Exercice 1 (Sécurité inconditionnelle)

Le but de cet exercice est d'étudier les conditions dans lesquelles les chiffrements par décalage, par substitution monoalphabétique ainsi que le chiffrement de Vigenère sont inconditionnellement sûrs.

1. Montrer que si un seul caractère est chiffré, le chiffrement par décalage est inconditionnellement sûr.
2. Quel est la plus grande taille de l'espace des messages en clair $\mathcal{M}$ pour laquelle le chiffrement par substitution monoalphabétique est inconditionnellement sûr ?
3. Montrer comment utiliser le chiffrement de Vigenère pour chiffrer tout mot de longueur t de façon inconditionnellement sûre.

Exercice 2 (Chiffrement de Vernam)

Lorsqu'on utilise le chiffrement de Vernam (Masque jetable) avec la clé $k = 0^\ell$, on obtient $\mathsf{Enc}_k(m) = 0^\ell \oplus m = m$ et le message est donc envoyé en clair ! Il est suggéré pour améliorer le cryptosystème d'éliminer 0^ℓ de l'ensemble des clés.(C'est-à-dire que l'algorithme Gen choisit la clé aléatoirement et uniformément parmi l'ensemble des chaînes binaires de longueur ℓ différentes de 0^ℓ). Ce nouveau système est-il inconditionnellement sûr ? Donner une démonstration qui justifie votre réponse. En particulier, si votre réponse est positive, expliquer pourquoi le chiffrement de Vernam n'est pas décrit de cette façon. Si votre réponse est négative, réconcilier celle-ci avec le fait que pour la clé 0^ℓ, le chiffrement ne change pas le texte en clair.

Exercice 3 (Chiffrement par substitution polyalphabétique)

Soit $\Gamma = (\mathsf{Enc}, \mathsf{Dec}, \mathcal{K}, \mathcal{M}, \mathcal{C})$ le cryptosystème suivant :

— L'ensemble $\mathcal{M}$ des messages en clair est l'ensemble des mots sur l'alphabet $\mathcal{A} = [\mathsf{a}, \mathsf{z}]$

— Une clé k est la donnée de n permutations $\langle \sigma_0, \ldots, \sigma_{n-1} \rangle$, où chaque σ_i est une permutation de $[\mathsf{a}, \mathsf{z}]$. L'ensemble des clés $\mathcal{K}$ est l'ensemble des séquences de permutations de $[\mathsf{a}, \mathsf{z}]$. La clé est choisie aléatoirement de façon uniforme parmi toutes les séquences de permutations.

— Soient $m = a_0 \ldots a_{\ell-1} \in \mathcal{M}$ et $k = \langle \sigma_0, \ldots, \sigma_{n-1} \rangle \in \mathcal{K}$.

$$\mathsf{Enc}_k(m) = \sigma_0(a_0)\sigma_1(a_1)\ldots\sigma_{i \mod n}(a_i)\ldots\sigma_{\ell-1 \mod n}(a_{\ell-1})$$

1. Définir l'algorithme de déchiffrement Dec.
2. Est-il facile de concevoir une attaque à texte clair choisi ?
3. Est-il facile de concevoir une attaque à texte clair connu ?
4. Est-il facile de concevoir une attaque à texte chiffré connu ?
   Pour chaque attaque, indiquer la longueur des textes nécessaires pour retrouver la clé.

Exercice 4 (Cryptanalyse du chiffrement par substitution monoalphabétique)

Le but de cet exercice est de se rendre compte par la pratique de la faiblesse du chiffrement par substitution monoalphabétique. Le texte en clair est entièrement écrit en minuscules, sans espaces ni accent. Instructions dans l'archive TP.

1. Retrouver le texte en clair :)
2. Verifiez si possible la Loi de Zipf .

Exercice 5 (Modes d'utilisation des chiffrements par blocs)

Soit $\Gamma = (\mathsf{Enc}, \mathsf{Dec}, \mathcal{K}, \mathcal{M}, \mathcal{C})$ un cryptosystème sur $\mathcal{M} = \mathcal{C} = \{0,1\}^n$. On souhaite utiliser ce cryptosystème pour chiffrer/déchiffrer des messages m (mots binaires) de taille quelconque $|m| > n$.

