# Principe et Algorithme Cryptographiques : examen

première session / documents et calculatrice autorisés / durée 2h / 2 pages

10 mai 2017

## 1 Paiement en ligne

La société AMOS et son prestataire WorldOffline offrent aux commerçants en ligne un mécanisme de paiement sécurisé. Lorsqu'il souhaite effectuer le paiement, l'acheteur envoie une requête HTTP POST à un serveur d'AMOS. Il est alors redirigé vers une page de paiement. Cette requête est composée de deux champs :

- `data` : un dictionnaire qui contient (entre autre) le montant de la transaction, le numéro de contrat du commerçant auprès de la banque, le numéro de carte de crédit de l'acheteur, sa date d'expiration et le cryptogramme au dos, ainsi qu'un identifiant unique de transaction.
- `seal` : SHA-256(`data` $\|$ $K$), où $K$ est une clef secrète partagée à l'avance entre AMOS et le commerçant.

A l'issue de la transaction sur la page de paiement, AMOS envoie une requête POST à une adresse spécifiée par le commerçant. Cette requête contient deux champs :

- `data` : un dictionnaire qui contient (entre autre) l'identifiant unique de la transaction, et un code réponse (paiement accepté, refus par la banque, timeout, problème technique, etc.)
- `seal` : SHA-256(`data` $\|$ $K$).

▷ **Question 1 :** Quel est le rôle du champ `seal` ? Quelle primitive cryptographique réalise-t-il ?

▷ **Question 2 :** De quelle manière est-ce que les acheteurs pourraient tricher si le `seal` n'était pas présent ?

▷ **Question 3 :** La société AMOS vérifie soigneusement que l'identifiant unique de transaction n'a jamais été utilisé précédemment. De quelle manière le commerçant pourrait tricher si ce n'était pas le cas ?

▷ **Question 4 :** Quel problème de sécurité majeur se pose si toutes ces requêtes sont envoyées « en clair » sur le réseau ?

▷ **Question 5 :** En réalité, ces requêtes sont envoyés en HTTPS. Expliquez par quel mécanisme ceci garantit à l'acheteur qu'il envoie son numéro de carte bancaire à une banque et pas à un serveur pirate.

▷ **Question 6 :** L'outil `openssl` révèle que la connection HTTPS utilise DH+RSA+SHA384+AES256CBC. Expliquez le rôle de chacun de ces algorithmes.

▷ **Question 7 :** Pourquoi ne pas avoir utilisé `seal` = SHA-256($K$ $\|$ `data`) ?

## 2 Retour sur les TPs

Le protocole STP est un protocole jouet créé exprès pour les TPs de PAC. Il permet à un client de s'identifier auprès d'un serveur avec un mot de passe. Voici comment il fonctionne :

1. Le client envoie son `username` au serveur.
2. Le serveur répond avec un `nonce` aléatoire.
3. Les deux calculent la clef de session $K$ = `password` $\|$ `nonce`.
4. Le client envoie une requête prédéfinie, toujours la même, chiffrée avec $K$.
5. La session du client est ouverte, et les échanges qui suivent sont chiffrés avec la clef $K$.

Dans le TP, la dernière étape se réalise en envoyant le dictionnaire `{'url': '/bin/stp/gateway', 'method': 'GET'}` chiffré avec $K$.

▷ **Question 8 :** Expliquez comment un adversaire passif qui espionne une session du protocole peut effectuer une attaque par dictionnaire « hors-ligne » sur le `password`.

▷ **Question 9 :** Le dictionnaire (anglais) fourni dans le TP contient $\approx 2^{16}$ mots, qui font tous 9 lettres (pour simplifier). Dans chacun des cas suivants, estimez le temps que prendrait une recherche exhaustive sur le mot de passe. Indiquez si cela vous semble réalisable par un particulier sans moyens importants.

a. Mon mot de passe est formé de 4 mots choisis aléatoirement dans le dictionnaire.

b. Mon mot de passe est formé d'un mot aléatoire suivi de 4 chiffres aléatoires.

c. Mon mot de passe est un mot de choisi aléatoirement, et chacune des lettres a été mise en majuscule avec probabililité 1/2.

# 3 Partage de secret

Cet exercice est consacré au schéma de partage de secret de Shamir. Il permet de partager un *secret* en autant de *parts* qu'on le souhaite. Lors du partage, un *seuil* $T$ est fixé : avec moins de $T$ parts, on n'obtient rien, tandis qu'avec $T$ parts on peut reconstituer le secret. L'acteur qui effectue le partage est nommé le *dealer*.

