Master CSI 2 2018-2019

Cryptologie Avancée — 4TCY903U

Responsables : G. Castagnos – G. Zémor

Examen — 17 décembre 2018

Durée 3h — Documents non autorisés

Partie G. Castagnos

Exercice 1. On rappelle le fonctionnement du chiffrement Elgamal :

— Soit GenDH un algorithme polynomial qui prend en entrée 1^k et retourne (G, n, g) : la description d'un groupe cyclique G, son ordre n de k bits et un générateur g.

— L'algorithme KeyGen appelle GenDH puis choisit x aléatoire avec probabilité uniforme dans $\mathbf{Z}/n\mathbf{Z}$ et calcule $h = g^x$. KeyGen retourne $pk = (G, n, g, h)$ et $sk = x$.

— L'algorithme Encrypt sur l'entrée (pk, m) avec $m \in G$ choisit r uniformément dans $\mathbf{Z}/n\mathbf{Z}$ et retourne $c = (g^r, mh^r)$

(a) Rappeler la définition précise de l'hypothèse garantissant la sécurité IND – CPA du chiffrement Elgamal.

Dans les quatre questions suivantes **(b-e)**, on suppose que la sortie de GenDH est de la forme (G, n, g) avec $G = ((\mathbf{Z}/p\mathbf{Z})^\times, \times)$ où p est premier et $n = p - 1$ (donc non premier), et g un générateur de $(\mathbf{Z}/p\mathbf{Z})^\times$. On rappelle que le symbole de Legendre $\left(\frac{x}{p}\right)$ d'un élément x de $(\mathbf{Z}/p\mathbf{Z})^\times$ vaut 1 si x est un carré modulo p et -1 si ce n'est pas un carré.

(b) Que vaut $\left(\frac{g}{p}\right)$?

(c) Soient x, y deux éléments de $\mathbf{Z}/(p-1)\mathbf{Z}$, et $X := g^x$ et $Y := g^y$. Montrer comment à partir de X et Y on peut calculer le symbole $\left(\frac{g^{xy}}{p}\right)$.

(d) En déduire que l'hypothèse énoncée en (a) est fausse pour $G = (\mathbf{Z}/p\mathbf{Z})^\times$. Pour cela décrire un algorithme $\mathcal{D}$ attaquant le problème sous-jacent et montrer que son avantage est non négligeable.

(e) Détailler comment l'algorithme $\mathcal{D}$ construit précédemment peut donner une attaque sur Elgamal. Comment peut on s'en protéger ?

Pour toute la suite de l'exercice, on suppose maintenant que la sortie de GenDH est de la forme (G, n, g) avec G un sous groupe d'ordre $n := q$ premier impair de $((\mathbf{Z}/p\mathbf{Z})^\times, \times)$ où p est premier et g un générateur de G. De plus on considère $\mathcal{H}$ un oracle aléatoire pouvant prendre en entrée des éléments de tout groupe cyclique G retourné par GenDH et retournant des valeurs de $\{0,1\}^k$. On définit une variante d'Elgamal dans le modèle de l'oracle aléatoire en modifiant l'algorithme de chiffrement comme suit :

— L'algorithme Encrypt sur l'entrée (pk, m) avec $m \in \{0,1\}^k$ choisit r uniformément dans $\mathbf{Z}/q\mathbf{Z}$ et retourne $c = (g^r, m \oplus \mathcal{H}(h^r))$

(f) Donner l'algorithme de déchiffrement Decrypt correspondant.

(g) Quel est l'intérêt de cette variante comparé au chiffrement d'Elgamal classique ?

On définit l'expérience que joue un algorithme $\mathcal{B}$ contre un problème nommé List − CDH :

$\mathbf{Exp}^{\mathsf{List-CDH}}_{\mathsf{GenDH},k}(\mathcal{B})$:

1. Lancer GenDH avec entrée 1^k pour obtenir G, q, g
2. Choisir $x, y \xleftarrow{\$} (\mathbf{Z}/q\mathbf{Z})$, et calculer $X = g^x$ et $Y = g^y$
3. $\mathcal{B}$ prend $G, q, g, (X, Y)$ en entrée et renvoie L un ensemble d'éléments de G
4. La sortie de l'expérience est 1 si $g^{xy} \in L$ et 0 sinon

Le succès de $\mathcal{B}$ pour résoudre List − CDH est $\Pr\left[\mathbf{Exp}^{\mathsf{List-CDH}}_{\mathsf{GenDH},k}(\mathcal{B}) = 1\right]$.

(h) Soit $\mathcal{A}$ un algorithme polynomial probabiliste contre le problème calculatoire Diffie-Hellman (CDH) avec succès non négligeable. Construire un algorithme $\mathcal{B}$ polynomial probabiliste contre List − CDH avec succès non négligeable.

(i) Réciproquement, soit un algorithme $\mathcal{B}$ polynomial probabiliste contre List − CDH avec succès non négligeable, construire un algorithme $\mathcal{A}$ polynomial probabiliste contre CDH avec succès non négligeable.

(j) Soit $\mathcal{A}$ un attaquant polynomial probabiliste IND−CPA avec avantage non négligeable contre la variante d'Elgamal dans le modèle de l'oracle aléatoire définie plus haut.

