chaque répertoire autonome contient une clé d'émetteur primaire. Elle est nécessaire pour réaliser les opérations

- d'authentification de l'émetteur primaire,
- d'écriture sécurisée,
- d'effacement sécurisé,

. d'invalidation d'un domaine,
. de déblocage du répertoire suite à trois mauvaises présentations d'un code porteur.

Les clés d'émetteurs secondaires (SK)

Les émetteurs secondaires d'une application sont les acteurs subordonnés à l'émetteur primaire. Il s'agit par exemple des gérants des différentes succursales d'une chaîne de magasins. Leur pouvoir est moindre que celui de l'émetteur primaire.
Les clés d'émetteurs secondaires permettent de réaliser les opérations

. d'authentification de l'émetteur secondaire et
. d'écriture sécurisée.

Les clés d'authentification (AK)

Ces clés permettent

. d'authentifier un terminal,
. de certifier des données,
. d'authentifier le porteur par chiffrement du PIN ou d'un AID.

Les clés d'effacement (EK)

Ces clés permettent de réaliser l'effacement sécurisé de mots dans les zones de transactions.

Les clés MAC

Les clés suivantes sont utilisées pour la génération et la vérification de signatures de messages externes (MAC) :

- la clé de génération de MAC: GK,
- la clé de vérification de MAC: VK,
- la clé de génération/vérification de MAC: GVK.

Le PIN et les AID sont chacun écrits dans des zones secrètes de taille suffisamment grande que pour permettre de les substituer. Le premier code est écrit à l'adresse la plus haute de la zone secrète. Le remplacement du code consiste à écrire le nouveau code à une adresse plus faible (comme dans une pile). Le système de sécurité utilise le premier code rencontré dans la zone secrète.
La gestion des clés IK, SK, AK et EK est différente de celle des codes PIN et AID. Une seule clé peut être écrite dans une zone

secrète, son remplacement est réalisé en créant une nouvelle zone secrète, en y écrivant la nouvelle clé et en invalidant la zone secrète contenant l'ancienne clé.

2.1.7. Les descripteurs de domaine

Les informations suivantes sont communes à tous les descripteurs :

- le niveau du domaine dans la hiérarchie (carte, application ou service),
- le type du domaine (répertoire, zone secrète, zone d'accès, zone publique ou zone de transactions),
- la longueur du domaine exprimée en mots, descripteur compris,
- le verrou (bit d'invalidation) indiquant l'état du domaine : domaine valide ou invalide,
- le checksum du descripteur, calculé par la carte lors de la création du domaine. Le checksum est le complément à 2 de la somme des octets formant le descripteur (sans les verrous). Il peut être exploité par le monde extérieur pour vérifier l'intégrité du contenu du descripteur.

Outre ces informations, la composition des différents descripteurs est détaillée ci-dessous.

a) Contenu des descripteurs de répertoires

- La référence du répertoire permet de l'identifier parmi les autres répertoires de même niveau.
- Les paramètres utilisés par le système de sécurité :
  . création de répertoires de niveau inférieur interdite / création de répertoires de niveau immédiatement inférieur autorisée,
  . conditions de création de répertoires de niveau inférieur,
  . conditions de création de zones de transactions,
  . numéro de la clé d'émetteur secondaire à présenter pour créer un répertoire de niveau inférieur si la condition de création de répertoires requiert une telle clé,
  . numéro de la clé d'émetteur secondaire à présenter pour créer une zone de transactions si la condition de création de zones de transactions requiert une telle clé,

. l'authentification du terminal est requise/n'est pas requise avant d'effectuer tout autre type d'authentification ou une écriture sécurisée,
. le répertoire dépend du système de sécurité du répertoire auquel il appartient/le répertoire possède son propre système de sécurité (répertoire autonome).

- Le verrou indique la phase de vie dans laquelle se trouve le répertoire : répertoire en phase de personnalisation ou en phase d'utilisation.

b) Contenu des descripteurs de zones de transactions

- Mode d'écriture : zone de transactions consommable en mode 'jeton' ou en mode 'mot'.
  Le mode 'jeton' permet de consommer les mots de la zone bit par bit par des écritures successives (surcharge du contenu précédent jusqu'à ce que tous les bits vaillent 1).
  Si le mode 'mot' est choisi, chaque écriture est suivie de la validation du mot. Une deuxième écriture sur ce mot devient alors impossible. La validation sera exposée au paragraphe 2.1.8.
- La référence de la zone : elle permet de distinguer la zone des autres zones de transactions du même répertoire. Lorsque la zone est pleine, il est possible d'en allouer une nouvelle ayant la même référence. Nous verrons que l'instruction de sélection de domaine permet dans ce cas de passer de l'une à l'autre.
- Les paramètres utilisés par le système de sécurité :
  . conditions d'effacement de données dans la zone,
  . conditions d'écriture de données dans la zone,
  . conditions de lecture de données dans la zone,
  . numéro de la clé d'émetteur secondaire à utiliser lorsque la condition d'écriture requiert une telle clé et/ou
    numéro de la clé d'effacement à mettre en oeuvre lorsque la condition d'effacement requiert une telle clé,
  . numéro de la clé d'émetteur secondaire à utiliser lorsque la condition de lecture requiert une telle clé,
  . option de mémorisation du type d'écriture de chaque mot : Si l'option n'est pas retenue, l'utilisateur dispose de 32 bits utiles pour chaque mot. Si la mémorisation est demandée, le

32ème bit de chaque mot indique s'il a été écrit par une écriture directe ou par une écriture sécurisée. L'utilisateur ne dispose dans ce cas que de 31 bits utiles pour chaque mot.

c) Contenu des descripteurs de zones publiques

La référence de la zone : elle permet de distinguer la zone des autres zones publiques appartenant au même répertoire.

d) Contenu des descripteurs de zones d'accès

- Le nombre maximum de présentations consécutives erronnées de code PIN ou AID autorisées avant blocage : trois présentations.
- Le nombre maximum de présentations consécutives erronnées de clé IK ou SK autorisées avant blocage de la carte : une présentation.
- Les conditions nécessaires pour débloquer un répertoire : présentation du PIN suivie de la clé IK.

e) Contenu des descripteurs de zones secrètes renfermant un code secret

- L'identifiant du code : il indique si le code est le PIN ou un AID.
- La condition de remplacement du code : cette condition est soit la présentation du code, soit la présentation du code et l'authentification de l'émetteur primaire.

f) Contenu des descripteurs de zones secrètes renfermant une clé secrète IK, SK, EK ou AK

- L'identifiant de la clé : indique le type de clé contenue dans la zone.
  Pour les clés d'émetteur secondaire, d'authentification et d'effacement, ce champ indique également le numéro de clé (il peut en effet exister 4 clés de ces types dans un répertoire).
- La version de la clé.

g) Contenu des descripteurs de zones secrètes renfermant une clé de signature de message (MAC)

- L'identifiant : indique si la zone contient une clé de génération de signature, une clé de vérification de signature ou une clé de génération et de vérification de signature.
- La version de la clé.

L'organisation d'une zone MAC est différente de celle d'une zone secrète contenant une clé IK, SK, EK ou AK. Le corps de la zone est composé de la clé, d'un mot de paramètres et d'une valeur initiale. Il sera décrit au point f. du paragraphe 2.1.11.3.

h) Les conditions d'accès et de création

Les conditions de création de répertoires et de zones de transactions, et les conditions d'accès en lecture, écriture et effacement dans une zone de transactions sont chacune exprimées dans un quartet. Deux bits concernent les conditions requises par le porteur (4 valeurs possibles) et les deux autres bits concernent les conditions requises par les émetteurs (4 valeurs possibles).

. Conditions requises par le porteur :
  - pas de protection,
  - présentation du PIN ou de l'AID,
  - présentation du PIN et de l'AID,
  - création et accès interdit.

. Conditions requises par l'émetteur pour la création d'un répertoire ou d'une zone de transactions :
  - pas de protection,
  - écriture du descripteur du nouveau domaine sécurisée par une clé d'émetteur secondaire,
  - authentification de l'émetteur primaire,
  - authentification de l'émetteur primaire et écriture du descripteur du nouveau domaine sécurisée par la clé de l'émetteur primaire.

. Conditions requises par l'émetteur pour l'effacement dans une zone de transactions :
  - pas de protection,
  - effacement sécurisé par une clé d'effacement,
  - authentification de l'émetteur primaire,

- authentification de l'émetteur primaire et effacement sécurisé par la clé de l'émetteur primaire.

. Conditions requises par l'émetteur pour l'écriture dans une zone de transactions :
  - pas de protection,
  - écriture du mot sécurisée par une clé d'émetteur secondaire,
  - authentification de l'émetteur primaire,
  - authentification de l'émetteur primaire et écriture du mot sécurisée par la clé de l'émetteur primaire.

Notons que lorsqu'une condition en écriture est remplie, elle génère également une autorisation en lecture selon l'ordre donné ci-dessous :
  - pas de protection,
  - authentification de l'émetteur secondaire,
  - authentification de l'émetteur primaire,
  - authentification de l'émetteur primaire.

Si aucune de ces conditions n'est remplie lors d'un accès en lecture, le système de sécurité analyse les conditions suivantes :

. Conditions requises par l'émetteur pour la lecture dans une zone de transactions :
  - pas de protection,
  - authentification de l'émetteur secondaire,
  - authentification de l'émetteur primaire ou secondaire,
  - authentification de l'émetteur primaire.

2.1.8. La validation de chaque mot

Lorqu'une zone de transactions est consommable en mode 'mot', toute écriture d'un mot dans cette zone sera suivie de sa validation automatique. La carte vérifie que les données effectivement écrites correspondent bien aux données qu'elle a reçues dans l'instruction d'écriture. Si l'écriture d'un mot est correcte, son bit de validation est positionné. Ce bit est la preuve de la bonne écriture physique du mot et permet d'éviter une écriture ultérieure (par surcharge des bits non écrits) du mot.
En cas d'incident d'écriture, le bit n'est pas positionné. Un mot non validé et non vierge est considéré comme un mot erroné et est signalé comme tel lors d'une lecture.

Une innovation importante par rapport aux anciens masques est que le bit de validation est transparent à l'utilisateur. Tout en assurant une sécurité optimale, TB100 offre donc 32 bits utiles par mot.

2.1.9. Définitions

- Pour rappel, un domaine est soit un répertoire, soit une zone.
- Le domaine courant est le dernier domaine sélectionné.
- Le répertoire actif est soit le répertoire courant si celui-ci possède son propre système de sécurité, soit le premier répertoire de niveau supérieur possédant un système de sécurité et incluant le répertoire courant.
- Le système de sécurité actif est le système de sécurité du répertoire actif associé au répertoire courant.
- Les clés actives sont les clés composant le système de sécurité du répertoire actif.

2.1.10. La zone d'accès

Une zone d'accès doit être créée dans chaque répertoire autonome. Cette zone est un des maillons importants du système anti-fraude géré par la carte. Elle permet de mémoriser le résultat (réussite ou échec) de :

. chaque authentification du porteur par PIN ou AID,
. chaque authentification de l'émetteur primaire ou secondaire,
. chaque écriture et effacement sécurisé,
. chaque invalidation d'un domaine.

Elle n'est pas mise à jour lors

. d'une authentification du terminal,
. d'une certification,
. d'une signature.

Le résultat de l'opération est enregistré

. dans la zone d'accès du répertoire carte s'il s'agit d'une authentification du porteur par PIN,
. dans la zone d'accès du répertoire actif lié à la clé pour toute autre opération.

Le système d'exploitation analyse les informations enregistrées dans cette zone afin de savoir quel code ou clé peut être

exploité. Ceci permet d'éviter les recherches exhaustives des clés et des codes.
En effet,

- après trois présentations fausses d'un code secret (PIN ou AID), ou après une mauvaise présentation d'une clé d'émetteur, le répertoire concerné passe à l'état bloqué.
  Il est possible de le débloquer en réalisant une authentification du porteur par PIN suivie d'une authentification de l'émetteur primaire du répertoire bloqué.
- après une mauvaise présentation d'un code porteur dans un répertoire donné, une clé d'émetteur primaire ou secondaire ne peut pas être présentée dans le même répertoire afin de ne pas pas permettre de rechercher une clé par une autre.

Recyclage d'une zone d'accès

Lorsque la zone d'accès d'un répertoire d'une carte équipée d'une mémoire EPROM est saturée, toute instruction exploitant un code ou une clé est inutilisable dans ce répertoire.
La mémoire de type EEPROM de la carte TB100 offre l'avantage de pouvoir recycler une zone d'accès avant saturation. Le recyclage est effectué automatiquement à condition de satisfaire aux conditions de sécurité.

La zone d'accès transitoire

Pendant les phases de fabrication et de personnalisation de la carte, une zone d'accès transitoire située à une adresse fixe est prévue pour mémoriser les résultats des présentations des clés de fabrication et de personnalisation.

2.1.11. Fonctionnalités

2.1.11.1. Le dialogue carte-terminal

La communication entre la carte et son environnement est conforme à la norme ISO 7816-3.
Les ordres envoyés par un terminal à une carte sont composés des cinq octets suivants :

- CI : la classe d'instruction,
- CODOP : le code hexadécimal de l'ordre à exécuter,
- P1, P2 et P3 : les paramètres de l'instruction.

Dans le cas d'un ordre entrant (flux de données vers la carte), les cinq octets sont suivis d'une chaîne de données de longueur 'P3'. La carte répond par l'envoi des mots d'état ME1 et ME2.
Dans le cas d'un ordre sortant (flux de données extrait de la carte), la réponse de la carte est une chaîne de données suivie des mots d'état ME1 et ME2.

Les mots d'état ME1 et ME2 :

- S'il y a une erreur de forme sur les cinq octets, l'ordre n'est pas exécuté et la valeur de ME1 indique la nature de l'erreur; ME2 n'est pas significatif.
- Si la forme de l'ordre est correcte et s'il est exécuté entièrement (même s'il s'est mal déroulé) ME1 indique le nombre de présentations de PIN ou de clés autorisées avant blocage et le type d'authentification qui a causé l'erreur ou le blocage éventuel. ME2 donne d'autres détails sur le déroulement de l'ordre.

2.1.11.2. Les instructions de base

a) Remise à zéro (RAZ)

La remise à zéro permet d'initialiser le composant; elle est obligatoire avant tout dialogue avec la carte. Elle n'est pas activée logiquement par l'envoi d'un ordre, mais est déclenchée physiquement par l'application d'un signal sur le contact 'RAZ'. Le format de la réponse renvoyée au terminal est imposé par la norme ISO 7816-3.

- Le contenu des quatre premiers octets (appelés 'octets systèmes') est spécifié dans la norme. Ils déterminent les caractéristiques de l'interface de la carte.
- Le contenu des cinq octets suivants (appelés 'octets historiques') n'est pas spécifié dans la norme et est propre au masque. Ces octets apportent l'information suivante :
  T1 : type de SPOM (SPOM 21)
  T2 : famille du masque (MP)
  T3-T4 : version du masque (TB100)
  T5 : phase de vie de la carte
- Les mots d'état ME1 et ME2.

Après la remise à zéro, le répertoire carte est le répertoire courant.

b) Sélection de domaine

Arguments :
P1 : /
P2 : mode de sélection
P3 : longueur du pattern envoyé (0, 1 ou 2 octet(s))
<D1/D1-D2> : pattern envoyé à la carte

Cette instruction permet de se déplacer dans l'arborescence afin de déterminer le domaine courant.
Il existe quatre modes de sélection :

- sélection du premier domaine dont le premier (les deux premiers) octet(s) du descripteur correspond(ent) au pattern fourni,
- sélection du dernier domaine dont le premier (les deux premiers) octet(s) du descripteur correspond(ent) au pattern fourni,
- sélection du domaine suivant le domaine courant et dont les deux premiers octets du descripteur sont égaux à ceux du domaine courant,
- sélection du domaine précédant le domaine courant et dont les deux premiers octets du descripteur sont égaux à ceux du domaine courant.

c) Lecture de données

Arguments :

P1-P2 : adresse de début de lecture dans le domaine

P3 : nombre d'octets à lire

Cette instruction permet de lire des données (maximum 256 octets) dans le domaine courant.

Lorsque le domaine courant est un répertoire, le contenu des zones appartenant au répertoire n'est pas lu. L'instruction permet alors de connaître la sous-structure du répertoire courant, elle renvoie

- les descripteurs des zones du répertoire courant,
- les descripteurs des sous-répertoires de niveau inférieur appartenant au répertoire courant,
- les mots vierges du répertoire courant.

Lorsque le domaine courant est une zone secrète, seuls les mots vierges sont renvoyés. Le contenu des mots écrits est remplacé par des zéros.

La lecture des zones d'accès et des zones publiques est libre; la lecture d'une zone de transactions est contrôlée par les conditions spécifiées dans son descripteur.

d) Ecriture de données

Arguments :

P1-P2 : adresse d'écriture du mot dans la zone

P3 : longueur des données fournies (4 octets)

D1-D4 : contenu du mot à écrire

Cette instruction permet d'écrire un mot dans la zone courante. Si l'option 'jeton' a été choisie lors de la création de la zone, plusieurs écritures peuvent être effectuées sur le même mot. Sinon, le mot est validé automatiquement et ne peut pas être surchargé.

e) Effacement de données

Arguments :

P1-P2 : adresse du premier mot à effacer dans la zone

P3 : longueur des données fournies (2 octets)

D1-D2 : adresse du premier mot suivant la fin de la chaîne à effacer

Cette instruction permet d'effacer une chaîne de mots dans la zone de transactions courante.

f) Recherche du prochain mot vierge

Arguments :
P1-P2 : adresse du début de la recherche
P3 : longueur des données fournies (0 octet)
Cette instruction permet de rechercher le premier mot vierge situé à partir de l'adresse 'P1-P2' dans le domaine courant.

g) Recherche du prochain mot non vierge

Arguments :
P1-P2 : adresse du début de la recherche
P3 : longueur des données fournies (0 octet)
Cette instruction permet de rechercher le premier mot non vierge situé à partir de l'adresse 'P1-P2' dans le domaine courant.

h) Recherche sans masque d'un mot écrit

Arguments :
P1-P2 : adresse du début de la recherche
P3 : longueur du pattern fourni (1 ou 2 octets)
D1/D1-D2 : pattern à rechercher
Cette instruction permet de rechercher dans la zone courante un mot dont le premier ou les deux premiers octet(s) correspond(ent) au pattern fourni. La recherche est impossible dans une zone secrète.

i) Recherche avec masque d'un mot écrit

Arguments :
P1-P2 : adresse du début de la recherche
P3 : longueur des données fournies (9 octets)
D1 : mode de recherche
D2-D5 : pattern à rechercher
D6-D9 : masque
Cette instruction permet de rechercher dans une zone un mot écrit qui, masqué par le mot D6-D9, correspond au mot D2-D5. Un bit à '0' dans le masque indique que la carte doit ignorer la valeur du bit correspondant du mot lu dans la zone.

Les quatre modes de recherche suivants peuvent être combinés :

- recherche vers les adresses croissantes,
- recherche vers les adresses décroissantes,
- recherche du premier mot dont la configuration est égale à celle du pattern,
- recherche du premier mot dont la configuration est différente de celle du pattern.

La recherche est impossible dans une zone secrète.

j) Lecture de résultat

Arguments :

P1-P2 : /

P3 : longueur des données renvoyées (8, 12 ou 20 octets)

Cette instruction permet de lire un résultat mémorisé par la carte suite à l'exécution de certaines instructions. La lecture doit être effectuée immédiatement après l'instruction concernée.

Après une recherche réussie, la lecture renvoie :

- l'adresse relative du mot trouvé,
- le nombre total de mots du domaine,
- le contenu du mot.

Après une remise à zéro, la lecture renvoie :

- le nombre total de mots du répertoire carte,
- le contenu du descripteur du répertoire carte.

Après une sélection de domaine réussie, la lecture renvoie :

- l'adresse relative du descripteur sélectionné,
- le nombre total de mots du domaine,
- le contenu du descripteur du domaine.

Après une sélection de zone secrète MAC réussie, la lecture renvoie :

- l'adresse relative du descripteur sélectionné,
- le nombre total de mots de la zone,
- le contenu du descripteur de la zone,
- le mot des paramètres,
- la valeur initiale (cette valeur est remplacée par des zéros si sa lecture est interdite).

Après un calcul de MAC, la lecture renvoie les 8 octets constituant la signature.

Après une certification, la lecture renvoie le résultat du calcul du certificat.

k) Lecture d'un nombre aléatoire

Arguments :
P1-P2 : /
P3 : longueur du nombre aléatoire renvoyé (8 octets)

Cette instruction génère et renvoie un nombre aléatoire de 64 bits utilisé pour le chiffrement de message. La carte mémorise ce nombre jusqu'à la fin de la session ou jusqu'à la prochaine génération d'un nombre aléatoire.

l) Directory

Arguments :
P1-P2 : adresse relative de début de directory
P3 : nombre d'octets à lire

Cette instruction fournit la liste des descripteurs des zones et des répertoires de niveau inférieur appartenant au répertoire courant, à partir de l'adresse 'P1-P2'.
Elle a donc le même effet qu'une lecture de données dans un répertoire.

m) Création d'un domaine.

Arguments :
P1-P2 : adresse relative du descripteur à écrire
P3 : longueur du descripteur (3 ou 7 octets)
D1-D3/D7 : contenu du descripteur

Cette instruction permet d'écrire un descripteur dans le répertoire courant afin de créer un nouveau domaine. La longueur du descripteur varie en fonction du type de domaine créé.

n) Ecriture des verrous

Arguments :
P1 : /
P2 : type de verrou (fabrication ou utilisation)
P3 : '00'

Cette instruction permet de basculer
- le verrou de fabrication de la carte pour marquer la fin de la phase de fabrication et de pré-personnalisation,
- le verrou d'utilisation d'un répertoire pour marquer la fin de sa phase de personnalisation.

2.1.11.3. Les instructions de sécurité

La plupart des instructions de sécurité utilisent l'algorithme DES (Data Encryption Standard) codé en ROM. Le DES est un algorithme public symétrique utilisant des clés secrètes. Développé en 1974 par IBM, il n'a jusqu'ici, et à notre connaissance, jamais été brisé.
Cet algorithme permet de chiffrer des messages de 8 octets en utilisant une clé de 56 bits. La fonction directe du DES peut être représentée comme suit:

$M' = DES\ (M, K)$

où M est le message d'entrée (64 bits)
K est la clé secrète (56 bits)
M' est le message chiffré résultant (64 bits)

La fonction inverse du DES permet de retrouver le message d'entrée M à partir du message chiffré M':

$M = DES^{-1}\ (M', K)$

Les procédures de sécurité qui utilisent le DES font intervenir le module de sécurité du terminal. Ce module est le coffre-fort qui contient les clés secrètes du terminal, il est capable de chiffrer et de déchiffrer les messages échangés avec la carte. Les fonctionnalités de ce module sont présentées au paragraphe 2.2.

a) L'authentification

Le but de l'authentification d'un intervenant (porteur, émetteur,...) est de vérifier son identité, son habilitation à effectuer certaines opérations. Un intervenant est authentifié si la clé ou le code qu'il a présenté(e) correspond à la valeur de la clé ou du code mémorisé(e) dans le répertoire actif de la carte.

