Schémas cryptographiques à clé publique à base de codes correcteurs d'erreurs proposés à la compétition du NIST

J.-P. Tillich
Inria, équipe SECRET

1er juin 2018

0. Cryptographie à clé publique post-quantique

- Cryptographie résistant à l'ordinateur quantique
- Tous les schémas à clé publique utilisés en pratique sont cassés par un ordinateur quantique, RSA, log discret
- Compétition du NIST fin 2017 : standardiser des solutions de remplacement

Les solutions de remplacement

- Cryptographie à base de réseaux
- Cryptographie basée sur les codes
- Cryptographie basée sur les systèmes algébriques multivariés
- Cryptographie basée sur les fonctions de hachage
- Cryptographie basée sur les isogénies
- ...

Propositions au NIST

	signatures	chiffrement/échange de clés
réseaux	5	20
codes	3	18
multivarié	8	3
hachage	2	0
isogénies	0	1
divers	3	5

1. Cryptographie basée sur les codes

Problème difficile en cryptographie

Problème 1. [Décodage]
Entrée : n, r, t *avec* $r < n$, *matrice de parité* $\boldsymbol{H} \in \mathbb{F}_q^{r \times n}$, $\boldsymbol{s} \in \mathbb{F}_q^r$
Question : $\exists$ *?* $\boldsymbol{e}$ *tel que*

$$\begin{cases} \boldsymbol{H}\boldsymbol{e}^\intercal & = & \boldsymbol{s}^\intercal \\ |\boldsymbol{e}| & \leqslant & t \end{cases}$$

où $|\boldsymbol{e}|$ = *poids de hamming de* $\boldsymbol{e} = \#\{i \in [\![1, n]\!], e_i \neq 0\}$.

Problème NP-complet

Le problème dual

$$\begin{aligned} \text{Code } \mathcal{C} &\stackrel{\text{def}}{=} \left\{ \boldsymbol{c} \in \mathbb{F}_q^n : \boldsymbol{H}\boldsymbol{c}^\intercal = 0 \right\} \\ \dim \mathcal{C} &= n - r = k \end{aligned}$$

Entrée : t , $\mathcal{C}$ sous-espace de dim k de $\mathbb{F}_q^n$, $\boldsymbol{y} \in \mathbb{F}_q^n$

Question : $\exists$? $\boldsymbol{c} \in \mathcal{C}$ tel que $|\boldsymbol{y} - \boldsymbol{c}| \leqslant t$.

$$\boldsymbol{H}\underbrace{(\boldsymbol{y} - \boldsymbol{c})}_{\boldsymbol{e}}{}^\intercal = \boldsymbol{H}\boldsymbol{y}^\intercal = \boldsymbol{s}^\intercal$$

$$\begin{aligned} \boldsymbol{y} &= \text{le mot que l'on veut décoder} \\ \boldsymbol{e} &= \boldsymbol{y} - \boldsymbol{c} = \text{l'erreur que l'on veut trouver} \end{aligned}$$

Un problème très étudié

Corr. t erreurs dans code de long. n et dim. k a un coût $\tilde{O}(2^{\alpha(\frac{k}{n},\frac{t}{n})n})$

Auteur(s)	année	$\max_{R,\tau} \alpha(R,\tau)$
Prange	1962	0.1207
Stern	1988	0.1164
Dumer	1991	0.1162
Bernstein, Lange, Peters	2011	
May, Meurer, Thomae	2011	0.1114
Becker, Joux, May, Meurer	2012	0.1019
May, Ozerov	2015	0.0966
Both, May	2017	0.0953
Both, May	2018	0.0885

Ces complexités coincident quand $t = o(n)$

- [CantoTorres, Sendrier, 2016] complexité $2^{-\log(1-R)t(1+o(1))}$ quand $t = o(n)$ et $R = k/n$
- Pas mieux que l'algorithme de Prange de 1962...