À partir de $\mathcal{A}$, construire un algorithme $\mathcal{B}$ et montrer qu'il a un succès non négligeable pour l'expérience $\mathbf{Exp}^{\mathsf{List-CDH}}_{\mathsf{GenDH},k}(\mathcal{B})$. Conclure sur la sécurité IND − CPA de ce chiffrement. Indications :

— $\mathcal{B}$ doit interagir avec $\mathcal{A}$ pour simuler l'expérience IND − CPA
— $\mathcal{B}$ doit simuler l'oracle aléatoire $\mathcal{H}$ auquel $\mathcal{A}$ a accès et utiliser la liste des requêtes de $\mathcal{A}$ pour résoudre le problème List − CDH
— Minorer la probabilité de succès de $\mathcal{B}$ en fonction de l'avantage de $\mathcal{A}$.

Exercice 2. Soit k un entier et soit f une permutation à trappe à sens-unique de $\{0,1\}^k$. Soit HC un algorithme polynomial déterministe, qui sous l'entrée $x \in \{0,1\}^k$ retourne un bit noté $\mathsf{HC}(x) \in \{0,1\}$. On dit que HC est un prédicat difficile pour f si pour tout algorithme polynomial probabiliste $\mathcal{A}$, si on choisit $x \xleftarrow{\$} \{0,1\}^k$ aléatoire avec probabilité uniforme, et si $p(k)$ désigne la probabilité que $\mathcal{A}$ retourne $\mathsf{HC}(x)$ sous l'entrée $(f, \mathsf{HC}, f(x))$, alors $\mathbf{Adv}^{\mathsf{HC}}(A) := |p(k) - 1/2|$ est négligeable.

On construit un schéma de chiffrement à clef publique comme suit. La clef publique est constituée de la donnée de f et de HC et la clef privée est la trappe permettant d'inverser f. Pour chiffrer un bit $m \in \{0,1\}$, on choisit $x \xleftarrow{\$} \{0,1\}^k$ aléatoire avec probabilité uniforme, et on pose $c = (f(x), \mathsf{HC}(x) \oplus m)$.

(a) Donner un algorithme de déchiffrement.

(b) Montrer que si HC est un prédicat difficile pour la permutation à trappe à sens-unique f, alors ce schéma de chiffrement est IND − CPA.

Partie G. Zémor

Exercice 3. On considère un code C de longueur n, donné par une matrice de parité $\mathbf{H}$ à $n/2$ lignes. On rappelle que mettre $\mathbf{H}$ sous forme systématique consiste à trouver une matrice de parité $\mathbf{H}'$ du même code C, dont une sous-matrice $(\mathbf{H}'_{ij})_{\substack{1\leq i\leq n/2 \\ j\in J}}$ où $|J| = n/2$, est la matrice identité $n/2 \times n/2$.

On suppose que l'on a mis $\mathbf{H}$ sous forme systématique suivant une partition $[1,n] = J \cup \overline{J}$ aléatoire des coordonnées. Soit $\mathbf{x}$ un mot de poids d du code C.

(a) Quel est la probabilité que $|\mathrm{supp}(\mathbf{x}) \cap J| = d-1$ et $|\mathrm{supp}(\mathbf{x}) \cap \overline{J}| = 1$? Comment peut-on reconnaître si on est dans un tel cas de figure ?

(b) Quel est approximativement le coût de chercher un mot de poids d de C de cette manière ?

(c) Sachant que systématiser la matrice $\mathbf{H}$ coûte de l'ordre de n^2 additions de vecteurs (n-uples), est-ce plus ou moins avantageux de chercher des partitions $(J, \overline{J})$ telles que $|\mathrm{supp}(\mathbf{x}) \cap J| = d-2$? Telles que $|\mathrm{supp}(\mathbf{x}) \cap J| = d-3$?

Exercice 4. On considère une source qui produit une matrice binaire $k \times n$ aléatoire $\mathbf{G}$, ainsi qu'un vecteur $\mathbf{y}$ qui est obtenu

1. soit en le choisissant uniformément dans $\mathbb{F}_2^n$,
2. soit en le choisissant de la forme $\mathbf{y} = \mathbf{x}\mathbf{G} + \mathbf{e}$, où $\mathbf{x}$ est choisi uniformément dans $\mathbb{F}_2^k$ et $\mathbf{e}$ est indépendamment choisi uniformément dans l'ensemble des vecteurs de poids t.

On fait l'hypothèse qu'il n'existe pas d'algorithme polynomial en n qui, étant donné $(\mathbf{G}, \mathbf{y})$ produit comme ci-dessus, décide avec un biais non-négligeable si on est dans le cas 1. ou le cas 2.

