Introduction à la cryptographie Post-Quantique

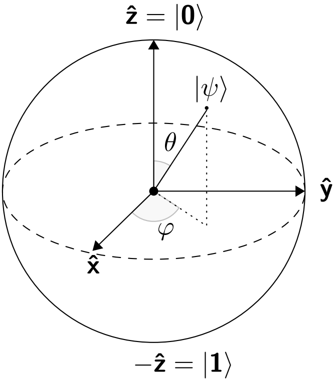
Pascal Lafourcade

Séminaire 27 mars 2025

Qbit

$$|\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \text{ avec } (\alpha, \beta) \in \mathbb{C}, \text{ tel que } \alpha\,|0\rangle + \beta\,|1\rangle = 1$$

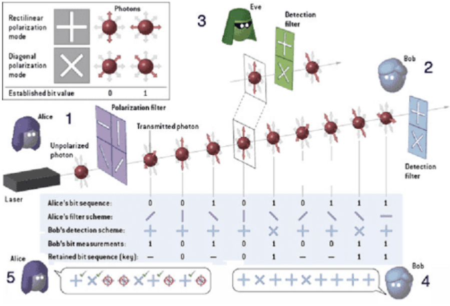
$$||\psi||^2 = |\alpha|^2 + |\beta|^2 = \alpha.\overline{\alpha} + \beta.\overline{\beta} = 1$$



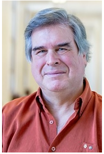
BB84

Théorème (Non-clonage (Wooters et Zurek, 1982))

Il est impossible de copier parfaitement un qubit dont l'état quantique est inconnu.



Charles Bennett



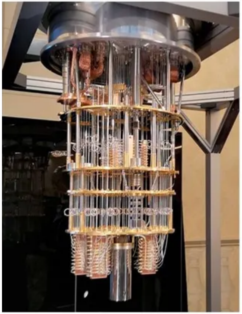
Gilles Brassard

Ordinateurs quantiques

- 1998 : 2 qbits, IBM
- 1999 : 3 qbits, IBM
- 2001 : 7 qbits, IBM
- 2017 : 50 qbits, IBM Q50
- 2019 : 53 qbits, Google Sycamore
- 2021 : 90 qbits, Rigetti Aspen-9
- 2021 : 127 qbits, IBM Eagle
- 2022 : 433 qbits, IBM Osprey
- Dec 2023 : 1 121 qubits, IBM Condor

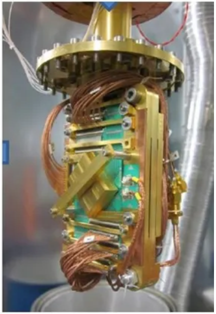
- 2011 : 128 qbits, One
- 2013 : 512 qbits, Two
- 2015 : 1152 qbits, 2X
- 2017 : 2048 qbits, 2000Q
- 2020 : 5760 qbits, Advantage
- 2024 : 7440 qbits, Advnatage2

Ordinateurs quantiques



IBM

rigetti



D:WAVE

Portes quantiques

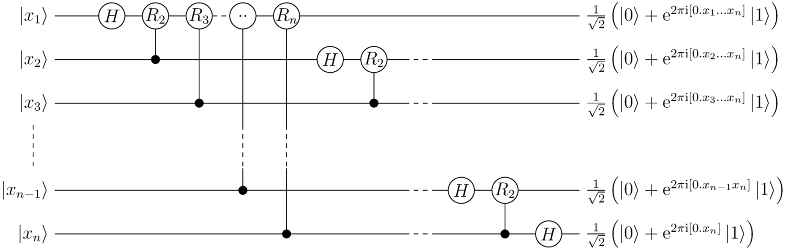
$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Circuits quantiques



Transformée de Fourrier quantique

Algorithmes quantiques

- Algorithme de Deutsch (1985) et Deutsch-Jozsa (1992)
- Algorithme de Simon (1994)
- Algorithme de Shor (1994)
- Algorithme de Grover (1996)

Shor et Grover

Algorithme de Shor (1994)

Calcule l'ordre d'un nombre en temps polynomial.

Définition de l'ordre

L'ordre de a est le plus petit entier r tel que $a^r \equiv 1 \mod N$

Algorithme de Grover (1996)

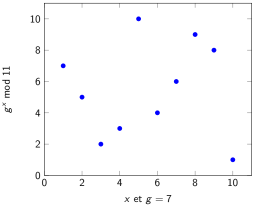
Trouver efficacement un élément qui satisfait une propriété dans une liste donnée.

