Exercice 1 (Cryptosystème de Merkle et Hellman)

Le but de cet exercice est d'étudier un cryptosystème à clé publique fondé sur la difficulté d'un problème complet de la classe *NP*. Il a cependant été cassé, ainsi que toutes les variantes qui en ont été dérivées. À ce jour, la cryptologie à clé publique ne repose pas sur l'hypothèse de la difficulté des problèmes *NP*-complet, mais sur la difficulté (supposée) de certains problèmes qui ont trait à la théorie des nombres. Néanmoins, ce cryptosystème dû à Merkle et Hellman a un intérêt historique puisqu'il fût l'un des premiers cryptosystèmes à clé publique publiés.

Le cryptosystème repose sur le problème SUBSETSUM (somme de sous ensembles) qui est *NP*-complet :

SUBSETSUM

entrée : Une suite σ de n nombres entiers $a_1, \ldots, a_n$ et un nombre S.

sortie : OUI s'il existe une sous-suite de nombres dont la somme est S, NON sinon

On s'intéresse à la version fonctionnelle, qui a les mêmes entrées mais qui demande en sortie un certificat, c'est-à-dire une sous-suite des a_i dont la somme vaut S s'il en existe. Cette sous-suite peut être encodée par un mot $b_1 \ldots b_n$ binaire de longueur n.

1. Soit $\sigma = (1, 5, 6, 11, 14, 20)$. Existe-t-il une solution pour les entrées $\sigma, 22$ et $\sigma, 24$?
2. Supposons que pour un certain ensemble d'instances, le problème SUBSETSUM est facile à condition de connaître une information supplémentaire. Plus précisément, on fait l'hypothèse suivante :

   il existe un ensemble Λ de suites d'entiers et il existe un ensemble de brèches B et une fonction $f : \Lambda \to B$ qui associe à chaque suite de Λ une brèche b tels que le problème

   entrée $\sigma \in \Lambda, b = f(\sigma)$ et S entier

   sortie un sous ensemble de σ dont la somme des éléments vaut S s'il en existe

   est facile.

   En déduire un cryptosystème à clé publique.
3. Une suite $a_1, \ldots, a_n$ est *super croissante* si pour tout $i, 1 < i \leq n$, $\sum_{j=1}^{i-1} a_j < a_i$. Montrer qu'il existe un algorithme polynomial qui résout le problème SUBSETSUM pour toute entrée (σ, S) où σ est une suite super croissante.
4. Dans le cryptosystème de Merkle et Hellman, la clé privée est une suite super croissante σ, et deux entiers q et N tels que N est plus grand que la somme des éléments de σ et q est premier avec N. La clé publique est la suite obtenue en multipliant chaque élément de σ par q modulo N.

   Quels sont les algorithmes de chiffrement et déchiffrement ? Soit $\sigma = (2, 3, 6, 13, 27, 52)$, $q = 31$ et $N = 105$. Chiffrer le message 011000 110101. Déchiffrer le message 280 333.
5. Si vous deviez implémenter ce cryptosystème, quelle taille (en bits) préconiseriez-vous pour la clé privée ?

Malheureusement, il existe des failles dans la transformation de la clé privée vers la clé publique qui permettent de retrouver la clé privée à partir de la clé publique ou de se passer de celle ci pour retrouver le message en clair. L'étude de cette attaque dépasse le cadre de ce TD. Voir Shamir, Adi,

A polynomial-time algorithm for breaking the basic Merkle - Hellman cryptosystem, Information Theory, IEEE Transactions on, vol.30, no.5, pp. 699-704, Sep 1984.

Exercice 2 (Cryptosystème de Rabin)

On note :
$C_4 = \{N \in \mathbb{N} \mid \exists p, q \text{ premiers}, |p| = |q|, p \equiv q \equiv 3 \pmod 4 \ , \ p \cdot q = N\}$
-$\mathbb{Z}_N = \{i \in \mathbb{N} \mid 0 \leq i \leq N-1\}$
-$\mathbb{Z}_N^* = \{i \in \mathbb{N} \mid 0 \leq i \leq N-1, \text{pgcd}(i, N) = 1\}$
-$Q_N = \{y \in \mathbb{Z}_N \mid \exists x \in \mathbb{Z}_N, x^2 \equiv y \pmod N\}$.

