Université Paris 8 — Année 2022–23 — J. Lavauzelle

Master 1 Mathématiques et applications

Cryptographie à clé publique – Solutions feuille de TD 6

10/03/2023

Retrouvez le sujet du TD et d'autres exercices à l'adresse :

`www.math.univ-paris13.fr/~lavauzelle/teaching/2022-23/clef-publique.html`

($\star$) exercice fondamental ($\star\star$) pour s'entraîner ($\star\star\star$) pour aller plus loin 💻 sur machine

Exercice 1. ($\star\star$) **Une proposition de schéma de signature.**

Dans cet exercice, on considère un nombre premier p pour lequel le problème du logarithme discret dans $\mathbb{F}_p^\times$ est supposé difficile. On note g un générateur du groupe cyclique $\mathbb{F}_p^\times$. Enfin, on considère une fonction de hachage $H : \{0,1\}^* \to \mathbb{Z}/(p-1)\mathbb{Z}$.

Un schéma de signature est décrit par les trois algorithmes suivants.

Algorithme 1 : Génération de clefs

Entrée : les paramètres du système
Sortie : une paire de clefs publique/privée

1 Tirer x aléatoirement dans $\mathbb{Z}/(p-1)\mathbb{Z}$.
2 Tirer y aléatoirement dans $\mathbb{Z}/(p-1)\mathbb{Z}$.
3 Calculer $X = g^x \mod p$ et $Y = g^y \mod p$.
4 La clef publique est $\texttt{pk} = (X,Y)$, la clef privée est $\texttt{sk} = (x,y)$.

Algorithme 2 : Signature

Entrée : un message $\boldsymbol{m} \in \{0,1\}^*$, la clé privée $\texttt{sk} = (x,y)$
Sortie : une signature $s \in \mathbb{Z}/(p-1)\mathbb{Z}$

1 Calculer $h = H(\boldsymbol{m}) \in \mathbb{Z}/(p-1)\mathbb{Z}$
2 Calculer et retourner l'élément $s = xh + y \in \mathbb{Z}/(p-1)\mathbb{Z}$.

Algorithme 3 : Vérification

Entrée : une signature $s \in \mathbb{Z}/(p-1)\mathbb{Z}$, un message $\boldsymbol{m} \in \{0,1\}^*$, la clé publique $\texttt{pk} = (X,Y)$
Sortie : vrai ou faux

1 Calculer $h = H(\boldsymbol{m}) \in \mathbb{Z}/(p-1)\mathbb{Z}$.
2 Calculer $a = g^s \mod p$ et $b = X^h Y \mod p$.
3 Faire le test $a \equiv b \mod p$ et retourner le booléen associé.

Question 1.– Vérifier que le schéma de signature est valide.

Question 2.– Proposer une attaque sur la clé privée $\texttt{sk} = (x,y)$. On précisera le moyen d'attaque utilisé.

Solutions de l'Exercice 1.

Solution Q1. On a bien

$$a \equiv g^s \equiv g^{xh+y} \equiv (g^x)^h g^y \equiv X^h Y \mod p$$

Solution Q2. L'idée est de chercher un système d'équations linéaires satisfaites par les éléments (x, y) de la clé privée. Si $\boldsymbol{m}$ et $\boldsymbol{m}'$ sont deux messages, alors on a :

$$\begin{aligned} s &= x\,H(\boldsymbol{m}) + y \mod (p-1) \\ s' &= x\,H(\boldsymbol{m}') + y \mod (p-1) \end{aligned}$$

Puis, si $H(\boldsymbol{m}) - H(\boldsymbol{m}')$ est inversible modulo $p-1$, on obtient :

$$x = (s - s')(H(\boldsymbol{m}) - H(\boldsymbol{m}'))^{-1} \mod (p-1)$$

et

$$y = s - x\,H(\boldsymbol{m}) \mod (p-1)\,.$$

De ces calculs, on déduit qu'il est possible de monter une attaque sur la clé, avec des **messages connus** (KMA). En effet, quelques messages (2 avec bonne probabilité) suffisent pour obtenir $H(\boldsymbol{m}) - H(\boldsymbol{m}')$ inversible modulo $p-1$, et il n'est pas nécessaire de les choisir. On retrouve ensuite x et y avec les formules ci-dessus.

Exercice 2. ($\star\star\star$) Compression de clefs ECDSA.

On considère une paire de clefs publique/privée pour le schéma de signature ECDSA, instantié dans une courbe elliptique E sur $\mathbb{F}_p$. L'équation de la courbe est donnée comme un paramètre publique, ainsi qu'un générateur G du plus grand sous-groupe cyclique $\mathbb{G}$ de $E(\mathbb{F}_p)$. On note enfin n l'ordre de $\mathbb{G}$. On se place dans un cas favorable où $n \sim p$.