Dans un système à cartes passives telles que les cartes à bande magnétique, le porteur introduit son code secret au terminal et ce dernier le compare au code inscrit sur la bande. Cette opération nécessite un transfert du code de la carte vers le terminal. Un fraudeur peut éventuellement agir durant ce transfert, d'où la faiblesse de ce système.
La carte à microcalculateur permet d'éliminer ce risque puisqu'elle effectue elle-même la comparaison. De plus, le code

ou la clé présenté(e) peut être chiffré(e) par le terminal avant d'être envoyé(e) à la carte. Le chiffrement est fonction d'un nombre aléatoire généré par la carte, afin de rendre inexploitable tout message lu sur la ligne par un éventuel fraudeur.

Authentification du terminal

Arguments :

P1-P2 : /
P3 : longueur des données fournies (10 octets)
D1-D2 : référence de la clé d'authentification AK
D3-D10 : message d'authentification

Cet ordre permet au répertoire actif de s'assurer que le terminal n'est pas contrefait. Ce dernier doit prouver qu'il connait la clé d'authentification AK (référencée par D1-D2) du répertoire actif.
L'authentification peut être réalisée à partir de n'importe quel répertoire actif et reste valable pour tous les répertoires sélectionnés jusqu'à la fin de la session. La procédure d'authentification est schématisée ci-après.

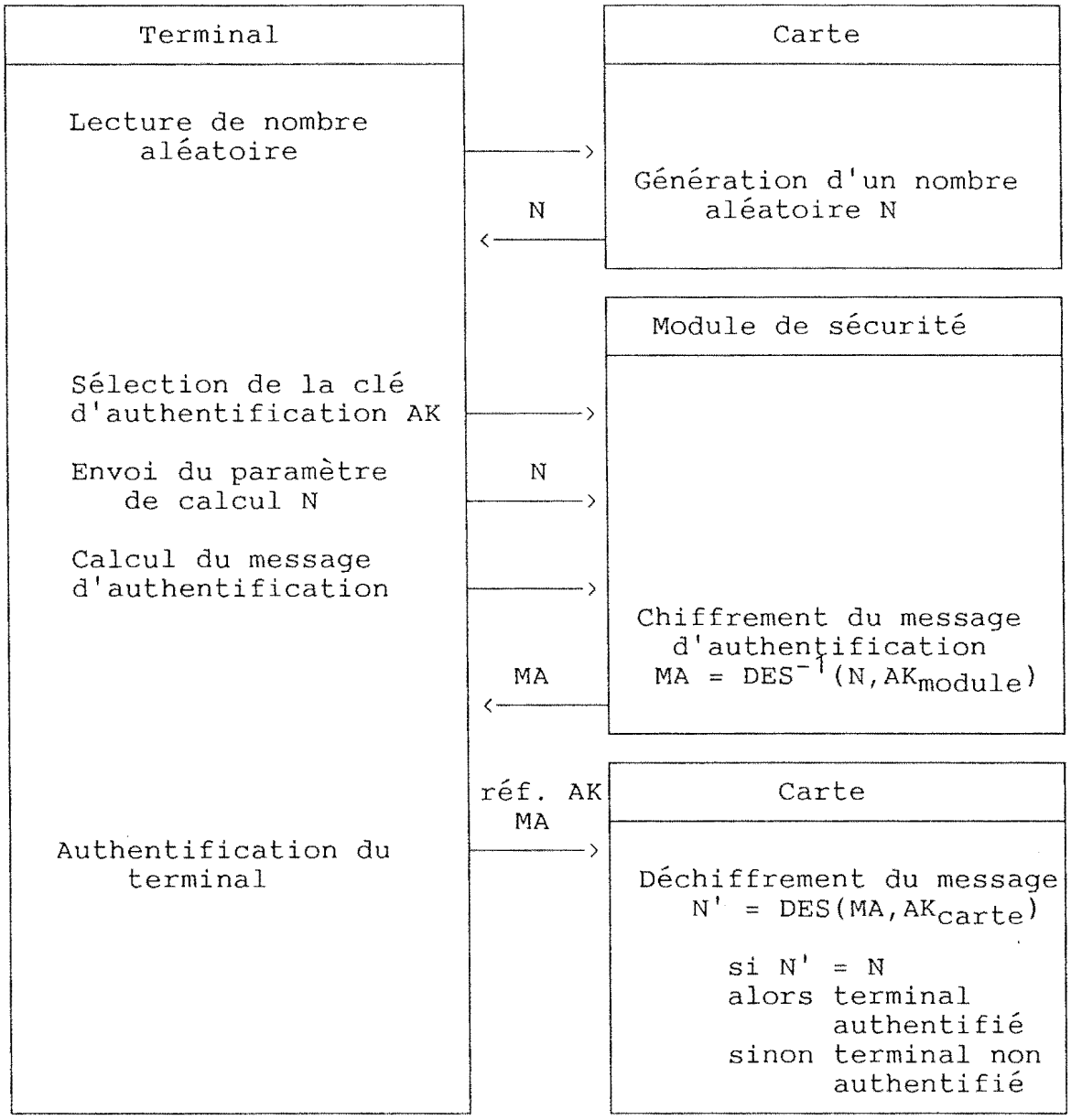


figure 7 : Authentification du terminal par une carte TB100

Authentification non chiffrée du porteur par PIN

Arguments :

P1-P2 : /

P3 : longueur du code fourni (8 octets)

D1-D8 : valeur du code PIN

Cet ordre permet au répertoire actif de s'assurer que le porteur de la carte est légitime. Ce dernier doit prouver qu'il connaît le code confidentiel (PIN) mémorisé dans la zone secrète du répertoire *carte*.

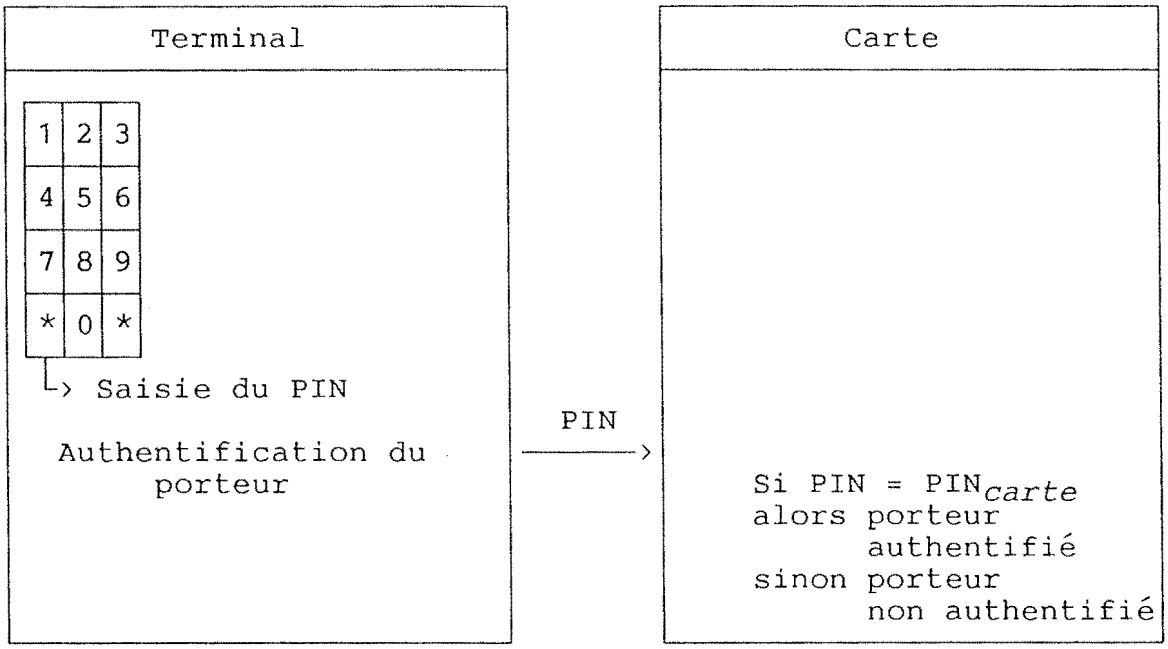


figure 8 : Authentification non chiffrée du porteur par PIN

Authentification chiffrée du porteur par PIN

Arguments :

P1-P2 : /

P3 : longueur des données fournies (10 octets)

D1-D2 : référence de la clé d'authentification AK utilisée pour le chiffrement

D3-D10 : message d'authentification

Le rôle de cet ordre est équivalent à celui de l'authentification non chiffrée du porteur par PIN.

Le code est chiffré par le module de sécurité du terminal avant d'être envoyé à la carte. Le chiffrement utilise la clé AK du répertoire actif référencée par D1-D2.

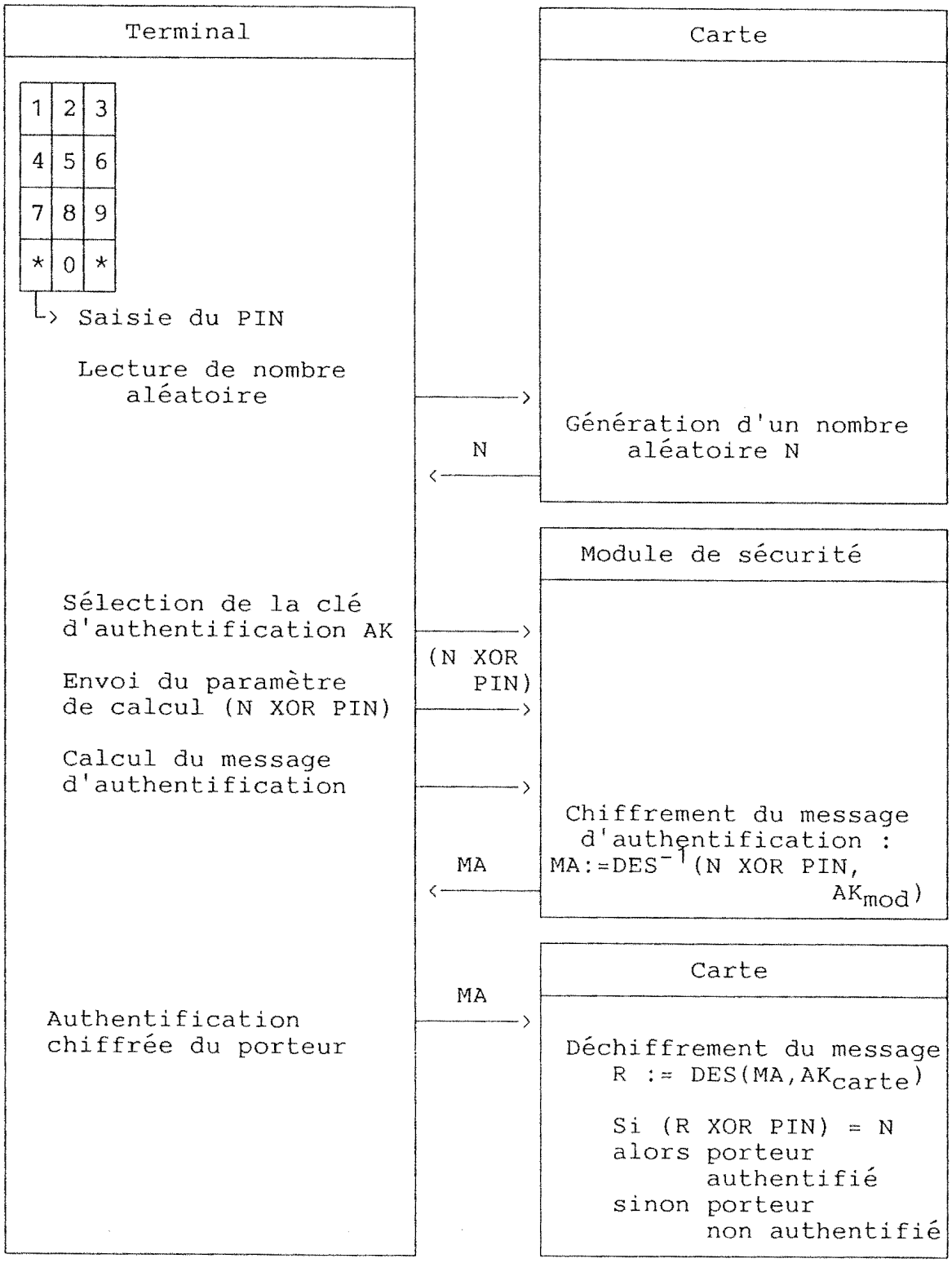


figure 9 : Authentification chiffrée du porteur par PIN

Authentification non chiffrée du porteur par AID

Arguments :

OR : '30'
P1-P2 : /
P3 : longueur du code fourni (8 octets)
D1-D8 : valeur du code AID

Cet ordre permet au répertoire actif de s'assurer que le porteur de la carte est légitime. Ce dernier doit prouver qu'il connaît le mot de passe additionnel AID mémorisé dans la zone secrète du répertoire *actif*.

Authentification chiffrée du porteur par AID

Arguments :

P1-P2 : /
P3 : longueur des données fournies (10 octets)
D1-D2 : référence de la clé d'authentification AK utilisée pour le chiffrement
D3-D10 : message d'authentification

Le rôle de cet ordre est équivalent à celui de l'authentification non chiffrée du porteur par AID.
Le code est chiffré par le module de sécurité du terminal avant d'être envoyé à la carte. Le principe de chiffrement est identique à celui utilisé pour l'authentification par PIN.

Authentification de l'émetteur par la clé IK

Arguments :

P1-P2 : /
P3 : longueur du message fourni (8 octets)
D1-D8 : message d'authentification

Cette instruction permet d'authentifier l'émetteur primaire du répertoire actif. Ce dernier doit prouver qu'il connaît la clé d'émetteur IK du répertoire actif.

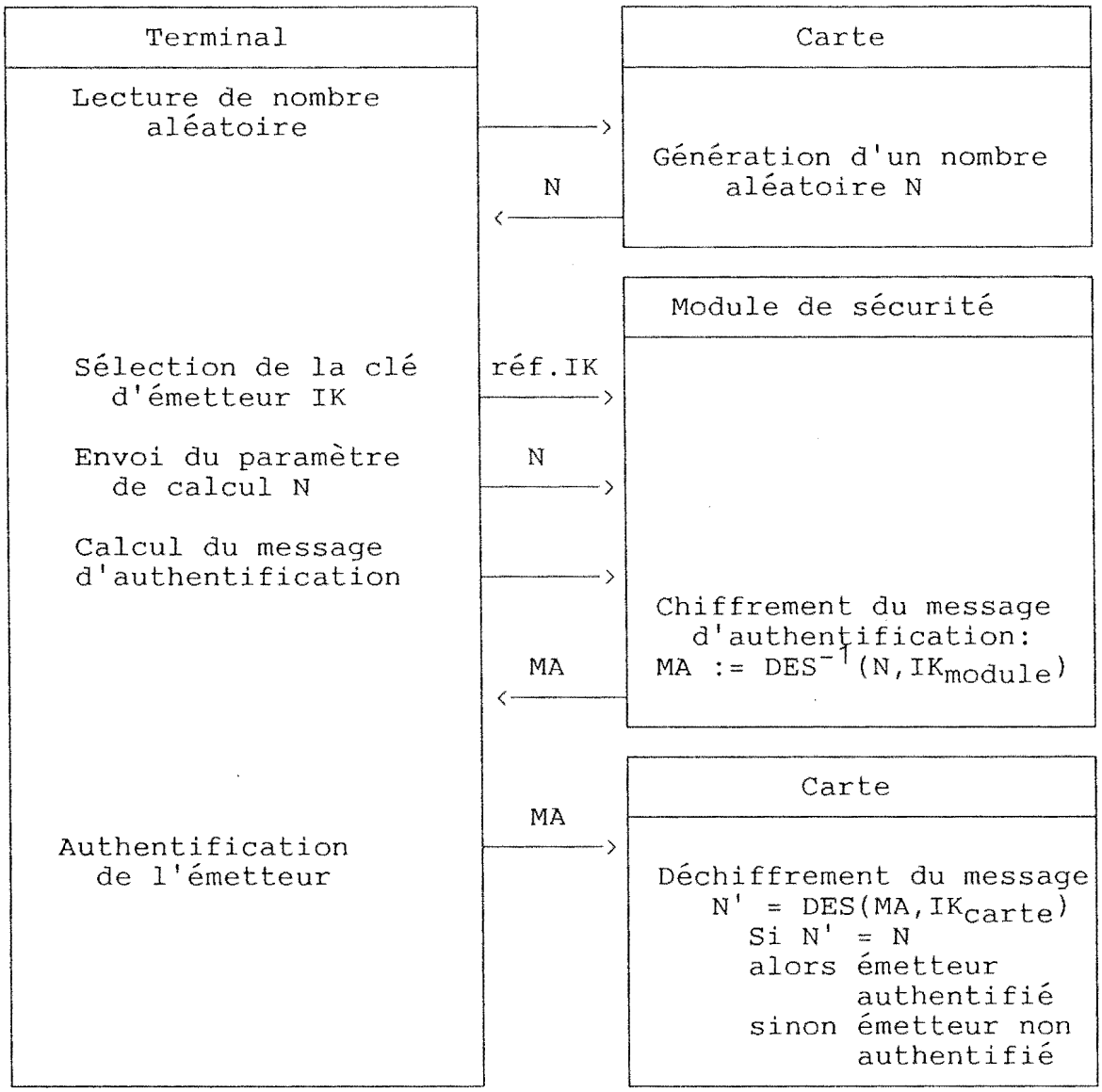


figure 10 : Authentification de l'émetteur primaire

Authentification de l'émetteur par clé SK

Arguments :

P1-P2 : /

P3 : longueur des données fournies (10 octets)

D1-D2 : référence de la clé de l'émetteur secondaire

D3-D10 : message d'authentification

Cette instruction permet d'authentifier un émetteur secondaire du répertoire actif. Ce dernier doit prouver qu'il connait la clé d'émetteur secondaire SK référencée par D1-D2.

Le principe de chiffrement est identique à celui utilisé pour l'authentification de l'émetteur primaire.

b) L'invalidation d'un domaine

Arguments :

P1-P2 : /

P3 : longueur du message fourni (8 octets)

D1-D8 : message chiffré

Cette instruction permet à l'émetteur primaire du répertoire actif d'invalider le domaine courant en écrivant le bit d'invalidation de son descripteur.

L'invalidation d'un domaine rend son évolution impossible. Lorsqu'un répertoire est invalidé, tous les domaines lui appartenant le deviennent aussi.

L'accès en lecture à un domaine invalidé est possible après une authentification de l'émetteur primaire du répertoire actif.

La procédure d'invalidation est similaire à l'authentification de l'émetteur primaire.

c) L'écriture sécurisée.

Cette procédure permet à une autorité habilitée (un émetteur primaire ou secondaire) de transférer des informations chiffrées, d'un host vers une carte en toute sécurité.

Elle permet également d'authentifier l'émetteur du message et donc de s'assurer qu'il est habilité à écrire les données dans la carte.

Les données envoyées constituent soit un mot à écrire dans une zone (code secret, clé secrète ou données confidentielles), soit un descripteur à écrire dans un répertoire afin de créer un nouveau domaine.

Ecriture sécurisée d'un mot (chiffrement avec IK)

Arguments :

P1-P2 : adresse relative du mot à écrire

P3 : longueur du message fourni (8 octets)

D1-D8 : message d'écriture

Cette instruction permet d'envoyer un mot chiffré à la carte, d'authentifier l'émetteur primaire et d'écrire le mot dans la zone spécifiée si l'authentification est correcte.

Le message d'écriture (chiffré par le DES avec la clé IK) contient les informations suivantes :

- l'adresse relative du mot à écrire (Adr),
- le mot à écrire (Mot),
- les deux premiers octets du descripteur de la zone destinataire (Id).

Pour rappel, les deux premiers octets d'un descripteur contiennent son niveau, son type et sa référence, ils permettent donc d'identifier la zone.

La procédure d'écriture sécurisée est schématisée ci-après.

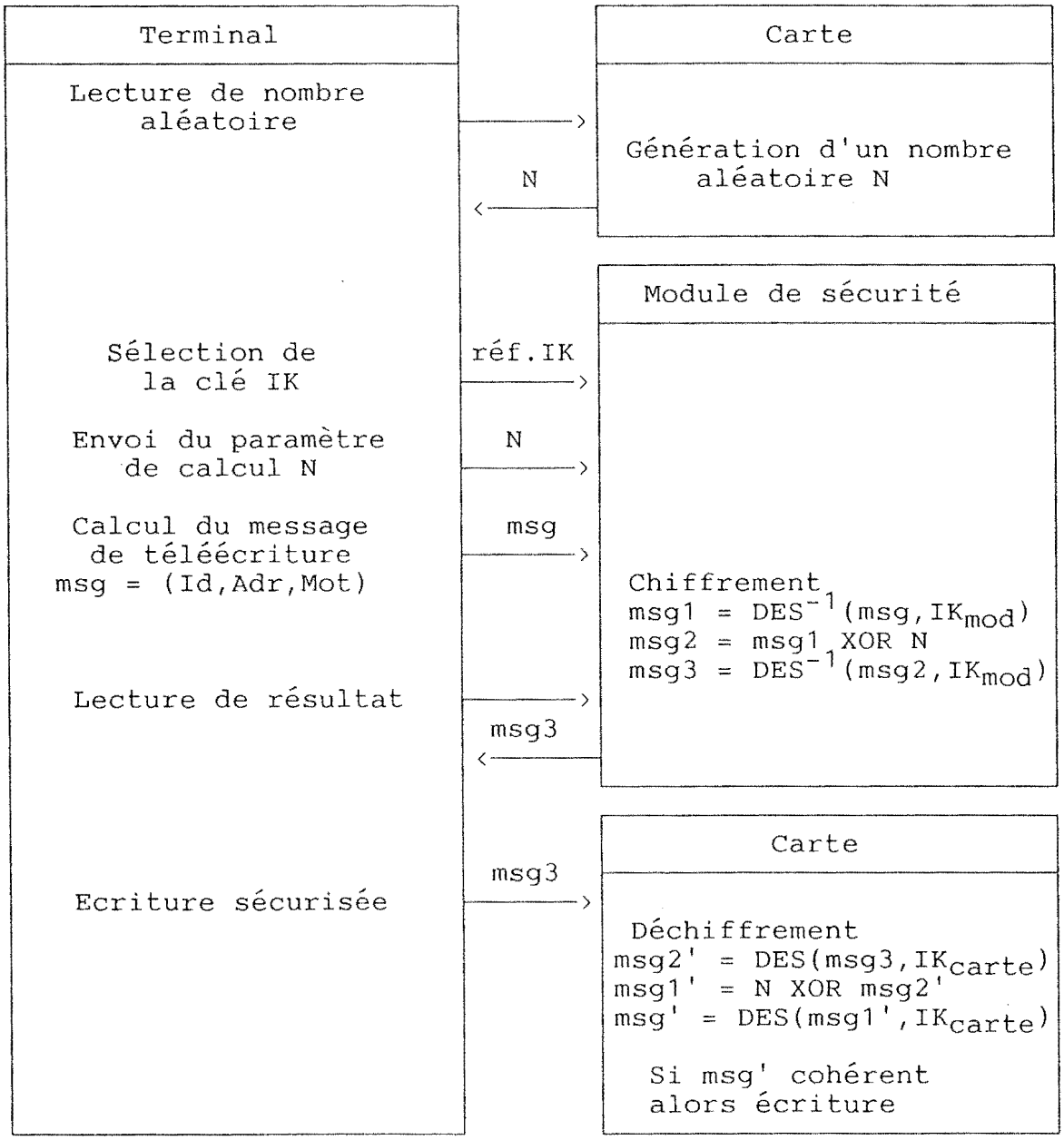


figure 11 : L'écriture sécurisée

Le contrôle de cohérence du message msg' consiste à vérifier que
- les deux premiers octets du descripteur correspondent aux deux premiers octets du domaine courant,
- les troisième et quatrième octets correspondent aux paramètres P1-P2 de l'ordre.