1. Soit p un nombre premier et $y \in Q_p$. Combien existe-t-il de racines carrées de $\dot{y}$ dans $\mathbb{Z}/p\mathbb{Z}$?
   Aide : $\langle \mathbb{Z}/p\mathbb{Z}, +, \cdot \rangle$ est un *corps* commutatif. On factorisera le polynome $X^2 - \dot{y}$ dans $\mathbb{Z}/p\mathbb{Z}[X]$.

2. Donner un algorithme efficace (de complexité polynomiale) pour le problème suivant (4RAC) :
   entrée : p premier tel que $p \equiv 3 \pmod 4$, $y \in Q_p$
   sortie : $\{x \in \mathbb{Z}_p \mid x^2 \equiv y \pmod p\}$.

   Soit $N = pq$ avec p, q premiers entre eux . On rappelle que $\psi : \dot{x} \mod N \mapsto (\dot{x} \pmod p, \dot{x} \pmod q)$ est un isomorphisme d'anneaux de $\mathbb{Z}/N\mathbb{Z}$ dans $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z}$.
3. Que valent $\psi^{-1}(\dot{1}, \dot{0}), \psi^{-1}(\dot{0}, \dot{1})$? Plus généralement, comment calculer $\psi^{-1}(y)$ pour $y \in \mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z}$?
4. Démontrer que, si la factorisation (p, q) de l'entrée N est connue, et $p, q \equiv 3 \pmod 4$, alors le problème suivant (4RAC) admet un algorithme de complexité polynomiale :
   entrée : $N \in C_4$, $y \in Q_N$
   sortie : $\{x \in \mathbb{Z}_N \mid x^2 \equiv y \pmod p\}$.
5. Montrer que, réciproquement, le problème (4FACTORISATION) ci-dessous, se réduit polynomialement à (4RAC) :
   entrée : $N \in C_4$
   sortie : $p \geq 2, q \geq 2$ tels que $p \cdot q = N$.
6. Définir une fonction à sens unique et à brèche secrète issue du problème (4RAC) ; en déduire un cryptosystème (dit de Rabin) à clés publiques.
7. Montrer qu'une attaque à texte chiffré est poynomialement équivalente à (4FACTORISATION).

8. Montrer qu'une attaque à texte clair est possible.

Exercice 3 (Cryptosystème RSA)

On considère ici le cryptosystème RSA (voir annexe) qui est fondé sur le problème algorithmique RacineIemeModulaire.

1. Montrer que, pour tout couple de clés (sk, pk), et tout message m, on a bien

$$\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m.$$

Aide : utiliser le théorème d'Euler ou l'un de ses corollaires.

2. Vérifier que, étant donnés des nombres premiers p, q et $N = p \cdot q$, on peut générer aléatoirement, selon une loi uniforme, un entier $e \in \mathbb{Z}_N$, tel que $\text{pgcd}(e, \varphi(N)) = 1$ en temps polynomial.

3. Vérifier que, étant donnés des nombres premiers p, q, l'entier $N = p \cdot q$, et $e \in \mathbb{Z}_N$, tel que $\text{pgcd}(e, \varphi(N)) = 1$, on peut calculer en temps polynomial un entier $d \in \mathbb{Z}_N$, tel que $d \cdot e \equiv 1 \pmod{\varphi(N)}$.

4. Montrer que les fonctions de chiffrement et de déchiffrement sont calculables en temps polynomial.

5. Démontrer que RacineIemeModulaire est plus facile que Factorisation (pour des réductions déterministes, en temps polynomial).

6. À partir de quelle hypothèse de théorie de la complexité, peut-on prouver qu'il est difficile :
   a- de déterminer la clé secrète à partir de la clé publique ?
   b- de déchiffrer les cryptogrammes sans connaître la clé secrète ?

Remarque : RacineIemeModulaire est plus facile que Factorisation ; l'inégalité en sens contraire est supposée vraie (espéré ?) mais non prouvée !

