Sécurité des cartes à puce

Estelle Pleinet, Tristan Porteries

23 janvier 2022

Table des matières

Résumé

Les cartes bancaires ont connu depuis leur création en 1984 de nombreuses évolutions matérielle de leur puce intégrée, passant de processeur 8bit avec 500 octet de mémoire à des processeur 32 bit couplé à des coprocesseur cryptographique. Ces améliorations sont le résultats d'améliorations de la sécurité des protocoles entre la carte bancaire et le terminal menant à des transactions. Même protocoles faisant l'objet d'attaques importantes comme a pu le démontré Serge Humpic en 1998.

Ce rapport aborde une description général du fonctionnement des cartes à puce puis se concentre sur les sécurités cryptographique des cartes à puces.

CHAPITRE 1

Description

1.1 Usage

Les cartes à puces forment une large classe de carte intégrant une puce électronique passive. Cette puce est alimentée au contact d'un terminal et ce dernier interroge la carte pour diverse raison en fonction des contextes. La communication entre la puce et le terminal se réalise généralement par un ensemble de broche de contact.

Ainsi il est possible de trouver des carte à puce pour les objectifs suivants :
— carte mémoire
— carte d'accès
— carte bancaire
— carte sim

La grande majorité d'entre elles permettent au terminal d'authentifier l'utilisateur afin d'autoriser un accès, un paiement ou encore un connexion à un réseau.

1.2 Communication

La communication avec le terminal peut se réaliser par des contacts électriques en façade de la puce. Ces contacts permettent l'alimentation de la puce

FIGURE 1.1 – Différent types de cartes à puce

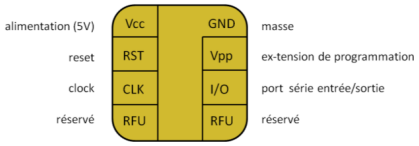

FIGURE 1.2 – Broche des cartes à puces bancaires selon la norme EMV

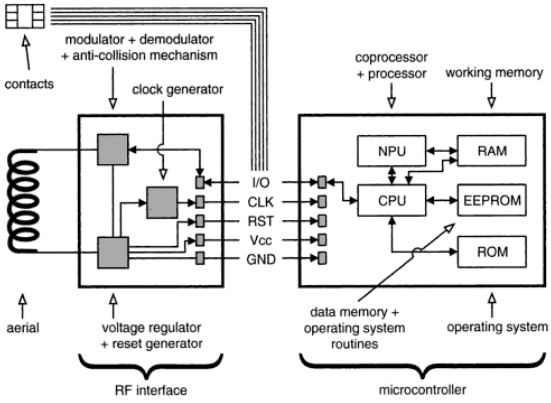

FIGURE 1.3 – Exemple de diagramme électronique d'une puce bancaire

et l'échange de donnée avec le terminal.

Il existe de nombreuse normes définissant ces broches ainsi que la forme des communications. Ces normes ont pour point communs d'avoir au moins les broches suivantes :

— alimentation
— horloge
— port de communication série bidirectionnel

Le terminal est client de la communication, c'est à dire qu'il est le seul à initier la communication en envoyant des requêtes à la puce. Cette dernière doit répondre, à son tour, par le même canal.

1.3 Architecture

Derrière ses broches se cache un circuit électronique regroupant divers composants. Ici aussi les composant varient en fonction du type de puce et de la norme utilisée. Pour les cartes bancaires, la puce sera composée de :

— micro contrôleur
— mémoire permanente privée (ROM)
— mémoire temporaire (RAM)

La mémoire permanente contiendra le programme de la puce lui permettant de répondre aux requêtes du terminal ainsi que des données publiques (utilisateur, numéro) et des secrets (code, clé privée RSA. . .). Ces données sont inscrites dans la mémoire au moment de la fabrication sans que les technologies actuelles ne puissent les extraire par un canal alternatif.

