Initiation à la cryptographie

Travaux Dirigés de cryptographie n°2

—Télécom Nancy 2A ISS—

1 Chiffrement à flot

Dans un chiffrement par flot dit synchrone, le texte chiffré est obtenu en combinant par une opération de groupe (généralement le XOR) le message clair avec une suite secrète, pseudo-aléatoire.

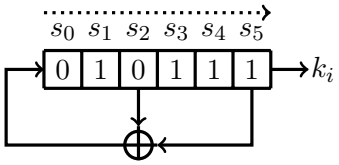
Les algorithmes de chiffrement par flot sont généralement plus rapides que ceux par bloc. Leur utilisation est donc privilégiée dans les applications à fortes contraintes de consommation électrique, comme les téléphones portables.

Les registres à décalage à rétroaction linéaire sont des dispositifs rapides pour créer des suites chiffrantes aux bonnes propriétés matérielles.

▸ **Exercice 1.** Quelques définitions sur les LFSR, et exemple simple

Un registre à décalage à rétroaction (FSR en anglais, comme feedback shift register) est un algorithme permettant d'engendrer une suite vérifiant une relation de récurrence. Les éléments de la suite sont des éléments de $(\mathbb{F}_2)^n$. Un tel registre de longueur ℓ se schématise par un registre de ℓ cellules contenant chacune n bits, et les $\ell \cdot n$ bits forment l'état interne du FSR. Ces cellules sont initialisées avec ℓ mots de n bits, et à chaque unité de temps, chaque mot de n bits est décalé d'une cellule vers la droite. Le contenu de la cellule la plus à droite sort du registre (et sert éventuellement pour un chiffrement) tandis que la cellule la plus à gauche reçoit en entrée un mot obtenu par combinaison du contenu du registre, souvent par XOR bien définis. Si la valeur de ce mot de *rétroaction* est obtenue par une combinaison linéaire des mots du registre, on parle de LFSR (linear FSR).

1. L'état interne à l'étape $i+1$ découle de l'état interne à l'étape i. Pour simplifier, regardons des mots de 1 bit uniquement. Donner les relations de récurrence du LFSR qui correspond au schéma suivant.



2. Calculer les premiers états à partir de 001000, ainsi que les bits produits. Que remarquez-vous ?
3. Toute suite produite par un LFSR est mécaniquement périodique. Quelle est la plus grande période possible de façon générale ? Dans notre cas particulier avec un registre qui comporte $\ell = 6$ cellules et uniquement des mots de 1 bit ? Ce LFSR est-il de période optimale pour un registre à 6 cellules ?
4. On suppose que lors de six itérations successives numérotées N à $N+5$, les bits produits par le LFSR sont, dans l'ordre : $0, 1, 0, 1, 0, 1$. Quel est l'état interne avant l'étape N ? Quels sont les bits suivants ?
5. Il y a plusieurs façons de représenter le fonctionnement d'un LFSR. L'une d'entre elle consiste à donner la matrice M qui permet de passer d'un état interne $v = (x_0^{(i)}, \ldots, x_5^{(i)})$ à l'état interne suivant $v' = (x_0^{(i+1)}, \ldots, x_5^{(i+1)})$, en effectuant le produit matriciel $v' = M \cdot v$. Quelle est ici cette matrice ?

6. Une autre manière très commune et plus concise de décrire un LFSR se fait via un polynôme. En notant $(a_1, \cdots, a_\ell) \in (\mathbb{F}_2)^\ell$ la première colonne de votre matrice, que l'on appelle les *coefficients de rétroaction linéaire*, on peut construire le *polynôme de rétroaction* suivant :

$$f(X) = 1 + \sum_{i=1}^{\ell} a_i X^i \in \mathbb{F}_2[X].$$

   Qui est ici le polynôme de rétroaction de notre LFSR ?
7. Si l'on considère maintenant le LFSR de polynôme de rétroaction $z^5+z^3+z^2+z+1$, comment peut on s'assurer rapidement que la période est nécessairement maximale ?