Question 1.– Rappeler une description de la clef publique et de la clef privée du système. Quelle est la taille mimimale de p pour obtenir une sécurité de 128 bits ? En déduire la taille minimale de la clef publique, lorsqu'on n'utilise aucune stratégie d'encodage particulière.

Question 2.– Donner une majoration la plus fine possible du nombre de bits nécessaires pour encoder un point fini de $E(\mathbb{F}_p)$.

Question 3.– Soit $P = (x_P, y_P)$ un point fini de $E(\mathbb{F}_p)$. Comment peut-on déduire la valeur de y_P à partir de celle de x_P, au signe près ? En déduire une description unique de P qui utilise au plus $\lceil \log_2 p \rceil + 1$ bits.

Solutions de l'Exercice 2.

Solution Q1. La clef privée est $a \in \{1, \ldots, n-1\}$ et la clé publique est $aG \in \mathbb{G}$. Sur une courbe elliptique, la résolution du problème du logarithme discret nécessite $O(\sqrt{n})$ opérations élémentaires de groupe, où n est l'ordre du groupe. Ici $n \sim p$, donc pour obtenir une sécurité de 128 bits, il faut prendre p de taille 256 bits.

Un point d'une courbe elliptique est constitué d'une abscisse et d'une ordonnée. Il faut donc 512 bits pour stocker aG.

Solution Q2. Comme il peut y avoir jusqu'à $p + 1 + 2\sqrt{p}$ points dans $E(\mathbb{F}_p)$, on a besoin d'au moins $\log_2(p + 1 + 2\sqrt{p}) \leq \lceil \log_2(p) \rceil + 1$ bits pour stocker un point de cette courbe elliptique.

Solution Q3. On suppose que l'équation de la courbe elliptique est $y^2 = x^3 + ax + b$. On peut déduire y_P^2 à partir de x_P en calculant simplement $x_P^3 + ax_P + b$. Puis, on extrait une racine carrée de la valeur

obtenue pour obtenir $\pm y_P$. Notons que le calcul d'une racine carrée est une opération efficace dans un corps fini $\mathbb{F}_p$ (algorithmes de Cipolla, de Tonelli-Shanks).

On peut donc décrire le point P avec $\lceil \log_2 p \rceil + 1$ bits : $\lceil \log_2 p \rceil$ bits pour décrire x_P, et un bit pour décrire le signe de y_P. C'est un encodage quasi-optimal au vu de la question précédente.

Exercice 3. ($\star\star\star$) Signature de cercle.

Une signature de cercle (*ring signature*) est une primitive cryptographique qui permet à chaque membre d'un « cercle » (= groupe d'utilisateur) de signe anonymement un message m au nom du cercle.

Voici quelques propriétés désirées dans une signature de cercle :

1. N'importe quel vérificateur extérieur doit pouvoir vérifier (et être convaincu) qu'une signature s d'un message m est émise au nom du cercle.
2. Par ailleurs, il doit être impossible de déterminer quel membre du cercle a émis la signature.
3. Enfin, le signataire ne doit pas avoir besoin de l'aide d'autres membres du groupe pour poubvoir signer un message.

Dans cet exercice, on présente une signature de cercle basée sur la signature RSA-FDH. La fonction de hachage utilisée est notée H, à valeur dans $\{0,1\}^t$. On suppose également que toutes les clefs publiques des membres du cercle sont authentifiées.

Pour commencer l'exercice, on considère un cercle composé de deux membres seulement : Alice et Bob. Leurs clefs publiques sont respectivement (n_A, e_A) et (n_B, e_B). On suppose que les modules n_A et n_B sont de taille $t+1$ bits, et on assimile tout élément de $\{0,1\}^t$ avec un entier inférieur à 2^t. La signature d'un message $\boldsymbol{m}$ par Alice est donc $H(\boldsymbol{m})^{d_A} \mod n_A$ où d_A est la clé privée associée à (n_A, e_A).

On note enfin $\oplus$ l'opération de `xor` (addition modulo 2) bit-à-bit pour des entiers vus comme des chaînes de bits de longueur t.

L'Algorithme 4 décrit les opérations à effectuer par Alice pour émettre une signature de cercle. Notons qu'une description similaire est possible pour Bob, en remplaçant simplement les données propres à Alice par celles propres à Bob, et réciproquement. L'Algorithme 5 décrit la vérification de la signature effectuée par une personne potentiellement extérieure au cercle.

