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 $1/q$.*

Solution. *Par définition il y a q^2 triplets DH, donc $q^3 - q^2$ triplets NDH. On a :*

$$\begin{aligned}
\Delta(X, Y) &= \frac{1}{2} \sum_{(a,b,c) \in \mathbb{Z}_q^3} |\Pr[X = (a, b, c)] - \Pr[Y = (a, b, c)]| \\
&= \frac{1}{2} \sum_{(a,b,c) \in DH} \left| \underbrace{\Pr[X = (a, b, c)]}_{=1/q^3} - \underbrace{\Pr[Y = (a, b, c)]}_{=0} \right| \\
&\quad + \frac{1}{2} \sum_{(a,b,c) \in NDH} \left| \underbrace{\Pr[X = (a, b, c)]}_{=1/q^3} - \underbrace{\Pr[Y = (a, b, c)]}_{=1/(q^3-q^2)} \right| \\
&= \frac{1}{2q} + \frac{1}{2}(q^3 - q^2) \left(\frac{1}{q^3 - q^2} - \frac{1}{q^3} \right) = \frac{1}{2q} + \frac{1}{2} \left(1 - \frac{q^3 - q^2}{q^3} \right) = \frac{1}{q} \quad .
\end{aligned}$$

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.

Solution. *Sous hypothèse DDH dans le groupe $\mathcal{G}$:*

$$\left| \Pr\left[\mathcal{D} \overset{G1}{\longmapsto} 1 \right] - \Pr\left[\mathcal{D} \overset{G2}{\longmapsto} 1 \right] \right| = \text{negl}$$

De plus par propriété de la distance statistique :

$$\left| \Pr\left[\mathcal{D} \overset{G2}{\longmapsto} 1 \right] - \Pr\left[\mathcal{D} \overset{G3}{\longmapsto} 1 \right] \right| = \frac{1}{q}$$

On utilise l'inégalité triangulaire pour conclure. Notez bien que le négligeable ici est en $\log q$, car q est la taille du groupe.

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 **impair** 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.*

Solution. *Comme le groupe est d'ordre q, $g^q = 1$. De plus $g^{q+1} = g$, donc $g^{(q+1)/2}$ est une racine carrée de g. (Parce que q est impair, ce que l'énoncé oubliait de préciser !).*

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 $g^{a/b}$

Solution. *P1 $\implies$ P2 est trivial.*

P2 $\implies$ P1 car on peut calculer $g^{a^2+b^2}$ puis $g^{(a+b)^2}$, diviser. On obtient g^{2ab}. On peut prendre la racine carrée (elle est unique car : tout entier k tel que $g^k = 1$ est divisible par q, l'ordre de $\mathcal{G}$, donc est multiple de q).

Comme on a équivalence on peut remplacer (P1 et P2) par (P12).

P12 $\implies$ P3 utilise le fait qu'avec P1, on peut calculer n'importe quelle puissance de la forme g^{a^k} pour un k fixé. En effet, comme on peut multiplier et passer au carré dans l'exposant, on a de quoi faire une exponentiation rapide (et c'est bien polynomial et déterministe).

Comme q est premier et $a \neq 0$, $a^{q-1} = 1 \mod q$ donc a fortiori $a^{q-2}a = 1 \mod q$, donc $a^{q-2} = 1/a \mod q$. Il suffit donc de calculer cette puissance.

P4 $\implies$ P3 est trivial.

P4 $\implies$ P12 car on peut d'abord calculer $g^{1/b}$ séparément, puis utiliser l'algorithme pour calculer $g^{a/(1/b)} = g^{ab}$.

P12 et P3 $\implies$ P4 est assez facile. On en déduit donc que P12 $\implies$ P4. Donc P12 et P4 sont équivalents. Et pour finir, comme P4 $\implies$ P3 et P12 $\implies$ P3 on a une équivalence de tous les problèmes.

Une autre réduction redondante :

P3 $\implies$ P2 car à partir de a et b on peut calculer les exposants : $1/a$, $1/b$, $1/a + 1/b$, $1/(1/a + 1/b) = ab/(a + b)$. En particulier remplaçons a par $(1 - r)$ et b par $(1 + r)$ (si on partait de g^r, on peut calculer ces exposants) alors on obtient l'exposant $(1 - r^2)/2$. En passant au carré : $(1 - r^2)$, et donc l'exposant r^2.

Dit de manière plus directe on calcule sur l'entrée g^a : g^{1-a} puis g^{1+a} puis $g^{1/(1-a)}$ puis $g^{1/(1+a)}$ puis $g^{1/(1-a)+1/(1+a)}$ puis $g^{(1-a^2)/2}$ puis g^{1-a^2} en passant au carré puis g^{a^2}.

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.*

Solution. *Soit un adversaire contre rDL. Sur l'entrée : g^r, g^{ar} on envoie g^{ar} à l'adversaire DL, et on envoie g^r à l'adversaire DL ; il trouve ar et r et on en déduit a.*

Soit un adversaire contre DL. Sur l'entrée : g, g^a on prend un r au hasard et on passe à la puissance, pour l'envoyer à l'adversaire rDL. Donc avantage ...