Pour cela le mot m est découpé en blocs $m_1, \ldots, m_\ell$, chacun de longueur n, tels que $m = m_1 m_2 \ldots m_\ell$. Le chiffrement c_j de chaque bloc m_j est une fonction des blocs $m_1, \ldots, m_j$. Cette fonction peut évidemment appliquer l'algorithme de chiffrement Enc sur l'un ou plusieurs de ces blocs.

1. Écrire les fonctions de déchiffrement des modes ECB (Electronic Codebook), CBC (Cipher-block Chaining), CFB (Cipher Feedback) et OFB (Output Feedback). Ces modes sont associées aux fonctions $c_j = \mathsf{Enc}_k(m_j)$, $c_j = \mathsf{Enc}_k(c_{j-1} \oplus m_j)$, $c_j = m_j \oplus \mathsf{Enc}_k(c_{j-1})$ et $c_j = m_j \oplus z_j$ avec $z_j = \mathsf{Enc}_k(z_{j-1})$.
2. Si une erreur de transmission venait à altérer l'un des bits de l'un des blocs du texte chiffré, pouvez-vous estimer pour chacun des 4 modes la différence entre le texte clair initial et le texte déchiffré ?
3. Indiquez dans chaque cas si le chiffrement ou déchiffrement peuvent chacun être parallélisés.
4. Définir un nouveau mode de chiffrement par blocs utilisant une et une seule fois les opérateurs Enc_k et $\oplus$ (XOR). Présente-t-il un intérêt ?

Exercice 6 (Attaque exhaustive de cryptosystèmes à clé secrète)

Soit $\Pi = (\mathsf{Enc}, \mathsf{Dec}, \mathcal{K}, \mathcal{M}, \mathcal{C})$ un cryptosystème à clé secrète sur l'ensemble de messages en clair $\mathcal{M}$. Soient $\mathcal{K}$ et $\mathcal{C}$ les ensembles des clés et des messages chiffrés de ce cryptosystème. Supposons les égalités $\mathcal{M} = \mathcal{C} = \mathcal{K}$ et supposons que l'on sache efficacement calculer pour un message fixé $m_0 \in \mathcal{M}$ toute image inverse de la fonction $f : \mathcal{K} \to \mathcal{C}$ définie par $f(k) = \mathsf{Enc}_k(m_0)$. Quelle attaque peut-on mener ? Préciser notamment sa nature et sa complexité en temps et en espace en fonction de celles du calcul de f^{-1}.

Exercice 7 (Générateur pseudo aléatoire matériel)

Un *registre à décalage à rétroaction linéaire (RDRL)* est un moyen matériel de générer efficacement des suites de bits pseudo aléatoires. Un RDRL consiste en un registre R à décalage à p positions. Initialement, le bit à la position i est le $i$ème bit de la clé k ($\forall i \in [0, p-1], R[i] = k[i]$). A chaque cycle d'horloge, un nouveau bit de la suite pseudo aléatoire est généré et les opérations suivantes sont exécutées :

— Le bit $R[0]$ est renvoyé (c'est le bit généré), comme dernier bit de la suite ;
— Les bits $R[1], \ldots, R[p-1]$ sont décalés d'un pas vers la gauche ;
— La nouvelle valeur de $R[p]$ est donnée par

$$\sum_{j=0}^{p-1} \alpha_j R[j] \mod 2$$

où $\alpha_j \in \{0,1\}$, pour chaque $j \in [0, p-1]$ (rétroaction linéaire).