Exercice 4 (Cryptosystème d'El-Gamal)

On considère le cryptosystème d'El-Gamal (voir annexe) qui est fondé sur le problème algorithmique de Diffie-Hellman.

1. Montrer que, pour tout couple de clés (sk, pk), et tout message m, on a bien
$$\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m.$$

2. Étudier la sécurité de ce système. Pour cela, démontrer que l'un (au moins) des problèmes de Diffie-Hellman (on précisera lequel ou lesquels) est plus facile que déchiffrer les cryptogrammes d'El-Gamal en ne connaissant que la clé publique.

3. Un attaquant sait résoudre le problème du logarithme discret (par exemple, parcequ'il a fabriqué un ordinateur quantique). Peut-il déchiffrer les cryptogrammes d'El-Gamal en ne connaissant que la clé publique ?

Exercice 5 (Échanges de clés)

L'objet de cet exercice est d'étudier différents protocoles d'échanges de clés, entre Alice et Bob, via un canal C, qui peut être écouté par Eve ; la sécurité repose sur la difficulté des problèmes de DiffieHellman. On considère les propriétés suivantes :

Fonctionnalité À la fin du protocole, les deux participants partagent une même clé.

Rafraîchissement de la clé Chaque participant est assuré que la clé est nouvelle.

Autocertification des clés Chaque participant est assuré que seul son partenaire (celui avec qui il communique via le canal C : A ou B ou E) pourra connaitre la clé de session échangée.

Authentification des clés Chaque participant est assuré de l'identité de l'autre participant (A ou B, mais pas E)

Confirmation de la clé Chaque participant est assuré que son partenaire connaît la clé.

On suppose que Bob (resp. Alice) a pour clé privée $sk_B = (p, \alpha, x_{Bob})$ (resp. $sk_A = (p, \alpha, x_{Alice})$ et pour clé publique $pk_B = (p, \alpha, y_{Bob})$ (resp. $pk_A = (p, \alpha, y_{Alice})$). Ils communiquent via un canal C qui peut être écouté par Eve (l'espion) et sur lequel Eve peut aussi envoyer des messages. Les couples (p, α) considérés sont formés d'un entier premier p et d'un générateur α de $\mathbb{Z}_p^*$ et sont supposés tels que le problème DiffieHellman est difficile. On considère les protocoles suivants :

Version 1 Alice envoie à Bob $\mathsf{EncElGamal}\big((p, \alpha, y_{Bob}), K\big)$. La clé échangée est K.

Version 2 Alice envoie à Bob $\alpha^{K_1} \mod p$. Bob envoie à Alice $\alpha^{K_2} \mod p$. La clé échangée est $K := \alpha^{K_1 \cdot K_2} \mod p$.

Version 3 Alice envoie à Bob $\alpha^{K_1} \mod p$. La clé échangée est $K := y_{Bob}^{K_1} \mod p$.

Version 4 Alice envoie à Bob $\alpha^{K_1} \mod p$. Bob envoie à Alice $\alpha^{K_2} \mod p$. La clé échangée est $K := \alpha^{K_1 \cdot x_{Bob} + K_2 \cdot x_{Alice}} \mod p$.

Version 5 Alice envoie à Bob : $\alpha^{K_1} \mod p$.
Bob envoie à Alice : $\alpha^{K_2} \mod p$,
La clé échangée est $K := \alpha^{K_1 \cdot K_2} \mod p$.
Alice envoie à Bob : $\mathsf{Enc}(K, S(sk_A, [\alpha^{K_1} \mod p], [\alpha^{K_2} \mod p]))$.
Bob envoie à Alice : $\mathsf{Enc}(K, S(sk_B, [\alpha^{K_2} \mod p], [\alpha^{K_1} \mod p]))$.
Enc désigne ici la fonction de chiffrement d'un cryptosystème symétrique (par exemple A.E.S) et S est un procédé de signature (par exemple RSA).

Pour chacune des propriétés définies plus haut et chaque protocole, déterminer si le protocole satisfait la propriété. N.B. Il y a donc 5×5 cas à étudier.