Ecriture sécurisée d'un mot (chiffrement avec SK)

Arguments :
P1-P2 : adresse relative du mot à écrire
P3 : longueur des données fournies (10 octets)
D1-D2 : référence de la clé SK
D3-D10 : message d'écriture

L'objectif et le fonctionnement de cet ordre sont similaires à ceux de l'instruction précédente.

Ecriture sécurisée d'un descripteur (chiffrement avec IK)

Arguments :
P1-P2 : adresse relative du descripteur à écrire
P3 : longueur des données fournies (9 ou 17 octets)
D1 : '00'
D2-D9/D17 : message d'écriture

Cet ordre permet de transférer un descripteur chiffré à la carte, d'authentifier l'émetteur primaire et d'écrire le descripteur dans le répertoire courant si l'authentification est correcte.

Le message d'écriture contient
- les deux premiers octets du descripteur du répertoire qui accueille le nouveau domaine,
- le nouveau descripteur,
- l'adresse relative du nouveau descripteur.

La procédure est similaire à la procédure d'écriture sécurisée d'un mot (chiffrement avec IK).

Ecriture sécurisée d'un descripteur (chiffrement avec SK)

Arguments :

P1-P2 : adresse relative du descripteur à écrire
P3 : longueur des données fournies (11 ou 19 octets)
D1 : '00'
D2-D3 : référence de la clé SK
D4-D11/D19 : message d'écriture

Le but et le principe de fonctionnement de cet ordre sont identiques à ceux de l'instruction précédente.

d) L'effacement sécurisé

Cette procédure permet de sécuriser l'effacement d'un ensemble de mots dans une zone de transactions. Le chiffrement utilise la clé d'émetteur IK ou une clé d'effacement EK. Le principe de la procédure d'effacement est identique à celui de l'écriture sécurisée, seul le contenu du message chiffré transféré diffère; il contient :

- les deux premiers octets du descripteur de la zone,
- l'adresse relative du premier mot à effacer,
- l'adresse relative du premier mot suivant la chaîne de mots à effacer.

Effacement sécurisé par la clé d'émetteur IK

Arguments :

P1-P2 : adresse relative du premier mot à effacer
P3 : longueur du message fourni (8 octets)
D1-D8 : message chiffré d'effacement

Effacement sécurisé par la clé d'effacement EK

Arguments :

P1-P2 : adresse relative du premier mot à effacer
P3 : longueur du message fourni (10 octets)
D1-D2 : référence de la clé EK
D3-DA : message chiffré d'effacement

e) La certification

La certification permet à un émetteur de s'assurer qu'un mot a été correctement enregistré dans la carte. Il peut s'agir d'un mot de donnée dans une zone, ou du premier mot d'un descripteur dans un répertoire.
Ce procédé est couramment utilisé dans les réseaux de cartes bancaires. En voici quelques exemples :

- dans une carte dont le crédit est limité à un plafond donné, l'organisme émetteur de la carte peut, lors d'une transaction avec la carte du porteur, vérifier à distance que ce plafond est correct,
- le résultat d'une transaction financière est mémorisé dans une zone de transactions de la carte et est transmis au système central pour créditer ou débiter le compte du propriétaire de la carte. L'organisme financier peut alors vouloir comparer les données reçues avec celles qui sont réellement écrites dans la carte.

L'ordre de certification

Arguments :

P1-P2 : adresse relative du mot à certifier
P3 : longueur des données fournies (10 octets)
D1-D2 : référence de la clé AK
D3-D10 : message d'entrée

Cette instruction permet de générer un certificat. Le message d'entrée D3-D10 contient

- l'adresse relative du mot à certifier et
- un nombre aléatoire provenant du terminal.

La procédure de certification

Le terminal envoie à la carte la référence de la clé AK utilisée pour le chiffrement et le message d'entrée.
La carte calcule le certificat sur base du message reçu, de la clé d'authentification AK et d'un paramètre interne (p.i.). Ce paramètre contient principalement les deux premiers octets du descripteur du domaine courant (réf. descr.) et le contenu du mot écrit à l'adresse spécifiée.
Le terminal lit ensuite le certificat et exécute lui-même le

calcul de certificat en utilisant le contenu présumé correct du mot. Si les deux certificats sont égaux, le terminal a la garantie de l'intégrité du mot.
Cette procédure est schématisée ci-dessous.

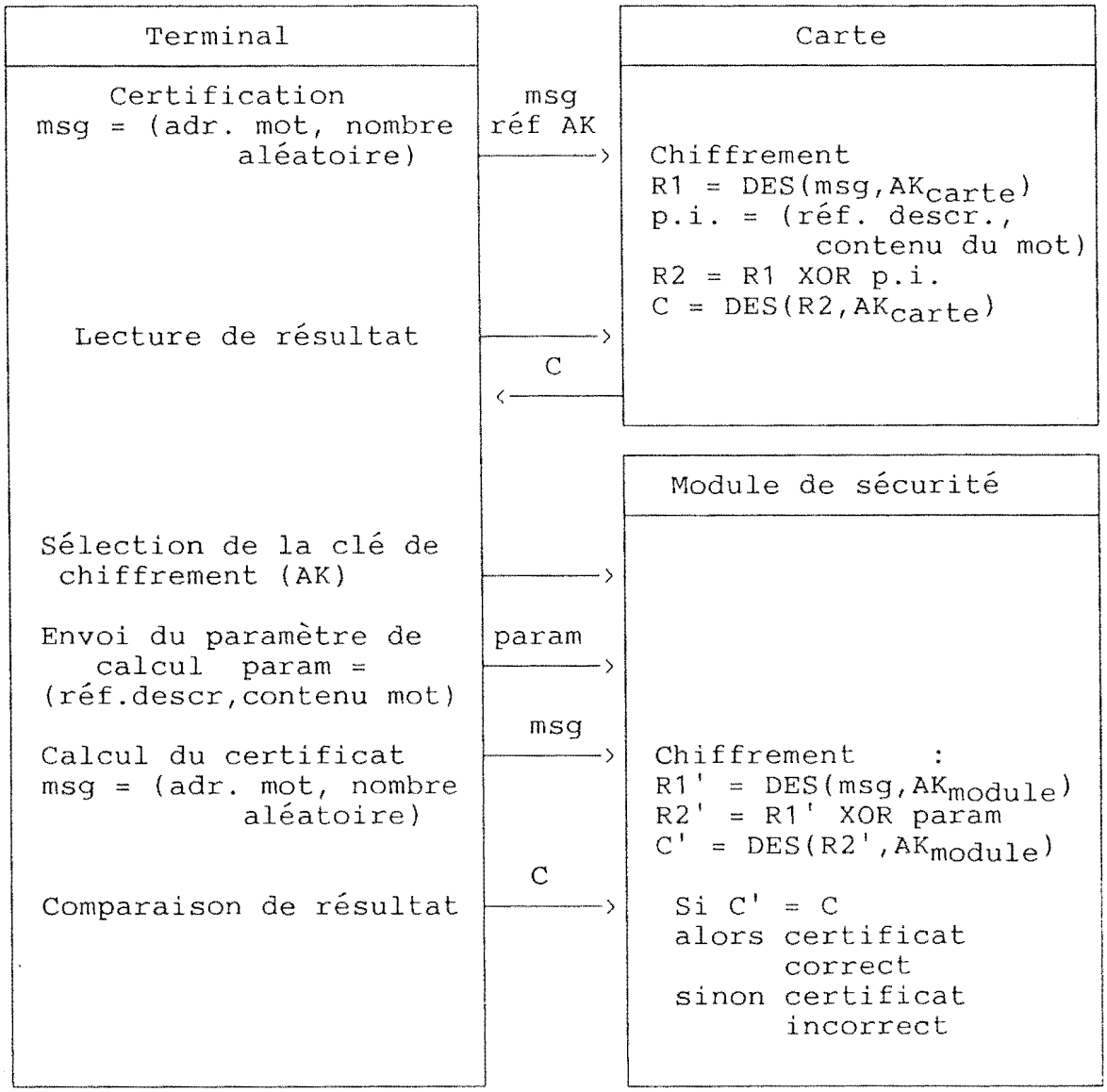


figure 12 : La certification

L'émetteur réalise une condensation et un chiffrement de toutes les données émises bloc par bloc depuis le début du dialogue, et envoie le résultat de ce chiffrement au destinataire à la suite du message. Après réception, le destinataire réalise le même calcul et compare la signature obtenue à celle de l'émetteur. Si elles sont égales, il a la garantie de l'authenticité et de l'intégrité du message.
La signature est obtenue par chiffrement chaîné des blocs de 64 bits composant le message de départ.

f) La signature de message

Lors d'une transaction, après authentification mutuelle, les deux correspondants peuvent encore se prémunir d'un fraudeur qui aurait attendu la fin de la phase d'authentification pour émettre des faux messages. La signature (Message Authentification Code) garantit l'authenticité et l'intégrité d'un message externe.

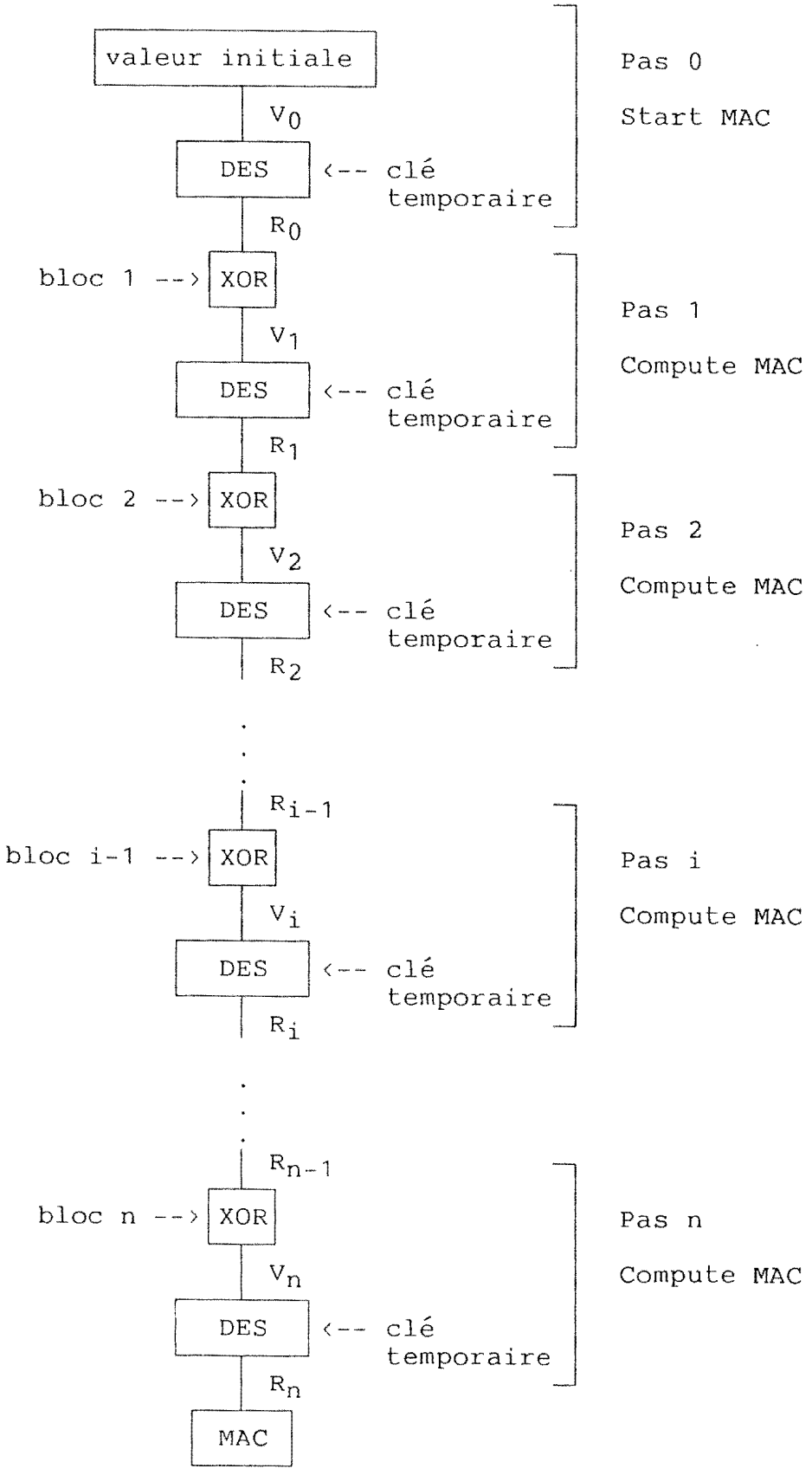


figure 13 : La signature de message

La signature est calculée par deux instructions spécifiques : d'abord l'instruction 'Start MAC', puis l'instruction 'Compute

MAC' autant de fois qu'il y a de blocs de 64 bits. Ces deux instructions utilisent une clé temporaire générée sur base des informations mémorisées dans la zone secrète MAC du répertoire actif.
Après son calcul, le MAC est lu par une 'lecture de résultat' et est envoyé au destinataire à la suite du message.
Le destinataire compare la signature reçue avec celle qu'il a calculée en utilisant l'instruction de 'vérification de MAC'.

Contenu de la zone secrète MAC

Les deux premiers mots de la zone constituent la clé secrète au départ de laquelle est générée la clé temporaire.
Le troisième mot contient les paramètres d'exploitation du MAC :

- le nombre minimum de pas exigés pour générer une signature,
- le type de valeur initiale utilisée par l'instruction 'start MAC' qui est :
  . soit une valeur interne constante mémorisée dans les quatrième et cinquième mots de la zone MAC,
  . soit un compteur interne dont la valeur est mémorisée dans les quatrième et cinquième mots de la zone MAC,
  . soit un nombre aléatoire généré par la carte,
  . soit une valeur externe : il s'agit par exemple d'un nombre aléatoire généré et transmis par l'émetteur d'un message.
- la clé temporaire doit être diversifiée au moins une fois avec le nombre aléatoire généré par la carte, ou la diversification n'est pas obligatoire,
- le chargement direct est ou n'est pas autorisé,
- les conditions requises du porteur pour pouvoir calculer une signature (cf point h du paragraphe 2.1.7.),
- la valeur interne est/n'est pas accessible en lecture.

Génération d'une clé temporaire

Arguments :

P1-P2 : /

P3 : longueur des données fournies (0, 2, 8 ou 10 octets)

<D1-D2> : référence de la clé MAC

<D3-D10> : valeur externe de diversification

ou <D1-D8> : valeur externe de diversification

Cet ordre permet de générer une clé temporaire utilisée dans la procédure de signature. La clé peut être diversifiée afin de la rendre unique pour chaque session. L'exploitation d'une telle clé n'est possible que si les droits d'utilisation spécifiés dans les paramètres de la zone secrète du MAC sont respectés.

Cet ordre peut être exécuté selon quatre modes différents:

- chargement direct de la clé de la zone MAC référencée par D1-D2
- chargement de la clé référencée par D1-D2 suivi d'une diversification par la valeur externe D3-D10
- diversification de la clé chargée précédemment par une valeur aléatoire interne
- diversification de la clé chargée précédemment par la valeur D1-D8.

Start MAC

Arguments :

P1-P2 : /

P3 : longueur des données fournies (0 ou 8 octets)

<D1-D8> : valeur externe

Cet ordre permet d'effectuer le premier pas d'un calcul de MAC en utilisant une valeur initiale (64 bits) qui est

- soit la valeur externe D1-D8,
- soit la valeur dont le type est spécifié dans le mot des paramètres de la zone secrète du MAC.
  Si la valeur initiale est un compteur ou une valeur interne, la carte cherche automatiquement sa valeur dans la zone secrète MAC précédemment sélectionnée par l'instruction de génération de clé temporaire.

Cette valeur est chiffrée en utilisant la clé temporaire générée précédemment. Le résultat R_0 est fourni en entrée de la première instruction 'Compute MAC' qui suit.

Compute MAC

Arguments :

P1-P2 : /

P3 : longueur des données fournies (8 octets)

D1-D8 : valeur externe (bloc i du message à signer)

Cet ordre effectue le $i^{ème}$ pas du calcul de MAC en utilisant le résultat R_0 de l'instruction 'Start MAC' ou le résultat R_{i-1} de la commande 'Compute MAC' précédente. Un 'OU exclusif' est réalisé entre ce résultat R_O ou R_{i-1} et le bloc i du message à signer. La valeur V_i ainsi obtenue est chiffrée avec la clé temporaire pour donner le résultat R_i.

Une lecture de résultat permet d'obtenir la signature si le nombre minimum de pas exigé pour sa génération a été respecté. Si la valeur initiale est un compteur, sa valeur est incrémentée lors de la lecture du résultat.

Vérification de MAC

Arguments :

P1-P2 : /

P3 : longueur des données fournies (8 octets)

D1-D8 : valeur de la signature reçue

Cette instruction permet au destinataire du message de comparer la signature reçue avec celle qu'il a calculée lui-même.

2.2. Le module de sécurité

2.2.1. Introduction

L'exécution des instructions mettant en oeuvre l'algorithme DES nécessite l'intervention du module de sécurité du terminal. Ce module doit être capable de calculer les fonctions directes et les fonctions inverses de celles de la carte.
Les modules de sécurité d'un réseau doivent en outre

- mémoriser les clés mères (rôle de coffre-fort) et être capables de (re)créer les clés diversifiées,
- permettre d'écrire de manière sécurisée les clés dans les cartes et dans les autres modules.

Ces modules possèdent une structure physique et logique analogue à celle de la carte, mais disposent d'un jeu d'instructions plus étendu. Ils se présentent soit sous la forme d'une carte, soit sous la forme d'un circuit intégré DIL (dual in line).

2.2.2. Typologie des modules

Un système gérant des cartes à micro-calculateur peut être schématisé comme suit :

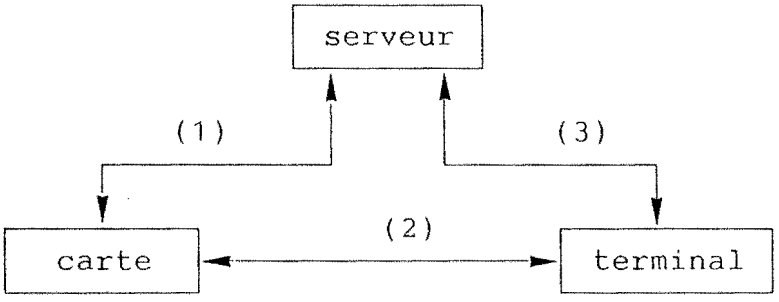


figure 14 : Schéma d'un système gérant un parc de cartes

- La liaison (1) est établie pour réaliser des transactions ON-LINE entre le serveur et la carte.
- La liaison (2) est établie pour réaliser des transactions OFF-LINE entre la carte et le terminal, sans intervention du serveur.
- La liaison (3) est utilisée par le serveur pour collecter les informations accumulées par le terminal.

Un tel système doit permettre l'introduction et la diffusion de clés au sein de ses différents éléments. Ces deux fonctions sont assurées par les liaisons suivantes :

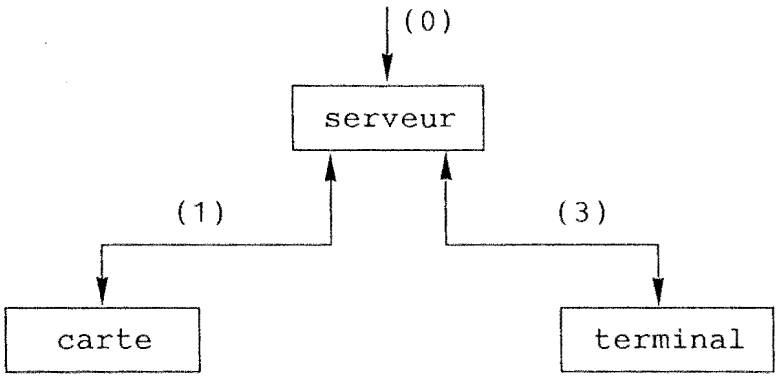


figure 15 : Diffusion des clés dans le système

- La liaison (0) permet d'introduire une clé dans le système.
- La liaison (1) permet au serveur de diffuser une clé dans la carte.
- La liaison (3) permet au serveur de doter le terminal d'une clé.

Notons que la liaison (2) de la figure 14 ne permet pas au terminal d'écrire une clé dans la carte.

Chaque type de liaison est sécurisé par un module de sécurité spécifique.

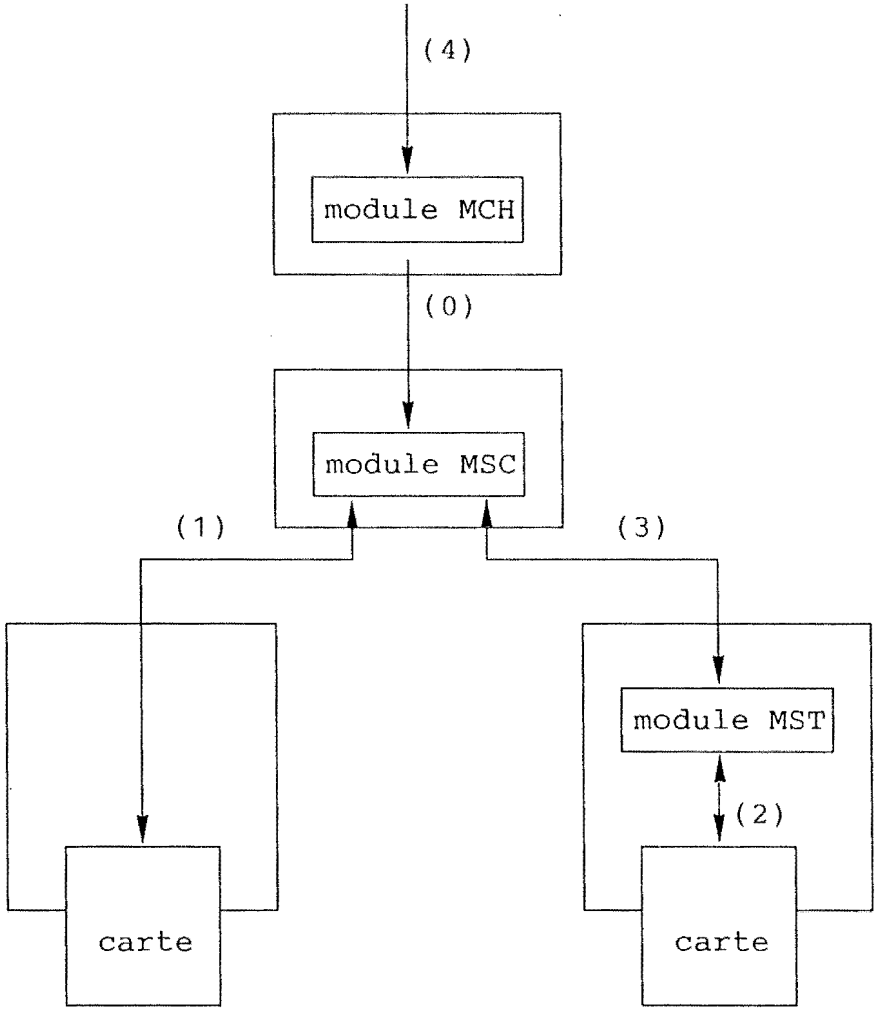


figure 16 : Les différents types de modules de sécurité

- Le module de chargement (MCH) permet de sécuriser l'introduction d'une clé dans le système par la liaison (4). Le MCH sécurise ensuite la diffusion de la clé vers le serveur (liaison (0)) par une écriture sécurisée.
- Le module de sécurité associé au central (MSC) permet quant à lui de sécuriser les transactions ON-LINE avec la carte (liaison (1)) et de diffuser une clé vers un terminal (liaison (3)) ou vers une carte (liaison (1)).
- Enfin, le module de sécurité associé au terminal (MST) permet de sécuriser les transactions OFF-LINE entre la carte et le terminal (liaison (2)).

