Corrigé TD numéro 4

Introduction à la Cryptographie

Exercice 1: Falsification existentielle du schéma de signature ElGamal

Soit p un nombre premier, g un générateur de $\mathbb{Z}_p^\star$ et une clé publique $y = g^x \in \mathbb{Z}_p^\star$ où $x \bmod (p-1)$ est la clé secrète. L'espace des messages est $M_n = \mathbb{Z}_p^\star$.

La signature du message $m \in \mathbb{Z}_p^\star$ est le couple (r,t) tel que $0 \leq r, t < p-1$ et qui satisfait l'équation $g^m = y^r r^t \bmod p$.

L'algorithme de signature génère tout d'abord $k \in \mathbb{Z}_p^\star$ aléatoirement t.q. $\gcd(k, p-1) = 1$. Puis, il calcule $r = g^k \bmod p$ et $t = (m - xr)/k \bmod (p-1)$.

(a) Vérifier que la vérification fonctionne.

(b) Montrer une falsification existentielle sur ce schéma.

(c) Montrer que ce schéma peut être cassé si le même random k est utilisé pour calculer la signature de deux messages différents m_0 et m_1.

(d) Est-ce que ce schéma est sûr si on prend $k, k+1, k+2, \ldots$ où k est une valeur aléatoire pour signer des messages différents $m_1, m_2, \ldots$?

(e) Si k est généré par un générateur congruentiel modulo $(p-1)$.

Corrigé:

(a) La vérification fonctionne car on a tout fait pour.

$$y^r \cdot r^t = g^{xr} \cdot (g^k)^{(m-xr)/k} = g^{xr+(m-xr)} = g^m \bmod p!$$

(b) Si on pose $r = y$ et $t = -r$ est une signature valide pour le message $m = 0$.

(c) Si on utilise 2 fois le même random k, alors on a 2 équations avec 2 inconnues:

$$kt_1 + xr_1 = m_1 \bmod (p-1) \text{ et } kt_2 + xr_2 = m_2 \bmod (p-1)$$

qu'on peut facilement résoudre si le déterminant du système est inversible: c'est-à-dire si $\gcd(p-1, t_2 r_1 - r_2 t_1) = 1$.

(d) Si k est un compteur, alors on peut encore écrire des équations qui font disparaître la valeur de k en combinant 2 équations.

(e) Si k est généré par un générateur congruentiel $k_{i+1} = a \cdot k_i + b \bmod (p-1)$, alors si a est connu et b inconnu, c'est encore facile. Si a et b sont inconnus, il est possible de faire disparaître k et d'écrire des équations quadratiques qu'on peut résoudre par relinéarisation: de $kt_1 + xr_1 = m_1$ et $(ak+b)t_2 + xr_2 = m_2$, on peut calculer en multipliant la première par at_2 et la seconde par t_1,

$$-bt_1t_2 + axt_2r_1 - xt_1r_2 = m_2at_2 - m_2t_1$$

qui est quadratique en ax. On créé une nouvelle variable X qui vaut ax et on collecte suffisamment d'équation pour résoudre un système linéaire inversible. On a 4 inconnues et enfin on retrouve a, x, X et b avec 8 signatures. On vérifie si $X = ax$.

Exercice 2: ρ-Pollard pour le logarithme discret

Soit p et q premiers et $q|(p-1)$, γ un élément de $\mathbb{Z}_p^\star$ qui génère le sous-groupe G d'ordre q et $\alpha \in G$. Soit la fonction F qui envoie les éléments de G sur $\{0, \ldots, q-1\}$. On définit la fonction $H : G \to G$ qui envoie β sur $\beta\alpha\gamma^{F(\beta)}$.

1. $i := 1, x := 0, \beta := \alpha, x := F(\beta), \beta' := H(\beta)$

2. Tantque $\beta \neq \beta'$, faire
   $x := (x + F(\beta)) \bmod q$, $\beta := H(\beta)$
   $x' := (x' + F(\beta')) \bmod q$, $\beta' := H(\beta')$
   $x' := (x' + F(\beta')) \bmod q$, $\beta' := H(\beta')$
   i:=i+1

3. si $i < q$, alors return $(x - x')i^{-1} \bmod q$

4. sinon échec.

On définit $\beta_1, \beta_2, \ldots$ comme $\beta_1 = \alpha$, et pour $i > 1$, $\beta_i = H(\beta_{i-1})$.

(a) Montrer qu'à chaque boucle, on a: $\beta = \beta_i = \gamma^x \alpha^i$ et $\beta' = \beta_{2i} = \gamma^{x'} \alpha^{2i}$.

(b) Montrer que si la boucle se termine avec $i < q$, la valeur de sortie est $\log_\gamma(\alpha)$.

(c) Soit j le plus petit index t.q. $\beta_j = \beta_k$ pour un indice $k < j$. Montrer que $j \leq q + 1$ et que la boucle se termine avec $i < j$ (et en particulier $i \leq q$).

(d) Si la fonction F se comporte comme une fonction aléatoire, c'est-à-dire choisit au hasard parmi toutes les fonctions de G vers $\{0, \ldots, q-1\}$. Montrer que cela implique que H est une fonction aléatoire, *i.e.* elle est uniformément distribuée sur G vers G.

