On rappelle qu'on utilise la notation $x \leftarrow U(\mathcal{X})$ pour signifier que x est tiré uniformément au hasard dans $\mathcal{X}$. Certaines questions sont plus faciles (demandant d'appliquer des techniques vues en cours / TD), d'autres sont plus difficiles ; le barême final en tiendra compte.

Caractérisation de DDH

Soit $\mathcal{G}$ un groupe cyclique d'ordre q engendré par g. Soit $\mathcal{P}$ la distribution uniforme sur $\mathcal{G}^3$. Soit $\mathcal{P}_{DH}$ la distribution uniforme sur l'ensemble des triplets Diffie-Hellman : (g^a, g^b, g^{ab}). Soit $\mathcal{P}_{NDH}$ la distribution uniforme sur l'ensemble des triplets non-DH : $(g^a, g^b, g^c), c \neq ab$. On rappelle l'expression de la distance statistique entre deux variables aléatoires discrètes sur un ensemble A dénombrable :

$$\Delta(X, Y) = \frac{1}{2} \sum_{a \in A} |\Pr[X = a] - \Pr[Y = a]| \quad .$$

Question 1. *Montrer que la distance statistique entre $\mathcal{P}$ et $\mathcal{P}_{NDH}$ est $1/q$.*

Question 2. *Soit $\mathcal{D}$ un algorithme probabiliste en temps polynomial (PPT) faisant des requêtes à une certaine distribution, que l'on va utiliser comme distingueur.*

On définit trois jeux de sécurité (G1), (G2), (G3) dans lesquels $\mathcal{D}$:

(G1) Reçoit des échantillons de la distribution $\mathcal{P}_{NDH}$
(G2) Reçoit des échantillons de la distribution $\mathcal{P}$
(G3) Reçoit des échantillons de la distribution $\mathcal{P}_{DH}$

À l'aide de ces jeux, montrer que sous l'hypothèse DDH dans le groupe $\mathcal{G}$, $\mathcal{P}_{NDH}$ et $\mathcal{P}_{DH}$ sont indistinguables.

Problèmes équivalents

On s'intéresse ici à des réductions *déterministes en temps polynomial* entre différents problèmes. On dit que le problème A se réduit au problème B s'il existe un algorithme déterministe R pour résoudre A qui effectue des appels à une sous-routine résolvant B, tel que le temps de calcul de R est polynomial en la taille de l'entrée (sans compter le coût des sous-routines). Deux problèmes A et B sont dit équivalents si A se réduit à B et B se réduit à A.

Soit $(\mathcal{G}, \cdot)$ un groupe cyclique d'ordre q premier et g un générateur (on suppose que multiplication et inversion dans $\mathcal{G}$ sont calculables en temps polynomial en $\log q$). Notez bien que g est fixé pour l'ensemble de cet exercice.

Question 3. *Montrer qu'un élément de $\mathcal{G}$ n'a qu'une seule racine carrée, et donner un algorithme en temps polynomial pour la calculer.*

Question 4. *Montrer que les problèmes suivants dans $\mathcal{G}$ sont équivalents :*

(P1) Problème Diffie-Hellman calculatoire (CDH) : étant donné g^a et g^b, calculer g^{ab}
(P2) Étant donné g^a, calculer g^{a^2}
(P3) Étant donné g^a avec $a \neq 0$, calculer $g^{1/a}$
(P4) Étant donné g^a et g^b avec $b \neq 0$, calculer $g^{a/b}$

Choix de générateurs

On introduit les problèmes rDL, rCDH et rDDH (et hypothèses de sécurité correspondantes) qui sont les mêmes que les problèmes DL, CDH et DDH vus en cours, sauf que le générateur g est maintenant distribué uniformément dans $\mathcal{G}\backslash\{1\}$ (au lieu d'être fixe). Par exemple, pour rDL, l'adversaire observe une entrée de la forme g^r, g^{ar} où r et a sont uniformes, et doit trouver a. Dans cet exercice vous avez le droit de réutiliser les résultats de l'exercice "problèmes équivalents" si besoin.

