TD4 - Elgamal et logarithme discret

(ElGamalnotCCAfr) **Exercise 1.** *CCA security for Elgamal*

Nous considérons le chiffrement ElGamal dans un groupe $\mathcal{G}$ où le problème CDH est difficile.

1. Montrer que le chiffrement ElGamal est "homomorphe pour la multiplication" : étant donnés c_1, c_2, le chiffrement de deux messages m_1 et m_2, il est possible de calculer un chiffrement valide de $m_1 \cdot m_2$ en ne connaissant pas m_1 et m_2.

Nous définissons maintenant l'expérience de sécurité suivante :

Definition 1 (Chosen Ciphertext Attack Security)**.** *Définissons deux expériences Exp_0 et Exp_1 :*

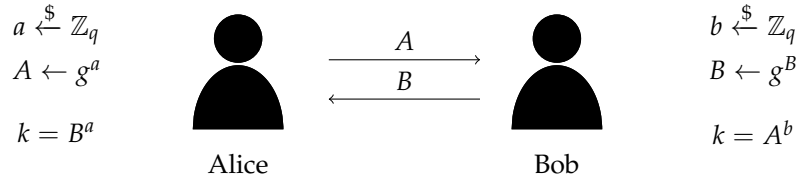
$\mathcal{C}$		$\mathcal{A}$
Échantillonner $(pk, sk) \leftarrow KeyGen(.)$		
	$\xrightarrow{pk}$	
	$\xleftarrow{c_i}$	
$m_i = Dec(sk, c_i)$	$\xrightarrow{m_i}$	
		Choisir m_0, m_1 $(\neq m_i)$
	$\xleftarrow{m_0, m_1}$	
Calculer $c \leftarrow Enc(pk, m_b)$		
	$\xrightarrow{c}$	
	$\xleftarrow{c'_i}$	*pour* $c'_i \neq c$
$m'_i = Dec(sk, c'_i)$	$\xrightarrow{m'_i}$	
		Calculer un bit b'*, le renvoyer*

l'avantage de l'adversaire est définie par : $Adv^{cca}(\mathcal{A}) = |\mathbb{P}\mathrm{r}[\mathcal{A} \rightarrow^{Exp_0} 1] - \mathbb{P}\mathrm{r}[\mathcal{A} \rightarrow^{Exp_1} 1]|$.
Si l'avantage de tout adversaire PPT est négligeable, alors on dit que le chiffrement est sûr pour la sécurité IND-CCA.

2. Montrez que le chiffrement ElGamal n'est pas sûr pour la sécurité IND-CCA.

(DH) **Exercise 2.** *Diffie-Hellman key exchange*

On considère l'échange de clé Diffie-Hellman comme suit :



1. Montrer que cet échange de clé est sûr sous l'hypothèse CDH (*i.e.* un adversaire qui observe A et B ne peut pas retrouver le secret partagé k).

2. Expliquer comment un adversaire peut se placer entre Alice et Bob et modifier les messages transmis (sans qu'Alice ou Bob ne s'en rende compte).

(indices) **Exercise 3.** *Calcul de log discret*

On s'intéresse à la méthode de calcul du log discret par calcul d'indices. L'idée est de trouver des relations entre les logs discrets de différents éléments qui serviront de base pour décomposer un élément quelconque.
On se place dans le groupe $\mathbb{G} = \mathbb{Z}_{101}^*$ et on prend $g = 11$.

1. Montrer que g est un générateur de $\mathbb{G}$.

2. Calculer g^2, g^3, g^4, et les décomposer en produits de nombres premiers.

3. En déduire des équations (modulaires) reliant $\log_g(2)$, $\log_g(3)$ et $\log_g(5)$.

4. Résoudre le système pour trouver $\log 2$, $\log 3$ et $\log 5$.

(Pollard) **Exercise 4.** *Pollard rho*

Soit $\mathbb{G}$ un groupe cyclique engendré par g, d'ordre premier q. Soit h un élément de $\mathbb{G}$ fixé. Soit $F : \mathbb{G} \longrightarrow \mathbb{Z}_q$ une fonction non identiquement nulle. On définit

$$H : \begin{array}{ccc} \mathbb{G} & \longrightarrow & \mathbb{G} \\ \alpha & \longmapsto & \alpha \cdot h \cdot g^{F(\alpha)}. \end{array}$$

On considère l'algorithme suivant :

Pollard ρ Algorithm

Input: $h, g \in \mathbb{G}$

Output: $x \in \{0, \ldots, q-1\}$ tel que $h = g^x$ of FAIL.

1. $i \leftarrow 1$
2. $x \leftarrow 0$, $\alpha \leftarrow h$
3. $y \leftarrow F(\alpha)$; $\beta \leftarrow H(\alpha)$
4. **while** $\alpha \neq \beta$ do
5. $\quad x \leftarrow x + F(\alpha) \bmod q; \alpha \leftarrow H(\alpha)$
6. $\quad y \leftarrow y + F(\beta) \bmod q; \beta \leftarrow H(\beta)$
7. $\quad y \leftarrow y + F(\beta) \bmod q; \beta \leftarrow H(\beta)$
8. $\quad i \leftarrow i + 1$
9. **end while**
10. **if** $i < q$ **then**
11. $\quad$ **return** $(x - y)/i \bmod q$
12. **else**
13. $\quad$ **return** FAIL
14. **end if**

Pour $i \in \mathbb{N}^*$, on définit (γ_i) par $\begin{cases} \gamma_1 := h \\ \forall\, i \geqslant 1, \gamma_{i+1} := H(\gamma_i) \end{cases}$.

1. Montrer que, dans la boucle **while** (lignes 4 à 9 de l'algorithme), on a $\alpha = \gamma_i = g^x h^i$ et $\beta = \gamma_{2i} = g^y h^{2i}$.

2. Montrer que si la boucle termine avec $i < q$, alors l'algorithme renvoie le logarithme discret de h en base g.

3. Soit j le plus petit entier tel que $\gamma_j = \gamma_k$ pour un certain $k < j$. Montrer que $j \leqslant q + 1$ et que la boucle termine avec $i < j$.

4. Montrer que si F est une fonction aléatoire (*i.e.*, choisie aléatoirement parmi les fonctions non nulles de $\mathbb{G}$ dans $\mathbb{Z}_q$), alors la complexité moyenne de l'algorithme est en $O(q^{1/2})$ multiplications dans $\mathbb{G}$.