(e) Si F est une fonction aléatoire, alors le temps moyen de l'algorithme est $O(q^{1/2}\mathsf{len}(q)(\mathsf{len}(p))^2)$ et que la proba d'échec est $\leq 1/q$.

Corrigé:

(a) À la fin de la boucle, on a ces deux équations car dans la boucle Tantque, on fait deux fois le calcul pour x' et une fois pour x.

(b) Si la boucle se termine avec $i < q$, alors on a $\beta = \beta'$ et donc compte tenu des équations précédentes, on a: $\gamma^x \alpha^i = \gamma^{x'} \alpha^{2i}$ et donc:

$$\frac{x - x'}{i} \bmod q$$

est le logarithme recherché de β en base α. Pour cela, il faut que i soit inversible modulo q et donc que $i < q$.

(c) Pour montrer que $j \leq q + 1$, on utilise le lemme des tiroirs. Si $j > q + 1$, β_i est une suite qui prend deux fois la même valeur et donc la boucle se termine avant. Pour montrer que $i < j$, on utilise le fait que les deux suites avancent et l'une avance deux fois plus vite. Avec un petit dessin, on s'apreçoit que la suite la plus rapide ne peut pas sauter au-dessus de la suite la plus lente sans que les deux suites prennent la même valeur.

(d) Il est facile de voir que si F est une fonction aléatoire, alors H est une fonction aléatoire. En effet, la fonction $\beta \mapsto \gamma^{F(\beta)}$ est aléatoire si F est aléatoire. Si F est aléatoire, on peut supposer que les valeurs de F sont indépendantes des entrées de F et donc qu'il y a indépendance entre les deux fonctions: $\beta \mapsto \beta\alpha$ et $\beta \mapsto \gamma^{F(\beta)}$. Comme la deuxième fonction donne une valeur aléatoire, le produit d'une valeur aléatoire par une valeur indépendante est une valeur aléatoire. Ainsi, la fonction H est une fonction aléatoire dès que F est aléatoire.

(e) Si la fonction F est aléatoire, alors on peut utiliser le paradoxe des anniversaires et on va obtenir une collision en temps $q^{1/2}$. De plus, à chaque étape on calcule une exponentiation modulaire dont le coût est $O(\mathsf{len}(q)\mathsf{len}(p)^2)$ et pour que l'algorithme fonctionne, il faut que $i < q$, ce qui se produit avec probabilité au moins égale à $1 - 1/q$.

Exercice 3: Problèmes équivalents à Diffie-Hellman

On dit qu'un problème A *se réduit de manière déterministe en temps polynomial* à un problème B s'il existe un algorithme déterministe R pour résoudre le problème A qui fait des appels à un sous-programme pour le problème B, où le temps d'exécution de R (ne comprenant pas le temps de calcul du sous-programme B) est polynomial en la taille de l'entrée.

On dit que A et B sont *équivalent de manière déterministe en temps polynomial* si A est réductible de manière déterministe en temps polynomial au problème B et si B est réductible de manière déterministe en temps polynomial au problème A.

Montrer que les problèmes suivants sont équivalents de manière déterministe en temps polynomial:

1. Étant donné un premier p, un premier $q|(p-1)$, et un élément $\gamma \in \mathbb{Z}_p^\star$ d'ordre q, et deux éléments $\alpha, \beta \in \langle\gamma\rangle$, calculer γ^{xy}, où $x = \log_\gamma \alpha$ et $y = \log_\gamma \beta$. Ce problème est appelé le *problème Diffie-Hellman*.

2. Étant donné un premier p, un premier $q|(p-1)$, et un élément $\gamma \in \mathbb{Z}_p^\star$ d'ordre q, et un élément $\alpha \in \langle\gamma\rangle$, calculer γ^{x^2}, où $x = \log_\gamma \alpha$.

3. Étant donné un premier p, un premier $q|(p-1)$, et un élément $\gamma \in \mathbb{Z}_p^\star$ d'ordre q, et deux éléments $\alpha, \beta \in \langle\gamma\rangle$, avec $\beta \neq 1 \bmod p$, calculer $\gamma^{xy'}$, où $x = \log_\gamma \alpha$ et y' est l'inverse multiplicatif modulo q de $y = \log_\gamma \beta$.

4. Étant donné un premier p, un premier $q|(p-1)$, et un élément $\gamma \in \mathbb{Z}_p^\star$ d'ordre q, et un élément $\alpha \in \langle\gamma\rangle$, avec $\alpha \neq 1 \bmod p$, calculer $\gamma^{x'}$, où x' est l'inverse multiplicatif modulo q de $x = \log_\gamma \alpha$.

