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

Master 1 Mathématiques et applications

Cryptographie à clé publique – Solutions feuille de TD 4

17/02/2023

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. ($\star$) ElGamal : application directe.

En guise d'exercice d'application, on considère le cryptosystème d'ElGamal « brut » dans le groupe multiplicatif $\mathbb{F}_p^\times$ où $p = 19$. On prend comme générateur $g = 2$.

Alice produit la clé privée $a = 5$.

Question 1.– Quelle est la clé publique ?

Question 2.– Chiffrer le message $m = 10$ avec l'aléa $k = 7$.

Question 3.– Déchiffrer $c = (12, 7)$ avec la clé privée d'Alice.

Solutions de l'Exercice 1.

Solution Q1. La clé publique est $\alpha = g^a = 2^5 \equiv 13 \mod p$.

Solution Q2. Pour chiffrer le message $m = 10$ avec la valeur aléatoire $k = 7$, on calcule $b_1 = g^k$ et $b_2 = m\alpha^k$. Concrètement, on obtient :

$$b_1 = 2^7 \equiv 14 \mod p \qquad \text{et} \qquad b_2 = 10 \times 2^{13} \equiv 5 \mod p$$

Solution Q3. On obtient $m' = 7 \times 12^{-5} \equiv 8 \mod p$.

Exercice 2. ($\star$) Attaque sur l'homomorphisme du chiffrement d'ElGamal.

Question 1.– Démontrer que le chiffrement d'ElGamal dans sa version « brute », présenté dans un groupe $(\mathbb{G}, \cdot)$, est homomorphe. Autrement dit, démontrez que si m et m' sont deux clairs de chiffrés $c = (c_1, c_2)$ et $c' = (c'_1, c'_2)$, alors un chiffré possible de $m \cdot m'$ est $(c_1 \cdot c'_1, c_2 \cdot c'_2)$.

Application. Bob souhaite acheter une maison à Clara. Pour cela, il doit transmettre sa promesse d'achat à Alice, une notaire. Sur cette promesse d'achat, on suppose qu'il inscrit uniquement la somme qu'il souhaite payer à Clara.

Alice, la notaire, souhaite utilise le chiffrement ElGamal « brut » dans le groupe multiplicatif $\mathbb{F}_p^\times$, afin de sécuriser la valeur entière (en euros) que Bob souhaite inscrire sur sa promesse d'achat.

Précisons que la valeur du nombre premier p a été choisie suffisamment grande par Alice, pour que le logarithme discret dans $\mathbb{F}_p^\times$ soit irrésoluble.

Question 2.– Supposons que Clara arrive à intercepter le message chiffré de Bob. Comment peut-elle modifier ce chiffré pour faire croire à Alice que Bob souhaite payer 2 fois plus que la somme initialement prévue ?

Question 3.– Que proposeriez-vous à la notaire pour empêcher cela ?

Solutions de l'Exercice 2.

Solution Q1. Soit n l'ordre de $\mathbb{G}$, et $\alpha = g^a$ la clef publique. Le chiffré c a été construit de la sorte : une valeur $k \in \mathbb{Z}/n\mathbb{Z}$ a été tirée aléatoirement, puis on a défini :

$$c = (c_1, c_2) = (g^k, m\alpha^k)$$

De même, $c' = (c'_1, c'_2) = (g^{k'}, m'\alpha^{k'})$ avec un certain $k' \in \mathbb{Z}/n\mathbb{Z}$ tiré aléatoirement et indépendamment de k. On a donc :

$$c_1 c'_1 = g^k g^{k'} = g^{k+k'}$$

d'une part, et d'autre part :

$$c_2 c'_2 = m\alpha^k m'\alpha^{k'} = (mm')\alpha^{k+k'}$$

Le couple $(c_1 c'_1, c_2 c'_2)$ correspond donc à un chiffré de mm' avec l'aléa $k + k'$.

Solution Q2. Clara intercepte $c = (c_1, c_2) = (g^k, m\alpha^k)$, où m est la somme en euros. pour faire croire que Bob souhaite payer deux fois plus, il lui suffit de transformer (c_1, c_2) en $(c_1, 2c_2)$.

Solution Q3. La notaire peut utiliser OAEP pour empêcher cette attaque.

Exercice 3. ($\star\star$) Chiffrement ElGamal avec aléa non-parfait.

Dans cet exercice, on s'intéresse au système de chiffrement ElGamal dans un groupe cyclique $\mathbb{G}$ d'ordre q. On note g un générateur de $\mathbb{G}$, et on rappelle qu'une paire de clefs consiste en un entier $a \in \{1, \dots, q-1\}$ (la clef privée) et l'élement de groupe $g^a \in \mathbb{G}$ (la clé publique).

On rappelle ci-dessous les algorithmes de chiffrement et déchiffrement d'ElGamal.

