Principe et Algorithme Cryptographiques : DS n°2

première session

Université de Lille-1
FIL — PAC

22 mai 2015

Ce sujet est composé de 3 pages. Tous les documents sont autorisés. Vos réponses doivent **absolument** être justifiées.

1 Cartes bleues

Une norme internationale nommée EMV spécifie le fonctionnement de la cryptographie dans les cartes bancaires contenant un microprocesseur (les plus répandues en Europe). On va admettre que ce microprocesseur contient une mémoire qui n'est pas facilement accessible, et qui permet donc de stocker des données secrètes. La carte bleue contient aussi des données non-secrètes, qui ont vocation à être communiquées au terminal de paiement : le numéro du compte bancaire à débiter, par exemple, ainsi que des clefs publiques et des certificats. Le système RSA est utilisé en toute circonstance.

L'usage de la cryptographie vise principalement à empêcher la fraude, par les clients (qui possède une carte bleue) ou par les marchands (qui possèdent un terminal de paiement). Dans ce système, chaque banque possède une ou des clef publiques. Une autorité de certification spécifique au système EMV fournit aux banques des certificats. Les marchands possèdent la clef publique de cette autorité.

On va supposer que des fraudeurs peuvent espionner les communications entre la carte et le terminal de paiement.

La norme prévoit deux niveau d'identification de la carte par le terminal.

Identification « statique ». Elle offre le niveau de sécurité le plus faible. La carte contient seulement une signature des données spécifiques à la carte (le numéro du compte, etc.) par la banque, ainsi que le certificat de la banque.

▷ **Question 1 :** Que doit vérifier le terminal ? Que doit-il contenir à l'avance pour cela ?

Le terminal doit posséder la clef publique de l'autorité de certification. Il doit récupérer le certificat de la banque sur la carte (c.a.d. la clef publique de la banque signée par l'autorité de certification), et vérifier ce certificat. Ensuite, il doit vérifier la signature des données contenues dans la carte, en se servant de la clef publique de la banque.
Note : le terminal ne connaît pas a priori la clef de la banque pour des raisons pratiques. Lorsqu'une banque se créé (on a vu apparaitre les banques en lignes ces dernières années), ou lorsqu'une banque change de clef, il faudrait mettre à jour des millions de terminaux de paiement dans le monde entier !

▷ **Question 2 :** Quelle manoeuvre de fraude est rendue impossible par cette vérification ?

On ne peut pas fabriquer une carte bleue pour n'importe quel numéro de compte (qu'on choisirait) : en effet, il faudrait pour ça ou bien forger une signature à la place de la banque, ou bien utiliser une clef de banque fictive, c'est-à-dire forger un certificat à la place de l'autorité de certification.

▷ **Question 3 :** Quelle autre manoeuvre de fraude *n'*est *pas* rendue impossible par cette vérification ?

Ceci n'empêche pas de *cloner* une carte. Comme ces données signées sont transférées de la carte vers le lecteur, on peut les intercepter, et les « incruster » dans une autre carte.

Identification « dynamique ». Elle offre un niveau de sécurité plus élevé. La carte contient ainsi des données bancaires, ainsi qu'une paire de clefs RSA. Pour authentifier la carte, le terminal lui envoie des données (le numéro de compte du marchand, la date, ainsi qu'un nombre aléatoire imprévisible), et la carte en renvoie une signature.

▷ **Question 4 :** Une partie des données présentes sur la carte doit toujours être signée par la banque. Lesquelles ?

La banque doit signer : le numéro de la carte bleue (comme avant) ainsi que la clef publique propre à la carte. En fait la banque doit établir un certificat pour la clef publique de la carte.

▷ **Question 5 :** Que doit récupérer le terminal sur la carte ? Que doit-il vérifier ?

Le terminal doit récupérer toutes les clefs publiques et les certificats : celui de la carte établi par la banque, et celui de la banque établi par l'autorité de certification. Il doit vérifier que toutes les clefs publiques sont correctement signées par leurs garants respectifs, et que la signature effectuée par la carte est valide.