Soit un adversaire contre rCDH. Sur l'entrée g^r, g^{ar}, g^{br} on veut calculer g^{arb}. On peut utiliser l'adversaire CDH pour calculer g^{abr^2} et ensuite $g^{abr^2(r)^{-1}}$ par exemple en utilisant l'exo précédent.

Dans le sens inverse il suffit aussi de randomiser l'entrée.

Soit un adversaire contre rDDH, il suffit de randomiser l'entrée aussi.

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 $\mathcal{G}$.

Solution. *Sous-groupe :* $(aN+1)(bN+1) = (a+b)N+1 \mod N^2$.

Les puissances de $(aN+1)$ *sont de la forme* $kaN+1$. *Les générateurs sont tous les* $(aN+1)$ *où* a *est premier avec* N *(et dispose donc d'un inverse modulo* N*).*

Le DL est facile car il se résout avec une division euclidienne par N.

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.*

Solution. *On montre que* $c^d = 1 \mod N$. *En effet* $c^d = g^{md}r^{Nd}$ *et* $g^{md} = 1 \mod N$ *et* $r^{Nd} = (r^N)^d = 1 \mod N$ *par définition de* $d = \phi(N)$. *L'algorithme de déchiffrement consiste donc à calculer le DL de* c^d. *En effet,* $r^d = 1 \mod N$ *donc si on écrit* $(r^d) = g^{r'}$ *on a :* $c^d = g^{md+Nr'}$ *donc on élimine* r' *en calculant le DL, qui est* $md \mod N$. *On multiplie ensuite par l'inverse de* d *modulo* N. *Cet inverse existe-t-il ? Bonne question ! (Que je ne m'étais pas posée au départ). En toute généralité non. Mais de toute façon si* d *n'est pas premier avec* N*, le cryptosystème est trivialement cassé).*

Si P *et* Q *sont tous les deux de* n *bits, alors on a* $2P > Q$ *et* $2Q > P$. *Donc on ne peut pas avoir* $P|(Q-1)$. *De plus on n'a pas* $P|P-1$ *(trivial) et donc on n'a pas* $P|(P-1)(Q-1)$ *; de même pour* Q *; par conséquent* d *est premier avec* N. *Ouf !*

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).

Solution. *Comme la preuve de IND-CPA pour ElGamal. On considère un adversaire IND-CPA* $\mathcal{A}$ *auquel on fait croire qu'il vit dans le monde IND-CPA, et on l'utilise pour implémenter un adversaire distingueur* $\mathcal{B}$.

$\mathcal{B}$ *reçoit deux messages* m_0, m_1 *de* $\mathcal{A}$. *Lorsque* $\mathcal{B}$ *reçoit* x *de son challenger, il construit le chiffré challenge suivant :* xg^{m_b} *où* b *est un choix de bit random. Ensuite, il envoie ce chiffré à* $\mathcal{A}$. $\mathcal{A}$ *renvoie un bit* b'.

Si $b = b'$, *alors* $\mathcal{B}$ *renvoie "distribution* (N, v)*". Sinon, il renvoie "distribution* (N, u)*".*

On démontre ensuite que l'avantage de $\mathcal{B}$ *dans le jeu du distingueur (probabilité qu'il réussisse moins 1/2) est égal à l'avantage de* $\mathcal{A}$ *dans le jeu IND-CPA.*

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.*

Solution. *On part d'un adversaire $\mathcal{A}$ contre IND-CPA de E', et on construit un adversaire $\mathcal{B}$ contre IND-CPA de E.*

$\mathcal{B}$ reçoit d'abord une clé publique pk. Il prend un bit b au hasard et fabrique une clé publique $\mathsf{pk}_0, \mathsf{pk}_1$ avec $\mathsf{pk}_b = \mathsf{pk}$ et pk_{1-b} nouveau tiré au hasard. Il envoie cette clé publique à $\mathcal{A}$.

$\mathcal{A}$ fabrique une paire $(m_0^0, m_1^0), (m_0^1, m_1^1)$ et l'envoie à $\mathcal{B}$. $\mathcal{B}$ envoie (m_b^0, m_b^1) à son challenger (la paire correspondant à la clé publique que le challenger connaît). Le challenger renvoie un chiffré challenge c^* qui est donc $c^* = \mathsf{Enc}\big(\mathsf{pk}_b, m_b^{b'}\big)$ pour un bit b' que l'on doit deviner.

$\mathcal{B}$ complète le chiffré challenge en chiffrant m_{1-b}^b, et l'envoie à $\mathcal{A}$. Alors :

- *Si $b = b'$ le chiffré challenge envoyé à $\mathcal{A}$ est bien formé, et il peut distinguer avec probabilité $1/2 + \epsilon$*
- *Si $b \neq b'$ le chiffré challenge envoyé à $\mathcal{A}$ est aléatoire, et il distingue avec probabilité $1/2$*

Donc au total l'avantage de $\mathcal{B}$ est 1/2 de celui de $\mathcal{A}$. (C'est la même preuve que celle d'un des exercices du TD "signatures numériques").

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

Solution. *Choisir deux messages $m_0 \neq m_1$. Envoyer à C les messages (m_0, m_1) et (m_1, m_0). C renvoie un chiffré challenge (c_1, c_2). On demande le déchiffrement de (c_2, c_1) qui donne (m_b, m_{b-1}). On a ainsi résolu le problème.*