Plan

Cryptographie Pré-quantique

Deux problèmes :

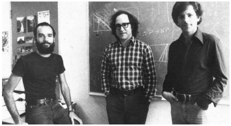
- Factorisation : $n = p \times q$ difficile de trouver p et q.
- Logarithme disctet : $g, p, g^x \bmod p$ difficile de trouver x.

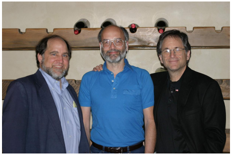


C'est deux problèmes sont cassés par l'algorithmes de Shor !

"**store-now, decrypt-later**"

Rivest Shamir Adelmann (RSA 1978)





Soit $n = pq$, p et q deux nombres premiers.

Clé Publique : (e, n)

Clé Secrète : d où $d = e^{-1} \mod \phi(n)$
et $pgcd(e, \phi(n)) = 1$

Chiffrement : $c = m^e \mod n$

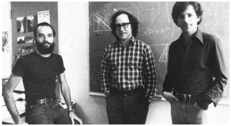
Déchiffrement: $m = c^d \mod n$

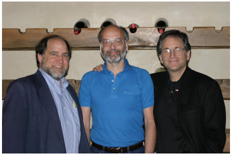
Correction

$c^d = m^{de} = m.m^{k\phi(n)} \mod n$

Rappel : Théorème d'Euler $\forall x \in (\mathbb{Z}/n\mathbb{Z})^*, x^{\phi(n)} = 1 \mod n$

Rivest Shamir Adelmann (RSA 1978)





Soit $n = pq$, p et$q$ deux nombres premiers.

Clé Publique : (e, n)

Clé Secrète : d où $d = e^{-1} \mod \phi(n)$
et $pgcd(e, \phi(n)) = 1$

Chiffrement : $c = m^e \mod n$

Déchiffrement: $m = c^d \mod n$

Correction

$c^d = m^{de} = m.m^{k\phi(n)} \mod n$

Rappel : Théorème d'Euler $\forall x \in (\mathbb{Z}/n\mathbb{Z})^*, x^{\phi(n)} = 1 \mod n$

Casser la factorisation permet de casser RSA !

Algorithme de factorisation (Shor), $N = pq$

- Choisir $1 < a < N$ au hasard.
- Si $d = pgcd(N, a) \neq 1$ alors d est un facteur de N
- Sinon $pgcd(N, a) = 1$ alors a est inversible modulo N, cad $\exists k, a^k \equiv 1 \bmod N$ (Euler)
- Calcule l'ordre de a cad : La période de $F_a(x) = a^x \bmod N$ est le plus petit entier w tel que $F_a(x) = F_a(x + w)$

$$\begin{aligned} a^{x+w} &\equiv a^x \bmod N \\ a^{x+w} a^{-x} &\equiv 1 \bmod N \\ a^w - 1 &\equiv 0 \bmod N \end{aligned}$$

- Si w est impair ou $a^{w/2} \not\equiv -1 \bmod N$, l'algorithme tire un nouveau a et réitère les différentes étapes.
- Sinon $d = pgcd(a^{w/2} - 1, N) \neq 1$ ou $d' = pgcd(a^{w/2} + 1, N) \neq 1$ d ou d' donne un facteur non-trivial de N

$\square$

Détails de l'algorithme de factorisation (Shor)

L'ordre w de a est pair et $a^{w/2} \equiv -1 \bmod N$.

- Comme w est pair cela permet d'obtenir $a^w - 1 \equiv (a^{w/2} - 1)(a^{w/2} + 1) \equiv 0 \bmod N$, car $x^2 - y^2 = (x - y)(x + y)$.
  Ainsi pour tout $q \in \mathbb{N}$, $(a^{w/2} - 1)(a^{w/2} + 1) = qN$.
  Ce qui signifie que N divise $(a^{w/2} - 1)(a^{w/2} + 1)$.
- Par définition de w, il s'agit du plus petit entier tel que $a^w \equiv 1 \bmod N$, comme $w/2 < w$,
  Il en découle que $a^{w/2} \not\equiv 1 \bmod N$
  Donc $a^{w/2} - 1 \not\equiv 0 \bmod N$.
  Ainsi $d = pgcd(a^{w/2} - 1, N)$ est un facteur non-trivial de N
- $a^{w/2} \equiv -1 \bmod N$ signifie que $a^{w/2} + 1 \equiv 0 \bmod N$, cad N divise $a^{w/2} + 1$.
  Donc $d' = pgcd(a^{w/2} + 1, N)$ donne un facteur d' non-trivial de N.