On cherche à montrer que sous cette hypothèse il n'existe pas d'algorithme $\mathcal{A}$ polynomial en n qui étant donnés une matrice aléatoire $\mathbf{G}$ comme ci-dessus et *deux* vecteurs $\mathbf{y}_1, \mathbf{y}_2$ produits :

1. soit en choisissant $\mathbf{y}_1, \mathbf{y}_2$ indépendants (au sens des probabilités) et uniformes dans $\mathbb{F}_2^n$,
2. soit en choisissant $\mathbf{y}_1 = \mathbf{x}_1\mathbf{G} + \mathbf{e}_1$ et $\mathbf{y}_2 = \mathbf{x}_2\mathbf{G} + \mathbf{e}_2$ indépendants et chacun de même loi que ci-dessus (aléatoire à distance t du code engendré par $\mathbf{G}$),

décide avec un biais non-négligeable si on est dans le cas 1. ou le cas 2.

(a) On suppose qu'un tel algorithme $\mathcal{A}$ existe. Supposons que l'on dispose d'une paire de vecteurs dont l'un est uniforme à distance t du code C engendré par $\mathbf{G}$, et l'autre est uniforme dans $\mathbb{F}_2^n$, mais nous ne savons lequel. Nous décidons aléatoirement d'appeler l'un des vecteurs $\mathbf{a}$ et l'autre $\mathbf{b}$. On tire à pile ou face et suivant le résultat on pose $\mathbf{y}_1 = \mathbf{a}$ et $\mathbf{y}_2 = \mathbf{b}$ ou $\mathbf{y}_1 = \mathbf{b}$ et $\mathbf{y}_2 = \mathbf{a}$. Puis on donne à l'algorithme $\mathcal{A}$ le triplet constitué de la matrice $\mathbf{G}$ et de $\mathbf{y}_1, \mathbf{y}_2$. Supposons que l'algorithme réponde encore «cas i.», $i = 1, 2$, comme s'il s'agissait d'une entrée légitime. Supposons que l'on déclare «$\mathbf{a}$ est uniforme» si $i = 1$ et «$\mathbf{a}$ est à distance t de C» si $i = 2$. Avec quelle probabilité a-t-on raison ?

(b) Supposons maintenant qu'on ait une instance $(\mathbf{G}, \mathbf{y})$, et que l'on souhaite déterminer si $\mathbf{y}$ est à distance t du code engendré par $\mathbf{G}$, ou bien uniforme dans $\mathbb{F}_2^n$. Montrer

comment on peut distinguer les deux cas avec un biais non-négligeable en utilisant l'algorithme $\mathcal{A}$ (toujours en supposant qu'il existe). En déduire que l'algorithme $\mathcal{A}$ ne peut pas exister.

(c) On considère le système de chiffrement asymétrique suivant : La clé publique est constituée d'une matrice $k \times n$ aléatoire uniforme $\mathbf{G}$, ainsi que deux vecteurs $\mathbf{y}_1 = \mathbf{x}_1\mathbf{G} + \mathbf{e}_1$ et $\mathbf{y}_2 = \mathbf{x}_2\mathbf{G} + \mathbf{e}_2$ où $\mathbf{x}_1$ et $\mathbf{x}_2$ sont aléatoires uniformes dans $\mathbb{F}_2^k$ et indépendants l'un de l'autre, et où $\mathbf{e}_1$ et $\mathbf{e}_2$ sont choisis aléatoirement parmi les vecteurs de poids t. La clé secrète est le couple $(\mathbf{x}_1, \mathbf{x}_2)$. Le message clair m est constitué de deux bits $m = (m_1, m_2)$ et le chiffré est obtenu à partir d'un vecteur $\mathbf{e} \in \mathbb{F}_2^n$ aléatoire de poids t et est constitué du triplet $\mathbf{G}\mathbf{e}^T$, $m_1 + \mathbf{y}_1\mathbf{e}^T$, $m_2 + \mathbf{y}_2\mathbf{e}^T$. Comment déchiffre-t-on ? Que doit-on supposer sur t pour que cela fonctionne ?

(d) Démontrer la sécurité CPA du chiffrement.

(e) Décrire une généralisation du procédé lorsque le clair m est constitué de k bits, $m \in \{0,1\}^k$.

(f) On considère la variante suivante : une matrice $k \times n$ aléatoire uniforme $\mathbf{G}$ est choisie, comme précédemment, ainsi que de k vecteurs $\mathbf{y}_1, \ldots, \mathbf{y}_k$ de la forme $\mathbf{y}_i = \mathbf{x}_i\mathbf{G} + \mathbf{e}_i$, pour $\mathbf{e}_i$ de poids t, $i = 1 \ldots k$, $2k < n$. On appelle $\mathbf{H}$ la matrice dont les lignes sont constituées des lignes de $\mathbf{G}$ ainsi que des vecteurs $\mathbf{y}_i$. La clé secrète consiste en la donnée des $\mathbf{e}_i, i = 1 \ldots k$. La clé publique consiste en une matrice $\mathbf{G}'$ de dimensions $k' \times n$, génératrice du code C' dont $\mathbf{H}$ est une matrice de parité. Le chiffré d'un message clair $m \in \mathbb{F}_2^{k'}$ est

$$m\mathbf{G}' + \mathbf{e}$$

où $\mathbf{e}$ est aléatoire de poids t. Comment peut-on déchiffrer à l'aide de la clé secrète ? Le système résiste-t-il à une attaque CPA ? Comment peut-on y remédier ?