▷ **Question 6 :** Pourquoi cette méthode améliore-t-elle la sécurité par rapport à la précédente ?

Ceci empêche le clonage des cartes. En effet, en produisant une signature, la carte démontre qu'elle possède une clef secrète, qui elle ne sort jamais de la carte... une clef secrète dont la clef publique est authentifiée par la banque. Donc, pour répondre au défi lancé par le terminal de paiement, soit il faut faire une fausse signature à la place de la banque, soit il faut faire ue fausse signature à la place d'une carte déjà existante.

Transaction. Une fois que la transaction a lieu (par exemple, l'utilisateur tape son code PIN pour l'approuver), la carte génère un Certificat de Transaction (CT) et le transmet au terminal. Celui-ci l'envoie à la banque pour obtenir le paiement.

Le Certificat de Transaction est constitué de la description de l'opération (numéros des comptes de départ et d'arrivée, montant, unité monétaire, *numéro de la transaction réalisé par la carte*, etc.) ainsi que d'un MAC de ces données. L'CBC-MAC est utilisé, dans lequel l'AES fait office de système de chiffrement par bloc. La clef qui sert à calculer ce MAC est stockée dans la mémoire secrète de la carte.

▷ **Question 7 :** Quelle fraude pourrait avoir lieu si le MAC n'était pas présent ?

Le terminal de paiement pourrait se faire passer pour n'importe qui et envoyer à la banque des transactions au profit de son propriétaire, et au détriment de clients choisis au hasard. Ou bien il pourrait augmenter le montant de transactions auxquelles les clients consentaient pourtant, etc.

▷ **Question 8 :** Quelle fraude pourrait avoir lieu si le numéro de la transaction en cours n'était pas inclu dans les données MACées ?

Le terminal de paiement pourrait envoyer plusieurs fois le même certificat de transaction, et du coup « rejouer » des transactions autant de fois qu'il le souhaite (le pauvre client serait prélevé plusieurs fois).

Dérivation de clef. La norme recommande que la clef de MAC de chaque carte soit obtenue en chiffrant le numéro de la carte par une clef « maitresse » détenue par la banque.

▷ **Question 9 :** Quel est l'intérêt de cette pratique (pour les banques) ?

Ceci évite aux banques d'avoir à générer pour chaque carte un nombre aléatoirement de bonne qualité, puis le stocker, en plus du numéro de carte, de compte, etc.

▷ **Question 10 :** Pourquoi faut-il évidemment stocker la clef *dérivée* sur chaque carte, et pas la clef maitresse ?

Si jamais quelqu'un parvient à accéder à la zone de mémoire « secrète » d'une carte en service, accéder à une clef dérivée compromet *cette* carte. Accéder à la clef maitresse compromet *toutes* les cartes.

2 Alice et Bob sont accusés de terrorisme

Alice, Bob, Charlie et Dorothée utilisent un échange de clef Diffie-Hellman *statique* pour communiquer entre eux. Ceci signifie que chacun possède une clef secrète (x_a pour Alice, x_b pour Bob, etc.). À chacune de ces clefs secrète correspond une clef publique $pk_a = g^{x_a} \bmod p$, etc. On précise qu'ils se sont mis d'accord, il y a longtemps, sur un grand nombre premier p et un « générateur » g de $\mathbb{Z}_p$ qui est d'ordre q (un nombre premier très grand).

Pour communiquer entre eux, Alice et Bob utilisent la méthode de Diffie-Hellman pour obtenir une clef partagée entre eux deux seulement : $s_{ab} = g^{x_a x_b} \bmod p$. Concrètement, nos amis utilisent ensuite l'AES en mode CBC pour chiffrer leurs échanges.

▷ **Question 11 :** Comment tirer une clef AES de s_{ab} ?