Exemple $N = 15$

$a = 2$
$pgcd(2, 15) = 1$, donc $w = 4$ car $2^4 = 16 = 1 \bmod 15$, donc
$d = pgcd(a^{w/2} - 1, N) = pgcd(3, 15) = 5$
$d' = pgcd(a^{w/2} + 1, N) = pgcd(5, 15) = 5$

$a = 3$
$pgcd(3, 15) \neq 1$ donc 3 divise 15

$a = 11$
$pgcd(11, 15) = 1$, donc $w = 2$ car $11^2 = 121 = 15 * 8 + 1 = 1 \bmod 15$, donc
$d = pgcd(a^{w/2} - 1, N) = pgcd(10, 15) = 5$
$d' = pgcd(a^{w/2} + 1, N) = pgcd(12, 15) = 3$

$a = 13$
$pgcd(13, 15) = 1$, donc $w = 4$ car
$13^4 = 28561 = 1804 * 15 + 1 = 1 \bmod 15$, donc
$d = pgcd(a^{w/2} - 1, N) = pgcd(168, 15) = 3$
$d' = pgcd(a^{w/2} + 1, N) = pgcd(170, 15) = 5$

Logarithme discret et Shor

Problème du logarithme discret

Retrouver s à partir de $y \equiv g^s \bmod p$ avec $0 \leq s < q$.

Appliquer Shor à $F_{g,y}$:

$$\begin{array}{rcl} F_{g,y} : \mathbb{Z}_q \times \mathbb{Z}_q & \to & G \\ (x, x') & \to & g^x y^{x'} \bmod p \end{array}$$

L'algorithme de Shor donne $(w, w') \in \mathbb{Z}_q \times \mathbb{Z}_q$ une période de cette fonction $F_{g,y}$, soit $F_{g,y}(x + w, x' + w') = F_{g,y}(x, x')$

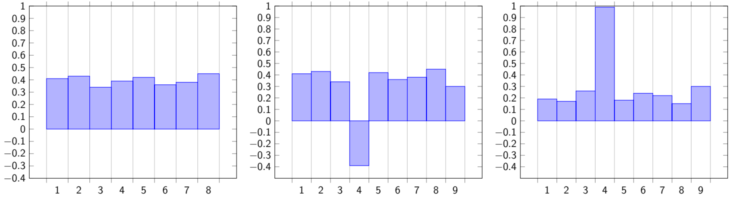
$$\begin{array}{rcl} g^x y^{x'} & \equiv & g^{w+x} y^{w'+x'} \bmod p \\ g^w y^{w'} & \equiv & 1 \bmod p \\ g^{w+sw'} & \equiv & 1 \bmod p \end{array}$$

Ainsi $w + sw' \equiv 0 \bmod q$

$s \equiv -w \cdot (w')^{-1} \bmod q$, si w' est inversible modulo qs.

Grover 1996

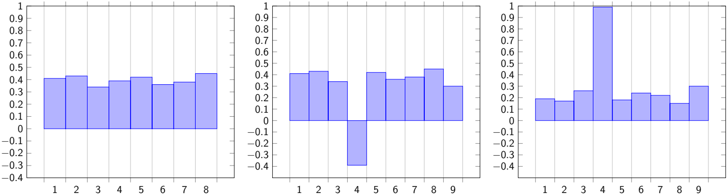
Trouve $x \in \{0,1\}^n$ avec $F(x)$ en $\sqrt{2^n}$ évaluations de F



Oracle quantique qui détermine x

Grover 1996

Trouve $x \in \{0,1\}^n$ avec $F(x)$ en $\sqrt{2^n}$ évaluations de F



Oracle quantique qui détermine x

Diminue légèrement la sécurité pour :

- les fonctions de hachages de $O(2^{\frac{N}{2}})$ à $O(2^{\frac{N}{3}})$
- les chiffrements symmétriques de $O(2^n)$ à $O(2^{\frac{n}{2}})$

Plan

Cryptographie Post-Quantique

- Fonctionne sur les ordinateurs classiques

- Résite à un ordinateur quantique

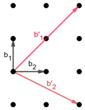
Les problèmes difficiles sous-jacents sont différents !