Exercice 6 (Pile ou face)

Alice et Bob veulent jouer à pile ou face mais ils n'ont pas toujours de pièce à lancer, n'ont pas confiance l'un dans l'autre et sont parfois à 15000 km de distance. Par contre ils connaissent la cryptologie.

1. Alice propose le protocole suivant : Bob tire aléatoirement un bit (selon une loi uniforme), ensuite Alice tire léatoirement un bit (loi uniforme). Enfin Alice et Bob calculent le ou exclusif des deux bits qui est le résultat du tirage. Ce jeu est-il équitable ?

2. Alice et Bob prennent un café. Alice propose d'utiliser le protocole suivant pour déterminer qui paie les cafés :
   - Alice choisit un bit a
   - Charles, en qui A et B ont confiance, tire un bit aléatoire b, selon une loi uniforme.
   - si $a \oplus b = 1$ Alice gagne, sinon Alice perd.
   Bob doit-il accepter ?

3. Alice et Bob sont à 15000 km de distance l'un de l'autre ; ils communiquent par téléphone, et ne se font pas confiance. Bob propose le protocole suivant. Soit $f : \{0,1\}^* \to \{0,1\}^*$.
   (a) Alice *choisit* un mot x et calcule $y = f(x)$;
   (b) Alice envoie y à Bob ;
   (c) Après avoir reçu y, Bob *choisit* un bit b et l'envoie à Alice ;
   (d)
   (e)

   Compléter les deux dernières étapes de ce protocole.
   On suppose que leur canal de communication est sûr : chaque joueur peut-il *vérifier* qui est le gagnant ?
   Quelle hypothèse doit satisfaire la fonction f pour que ce jeu soit *équitable* ?
   Qui est favorisé si f ne satisfait pas l'hypothèse ?
4. Voici un autre protocole qui repose sur l'utilisation d'un cryptosystème « commutatif » :

$$\forall k1, k2 \in \mathcal{K}, \forall m \in \mathcal{M} : \mathsf{Dec}_{k1}(\mathsf{Enc}_{k2}(\mathsf{Enc}_{k1}(m))) = \mathsf{Enc}_{k2}(m)$$

   Cette propriété n'est en général pas vérifiée par les cryptosystèmes à clé secrète. Elle est par contre vérifiée par le systèmes à clé publique RSA.
   (a) Alice et Bob génèrent une paire (clé privée, clé publique) que l'on notera respectivement (sa, pa) et (sb, pb).
   (b) Alice choisit aléatoirement deux nombres $m_0 \equiv 0 \pmod 2$ et $m_1 \equiv 1 \pmod 2$ (m_0 peut être vu comme "pile" et m_1 comme "face").
   (c) Alice choisit une permutation σ de $\{0,1\}$ et envoie à Bob le message $c_0 := \mathsf{Enc}_{pa}(m_{\sigma(0)})$, puis le message $c_1 := \mathsf{Enc}_{pa}(m_{\sigma(1)})$.
   (d) Bob choisit un bit $i \in \{0,1\}$ et renvoie à Alice le résultat $\mathsf{Enc}_{pb}(c_i)$.
   (e) Alice déchiffre ce message avec sa clé privée et envoie à Bob le résultat $\mathsf{Dec}_{sa}(\mathsf{Enc}_{pb}(c_i)) = \mathsf{Dec}_{sa}(\mathsf{Enc}_{pb}(\mathsf{Enc}_{pa}(m_{\sigma(i)}) = \mathsf{Enc}_{pb}(m_{\sigma(i)}$.
   (f) Bob déchiffre le message et envoie le résultat $m_{\sigma(i)}$ à Alice.
   (g) Alice vérifie que $m_{\sigma(i)} \in \{m_0, m_1\}$ et lit le résultat du lancer.
   (h) Alice et Bob révèlent leurs clés secrètes.
   (i) si $\sigma(i) = 1$ Alice gagne, sinon Alice perd.

   Montrer que :
   - ce jeu est équitable
   - ce jeu est protégé contre la triche : chaque participant détecte immédiatement toute tentative de tricherie.