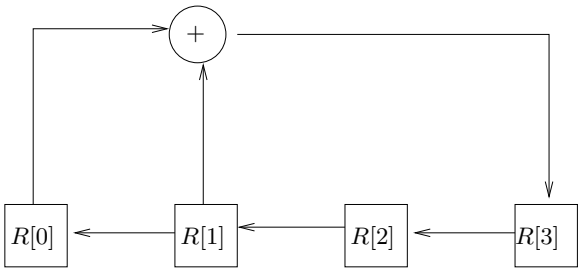


FIGURE 1 – Un registre à décalage et à rétroaction linéaire

1. Donner l'équation qui correspond au registre de la Figure 1. Montrez que la suite générée est ultimement périodique. Quelle est la période maximale ? Étudiez les variations de la période en fonction de la valeur de la clé.
2. *Attaque à texte clair connu.* Soit le RDRL défini par l'équation

$$\forall n \geq 0, z_{n+p} = \sum_{j=0}^{p-1} \alpha_j z_{n+j} \mod 2 \tag{1}$$

où $\alpha_j, 0 \leq j \leq p-1$ sont des constantes $\in \{0,1\}$ inconnues de l'attaquant. Un message m est chiffré par Alice en $c = m \oplus s$ où s est la suite de bits générée par le RDRL décrit à l'équation (1) initialisé avec la clé k.

L'attaquant obtient le texte chiffré c et connaît également les ℓ premiers bits $m|_{0..\ell-1}$ du message en clair. En supposant que p est connu, expliquez comment l'attaquant peut calculer la suite s et ainsi retrouver la totalité du texte en clair.

3. Que pensez vous de l'utilisation de RDRL dans le cadre d'applications cryptographiques ? Argumentez votre réponse.

Exercice 8 (DES)

Le Data Encryption Standard (DES) a une propriété étonnante. Le but de cet exercice est de démontrer cette propriété. Pour plus de détails, la figure 2 présente le workflow du DES. Considérons un nombre A représenté en binaire composé de n bits. Notons A' le complément de A, que l'on définit comme $A' = A \oplus \{1\}^n$ (chaque bit de A' est l'inverse de A). Le DES a la propriété étonnante suivante. Si

$$y = DES(k, x)$$

avec x le message clair représenté en tant que nombre binaire, y le message chiffre représenté en tant que nombre binaire et k la clé, alors,

$$y' = DES(k', x')$$

1- Démontrer que pour tout A et B nombres en binaire de même longueur, nous avons :

$$A' \oplus B' = A \oplus B$$

et

$$A' \oplus B = (A \oplus B)'$$

2- Étude des clés.
- Montrez que $PC-1(k') = (PC-1(k))'$
- Montrez que $LS_i(C'_{i-1}) = (LS_i(C_{i-1}))'$
- À partir des 2 résultats précédents, montrez que pour i = 1, 2 ,...,16, si k_i est une clé généré a partir de k, alors k'_i est une clé généré à partir de k'.

3- Étude des messages.
- Montrez que $IP(x') = (IP(x))'$
- Montrez que $E(R'_i) = (E(R_i))'$

4- Étude des messages et clés.
- En utilisant tout les résultats précédents, montrez que si R_{i-1}, L_{i-1}, k_i génèrent R_i, alors R'_{i-1}, L'_{i-1}, k'_i génèrent R'_i
- Déduisez en la propriété de l'exercice.

5- Utilisation de la propriété.
- À votre avis, cette propriété est elle valide si on change les valeurs des boites de substitutions dans le réseau de Feistel ?
- Comment cette propriété affecte l'exploration des clés lorsque l'on cherche a casser un chiffrement avec une recherche exhaustive ?