CHAPITRE 2

Protocoles sécurisé

Le reste du rapport traite des échanges sécurisés entre la puce d'une carte bancaire et un terminal de paiement. Ces échanges se regroupe en 3 niveaux de sécurité de plus en plus difficile à tromper

1. authentification du code de déverrouillage
2. authentification des données de la puce
3. authentification en ligne des données de la puce

En fonction de leur temps de calcul certaines de ces sécurités ne sont pas toujours appliquées, par exemple la authentification en ligne ne se fait que à partir d'un seuil de transaction (environ à 100€) et dans 20% des cas.

2.1 Vérification du code

Cette étape assure que l'utilisateur possède un secret partagé avec la carte bancaire : le code d'accès.

Ce code est demandé par le terminal puis transmis à la puce. Dans les anciennes normes ce code était transmis en clair à la puce, le nouvelle normes impose un chiffrage asymétrique (RSA) de ce code.

Une fois le code transmit à la carte celle ci lit sa mémoire interne afin de retrouver le code enregistré et de le comparer avec le code reçu. Si la comparaison est effective la puce indique au terminal que l'authentification de la transaction peut continuer.

Cette vérification empêche les attaques simples mais n'empêche pas un attaquant de créer une carte répondant toujours OK à la réception d'un code. Ce type de carte est nommé une yescard. Pour lutter contre cette attaque d'autre méthode vont viser à authentifier la carte.

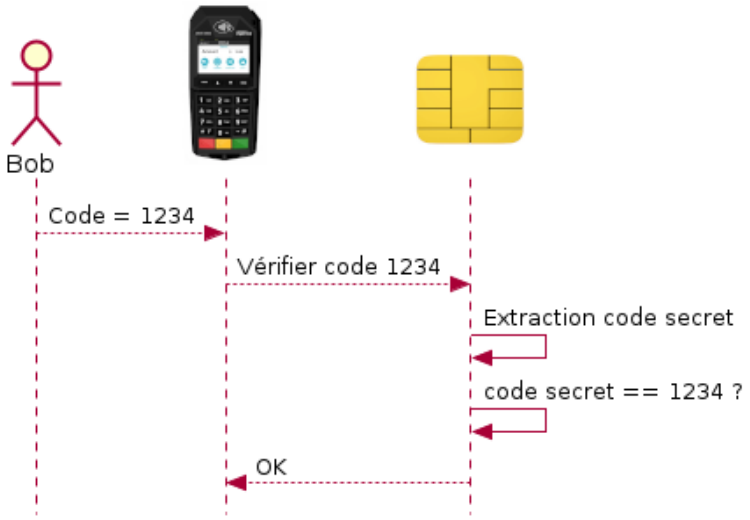

FIGURE 2.1 – Séquence de authentification du code secret

2.2 Authentification de la carte

Après la vérification du code de la carte d'autres authentifications peuvent se rajouter. Ses authentifications sont regroupés en deux catégories en fonction de leur accès aux ressources externes. Elles permettent de s'assurer que la puce contient des informations générer par la banque au lieu d'informations copiées ou créées par des attaquants.

La première catégorie concerne les authentifications hors lignes grâce à deux méthodes, une statique rapide mais faible et l'autre dynamique empêchant les attaquants de copier les données publiques d'une carte.

La deuxième catégorie rassemble les méthodes d'authentifications en ligne se basant sur un défi généré aléatoirement par un centre de contrôle de la banque émettrice de la carte.

2.2.1 Authentification en ligne

L'authentification en ligne et une des première méthode d'authentification des carte bancaire bénéficiant d'une sécurité importante. Cette méthode se base sur le partage d'un secret entre la carte et un centre de contrôle, ce dernier identifié la carte par ses informations publiques (numéro, nom...) et génère un nombre aléatoire du défi. Ce nombre est transmit au terminal puis à la puce. La puce et le centre de contrôle calculent alors le chiffrage avec leur secret partagé, chacun de leur côté, puis la puce retransmet par l'intermédiaire du terminal la valeur chiffrée au centre qui à son tour compare la valeur et accepte la transaction le cas échéant.

Plus concrètement la méthode de chiffrement est symétrique et dépend d'un secret identique dans la puce et le centre de contrôle. Au commencement de cette méthode, des algorithmes de chiffrement léger comme Telepass1 nécessitant très peu de temps de calcul et d'espace mémoire, par la suite DES s'est

imposé comme l'algorithme majoritaire suffisamment performant pour cette authentification.