Le problème c'est que s_{ab} est trop gros, et on ne sait pas vraiment si les valeurs possibles sont uniformément réparties. Il suffit de garder les 128 bits de poids faible de SHA-256(s_{ab}).
En fait, il est également possible d'utiliser directement les 128 bits de poids faibles de s_{ab}. Ceci n'est pas évident à priori, car rien ne dit que les bits de poids faibles de s_{ab} sont uniformément distribués et imprédictibles, même lorsque x_a et x_b le sont. Il a été démontré (avec des techniques sophistiquées) que c'est bien le cas en 2009, dans l'article *Optimal Randomness Extraction from a Diffie-Hellman Element*, par Céline Chevalier, Pierre-Alain Fouque, David Pointcheval et Sébastien Zimmer.

L'Etat intercepte tous les échanges chiffrés et possède les clefs publiques de tout le monde. La police accuse (à tort) Alice et Bob de participer à un complot terroriste. Trainés au tribunal, Alice et Bob sont sommés de révèler leurs clefs secrètes x_a et x_b pour que leurs conversations soient rélévées.

Alice et Bob refusent de révéler x_a et x_b, mais acceptent de révéler s_{ab}, au nom du droit à la vie privée de Charlie et Dorothée.

▷ **Question 12 :** Justifiez leur décision.

Avec x_a, le juge pourrait calculer $s_{ac} = {pk_c}^{x_a} \bmod p$, et donc espionner aussi les conversations entre Alice et Charlie. Or Charlie, lui, n'est pas suspecté. Au passage, dans tous les cas de figure le juge ne peut pas apprendre s_{cd}.

Alice et Bob, assez énervés par les tracas qu'ils subissent, décident de jouer aux plus malins : au lieu de transmettre s_{ab} au juge, ils lui donnent une valeur pseudo-aléatoire s_{bidon} qui est différente de s_{ab}. Le juge essaye de déchiffrer leurs échanges chiffrés avec s_{bidon}, et obtient du charabia incompréhensible. Alice et Bob tentent de se justifier en prétendant que depuis le début ils s'envoie des messages pseudo-aléatoires chiffrés, juste pour pour embêter les dispositifs d'espionnage de masse de la population.

Le juge se rend compte qu'il ne possède pas de moyen infaillible de démontrer qu'Alice et Bob lui mentent.

▷ **Question 13 :** Pourquoi ?

Le juge pourrait avoir deux stratégies pour « confirmer » que la chaine de bits qu'on lui a remise est bien la clef partagée entre Alice et Bob.
Tout d'abord, il pourrait essayer de s'en servir pour effectuer le déchiffrement et voir ce qui se passe. Cette option, ici, est exclue, car si Alice et Bob s'étaient vraiment envoyé des messages pseudo-aléatoires, alors le texte clair ne pas distinguable du charabia qu'on obtiendrait en déchiffrant avec une mauvaise clef. Du coup, que le juge déchiffre avec la bonne clef ou avec la mauvaise clef, il obtiendrait sensiblement la même chose. Attention cependant, ceci passe sous silence d'éventuelles vérifications d'intégrité dans les paddings.
L'autre stratégie que le juge pourrait utiliser consiste à déterminer si (pk_a, pk_b, s_{bidon}) forme bien un triplet Diffie-Hellman. Ceci est exactement le problème Diffie-Hellman décisionnel, et il y a tout un tas de situations où ne sait pas le résoudre (par exemple, si g est un élément d'ordre élevé qui est un résidu quadratique).

Le greffier propose alors le protocole suivant, pour établir l'éventuelle mauvaise fois d'Alice et Bob. Il s'agit d'un protocole qui sert à tester si un nombre s est bien égal à $g^{x_a x_b} \bmod p$, mais sans révéler ni x_a ni x_b.