Exercice 9 (Signature à clés publiques)

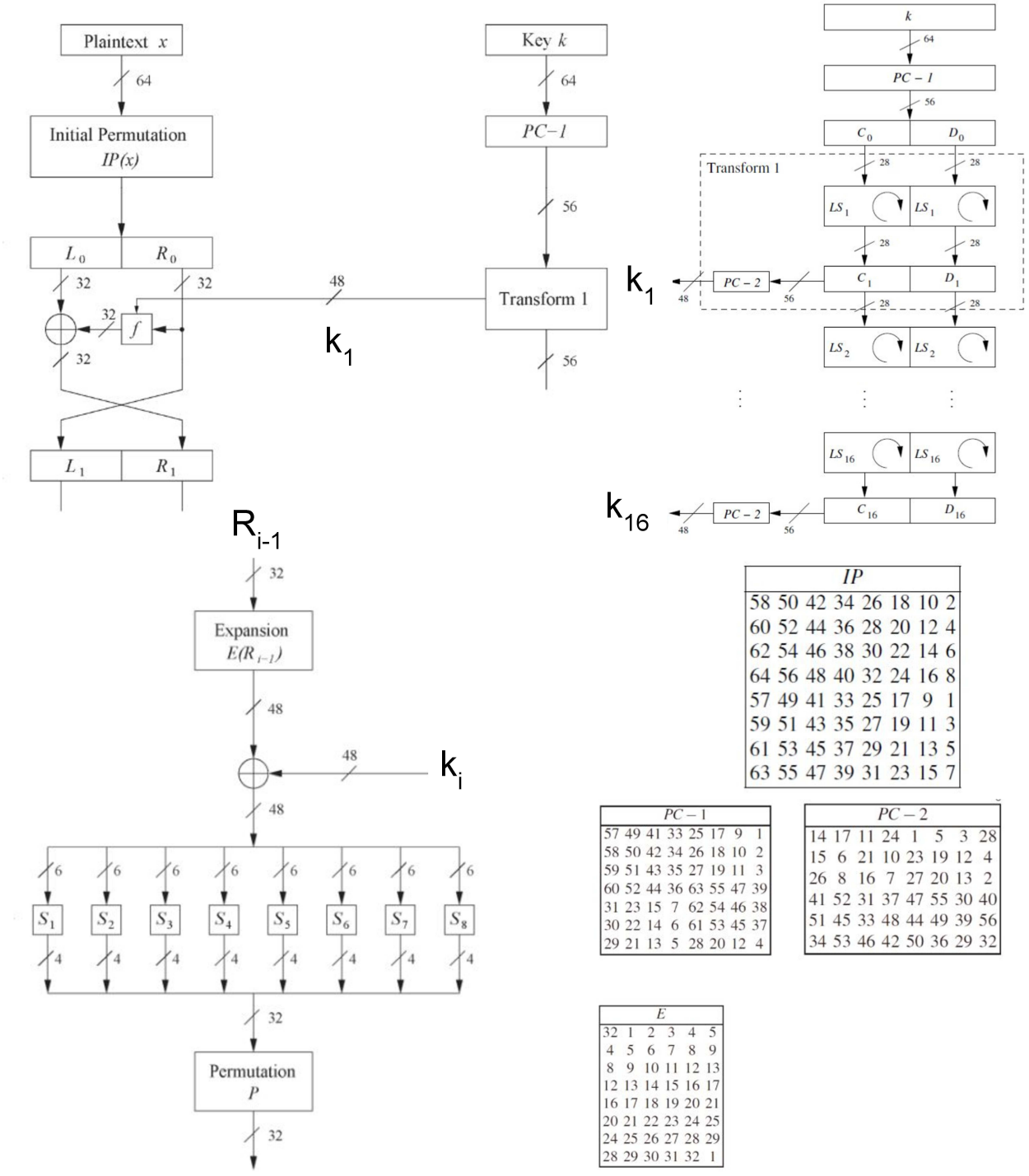
On souhaite construire un système de signatures à *clés publiques*. La différence avec les systèmes d'authentification vus en cours est que la clé se compose de deux parties (sk, pk). La partie *privée* sk n'est connue que de celui/celle qui signe ; la partie publique, connue de tous est utilisé pour vérifier que la signature d'un message est valide. Plus précisément, un système de signatures à clés publiques sur un espace de messages $\mathcal{M}$ se compose de trois algorithmes (G, S, V) :
- G est un algorithme probabiliste qui retourne un couple (sk, pk). sk est la clé secrète qui sert à signer et pk est la clé publique utilisée pour vérifier.
- L'algorithme S prend en entrée une clé secrète sk et un message m. Il produit en sortie une signature σ.
- L'algorithme V prend en entrée une clé publique pk, un message m et une signature σ. Il produit en sortie un Booléen 0 (interprété comme "rejette") ou 1 (interprété comme "accepte").