Algorithme de Prange

$$\begin{aligned} \boldsymbol{H}\boldsymbol{e}^\intercal &= \boldsymbol{s}^\intercal \\ \boldsymbol{e} &= (e_i)_{1\leqslant i\leqslant n} \\ |\boldsymbol{e}| &\leqslant t \end{aligned}$$

$\boldsymbol{H}\boldsymbol{e}^\intercal = \boldsymbol{s}^\intercal$: $n-k$ équations, n inconnues.

- Idée : espérer que $e_i = 0$ sur un ensemble de taille k, par exemple $e_{n-k+1} = \cdots = e_n = 0 \Rightarrow (n-k)$ inconnues $e_1, \ldots, e_{n-k}$.
- $\boldsymbol{H} = \begin{pmatrix} \boldsymbol{H}_1 & \boldsymbol{H}_2 \end{pmatrix}$, $\boldsymbol{H}_1 \in \mathbb{F}_q^{(n-k)\times(n-k)}$, $\boldsymbol{H}_2 \in \mathbb{F}_q^{(n-k)\times k}$, si $\boldsymbol{H}_1$ inversible alors $\boldsymbol{e}_{[\![1,n-k]\!]}{}^\intercal = \boldsymbol{H}_1^{-1}\boldsymbol{s}^\intercal$. Si $|\boldsymbol{e}_{[\![1,n-k]\!]}| \leqslant t$, c'est gagné, sinon essayer d'autres positions, jusque cela marche. Si $t = o(n)$,
  proba qu'un essai marche $\approx \left(\frac{n-k}{n}\right)^t = (1-R)^t$

Cryptographie basée sur les codes

Code $\mathcal{C} \stackrel{\text{def}}{=} \{\boldsymbol{c} \in \mathbb{F}_q^n : \boldsymbol{H}\boldsymbol{c}^\intercal = 0\}$

- Choisir un code qui a un algo de décodage efficace permettant de corriger t erreurs
- Clé publique : matrice de parité aléatoire $\boldsymbol{H}_{\text{rand}} = \boldsymbol{Q}\boldsymbol{H}$ du code où $\boldsymbol{Q}$ est une matrice inversible dans $\mathbb{F}_q^{r \times r}$
- Clé secrète : structure qui permet le décodage
- Chiffrement de Niederreiter

$$\begin{aligned} \text{message} &= \boldsymbol{e} \in \mathbb{F}_q^n \text{ de poids } t \\ \text{chiffré} &= \boldsymbol{H}\boldsymbol{e}^T \end{aligned}$$

Deux approches

- Choisir un code (qui a un décodage efficace)
- Choisir une famille de codes avec une réduction au problème de décodage générique d'un code linéaire.

Histoire

- 1978 McEliece : codes de Goppa binaires
- 1986 variante de Niederreiter basée sur des codes GRS
- 1991 Gabidulin, Paramonov, Tretjakov : codes de Gabidulin
- 1994 Sidelnikov : codes de Reed-Muller
- 1996 Janwa-Moreno : codes géométriques
- 199* un million de propositions avec des codes LDPC
- 2003 Alekhnovich : système d'Alekhnovich
- 2005 Berger-Loidreau : sous-codes de codes GRS
- 2006 Wieschebrink, codes GRS + colonnes aléatoires dans mat. génératrice
- 2008 Baldi-Bodrato-Chiaraluce : codes MDPC basés sur des codes LDPC
- 2010 Bernstein, Lange, Peters : codes de Goppa sauvages
- 2012 Misoczki-Tillich-Barreto-Sendrier : codes MDPC

- 2012 Löndahl-Johansson : codes convolutifs
- 2013 Gaborit, Murat, Ruatta, Zémor : codes LRPC
- 2014 Shrestha, Kim : codes polaires
- 2014 Hooshmand, Shooshtari, Eghlidos, Aref : sous-codes de codes polaires