2.2.3. Les intervenants

Un intervenant est une entité qui définit des clés nécessaires à l'exploitation d'un parc de cartes. On distingue quatre types d'intervenants :

- l'encarteur,
- l'émetteur de la carte,
- les émetteurs de répertoires applications,
- les émetteurs de répertoires services.

Chaque intervenant possède un réseau de modules de sécurité permettant de contrôler l'exploitation des clés qu'il a définies.

2.2.3.1. L'encarteur

L'encarteur définit deux clés :

- la clé de fabrication qui est confiée au fabricant, écrite dans la carte par ce dernier, puis utilisée par l'encarteur pour écrire la clé de personnalisation,
- la clé de personnalisation qui est écrite par l'encarteur puis confiée à l'émetteur de la carte qui doit la présenter pour écrire les clés qu'il a définies.

L'encarteur utilise quatre modules :

- un MCH pour générer ses deux clés,
- un MSC contenant la clé de fabrication maître. Ce module est confié au fabricant afin de lui permettre d'écrire les clés de fabrication diversifiées dans les cartes,
- un MSC contenant les clés de fabrication et de personnalisation maîtres. L'encarteur conserve ce "coffre-fort" et l'utilise pour écrire les clés de personnalisation diversifiées dans les cartes,
- un MSC contenant la clé de personnalisation maître. Il est confié à l'émetteur de cartes afin que celui-ci puisse personnaliser les cartes.

2.2.3.2. Les émetteurs de niveau carte, application et service :

Chaque émetteur définit cinq groupes de clés propres à son niveau :

- la clé d'émetteur primaire,
- la ou les clés d'émetteurs secondaires,
- la ou les clés d'authentification,

- la ou les clés d'effacement,
- la ou les zones MAC.

Chaque émetteur utilise trois modules :

- un MCH pour générer les clés qu'il a définies,
- un module de personnalisation (MSC) contenant les clés qu'il a générées. Le MSC de l'émetteur de niveau carte contient en plus la clé de personnalisation transmise par l'encarteur. L'émetteur conserve ce module pour personnaliser les cartes.
- les modules de contrôle du parc de cartes en utilisation (MSC ou MST). Ils contiennent les clés que l'émetteur a définies, et sont répartis dans les serveurs et terminaux du système.

2.2.4. Fonctionnalités particulières des modules

Le jeu d'instructions des modules est plus étendu que celui des cartes. Les instructions spécifiques aux modules utilisées dans les procédures de sécurité vues au paragraphe 2.1.11.3. sont décrites brièvement ci-dessous.

2.2.4.1. Sélection de clé de chiffrement

Cet ordre permet d'indiquer au module la référence de la clé du répertoire courant qu'il doit utiliser, sous une forme diversifiée ou non, pour une des opérations suivantes :

- le calcul de message d'authentification,
- le calcul de message d'écriture sécurisée ou d'effacement,
- le contrôle de certificat,
- le calcul de message de transfert de clé.

Cette instruction doit être précédée de la sélection du répertoire dans lequel la clé est mémorisée.

2.2.4.2. Fourniture de paramètre de calcul

Cet ordre permet de communiquer au module le paramètre à mettre en oeuvre pour

- calculer un message d'authentification d'un terminal ou d'un émetteur; le paramètre de calcul est alors le nombre aléatoire généré par la carte,
- calculer un message d'authentification d'un porteur; le paramètre de calcul est alors le nombre aléatoire généré par

la carte appliqué à la valeur du PIN par un 'OU exclusif' (N XOR PIN),
- calculer un message d'écriture sécurisée; le paramètre est alors le nombre aléatoire généré par la carte,
- calculer un message d'effacement sécurisé; le paramètre est alors le nombre aléatoire généré par la carte,
- calculer un certificat; le paramètre de calcul est alors la paramètre interne mis en jeu par la carte.

2.2.4.3. Calcul de message d'authentification

Cet ordre permet à une autorité habilitée d'obtenir le message grâce auquel elle pourra s'authentifier par rapport à une carte. Le module met en oeuvre la clé désignée par le précédent ordre de sélection de clé de chiffrement et le nombre aléatoire transmis dans le dernier ordre de fourniture de paramètre de calcul. Le module renvoie automatiquement le message.

2.2.4.4. Calcul de message d'écriture sécurisée et de message d'effacement sécurisé

Cet ordre permet à une autorité habilitée d'obtenir le message qui lui permettra de réaliser l'écriture sécurisée d'un mot ou d'un descripteur, ou l'effacement sécurisé d'une chaîne de mots dans une carte.
Le module met en oeuvre la clé désignée par le précédent ordre de sélection de clé de chiffrement et le nombre aléatoire transmis dans le dernier ordre de fourniture de paramètre de calcul.
Une lecture de résultat permet d'obtenir le message chiffré.

2.2.4.5. Calcul de certificat

Cet ordre permet au module de recalculer le certificat émis par la carte.
Le module met en oeuvre la clé désignée par le précédent ordre de sélection de clé de chiffrement et le paramètre interne de la carte transmis dans le dernier ordre de fourniture de paramètre de calcul.

2.2.4.6. Comparaison de certificat

Cet ordre permet au module de comparer le certificat émis par une carte à celui qu'il a recalculé par un ordre de calcul de certificat.
Le module reçoit comme argument le certificat calculé par la carte.

2.3. Exemples d'utilisation des cartes TB10 et TB100

2.3.1. La carte Bergamo en Italie

Une carte multi-émetteurs est étudiée actuellement pour la ville de Bergamo et la région de Lombardia en Italie. Elle devrait permettre de remplacer deux cartes mono-émetteurs implémentées sur masque M9 : d'une part une carte contenant toutes les données anagraphiques d'une famille utilisée par les habitants pour obtenir des papiers d'état civil, et d'autre part la carte bancaire OSCAR.
La carte Bergamo est composée de trois applications :

- l'application 'commune' contenant les données anagraphiques,
- l'application bancaire Oscar,
- l'application 'prépayement' permettant de payer des services tels que le parking, l'accès à la partie historique de la ville...

2.3.2. Postomat en Suisse

La carte Postomat développée en Suisse pour la ville de Bielle permet :

- d'utiliser les services municipaux,
- d'effectuer des paiements dans les commerces,
- de téléphoner.

2.3.3. L'Université de Rome

L'Université de la Sapienza à Rome distribue à chaque étudiant une carte permettant :

- d'accéder aux examens,
- de gérer les diplômes et autres documents scolaires,
- d'accéder à la bibliothèque,
- d'accéder au restaurant universitaire,
- et prochainement d'assurer les services bancaires.

2.3.4. La carte professionnelle de santé

Cette carte est étudiée dans le cadre d'un projet orchestré par le Ministère de la Santé et des Affaires Sociales en France. Les principaux objectifs de ce projet visent à remplacer la feuille de soin (700 millions d'exemplaires par an) par une

transaction signée, et à faciliter les remboursements. Deux cartes seront mises en circulation : la carte d'assuré social (carte Vital ou autre) et la carte professionnelle de santé (CPS).

La carte de l'assuré est une carte mono-émetteur de type M9. Elle contient des informations concernant l'assuré et permet notamment de l'identifier.

La carte professionnelle de santé est détenue par le prestataire de services (médecin, pharmacien, infirmier...). Il s'agit d'une carte d'habilitation qui permet au professionnel d'exploiter la carte de l'assuré. La nature des informations lisibles est fonction du critère d'accès propre au type de profession (un généraliste n'a pas accès aux mêmes informations qu'un spécialiste). Cette carte est implémentée sur un masque multi-émetteurs TB100 contenant au moins les deux applications suivantes :

- le contrôle d'accès télématique au fichier de sécurité sociale,
- la description du niveau d'accès à la carte de l'assuré.

La rédaction de la feuille de soin est remplacée par la procédure suivante : lors de la consultation, le prestataire introduit sur son terminal les informations caractérisant les soins ou/et produits délivrés. Les identités respectives des deux parties sont lues sur chaque carte et chacun 'signe' la transaction en présentant son PIN. En fin de journée, un serveur collecte toutes les 'feuilles de soins' ainsi rédigées et les traite afin d'effectuer les remboursements.

Chapitre 3 : Description de la carte MCOS

3.1. Introduction

La carte à micro-calculateur MCOS (Multi Application Chip Operating System) de Gemplus est une carte multi-émetteurs équipée d'un composant à mémoire EEPROM. Le noyau de base de son système d'exploitation est celui de la carte mono-émetteur COS du même constructeur. Ce noyau est enrichi de fonctions exploitant la structure multi-répertoires de MCOS, l'effacement de sa mémoire EEPROM et son algorithme DES.
Le noyau de base commun permet d'assurer une certaine compatibilité entre les produits COS et MCOS. En effet, un répertoire (racine ou application) d'une carte MCOS offre les mêmes possibilités qu'une carte COS. Une application implémentée sur une carte COS n'exploitant que son noyau de base peut donc être portée sans difficulté dans un répertoire de la carte multi-émetteurs.
Certaines instructions de la carte MCOS existent en version simple et en version étendue. La version simple est directement compatible avec les cartes COS EPROM et la version étendue exploite pleinement les possibilités de la carte MCOS.

3.1.1. Philosophie de Gemplus

La philosophie de Gemplus est d'offrir un code ROM simple, ouvert, voire même rustique auquel on peut ajouter des sous-programmes afin de satisfaire à une application particulière. L'objectif de Gemplus est donc de faciliter le prototypage et de toucher des petits marchés sans investir immédiatament dans l'élaboration d'un code ROM.
Cette souplesse n'est possible qu'au détriment de la sécurité. La possibilité d'ajouter des sous-programmes augmente en effet les risques de fraude au niveau du masque.

3.1.2. Quelques mots à propos de la carte COS

Il existe actuellement plusieurs versions de la carte COS :

- la version de base est implémentée sur un composant de capacité réduite (1 K octets EPROM, 2 K octets ROM),
- une version plus étendue (par des instructions supplémentaires) est implémentée sur un composant doté de mémoires de plus grande capacité (4 K octets EPROM, 3 K octets ROM),
- la version avec DES est également implémentée sur le composant doté d'une mémoire EPROM de 4 K octets. L'algorithme DES fait partie de la bibliothèque de sous-programmes de Gemplus. Il est codé en EPROM, ce qui réduit l'espace disponible pour l'application qu'elle contient. Le DES permet d'ajouter des instructions chiffrées au noyau de base,
- une dernière version de la carte COS est implémentée sur un composant à mémoire effaçable (2K EEPROM). Ce type de composant permet de réécrire dans une zone.

La mémoire des cartes COS est divisée en 5 grandes parties :

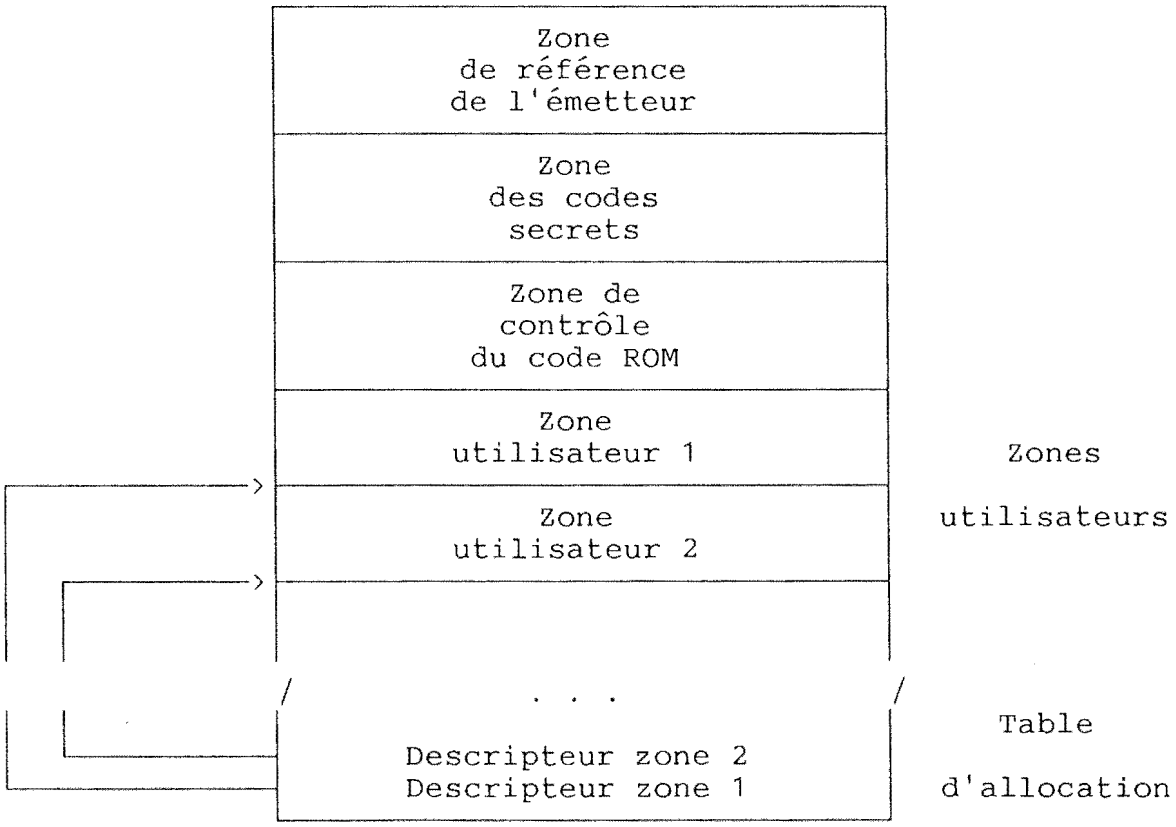


figure 17 : Structure des cartes COS

- La zone de référence de l'émetteur contient
  . la référence de la carte, de l'émetteur et de l'application,
  . le code secret de fabrication et le code secret de l'émetteur.
- La zone des codes secrets contient 7 codes secrets de base et 2 codes de substitution qui permettent de remplacer 2 des 7 codes de base.
- La zone de contrôle du code ROM contient les paramètres de fonctionnement du code ROM, le mécanisme de ratification des codes secrets, et éventuellement les adresses des sous-programmes ajoutés en EPROM.
- La majeure partie de la mémoire contient les zones utilisateurs (au maximum 63). Une zone utilisateur est soit un fichier de données soit un sous-programme. Chacune est constituée de 1 à 256 mots de 4 octets. Les fichiers sont protégés indépendamment les uns des autres en écriture et en lecture par les 7 codes secrets de base.
  Cette partie s'étend à partir d'une adresse fixe, dans le sens croissant des adresses, jusqu'à saturation de l'EPROM.
- La table d'allocation contient les descripteurs des zones utilisateurs. Un descripteur est ajouté à la table lors de chaque nouvelle allocation de zone. Les descripteurs sont insérés dans le sens décroissant des adresses, à partir des adresses hautes de la mémoire, jusqu'à saturation de celle-ci. Les descripteurs contiennent les caractéristiques des zones (notamment les numéros de codes secrets protégeant la lecture et l'écriture) et l'adresse de la zone suivante.

Le tableau suivant indique les instructions disponibles dans les différentes cartes COS :

	COS 1K EPROM	COS 4K EPROM	COS 2K EEPROM	COS DES
Noyau de base				
Remise à zéro	X	X	X	X
Ecriture d'un mot mémoire	X	X	X	X
Ecriture zone utilisateur	X	X	X	X
Création d'un descripteur	X	X	X	X
Modification d'un descripteur	X	X	X	X
Lecture d'un mot mémoire	X	X	X	X
Lecture zone utilisateur	X	X	X	X
Lecture table d'allocation	X	X	X	X
Présentation d'un code secret	X	X	X	X
Chargement d'un code secret	X	X	X	X
Substitution d'un code secret	X	X	X	X
Annulation d'un code secret	X	X	X	X
Réhabilitation code secret	X	X	X	X
Instructions particulières				
Branchement sous-programme	X	X		
Consommation en mode jeton		X		X
Récherche sur argument		X		X
Positionnement sur une zone		X	X	X
Checksum d'une zone		X		X
Statut de la carte		X	X	X
Réécriture dans une zone			X	
Sélection clé de session				X
Calcul clé de session				X
Présentation chiffrée de code				X
Lecture chiffrée				X
Ecriture chiffrée				X

figure 18 : Fonctionnalités des cartes COS

3.2. Présentation physique de la carte MCOS

Le principe d'encartage (composant et bouton de contact) est identique à celui utilisé chez BULL CP8, les contacts sont également disponibles en position haute ou basse.
Le composant de la carte MCOS est constitué des éléments suivants :

- microprocesseur MOTOROLA 6805 8 bits,
- ROM 6 K octets,
- RAM 160 octets,
- EEPROM 2 K octets.

L'interface physique avec le monde extérieur a les mêmes caractéristiques que celles de la carte TB100.

3.3. Description de la mémoire EEPROM de la carte MCOS

3.3.1. Organisation logique de la mémoire EEPROM

L'EEPROM est divisée logiquement en une structure arborescente multi-répertoires à deux niveaux. Un premier niveau correspond à l'ensemble de la carte : le répertoire carte (répertoire racine). Il est partagé en entités du deuxième niveau : les répertoires application. Leur nombre maximum est de 31.
Les répertoires (racine et application) sont constitués de fichiers (zones) contenant des données ou des clés secrètes. Le masque est prévu pour gérer 255 fichiers mais la taille actuelle de l'EEPROM limite à 160 le nombre total de descripteurs.
Tous les répertoires sont indépendants et peuvent être associés à des applications différentes. Chaque répertoire application est contrôlé par son propre bloc de sécurité et/ou par le bloc de sécurité du niveau carte.
Le système de sécurité est assez complexe et peut être configuré de multiples manières. Dans sa configuration par défaut, le bloc de sécurité du niveau carte délivre des autorisations globales et les blocs de sécurité associés aux répertoires applications délivrent des autorisations propres aux répertoires.

Le schéma ci-dessous illustre cette organisation.

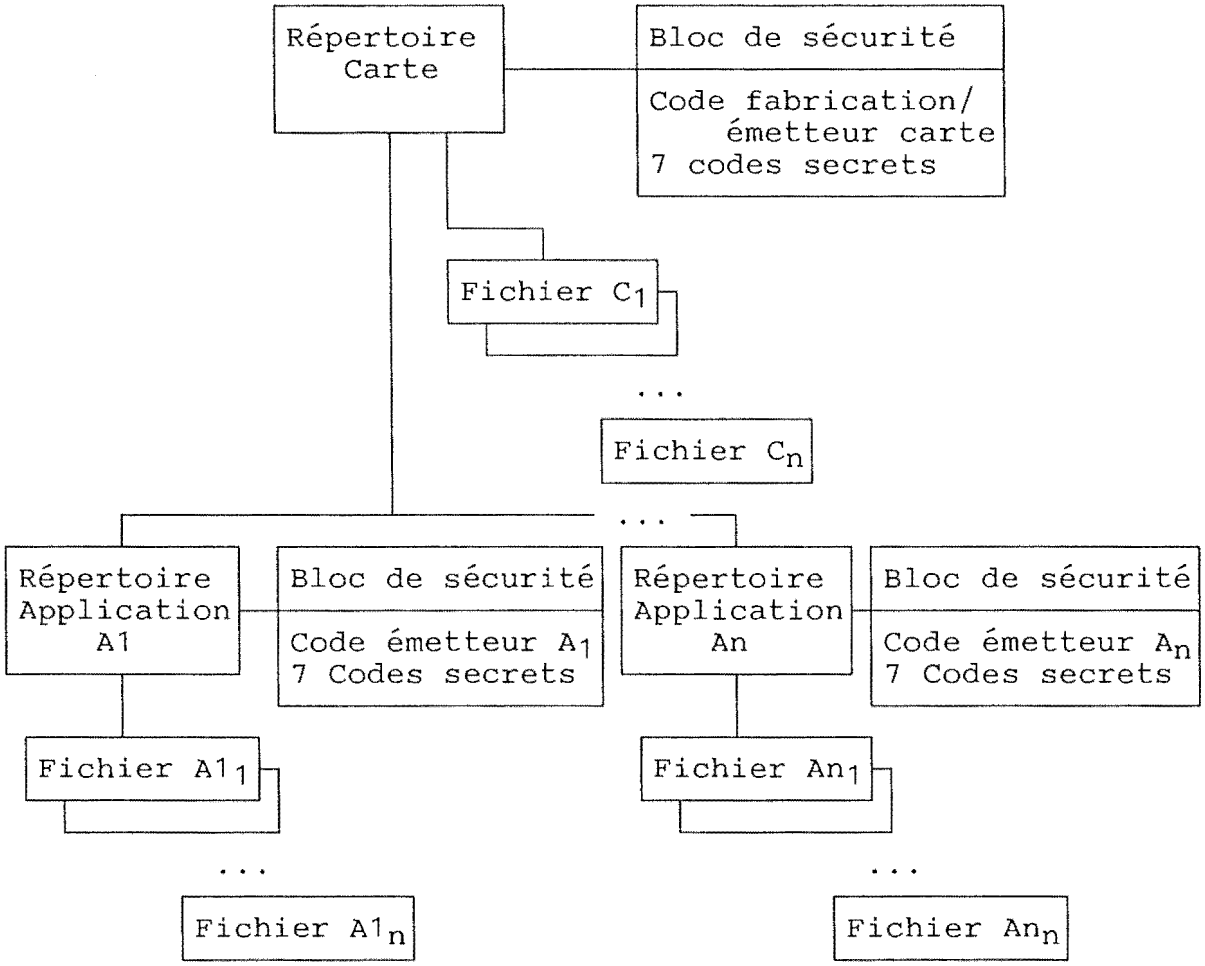


figure 19 : Organisation de l'EEPROM de la carte MCOS

3.3.2. Structure physique de la mémoire EEPROM

L'EEPROM est constituée de mots de 32 bits. Elle est structurée de la manière suivante :

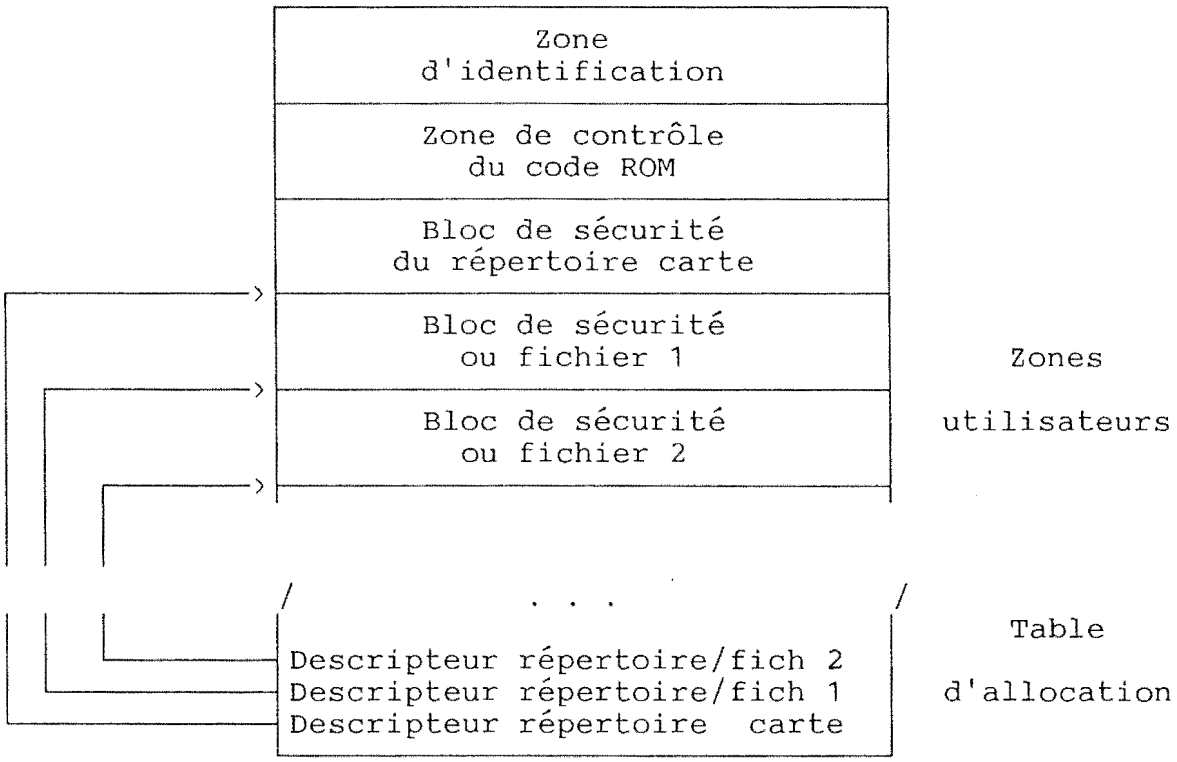


figure 20 : Mapping de la mémoire EEPROM de la carte MCOS

3.3.2.1. La zone d'identification

Cette zone contient

- la référence de la carte (numéro de série),
- la référence de l'émetteur de la carte.