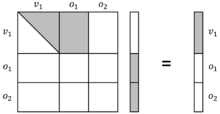
Compétition du NIST lancée en 2017

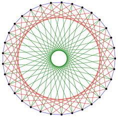
- 30 novembre 2017 : 69 sousmissions Round 1
- 30 janvier 2019 : 26 sousmissions choisies pour le Round 2
- 22 juillet 2020 : 7+8 sousmissions choisies pour le Round 3
- 5 juillet, 2022 :
  - KEM : Kyber
  - Signature : Dilithium, Falcon, SPHINCS+
- 13 août 2024, NIST publie les standards :
  - FIPS 203 (Kyber),
  - FIPS 204 (Dilithium)
  - FIPS 205 (SPHINCS+)
  - FIPS 206 (FALCON à venir)
- 10 mars 2025, NIST annonce le vainqueur du Round 4 : KEM HQC

5 familles de problèmes difficiles

- Fonctions de hachage
- Réseaux Euclidiens (Lattices)
- Codes
- Systèmes Multivariés
- Isogénies

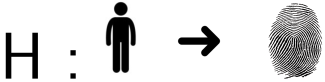




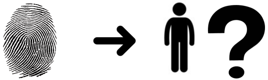
Plan

Fonctions de hachage (SHA-1, SHA-3)

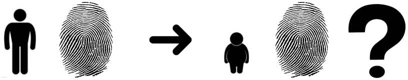


Properties

- First Pré-image

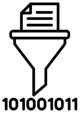


- Second Pré-image



- Collision

Signature de Lamport, 1979



Génération de clés

- $\forall i, i = 1...k, \forall b \in \{0,1\}$, choisir $x_{i,b}$
- Calculer $y_{i,b} = H(x_{i,b})$
- $sk = (x_{i,b})_{i,b}$
- $pk = (y_{i,b})_{i,b}$

$k = 4$

$$sk = \begin{cases} x_{1,0}, x_{2,0}, x_{3,0}, x_{4,0} \\ x_{1,1}, x_{2,1}, x_{3,1}, x_{4,1} \end{cases}$$

$$pk = \begin{cases} y_{1,0}, y_{2,0}, y_{3,0}, y_{4,0} \\ y_{1,1}, y_{2,1}, y_{3,1}, y_{4,1} \end{cases}$$

Signature de $m = m_1...m_k$ avec sk

- $\forall i, i = 1...k, \sigma_i = x_{i,m_i}$
- $\sigma = (\sigma_i)_i$

$m = 0110$

$\sigma = (x_{1,0}, x_{2,1}, x_{3,1}, x_{4,0})$

Vérification avec pk

Vérifier si $\forall i, H(\sigma_i) = y_{i,m_i}$

$H(\sigma) = (y_{1,0}, y_{2,1}, y_{3,1}, y_{4,0})$

Signature de Winternitz, 1989

Génération de clés

- w est un paramètre choisi par l'utilisateur, $l = \frac{k}{w}$, $B = 1 + \left\lceil \frac{log_2 l}{w} \right\rceil$
- Construire $l + B$ chaînes
  Chaque chaîne part de y_i^0 et finit par $y_i^{2^l-1} = H^{2^l-1}(y_i^0) = z_i$.
- Clé publique : $z = h(z_1||z_2||\ldots||z_{l+B})$
- Clé secrète : $(y_i^0)_{0\leq i\leq l+B}$

Signature de $(x_1, \ldots, x_k)$

- Calculer $C = (x_{k+1}, \ldots, x_{k+B}) = \sum_{i=1}^{l}(2^w - 1 - x_i)$
- Calculer $a_i = H^{x_i}(y_i)$, pour $1 \leq i \leq l + B$
- La signature $sig_k(x_1, \ldots, x_k) = (a_1, \ldots, a_{l+B})$

Vérification

- Calculer $C = (x_{k+1}, \ldots, x_{k+B}) = \sum_{i=1}^{l}(2^w - 1 - x_i)$
- Pour $1 \leq i \leq l + B$ calculer $a_i = H^{x_i}(y_i)$
- Vérifier que $z = h(z_1||z_2||\ldots||z_{l+B})$

Exemple, $k = 9$, $w = 3$, donc $l = 3$

$B = 1 + \left\lceil \frac{log_2 l}{w} \right\rceil = 2$

- Clé secrète : $(y_i^0)_{0 \le i \le l+B}$
- Clé publique : $z = H(z_1||z_2||z_3||z_4||z_5)$

Signature de x : $\sigma = (a_1, a_2, a_3, a_4, a_5)$

- $x = 011\,101\,001$, donc $x_1 = 011 = 3$, $x_2 = 101 = 5$, et $x_3 = 001 = 1$

$$
\begin{array}{l}
y_1^0 \to y_1^1 \to y_1^2 \to y_1^3 \to y_1^4 \to y_1^5 \to y_1^6 \to y_1^7 = z_1 \\
y_2^0 \to y_2^1 \to y_2^2 \to y_2^3 \to y_2^4 \to y_2^5 \to y_2^6 \to y_2^7 = z_2 \\
y_3^0 \to y_3^1 \to y_3^2 \to y_3^3 \to y_3^4 \to y_3^5 \to y_3^6 \to y_3^7 = z_3
\end{array}
$$