Histoire

- 1978 McEliece : codes de Goppa binaires
- 1986 variante de Niederreiter basée sur des codes GRS
- 1991 Gabidulin, Paramonov, Tretjakov : codes de Gabidulin
- 1994 Sidelnikov : codes de Reed-Muller
- 1996 Janwa-Moreno : codes géométriques
- 199* un million de propositions avec des codes LDPC
- 2003 Alekhnovich : système d'Alekhnovich
- 2005 Berger-Loidreau : sous-codes de codes GRS
- 2006 Wieschebrink, codes GRS + colonnes aléatoires dans mat. génératrice
- 2008 Baldi-Bodrato-Chiaraluce : codes MDPC basés sur des codes LDPC
- 2010 Bernstein, Lange, Peters : codes de Goppa sauvages
- 2012 Misoczki-Tillich-Barreto-Sendrier : codes MDPC

- 2012 Löndahl-Johansson : codes convolutifs
- 2013 Gaborit, Murat, Ruatta, Zémor : codes LRPC
- 2014 Shrestha, Kim : codes polaires
- 2014 Hooshmand, Shooshtari, Eghlidos, Aref : sous-codes de codes polaires

2. Réduire la taille de la clé publique

- Un inconvénient de la cryptographie à base de codes : la taille des clés publiques

Code de Goppa binaire (McEliece)

$$
\begin{aligned}
t &= \Theta\left(\frac{n}{\log n}\right) \\
\text{taille de clé } K &= \Theta\left(n^2\right) \\
\text{sécurité (bits) } \log_2 S &= \Theta\left(t\right) \\
&\Downarrow \\
\log_2 S &= \Theta\left(\frac{\sqrt{K}}{\log K}\right)
\end{aligned}
$$

Codes quasi-cycliques

matrice génératrice $\boldsymbol{G} = \left[\begin{array}{c|c|c} & & \\ \hline \cdots & \boldsymbol{C}_i & \cdots \\ \hline & & \end{array}\right]$ formée par un nombre constant de matrices circulantes

$$\boldsymbol{C}_i = \begin{bmatrix} c_0 & c_1 & \cdots & c_{p-2} & c_{p-1} \\ c_{p-1} & c_0 & c_1 & \cdots & c_{p-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & c_{p-1} & c_0 & c_1 \\ c_1 & \cdots & \cdots & c_{p-1} & c_0 \end{bmatrix}$$

codes alternants/Goppa quasi-cycliques

- 2005 Gaborit : sous-codes quasi-cycliques de codes BCH.
- 2007 Otmani,Tillich, Dallot : attaque.
- 2009 Berger, Cayrel, Gaborit, Otmani : codes alternants quasi-cycliques.
- 2009 Misoczki, Barreto : codes de Goppa quasi-dyadiques.
- 2010 Faugère, Otmani, Perret, Tillich/Gauthier, Leander : presque tous les paramètres des schémas proposés en 2009 ont été cassés par des attaques algébriques (rendues possibles à cause de la réduction du nombre d'inconnues).
- 2015 Faugère, Otmani, Perret, Portzamparc, Tillich réduit le problème de retrouver la clé secrète d'un code de Goppa quasi-cyclique à celle de retrouver la clé secrète d'un code de Goppa plus petit.

Un isomorphisme d'anneau

$\mathcal{R}_p$ L'anneau des matrices circulantes de taille p sur $\mathbb{F}_q$.

$$\mathcal{R}_p \cong \mathbb{F}_q[X]/(X^p - 1)$$

$$\begin{bmatrix} c_0 & c_1 & \cdots & c_{p-2} & c_{p-1} \\ c_{p-1} & c_0 & c_1 & \cdots & c_{p-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & c_{p-1} & c_0 & c_1 \\ c_1 & \cdots & \cdots & c_{p-1} & c_0 \end{bmatrix} \mapsto \quad c_{p-1}X^{p-1} + \cdots + c_1 X + c_0$$

Changer le code pour avoir une preuve de sécurité

[Sendrier, 2010] réduction de sécurité : si un attaquant peut casser le système de McEliece/Niederreiter avec un code de Goppa $[n, k]$ corrigeant t erreurs, alors ou bien

(i) il est capable de décoder t erreurs de manière efficace dans un code linéaire générique de paramètres $[n, k]$

(ii) il est capable de distinguer un code de Goppa binaire $[n, k]$ d'un code linéaire générique.

