Protocole de la carte bancaire

IT218, Paul Dorbec

1 Les protocoles de la carte bancaire

Nous nous intéressons ici au protocole utilisé au cours d'un paiement par carte bancaire chez un commerçant.

Il faut noter que chaque carte bancaire contient dans sa puce les informations de la carte (nom, prénom, numéro de carte, date de validité) que l'on désigne par *data*, une signature $\mathcal{S}$, et un calculateur lui permettant de coder un message grâce à un système de codage symétrique (le DES) de clé K, qui est propre à la carte. En outre, le code confidentiel y est aussi stocké de façon cryptée. La signature $\mathcal{S}$ n'est pas calculée par la carte, elle est directement enregistrée dans la carte. Elle se calcule à partir des informations *data*. On commence par calculer le résultat $f(data)$ d'une fonction de hachage f, que l'on encode ensuite avec la clé privée K_B^{-1} d'un système à clé publique (RSA) :

$$\mathcal{S} = E_{K_B^{-1}}\Big(f(data)\Big)$$

Les clés K_B (publique) et K_B^{-1} (privée) ainsi que la fonction de hachage f sont propres à la banque[1], donc communes à toutes les cartes.

Le protocole de paiement peut se décrire ainsi :

1. Alice (**A**) entre sa carte bancaire dans le terminal (**T**) .
2. Le commerçant saisit le montant de la transaction sur **T**

Authentification de la carte :

3. La puce de la carte (**P**) fournit à **T** les informations *data* ainsi que sa signature $\mathcal{S}$
4. **T** compare $f(data)$ avec $D_{K_B}(\mathcal{S})$

Vérification du code secret :

5. **T** demande le code à **A**
6. **A** fournit son code
7. **T** transmet le code à **P**

[1] en fait de banque, il s'agit d'un serveur central dédié à la vérification distante des cartes

8. **P** atteste de la validité du code auprès de **T** .

Authentification en ligne : dans les terminaux, cette partie est effectuée dans environ 20% des cas et uniquement lorsque le montant de la transaction dépasse 100 euros.

9. **T** demande à la banque (**B**) d'initialiser une session d'authentification
10. **B** envoie un nombre aléatoire x à **P**
11. **P** calcule $y = E_K(x)$ et transmet y à **B**
12. **B** compare y avec $E_K(x)$ où K est la clé que **B** associe à la puce **P**
13. **B** confirme à **T** la validité de la carte

Question 1.1 *Expliquez le rôle de l'authentification en ligne. Selon vous, pourquoi n'est elle pas systématiquement effectuée ? Pourquoi est elle nécessaire de temps en temps ?*

Question 1.2 *Dans la première étape, c'est le résultat d'une fonction de hachage qui est encodé avec la clé secrete K_B^{-1}. Expliquez ce choix (notamment par rapport à l'étape 4 du protocole).*

Question 1.3 *Pourquoi un système cryptographique à clé publique est nécessaire pour l'étape 4 ? Pourquoi la banque peut-elle se permettre d'utiliser un système à clé symétrique pour l'authentification ?*

Ce système présente deux failles de sécurité, une logique et une cryptographique. Ces deux failles ont été négligées jusqu'en 1998 parce que les banques considéraient que la sécurité du système reposait aussi d'une part sur le secret du protocole employé, et d'autre part sur la difficulté de créer une réplique de carte.

La faille cryptographique vient de la taille trop faible des clés du système à clé publique. En 1998, Serge Humpich a ainsi réussi à déduire K_B^{-1} de K_B et à créer ce qui fut appelé les "yescards".

Question 1.4 *Expliquez comment à partir de la connaissance de K_B^{-1}, Serge Humpich a pu créer une carte bancaire acceptée dans les terminaux.*

Question 1.5 *Selon vous, quelle est la faiblesse logique du protocole. Comment Mallory peut-il se créer une carte et faire tous les achats (<100 euros) qu'il souhaite ? (indice : emprunter sa carte quelques heures à Alice)*

Petit complément d'information : Pour compenser la faiblesse cryptographique, les anciennes clés de 320 bits utilisées pour le système de codage RSA furent remplacées par des clés de 768 bits. Mais la faille logique ne fut pas corrigée avant mai 2007. Une des dernières escroqueries liées à cette faille date de février 2007 et correspondrait à un montant de près de 640 000 euros.

Le protocole suivant permet de pallier à cette faille (seules les étapes d'authentification de la carte et de vérification du code secret sont détaillées). Il utilise deux nouvelles clé K_C et K_C^{-1} d'un système à clé publique. Ces clés sont propres à la carte.

1. **P** fournit à **T** les informations *data*, sa signature $\mathcal{S}$ et $x = E_{K_B^{-1}}(K_C)$
2. **T** compare $f(data)$ avec $D_{K_B}(\mathcal{S})$ et décode $K_C = D_{K_B}(x)$
3. **T** génère un nombre aléatoire N_T et l'envoie à **P**
4. **P** code N_T et envoie $y = E_{K_C^{-1}}(N_T)$ à **T**
5. **T** compare N_T et $D_{K_C}(y)$
6. **T** demande le code à **A**
7. **A** fournit son code (exemple : 1234)
8. **T** transmet le code crypté $E_{K_C}(1234)$ à **P**
9. **P** atteste de la validité du code auprès de **T** .

Question 1.6 *Pourquoi ce protocole compense-t-il la faille logique ?*