- $C = (7-3) + (7-5) + (7-1) = 4 + 2 + 6 = 12$ donc $C = 001\,100$

$$
\begin{array}{l}
y_4^0 \to y_4^1 \to y_4^2 \to y_4^3 \to y_4^4 \to y_4^5 \to y_4^6 \to y_4^7 = z_4 \\
y_5^0 \to y_5^1 \to y_5^2 \to y_5^3 \to y_5^4 \to y_5^5 \to y_5^6 \to y_5^7 = z_5
\end{array}
$$

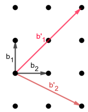
- $\sigma = (y_1^3, y_2^5, y_3^1, y_4^1, y_5^5)$

Vérification de $x = (x_1, x_2, x_3)$ et $\sigma = (a_1, a_2, a_3, a_4, a_5)$ avec z

- Calcul de C pour la création de x_3 et x_5
- Vérification de (a_4, a_5)
- Vérification de $z = h(z_1||z_2||\ldots||z_5)$

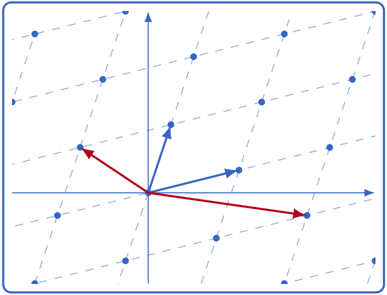
Plan

Réseaux Euclidiens



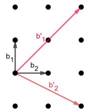
Definition

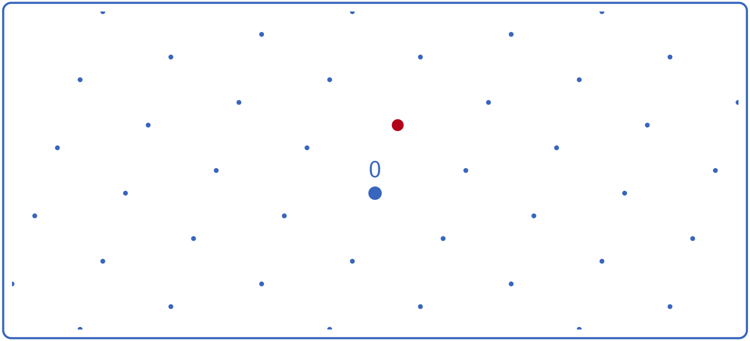
Un *réseau* est un sous-groupe discret $\mathcal{L}$ de $\mathbb{R}^n$ où n un entier positif.



- Tour ensemble B de vecteru libres qui générent $\mathcal{L}$ est appelé une base.
- Il y a une infinité de bases.
- Certaine sont meilleures : orthogonalité, petits vecteurs.

Probleme difficile sur les lattices (SVP)

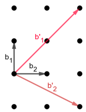




Shortest Vector Problem (SVP) : Trouver un vecteur petit de $\mathcal{L} \setminus \{0\}$.
$||v|| = \sqrt{\sum_{i=i}^{n} v_i^2}$

Deux problèmes difficiles sur les Lattices

SIS and LWE



Short Integer Solution (SIS)
Soit $q, n \in \mathbb{N}$.

`Entrée :` $A \stackrel{\mathcal{U}}{\leftarrow} \mathrm{M}_n(\mathbb{Z}/q\mathbb{Z})$

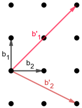
`But :` Trouver **un petit vecteur** $s \in \mathbb{Z}^n \mid As = 0 \bmod q$

Learning With Error (LWE)
Soit $q, n, m \in \mathbb{N}$.

`Entrée :` $(A, \quad b = As + e)$,
où $A \stackrel{\mathcal{U}}{\leftarrow} \mathrm{M}_{m,n}(\mathbb{Z}/q\mathbb{Z})$, $s \stackrel{\mathcal{D}_s}{\leftarrow} (\mathbb{Z}/q\mathbb{Z})^n, e \stackrel{\mathcal{D}_e}{\leftarrow} \mathbb{Z}^m$

`But :` Trouver s.

Signature : FALCON, 2017



Fast Fourier **l**attice-based **co**mpact signatures over **N**TRU.