Exercice 7 (Preuve à divulgation nulle [1])

Paul prétend avoir résolu la grille de sudoku #2349 du journal l'Univers. Un problème de sudoku se présente sous la forme d'une grille de dimension 9×9, dont certaines cases sont pré-remplies avec

1. L'exercice est tirée de R. Gradwohl, M. Naor, B. Pinkas and, G. Rothblum, *Cryptographic and Physical Zero-knowledge Proof Systems for Solutions of Sudoku Puzzles*. Theory Comput. Syst. 44(2) : 245-268 (2009) `http://www.wisdom.weizmann.ac.il/~naor/PAPERS/sudoku_abs.html`

des entiers entre 1 et 9. Une solution au problème est l'attribution d'un entier $\in \{1, \ldots, 9\}$ à chaque case non pré-remplie de telle sorte que chaque ligne, colonne ou sous grille 3×3 ne comporte pas deux fois le même entier. Victoria a néanmoins quelques doutes et souhaiterait que Paul lui montre sa solution afin de pouvoir vérifier ses dires. Au lieu de cela, Paul propose d'effectuer le protocole suivant :

- **Paul** dessine au sol une grille de taille 9×9. Sur chaque case, il dépose 3 cartes. Si la case correspond à une case pré-remplie du problème, les trois cartes sont déposées faces ouvertes ; leur valeur est celle indiquée dans la case correspondante. Sinon, les trois cartes sont déposées faces cachées.
- **Victoria** Pour chaque ligne/colonne/sous-grille, choisit (aléatoirement) une des trois cartes de chaque case dans la ligne/colonne/sous-grille correspondante.
- **Paul** rassemble les cartes choisies par Victoria en tas : un tas par ligne/colonne/sous-grille. Il mélange ensuite chacun des tas séparément.
- **Victoria** récupère les tas et vérifie qu'aucun d'entre eux ne contient deux cartes de même hauteur. Si c'est le cas, Victoria **accepte** l'affirmation de Paul et **rejette** sinon.

1. Montrer que si Paul a effectivement résolu la grille, et que Paul et Victoria suivent le protocole, Victoria **accepte** toujours. Le système de preuve est dit *consistant*.
2. Supposer que Paul ne connaît pas la solution du problème. Montrer que Victoria accepte avec probabilité au plus 1/3, même si Paul ne respecte pas le protocole. Bonus : montrer que cette probabilité est en fait au plus 1/9. Le protocole est dit *robuste* : Victoria rejette avec une probabilité non nulle les assertions fausses.
3. En déduire un protocole dans lequel Victoria accepte avec probabilité $< 1/3^{10}$ lorsque Paul ne connaît pas la solution.
4. Démontrer que le système de preuve est *à divulgation nulle* : le protocole ne révèle rien sur la solution du problème, même si Victoria ne respecte pas sa partie du protocole.

Un procédé de mise en gage est la donnée d'un algorithme Commit qui prend en paramètre des couples $\in \mathcal{M} \times \mathcal{K}$ et vérifie les propriétés

- *Engagement* Il n'existe pas $m \neq m' \in \mathcal{M}, k, k' \in \mathcal{K}$ tels que $\mathsf{Commit}(m, k) = \mathsf{Commit}(m', k')$.
- *Dissimulation* Étant donné $\mathsf{Commit}(m, k)$, le problème de calculer m (ou toute fonction non triviale $f(m)$) est difficile.

5. Supposer l'existence de procédés d'engagement. Donner un protocole de preuve à divulgation nulle pour le problème du sudoku qui repose sur de tels procédés plutôt que sur la manipulation de paquets de cartes.
6. Le problème sudoku est NP-complet [2]. Paul prétend qu'un graphe G est 3-coloriable. Peut-il le démontrer à Victoria sans indiquer comment le 3-colorier ?

2. Takayuki Yato, Complexity and Completeness of Finding Another Solution and its Application to Puzzles, Master thesis, U. of Tokyo, Jan 2003 `http://www-imai.is.s.u-tokyo.ac.jp/~yato/data2/MasterThesis.ps`