Algorithme 1 : Algorithme de chiffrement

Entrée : un message $m \in \mathbb{G}$, la clé publique $\alpha = g^a \in \mathbb{G}$
Sortie : un chiffré $c = (c_1, c_2) \in \mathbb{G}^2$

1 Choisir aléatoirement $r \in \mathbb{Z}/q\mathbb{Z}$.
2 Calculer $c_1 = g^r$.
3 Calculer $c_2 = m\alpha^r$.
4 Retourner $c = (c_1, c_2)$.

Algorithme 2 : Algorithme de déchiffrement

Entrée : un chiffré $c = (c_1, c_2) \in \mathbb{G}^2$, la clé privée $a \in \mathbb{Z}/q\mathbb{Z}$
Sortie : un message $m' \in \mathbb{G}$

1 Calculer $x = c_1^a$.
2 Calculer $m' = c_2/x$.
3 Retourner m'.

Question 1.– Dans cette question, on suppose que :

— le groupe $\mathbb{G}$ est un groupe multiplicatif $\mathbb{F}_p^\times$

— à l'étape 1 de l'algorithme de chiffrement, Bob choisit r uniformément dans $\{0, \ldots, 2^{32} - 1\}$

Que dire de la sécurité du système dans ce contexte ? Donner une réponse quantifiée en nombre de bits, et justifier.

On suppose maintenant, et dans toute la suite de l'exercice, que lors de l'étape 1 de l'algorithme de chiffrement, Bob utilise un générateur d'aléa de mauvaise qualité, défini ainsi :

— la première valeur aléatoire r_0 est engendrée par un tirage uniforme dans $\mathbb{Z}/q\mathbb{Z}$

— les valeurs aléatoires notées $r_1, r_2, \ldots, r_i, \ldots$ qui sont ensuite engendrées par le générateur, satisfont :

$$r_{i+1} = ur_i + v \mod q$$

où u et v sont des éléments fixes de $\mathbb{Z}/q\mathbb{Z}$.

Question 2.– Supposons qu'un attaquant connaisse u et v. Présenter une attaque contre le système permettant de déchiffrer un chiffré $\boldsymbol{c} = (c_1, c_2)$. On indiquera **précisément** le mode d'attaque.

Question 3.– On suppose maintenant que les valeurs de u et v sont inconnues de l'attaquant, mais restent inférieures à une constante K. Peut-on adapter l'attaque précédente ? Si oui, donner la complexité de l'attaque en fonction de K.

Solutions de l'Exercice 3.

Solution Q1. Première réponse (correcte mais incomplète). La sécurité est d'au plus 32 bits. Étant donné un chiffré (c_1, c_2), on peut simplement faire une recherce exhaustive sur r : on calcule tous les g^i et on teste l'égalité avec c_1. Puis lorsqu'on a obtenu la bonne valeur $i = r$, on calcule $c_2/\alpha^r = m$.

Seconde réponse. Ici, on peut même accéléler la recherche de r par une méthode de type *baby-step giant step*. On calcule g^i pour i allant de 0 à $2^{16} - 1$. Puis, on calcule $c_1 g^{-2^{16}j}$, avec j croissant de 0 à $2^{16} - 1$, jusqu'à obtenir l'un des g^i précalculés. Cette méthode a un coût algorithmique maximal de 2^{16} « opérations » lorsque le test d'appartenance à une liste est efficace (cf. cours). La sécurité est donc d'au plus d'au plus 16 bits.

Solution Q2. Voici une attaque à **clair connu**. L'attaquant obtient d'abord la paire clair/chiffré $(m, (c_1, c_2))$. On a donc

$$c_1 = g^{r_0} \quad \text{et} \quad c_2 = mg^{ar_0}$$

Puis, l'attaquant demande à attaquer le prochain message chiffré. On lui fournit donc (c_1', c_2') avec

$$c_1' = g^{r_1} = g^{ur_0+v} \quad \text{et} \quad c_2' = m'g^{ar_1} = m'g^{a(ur_0+v)}$$

Montrons que l'attaquant peut retrouver m'. Pour cela, on note que

$$c_2' = m'(g^{ar_0})^u (g^a)^v = m' c_2^u \alpha^v$$

Les valeurs de c_2', c_2, u, v et α sont connues de l'attaquant. Par conséquent, il peut retrouver m' par deux exponentiations que quelques opérations de groupe supplémentaires.

Solution Q3. La recherche de u et v de manière exhaustive ajoute un facteur en $O(K^2)$ à la complexité de l'attaque précédente.

Exercice 4. ($\star\star$) Une variante du chiffrement ElGamal.

Dans cet exercice, on se place dans le corps $\mathbb{F}_p$, avec p premier, et on considère g un générateur de $\mathbb{F}_p^\times$.