Problème : [Faugère-Gauthier-Otmani-Perret-Tillich, 2011] Distingueur algébrique des codes de Goppa en rendement R proche de 1.

Un modèle probabiliste de l'attaquant

Un (T, ϵ) adversaire $\mathcal{A}$ pour **Nied**$(\mathcal{K}_{n,k,t})$ est un algorithme qui en temps T vérifie

$$\mathsf{Prob}_{\boldsymbol{H},\boldsymbol{e}}(\mathcal{A}(\boldsymbol{H}, \boldsymbol{H}\boldsymbol{e}^{\mathsf{T}}) = \boldsymbol{e} | \boldsymbol{H} \in \mathcal{K}_{n,k,t}) \geqslant \epsilon$$

La plupart des attaques fournissent un adversaire pour **Nied**$(\mathcal{K}^{\mathsf{lin}}(n,k))$ au lieu de **Nied**$(\mathcal{K}^{\mathsf{Goppa}}(n,k,t))$.

Comment le distingueur apparait

$$\textbf{Adv} \stackrel{\text{def}}{=} \text{Prob}(\mathcal{A}(\boldsymbol{H}, \boldsymbol{H}\boldsymbol{e}^\intercal) = \boldsymbol{e} | \boldsymbol{H} \in \mathcal{K}^{\text{Goppa}}_{n,k,t}) - \text{Prob}(\mathcal{A}(\boldsymbol{H}, \boldsymbol{H}\boldsymbol{e}^\intercal) = \boldsymbol{e} | \boldsymbol{H} \in \mathcal{K}^{\text{lin}}_{n,k})$$

Distingueur D :

entrée : $\boldsymbol{H} \in \mathbb{F}_q^{(n-k)\times n}$

Etape 1 : tirer aléatoirement $\boldsymbol{e} \in \mathbb{F}_q^n$ de poids t

Etape 2 : si $\mathcal{A}(\boldsymbol{H}, \boldsymbol{H}\boldsymbol{e}^\intercal) = \boldsymbol{e}$ alors retourner 1, sinon retourner 0.

Avantage de $D \stackrel{\text{def}}{=} |\textbf{Adv}|$.

algorithme de décodage de codes linéaires ou distingueur de codes de Goppa

Proposition 1. *[Sendrier, 2010] Si $\exists(T,\epsilon)$-adversaire contre* **Nied**$(\mathcal{K}^{Goppa}_{n,k,t})$ *, alors de deux choses l'une*

(i) on a un $(T,\epsilon/2)$-adversaire contre **Nied**$(\mathcal{K}^{lin}(n,k)$ *(i.e. un decodeur de codes linéaires en temps T avec probabilité de succès $\geqslant \epsilon/2$).*

(ii) on a un distingueur entre $\boldsymbol{H} \in \mathcal{K}^{Goppa}_{n,k,t}$ et $\boldsymbol{H} \in \mathcal{K}^{lin}_{n,k}$ en temps $T + O(n^2)$ et avantage $\geqslant \epsilon/2$.

Codes QC-MDPC

Code de matrice de parité

$$\boldsymbol{H}_{\text{secret}} = \left[\begin{array}{c|c} h_0 & h_1 \\ \circlearrowright & \circlearrowright \end{array}\right]$$

avec

$$\begin{array}{rcl} h_0(X) & & \text{inversible dans } \mathbb{F}_2[X]/(X^p-1) \\ |h_i(X)| & = & w = \Theta\left(\sqrt{p}\right) \end{array}$$

- peut corriger $\Theta\left(\sqrt{p}\right)$ avec l'algorithme de Gallager.

