Théorie de l'information

D. Kohel

08/02/2021

Cryptographie symétrique

- fonctions booléenes (Boole)

$$\mathbb{F}_2^n \xrightarrow{E_K} \mathbb{F}_2^n \xrightarrow{\pi_i} \mathbb{F}_2$$

fonction booléen.

- parfaitement sûr ?

Cryptographie à clé publique

- basée sur des problèmes difficiles
  - problème = problème computationnel, dont sa résolution est un algorithme
  - difficile = caractérisation de sa complexité (du meilleur algorithme connu)
    ≠ temps polynôme
    = temps exponentiel.
- On a vu que la clé publique détermine l'application

$E_K : \mathcal{M} \longrightarrow \mathcal{C}$, ainsi que son inverse. Il suit que

$$H(\mathcal{M}|\mathcal{C}) = H(\mathcal{C}|\mathcal{M}) = 0$$
$$I(\mathcal{M}, \mathcal{C}) = H(\mathcal{M}) = H(\mathcal{C}).$$

Le plus loin possible d'être parfaitement sûr.

— Par conséquant, la securité reste entièrement sur la difficulté du problème computationnel.

Cryptographie pré-quantique (1976 – 1994)

RSA : basé sur le problème de factorisation des entiers — Diffie et Hellman — Shor

(= attaque contre le problème RSA dans $(\mathbb{Z}/N\mathbb{Z})^*$)

ElGamal: basé sur la difficulté du problème du log discret dans $\mathbb{F}_p^*$ ($\Rightarrow$ attaque contre le problème de Diffie et Hellman).

ECC (Cryptographie à base des courbes elliptique): basé sur la difficulté sur la difficulté du problème de log discret sur une courbe elliptique

— Pour RSA et ElGamal on a des algorithmes (classiques) qui sont sous-exponentiels.

— Pour ECC, les meilleurs algorithmes connus sont pleinement exponentiels.

Cryptographie post-quantique

Algorithme de Shor (1994) permet de trouver une période d'une fonction

$$f : \mathbb{Z}^n \longrightarrow X$$
$$(x_1, \dots, x_n) \longmapsto f(x_1, \dots, x_n).$$

Un période est un élément $(a_1, \dots, a_n)$ tel que

$$f(x_1 + a_1, \dots, x_n + a_n) = f(x_1, \dots, x_n)$$

pour tout $(x_1, \dots, x_n) \in \mathbb{Z}^n$.

L'algorithme de Shor permet de résoudre le problème de factorisation ainsi que les logarithmes discrets.

On suppose que la sortie de l'algorithme est une période aléatoire. Deux itérations

donne des solutions indépendents.

Problème de factorisation

Soit N un entier impair composé.
Soit $a \in (\mathbb{Z}/N\mathbb{Z})^*$ un élément aléatoire.

$$f : \mathbb{Z} \longrightarrow (\mathbb{Z}/N\mathbb{Z})^*$$
$$x \longmapsto a^x = f(x)$$

Soit w une période (telle que $f(x+w)=f(x)$), produit par l'algorithme de Shor.
Alors $a^w = 1$ dans $(\mathbb{Z}/N\mathbb{Z})^*$.
Si w est pair, $w = 2r$, on pose $u = a^r$, avec $u^2 = 1$.
Si $u \neq \pm 1$ on obtient la factorisation

$$\text{pgcd}(u-1, N)\,\text{pgcd}(u+1, N) = N.$$

Si $u=1$ et r est pair, on remplace r avec $r/2$, et u avec $a^{r'} = r'$

Si $u=-1$ (ou $u=1$ et r impair) on choisit un autre a et on repète l'algorithme de Shor.

<u>Problème du log discret</u>

Soit $G=\langle a \rangle$ un groupe cyclique et $b=a^k$; $|G|=q$ premier. On définit la fonction $f: \mathbb{Z}^2 \longrightarrow G$ par

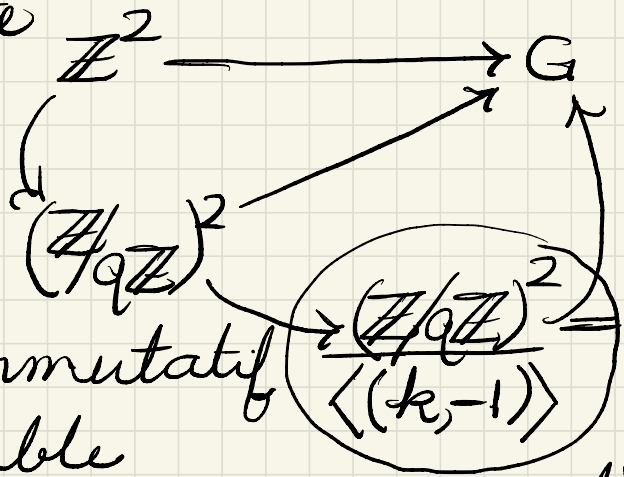
$$(x_1, x_2) \longmapsto a^{x_1} b^{x_2}$$

Si $f(x_1+w_1, x_2+w_2) = f(x_1, x_2)$ alors

$$1 = f(w_1, w_2) = a^{w_1} b^{w_2},$$

et si $w_2 \neq 0 \pmod q$, on a $k = -w_1/w_2 \pmod q$ est $\log_a(b)$.