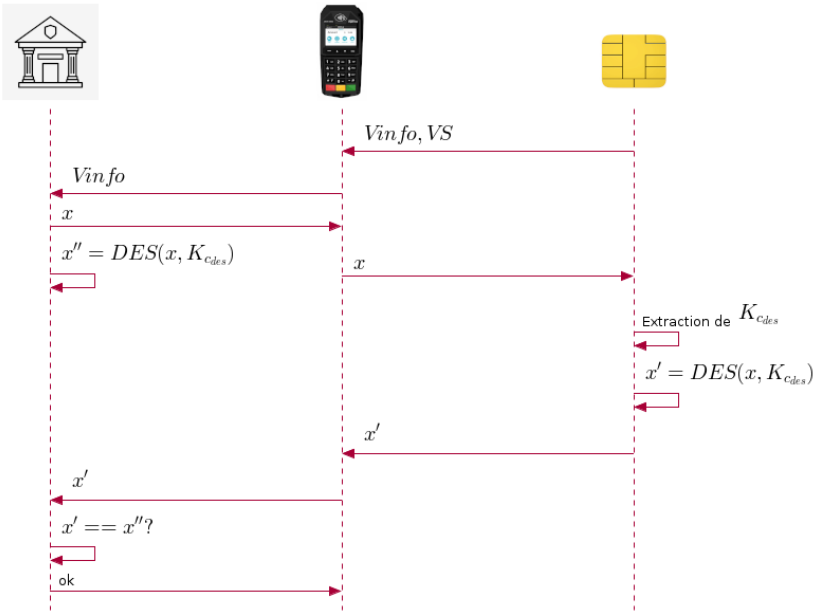

FIGURE 2.2 – Séquence d'authentification en ligne avec comme secret partagé $K_{c_{DES}}$, $Vinfo$ dénote un tuple contenant les informations publique de la puce permettant son identification.

L'avantage de cette authentification est sa robustesse définie par la sécurité du protocole DES et la protection des données secrètes sur la puce et le centre de contrôle. En contrepartie la communication avec un centre de contrôle apporte un délai de traitement conséquent pouvant introduire des files d'attente dans les magasins, c'est pourquoi cette méthode est préféré seulement à partir d'un seuil élevé (par exemple 100€) et dans 20% des cas afin de limiter les potentiels fraudes.

2.2.2 Authentification hors ligne

Dans le cas ou la méthode en ligne est refusée, l'authentification se passe seulement hors ligne avec des informations connus par le terminal. Ces informations sont des clés publiques de cryptage asymétrique (typiquement RSA) pouvant être connu de tous sans pour autant dévoiler tout le secret sans la clé privée.

Authentification statique hors ligne

La première authentification hors ligne possible par les premières puces de carte bancaire étaient limitées par leurs puissances de calcul. Ainsi les premières authentifications sont dites statique car le terminal est responsable de tout le calcul cryptographique nécessitant seulement des informations complémentaire de la puce.

La banque émettrice de la carte possède une paire de clés asymétrique $K_{b_{priv}}$ et $K_{b_{pub}}$, la clé publique $K_{b_{pub}}$ est connu du terminal. Lors de la construction de la puce les informations publiques de la carte sont définies par le tuple $Vinfo = (numero, nom, prenom, date\_de\_validite)$. Conjointement une signature de ses informations est calculé grâce à $K_{b_{priv}}$ de la manière suivante : $VS = RSA(Vinfo, K_{b_{priv}})$. Ces deux informations sont inscrites dans la mémoire secrète de la puce.

Après une vérification du code, la carte transmet $Vinfo$ et VS au terminal, ce dernier déchiffre VS avec la clé publique de la banque $Vinfo' = RSA(VS, K_{b_{pub}})$ et compare $Vinfo'$ avec $Vinfo$, dans le cas d'informations signées par la banque la comparaison est effective.