Question 5. *Montrer que :*

1. *Les hypothèses (DL, DDH, CDH) impliquent respectivement les hypothèses (rDL, rDDH, rCDH) ;*
2. *L'hypothèse rDL implique l'hypothèse DL ;*
3. *L'hypothèse rCDH implique l'hypothèse CDH.*

Cryptosystème de Paillier

Le cryptosystème de Paillier est un schéma de chiffrement à clé publique utilisant des propriétés similaires à RSA.

Cryptosystème de Paillier

$\mathsf{KeyGen}(1^n)$:

- Générer deux premiers P, Q de n bits
- Soit $g = (N+1) \mod N^2 \in \mathbb{Z}_{N^2}^*$
- Calculer $N = PQ$, $d = \phi(N) = (P-1)(Q-1)$
- $\mathsf{pk} = N$, $\mathsf{sk} = d$

$\mathsf{Enc}(\mathsf{pk}, m)$:

- $r \leftarrow U(\mathbb{Z}_{N^2}^*)$
- $c := g^m r^N \in \mathbb{Z}_{N^2}^*$
- Renvoyer c

Dec ...

Question 6. *Soit $\mathcal{G} := \{(aN+1) \mod N^2, a \in \{0, \ldots, N-1\}\}$. Montrer que $\mathcal{G}$ est un sous-groupe multiplicatif de $\mathbb{Z}_{N^2}^*$ d'ordre N.*

Quels éléments de $\mathcal{G}$ sont générateurs ? Montrer que le problème du logarithme discret est facile dans $\mathcal{G}$.

Question 7. *Montrer que si $c = \mathsf{Enc}(\mathsf{pk}, m)$, alors $c^d = 1 \mod N$ (i.e., $c^d \in \mathcal{G}$). En déduire l'algorithme de déchiffrement.*

Indice : *calculer un logarithme discret.*

Question 8. *Montrer la sécurité IND-CPA sous l'hypothèse :*

- *Soit $N = PQ$ le produit de deux premiers de n bits*
- *Soit $u \leftarrow U(\mathbb{Z}_{N^2}^*)$*
- *Soit v uniforme dans le groupe $\{r^N : r \in \mathbb{Z}_{N^2}^*\}$*

alors la distribution (N, u) est calculatoirement indistinguable de (N, v).

L'une des propriétés intéressantes du cryptosystème de Paillier est son homomorphisme : si $c_0 = \mathsf{Enc}(\mathsf{pk}, m_0)$ et $c_1 = \mathsf{Enc}(\mathsf{pk}, m_1)$ alors $c_0 \cdot c_1$ est un chiffré valide pour $m_0 + m_1 \mod N$.

Collisions des collisions, tout n'est que collision

On admet le résultat suivant.

Proposition 1. *Soit* $H : \{0,1\}^n \to \{0,1\}^n$ *une fonction aléatoire. Soit* $Y(H)$ *le nombre d'élements de* $\{0,1\}^n$ *sans préimage par* H*. Alors :*

$$\Pr\left[Y(H) > 0.4 \times 2^n\right] \leq 0.0987^{2^n}$$

où la probabilité est prise sur le choix aléatoire et uniforme de la fonction H*.*

Pour utiliser plus facilement cette proposition, notez que faire une séquence de 2^n tirages uniformes dans $\{0,1\}^n$ est la même chose que tirer une fonction $\{0,1\}^n \to \{0,1\}^n$ au hasard.

On considère dans cet exercice des algorithmes de *fusion* opérant sur des *listes* de bit-strings, ordonnées selon l'ordre lexicographique. Si $s \in \{0,1\}^*$, on note $s|_t \in \{0,1\}^t$ la troncation de s à son préfixe de t bits. On note $x_1 \| x_2$ la concaténation des bit-strings x_1 et x_2 et $x_1 \oplus x_2$ leur XOR bit à bit. Enfin, on note 0_t une chaîne de t zéros.