1. Alice et Bob choisissent au hasard un nombre $u \in \mathbb{Z}_p$.
2. Ils calculent $\alpha \leftarrow g^u \bmod p$ et $\beta \leftarrow {pk_b}^u \bmod p$ puis envoient le tout au juge.
3. Le juge choisit au hasard un nombre $d \in \mathbb{Z}_p$ (le « défi ») et l'envoie aux suspects.
4. Alice calcule $r = u + dx_a \bmod q$ (la « réponse ») et l'envoie au juge.
5. Le juge vérifie que :
$$\begin{cases} g^r & \equiv \alpha \cdot {pk_a}^d & \bmod p \\ {pk_b}^r & \equiv \beta \cdot s^d & \bmod p \end{cases}$$
Si ce n'est pas le cas, c'est qu'Alice et Bob ont triché pendant l'exécution du protocole, ou bien que s n'est pas leur clef partagée.

Alice et Bob sont honnêtes

Dans ce cas-là, la valeur de s que le juge possède est bien $s = s_{ab}$, c'est-à-dire $s \equiv g^{x_a x_b} \bmod p$.

▷ **Question 14 :** Justifiez que le juge va toujours exécuter le protocole de manière satisfaisante.

▷ **Question 15 :** Expliquez pourquoi le juge n'apprend pas les clefs secrètes x_a et x_b.

Tout d'abord, le juge n'apprend pas la valeur u. En effet, si le juge pouvait déduire u de a et b, c'est qu'il serait capable de résoudre le problème du logarithme discret.
Ensuite, le juge apprend $r = u + dx_a$. Le point important consiste à justifier qu'il ne peut pas en déduire x_a. En effet, il connaît d. L'argument, c'est que u est inconnu du juge et complètement aléatoire : u agit comme un masque jetable qui empêche le juge de connaître dx_a, et donc de connaître x_a.

Alice et Bob trichent

Note : *les 4 questions suivantes sont un peu plus mathématiques, mais vous êtes largement guidés...*

Dans ce cas-là, la valeur s que le juge possède n'est pas s_{ab}. On écrit donc $s = g^z \bmod p$ (avec $z \not\equiv x_a x_b \bmod q$).

On va voir que quoi que fassent Alice et Bob, le protocole échoue avec très forte probabilité. Alice et bob peuvent faire n'importe quoi, y compris calculer α, β et r autrement que ce qui est prévu. On supposer qu'ils renvoient $\alpha = g^{u_1} \bmod p$ et $\beta = g^{u_2} \bmod p$.

▷ **Question 16 :** Quelle est la valeur « normale » de u_2 qui devrait se présenter si Alice et Bob respectaient le protocole ?

▷ **Question 17 :** Le juge effectue d'abord la première des deux vérifications. Montrez que si cette première congruence est vraie, alors on a : $r \equiv u_1 + dx_a \mod q$.

▷ **Question 18 :** Le juge effectue ensuite la deuxième vérification. Montrez que si la deuxième congruence est vraie, alors on a : $u_1 x_b - u_2 \equiv d(z - x_a x_b) \mod q$.

▷ **Question 19 :** On rappelle qu'Alice et Bob doivent choisir u_1 et u_2 *avant* de connaître le défi d. Montrez qu'ils n'ont quasiment aucune chance de s'en sortir.

Pour réussir à passer les tests, Alice et Bob doivent réussir à vérifier la congruence de la question précédente. Le hic, c'est qu'ils doivent envoyer α et β, c.a.d. choisir u_1 et u_2 *avant* de connaître d. Ils peuvent s'y prendre comme ils veulent : le choix de α et β

Le juge a autre chose à faire

Comme il ne veut pas avoir à s'occuper de cette affaire manifestement boiteuse, le juge propose la modification suivante du protocole : au lieu qu'il fournisse un défi aléatoire, Alice et Bob calculent de leur côté $d = H(pk_a \parallel pk_b \parallel s \parallel \alpha \parallel \beta)$. Du coup, il suffit qu'ils fassent les calculs dans leur coin, et le juge n'a plus qu'à vérifier la « preuve ».

▷ **Question 20 :** Est-ce raisonnable ?

Ceci est raisonnable.