Génération de clés

1. Clé publique : $A \xleftarrow{\$} R_q^{m \times k}$
2. Clé secrète : $B \xleftarrow{\$} R_q^{m \times k}$ de petite norme tel que $A \cdot B = 0 \bmod q$

Signature de m

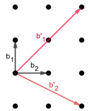
1. Calculer $c = A^{-1} \cdot H(m)$
2. Calculer $v \in B \cdot R_q$ (partie difficile)
3. $\sigma = c - v$

Vérification

$A \cdot \sigma = H(m)$ et σ est de petite norme

$$
\begin{aligned}
A \cdot \sigma &= A \cdot (c - v) \\
&= A \cdot (A^{-1} \cdot H(m) - v) \\
&= H(m) - A \cdot v \\
&= H(m) - A \cdot B \cdot R_1 \\
&= H(m)
\end{aligned}
$$

KYBER, 2011 par Lyubashevski, Peikert et Regev



Soient n, m, et q des entiers.

Génération des clés

- Clé secrète : $s \in R$, choisi petit.
- Clé publique : $(a, b) = (a, b = a.s + e)$, où a random et $e \in R$ petit.

Chiffrement de $m =$ polynôme à coefficients 0 ou 1

- Choisir r, e_1, e_2 petits dans R.
- Caculer $c = (a.r + e_1, b.r + e_2 + \lfloor q/2 \rfloor .m) \in R_q \times R_q$

Déchiffrement de $c = (u, v)$

- Calculer
$$\begin{aligned} v - u.s &= a.s.r + e.r + e_2 + \lfloor q/2 \rfloor .m) - a.s.r - s.e_1 \\ &= (r.e - s.e_1 + e_2) + \lfloor q/2 \rfloor .m) \end{aligned}$$

Pour chaque coordonnée de m, le clair est 0 si le résultat est plus proche de 0 que de $\lfloor q/2 \rfloor$, et 1 sinon.

Plan

Codes correcteurs

- Message émis avec redondance m pour corriger t erreurs
- Message reçu m'
- Création d'une matrice génératirce G
- Calcule de la matrice de contrôle H
- Syndrome $y = Hm'$

Si $y = 0$ pas d'erreur dans m sinon on peut corriger et retrouver m

Codes correcteurs

- Message émis avec redondance m pour corriger t erreurs
- Message reçu m'
- Création d'une matrice génératirce G
- Calcule de la matrice de contrôle H
- Syndrome $y = Hm'$

Si $y = 0$ pas d'erreur dans m sinon on peut corriger et retrouver m

Problème : Décodage de syndrome

- Entrée : Matrice H, syndrome y et un poids w
- Problème : Trouver e de poids w avec $He = y$

Théorème : Berlekamp, McEliece, van Tilborg 1978

Décodage d'un syndrome est NP-complet.

Chiffrement de McEliece 1978

Génération des clés

- Soit un code correcteur de t erreurs, G une matrice génératrice
- Choisir S une matrice inversible
- Choisir une matrice de permutation P
- Clé publique : $(\mathcal{G} = S \cdot G \cdot P, t)$
- Clé privée : (S, G, P)

Chiffrement de m

- Choisir aléatoirement e "avec moins" de t erreurs
- Calculer $c = m \cdot \mathcal{G} + e$

Déchiffrement de c

- Calculer $a = c \cdot P^{-1} = m \cdot S \cdot G + eP^{-1}$
- Correction des erreurs sur a pour obtenir $b = m \cdot S$
- Résoudre le système linéaire $b = m \cdot S$, pour m

HQC : Hamming Quasi-Cyclic

Génération des clés $(\mathbf{pk}, \mathbf{sk})$

$\mathbf{h} \xleftarrow{\$} \mathcal{R}$, la matrice génératirce $\mathbf{G} \in \mathbb{F}^{k \times n}$ de $\mathcal{C}$, $\mathbf{sk} = (\mathbf{x}, \mathbf{y}) \xleftarrow{\$} \mathcal{R}^2$ tel que $\omega(\mathbf{x}) = \omega(\mathbf{y}) = w$, soit $\mathbf{pk} = (\mathbf{h}, \mathbf{s} = \mathbf{x} + \mathbf{h} \cdot \mathbf{y})$,

Chiffrement de $\mathbf{m}$ avec $\mathbf{pk}$: $\mathbf{c} = (\mathbf{u}, \mathbf{v})$.