▸ **Exercice 2.** LFSR et polynôme de rétroaction

Soit $(s_n)_{n\in\mathbb{N}}$ une suite engendrée par un LFSR avec le polynôme de rétroaction de degré ℓ suivant :

$$f(X) = 1 + \sum_{i=1}^{\ell} a_i X^i \in \mathbb{F}_2[X].$$

1. Montrer qu'il existe un polynôme $g(X) \in \mathbb{F}_2[X]$ de degré plus petit strictement que celui de$f$ tel que $s(X) = g(X)/f(X)$.
2. Le *polynôme de rétroaction minimal* d'une suite binaire périodique est le polynôme de rétroaction de plus bas degré parmi les polynômes de rétroaction de tous les LFSR qui génèrent cette suite. Montrer que le polynôme de rétroaction minimal de $(s_n)_{n\in\mathbb{N}}$ est (l'unique) polynôme $f_0(X) \in \mathbb{F}_2[X]$ tel qu'il existe $g_0 \in \mathbb{F}_2[X]$ avec $\deg(g_0) < \deg(f_0)$, $\mathrm{pgcd}(g_0, f_0) = 1$ et $s(X) = g_0(X)/f_0(X)$. Vous pouvez laisser l'unicité en exercice pour plus tard.
3. Nous rappelons qu'un polynôme est dit primitif si l'une de ses racines engendre le groupe multiplicatif de $((\mathbb{F}_2)^\ell)^*$. Montrer que si le polynôme de rétroaction minimal de $(s_n)_{n\in\mathbb{N}}$ est primitif alors la suite est de période maximale.

▸ **Exercice 3.** Malléabilité des chiffrements par flot

Dans cet exercice, nous considérons un chiffrement par flot, noté E, paramétré par une clé secrète K et un vecteur d'initialisation IV.

1. Rappelez le principe général de fonctionnement d'un chiffrement par flot. Étant donnés un message en clair M, une clé K et un vecteur d'initialisation IV, comment le chiffré C est-il obtenu ?

   Supposons qu'Alice ait envie de faire un virement bancaire de 100 euros à Mallory. Pour cela, elle utilise un système de chiffrement par flot E dont seules elle et sa banque connaissent la clé privée K. Alice chiffre donc l'ordre de virement M qu'elle envoie alors à sa banque.

   Mallory est capable d'intercepter et de modifier ce message chiffré C avant que la banque d'Alice ne le reçoive. Elle ne connaît pas M, mais elle sait que les ordres de virement sont des chaînes de caractères de la forme suivante :

$$M = \langle \textit{date} \rangle : \langle \textit{nonce} \rangle : \langle \textit{émetteur} \rangle : \langle \textit{destinataire} \rangle : \langle \textit{montant} \rangle : \langle \textit{commentaire} \rangle$$

   où *nonce* est une chaîne aléatoire de 8 chiffres décimaux, que la banque aura transmise à Alice juste avant que celle-ci ne prépare son ordre de virement.
2. À quoi sert ce *nonce* ?

   Dans le cas d'Alice et Mallory, le message est donc de la forme suivante :

$$M = \texttt{2019-01-28:}\langle \textit{nonce} \rangle\texttt{:Alice:Mallory:100:}\langle \textit{commentaire} \rangle$$

3. Comment Mallory peut-elle faire pour obtenir 999 euros de la part d'Alice ?
4. Quelle contre-mesure est-il possible de mettre en œuvre pour empêcher ce genre d'attaque ?

2 Chiffrement par bloc

▸ **Exercice 4.** Réseaux de Feistel

1. Décrire la fonction inverse d'un tour de réseau de Feistel.
2. Nommons F la fonction interne d'un réseau de Feistel pour $2n = 32 + 32$. Pour une valeur de clé de tour K fixée, la fonction $x \mapsto F(K, x)$ est donc une fonction de 32 bits vers 32 bits. Combien existe-t-il de fonctions distinctes de 32 bits vers 32 bits ?
3. Si on considère l'ensemble des valeurs possibles de la clé de tour (qu'on suppose sur 48 bits), que pouvez-vous dire de la proportion que représentent les fonctions de la forme $x \mapsto F(K, x)$ qui peuvent être ainsi construites, parmi toutes les fonctions de 32 bits vers 32 bits ?
4. Considérons un réseau de Feistel à 16 tours, avec une clé maîtresse de 56 bits. Montrer que pour une valeur donnée de la clé maîtresse, la fonction de chiffrement est une permutation d'un ensemble à 2^{64} éléments. Combien de telles permutations existent ? Que pouvez-vous dire de l'ensemble des permutations qui peuvent être construites grâce à cette structure de réseau de Feistel, en faisant varier les clés ? Quelle est leur proportion ?