On note que :



est commutatif

L'ensemble des périodes $\Omega \subseteq \mathbb{Z}^2$ est un groupe d'ordre q

est un sous-groupe d'indice q avec

$$(q\mathbb{Z})^2 \subseteq \Omega \subseteq \mathbb{Z}^2$$

(q au-dessus de $(q\mathbb{Z})^2$ et $\mathbb{Z}^2$, q sous $(q\mathbb{Z})^2$, q sous Ω)

Trois axes pour des cryptosystèmes post-quantiques

- à base de codes (McEliece)
- à base de réseaux (NTRU)
- à base de graphes d'isogénies de courbes elliptiques. (supersingulières)

McEliece

- Pas très bien accepté comme protocole pré-quantique (trop lente trop grande taille de clé, par rapport à RSA, ElGamal ou ECC)
- Candidat post-quantique
- Basé sur la difficulté (computionnelle) de décodage d'un code linéaire général.
- cryptosystèmes à clé publique.

Cryptosystème de McEliece

Génération de clés (Alice)

- Choix de code $C \subseteq \mathbb{F}_2^n$, $k = \dim_{\mathbb{F}_2} C$ t-correcteur ($d(C) = d \geq 2t+1$), avec un algorithme efficace pour décodage : $D : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^k$ tel que $D(c+e) = D(c)$ pour tout vecteur e (d'erreur) de poids $\leq t$.
- Soit G une matrice génératrice de C dans $M_{k \times n}(\mathbb{F}_2)$, dont les lignes forment une base de C.
- Choix de $S \in GL_k(\mathbb{F}_2)$ et $P \in O_n(\mathbb{F}_2)$, matrice de permutation (pour laquelle il existe un 1 par ligne / colonne, zéro sinon)

- Alice pose $\hat{G} = SGP$.
- La clé publique est $(\hat{G}, t)$.
- La clé privée est (S, G, P, D).

Chiffrement (Bob)

Soit $m \in \mathbb{F}_2^k$ (un message à chiffrer).

Bob calcule $c' = m\hat{G} \in \mathbb{F}_2^n$.

Bob choisit un vecteur $e \in \mathbb{F}_2^n$, vecteur d'erreur, de poids t.

N.B. Il y a $\binom{n}{t}$ tels vecteurs.

Le texte chiffré est $c = c' + e$.

N.B. c' est un mot de code de $\hat{C}$ (engendré par $\hat{G}$), toujours t-correcteur d'erreur.

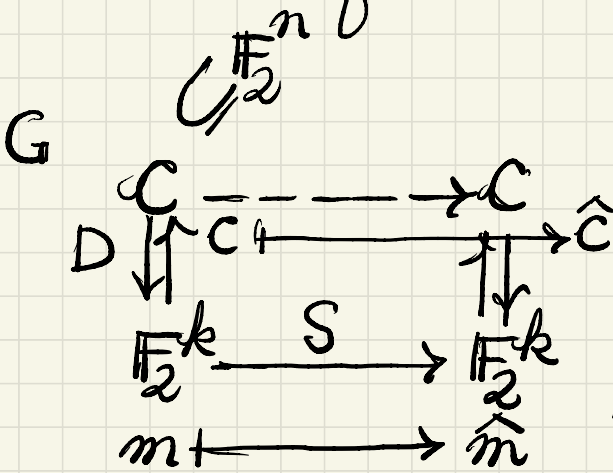
Déchiffrement (Alice)

Alice calcule $\hat{c} = cP^t$ (N.B. $P^t = P^{-1}$)

Alice décode $D(\hat{c}) = \hat{m} \in \mathbb{F}_2^k$.

Alice calcule $m = \hat{m}S^{-1}$.

On note que :



SG = une autre matrice génératrice pour C

La matrice de permutation P donne un nouveau code

$$C \xrightarrow{P} \hat{C}$$
$$\hat{c} \longmapsto c'$$

avec matrice génératrice G pour $\hat{C}$

<u>Preuve</u> On note $\hat{c} = cP^t$

(que le déchiffrement donne m)

$$= (m\hat{G})P^t + eP^t$$
$$= (mSG)PP^t + \hat{e}$$

où $\hat{e}$ est un vecteur de poids $\leq t$.

$$D(\hat{c}) = D(\hat{m}G + \hat{e}) = \hat{m}G + \hat{e}$$
$$= D(\hat{c} + \hat{e}) = D(\hat{c}) = \hat{m} \ (= mS).$$

Alors $D(\hat{c})S^{-1} = \hat{m}S^{-1} = m$. □

Taille des clés

McEliece (1978) : $n=1024$, $k=524$, $t=50$

Alors que la matrice $\hat{G}$ aura $524(1024) \sim 525$ kb

N.B. Diffie-Hellman 1976
RSA 1977
ECC – Koblitz, Miller 1977
ElGamal 1986

Actuellement (2021) :

NIST Post-quantum encryption competition

- NTS-KEM (à base de codes)

$$[n,k,t] \in \{[6688,128,13], [6960,119,13], [8192,128,13]\}$$

$[n,k,t] = [6960, 5413, 119]$

Ces paramètres ont été jugés suffisants pour la sécurités par des auteurs.

cf. Daniel Augot, Initial recommendations of long-term secure post-quantum systems, 2015.