3.3.2.2. La zone de contrôle du code ROM

- L'octet des verrous indique la phase de vie de la carte :
  . verrou de fabrication : carte en cours de fabrication/ fin de la fabrication,
  . verrou de personnalisation : carte non personnalisée/ carte personnalisée.

  Cet octet contient également un bit indiquant si le jeu d'instructions cryptographiques est contrôlé par code secret.
- L'adresse du filtre éventuel : le filtre est un programme qui peut être inséré en EEPROM afin d'étendre et de transformer le masque initial. Mis en oeuvre lors de l'exécution de chaque

instruction du masque initial, il permet de modifier le déroulement des instructions et d'en ajouter de nouvelles.

- Les options du code ROM définissent les différents paramètres du système d'exploitation de la carte. Les mécanismes concernés par ces options sont détaillés tout au long de ce chapitre.
  . 'masque statut clé de session' : si ce masque vaut 1, le statut de clé de session est conservé lors d'un changement de répertoire,
  . 'masque statut authentification' : si ce masque vaut 1, le statut d'authentification est conservé lors d'un changement de répertoire,
  . substitution standard de code secret/substitution étendue de code secret,
  . condition de modification de fichier : la modification de fichier dépend
    * soit des conditions d'accès en écriture (Cr) et en réécriture (Cu),
    * soit de la condition de création et de modification de fichier (Cf),
  . condition de création de fichiers : la création de fichier dépend
    * soit uniquement de la condition de création et de modification de fichier (Cf) du répertoire courant,
    * soit de la présentation du code 0.

3.3.2.3. Le bloc de sécurité du répertoire carte

L'adresse de ce bloc est écrite dans le descripteur du répertoire carte lors de la fabrication. Sa valeur par défaut est l'adresse du mot suivant la zone de contrôle du code ROM. Cette adresse peut être modifiée en cours de personnalisation afin d'ajouter le code exécutable du filtre après la zone de contrôle du code ROM.

La structure de ce bloc est la suivante :

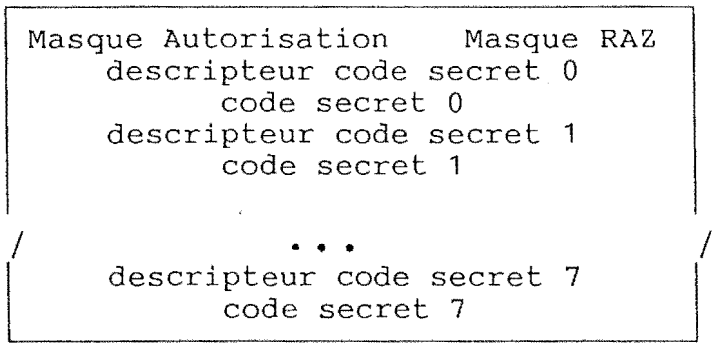


figure 21 : Structure d'un bloc de sécurité

- Le masque d'autorisation et le masque de RAZ ont chacun une longueur de deux octets. Leurs rôles seront détaillés au paragraphe 3.3.
- Le code secret 0 (4 octets) contient en premier lieu le code secret de fabrication, chargé au cours de la fabrication afin de sécuriser la carte entre le fabricant et l'émetteur.
  Au cours de la personnalisation, ce code est substitué par le code secret de l'émetteur de la carte.
- Les quatorze mots suivants contiennent les sept autres codes secrets et leurs descripteurs respectifs. Ces codes sont utilisés pour protéger les différents types d'accès aux fichiers ainsi que les créations dans la carte.
  Un code secret est toujours présenté à la carte sur 8 octets, mais seuls les quartets de poids faible de chaque octet sont mémorisés dans le mot correspondant du bloc de sécurité.

3.3.2.4. Les zones utilisateurs

Une zone utilisateur est soit un fichier de données appartenant à un répertoire (carte ou application), soit un bloc de sécurité attaché à un répertoire application. Ces zones sont allouées dans le sens croissant des adresses, à la suite du bloc de sécurité de la carte.
Les caractéristiques de chaque fichier sont écrites lors de sa création dans un descripteur alloué dans la table d'allocation.
Un bloc de sécurité est automatiquement alloué lors de la création d'un répertoire contrôlé par son propre système de sécurité. Les blocs de sécurité des répertoires application ont la même structure que celui du répertoire carte, ils ne contiennent pas de masque de RAZ.

3.3.2.5. La table d'allocation

Cette zone permet de maintenir une cartographie de l'EEPROM ainsi que l'information sur les protections attachées aux différents fichiers et répertoires. Elle contient les descripteurs des répertoires et des fichiers. Les descripteurs sont alloués dans le sens décroissant des adresses mémoire.
MCOS est prévu pour gérer un maximum de 255 descripteurs mais la taille actuelle de l'EEPROM (2 K octets) limite ce nombre à 160 (cas d'un seul répertoire constitué de fichiers d'un seul mot).

3.3.2.6. La zone de test

Cette zone est libre d'accès et permet de réaliser des tests de lecture, écriture et réécriture pendant toute la vie de la carte.

3.3.3. Les descripteurs de répertoires

La création de nouveaux répertoires est contrôlée par le code de fabrication au cours de la personnalisation et par le code de l'émetteur de la carte durant la phase d'utilisation.
Les descripteurs de répertoire contiennent les informations suivantes :

- l'identifiant du répertoire (1 à 254),
- le type de descripteur : descripteur de répertoire,
- le numéro de répertoire (0 à 31). Ce numéro est alloué automatiquement par la carte (incrémenté lors de chaque création). Il doit être indiqué dans les descripteurs de fichiers pour les rattacher à leurs répertoires respectifs.
- Cc : la condition d'accès en création et réhabilitation de codes secrets,
- Cf : la condition d'accès en création et modification de fichier,
- un indicateur mentionnant si le répertoire est créé avec ou sans bloc de sécurité.
  Si un bloc de sécurité est demandé, 17 mots lui sont alloués lors de la création du répertoire, sinon, le répertoire créé est automatiquement rattaché au bloc de sécurité du répertoire carte.
- un indicateur mentionnant si la réécriture est ou n'est pas disponible,
- l'adresse (exprimée en mots) de la zone suivante,

- le bit de parité de l'adresse,
- le bit de validation du descripteur : il est positionné par MCOS après vérification de la bonne écriture du descripteur. Si le descripteur est mal écrit, une lecture de la table d'allocation indique que le descripteur est invalide.

Une condition d'accès (Cc ou Cf) est exprimée dans un quartet. Sa valeur (0 à 15) correspond au poids du bit du registre d'autorisation devant être positionné pour autoriser l'action correspondante. Ce registre est affecté lors de la présentation de chaque code secret. Une condition à 0 indique que l'accès est libre.

3.3.4. Les descripteurs de fichiers

Les descripteurs de fichiers contiennent les informations suivantes :

- le type du fichier (1 à 255),
- la nature du fichier : fichier standard ou fichier pouvant contenir des clés secrètes,
- le type de descripteur : descripteur de fichier,
- le numéro de répertoire auquel est rattaché le fichier,
- Cr : la condition d'accès en lecture,
- Cw : la condition d'accès en écriture,
- Cu : la condition d'accès en réécriture et effacement,
- le verrou Lu : son positionnement permet d'interdire la réécriture et l'effacement dans le fichier,
- le verrou Lw : son positionnement permet d'interdire l'écriture dans le fichier,
- le verrou Lr : son positionnement permet d'interdire la lecture dans le fichier,
- le verrou La : il indique si les accès au fichier doivent être chiffrés ou non,
- l'adresse (exprimée en mots) de la zone suivante,
- le bit de parité de l'adresse,
- le bit de validation du descripteur : ce bit est positionné par le MCOS de la même manière que pour les descripteurs de répertoires.

Les valeurs des conditions d'accès (comprises entre 0 et 15) déterminent le poids du bit du registre d'autorisation devant être positionné pour que l'accès correspondant soit autorisé.
Une autorisation en réécriture et effacement libère également

l'accès en écriture et en lecture. Une autorisation en écriture libère l'accès en lecture.

Le type de fichier peut prendre toute valeur comprise entre 1 et 255. Certaines fonctions dépendent de cette valeur :

. [1 à 127] : les valeurs des conditions et des verrous d'accès sont fournies lors d'une lecture de la table d'allocation,

. [248 à 255] : une instruction de checksum peut être effectuée sur le fichier considéré.

La longueur du fichier est comprise entre 4 et 1024 octets. Elle est spécifiée lors de sa création mais n'est pas indiquée dans son descripteur. Elle est déductible de l'adresse du début du fichier écrite dans le descripteur précédent, et de l'adresse de la zone suivante spécifiée dans le descripteur alloué ici.

3.3.5. Les descripteurs de codes secrets

La structure des huit descripteurs est identique. Chacun contient les informations suivantes :

- le bit de validation du descripteur : indique si le code n'est pas encore initialisé ou s'il est initialisé et validé,
- le bit d'annulation du code,
- le type de présentation du code : chiffré ou en clair,
- l'octet de ratification,
- la valeur d'autorisation (2 octets),

La valeur d'autorisation est appliquée au registre d'autorisation par un 'OU logique' lors d'une présentation correcte du code. Elle détermine donc les bits du registre qui sont basculés suite à la présentation du code.

L'annulation d'un code secret est provoquée par une instruction spécifique. Elle peut entraîner l'interdiction d'accès (en écriture et/ou en lecture et/ou en effacement et réécriture) à tous les fichiers protégés par ce code secret si sa valeur d'autorisation n'a pas d'équivalent.

L'octet de ratification est utilisé comme suit :

- les 3 bits de poids faible pour la ratification des instructions de présentation, d'annulation et de substitution du code,

- les 3 bits de poids fort pour la ratification de l'instruction de réhabilitation du code.

La ratification est donc ici systématique et il n'existe plus de zone de ratification (zone d'accès) globale comme dans les cartes COS EPROM. Ce système permet d'économiser un grand nombre d'octets sans pour autant nuire à la sécurité du produit.
Le code secret 0 est ratifiable et n'est verrouillé, comme les codes 1 à 7, qu'après trois présentations fausses successives.

3.3.6. Le système de sécurité

Chaque action dans un répertoire est contrôlée par son système de sécurité. Ce système est composé

- du bloc de sécurité associé au répertoire,
- du descripteur du répertoire,
- des statuts de clé de session et d'authentification,
- du registre d'autorisation.

Lorsqu'un répertoire est créé sans bloc de sécurité, il est sous contrôle de celui de la carte.
Le système de sécurité d'un fichier est composé du système de sécurité du répertoire auquel il appartient et des conditions d'accès écrites dans le descripteur du fichier.

3.3.6.1. Le statut de clé de session

Ce statut est affecté lorsque la carte a calculé une clé de session. Cette clé est mise en oeuvre par toutes les instructions chiffrées exécutées au cours d'une session.
Elle est calculée par l'exécution de l'instruction de 'sélection de clé' suivie directement de l'exécution de l'instruction de 'calcul de clé de session'.

Sélection d'une clé :

Cet ordre permet de sélectionner une clé secrète Ks destinée à calculer la clé de session. Les clés secrètes sont mémorisées dans des fichiers interdits en écriture et en lecture. Le terminal fournit à la carte le numéro de fichier (n°) et la position (Po) dans ce fichier où se trouve la clé secrète Ks, ainsi qu'un nombre aléatoire N.
La carte commence par désactiver les statuts de session et d'authentification. Elle calcule ensuite un cryptogramme (avec

la fonction directe du DES) qu'elle conserve en RAM jusqu'à l'exécution de l'instruction de calcul de clé de session.

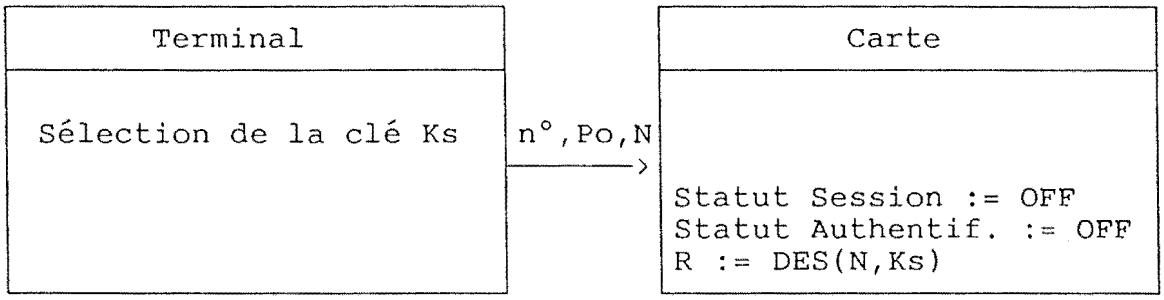


figure 22 : Sélection d'une clé

Calcul de clé de session :

Cet ordre permet de calculer une clé de session S en chiffrant le cryptogramme R (calculé lors de la sélection de clé) et un nombre aléatoire N' (généré par la carte) avec la clé Ks sélectionnée. Si l'instruction se termine avec succès, le statut 'clé de session' est basculé. La clé de session est mémorisée en RAM.

La clé Ks est également mémorisée dans le module de sécurité du terminal, ce dernier pourra donc calculer la clé de session en utilisant le cryptogramme R et le nombre aléatoire N' renvoyés par la carte.

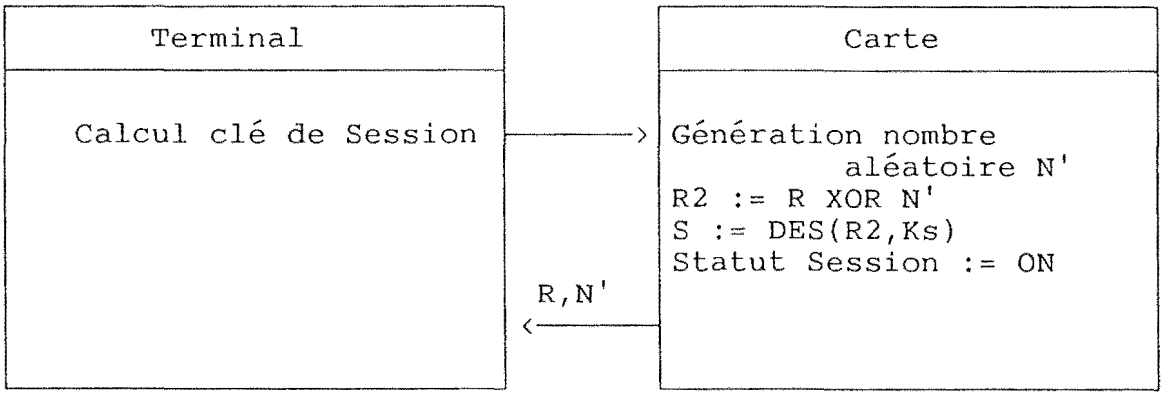


figure 23 : Calcul d'une clé de session

3.3.6.2. Le statut d'authentification

Ce statut est affecté lorsque le terminal est authentifié par la carte. Cette authentification est réalisée par une présentation chiffrée d'un code secret :

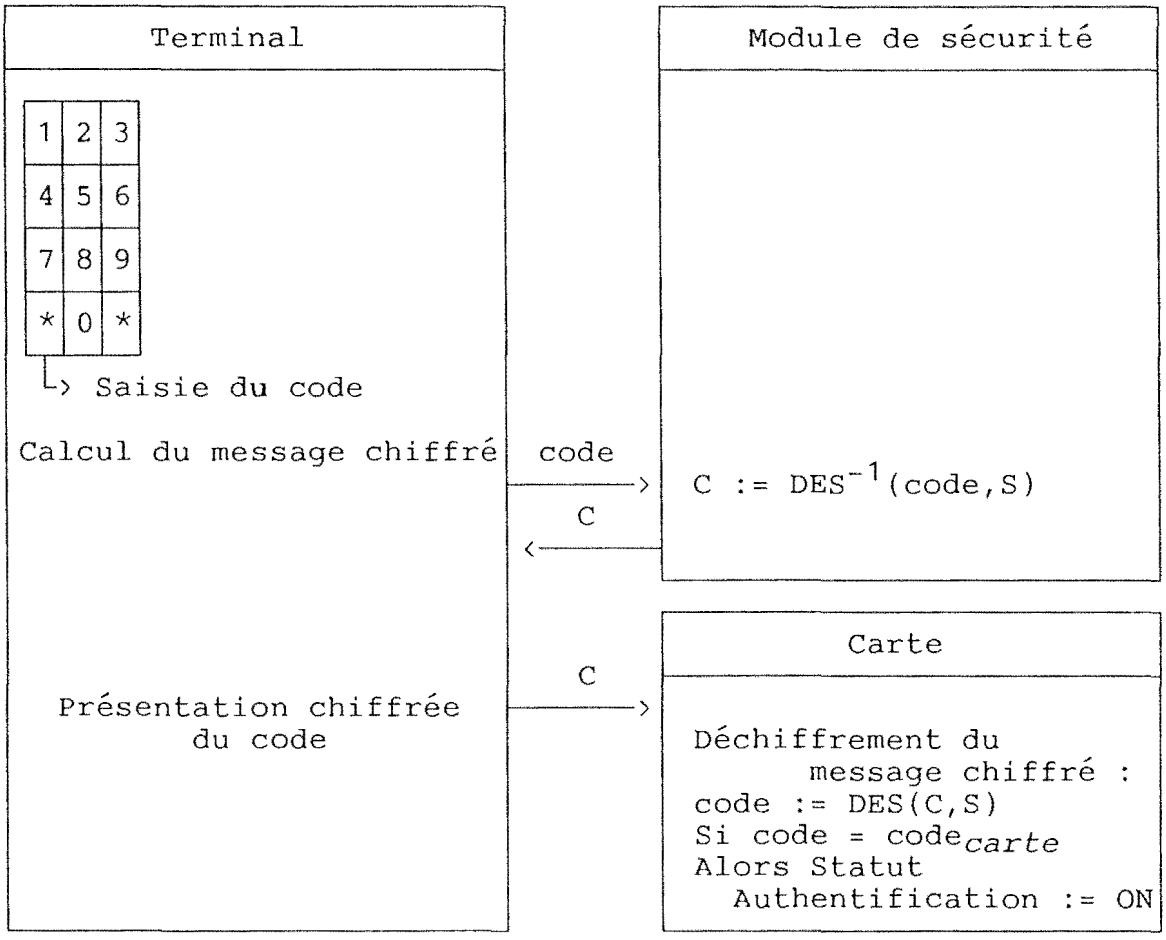


figure 24 : Présentation chiffrée d'un code secret

Lors d'un changement de répertoire, si le nouveau répertoire sélectionné est sous le contrôle d'un bloc de

sécurité différent de celui de l'ancien répertoire courant, le 'masque statut clé de session' et le 'masque statut authentification' sont respectivement appliqués aux statuts d'authentification et de clé de session par un 'ET logique'. Si l'un de ces masques a une valeur nulle, le statut correspondant est remis à zéro.
Lorsque des autorisations liées à des présentations chiffrées ont été mémorisées dans le registre d'autorisation, la remise à zéro du statut d'authentification invalide ces autorisations. Seules les autorisations liées aux présentations en clair sont maintenues.
La valeur du masque du statut d'autorisation détermine donc s'il y a héritage de droits lors d'un changement de répertoire.

3.3.6.3. Le registre d'autorisation

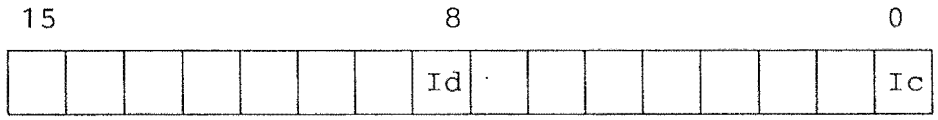


figure 25 : Le registre d'autorisation

Ce registre de 16 bits est remis à zéro lors de l'initialisation de la carte. Son contenu est activé lors de la présentation correcte d'un code secret, par l'application de la valeur d'autorisation du code au registre par un 'OU logique'. Les bits ainsi activés autorisent certaines actions protégées par les conditions d'accès correspondantes.

Les bits Ic et Id ont des fonctions particulières; elles seront vues lors de la description des instructions.

3.3.6.4. Les valeurs d'autorisation

La valeur d'autorisation d'un code secret est écrite dans le descripteur du code lors de sa création. Elle définit les positions des bits qui seront activés dans le registre d'autorisation après une présentation correcte du code.
La valeur d'autorisation du code secret 0 du répertoire carte est initialisée lors de la fabrication à la valeur '0101'$_H$. Sa présentation correcte permet donc de basculer les bits Ic et Id du registre d'autorisation, délivrant ainsi des autorisations particulières. L'émetteur a la possibilité de modifier cette valeur au cours de la personnalisation par une instruction

'd'écriture mémoire'.

