

Calcul sécurisé – Feuille d'exercices numéro 4

Université Paris-Saclay – M1 Informatique/MINT

14 mars 2018

Exercice 1

Considérons l'algorithme de chiffrement ElGamal, défini de la façon suivante :

- p est un nombre premier tel que le problème du logarithme discret dans $\mathbb{Z}/p\mathbb{Z}$ soit difficile, α est un générateur de $(\mathbb{Z}/p\mathbb{Z})^*$.
- La clé secrète est un entier a tel que $2 \leq a \leq p - 2$.
- La clé publique est constituée de p , α et $\beta = \alpha^a \bmod p$.
- Pour chiffrer un message $x \in (\mathbb{Z}/p\mathbb{Z})^*$, on tire un entier $k \in \mathbb{Z}/(p - 1)\mathbb{Z}$ aléatoire. Le chiffré est alors (y_1, y_2) , où

$$\begin{cases} y_1 = \alpha^k \bmod p \\ y_2 = x\beta^k \bmod p \end{cases}$$

Expliciter la propriété d'homomorphie du schéma de chiffrement ElGamal

Exercice 2

On considère un schéma de chiffrement complètement homomorphe

$$Gen \rightarrow (sk, pk), \quad Enc_{pk}(x) \rightarrow y, \quad Dec_{sk}(x) = y, \quad Eval_{pk}(f, y) \rightarrow z$$

où $x \in \mathbb{Z}_2^n$ et f est une fonction de \mathbb{Z}_2^n dans \mathbb{Z}_2^n (définie par un circuit avec des portes logiques AND, OR et NOT).

On suppose qu'on a toujours

$$Dec_{sk}(Eval_{pk}(f, Enc_{pk}(x))) = f(x)$$

Si Dec_{sk} est une bijection, montrer que le schéma de chiffrement n'est pas sûr : il est facile de déchiffrer en ayant uniquement la clé publique.

Exercice 3

On considère un schéma FHE qui chiffre de façon probabiliste chaque bit en un chiffré de n bits. Pour simplifier, on suppose que les chiffrés ont une distribution de probabilité uniforme dans l'espace de chaînes de n bits. On veut stocker 1 Terabit de données.

1. Approximativement, à partir de quelle valeur de n peut-on dire que la probabilité d'avoir deux bits ayant le même chiffré devient-elle négligeable ?
2. Quelle est la taille des données une fois chiffrées ?

Exercice 4

Soit $f : (\mathbb{Z}/2\mathbb{Z})^n \rightarrow \mathbb{Z}/2\mathbb{Z}$ une fonction.

1. Montrer que cette fonction peut s'écrire comme un polynôme en n variables de degré total au plus n .
2. Soit E un schéma de chiffrement homomorphe (qui chiffre des bits). On suppose que le chiffré d'un message clair contient un "bruit" de taille λ , et que l'évaluation homomorphique de la multiplication (resp. l'addition) multiplie (resp. additionne) les bruits. Donner une borne supérieure sur la taille de la sortie de $Evaluate(f, c_1, \dots, c_n)$ où les c_i sont des chiffrés ayant un bruit λ .

Exercice 5

1. Trouver le PGCD approché, à trois chiffres, des nombres 195051, 257797, 328385 et 360776.

Indication : tester tous les "petits" bruits possibles sur une paire de ces nombres jusqu'à obtenir un PGCD suffisamment grand, puis vérifier avec les autres nombres.

2. Trouver le PGCD approché p , à quatre chiffres, des nombres 26978617, 23646450, 33970508 et 69181912, où le premier nombre est un multiple exact de p .

Remarque : factoriser directement 26978617 est considéré comme de la triche.

Exercice 6

On considère un schéma de chiffrement complètement homomorphe

$$Gen \rightarrow (sk, pk), \quad Enc_{pk}(x) \rightarrow y, \quad Dec_{sk}(x) = y, \quad Eval_{pk}(f, y) \rightarrow z$$

On suppose qu'on a toujours

$$Dec_{sk}(Eval_{pk}(f, Enc_{pk}(x))) = f(x)$$

On considère le protocole suivant entre $A(sk)$ et $B(pk)$, où A connaît un secret sk et où les deux participants connaissent pk , une fonction f , et le chiffré $y = Enc_{pk}(x)$ d'une valeur inconnue x :

1. **for** $i = 1$ to n **do**
2. B chooses randomly a bit b_i
3. **if** $b_i = 0$ **then** B takes $z_i = Enc_{pk}(0)$ **else** B takes $z_i = Eval_{pk}(f, y)$
4. B sends z_i to A
5. A computes $t_i = Dec_{sk}(z_i)$
6. **if** $t_i = 0$ **then** A sets $b'_i = 0$ **else** A sets $b'_i = 1$
7. A sends $Commit(b'_i)$ to B
8. B reveals b_i and how he computed z_i
9. A checks that z_i was well computed
10. A opens his commitment
11. B checks that the commitment is correct and gets b'_i
12. B checks that $b_i = b'_i$
13. **end for**

Le protocole s'interrompt dès qu'une des vérifications échoue.

1. En supposant que le protocole est exécuté correctement et que n est suffisamment grand, montrer que le protocole réussit si et seulement si $f(x) \neq 0$.
2. Montrer que si b_i est toujours choisi égal à 1, alors A peut tricher et faire réussir le protocole même si $f(x) = 0$.
3. Montrer que si l'étape 9 est omise dans le protocole, alors B peut tricher et apprendre la valeur de x .