Corrigé:

Pour montrer l'équivalence entre les deux premiers problèmes: il est clair que si je sais résoudre le premier problème, je sais résoudre le second car il suffit d'appeler un algorithme qui résout le premier sur les entrées $\alpha = \gamma^x$ et $\beta = \gamma^x$, pour qu'il retourne γ^{x^2}. Pour l'autre sens, il suffit d'utiliser l'égalité suivante: $xy = \frac{(x+y)^2 - x^2 - y^2}{2}$ et de demander γ^{x+y}, γ^x et γ^y à un algorithme qui résout le second problème. Enfin, il faut bien s'assurer qu'on puisse diviser par 2, ce qui est le cas car l'ordre du sous-groupe engendré par γ est q qui est premier.

Si on sait résoudre, 1 et 4, alors on peut facilement résoudre 3. Réciproquement, si on sait résoudre 4 et 3, on peut résoudre 1. Montrons par exemple que 4 est équivalent à 2 et ce sera terminé. Pour ce faire, l'idée est de remarquer que pour calculer l'inverse de x modulo q, on peut utiliser la formule suivante: $1/x = x^{q-2} \bmod q$ car d'après le petit théorème de Fermat: $x^{q-1} = 1 \bmod q$. Ainsi, il suffit de calculer $\gamma^{x^{q-2}}$. Pour

ce faire, il suffit de calculer l'algorithme square-and-multiply à partir de γ^x, on peut calculer γ^{x^2} avec un algorithme qui résoud le problème 2. On peut décomposer alors l'exposant $(q-2)$ en binaire et effectuer les multiplications et les élévations au carré nécessaires.

Exercice 4: Signature GHR (Gennaro-Halevi-Rabin)

Soit $N = pq$ un module RSA où p et q sont deux nombres premiers de k bits. La clé publique est consitutée du module N et d'une valeur aléatoire $y \in \mathbb{Z}_N^*$. La clé secrète est la factorisation de N ou de manière équivalente $\varphi(N)$.

Pour signer un message m, on calcule $e = H_k(m)$ et $d = e^{-1} \bmod \varphi(N)$. La signature est σ tel que $\sigma = y^d \bmod N$.

La vérification consiste à calculer

$$\sigma^{H_k(m)} \stackrel{?}{=} y \bmod N$$

(a) Montrer le lemme suivant: étant donné $x, y \in \mathbb{Z}_N^*$ et $a, b \in \mathbb{Z}$ tels que $x^a = y^b \bmod N$ et $\gcd(a,b) = 1$, alors on peut efficacement calculer $\tilde{x} \in \mathbb{Z}_N^*$ tel que $\tilde{x}^a = y \bmod N$.

Le strong-RSA problème est le suivant: étant donné un module N et un élément y, trouver $e > 1$ et x tels que $x^e = y \bmod N$.

(b) Montrer que ce schéma de signature est sûr dans le modèle de l'oracle aléatoire si la fonction $H_k(\cdot)$ est sans collision et que ses sorties sont des nombres premiers.

Corrigé:

(a) Comme $\gcd(a,b) = 1$, il existe deux entiers u et v tels que $au + bv = 1$. En élevant à la puissance v l'équation, $x^a = y^b$, on obtient:

$$x^{va} = y^{vb} = y^{1-ua} \bmod N,$$

et donc

$$x^{va} \cdot y^{ua} = y = (x^v \cdot y^u)^a \bmod N.$$

On a donc $\tilde{x} = (x^v \cdot y^u) \bmod N$.

(b) Supposons qu'il existe un adversaire contre le schéma de signature qui arrive à forger un message dont il n'aurait pas vu auparavant une signature et montrons comment à partir de cet adversaire on peut construire un algorithme qui résout le strong-RSA problème. En entrée d'un algorithme qui résout le strong-RSA problème, on a une entrée y et N un module RSA. on recherche x et $e > 1$ tels que $y = x^e \bmod N$. On doit construire un algorithme qui va utiliser notre adversaire. On va donc lui faire croire qu'il est en face d'un challengeur et s'il nous retourne une forgerie, alors on doit pouvoir l'utiliser pour retrouver x et e. Enfin, comme on est dans le modèle de l'oracle aléatoire, toute requête de hachage doit passer par l'algorithme qui doit résoudre le strong-RSA problème. Il faut donc montrer comment cet algorithme répond à une requête de hachage et à une requête de signature. Le principe est le même que dans la signature RSA-FDH. Initialement, je calcule plein de nombre premier e_i et je fais le pari que l'adversarire va répondre correctement pour la réponse de hachage e_k. Je calcule alors le produit: $E = \prod_{i \neq k} e_i$. Je lui envoie la clé publique $(y' = y^E \bmod N, N)$.

Je réponds à une requête de hachage m_i en renvoyant e_i et quand on me pose une requête de signature pour le message m_i, je renvoie $\sigma = y^{\prod_{j \neq i} e_j} \bmod N$. Il est aisé de voir alors que $\sigma^{H(m_i)} = \sigma^{e_i} = y^E = y' \bmod N$. À la fin, il va retourner $\tilde{\sigma}$ tel que $y^E = \tilde{\sigma}^{e_k}$. J'utilise alors le résultat de la question précédente comme $\gcd(E, e_k) = 1$ pour calculer σ tel que $\sigma^{e_k} = y \bmod N$: en fait $\sigma = \tilde{\sigma}^u \times y'^v$ où u et v sont tels que $uE + ve_k = 1$.