Remarques :

- Un accès à une zone peut être protégé par différents codes secrets. Il faut pour cela donner les mêmes valeurs d'autorisation à des codes secrets différents.
- Un même code secret peut permettre d'accéder à des zones ayant des protections d'accès différentes. Il faut pour cela donner une valeur d'autorisation plus étendue à ce code secret.

3.3.6.5. Les masques d'autorisation

Chaque bloc de sécurité contient un tel masque. Il permet de privilégier le rôle du code 0 en interdisant qu'un autre code (1 à 7) génère les mêmes autorisations.

Un bit positionné dans ce masque signifie que le bit de même poids peut être utilisé dans les valeurs d'autorisation des codes secrets 1 à 7 du bloc de sécurité. Si cette condition n'est pas respectée lors du chargement d'un code secret, l'instruction est refusée.

Le masque d'autorisation du bloc de sécurité du répertoire carte est initialisé lors de la fabrication à la valeur $'FEFE'_H$, interdisant le positionnement des bits Id et Ic par la présentation des codes 1 à 7. L'émetteur de la carte peut modifier cette valeur au cours de la personnalisation par une instruction 'd'écriture mémoire'.

Un code standard (code 1 à 7) peut avoir des fonctions identiques à celles du code émetteur. Il faut pour cela modifier le masque d'autorisation de manière à ce qu'il permette d'activer des autorisations de type émetteur.

3.3.6.6. Le masque de RAZ

Ce masque est contenu dans le bloc de sécurité du répertoire carte uniquement.

Lors d'un changement de répertoire, si le nouveau répertoire sélectionné est contrôlé par un système de sécurité différent de celui qui contrôlait le dernier répertoire courant, le masque de RAZ est appliqué par un 'ET logique' au registre d'autorisation. Un bit nul dans ce masque signifie que le bit correspondant du registre d'autorisation est dans ce cas remis à zéro.

Le masque est initialisé lors de la fabrication à la valeur $'00FF'_H$. Cette valeur permet l'héritage de droit du répertoire

carte vers les répertoires application. Il peut recevoir une nouvelle valeur au cours de la personnalisation afin de modifier ou de supprimer cet héritage.

3.4. Fonctionnalités

3.4.1. Le dialogue carte-terminal

Le dialogue carte-terminal est conforme à la norme ISO 7816-3 abordée au paragraphe 2.1.11.1.

3.4.2. Les instructions de base

a) La remise à zéro

La remise à zéro est provoquée physiquement par l'application d'un signal sur le contact 'RAZ'. Les octets 'historiques' renvoyés par la carte apportent l'information suivante :

- la référence du composant,
- la référence du code ROM,
- la version du code ROM,
- la référence de l'émetteur et de l'application,
- le nombre de répertoires créés dans la carte,
- l'état de la ratification pour les 8 codes secrets du répertoire carte,
- l'octet des verrous,
- les mots d'état ME1 et ME2.

Après une remise à zéro, le répertoire courant est le répertoire carte et le pointeur de fichier courant vaut 0 pour indiquer qu'aucun fichier n'est sélectionné.

b) Statut de la carte

Arguments :

P1 : '00'
P2 : '00', '01' ou '02'
P3 : longueur des données renvoyées (14 ou 32 octets)

Cet ordre renvoie des caractéristiques techniques relatives à l'état de la carte et du répertoire courant. La nature des informations fournies est fonction de la valeur du paramètre P2.

cas 1 :

- numéro de série (référence de la carte),

- nombre de fichiers alloués dans le répertoire courant,
- nombre d'octets de l'EEPROM non alloués,
- numéros des codes secrets (du bloc de sécurité courant) présentés faux une fois,
- numéros des codes secrets (du bloc de sécurité courant) présentés faux deux fois,
- numéros des codes secrets (du bloc de sécurité courant) présentés faux trois fois.

cas 2 :

La carte renvoie les descripteurs des 8 codes secrets du bloc de sécurité du répertoire courant. Le bit Id du registre d'autorisation doit dans ce cas être positionné.

cas 3 :

La carte renvoie la zone de contrôle du code ROM et le masque de RAZ. Le bit Ic du registre d'autorisation doit dans ce cas être positionné.

c) Ecriture mémoire

Arguments :

P1-P2 : adresse en octets du premier octet à écrire
P3 : longueur des données à écrire (1 à 32 octets)
D1-Dn : données à écrire

Utilisé principalement pendant la phase de personnalisation, cet ordre est le seul permettant d'écrire la référence de l'émetteur de la carte et le contenu de la zone de contrôle du code ROM.
L'espace mémoire accessible en adressage direct à l'aide de cet ordre dépend de la phase de vie de la carte relatée par l'octet des verrous, et du bit Ic du registre d'autorisation : tant que la phase de personnalisation n'est pas terminée, et à condition que le bit Ic du registre d'autorisation vaille 1, il est possible d'écrire dans toute la mémoire sauf dans les deux premiers mots qui contiennent la référence de la carte.
Trois mots particuliers sont toujours accessibles en adressage indirect :

- les deux mots de test,
- le mot des verrous.

d) Lecture mémoire

Arguments :

P1-P2 : adresse en octets du premier octet à lire

P3 : longueur des données à lire (1 à 32 octets)

Utilisé principalement pendant la personnalisation, cet ordre permet notamment de lire la zone de contrôle du code ROM.

Lorsque les conditions d'accès en adressage direct (cf ordre d'écriture mémoire) sont respectées, toute la mémoire est lisible. En dehors de ces conditions, seule la zone d'identification est accessible en lecture.

Trois mots particuliers sont toujours accessibles en adressage indirect :

- les deux mots de test,
- le mot des verrous.

3.4.3. Instructions de gestion des répertoires

a) Création d'un répertoire

P1-P2 : '0001'

P3 : longueur des données fournies (2 ou 8 octets)

D1-D2/D8 : données de création

Cette instruction permet de créer un nouveau répertoire. Elle ne fonctionne que si le bit Ic du registre d'autorisation est positionné.

Les deux premiers octets fournis à la carte sont :

- l'identifiant du répertoire (1 à 254), et
- l'octet contenant les options du répertoire créé :
  . bloc de sécurité associé/pas de bloc sécurité associé,
  . l'instruction de réécriture est interdite/l'instruction de réécriture est disponible.

Lorsque la longueur des données envoyées est limitée à deux octets, les valeurs suivantes sont chargées par défaut :

- la condition d'accès en création et réhabilitation de codes secrets : Cc = 8 pour imposer que le bit Id du registre d'autorisation soit positionné,
- la condition d'accès en création et modification de fichiers : Cf = 0 pour un accès libre,
- le masque d'autorisation : $'FE00'_H$,
- la valeur d'autorisation du code 0 : $'0100'_H$.

Lorsque la longueur des données envoyées est de 8 octets (version étendue), six octets supplémentaires sont fournis :

- la condition d'accès en création et réhabilitation de codes secrets (Cc : valeur entre 0 et 15),
- la condition d'accès en création et modification de fichier (Cf : valeur entre 0 et 15),
- le masque d'autorisation,
- la valeur d'autorisation du code 0.

Avant de créer et de valider le descripteur, MCOS vérifie que les données sont cohérentes et qu'il reste au moins 19 mots libres. Si un bloc de sécurité est demandé, MCOS y écrit le masque d'autorisation, la valeur d'autorisation du code 0 et une valeur nulle dans le code 0. Ce code 0 pourra être substitué par l'émetteur de la carte. Ce nouveau code sera fourni à l'émetteur de l'application en guise de code de fabrication du répertoire. L'émetteur de l'application pourra le substituer à son tour.

b) Sélection d'un répertoire

Arguments :

P1 : '00'
P2 : identifiant du répertoire (1 à 255)
P3 : longueur des données renvoyées par la carte (8 octets)

Cet ordre positionne le pointeur de répertoire courant et renvoie des informations relatives au nouveau répertoire sélectionné :

- le numéro de répertoire (0 à 31),
- l'octet d'options (cf instruction de création de répertoires),
- les conditions d'accès en création et réhabilitation de codes secrets (Cc),
- les conditions d'accès en création et modification de fichiers (Cf),
- le masque d'autorisation,
- la valeur d'autorisation du code 0.

Si le répertoire n'existe pas, le pointeur de répertoire courant est inchangé.

Si le répertoire sélectionné est contrôlé par un bloc de sécurité différent de celui du répertoire courant précédent (changement de répertoire actif), la carte va modifier le registre d'autorisation par application du masque de RAZ. De

plus, les statuts de clé de session et d'authentification sont positionnés conformément à leurs masques respectifs (cf options du code ROM).

c) Lecture des descripteurs de fichiers

Arguments :

P1 : déplacement exprimé en nombre de descripteurs
P2 : lecture simple (1 mot par descripteur)
ou lecture étendue (2 mots par descripteur)
P3 : longueur des données à lire (4 à 256 octets)

Cet ordre permet de lire les descripteurs des fichiers du répertoire courant. La longueur des données renvoyées est de 2 mots par descripteur (lecture étendue) ou d'un seul mot (lecture simple) pour assurer la compatibilité avec les cartes COS EPROM.
Le premier mot contient :

- le numéro de fichier,
- le type de fichier,
- la condition d'accès en lecture (Cr),
- la condition d'accès en écriture (Cw),
- la longueur du fichier exprimée en mots.

Le deuxième mot contient :

- la condition d'accès en réécriture et effacement (Cu),
- les verrous Lu, Lr, Lw et La,
- le bit de parité de l'adresse,
- la nature du fichier (fichier standard ou fichier contenant des clés secrètes),
- le numéro de répertoire auquel appartient le fichier.

Les descripteurs sont renvoyés dans l'ordre du plus récent au plus ancien. Si le déplacement P1 vaut 0, le premier descripteur renvoyé est donc le dernier alloué.
Si une erreur de parité a été détectée sur une adresse de fichier, le bit de parité est égal à 1 pour les deux fichiers contigus perturbés par cette erreur.
Si un fichier n'a pas été alloué de façon correcte (non validé), son descripteur est renvoyé avec un type égal à 0.
Cet ordre est notamment utilisé après la création d'un fichier pour connaître son numéro.

3.4.4. Les instructions de génération de clé de session

Les instructions de sélection de clé et de calcul de clé de session ont été exposées au paragraphe 3.3.6.1. Leurs arguments sont donnés ci-dessous.
La clé de session est utilisée par toutes les instructions chiffrées.

a) Sélection de clé

Arguments :
- P1 : position de la clé (en mots) dans le fichier
- P2 : numéro de fichier contenant la clé (0 à 255)
- P3 : longueur du nombre aléatoire fourni (8 octets)
- D1-D8 : nombre aléatoire N fourni à la carte

b) Calcul de la clé de session

Arguments :
- P1-P2 : '0000'
- P3 : longueur des données renvoyées (16 octets)

3.4.5. Les instructions de gestion des fichiers

a) Création d'un fichier

Arguments :
- P1-P2 : '0000'
- P3 : longueur des données fournies (4, 5 ou 8 octets)
- D1-Dn : message de création (éventuellement chiffré)

Cet ordre permet de créer un fichier dans le répertoire courant. Le message de création contenant les caractéristiques du nouveau fichier peut être transmis à la carte en clair ou de manière chiffrée.
Chargement non chiffré : (longueur = 4 ou 5 octets)
Le message de création contient les informations suivantes :
- tt : type du fichier (1 à 255)
- Cr : condition d'accès en lecture (0 à 15)
- Cw : condition d'accès en écriture (0 à 15)
- Lg : longueur du fichier en mots ('00' à 'FF' avec '00' pour 256 mots)
- < Cu > : condition d'accès en réécriture/effacement (0 à 15)

Le fichier est créé dans le répertoire courant si la condition de création de fichier du répertoire est satisfaite. Les données fournies doivent être conformes au contenu du descripteur.
Lorsque la longueur du message de création est de 4 octets, le fichier est interdit en réécriture et en effacement (le verrou Lu est automatiquement positionné).
Lorsque la longueur est de 5 octets, le cinquième octet fourni contient la condition d'accès en réécriture et en effacement Cu, et deux paramètres supplémentaires :

- type d'accès au fichier : accès en clair ou accès chiffré,
- nature du fichier : le fichier peut ou ne peut pas contenir de clés secrètes.

Si le dernier descripteur alloué comporte une erreur de parité, alors l'allocation d'un nouveau fichier sera refusée.
La seule modification possible d'un descripteur est le basculement des verrous à l'aide de l'instruction de modification de descripteur.

Chargement chiffré : (longueur du message = 8 octets)
Une authentification du terminal est nécessaire avant l'exécution de cet ordre.
Le message de création contient les 5 octets du message de chargement non chiffré, plus un octet contenant l'identifiant du répertoire courant.
Ce message est chiffré par la fonction inverse du DES avec la clé de session S avant d'être envoyé à la carte :

$$\text{message chiffré} = DES^{-1}(\text{message de création}, S)$$

Après réception et déchiffrement, la carte vérifie la cohérence des données. Le reste de l'instruction est identique au chargement en clair.

b) Modification d'un descripteur de fichier

Arguments :

P1 : '00'
P2 : numéro de fichier (0 à 255)
P3 : longueur des données fournies (1 octet)
D1 : données de modification d'accès

Cet ordre permet d'interdire l'accès en lecture, et/ou en écriture et/ou en réécriture et en effacement à un fichier en basculant respectivement les verrous Lr, Lw et Lu du descripteur

du fichier. Il est également possible de positionner le verrou La afin d'exiger des accès chiffrés au fichier.

c) Sélection d'un fichier

Arguments :

P1 : mode de recherche dans la table d'allocation courante

P2 : type du fichier recherché (0 à 255)

P3 : longueur des données renvoyées (4 octets pour que la carte renvoie 1 mot par descripteur ou 8 octets pour qu'elle renvoie 2 mots par descripteur)

Cet ordre permet de positionner le pointeur de fichier courant sur un fichier dont on connait le type, sans devoir lire la table d'allocation.

L'indicateur de mode de recherche peut prendre quatre valeurs :

0 : recherche du fichier suivant de type 'P2' à partir de la position en cours. Si le pointeur de fichier courant était indéfini, le premier fichier de type 'P2' est sélectionné,

1 : recherche du fichier précédent de type 'P2' à partir de la position en cours. Si le pointeur de fichier courant était indéfini, le dernier fichier de type 'P2' est sélectionné,

2 : recherche du premier fichier de type 'P2',

3 : recherche du dernier fichier de type 'P2'.

Les caractéristiques du descripteur du fichier cible sont renvoyées sous le même format que dans l'instruction de lecture des descripteurs de fichiers vue plus haut, en fonction de la longueur spécifiée.

Si le fichier n'est pas trouvé, le pointeur vaut zéro.

d) Ecriture fichier

Arguments :

P1 : déplacement en mots dans le fichier (0 à 255)

P2 : numéro du descripteur de fichier (0 à 255 avec 0 pour le fichier courant)

P3 : longueur des données fournies (0 à 255 octets)

D1-Dn : données à écrire

Cet ordre permet d'écrire et de modifier *une chaîne de données* (par surcharge des bits restés à 0) dans le fichier numéro 'P2' du répertoire courant, à partir du 'P1'^{ième} mot.

e) Ecriture chiffrée fichier

Arguments :

P1 : déplacement en mots dans le fichier (0 à 255)
P2 : numéro du descripteur de fichier (1 à 255)
P3 : longueur du message d'écriture (8 octets)
D1-D8 : message d'écriture

Cet ordre permet d'écrire et de modifier *un mot* (par surcharge des bits restés à 0) de manière sécurisée dans le fichier numéro 'P2' du répertoire courant, à partir du 'P1'^{ième} mot.

Le message d'écriture envoyé à la carte contient les informations suivantes :

- le contenu du mot à écrire,
- les paramètres P1 et P2 de l'instruction,
- les conditions d'accès au fichier Cr, Cw et Cu,
- les verrous La, Lr, et Lu du descripteur du fichier,
- la nature du fichier (K) : fichier standard ou fichier contenant des clés secrètes.

Il est chiffré avec la clé de session S :

message d'écriture = $DES^{-1}((mot/P1/P2/Cr/Cw/Cu/La/Lr/K/Lu),S)$

Après déchiffrement à l'aide de la fonction directe du DES et de la clé de session, la carte vérifie que

- les paramètres P1 et P2 reçus dans le message correspondent aux paramètres reçus dans l'ordre,
- les données Cr, Cw, Cu, La, Lr, K et Lu reçues dans le message sont égales aux caractéristiques du descripteur du fichier courant.

Si les données sont cohérentes, la suite de l'instruction est identique au déroulement de l'écriture non chiffrée.

Cette instruction ne fonctionne que si le terminal a été authentifié au cours de la session (statut d'authentification activé).

f) Réécriture fichier

Arguments :

P1 : déplacement en mots dans le fichier (0 à 255)
P2 : numéro du descripteur de fichier
(0 à 255 avec 0 pour le fichier courant)
P3 : longueur des données à écrire (0 à 255 octets)
D1-Dn : données à écrire

Cet ordre permet de réécrire *une chaîne de données* (par effacement et écriture) dans le fichier numéro 'P2' du répertoire courant, à partir du 'P1'^{ième} mot.

g) Réécriture chiffrée fichier

Arguments :

P1 : déplacement en mots dans le fichier (0 à 255)
P2 : numéro du descripteur de fichier (1 à 255)
P3 : longueur du message chiffré (8 octets)
D1-D8 : message de réécriture C

Cet ordre permet de réécrire *un mot* de manière chiffrée dans le fichier numéro 'P2' du répertoire courant, à partir du 'P1'^{ième} mot.
Le principe de chiffrement est similaire à celui de l'écriture chiffrée. Cette instruction ne fonctionne que si le terminal a été authentifié au cours de la session (statut d'authentification positionné).

h) Lecture fichier

Arguments :

P1 : déplacement en mots dans le fichier (0 à 255)
P2 : numéro du descripteur de fichier
(0 à 255 avec 0 pour le fichier courant)
P3 : longueur des données à lire
(0 à 255 octets avec 0 pour 256)

Cet ordre permet de lire *une chaîne d'octets* de longueur 'P3' dans le fichier numéro 'P2' du répertoire courant, à partir du 'P1'^{ième} mot.

i) Lecture sécurisée fichier

Arguments :

P1 : déplacement en mots dans le fichier (0 à 255)
P2 : numéro du descripteur de fichier (1 à 255)
P3 : longueur du message chiffré renvoyé (8 octets)

Cet ordre permet de lire *un mot* et de rechercher la position du premier mot vierge dans le fichier numéro 'P2' du répertoire courant, à partir du 'P1'^{ième} mot.
Le message chiffré renvoyé par la carte contient les informations suivantes :

- le mot lu,

- ses coordonnées P1 et P2,
- la position Po du premier mot vierge,
- l'identifiant Id du répertoire courant.

Ce message est chiffré par la fonction directe du DES avec la clé de session S :

message chiffré = DES((mot/P1/P2/Po/Id) , S)

Après déchiffrement par la fonction inverse du DES avec la clé de session, le terminal contrôle la cohérence des informations reçues et compare les coordonnées P1 et P2 avec celles qu'il a envoyées dans l'ordre. Si la cohérence est respectée, le contenu du mot et le déplacement Po sont considérés comme valides.

j) Effacement fichier

Arguments :

P1 : '00'
P2 : numéro du descripteur de fichier
(0 à 255 avec 0 pour le fichier courant)
P3 : '00'

Cet ordre permet d'effacer tout le contenu du fichier numéro 'P2' du répertoire courant.

k) Checksum fichier

Arguments :

P1 : '00'
P2 : numéro du fichier à contrôler (1 à 255)
P3 : longueur du résultat renvoyé (2 octets)
D1-D2 : résultat du checksum

Cet ordre permet de vérifier l'intégrité d'un fichier en calculant son checksum. Le résultat sur 16 bits est obtenu en additionnant tous les octets du fichier modulo 2^{16}.
Cette instruction ne fonctionne que si le type du fichier est supérieur ou égal à 248.

3.4.6. Les instructions de gestion des codes secrets

a) Chargement d'un code secret

Arguments :

P1 : mode de présentation (chiffré ou en clair)
P2 : numéro de code secret (1 à 7)
P3 : longueur des données fournies (8 ou 10 octets)
D1-D8 : code secret en clair ou message chiffré
‹D9-D10›: valeur d'autorisation activée par le code secret

Cet ordre permet de charger les codes 1 à 7 et de compléter leurs descripteurs. Le chargement n'est autorisé que si le bit Id du registre d'autorisation vaut 1 ou si la condition d'accès en création de code secret (Cc) est remplie.

Chargement en clair : (longueur = 8 ou 10 octets)
Le code secret est transféré sur 8 octets mais seuls les quartets de poids faible de chaque octet sont mémorisés dans la carte.
Si le chargement non étendu est utilisé (longueur = 8 octets), MCOS donne une valeur d'autorisation par défaut au code chargé . Cette valeur est fonction du numéro de code secret et du type de répertoire auquel il appartient :

- Pour les codes secrets du répertoire carte, le bit considéré pour la valeur d'autorisation est le bit dont le poids est égal au numéro du code secret.
  exemple : la valeur d'autorisation du code 6 vaut
  '0000 0000 0100 0000'$_B$ ou '0040'$_H$.
  Les valeurs d'autorisation des codes du répertoire carte sont donc des valeurs d'autorisation globales. Elles sont conservées lors d'un changement de répertoire si le masque de remise à zéro par défaut ('00FF') est inchangé.
- Pour les codes secrets d'un répertoire application, le bit considéré pour la valeur d'autorisation est le bit dont le poids est égal au numéro du code secret + 8.
  exemple : la valeur d'autorisation du code 6 vaut
  '0100 0000 0000 0000'$_B$ ou '4000'$_H$.
  Ces autorisations sont locales et sont annulées lors d'un changement de répertoire si le masque de remise à zéro vaut '00FF'.

Si le chargement étendu est utilisé (longueur = 10 octets), MCOS vérifie que la valeur d'autorisation reçue est compatible avec le masque d'autorisation du bloc de sécurité du répertoire concerné.
Après le chargement, la carte vérifie que le code et son descripteur sont écrits correctement. Si c'est le cas, le descripteur est validé.

Chargement chiffré : (longueur = 8 octets)
Une authentification du terminal est nécessaire avant l'exécution de cet ordre.
Le message envoyé contient les informations suivantes :

y	C0	0	C1	0	C2	Cs	C3	V0	C4	V1	C5	V2	C6	V3	C7

- 'y' est le quartet de poids fort du mode de présentation P1 de l'ordre (présentation du code chiffré ou en clair),
- les quartets 'C0' à 'C7' correspondent aux 32 bits du code secret réellement mémorisés dans la carte,
- les quartets 'V0' à 'V3' correspondent aux 16 bits de la valeur d'autorisation,
- Cs est le quartet de poids fort du numéro de code secret.

Ces données sont chiffrées par la fonction inverse du DES avec la clé de session S :

$$\text{message chiffré} = DES^{-1}(\text{données} , S)$$

Après déchiffrement, la carte vérifie que
- les données y et Cs du message correspondent aux paramètres P1 et P2 de l'instruction,
- les troisième et cinquième quartets valent 0.