Générer $\mathbf{e} \xleftarrow{\$} \mathcal{R}$, $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2) \xleftarrow{\$} \mathcal{R}^2$ tel que $\omega(\mathbf{e}) = w_{\mathbf{e}}$ et $\omega(\mathbf{r}_1) = \omega(\mathbf{r}_2) = w_{\mathbf{r}}$, soit $\mathbf{u} = \mathbf{r}_1 + \mathbf{h} \cdot \mathbf{r}_2$ et $\mathbf{v} = \mathbf{mG} + \mathbf{s} \cdot \mathbf{r}_2 + \mathbf{e}$

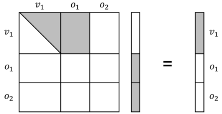
Déchiffrement de $\mathbf{c}$ avec $\mathbf{sk}$

$\mathcal{C}$.Decode$(\mathbf{v} - \mathbf{u} \cdot \mathbf{y})$.

$$\begin{aligned} \mathbf{v} - \mathbf{u} \cdot \mathbf{y} &= mG + sr_2 + e - (r_1 + hr_2)y \\ &= mG + r_2x + hy_r2 + e - r_1y - hr_2y \\ &= mG + r_2x + e - r_1y \end{aligned}$$

Plan

Problème difficile sur les systèmes multivariés



Soit l'ensemble d'équations E :

$$\begin{cases} y_1 = x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_5 + x_3x_4 + x_3x_5 \\ y_2 = x_1x_3 + x_1x_4 + x_1x_2 + x_2x_4 + x_2x_5 + x_3x_4 + x_4x_5 \\ y_3 = x_1x_2 + x_1x_4 + x_2x_3 + x_4 + x_5 \\ y_4 = x_1x_5 + x_3x_5 + x_2x_3 + x_2x_4 + x_3x_4 \\ y_5 = x_1x_2 + x_1x_3 + x_1x_5 + x_2x_5 + x_4x_5 \end{cases}$$

À partir de x_i et E, c'est facile de calculer y_i

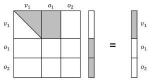
À partir de y_i et E c'est difficile de trouver x_i

Problème difficile

$$f_x(x_1, x_2, \ldots, x_n) = \sum_{1 \leq i < j \leq n} a_{ij}x_ix_j + \sum_{i=1}^{n} b_ix_i^2 + c$$

Trouver $(s_1, s_2, \ldots, s_n)$ tel quel $f_x(s_1, s_2, \ldots, s_n) = d_i$, for $i \leq i \leq m$.

HFE: Hidden Field Equations



Soit $g(X) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} a_{ij} X_i^{q^i + q^j} + x_j + \sum_{i=0}^{n-1} b_i X_i^{q^i} + c$

Génération des clés

- Clé secrete : R et S deux transformations affines inversibles
- Clé publique : La fonction suivante sur $x = (x_1, x_2, \dots, x_n)$:

$$g^{pub} = R(g(S(x)))$$

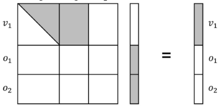
Chiffrement de $a = (a_1, a_2, \dots, a_n)$

$c = g^{pub}(a)$

Déchiffrement de c

1. Calculer $y' = R^{-1}(c)$
2. Trouver les solutions z de $g(X) = y'$
3. Calculer $a = S^{-1}(z)$.

Exemple de chiffrements multivariés



- MIA : Matsumoto Imai Scheme A, par Tsutomu Matsumoto et Ideki Imai en 1985
- STS : Sepwise Triangular Systems, par Copperslith, Stern, Vaudenay en 1993
- HFE : Hidden Field Equations, par Patarin 1996
- QUARTZ : par Courtois en 1996
- UOV : Unbalanced Oil and Vinegar, par Patarin en 1997
- SFLASH par Patarin, Courtois, Goubin en 2003

Finalistes du NIST

- MQDSS
- HFEv-: GUI, GeMSS, DualModeMS
- **Rainbow**, L(ifted)UOV, HiMQ3 (a version of TTS)

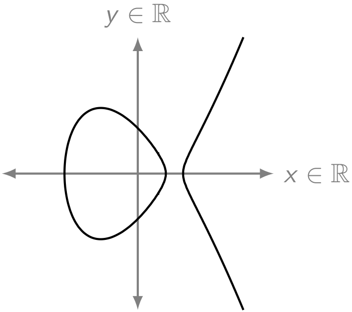
Rainbow en 2004 par Jintai Ding et Dieter Schmidt

Beaucoup sont cassés !