On s'intéresse à une variante du chiffrement ElGamal. La clé privée est toujous un élément aléatoire $a \in \mathbb{Z}/(p-1)\mathbb{Z}$, et la clé publique est toujours $\alpha = g^a$. En revanche, l'espace des clairs du système est $\mathbb{F}_p$, et celui des chiffrés est $\mathbb{F}_p \times \mathbb{F}_p^\times$. Enfin, l'algorithme de chiffrement est le suivant.

Algorithme 3 : Algorithme de chiffrement

Entrée : un message $m \in \mathbb{F}_p$, une clé publique α
Sortie : un chiffré $c = (c_1, c_2) \in \mathbb{F}_p^\times \times \mathbb{F}_p$

1 Choisir aléatoirement $r \in \mathbb{Z}/(p-1)\mathbb{Z}$.
2 Calculer $c_1 = g^r \mod p$.
3 Calculer $c_2 = \alpha^r + m$.
4 Retourner $c = (c_1, c_2)$.

Question 1.– Décrire précisément l'algorithme de déchiffrement associé (entrées, sortie, étapes), ainsi que sa complexité en fonction de p.

Question 2.– Supposons que Bob réutilise le même aléa à chaque chiffrement. Présenter une attaque contre le système en indiquant le mode d'attaque utilisé (c'est-à-dire, les moyens de l'attaquant).

Question 3.– Pourriez-vous instancier ce cryptosystème dans le groupe de points d'une courbe elliptique (au lieu de $\mathbb{F}_p$) ? Justifier : si oui, préciser les changements à effectuer ; si non, donner les obstacles.

Solutions de l'Exercice 4.

Solution Q1.

Algorithme 4 : Algorithme de déchiffrement

Sortie : un chiffré $c = (c_1, c_2) \in \mathbb{F}_p \times \mathbb{F}_p^\times$, , une clé privée a
Entrée : un message $m \in \mathbb{F}_p$

1 Calculer $m' = c_2 - (c_1)^a \mod p$.
2 Retourner m'.

Sa complexité est $O(\log(p))$ opérations (multiplications, carrés ou additions) dans $\mathbb{F}_p$.

Solution Q2. On peut monter une attaque à clair connu, par exemple. Soit $(c_1 = g^r, c_2 = \alpha^r + m)$ à déchiffrer. On demande un échantillon clair/chiffré quelconque, disons $(c_1' = g^r, c_2' = \alpha^r + m')$ où m' est connu de l'attaquant. Il suffit alors de calculer $m = c_1 - c_1' + m'$.

Solution Q3. Il faut voir que, dans le système décrit ci-dessus, il faut savoir additionner et multiplier dans la structure algébrique proposée. C'est possible dans $\mathbb{F}_p$, mais impossible (directement) dans $E(\mathbb{F}_p)$.

On peut donc répondre « non » avec la justification ci-dessus.

Une réponse « oui » était également possible : l'idée est de considérer une fonctionde de hachage $H : E(\mathbb{F}_p) \to \{0,1\}^t$, en supposant que le message est de longueur t (quitte à le décomposer). Puis, étant donnée une paire de clefs $(\mathtt{sk} = a, \mathtt{pk} = A = aG)$, le chiffré devient $(C_1, \boldsymbol{c}_2) = (rG, H(rA) \oplus \boldsymbol{m})$, et le déchiffrement consiste à calculer $H(aC_1) \oplus \boldsymbol{c}_2$.

Exercice 5. 💻 (⋆) Implantation de *baby-step giant-step*.

Question 1.– Implanter l'algorithme de calcul de logarithme discret dit « pas de bébé – pas de géant », dans le groupe multiplicatif d'un corps fini $\mathbb{F}_p^\times$.

Question 2.– Trouver les logarithmes discrets de $y \in \mathbb{F}_p^\times$ en base g pour les valeurs de p, g et y suivantes :

p	g	y
101	2	78
10007	5	8804
1000003	2	832469
100000007	5	29220559
10000000019	2	9521998688
1000000000039	3	855427796771

Jusqu'à quelle valeur de p le temps de calcul du logarithme discret par l'algorithme « pas de bébé – pas de géant » reste-t-il raisonnable sur votre machine ?

Et pour la recherche exhaustive ?

Solutions de l'Exercice 5.

Solution Q1. Voir fichiers annexes.

Solution Q2. On trouve les valeurs suivantes :

p	g	y	ℓ
101	2	78	36
10007	5	8804	131
1000003	2	832469	54692
100000007	5	29220559	44892073
10000000019	2	9521998688	2816033294
1000000000039	3	855427796771	303725427205

Expérimentalement, le temps de calcul prend plus de 10 secondes pour

— $p > 2^{27}$ avec la recherche exhaustive,
— $p > 2^{47}$ avec l'algorithme *baby-step giant-step*.

Bien entendu, ces valeurs dépendent de la machine, du langage de programmation choisi, de l'optimisation du code, etc.