Date Limite : **11h59 28 avril 2021**. Par **équipe de 4**
Rendu : Un seul fichier **pdf uniquement** à déposer dans thor
Fichier préparé de préférence à partir de sources LaTeX. Il pourra contenir des scans/photos de bonne qualité d'un document manuscrit.

Exercice 1 (double-DES)

Exercice 8 de la feuille de TD. Il a été suggéré d'augmenter la sécurité du DES en appliquant deux fois la fonction de chiffrement du DES avec deux clés différentes. Soit $DES_k : \{0,1\}^\ell \to \{0,1\}^\ell$ la fonction de chiffrement par bloc du DES. On note n la taille des clés k. La nouvelle fonction de chiffrement est alors $F_{k_1,k_2} : x \longrightarrow DES_{k_2}(DES_{k_1}(x))$.

Supposons que l'on dispose d'un ensemble de paires textes clairs/textes chiffrés $\{(x_1, y_1), \ldots, (x_a, y_a)\}$ obtenus avec la même clé (k_1, k_2), c'est-à-dire $y_i = F_{k_1,k_2}(x_i), \forall i, 1 \leq i \leq a$.

Q1 Montrer que $\forall i, 1 \leq i \leq a, DES_{k_1}(x_i) = DES^{-1}_{k_2}(y_i)$. Expliquez pourquoi le nombre de clés (k, k') telles que $\forall i, 1 \leq i \leq a, F_{k,k'}(x_i) = y_i$ est approximativement $2^{2n-a\ell}$.

Q2 En supposant que $a \geq \frac{2n}{\ell}$, concevoir une attaque de complexité en temps $O(2^n)$ qui permet de retrouver la clé (k_1, k_2). On pourra pré-calculer deux listes triées. Chaque élément de chacune des liste est un couple (z, k) où k est une clé et z est obtenu par application de DES_k ou DES^{-1}_k. Évaluez la complexité en temps et en espace de votre attaque.

Q3 Que peut-on dire sur la sécurité de double DES ?

Exercice 2 (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 recherché 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, i.e., $\mathcal{M} = \{0,1\}^b$. Soit $f : \mathcal{X} \to \mathcal{Y}$ une fonction. On considère le système de signatures défini ainsi : On note $x[i]$ le ième bit de x.

— $G()$: choisir aléatoirement $2b$ éléments de 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 $pk^1 = [f(x_1^1), \ldots, f(x_b^1)]$.

— $S(sk, m)$. 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.

Q4 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.

Q5 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.

Q6 Un attaquant est capable d'effectuer 100.10^9 calculs de f par seconde. Quelle taille des espaces d'entrée X et de sortie X de la fonction f préconisez-vous ? Justifiez votre réponse.

Q7 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 1) :

— 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.

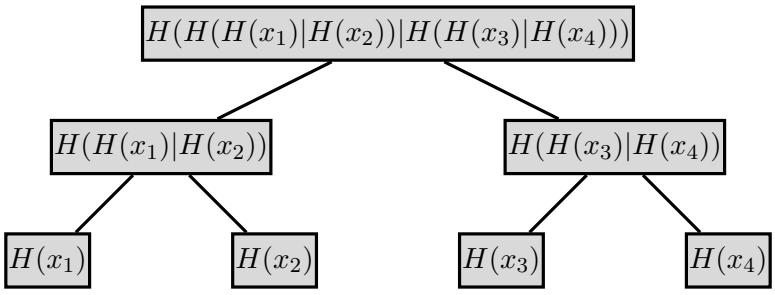

FIGURE 1 – Arbre de Merkle. | dénote la concaténation.

Q8 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 ?

Q9 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 consistent à publier N clés du système à usage unique.).

Q10 Pour quelle(s) raison(s) n'est-il pas faisable pour un attaquant de produire une contrefaçon ?