Algorithme 4 : Algorithme de signature de cercle (opéré par Alice)

Entrée : un message $\boldsymbol{m}$, les clés publiques $(n_A, e_A), (n_B, e_B)$, la clé privée d_A d'Alice
Sortie : une signature de cercle s

1 Hacher le message $\boldsymbol{m}$ en $h = H(\boldsymbol{m})$.
2 Tirer aléatoirement $s_B \leftarrow (\mathbb{Z}/n_B\mathbb{Z})^\times$.
3 Calculer $z_B = s_B^{e_B} \mod n_B$.
4 Calculer $s_A = (z_B \oplus h)^{d_A} \mod n_A$.
5 Retourner $s = (s_A, s_B)$.

Algorithme 5 : Algorithme de vérification de signature de cercle

Entrée : un message $\boldsymbol{m}$, les clés publiques $(n_A, e_A), (n_B, e_B)$, une signature $s = (s_A, s_B)$
Sortie : accepte/refuse

1 Calculer $x = (s_A^{e_A} \mod n_A) \oplus (s_B^{e_B} \mod n_B)$.
2 Vérifier que $x = H(\boldsymbol{m})$.

Question 1.– Vérifier que l'Algorithme 5 est valide, c'est-à-dire qu'il accepte toute signature effectuée honnêtement par Alice (ou Bob).

Question 2.– Expliquer en quoi la signature est anonyme pour un signataire quelconque du cercle. C'est-à-dire, argumenter sur le fait que le vérificateur ne peut pas savoir qui, parmi Alice et Bob, a produit la signature $s = (s_A, s_B)$.

On dit qu'une fonction de hachage est résistante au calcul de préimage si, étant donné un haché h, il est impossible calculatoirement de trouver un message $\boldsymbol{m}$ tel que $H(\boldsymbol{m}) = h$.

Question 3.– Supposons que la fonction de hachage H ne soit pas résistante au calcul de préimage. Donner une attaque sur le schéma de signature de cercle. On précisera le type d'attaque et les moyens donnés à l'attaquant.

Question 4.– Dans un contexte où l'on souhaite avoir une sécurité à long terme, quelle serait la taille de la signature de cercle ?

Une méthode pour falsifier une signature de $\boldsymbol{m}$ consiste à créer deux tableaux de valeurs de la forme $v_A := u_A^{e_A} \mod n_A$ et $v_B := u_B^{e_B} \mod n_B$ (où les u_A et u_B sont tirées aléatoirement), et de chercher une « collision » entre les tableaux de la forme $v_A \oplus H(\boldsymbol{m}) = v_B$. Une fois cette collision trouvée, on émet la signature (u_A, u_B) associé au message $\boldsymbol{m}$.

Question 5.– Supposons ici que $t = 256$.

1. Préciser si l'attaque décrite ci-dessus induit une falsification existentielle ou universelle. Justifier clairement.
2. Quelle est la taille approximative des tableaux nécessaires pour avoir une falsification avec probabilité proche de 0.5 ?

Question 6.– Généraliser le schéma de signature de cercle à K utilisateurs au lieu de 2 (Alice et Bob). On présentera l'algorithme de signature de l'un de ces K utilisateurs, ainsi que l'algorithme de vérification à effectuer.

Solutions de l'Exercice 3.

Solution Q1. Notons $h = H(\boldsymbol{m})$ et calculons la valeur de x dans l'étape 1 de l'Algorithme de vérification. D'une part, on a :

$$s_A^{e_A} \equiv (z_B \oplus h)^{d_A e_A} \equiv z_B \oplus h \mod n_A$$

D'autre part, $s_B^{e_B} \mod n_B$ est défini comme égal à l'entier z_B. Par conséquent :

$$x = (s_A^{e_A} \mod n_A) \oplus (s_B^{e_B} \mod n_B) = z_B \oplus h \oplus z_B = h\,.$$

Solution Q2. Dans le cas où Alice a produit la signature, on a :

$$s = ((s_B^{e_B} \oplus h)^{d_A} \mod n_A\,,\, s_B), \quad \text{avec } s_B \text{ aléatoire.}$$

Dans le cas où c'est Bob, on obtient :

$$s' = (s_B\,,\, (s_A^{e_A} \oplus h)^{d_B} \mod n_B), \quad \text{avec } s_A \text{ aléatoire.}$$

Si le signataire savait différencier qui d'Alice ou Bob a produit la signature, alors il saurait différencier un élément de la forme $(s_B^{e_B} \oplus h)^{d_A} \mod n_A$ (où h est fixé mais s_B est aléatoire dans $(\mathbb{Z}/n_B\mathbb{Z})^\times$), d'un élément aléatoire $s_A \in (\mathbb{Z}/n_A\mathbb{Z})^\times$. C'est impossible car $(s_B^{e_B} \oplus h)^{d_A} \mod n_A$ est aléatoire dans $(\mathbb{Z}/n_A\mathbb{Z})^\times$ (l'application $y \mapsto (y^{e_B} \oplus h)^{d_A} \mod n_A$ étant une bijection).