Codes QC-MDPC

On publie $h(x) = h_1(x)h_0^{-1}(x) \mod x^p - 1$

$$\boldsymbol{H}_{\text{public}} = \begin{array}{|c|c|}\hline 1 & h \\ \diagdown & \circlearrowright \\ 1 & \\ \hline \end{array}$$

$$\begin{aligned} \text{message} &= (e_0(x), e_1(x)) \text{ avec } |e_0(x)| + |e_1(x)| = t = \Theta(\sqrt{p}) \\ \text{chiffré} &= c(x) = e_0(x) + h(x)e_1(x) \end{aligned}$$

Sécurité/Taille de clé

$$
\begin{aligned}
t &= \Theta\left(\sqrt{n}\right) \\
\text{taille de clé } K &= \Theta\left(n\right) \\
\text{sécurité (bits) } \log_2 S &= \Theta\left(t\right) \\
&\Downarrow \\
\log_2 S &= \Theta\left(\sqrt{K}\right)
\end{aligned}
$$

Sécurité	128 bits	256 bits
Taille de clé McEliece	200kB	1MB
Taille de clé QC-MDPC	1.3kB	4.1kB

Déchiffrement

$$
\begin{aligned}
s(x) &= c(x)h_0(x) \\
&= \left(e_0(x) + h(x)e_1(x)\right) h_0(x) \\
&= \left(e_0(x) + h_1(x)h_0^{-1}(x)e_1(x)\right) h_0(x) \\
&= e_0(x)h_0(x) + e_1(x)h_1(x) \\
s(x) &= s_0 + \cdots + s_{p-1}x^{p-1} \\
e_i(x) &= e_{i0} + \cdots + e_{ip-1}x^{p-1} \\
h_0(x) &= x^{i_0} + \underbrace{\cdots}_{\Theta(\sqrt{p}) \text{ termes}} \\
s_{i_0} &= e_{00} + R \\
\mathsf{Prob}(R = 1) &= \frac{1}{2} - \varepsilon
\end{aligned}
$$

Déchiffrement (II)

- Le bit e_{00} intervient dans le calcul de $\Theta(\sqrt{p})$ bits de syndrôme : $i_0, \ldots, i_{w-1}$ si $h_0(x) = x^{i_0} + \cdots + x^{i_{w-1}}$.
- $e_{00} =$ vote majoritaire sur les s_{i_j}
- Se généralise aux autres bits d'erreur
- Peut être fait itérativement. [Tillich, 2018] : Proba(erreur) $= 2^{-\alpha n}$

Réduction de sécurité dans le cas des codes QC-MDPC

[Misoczki-Tillich-Sendrier-Barreto, 2013]

1. Caractère pseudo-aléatoire de la clé publique : décider s'il existe un mot de poids $w = \Theta\left(\sqrt{n}\right)$ dans un code QC de rendement $\frac{1}{2}$.
   $\exists h_0(x),\ h_1(x)$? tel que
   (a) $|h_i(x)| = w$
   (b) $h(x)h_0(x) + h_1(x) = 0 \mod x^p - 1$
2. décodage d'un code QC-générique de rendement $\frac{1}{2}$:
   trouver $e_0(x),\ e_1(x)$ tel que
   (a) $|e_i(x)| = w$
   (b) $h(x)e_0(x) + e_1(x) = s(x) \mod x^p - 1$

Schéma d'échange de clés

- Soumission au NIST BIKE [Aguilar, Aragon, Barreto, Bettaieb, Bidoux, Blazy, Deneuville, Gaborit, Gueron, Güneysu, Misoczki, Persichetti, Sendrier, Tillich, Zémor]
- Clés éphémères

Schéma d'échange de clés

$h_i,\ e_i \in F_2[x]/(x^p-1),\ |h_i| = \Theta(\sqrt{p}),\ |e_i| = \Theta(\sqrt{p})$

Alice		Bob
$(h_0, h_1, h = h_1 h_0^{-1})$		
	$\xrightarrow{\quad h \quad}$	
		$K = \mathsf{hash}(e_0, e_1)$
		$c = e_0 + h e_1$
	$\xleftarrow{\quad c \quad}$	
$s = c h_0$		
$= e_0 h_0 + e_1 h_1$		
$(e'_0, e'_1) = \mathsf{decode}_{h_0,h_1}(s)$		
$K' = \mathsf{hash}(e'_0, e'_1)$		