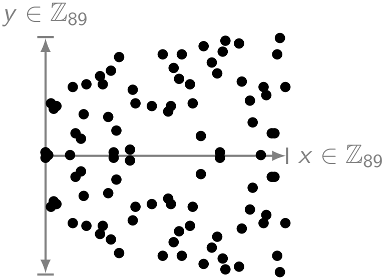
Plan

Courbes Elliptiques

$$y^2 = x^3 + ax + b$$



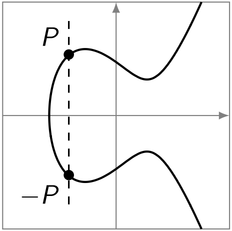
$y^2 = x^3 - 2x + 1$ over $\mathbb{R}$



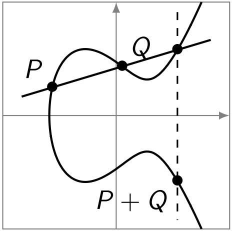
$y^2 = x^3 - 2x + 1$ over $\mathbb{Z}_{89}$

$E(K) = \{(x, y)$ tel que $y^2 = x^3 + ax + b\}$ plus un point "à l'infini"
Weierstrass : $\Delta = -16(4a^3 + 27b^2) \neq 0$
Si K n'est pas de caractérisitque 2 ou 3

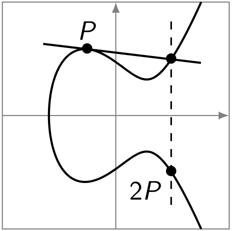
Lois de groupe



Inverse $-P$



Addition $P+Q$



Double $P+P$

$$P + R + Q = \mathcal{O} \Rightarrow R = -(P + Q)$$

$$R + S + \mathcal{O} = \mathcal{O} \Rightarrow R = -S$$

Isogénie

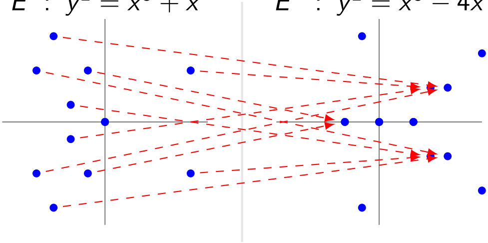
Définition

Une isogénie $\varphi : E_1 \to E_2$ est un (non-trivial) homomorphism de groupe défini par $f_i, g_i \in k[x, y]$, avec

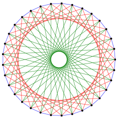
$$\varphi(x, y) = \left(\frac{f_1(x, y)}{g_1(x, y)}, \frac{f_2(x, y)}{g_2(x, y)} \right)$$

Exemple : $(x, y) \mapsto \left(\frac{(x^2+1)}{x}, \ \frac{y(x^2-1)}{x^2} \right)$ sur $\mathbb{N}_{11}$

$E \ : \ y^2 = x^3 + x$ $\qquad$ $E' \ : \ y^2 = x^3 - 4x$



Supersingular isogeny Diffie–Hellman key exchange



SIKE : Supersingular Isogeny Key Encapsulation

- SIDH proposé par Feo, Jao and Plût, PQCrypto 2011
- SIDH est vulnérable à une attaque “key-recovery” en juillet 2022

NIST

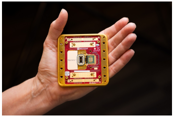
15 septembre 2022 :

SIKE and SIDH are insecure and should not be used

Plan

Changements en cours

- 2014 : La Fondation Linux a créé la Post-Quantum Cryptography Alliance (PQCA)
- Septembre 2023 : PQXDH protocol (Signal)
  devient quantum resistant
- Février 2024 : PQ3 protocol (imessage)
  devient quantum resistant
- Avril 2024 :
  Google Chrome $>$ 124 utilise KEM Kyber768 pour TLS 1.3
- Mai 2024 :
  migre vers Kyber pour l'échange de clés TLS
- Février 2025 : 1 million de qbits topologiques, Majorana



Hybridation

Le meilleur des deux mondes

Standardisation 2023

NIST
24 Août 2023

- ML-KEM : CRYSTALS-Kyber
- KEM : HQC
- ML-DSA : CRYSTALS-Dilithium
- SLH-DSA : SPHINCS+
- FN-DSA : Falcon (en cours)

	Clé publique	Chiffré	Encapsulation	Décapsulation
ML-KEM	800	2 420	45 200	34 572

	Taille en bytes		Temps en cycles	
	Clé publique	Signature	Signature	Vérification
ML-DSA	1 312	2 420	333 013	118 412
FN-DSA	897	666	386 678	82 340
SLH-DSA	32	17 088	1 100 000 000	1 190 000

A retenir

Novembre 2024

Algorithmes	Transition
ECDSA	Déprécié après 2030
	Interdit après 2035
RSA	Déprécié après 2030
	Interdit après 2035

Conclusion

Merci pour votre attention

pascal.lafourcade@uca.fr