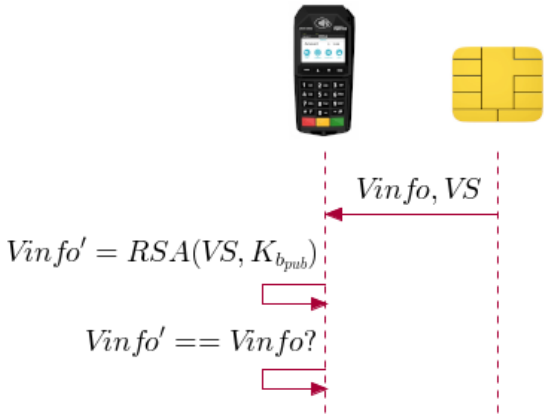

FIGURE 2.3 – Séquence d'authentification hors ligne statique

Failles

L'authentification statique hors ligne permet de s'assurer que les informations sont justes et d'empêcher les attaquants de créer de fausses cartes de toute pièce. Cependant un attaquant peut toujours copier les informations publiques de la carte ($Vinfo$ et VS) et les inscrire dans une yescard carte, outrepassant la vérificaiton du code et se faisant passer pour une autre carte.

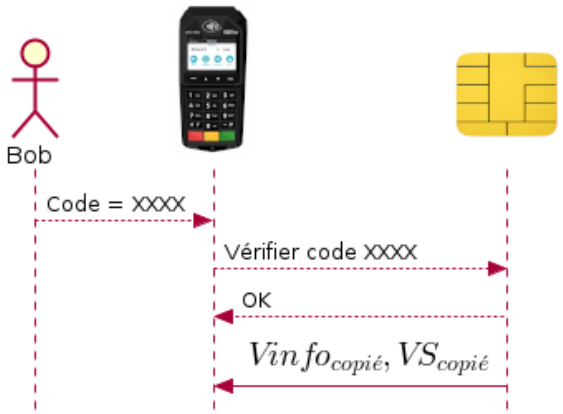

FIGURE 2.4 – Séquence d'une attaque par une yescard copiant les $Vinfo$ et VS d'une carte existante.

En 1998 Serge Humpic dévoila une attaque ignorant l'authentification statique. Cette attaque se base sur deux failles, la première profite du changement de session possible par l'utilisation d'une yescard, la deuxième bénéficie du déchiffrage de la clé privée $K_{b_{priv}}$. À cette époque cette clé utilisait 320 bits la rendant vulnérable au factorisation.

Serge Humpic possédait alors la capacité de créer des yescard contenant des informations nouvelles (compte bancaire inexistant) mais pour autant valide pour la banque.

Suite à l'affaire Humpic le groupement des carte bancaires à pris la décision d'augmenter la taille des clé vers un nouveau standard empechant toute factorisation à l'heure actuelle. Dans le même effort un nouveau protocole fut adopté vers le début des années 2000 découlant de la norme EMV.

Authentification dynamique hors lignes

La norme EMV introduit une authentification fusionnant le fonctionnement de l'authentification en ligne et celle hors ligne statique : le terminal transmettra un défi à la puce qui devra produire le résultat attendu en fonction d'un secret dans sa mémoire caché. Cette authentification est appelé dynamique par la nécessité d'un calcul de chiffrement sur la puce.

Plus concrètement chaque puce au moment de sa fabrication se verra attitré une paire de clé RSA $K_{c_{priv}}$ et $K_{c_{pub}}$, ainsi qu'une signature de la clé publique par une autorité de certification externe : $C = RSA(K_{c_{pub}}, K_{a_{priv}})$, avec $K_{a_{priv}}$ et $K_{a_{pub}}$ la paire de clé de l'autorité de certification. Le terminal connait $K_{b_{pub}}$ et $K_{a_{pub}}$.

Au début de la transaction après la vérification du code, la puce transmettra $Vinfo$, VS, $K_{c_{pub}}$ et C, le terminal commencera pas vérifier $Vinfo$ grâce à VS selon la procédure défini par l'authentification statique. Ensuite de la même façon la justesse de $K_{c_{pub}}$ est vérifié avec la signature C en calculant le déchiffré $K'_{c_{pub}} = RSA(S, K_{a_{pub}})$ et en comparant $K_{c_{pub}}$ et $K'_{c_{pub}}$. Si toutes les comparaisons sont réussites alors les informations $Vinfo$ et $K_{c_{pub}}$ sont reconnus comme émise par la banque et ne peuvent pas être créées par des attaquants.