Dans tout le processus, on suppose qu'on grand nombre premier $p$ (occupant $k$ bits) a été fixé et qu'il est connu de tous. Pour partager un secret $S$ (qui est un nombre $S$ avec $0 \leq S < p$) avec un seuil de $T \geq 2$, le *dealer* choisit uniformément au hasard $T-1$ élements de $\mathbb{Z}_p$ notés $A_1, \ldots, A_{T-1}$. La $i$-ème part ($i \geq 1$) s'obtient en calculant :

$$P_i = \left(i^{T-1} \cdot A_{T-1} + \cdots + i^2 \cdot A_2 + i \cdot A_1 + S\right) \ \% \ p.$$

▷ **Question 10 :** Quelle est la complexité, en fonction de $k$ et $T$, du calcul d'une part ?

▷ **Question 11 :** Quel gros problème y a-t-il si on autorise la part avec $i = 0$ ?

▷ **Question 12 :** Que se passe-t-il si le *dealer* publie accidentellement les coefficients $A_1, \ldots, A_{T-1}$ ?

Le mécanisme offre la *sécurité inconditionnelle*, comme on va le démontrer.

▷ **Question 13 :** Supposons qu'on dispose d'une seule part $P_i$ d'un secret partagé. Montrez que pour tout secret $S$, il existe des valeurs des coefficients $A_1, \ldots, A_{T-1}$ qui produisent la même valeur de $P_i$.

▷ **Question 14 :** Concluez-en qu'un adversaire qui aurait une puissance de calcul énorme (qui serait par exemple capable d'effectuer $2^k$ opérations) ne pourrait pourtant pas retrouver $S$ à partir de $P_i$.

Le résultat ci-dessus s'étend aussi au cas où l'adversaire possède $T-1$ parts, mais ça on va l'admettre. On va maintenant voir comment le secret peut être reconstitué avec $T$ parts.

▷ **Question 15 :** Supposons que $T = 2$. Expliquez comment, avec deux parts $P_i$ et $P_j$ ($i \neq j$), on peut reconstituer le secret $S$.

▷ **Question 16 :** Expliquez comment le processus se généralise lorsque $T > 2$. Comment la complexité évolue-t-elle en fonction de $T$ ?

Ce système de partage de secret possède une propriété de morphisme. On partage un premier secret $S$ avec les coefficients $A_1, \ldots, A_{T-1}$, ce qui donne les parts $P_1, P_2, \ldots$. Ensuite, on partage un deuxième secret $S'$ avec d'autres coefficents $A'_1, \ldots, A'_{T-1}$, ce qui donne les parts $P'_1, P'_2, \ldots$.

▷ **Question 17 :** Montrez que les parts $(P_i + P'_i) \ \% \ p$ correspondent à un partage du secret $(S + S') \ \% \ p$. Quels seraient les coefficients correspondants ?

En cette période électorale, on va voir une application du partage de secret aux élections. Imaginons un référendum où les citoyens peuvent voter « 0 » ou « 1 ». Le résultat de l'élection (qu'on note $R$) est le nombre de votes « 1 ». Il y a $N$ bureaux de vote, numérotés de 1 à $N$, et chaque bureau connaît tous les électeurs. Pour voter, un électeur partage son vote (0 ou 1) en $N$ parts, avec un seuil de $N$, et envoie par un canal sécurisé une part à chaque bureau (la $i$-ème part au $i$-ème bureau).

▷ **Question 18 :** Imaginons que le résultat $R$ de l'élections soit un secret partagé entre les bureau de vote. Expliquez comment le $i$-ème bureau de vote peut calculer la $i$-ème part du partage de $R$ à partir des votes.

▷ **Question 19 :** Expliquez comment, une fois le scrutin terminé, les $N$ bureaux de vote peuvent obtenir le résultat de l'élection.

▷ **Question 20 :** Supposons qu'un des bureaux est honnête et refuse de publier les parts qu'il reçoit des électeurs. Les autres bureaux, qui sont tous malhonnêtes et qui peuvent collaborer entre eux, peuvent-ils apprendre pour qui un électeur a voté ?

▷ **Question 21 :** Ce protocole, tel qu'il est décrit, souffre de nombreux défauts. Parmis ceux-ci, un électeur mécontent, ou un bureau de vote malhonnête, peuvent gravement fausser l'issue de l'élection. Expliquez comment ils pourraient s'y prendre (il est possible d'y remédier... mais c'est assez compliqué !).