A minima, il est requis que la propriété de *validité* suivante soit satisfaite :

$$\forall (sk, pk) \text{ retourné par } G(), \forall m \in \mathcal{M} : V(pk, m, S(sk, m)) = 1$$

La propriété de sécurité recherchée est la même que celle vue en cours pour les systèmes de signature à clés secrètes : contrefaçon existentielle. La seule différence est que l'adversaire reçoit, en plus de messages et leur signature, la clé publique.

Signature à usage unique On souhaite signer des messages de b bits, c-à-d $\mathcal{M} = \{0,1\}^b$. Soit $f : \mathcal{X} \to \mathcal{Y}$ une fonction. On considère le système de signatures défini ainsi :
On note $m[i]$ le ième bit de m.
- $G()$: choisir aléatoirement $2b$ éléments de $\mathcal{X}$ notés $x_1^0, \ldots, x_b^0, x_1^1, \ldots, x_b^1$. La clé secrète est le couple (sk^0, sk^1) où $sk^0 = [x_1^0, \ldots, x_b^0]$ et $sk^1 = [x_1^1, \ldots, x_b^1]$; la clé publique est le couple (pk^0, pk^1) où $pk^0 = [f(x_1^0), \ldots, f(x_b^0)]$ et $pk^1 = [f(x_1^1), \ldots, f(x_b^1)]$.



IP							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

PC − 1							
57	49	41	33	25	17	9	1
58	50	42	34	26	18	10	2
59	51	43	35	27	19	11	3
60	52	44	36	63	55	47	39
31	23	15	7	62	54	46	38
30	22	14	6	61	53	45	37
29	21	13	5	28	20	12	4

PC − 2							
14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

E					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

FIGURE 2 – Le texte clair x est chiffré avec la clé k. Le schéma en haut a gauche présente la première itération de chiffrement du DES. Le schéma a droite montre les décalages des clés. Le schéma en bas a gauche montre la fonction f, aussi appelé réseau de Feistel, pour une itération. Les permutations ou extensions sont données à titre informatif.

— $S(sk, m)$. La signature σ est un vecteur de taille b. Pour chaque $i, 1 \leq i \leq b$,

$$\sigma[i] = \begin{cases} sk^0[i] \text{ si } m[i] = 0 \\ sk^1[i] \text{ sinon} \end{cases}$$

— $V(pk, m, \sigma)$. L'algorithme de vérification procède de manière évidente. Si pour tout $i, 1 \leq i \leq b$, $f(\sigma[i]) = pk^{m[i]}[i]$ alors 1 est retourné, et 0 si ce n'est pas le cas.

1. Indépendamment de la fonction f, démontrez qu'il existe une attaque dès que l'attaquant peut obtenir la signature d'au moins deux messages. Décrire sous forme de pseudo-code l'attaque. L'attaquant connaît f, celle-ci faisant partie des algorithmes de génération de clé et de vérification.
2. On se limite à la signature d'un seul message (Si Sara souhaite signer plusieurs messages, elle doit générer un nouveau couple (sk, pk) pour chaque message.). Quelle propriété vue en cours doit satisfaire f pour le système soit sûr ? Démontrez que si f satisfait cette propriété alors tout attaquant efficace n'a qu'une probabilité négligeable de produire une contrefaçon (l'attaquant ne peut demander que la signature d'un unique message.). On pourra supposer qu'une attaque existe et en déduire que f ne satisfait pas la propriété désirée.
3. Un attaquant est capable d'effectuer 100.10^9 calculs de f par seconde. Quelle taille des espaces d'entrée $\mathcal{X}$ et de sortie $\mathcal{Y}$ de la fonction f préconisez-vous ? Justifiez votre réponse.
4. On souhaite signer un message de taille arbitraire. Comment procéder ?