Si ces tests sont concluants, la suite de l'instruction est équivalente au chargement en clair.

b) Présentation d'un code secret

Arguments :

P1	: '00'
P2	: numéro du code secret (0 à 7)
P3	: longueur des données fournies (8 octets)
D1-D8	: code secret en clair ou message chiffré

Présentation en clair :

Si le code secret est non valide ou annulé, MCOS renvoie une réponse négative.

Sinon, il consulte l'octet de ratification du code. Si les 3 bits de poids faible valent déjà 1, le code secret est bloqué, et l'instruction est terminée par un échec. Sinon, il positionne à 1 le bit de poids le plus faible valant zéro, puis compare le code reçu au code mémorisé en EEPROM. S'il n'y a pas égalité, l'instruction est terminée par un échec. Sinon, MCOS applique la valeur d'autorisation au registre d'autorisation par un 'OU logique' et remet l'octet de ratification à zéro.

Présentation chiffrée :

Le code secret est dans ce cas chiffré avant d'être envoyé à la carte :

$$\text{message chiffré} = DES^{-1}(\text{Code secret} , S)$$

Après déchiffrement, la carte exécute les mêmes opérations que pour la présentation en clair.

Si le code est correct, le terminal est authentifié et le statut d'authentification est activé.

c) Substitution d'un code secret

Arguments :

P1	:	'00'
P2	:	numéro de code secret à remplacer(0 à 7)
P3	:	longueur des données fournies (8 ou 16 octets)
D1-D8	:	code secret de substitution en clair ou chiffré (si la longueur = 8)
D1-D16	:	ancien code suivi du code de substitution en clair ou chiffré (si la longueur = 16).

Cet ordre permet de remplacer un code secret (par effacement et réécriture).

L'instruction exécutée avec une longueur égale à 8 octets doit être précédée d'une présentation correcte de l'ancien code secret.

L'instruction exécutée avec une longueur égale à 16 octets vérifie que l'ancien code fourni dans les octets D1-D8 est correct avant de le remplacer par la valeur D9-D16.

Substitution en clair :

Les données D1-D8 ou D1-D16 sont fournies à la carte en clair.

Substitution chiffrée :

Les données de substitution (octets D1-D8 ou D9-D16) respectent le format suivant :

0	C0	0	C1	0	C2	Cs	C3	X	C4	X	C5	X	C6	X	C7

- les quartets 'C0' à 'C7' correspondent aux 4 octets du nouveau code secret à écrire dans la carte,
- Cs est le numéro de code secret.

Ces données sont chiffrées avant d'être envoyées à la carte :

$$C = DES^{-1}(\text{données de substitution , S})$$

Après déchiffrement, la carte vérifie que les premier, troisième et cinquième quartets valent 0 et que le quartet Cs correspond au paramètre P2 de l'ordre, puis elle effectue le remplacement.
Si la longueur vaut 16 octets, l'ancien code (D1-D8) doit être chiffré de la même manière que dans l'instruction de présentation chiffrée de code.

d) Annulation d'un code secret

Arguments :

P1 : '00'
P2 : numéro de code secret (0 à 7)
P3 : longueur du code secret fourni (8 octets)
D1-D8 : code secret

Cet ordre permet d'interdire l'utilisation d'un code secret. Tous les accès protégés par ce code deviennent alors impossibles si aucun autre code n'a une valeur d'autorisation équivalente à celle du code annulé.
Avant d'annuler le code, la carte vérifie que la valeur D1-D8 correspond à la valeur du code mémorisée dans le bloc de sécurité. L'ordre est refusé si le code est verrouillé.
Si le code secret doit être présenté chiffré, les octets D1-D8 doivent être conformes aux données fournies dans l'instruction de présentation chiffrée de code.

e) Réhabilitation d'un code secret

Arguments :

P1 : '00'
P2 : numéro de code secret (0 à 7)
P3 : longueur du code secret fourni (8 octets)
D1-D8 : code secret

Cet ordre permet de réhabiliter un code secret bloqué. Son exécution nécessite que le bit Id du registre d'autorisation soit activé ou que la condition d'accès en réhabilitation Cc soit satisfaite.
MCOS examine les 3 bits de poids fort de l'octet de ratification du code. S'ils valent tous 1, le code est définitivement bloqué. Sinon le premier bit vierge est basculé à 1. Dans ce dernier cas, le code présenté est comparé au code mémorisé en EEPROM. S'il est incorrect, l'instruction se termine par un échec. S'il est correct, l'octet de ratification est remis à zéro et le code est débloqué.
Si le code secret doit être présenté chiffré, les octets D1-D8 doivent être conformes aux données fournies dans l'instruction de présentation chiffrée de code.

Un code 0 ne peut être réhabilité que si un autre code a une valeur d'autorisation permettant de basculer le bit Id ou le bit activant Cc. Pour qu'un des codes 1 à 7 active le bit Id, il faut avoir modifié la valeur du masque d'autorisation puisque sa valeur par défaut interdit d'utiliser le bit Id dans la valeur d'autorisation d'un code secret autre que le code 0.

Chapitre 4 : Comparaison des cartes MCOS et TB100

4.1. Performance et capacité des deux cartes

4.1.1. Temps d'exécution

Les temps d'exécution de quelques instructions sont repris dans le tableau ci-dessous. Ces temps ont été mesurés entre le moment où la carte reçoit le premier octet de l'ordre et le moment où elle renvoie le dernier mot d'état.

Instructions	TB100	MCOS
Remise à zéro	54 ms	36 ms
Lecture d'un mot	20 ms	18 ms
Lecture de 16 mots	108 ms	95 ms
Ecriture d'un mot	54 ms	21 ms
Sélection d'un répertoire application	29 ms	22 ms
Sélection du répertoire carte	19 ms	22 ms
Sélection d'une zone/d'un fichier	19 ms	22 ms
Création d'un répertoire application	404 ms	(*)
Création d'un répertoire sans bloc de sécurité		27 ms
Création d'un répertoire avec bloc de sécurité (1)		36 ms
Création d'un répertoire avec bloc de sécurité (2)		39 ms
Création d'une zone/d'un fichier	69 ms	28 ms
Présentation non chiffrée du PIN/du code 2	48 ms	30 ms
Génération d'un nombre aléatoire	67 ms	/
Authentification du terminal	81 ms	/
Sélection de clé de chiffrement	/	67 ms
Calcul de clé de session	/	77 ms
Présentation chiffrée du PIN/du code 3	99 ms	74 ms
Ecriture chiffrée d'un mot	167 ms	71 ms

figure 26 : comparaison des temps d'exécution

(*) L'instruction de création de répertoire dans MCOS peut être exécutée de trois manières différentes.

Ces chiffres doivent être interprétés en tenant compte de plusieurs aspects.
Premièrement, pour réaliser une certaine opération dans une carte, il est parfois nécessaire d'exécuter plusieurs instructions successivement. Par exemple, le calcul d'une clé de session dans MCOS doit obligatoirement être précédé d'une sélection de clé. Les temps d'exécution de ces deux instructions doivent donc être cumulés.
Deuxièmement, il est nécessaire de considérer les différentes opérations que chaque carte effectue au cours de l'exécution d'un ordre. Cet aspect concerne notamment l'instruction d'écriture : TB100 valide les données après leur écriture alors que la validation n'existe pas dans MCOS.
Troisièmement, le temps d'exécution d'une même instruction varie au cours de la vie d'une carte en fonction de la taille et de l'état des différents domaines créés et exploités dans l'EEPROM, mais aussi en fonction de l'adresse des données traitées.
Le lecteur trouvera ci-dessous une interprétation des différents temps d'exécution mesurés.

- Remise à zéro :
  Lors de l'initialisation, chaque carte renvoie l'état de la ratification des codes et clés de son répertoire racine.
  La carte MCOS doit pour cela consulter les octets de ratification du bloc de sécurité du répertoire carte. Le temps nécessaire à cette opération est invariable puisque ces octets sont mémorisés à un endroit fixe du bloc.
  TB100 doit balayer la zone d'accès de son répertoire carte. Le temps requis à cet effet dépend de la longueur et de l'état de la zone d'accès.
- Lecture d'un mot et lecture de 16 mots :
  L'écart observé ici est justifié par le fait que TB100 consulte le bit de validation de chaque mot lu. Lorsqu'elle lit un mot invalidé, la carte renvoie 4 octets valant '00' et indique la nature du problème dans le deuxième mot d'état.
- Ecriture d'un mot :
  La différence entre ces deux temps d'exécution est

essentiellement due à l'absence de la validation de l'écriture dans la carte MCOS.

- Sélection d'un répertoire application :
  Les deux résultats montrent l'intérêt de la table d'allocation de MCOS. En effet, la carte consulte cette table afin de connaître l'adresse du répertoire, puis elle pointe directement à cette adresse afin de le sélectionner. Le temps de sélection varie donc peu dans MCOS. Par contre, si une carte TB100 contient un nombre élevé de répertoires applications de taille importante, le temps de sélection d'un répertoire est d'autant plus long qu'il est situé dans les adresses hautes de la mémoire.
- Création d'un répertoire application, création d'un répertoire sans bloc de sécurité, création d'un répertoire avec bloc de sécurité (1) et (2) :
  La forte différence observée ici s'explique par le fait que la carte TB100 contrôle la virginité de chaque mot alloué au répertoire lors de sa création.
  Lors de la première création d'un répertoire avec bloc de sécurité dans MCOS (1), 2 octets ont été fournis à la carte. Les conditions d'accès, le masque d'autorisation et la valeur d'autorisation du code 0 ont donc été chargés avec des valeurs par défaut. Dans la deuxième création avec bloc de sécurité (2), 6 octets supplémentaires ont été envoyés à la carte.
- Création d'une zone/d'un fichier :
  Même remarque que pour la création d'un répertoire.
- Présentation non chiffrée du PIN/du code 2 du répertoire carte :
  Le faible temps d'exécution nécessaire pour présenter un code secret dans MCOS est rendu possible par la structure des blocs de sécurité.
  Premièrement, chaque code est mémorisé à un endroit fixe du bloc de sécurité. Le système d'exploitation ne doit donc pas chercher le code lors de sa présentation. Notons que cet avantage n'est possible qu'au détriment de la souplesse du système de sécurité. En effet, le nombre de codes est limité à 8 dans un bloc de sécurité de MCOS. De plus, 17

octets sont automatiquement réservés pour le bloc de sécurité s'il est demandé lors de la création d'un répertoire. Cet espace est alloué inutilement si les 8 codes ne sont pas utilisés.

Deuxièmement, chaque descripteur de code secret contient son propre octet de ratification, ce qui permet un contrôle rapide des résultats des présentations antérieures de ce code.

Cependant, bien que le PIN soit mémorisé dans le répertoire racine d'une carte TB100, il peut être présenté même si le répertoire actif n'est pas le répertoire carte. Par contre, si le PIN d'une carte MCOS est attribué à un code secret de son répertoire racine, ce répertoire doit être actif pour pouvoir le présenter. Lorsque le répertoire actif est un répertoire application, la présentation du PIN (30 ms) doit donc être précédée de la sélection de l'entité carte (22 ms) et suivie d'une nouvelle sélection pour revenir au répertoire application de départ (22 ms).

- Authentification du terminal :

  Dans TB100, l'instruction d'authentification du terminal (81 ms) doit être précédée de la génération d'un nombre aléatoire (67 ms).

  Dans MCOS, le terminal est authentifié après une présentation chiffrée d'un code secret (74 ms). Cette instruction doit être précédée d'une sélection de clé (67 ms) et du calcul d'une clé de session (77 ms).

  Le temps total d'exécution requis par la carte au cours d'une procédure d'authentification du terminal est donc de 148 ms pour TB100 et de 218 ms pour MCOS.

- Présentation chiffrée du PIN/du code 3 :

  La présentation chiffrée du PIN dans TB100 doit également être précédée d'une génération d'un nombre aléatoire. Le temps total d'exécution requis par la carte TB100 au cours d'une procédure d'authentification du porteur par PIN est donc de 166 ms. Dans MCOS, la procédure correspond à l'authentification du terminal avec un temps d'exécution de 218 ms.

- Ecriture chiffrée d'un mot :
  L'écriture chiffrée est plus longue dans TB100 puisque d'une part, la carte valide le mot, et d'autre part, le processus de chiffrement exécute deux fois l'algorithme DES.
  Il faut également tenir compte ici de la nécessité de la génération d'un nombre aléatoire dans TB100 et du calcul d'une clé de session dans MCOS.
  Dans MCOS, l'écriture chiffrée doit obligatoirement être précédée de l'authentification du terminal. Dans TB100, une option permet de l'exiger.

Remarque : Nous avons vu que certaines instructions doivent être précédées de la génération d'un nombre aléatoire dans TB100, et du calcul d'une clé de session dans MCOS.
Le nombre aléatoire généré par une carte TB100 reste valable durant toute une session. Le temps nécessaire à sa génération ne doit donc être considéré qu'une seule fois.
La clé de session calculée par une carte MCOS peut être utilisée au cours de toute une session si le masque de statut de clé de session vaut 1. Si ce masque vaut 0, une nouvelle clé devra être calculée après chaque sélection d'un nouveau répertoire autonome.

4.1.2. Capacité de mémorisation

La capacité de mémorisation des deux composants est sensiblement différente :

	TB100	MCOS
ROM	6 K octets	6 K octets
RAM	128 octets	160 octets
EEPROM	3 K octets	2 K octets

L'EEPROM de la carte TB100 est en partie utilisée par le système d'exploitation, notamment pour les bits de validation. L'espace disponible pour les applications est de 679 mots, soit 2716 octets. Dans MCOS, cet espace représente 505 mots.
Une carte multi-émetteurs est destinée à contenir plusieurs applications. La taille de la mémoire utilisateur est une des

premières limites de ce type de carte. La capacité de TB100 est donc appréciable, d'autant plus que l'ajout éventuel d'un filtre dans la mémoire EEPROM de la carte MCOS réduit encore la taille disponible pour les applications.

4.2. Organisation logique de la mémoire EEPROM

TB100 offre une arborescence à trois niveaux, soit un niveau en plus que dans MCOS. Ce troisième niveau, constitué par les répertoires services, est notamment utilisé dans la carte de l'Université de Rome.

4.3. Organisation physique de la mémoire EEPROM

Le mapping des deux cartes est très différent.
Dans TB100, l'espace de l'EEPROM est découpé progressivement en blocs imbriqués : le répertoire carte, les répertoires applications, les répertoires services et les zones. Le descripteur de chacun de ces blocs précède son corps. La taille de chaque bloc doit donc être spécifiée lors de sa création.
Dans MCOS, les caractéristiques des répertoires et des fichiers sont regroupées dans la table d'allocation, et les données leur appartenant sont rassemblées dans une autre partie de la mémoire. La taille d'un répertoire application ne doit pas être précisée lors de sa création. Les fichiers alloués dans un répertoire ne sont pas mémorisés les uns à la suite des autres. Ceci est possible grâce à la table d'allocation qui lie les fichiers à leurs répertoires respectifs. Il est donc possible de créer des fichiers dans un même répertoire jusqu'à concurrence du nombre maximum de descripteurs de la carte.

La solution de MCOS est souple mais nécessite une grande discipline de la part de chaque gestionnaire d'application. Il faut en effet que chacun respecte le quota d'espace mémoire qu'il a réservé auprès de l'organisme émetteur de la carte.
Ce problème ne se pose pas dans la carte TB100 puisque l'espace disponible pour un répertoire n'est pas dynamique. Soulignons également l'intérêt du contrôle de virginité de l'espace alloué à un répertoire lors de sa création. Ce contrôle garantit à

l'émetteur d'une application de TB100 qu'il dispose réellement de la mémoire qu'il a achetée.

4.4. Les descripteurs

L'objectif de ce paragraphe n'est pas de comparer rigoureusement le contenu des descripteurs des domaines des deux cartes, mais d'analyser la méthode de gestion et la sécurité des différents domaines des deux produits.

- Le niveau d'un domaine n'a pas de raison d'exister dans les descripteurs de MCOS puisque le descripteur du répertoire carte est écrit à une adresse fixe (l'adresse haute de la table d'allocation), et qu'il n'y a qu'un autre niveau.
- Le type de domaine de TB100 a pour équivalent dans MCOS le type de descripteur (répertoire ou fichier).

4.4.1. Les descripteurs de répertoires

- TB100 distingue les différents répertoires d'un même niveau par leur référence. MCOS les repère par leur identifiant.
- La longueur, obligatoire dans TB100, n'est pas spécifiée dans MCOS.
- Les conditions de création de répertoires de TB100 permettent d'exiger des combinaisons de conditions du porteur et de l'émetteur.
  Dans MCOS, l'unique condition de création d'un répertoire est la présentation du code 0 du répertoire carte.
- Les conditions de création de zones de transactions de TB100 correspondent à la condition de création et de modification de fichier de MCOS. Ici aussi, les combinaisons ne sont pas possibles.
- Dans TB100, il existe une option permettant d'exiger une authentification du terminal avant d'effectuer
  . une authentification du porteur ou de l'émetteur,
  . une écriture sécurisée, ou
  . un effacement sécurisé
  dans le répertoire actif.

Dans MCOS, une authentification du terminal est toujours exigée avant d'exécuter

. une écriture sécurisée, ou

. une réécriture sécurisée

dans la carte.

- L'option d'autorisation de création de répertoires de niveau inférieur dans TB100 n'a pas d'équivalent dans MCOS puisqu'il n'existe que deux niveaux.
- Il n'y a pas de verrou d'invalidation dans MCOS. L'invalidation d'un répertoire doit être réalisée en modifiant les descripteurs (positionnement des verrous d'accès) de tous les fichiers qu'il contient.
- Les descripteurs de répertoire de TB100 contiennent un verrou d'utilisation. Lors de l'allocation d'un nouveau répertoire autonome, ce verrou est basculé après avoir créé le système de sécurité minimal du répertoire, à savoir la clé de l'émetteur primaire et la zone d'accès. Après le positionnement de ce verrou, il est impossible de créer une nouvelle zone d'accès dans le répertoire.

  Ce verrou n'existe pas dans MCOS. Ceci est justifié par le fait que

  . le bloc de sécurité est créé automatiquement s'il est demandé lors de la création du répertoire,

  . il n'y a pas de zone de ratification commune aux différents codes, chaque code est ratifié dans un octet prévu à cet effet dans son descripteur,

  . le code 0 est chargé automatiquement avec une valeur nulle lors de la création du répertoire. Il doit ensuite être substitué par la clé de l'émetteur.
- Le checksum permet de vérifier l'intégrité d'un descripteur de TB100 au cours de son exploitation.

  Par contre, le bit de validation d'un descripteur de MCOS ne garantit pas l'intégrité de son contenu au cours de la vie de la carte.

4.4.2. Les descripteurs de zones de transactions de TB100 et les descripteurs de fichiers de MCOS

- Une zone de transactions de TB100 peut être consommée en mode jeton ou en mode mot (avec validation). Ce dernier mode est généralement plus utilisé que le premier.
  Cette option n'existe pas dans MCOS puisque les mots ne sont pas validés après leur écriture. Il est donc toujours possible d'écrire sur un mot non vierge par surcharge des bits valant 0.
- Une zone de transactions est repérée par son niveau, son type et sa référence écrits dans son descripteur. Lorsqu'une zone de transactions est saturée, il est possible de créer une nouvelle zone ayant la même référence. Les différents modes d'exécution de l'instruction de sélection de domaine permettent de se déplacer d'une zone à l'autre.
  Dans MCOS, un fichier est désigné par son numéro de descripteur. Ce numéro est unique, il est attribué par le système lors de la création du fichier.
- La longueur d'une zone de TB100 est écrite dans son descripteur.
  La longueur d'un fichier de MCOS n'est pas spécifiée dans son descripteur. Notons que cette longueur est limitée à 256 octets alors que la longueur d'une zone de transactions de TB100, exprimée en mots sur 2 octets, n'est limitée que par la taille du répertoire auquel elle appartient.
- Les conditions d'effacement, d'écriture et de lecture dans une zone de transactions peuvent exiger des *combinaisons* de conditions du porteur et de l'émetteur.
  Dans MCOS, les combinaisons de conditions d'accès à un fichier sont impossibles.
- Dans TB100, il est possible d'écrire dans une même zone, de manière chiffrée et en clair. Par option, le type d'écriture (chiffrée ou en clair) de chaque mot d'une zone de transactions peut être mémorisé dans son $32^{ème}$ bit. Ce dispositif permet par exemple d'exiger une écriture ON-LINE (écriture chiffrée) après un nombre déterminé d'écritures

OFF-LINE (écritures non chiffrées).
Dans MCOS une option détermine si tous les accès au fichier sont réalisés en clair ou de manière chiffrée. Il n'est pas possible d'alterner ces deux types d'accès. Néanmoins, si l'option 'accès en clair' a été choisie lors de la création du fichier, il est possible de la modifier (définitivement) pour ensuite accéder au fichier de manière chiffrée.

- Il n'y a pas de verrou d'invalidation d'un domaine dans MCOS. Il est possible d'interdire les accès en lecture, et/ou en écriture et/ou en réécriture à un fichier en basculant les verrous d'accès correspondants de son descripteur.

4.4.3. Les descripteurs de zones secrètes de TB100 et les descripteurs de codes secrets de MCOS

Avant d'analyser le contenu de ces descripteurs, considérons le rôle et le type de mémorisation des différents codes et clés des deux cartes.

Les normes ISO relatives aux cartes sont en constante évolution. Leur objectif est notamment d'attribuer un rôle précis à chaque clé ou code secret défini dans une carte.

Nous avons vu que la carte TB100 respecte rigoureusement cet objectif. En effet, le rôle de chaque clé ou code est déterminé lors de l'inscription du masque en ROM.

Dans MCOS, le rôle des 8 codes secrets d'un bloc de sécurité n'est pas imposé par le masque. La fonction de chacun est déterminée lors de la personnalisation par l'attribution de valeurs d'autorisation particulières, et éventuellement par la modification des masques d'autorisation et de remise à zéro.

La clé de personnalisation n'existe pas dans MCOS.

Le code émetteur du répertoire carte de MCOS ne nécessite pas de zone particulière puisqu'il est écrit par substitution du code de fabrication.

Les clés secrètes de TB100 sont stockées dans des zones secrètes particulières. Les clés secrètes de MCOS sont mémorisées dans des fichiers normaux. On ne peut donc pas les

sélectionner en mentionnant leur identifiant, leur numéro et leur version comme dans TB100.

