CRTYPA

Ludovic Perret

Sorbonne Universités, UPMC Univ Paris 06, INRIA Paris
LIP6, PolSyS Project, Paris, France

Abstract.

1 Rappels

1.1 Chiffrement à clé secrète

On rappelle ici quelques outils permettant de garantir la **confidentialité** d'une donnée.

1.2 Chiffrement par flot

Definition 1. *Un* **chiffrement par flot** *est un chiffrement à clef secrète. Il est constitué d'un premier algorithme* $\mathrm{SC} : \mathbb{F}_2^t \to \{0,1\}^n$ *qui prend en entrée une clé secrète* $K \in \mathbb{F}_2^k$*. La fonction* SC_K *permet de générer une* **suite chiffrante** *à partir de la clef secrète* K*. On chiffre alors une message* $m \in \{0,1\}^n$ *par:*

$$c = \mathrm{SC}(K) \oplus m.$$

Pour déchiffrer, on calcule:

$$m = \mathrm{SC}(K) \oplus c.$$

Example 1. RC4 est un exemple célèbre de chiffrement par flot. **Il ne faut surtout pas utiliser RC4 en pratique.**

1.3 Chiffrement par bloc

Definition 2. *Un* **chiffrement par bloc** *est un chiffrement à clef secrète. C'est une fonction* $E_K : \mathbb{F}_2^n \to \mathbb{F}_2^n$*, paramétrée par une clef secrète* $K \in \mathbb{F}_2^k$*, qui opère sur un bloc de taille fixe. On associe à* E_K *une fonction de déchiffrement* $D_K : \mathbb{F}_2^n \to \mathbb{F}_2^n$ *telle que:*

$$D_K\big(E_K(m)\big) = m, \quad \forall m \in \mathbb{F}_2^n.$$

Ainsi, pour chiffrer un message $m \in \mathbb{F}_2^n$*, on calcule:*

$$c = E_K(m) \in \mathbb{F}_2^n.$$

Pour déchiffrer $c \in \mathbb{F}_2^n$*, on fait:*

$$m = D_K(m) \in \mathbb{F}_2^n.$$

Example 2. En chiffrement par bloc, le standard est AES128 dans lequel $n = 128$ (taille du bloc) et $t = 128$ (taille de la clé secrète).

Definition 3. *Un* **mode opératoire** *pour un chiffrement par bloc est une algorithme dont l'objectif est de chiffrer un message de taille quelconque* $m \in \{0,1\}^*$ *avec un chiffrement par bloc* $E_K : \mathbb{F}_2^n \to \mathbb{F}_2^n$*. Le principe est de découper le message en des blocs de taille* n *et d'utiliser* E_K *sur chaque bloc.*

Example 3. Soit $m = (m_1, \ldots, m_t) \in (\mathbb{F}_2^n)^t$.

- Un mode opératoire simple est le mode `ECB` qui consiste à chiffrer chaque bloc du message m indépendamment. C'est à dire, on calcule:
$$c_i = E_K(m_i), \forall i, 1 \le i \le t.$$
On déchiffre par:
$$m_i = D_K(c_i), \forall i, 1 \le i \le t.$$
Il ne faut surtout pas utiliser `ECB` en pratique.
- Le mode `CBC` fonctionne de la manière suivante. Nous avons un vecteur public d'initialisation $c_0 = \mathrm{IV} \in \mathbb{F}_2^n$. On chiffre comme:
$$c_i = E_K(m_i \oplus c_{i-1}), \forall i, 1 \le i \le t.$$
Pour le déchiffrement, nous avons:
$$m_i = D_K(c_i) \oplus c_{i-1}, \forall i, 1 \le i \le t.$$
- Le mode `CTR` fonctionne de la manière suivante. Nous avons un vecteur public d'initialisation $\mathrm{IV} \in \mathbb{F}_2^n$. On chiffre comme:
$$c_i = m_i \oplus E_K(\mathrm{IV} \oplus i), \forall i, 1 \le i \le t.$$

1.4 Hachage

Definition 4. *Une* **fonction de hachage** *est une fonction qui prend comme entrée une donnée de taille quelconque et retourne une* **emprunte** *de taille fixe. Autrement dit, une fonction de hachage est une fonction de la forme $H : \{0,1\}^* \to \mathbb{F}_2^n$. Une* **fonction de hachage cryptographique** *est une fonction de hachage $H : \{0,1\}^* \to \mathbb{F}_2^n$ telle que:*

- *H est facilement évaluable, i.e. $\forall D \in \{0,1\}^*$, $H(D)$ est calculable en temps polynomial.*
- *H est* **résistante à la pré-image***, i.e. $\forall h \in \mathbb{F}_2^n$, il est difficile de trouver $D \in \{0,1\}^*$ tel que $H(D) = h$.*
- *H est* **résistante à la seconde pré-image***, i.e. $\forall D \in \{0,1\}^*$ fixé, il est difficile de trouver $D' \in \{0,1\}^*$ tels que*
$$H(D) = H(D') \text{ et } D \neq D'.$$
- *H est* **résistante à la collision***, i.e. il est difficile de trouver un couple $(D, D') \in \{0,1\}^*$ tels que*
$$H(D) = H(D') \text{ et } D \neq D'.$$

Remark 1. Soit $H : \{0,1\}^* \to \mathbb{F}_2^n$ une fonction de hachage. Le **paradoxe des anniversaires** permet de trouver une collision en $O(2^{n/2})$ évaluations de H avec une forte probabilité.

Remark 2.
- `MD5` fonction de hachage dans laquelle $n = 128$. **Il ne faut surtout pas utiliser `MD5` en pratique.** On trouve, par exemple, des collisions dans `MD5` (quasiment) en temps réel.
- `SHA1` fonction de hachage dans laquelle $n = 160$. **Il ne faut surtout pas utiliser `SHA1` en pratique.** Google, en collaboration avec de nombreux chercheurs, a annoncé le calcul d'une collision pour `SHA1`. Cette collisions a nécessité de l'ordre de $2^{63.1}$ évaluations de `SHA1`.
- On peut utiliser les fonctions de hachage la famille `SHA2` (`SHA256`, `SHA384`, ou `SHA512`)
- Le nouveau standard est `SHA3`.

1.5 Authentification à clé secrète

Definition 5. *Un* **Message Authentication Code (MAC)** *est une fonction $\mathrm{MAC}_K : \{0,1\}^* \to \mathbb{F}_2^n$ qui est paramétrée par une clef secrète $K \in \mathbb{F}_2^k$. Elle prend en entrée une donnée de taille quelconque et retourne un authentifiant de taille fixe. Dans ce modèle, l'émetteur et le destinataire partagent une clef secrète $K \in \mathbb{F}_2^t$. Ainsi, on associe à une donnée $D \in \{0,1\}^*$ un authentifiant $T = \mathbb{F}_2^n$. L'authentifiant T est ainsi envoyé avec la donnée D.*