▸ **Exercice 5.** Chiffrement par bloc avec peu de bits.

On considère un chiffrement par bloc E_K, avec une petite taille de bloc n.

1. Pour $n = 1$, $n = 2$, $n = 3$, $n = 4$, donner une borne sur le nombre de fonctions E_K possibles.
2. À partir de quelle valeur de la taille de bloc devient-il impossible de faire l'énumération exhaustive de l'ensemble des possibilités pour E_K, dans la perspective d'une attaque ?

▸ **Exercice 6.** Amélioration d'un système de chiffrement : le DES

Monsieur X utilise pour chiffrer ses données privées le cryptosystème DES, paramétré par une clé secrète k de 56 bits connue de lui seul. Comme Monsieur X a entendu dire que 56 bits étaient bien peu de nos jours, il envisage de rendre plus sûr le stockage de ses données en chiffrant une seconde fois toutes ses données, avec la clé DES $k' = k + 1$ (pour chaque donnée en clair m, la donnée chiffrée est donc $c = \mathrm{DES}_{k+1}(\mathrm{DES}_k(m))$, où k désigne la clé).

1. Est-ce une bonne idée ?
2. Discuter les avantages et/ou les inconvénients.
3. Monsieur X pense à une autre amélioration possible. Il va chiffrer une fois avec DES, et une fois avec AES128. Comme AES128 a besoin de clés de 128 bits, il va paramétrer son chiffrement DES par sa clé secrète k, et pour son chiffrement AES128 la même clé secrète k, mais avec des zéros pour faire le remplissage. Est-ce mieux ?
4. Quelle erreur fondamentale Monsieur X commet-il, eu égard aux principes de Kerckhoffs ?

3 Clé secrète

▸ **Exercice 7.** Authentification de type défi–réponse

Il existe des protocoles permettant d'authentifier une entité A auprès d'une entité B. Cela présuppose donc que A sache effectivement que l'entité vérificatrice est bien B, et pas un attaquant C qui se fait passer pour B. Or lors de la plupart des connexions, rien ne l'en assure. Il faudrait alors que B s'authentifie également auprès de A. C'est ce qu'on appelle l'*authentification mutuelle*.

L'idée générale est alors de reprendre les protocoles qui existent pour l'authentification d'une entité et de l'appliquer de manière symétrique pour authentifier B auprès de A. Nous allons voir sur deux exemples qu'il est tout de même nécessaire de prévoir quelques ajustements.

1. Expliquer pourquoi il n'est pas possible de faire de l'authentification mutuelle par mot de passe.
2. Suggérer une situation dans lesquelles une interception de mot de passe est possible en l'absence d'authentification du serveur.

   On cherche maintenant à utiliser une authentification de type défi–réponse utilisant un système à clé secrète. Considérons le protocole suivant qui utilise un chiffrement à clé secrète. A et B partagent au préalable une clé secrète K.

   (*i*) A tire une valeur aléatoire r_A et l'envoie à B ;

   (*ii*) B tire une valeur aléatoire r_B et calcule $\beta = E_K(r_A, r_B)$. B envoie β à A ;

   (*iii*) A calcule $D_K(\beta)$. S'il n'y a pas eu d'attaque, il retrouve r_A : B s'est authentifié.
   A prend connaissance de r_B. Il envoie r_B à B : A s'est authentifié.
3. Trouver une attaque de ce protocole par rejeu. On donne les éléments de départ de l'approche. Le participant A est honnête, et l'attaquant C (malhonnête !) se fait passer pour B. C va, parallèlement à la tentative d'authentification mutuelle émanant de A (vers B, pense-t-il) appelée « session 1 », initier une session d'authentification vers A (en faisant croire qu'elle émane de B), qu'on appellera « session 2 ». Les messages de ces deux sessions s'entrelacent. Les premières étapes sont (exactement dans cet ordre) :
   — (session 1) : A envoie r_A à C.
   — (session 2) : C envoie r_A à A.
   Compléter, et expliquer d'où provient le problème.
4. Suggérer une amélioration.