Le terminal va alors tirer un nombre aléatoire N et le transmettre à la puce, cette dernière devra chiffrer N avec sa clé privée $K_{c_{priv}}$, $N' = RSA(N, K_{c_{priv}})$ et renvoyer la valeur au terminal. Une fois le résultat reçu le terminal va déchiffré N' avec la clé publique de la puce pour trouver $N'' = RSA(N', K_{c_{pub}})$ et le comparer avec le N d'origine.

Une égalité entre N et N'' implique que la puce possède la clé privée associé à $K_{c_{pub}}$. Il resterait alors à l'attaquant la possibilité de créer une carte possédant une paire de clé RSA généré par celui ci et d'envoyer le $K_{c_{pub}}$ désiré au terminal. Néanmoins la signature de la clé publique de la puce par la clé privée de l'autorité de certification empêche l'utilisation d'une paire de clé nom certifié.

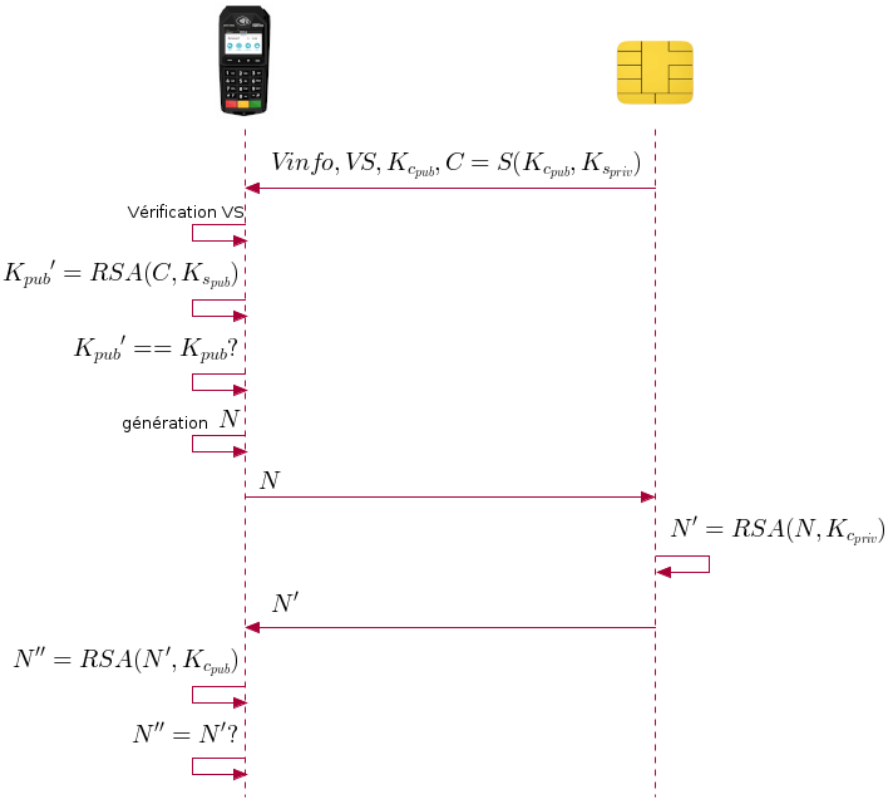

FIGURE 2.5 – Séquence d'authentification dynamique

Cette méthode d'authentification combine la nécessité d'un secret(clé privée) connu seulement par la puce et l'obligation d'avoir un secret conformant (clé privée et publique cohérente) à la signature d'une autorité de certification.

Conclusion

La sécurité des cartes à puce bancaire sont constamment en amélioration avec des algorithmes de plus en plus robuste empêchant au mieux les attaquant d'usurper des cartes ou de créer de nouvelles cartes avec cependant un compromis sécurité et temps d'authentification.

Les différentes méthodes expliquées dans ce rapport ont toutes bénéficiées des progrès en cryptographie et notamment de l'invention de la cryptographie asymétrique omniprésente dans les échanges avec une puce.