Les descripteurs des zones secrètes de TB100 contenant des codes secrets et les descripteurs des codes secrets d'un bloc de sécurité de MCOS

- Le niveau et le type d'une zone secrète de TB100 n'ont pas d'équivalent dans MCOS puisque le descripteur et le code font partie d'un bloc de sécurité attaché directement à un répertoire.
- L'invalidation d'une zone secrète de TB100 n'est pas utilisée dans le même but que l'annulation d'un code secret de MCOS.
  L'invalidation d'une zone secrète de TB100 permet de changer la valeur d'une clé en toute sécurité. L'émetteur primaire doit pour cela invalider la zone secrète contenant la clé afin d'interdire son utilisation. Il crée ensuite une nouvelle zone secrète et y écrit la nouvelle version de la clé.
  L'annulation d'un code secret de MCOS permet d'interdire définitivement l'utilisation d'un code. Si aucun code du répertoire actif n'a une valeur d'autorisation comparable à celle du code annulé, l'accès à certains fichiers peut être définitivement interdit.
- L'identifiant du code n'existe pas dans MCOS puisque la position du code dans le bloc de sécurité correspond à son numéro. De plus, son action est liée à sa valeur d'autorisation.
- La condition de remplacement d'un code lié à un répertoire de TB100 indique si l'authentification de l'émetteur primaire est exigée en plus de la présentation du code. Cette condition est indépendante de celles des autres codes d'un même répertoire.
  Dans MCOS, il est également possible d'exiger la présentation du code émetteur pour toute substitution de code. Ceci est possible à deux conditions :
  . le descripteur du code doit être configuré de manière à imposer une présentation chiffrée du code. Une clé

de session doit dans ce cas être calculée avant la substitution,

. la zone de contrôle du code ROM doit être configurée de telle manière que les instructions de sélection de clé et de calcul de clé de session soient contrôlées par code secret.
Le registre d'autorisation doit dans ce cas avoir une valeur particulière pour qu'une clé de session puisse être calculée. En donnant une valeur d'autorisation adéquate au code émetteur, on obtient la condition désirée.

- Dans TB100, une clé ou un code n'est jamais effacé(e). La substitution est réalisée par l'écriture d'une nouvelle clé ou d'un nouveau code. Chaque mot écrit dans une zone secrète est validé comme dans les zones de transactions. En cas d'erreur d'écriture, il suffit d'invalider la zone et d'en créer une nouvelle.
Dans MCOS, un code est substitué par effacement et réécriture. Une erreur éventuelle de réécriture est signalée par le deuxième mot d'état renvoyé par la carte. Ce type d'erreur est très gênant. En effet, s'il est impossible de changer de numéro de code, ou si tous les autres codes du bloc de sécurité sont utilisés, la seule solution est de créer un nouveau répertoire....

4.5. La portabilité

Ce paragraphe aborde la portabilité d'une application implémentée sur une carte vers un autre type de carte d'un même constructeur.

En ce qui concerne les produits de BULL CP8, une application résidant sur une carte TB10 peut être directement portée sur une carte TB100, mais un masque simplifié dérivé des masques TB10 et TB100 n'est pas encore disponible.

Comme nous l'avons vu dans l'introduction du chapitre 3, la carte MCOS est compatible avec les cartes COS si les applications qu'elles contiennent n'utilisent que les instructions de base. Les instructions n'existant pas dans le

masque MCOS peuvent néanmoins être ajoutées dans le programme filtre, avec les inconvénients que comporte cette solution.

4.6. Souplesse du masque

Gemplus offre un code ROM simple accompagné d'une bibliothèque de sous-programmes à ajouter en EEPROM, pour adapter rapidement un masque à une spécification donnée, et pour faciliter le prototypage.
Cette optique réduit peut-être le travail d'adaptation d'un masque à une application donnée, mais complique la phase de personnalisation puisque ces sous-programmes ne sont pas écrits lors de la fabrication du composant. De plus, l'ajout de ce code réduit la taille de l'EEPROM disponible pour l'utilisateur.
Il faut souligner le danger éventuel d'un tel procédé : si cette tâche est confiée à l'émetteur des cartes, il a la possibilité d'introduire du code à sa guise, pouvant endommager ou pirater le code ROM. Si cette tâche est réservée à l'encarteur, elle perd son intérêt. L'encarteur dispose des appareils adéquats (émulateurs) lui permettant de réaliser le prototypage.
S'il faut parfois ajouter des fonctions à un masque pour répondre à une offre, d'autres fonctions peuvent par contre être supprimées, la place ainsi récupérée compensant l'ajout de code.
La mise au point d'un masque spécifique sur émulateur et sa gravure par le fabricant est la politique suivie par BULL CP8. L'optique de GEMPLUS est intéressante pour une petite quantité de cartes.

4.7. La sécurité

4.7.1. Les clés et les codes

Nous avons vu au paragraphe 2.1.5 l'ensemble des différents codes et clés qui peuvent constituer le système de sécurité d'un répertoire de la carte TB100. Leur nombre varie

d'un répertoire à un autre. Seul le système de sécurité minimal est obligatoire pour un répertoire autonome.

Le système de sécurité du répertoire carte et de chaque répertoire application autonome de MCOS est composé

- d'un bloc de sécurité pouvant contenir 8 codes secrets,
- et des clés de chiffrement (en nombre illimité) mémorisées dans des fichiers.

Ce type de gestion n'est pas très souple puisqu'il alloue automatiquement 17 mots pour chaque bloc de sécurité demandé. Cet espace est gaspillé inutilement si l'application ne requiert pas 8 codes secrets.

4.7.2. La validation de chaque mot

L'intérêt de la validation de l'écriture de chaque mot dans une zone de TB100 a déjà été montré dans ce chapitre. Ce mécanisme apporte une double garantie. Premièrement, le bit de validation est la preuve de l'écriture correcte du mot. Deuxièmement, il garantit l'intégrité du mot : il est en effet impossible de réécrire sur un mot validé.

Le bit de validation n'existe pas dans MCOS. Ce masque dispose de l'instruction de checksum qui permet de vérifier l'intégrité d'un fichier, mais n'apporte aucune preuve de la bonne écriture physique de chaque mot. Il est donc impossible, en lisant un fichier, de déterminer quel mot à été mal écrit.

4.7.3. La diversification

L'analyse de la documentation et des quelques cartes dont nous disposons ne reflète pas l'utilisation de la diversification dans la carte MCOS.

4.7.4. Le module de sécurité

Il a été supposé dans ce document que les terminaux et les serveurs dialoguant avec des cartes MCOS sont munis de modules de sécurité analogues à ceux de BULL CP8.

4.7.5. La ratification

La ratification est traitée différemment dans les deux cartes étudiées.
Chaque répertoire autonome de la carte TB100 est doté d'une zone d'accès commune à tous les codes et clés du répertoire. Après une présentation erronée, aucun autre code ou clé que celui ou celle qui a provoqué l'erreur ne peut être présenté(e).
Dans MCOS, chaque code dispose de son propre octet de ratification. Ce mécanisme est peu coûteux en temps et en espace mémoire. Une mauvaise présentation ou le blocage d'un code n'empêche pas la présentation des autres codes. Seules les fonctionnalités liées au code mal présenté sont éventuellement suspendues.

4.8. Les fonctionnalités

4.8.1. L'accès à une zone de TB100 et l'accès à un fichier de MCOS

Toutes les instructions d'accès à une zone de TB100 traitent la zone courante. La sélection de la zone est donc indispensable avant d'y accéder.
Dans MCOS, un fichier peut être désigné de deux manières :

- soit le numéro du fichier est indiqué dans le paramètre P2 de chaque instruction d'accès au fichier (ce mode est possible puisque le numéro de fichier est un identifiant),
- soit le fichier est sélectionné avant d'y accéder, le paramètre P2 des instructions d'accès qui suivent peut alors être fixé à la valeur '00' pour désigner le fichier courant.

Le lecteur trouvera ci-dessous une comparaison de l'ensemble des instructions des deux cartes. Les paragraphes 4.8.2. et 4.8.3. font référence aux instructions de TB100. Les instructions propres à MCOS sont reprises au paragraphe 4.8.4.

4.8.2. Les instructions de base

A. Remise à zéro (RAZ)

La remise à zéro de MCOS renvoie davantage d'informations que celle de TB100 :

- la référence de l'émetteur de la carte,
- le nombre de répertoires dans la carte.

B. Sélection de domaine

Cette instruction unique permet de sélectionner n'importe quel domaine de l'arborescence (répertoire ou zone). Le domaine est recherché sur base de son niveau, de son type et éventuellement de sa référence.
Une instruction de lecture de résultat est nécessaire pour obtenir le contenu du descripteur du domaine sélectionné.
Dans MCOS, il existe deux instructions de sélection :

- La sélection d'un répertoire :
  Le répertoire est recherché sur base de son identifiant (unique dans la carte). L'instruction renvoie les caractéristiques du répertoire.
- La sélection d'un fichier :
  La recherche est basée sur le type de fichier. Elle renvoie les caractéristiques du fichier.

C. Lecture de données

L'instruction de lecture de données permet de lire des informations dans une zone ou dans un répertoire pendant les phases de personnalisation et d'utilisation.
Dans MCOS, il existe trois instructions de lecture :

- la lecture fichier est uniquement utilisable dans un fichier,
- la lecture mémoire permet de lire tout le contenu de l'EEPROM en phase de personnalisation,
- la lecture des descripteurs de fichiers renvoie les caractéristiques des fichiers du répertoire courant.

D. Ecriture de données

L'écriture de données permet d'écrire des informations dans une zone pendant la phase de personnalisation et pendant la phase d'utilisation.
Dans MCOS, il existe deux instructions d'écriture :

- l'écriture fichier n'est utilisable que dans un fichier,
- l'écriture mémoire permet d'écrire dans toute l'EEPROM (sauf sur la référence de la carte) en phase de personnalisation.
  Etant donné l'absence de validation dans MCOS, des précautions supplémentaires s'imposent afin d'interdire l'utilisation de cette instruction pour écrire sur le filtre éventuel.

E. Effacement de données

Une chaîne de mots de longueur variable peur être effacée dans une zone de transactions de TB100.
Dans MCOS, l'instruction d'effacement de fichier ne permet que d'effacer l'intégralité d'un fichier. Néanmoins, la carte MCOS dispose d'une instruction de réécriture dans un fichier.

F. Les instructions de recherche

Il n'existe pas d'instruction de recherche dans MCOS. Ces instructions sont pourtant très utiles, notamment lors de l'écriture d'une transaction dans un fichier : il est nécessaire de connaître l'adresse du premier mot vierge qui peut contenir les données à écrire.
Ces instructions peuvent sans doute être codées en EEPROM, mais nous avons vu les inconvénients de ce procédé.

G. Lecture de résultat

- L'exécution de cette instruction après une remise à zéro renvoie le nombre total de mots du répertoire carte et le contenu de son descripteur.
  Dans MCOS, le contenu du descripteur du répertoire carte peut être obtenu lors de sa sélection.
- La lecture de résultat après une sélection de domaine renvoie l'adresse relative du descripteur sélectionné, le nombre total de mots du domaine et le contenu de son descripteur.
  Dans MCOS, la sélection d'un domaine renvoie automatiquement les caractéristiques de son descripteur.

H. Génération de nombre aléatoire

Cette instruction n'existe pas telle quelle dans MCOS, la carte génère un nombre aléatoire et le renvoie lors de l'exécution de l'instruction de calcul de clé de session.

I. Directory

Cette instruction a la même fonction que l'instruction de lecture des descripteurs de fichiers d'un répertoire de la carte MCOS.

J. Création d'un domaine

Les zones et les répertoires d'une carte TB100 sont créés par la même instruction. La longueur du descripteur écrit est fonction du type de domaine créé.
Dans MCOS, il existe deux instructions différentes pour la création d'un répertoire et d'un fichier. Cette distinction est sans doute due au fait que la carte peut allouer automatiquement un bloc de sécurité lors de la création d'un répertoire, alors que la création d'un fichier est beaucoup plus simple.

K. Ecriture des verrous

Cette instruction permet de brûler le verrou d'utilisation d'un répertoire carte, application ou service, ainsi que le verrou de fabrication de la carte. Dans MCOS, il n'y a pas de verrou d'utilisation pour les répertoires applications. Les verrous de fabrication et d'utilisation de la carte peuvent être basculés par une instruction d'écriture mémoire utilisée en adressage indirect.

4.8.3. Les instructions de sécurité

4.8.3.1. L'authentification.

A. Authentification du terminal.

L'authentification du terminal par une carte TB100 est réalisée par une instruction spécifique. La procédure est détaillée au paragraphe 2.1.11.3.
L'authentification réalisée dans un répertoire actif est valable pour toute la session dans les autres répertoires de la carte.

Il n'existe pas d'ordre propre à cette fonction dans MCOS. L'authentification du terminal (décrite au paragraphe 3.3.6) est effectuée par une présentation chiffrée d'un des codes secrets du répertoire courant.
Le terminal est authentifié par le fait qu'il connait la clé Ks utilisée par la carte pour calculer la clé de session. En effet, le module de sécurité du terminal doit calculer la clé de session sur base

- de la clé Ks mémorisée dans un de ses fichiers, et
- des éléments renvoyés par la carte après qu'elle ait elle-même calculé la clé de session (le résultat R calculé par l'instruction de sélection de clé, et le nombre aléatoire généré par la carte).

L'authentification réalisée dans un répertoire actif n'est valable pour toute la session dans les autres

répertoires actifs de la carte que si le masque de statut d'authentification vaut 1.

B. Authentification du porteur par PIN ou AID (en clair)

Ces instructions ont pour équivalent dans MCOS l'instruction de présentation des codes 1 à 7.

C. Authentification chiffrée du porteur par PIN ou AID

Ces instructions ont pour équivalent dans MCOS l'instruction de présentation chiffrée des codes 1 à 7.

D. Authentification de l'émetteur par la clé IK

Dans TB100, l'authentification de l'émetteur est toujours chiffrée.
Dans MCOS, la configuration par défaut attribue le code 0 à l'émetteur. Ce code peut être présenté de manière chiffrée ou en clair. Ce dernier choix paraît peu sécuritaire.

E. Authentification de l'émetteur par clé SK

Dans MCOS, les émetteurs secondaires peuvent se voir attribuer un des codes 1 à 7. Le nombre de ces codes est cependant réduit. La présentation de ce code peut ici aussi être non chiffrée alors que l'authentification de l'émetteur secondaire est toujours chiffrée dans TB100.

4.8.3.2. L'invalidation d'un répertoire ou d'un fichier

L'invalidation est notamment utilisée
- pour remplacer une clé,
- lorsque le contenu d'une zone de transactions est saturé,
- pour empêcher l'évolution d'un répertoire en cas de fraude,
- ou lorsqu'un incident apparaît dans une zone.

Il n'existe pas d'instruction équivalente dans MCOS. L'instruction de 'modification de fichier' peut cependant

être utilisée pour interdire les accès à un fichier en basculant les verrous d'accès de son descripteur.

4.8.3.3. L'écriture sécurisée

A. Ecriture sécurisée d'un mot (chiffrement avec la clé IK ou avec la clé SK)

Cette instruction utilise la clé d'émetteur primaire ou secondaire du répertoire actif. Si un autre répertoire actif est sélectionné, la clé de ce nouveau répertoire doit être mise en oeuvre.
Dans MCOS, un message d'écriture sécurisée est chiffré avec la clé de session. Cette clé est calculée en fonction d'une clé Ks appartenant soit absolument au répertoire actif, soit à un répertoire quelconque de la carte. Ceci est fonction de la valeur du masque du statut de clé de session :

- si ce masque est positionné, le statut de clé de session n'est pas remis à zéro lors de la sélection d'un nouveau répertoire actif. Une clé de session calculée dans un répertoire reste alors valable dans tous les autres répertoires sélectionnés au cours d'une même session.
- si ce masque n'est pas positionné, le statut est annulé lors de la sélection d'un nouveau répertoire actif et une nouvelle clé de session doit être calculée pour pouvoir utiliser une instruction chiffrée.

B. Ecriture sécurisée d'un descripteur (chiffrement avec la clé IK ou avec la clé SK)

Cette instruction permet de créer des zones et des répertoires en toute sécurité.
Dans MCOS, il n'existe pas d'instruction de création sécurisée d'un répertoire. Seule la création d'un fichier peut être chiffrée.
La remarque concernant l'utilisation de la clé de session

dans l'instruction d'écriture sécurisée par la clé IK est valable ici également.

4.8.3.4. L'effacement sécurisé (chiffrement avec la clé IK ou avec la clé SK)

Cette instruction n'a pas d'équivalent dans MCOS. Néanmoins, cette carte dispose de l'instruction de réécriture sécurisée dans un fichier.

4.8.3.5. La certification

La certification n'est pas implémentée dans le masque de MCOS. Elle est pourtant très utilisée, notamment dans les cartes bancaires.

4.8.3.6. La signature de message

Le développement croissant du courrier électronique et des télécommunications en général accroît la nécessité d'utilisation de signatures électroniques. La possibilité de génération et de vérification de signatures par la carte TB100 est donc un atout non négligeable.

4.8.4. Instructions propres à MCOS

4.8.4.1. Lecture chiffrée

Cette instruction n'existe pas dans TB100.

4.8.4.2. Chargement d'un code secret

Dans TB100, les instructions d'écriture ou d'écriture sécurisée sont utilisées pour charger les codes secrets dans les zones secrètes.

4.8.2.3. Substitution d'un code secret

Un code secret n'est jamais effacé dans TB100; sa substitution est réalisée par l'écriture de sa nouvelle valeur à l'aide des instructions d'écriture ou d'écriture sécurisée.

4.8.2.4. Annulation d'un code secret

L'annulation d'un code secret de MCOS est irréversible et interdit toute utilisation ultérieure de ce code.
Dans TB100, l'instruction d'invalidation d'une zone secrète contenant un code secret rend également le code inutilisable, mais une nouvelle zone secrète peut ensuite être créée pour le remplacer.

4.8.2.5. Réhabilitation d'un code secret

Les conditions de réhabilitation d'un code secret de MCOS sont déterminées lors de la création du répertoire auquel il est rattaché. Ces conditions correspondent à la présentation d'un des 8 codes secrets du répertoire traité ou du répertoire racine.
Dans TB100, l'unique moyen de réhabilitation est la présentation du PIN suivie de l'authentification de l'émetteur primaire.

4.8.2.6. Statut de la carte

Cette instruction n'existe pas dans TB100. La plupart des informations qu'elle renvoie peuvent cependant être obtenues par les ordres de lecture.

Conclusion

L'organisation, les fonctionnalités et les mécanismes gérés par les cartes TB100 et MCOS ont été décrits dans les second et troisième chapitres. Le propos du dernier chapitre était de comparer les caractéristiques de ces deux produits.

Le but de ce mémoire n'était pas de déterminer si une carte est meilleure que l'autre dans l'absolu, ni de choisir l'une d'entre elles relativement à un critère donné. Il est néanmoins possible de dégager les particularités intéressantes de chaque carte.

Il ressort de l'analyse de la carte MCOS que la table d'allocation, le système de ratification des codes secrets, l'instruction de lecture chiffrée et la compatibilité avec les cartes COS sont des atouts appréciables. L'ajout de sous-programmes en EEPROM peut être utile dans des cas bien précis.

La carte TB100 se distingue par sa logique d'exploitation, ainsi que par la souplesse et la richesse de ses fonctionnalités et de son système de sécurité. Nous retiendrons particulièrement la validation de l'écriture, le contrôle de virginité de l'espace alloué à un domaine, la possibilité d'exiger des combinaisons de conditions d'accès et de création, le rôle précis de chaque clé et code, les instructions de certification et de signature électronique, la capacité de mémorisation de l'EEPROM, l'existence du niveau service.

Glossaire

AID : Alternate IDentification. Code secret du porteur constitué d'une information personnelle relative au porteur et immuable au cours de sa vie.

AK : Authentification Key. Clé secrète utilisée pour authentifier un terminal ou pour chiffrer des données.

Carte à micro-calculateur : carte à micro-circuit dont le composant est un MAM.

Carte à micro-circuit / carte à puce : carte de format 'carte de crédit' équipée d'un composant électronique.

Code secret : information confidentielle permettant d'authentifier une entité. Si le code doit être présenté à la carte de manière sécurisée, il est chiffré avec une clé secrète.

Clé secrète : information confidentielle permettant de chiffrer et de déchiffrer des données. Une clé secrète permet également d'authentifier (de manière sécurisée) l'entité qui la détient.

DES : Data Encryption Standard. Algorithme de chiffrement à clé secrète et réversible.

Domaine : noeud de l'arborescence de la mémoire EEPROM représentant un répertoire ou une zone.

Emetteur : entité responsable de l'émission et de la gestion de la carte ou d'un répertoire.

IK : Issuer Key. Clé secrète de l'émetteur principal d'un répertoire.

EEPROM : Electrically Erasable Programmable Read Only Memory. Mémoire programmable dont le contenu peut être effacé (en partie ou entièrement) par application d'un signal électrique.

EPROM : Erasable Programmable Read Only Memory.
Le contenu de ce type de mémoire peut être effacé (dans son intégralité uniquement) en l'exposant à un rayonnement ultraviolet. Cette propriété n'est pas utilisée dans une carte à puce puisque l'effacement intégral de son contenu n'est pas intéressant.
Ce type de mémoire est utilisé dans les cartes à puce

plutôt qu'une PROM (Programmable Read Only Memory) car une PROM est inscriptible une seule fois dans son intégralité alors qu'une EPROM peut être écrite progressivement.

MAC : Message Authentification Code. Signature électronique.

MAM : Microcalculateur Autoprogrammable Monolythique (SPOM en anglais). Microcircuit composé d'un microprocesseur, de mémoires RAM, ROM et (E)EPROM.

Masque : programme de gestion de la carte écrit dans la ROM du micro-calculateur, et exécuté par le microprocesseur.

Mémoire programmable / mémoire utilisateur : EEPROM ou EPROM.

Module de sécurité : appareil assurant les fonctions de sécurité et la gestion des clés secrètes au sein des serveurs et terminaux d'un réseau de cartes à micro-calculateur.

Mot : ensemble de 4 octets.

PIN : Personnal Identification Number. Code secret identifiant le porteur d'une carte.

RAM : Random Access Memory. Mémoire vive, à accès aléatoire.

ROM : Read Only Memory. Mémoire accessible en lecture uniquement.

Ratification : enregistrement du résultat de la présentation d'un code secret ou d'une clé secrète dans la mémoire d'accès.

Répertoire : espace de la mémoire EEPROM réservé à un gestionnaire d'application, constitué de zones et éventuellement de répertoires de niveau inférieur.

SK : Secondary Key. Clé secrète de l'émetteur secondaire d'un répertoire.

SPOM : Self Programmable One-Chip Microcomputer (cf MAM).

XOR : OU exclusif.

Zone : (fichier) ensemble d'informations du même type.

Liste des figures

Chapitre 2

Chapitre 3

Chapitre 4

Bibliographie

- BRIGHT, R.
  1988 Smart Cards : Principles, Practice, Applications
  (Ellis Horwood Limited)
- BULL CP8
  1990 The TB100 IC-Card Operating System User's Guide
  (réf. TU0187A01)
  1989 Guide Utilisateur de la Carte MP
  (réf. TU0167F01)
- CIVADE A.
  1990 Gemplus : "Fin 91, nous serons implantés au Japon"
  (dans : Finance et informatique n°4/septembre 1990)
- Documents sur la carte MCOS
- GANNE R., SALOMONI B.
  1990 La carte à mémoire
  (Eyrrolles Paris)
- GUEZ F., ROBERT C., LAURET A.
  1988 Les cartes à microcircuit, techniques et applications
  (Masson Paris)
- HAWKES P.
  1990 Identification des titulaires de cartes
  Nouvelles Technologies
  (dans : Bancatique n°60/mai 1990)
- HUBIN J.
  1990 Notes pour un cours de sécurité : cryptologie
  (FNDP 1990-1991)
- JOLY L-N.
  1990 Carte à puce et management des risques.
  (dans : Bancatique n°61/juin 1990)
- LE GOFF T.
  1987 L'irrésistible montée de la puce
  (dans : L'ordinateur Individuel n°93/juin 1987)