Un schéma avec une réduction au problème de décodage

Alice

(h_0, h_1)

$(f_0 = h_1 + rh_0, f_1 = r)$

Bob

$\xrightarrow{(f_0,f_1)}$

$K = \mathsf{hash}(e_0, e_1)$

$(c_0, c_1) = (e + e_1 f_0, e_0 + e_1 f_1)$

$\xleftarrow{(c_0,c_1)}$

$$\begin{array}{lcl} s & = & c_0 + c_1 h_0 \\ & = & e + e_0 h_0 + e_1 h_1 \\ (e'_0, e'_1) & = & \mathsf{decode}_{h_0,h_1}(s) \\ K' & = & \mathsf{hash}(e'_0, e'_1) \end{array}$$

Décodage d'un code quasi-cyclique

Problème 2. [décodage d'un code quasi-cyclique DDCQC-$(2,1)$]
Entrée : h, $s \in \mathcal{R} \stackrel{def}{=} F_2[x]/(x^p-1)$, *entier* $t > 0$
Problème : $\exists$ *?* $e_0, e_1 \in \mathcal{R}$ *t.q.* $|e_0| + |e_1| \leqslant t$ *et* $e_0 + e_1 h = s$

Problème 3. [décodage d'un code quasi-cyclique DDCQC$(3,1)$]
Entrée : h_0, h_1, s_0, $s_1 \in \mathcal{R}$, *entier* $t > 0$
Problème : $\exists$ *?* $e_0, e_1, e_2 \in \mathcal{R}$ *t.q. (i)* $|e_0| + |e_1| + |e_2| \leqslant 3t/2$ *(ii)* $e_0 + e_2 h_0 = s_0$, *(iii)* $e_1 + e_2 h_1 = s_1$.

Distinguer

Alice

(h_0, h_1)

$(f_0 = h_1 + rh_0, r)$

Bob

$\xrightarrow[\text{DDCQC(2,1)}]{(f_0,r) \text{ ou } (f_0^*,r^*)}$

$K = \mathsf{hash}(e_0, e_1)$

$c_0 = e + e_1 f_0$

$c_1 = e_0 + e_1 r$

$\xleftarrow{(c_0,c_1) \text{ ou } (c_0^*,c_1^*)}$

$$\begin{array}{lcl} s & = & c_0 + c_1 h_0 \\ & = & e + e_0 h_0 + e_1 h_1 \\ (e'_0, e'_1) & = & \mathsf{decode}(s) \\ K' & = & \mathsf{hash}(e'_0, e'_1) \end{array}$$

Distinguer (II)

Alice

(h_0, h_1)

$(f_0 = h_1 + rh_0, r)$

Bob

$\xrightarrow[\text{DDCQC}(2,1)]{(f_0,r) \text{ ou } (f_0^*,r^*)}$

$K = \mathsf{hash}(e_0, e_1)$

$c_0 = e + e_1 f_0$

$c_1 = e_0 + e_1 r$

$\xleftarrow[\text{DDCQC}(3,1)]{(c_0,c_1) \text{ ou } (c_0^*,c_1^*)}$

$$\begin{array}{lcl} s & = & c_0 + c_1 h_0 \\ & = & e + e_0 h_0 + e_1 h_1 \\ (e'_0, e'_1) & = & \mathsf{decode}(s) \\ K' & = & \mathsf{hash}(e'_0, e'_1) \end{array}$$

Conclusion

- Schémas avec une réduction au décodage de codes quasi-cycliques
- Situation très stable par rapport au décodage de codes linéaires
- Situation très stable par rapport au décodage de codes linéaires quasi-cycliques ?
- Réduction décision $\leftrightarrow$ recherche ?
- Signatures à base de codes ?
- Pire cas/cas moyen ?