Solution Q3. On procède très simplement comme suit :

1. tirer aléatoirement s_A dans $(\mathbb{Z}/n_A\mathbb{Z})^\times$ et s_B dans $(\mathbb{Z}/n_B\mathbb{Z})^\times$,
2. calculer $x = (s_A^{e_A} \mod n_A) \oplus (s_B^{e_B} \mod n_B)$,
3. calculer $\boldsymbol{m}$ tel que $H(\boldsymbol{m}) = x$ (car H n'est pas résistante aux préimages),

4. émettre la signature $s = (s_A, s_B)$ du message $\boldsymbol{m}$.

C'est une attaque par falsification existentielle à clé seule.

Solution Q4. Pour une sécurité à long terme on doit prendre n_A et n_B de taille ≥ 3072 bits, soit une signature de cercle de taille 6144 au moins.

Solution Q5.

1. C'est une falsification universelle : on peut fixer $\boldsymbol{m}$ avant de monter l'attaque.
2. D'après le paradoxe des anniversaires, il faut que la taille T des tableaux vérifie

$$T \geq 2^{t/2} = 2^{128}.$$

Solution Q6. Notons $(n_1, e_1), \dots (n_K, e_K)$ les clefs publiques des K membres du cercle, et supposons pour simplifier que le signataire soit le premier membre du cercle. Alors, pour signer un message $\boldsymbol{m}$ on définit l'algorithme suivant :

1. Tirer aléatoirement $s_2 \in (\mathbb{Z}/n_2\mathbb{Z})^\times, \dots, s_K \in (\mathbb{Z}/n_K\mathbb{Z})^\times$.
2. Calculer $z_i = s_i^{e_i} \mod n_i$ pour tout $i \in \{2, \dots, K\}$.
3. Calculer $s_1 = \left(z_2 \oplus \cdots \oplus z_K \oplus H(\boldsymbol{m})\right)^{d_1} \mod n_1$.
4. Retourner $s = (s_1, \dots, s_K)$.

Pour vérifier la signature de $\boldsymbol{m}$, on effectue alors le test :

$$(s_1^{e_1} \mod n_1) \oplus (s_2^{e_2} \mod n_2) \oplus \cdots \oplus (s_K^{e_K} \mod n_K) = H(\boldsymbol{m})$$

Exercice 4. 💻 Découverte d'un certificat.

Cet exercice « pratique » a pour but de vous faire découvrir les informations incluses dans un certificat.

Dans un navigateur de votre choix, entrer l'URL de l'université

`https://www.univ-paris8.fr/`

Chercher ensuite l'emplacement des certificats dans la barre d'adresse. Par exemple, sous `Mozilla Firefox`, on l'obtient en

1. cliquant d'abord sur le cadenas à gauche de l'adresse,
2. puis sur le chevron à droite de « connexion sécurisée »,
3. puis « plus d'information ».

Un bouton « Afficher le certificat » est alors disponible.

Question 1.– Combien de certificats trouve-t-on ? Pour chacun des certificats, préciser les émetteurs et sujets correspondants. Comment expliquer la présence de plusieurs certificats ? Que dire de celui qui a comme sujet *DigiCert Inc* ?

Question 2.– Pour chacun de ces certificats, quel algorithme de signature a été utilisé ?
Trouver également :

— la clef publique utilisée pour la signature
— la signature obtenue,
— la durée de validité,
— le contexte d'utilisation du certificat.

Solutions de l'Exercice 4.

Voir démonstration en cours.

Exercice 5. ($\star$) Un nouveau protocole d'identification ?.

Alice souhaite s'identifier auprès de Bob. On suppose qu'Alice et Bob détiennent un secret commun $x \in \{0,1\}^t$, pour $t \geq 1$, qui doit servir à plusieurs identifications. Le protocole suivant est proposé :

1. Bob choisit une chaîne aléatoire $r \in \{0,1\}^t$ et l'envoie à Alice
2. Alice calcule $y = r \oplus x$ et renvoie y à Bob.
3. Bob vérifie que $x = r \oplus y$.

Question 1.– Pourquoi ce protocole ne peut pas être utilisé pour plusieurs identifications ?

Question 2.– Quelle étape d'un protocole d'identification à trois passes manque t-il dans ce protocole ?

Solutions de l'Exercice 5.

Solution Q1. En observant les valeurs transmises entre Alice et Bob, un attaquant peut retrouver le secret x en effectuant le `xor` du défi r et de $y = r \oplus x$.

Solution Q2. Il manque l'étape d'engagement, dans laquelle Alice publie une valeur v liée à son secret, mais qui ne le révèle pas. Du fait de l'absence de cette étape, Alice est forcée de retourner une valeur y, qui couplée avec le défi r, révèle le secret r.