L'opération de t-*fusion* de deux listes L_1, L_2 construit la liste $L_1 \bowtie_t L_2$ définie comme suit :

$$L_1 \bowtie_t L_2 := \{(x_1 \oplus x_2) \| x_1 \| x_2, x_1 \in L_1, x_2 \in L_2, (x_1 \oplus x_2)|_t = 0_t\} \ .$$

Formellement, il faut en effet conserver l'information des paires (x_1, x_2) ayant mené à un élément $(x_1 \oplus x_2)$ donné. Informellement, on se concentre sur les n premiers bits, et on suppose qu'on sait toujours retrouver les paires correspondantes (ce qui simplifie l'écriture des algorithmes). On écrira donc plutôt :

$$L_1 \bowtie_t L_2 := \{(x_1 \oplus x_2), x_1 \in L_1, x_2 \in L_2, (x_1 \oplus x_2)|_t = 0_t\} \ .$$

Question 9. *Soient* L_1 *et* L_2 *deux listes de taille* 2^n *dont les éléments sont distribués uniformément dans* $\{0,1\}^{n+m}$*. Montrer qu'il existe une constante universelle* c *telle qu'avec probabilité* $1 - \mathrm{negl}(n)$*, il existe une liste* $L' \subseteq L_1 \bowtie_n L_2$ *de taille* $\geq c2^n$ *avec les propriétés suivantes :*

- *Les éléments de* L' *(correspondant aux XORs* $(x_1 \oplus x_2)$*) sont distribués uniformément dans* $0_n \| \{0,1\}^m$ *;*
- *La liste* L' *peut être calculée en temps* $\mathcal{O}(2^n)$*.*

Indice : *ne gardez pas toutes les paires formant des collisions partielles ; assurez-vous de l'indépendance des paires que vous avez conservées.*

On cherche maintenant à résoudre le problème du k-XOR :

> Soit $H : \{0,1\}^n \to \{0,1\}^n$ une fonction aléatoire, trouver un k-uplet $(x_1, \ldots, x_k)$ d'élements distincts tels que :
>
> $$H(x_1) \oplus \ldots \oplus H(x_k) = 0_n \ .$$

Question 10. *Dans cette question* $k = 4$*. On propose l'algorithme suivant :*

1. *Construire quatre listes* L_1, L_2, L_3, L_4 *de taille* $\mathcal{O}\left(2^{n/3}\right)$ *de bit-strings de la forme* $H(x) \| x$ *où les entrées* x *sont tirées dans* $\{0,1\}^n$ *sans remise (la constante du* $\mathcal{O}$ *étant choisie plus tard).*
2. *Utiliser l'algorithme de la question précédente pour construire une liste* L_{12} *en* $n/3$*-fusionnant* L_1 *et* L_2*.*

3. *De même, fusionner L_3 et L_4 pour obtenir une liste L_{34}.*
4. *Chercher une collision sur les n premiers bits entre L_{12} et L_{34} ; la renvoyer si elle est trouvée.*

Montrer que cet algorithme réussit avec probabilité au moins constante, résout le problème du 4-XOR et a complexité $\mathcal{O}\left(2^{n/3}\right)$.

Question 11. *Modifier l'algorithme pour prendre une autre entrée $c \in \{0,1\}^n$, et renvoyer un 4-uplet $(x_1, \ldots, x_4)$ tel que $H(x_1) \oplus \ldots \oplus H(x_4) = c$.*

Question 12. *Montrer que pour toute constante k, il existe un algorithme en temps $\mathcal{O}\left(2^{\frac{n}{1+\lfloor \log_2 k \rfloor}}\right)$ résolvant le problème du k-XOR avec probabilité constante.*

Indice : *considérez d'abord le cas où k est une puissance de 2.*

Extension de l'espace de messages

Soit un schéma de chiffrement à clé publique $E = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ avec espace de messages $M = \{0,1\}^n$. On définit un schéma de chiffrement $E' = (\mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}')$ où :

- KeyGen' appelle KeyGen deux fois pour obtenir $(\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1)$ et renvoie les paires $\mathsf{pk}' = (\mathsf{pk}_0, \mathsf{pk}_1)$, $\mathsf{sk}' = (\mathsf{sk}_0, \mathsf{sk}_1)$
- $\mathsf{Enc}'(m_0, m_1) = \mathsf{Enc}(\mathsf{pk}_0, m_0)\,, \mathsf{Enc}(\mathsf{pk}_1, m_1)$
- $\mathsf{Dec}'(\mathsf{sk}', (c_0, c_1)) = \mathsf{Dec}(\mathsf{sk}_0, c_0)\,, \mathsf{Dec}(\mathsf{sk}_1, c_1)$

Question 13. *Montrer que si E est IND-CPA, alors E' est IND-CPA.*

Question 14. *Montrer que E' n'est pas IND-CCA.*