Signature à usage multiple Sara souhaite maintenant signer N messages. Elle peut générer N couples $((sk_1^0, sk_1^1), (pk_1^0, pk_1^1)), \ldots, ((sk_N^0, sk_N^1), (pk_N^0, pk_N^1))$, publier les N clés publiques correspondantes et utiliser (sk_i^0, sk_i^1) pour signer le i-ème message. Le destinataire, Victor, fournira alors pk_i en entrée de l'algorithme de vérification. Cette approche a bien sûr l'inconvénient qu'un grand nombre de clés doit être manipulé. Essayons de réduire ce nombre.

Soit $x_1, \ldots, x_N \in \mathcal{X}$ et $H : \mathcal{X} \to \mathcal{Y}$ une fonction de hachage. Pour simplifier, on supposera que N est une puissance de 2. Un *arbre de Merkle* est un arbre binaire tel que (Voir Figure 3) :

— Les feuille contiennent les hachés $H(x_1), \ldots, H(x_N)$.
— Chaque nœud interne contient le haché de la concaténation des valeurs de ses fils.

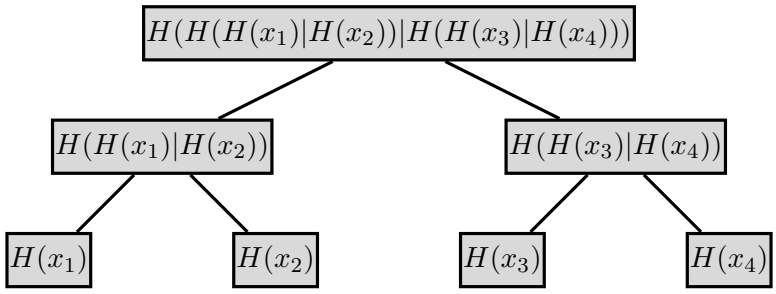


FIGURE 3 – Arbre de Merkle. | dénote la concaténation.

1. Victor ne connaît que la racine r de l'arbre. Soit $x \in \{x_1, \ldots, x_N\}$. Comment procéder pour que Sara, qui a contruit l'arbre, convainque Victor que x fait bien partie des x_i à partir desquels l'arbre a été construit ? A contrario, pourquoi n'est-t-il pas faisable de persuader Victor que $x' \notin \{x_1, \ldots, x_N\}$ fait partie des éléments à partir desquels l'arbre a été généré ?

2. En utilisant un ou plusieurs arbres de Merkle, donner un système de signature à clé publique qui permet de signer N messages et dans lequel la clé publique est constitué d'un seul élément (c'est-à-dire une diminution d'un facteur $O(N)$ de la taille de la clé publique par rapport à la solution naïve qui consiste à publier N clés du système à usage unique).
3. Pour quelle(s) raison(s) n'est-il pas faisable pour un attaquant de produire une contrefaçon ?

Exercice 10 (Inversion d'une fonction à sens unique)

Soit $f : \mathcal{K} \to \mathcal{K}$ une fonction pour laquelle le calcul de $f(x)$ pour un quelconque $x \in \mathcal{K}$ a des complexités en temps et en espace $O(1)$. L'objet de cet exercice est de fournir différents algorithmes de calcul de f^{-1} et d'en estimer les complexités. Dans chacun des cas suivants, calculer la complexité en temps et en espace des attaques suivantes :

1. Attaque exhaustive sans précalcul
2. Attaque exhaustive avec précalcul

   Une attaque exhaustive peut se dérouler en deux étapes :
   — un précalcul calculant une table contenant tout couple $(f(k), k)$ avec $k \in \mathcal{K}$.
   — l'attaque proprement dite.
3. Compromis espace-temps

   L'intérêt du compromis espace-temps est d'obtenir une complexité en temps de l'attaque en $O(|\mathcal{K}|^c)$ et une complexité en espace du précalcul en $O(|\mathcal{K}|^d)$ avec $0 < c < 1, 0 < d < 1$ (la complexité en temps du précalcul est $O(|\mathcal{K}|)$. Hellman implémente cette idée (voir l'ouvrage Cryptography Theory and Pratice de D. Stinson) et obtient une attaque du D.E.S avec $c = d = \frac{2}{3}$.

   Indication : lors du précalcul, on calcule une table contenant $k_0, f^T(k_0), f^{2 \cdot T}(k_0), \ldots$, où k_0 est fixé et $1 < T < |\mathcal{K}|$ est un entier fixé que vous choisirez. Afin de simplifier, nous supposerons que f est bijective et possède un unique cycle c'est-à-dire qu'il existe $k_0 \in \mathcal{K}$ tel que tout élément $k \in \mathcal{K}$ est de la forme $f^n(k_0)$ où n est un entier.

On pourra donner une valeur précise des complexités en temps et en espace des différentes attaques en supposant :
— un espace $\mathcal{K}$ de cardinalité 2^{56} (cas du D.E.S) ou 2^{128} (cas de A.E.S)
— qu'une instruction élémentaire s'exécute en $10^{-9}s$.
— que 10 Gigaoctets coûtent 1 euro.
D'autres hypothèses peuvent être faites : en 1993, Wiener proposa de construire une machine d'un million de dollars contenant 57600 puces, réalisant chacune 50000 chiffrements D.E.S par seconde.

Exercice 11 (Fonction à sens unique)

1- Soit $f : \{0,1\}^* \to \{0,1\}^*$ une application calculable, injective. L'inverse de f, notée f^{-1}, est une fonction (partielle). Est-elle calculable ?
2- Soit $f : \{0,1\}^* \to \{0,1\}^*$ une application calculable (non nécessairement injective). Montrer que l'on peut construire une application $\bar{f} : \mathbb{N} \times \{0,1\}^* \to \{0,1\}^*$ calculable, telle que

$$\forall n \in \mathbb{N}, \forall x \in \{0,1\}^n, f(\bar{f}(n, f(x))) = x.$$

Voici une définition formelle de la notion de fonction à sens unique. Une expérience dans laquelle un « challenger » cherche à vérifier qu'un « adversaire » $\mathcal{A}$ est capable d'inverser une fonction f est définie comme suit : Expérience $\textsc{Invert}_{\mathcal{A},f}(n)$

- Le challenger choisit aléatoirement $x \in \{0,1\}^n$ et calcule $y = f(x)$
- $\mathcal{A}$ reçoit (n, y) en entrée et produit x'
- Le résultat de l'expérience est 1 si $f(x') = y$ et 0 sinon.

Définition Une fonction $f : \{0,1\}^* \to \{0,1\}^*$ est *à sens unique* si et seulement si :

- Il existe un algorithme de complexité polynomiale en temps qui sur chaque entrée x produit $f(x)$;
- Pour tout algorithme probabiliste de complexité polynomiale en temps $\mathcal{A}$, il existe une fonction négligeable negl [1] telle que

$$Pr[\textsc{Invert}_{\mathcal{A},f}(n) = 1] \leq \mathsf{negl}(n)$$

3- Démontrer que l'existence de fonctions à sens unique implique que P $\neq$ NP.
4- Soit $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ la fonction $p, q \longrightarrow p \cdot q$. Cette application f est-elle à sens unique ?
N.B. On considérera des bijections $\alpha : \{0,1\}^* \to \mathbb{N} \times \mathbb{N}$ et $\beta : \mathbb{N} \to \{0,1\}^*$, telles que $\alpha, \alpha^{-1}, \beta, \beta^{-1}$ sont calculables en temps polynomial, puis on appliquera la définition ci-dessus à l'application

$$\beta \circ f \circ \alpha : \{0,1\}^* \to \{0,1\}^*.$$

1. Une fonction negl est *négligeable* si pour tout entier d, il existe un entier n_d tel que $\forall n \geq n_d, \mathsf{negl}(n) \leq \frac{1}{n^d}$.