

UNIVERSITÉ DE BORDEAUX

MASTER 2 : CRYPTOLOGIE ET SÉCURITÉ
INFORMATIQUE

PROJET DE FIN D'ÉTUDES

Wave - Un procédé de signature à base de codes correcteurs

Suzanne LANSADÉ
Eva PALANDJIAN

Encadrant:
Gilles ZEMOR

Février, 2020



Contents

Introduction	2
1 Le schéma de signature Wave	3
1.1 La famille de codes $(U, U+V)$ -généralisés	3
1.2 Le principe de signature	6
1.3 Le décodage avec trappe	7
1.4 Implémentation et choix de paramètres	11
2 Uniformisation des signatures et syndromes	12
2.1 Une fuite d'information	12
2.2 La méthode du rejet	13
2.3 Choix des algorithmes de décodage	16
2.4 Estimation du nombre de rejet	16
2.5 Une famille de fonctions uniformément distribuée	19
3 Sécurité du schéma	19
3.1 Sécurité EUF-CMA	19
3.1.1 Définitions	19
3.1.2 Réduction au problème DOOM	20
3.1.3 Preuve formelle de la réduction	21
3.2 Indistinguabilité des codes $(U, U+V)$ -généralisés	23
Conclusion	24
ANNEXES	25

Introduction

- Passage au post-quantique \rightarrow On cherche des alternatives à RSA et DH.
- Appel d'offre NIST
- On aimerai bien un système de signature utilisant les codes, puisque le problème du décodage d'un code aléatoire est NP-complet.
- Ça marche pas mal pour le chiffrement, mais encore très compliqué pour les signatures, comme en témoigne le tableau des soumissions au NIST pour le second tour.

	Proportion	Échange de clefs	Signature
Réseaux	46%	9	3
Codes	28%	7	0
Fonctions de hachage	4%	0	1
Multi-varié	15%	0	4
Isogénies	4%	1	0
Preuves ZK	4%	0	1

Figure 1: Comparaison des soumissions au NIST du second tour.

Alors pourquoi est-il si compliqué de faire des signatures avec des codes correcteurs ?

- car il est difficile de tomber dans l'ensemble des syndromes facilement décodables, ie. difficile de créer une fonction de hachage qui envoie le message m dans l'ensemble des syndromes possibles. En effet, pour un décodage au sens stricte il faudrait un syndrome s associé à un unique mot c du code, le plus proche de m .

quand on chiffre \rightarrow ne pose pas de problème.

quand on signe \rightarrow pose un problème car il est dur de trouver un syndrome de cette sorte.

Les solutions à ce problème ont pour l'instant soit été cassées, soit impraticables.

La solution de Wave est d'enlever la restriction au mot le plus proche. On cherche maintenant une famille de codes permettant de trouver, pour un mot quelconque, un des mots de code à distance ω . En particulier, la distance de décodage ω est très grande, ce qui assure typiquement de l'existence d'un

mot de code à distance ω .

Explication rapide du schéma wave + on cherchera à rendre les sorties uniformes + preuves de sécurité.

1 Le schéma de signature Wave

Nous allons détailler dans cette section le schéma de signature Wave. C'est un schéma de type hache et signe à base de codes correcteurs. Pour des raisons de clarté nous oublierons dans un premier temps la problématique du hachage. Nous la réintroduirons en fin de rapport afin de proposer une preuve formelle de la sécurité du schéma, où la fonction de hachage est alors nécessaire. Pour l'instant, nous considérerons que l'entrée de l'algorithme de signature est déjà un syndrome du code considéré.

Le schéma de signature Wave s'appuie sur une famille de codes appelés des codes $(U, U + V)$ -généralisés. La structure de ces codes nous permettra de proposer un algorithme de décodage \mathcal{D} utilisant une trappe T et donnant un avantage par rapport à un algorithme de décodage générique. Ce système s'appuie aussi sur la notion de fonctions GPV en moyenne, que nous détaillerons.

1.1 La famille de codes $(U, U + V)$ -généralisés

Pour définir la famille de code utilisée dans le schéma de signature Wave, on part de deux codes de longueur $n/2$ aléatoires de dimensions respectives k_u et k_v . On a premièrement la définition suivante :

Définition 1.1. (*Code $(U, U + V)$.*) Un code $(U, U + V)$ est un code de longueur n et de dimension $k = k_U + k_V$ et tel que :

$$(U, U + V) = \{(u, u + v) \text{ tel que } u \in U \text{ et } v \in V\}$$

En revanche, pour le système de signature Wave, nous n'utiliserons pas cette définition. En effet, on peut facilement tirer des informations sur la structure de ces codes.

Définition 1.2. (*hull*). Le hull d'un code \mathcal{C} est défini comme $\text{hull}(\mathcal{C}) := \mathcal{C} \cap \mathcal{C}^\perp$

Proposition 1.3. *Soit \mathcal{C} un code aléatoire binaire de longueur n et de dimension k , alors on s'attend à avoir $\dim(\text{hull}(\mathcal{C})) \sim \mathcal{O}(1)$. De plus :*

$$\mathbb{P}(\dim(\text{hull}(\mathcal{C})) \leq t) \geq 1 - \mathcal{O}(2^{-t})$$

Proposition 1.4. *Soit UV un code $(U, U+V)$ binaire permuté de longueur n tel que $k_U > k_V$, alors nous avons avec probabilité $1 - \mathcal{O}(2^{k_U - k_V})$*

$$\dim(\text{hull}(UV)) = k_U - k_V$$

Preuve. En annexe. □

Ainsi nous ne pouvons pas utiliser les codes $(U, U+V)$ binaires où $\dim(U) > \dim(V)$ puisque nous pouvons facilement les distinguer de codes aléatoires. En effet, il suffit de calculer la dimension de leur *hull* puis de le comparer au résultat de la proposition 1.3. Pour résoudre ce problème, nous poserons $q = 3$ pour toute la suite du rapport. Nous utiliserons également une nouvelle famille de codes, les codes $(U, U+V)$ -généralisés :

Définition 1.5. *(codes $(U, U+V)$ -généralisés) Soient n un entier pair et a, b, c, d quatres vecteurs de $\mathbb{F}_q^{n/2}$ tels que pour tout $i \in \llbracket 1, n/2 \rrbracket$:*

$$a_i c_i \neq 0 \tag{1}$$

$$a_i d_i - b_i c_i \neq 0 \tag{2}$$

Soient U et V deux codes définis comme précédemment. Le code $(U, U+V)$ -généralisé correspond à l'ensemble :

$$\{(a.u + b.v, c.u + d.v) \text{ tel que } u \in U \text{ et } v \in V\}$$

où $x.y$ est le produit coordonnée par coordonnée des x_i et y_i .

Les conditions sur les vecteurs a, b, c, d permet de garantir que :

- toutes les coordonnées de $u \in U$ apparaîtront deux fois, ce qui sera nécessaire pour utiliser la structure du code dans notre algorithme de décodage (par l'équation (1)).
- la dimension du code $(U, U+V)$ -généralisé sera la somme des dimensions des codes U et V (par l'équation (2)).

Sans perte de généralité, nous posons pour toute la suite les vecteurs a, b, c, d tels que $a_i d_i - b_i c_i = 1$ pour tout $i \in \llbracket 1, n/2 \rrbracket$.

Proposition 1.6. *Soient U, V, a, b, c et d définis comme précédemment. Soit UV le code $(U, U + V)$ -généralisé associé. Alors*

$$k = \dim(UV) = k_U + k_V.$$

De plus soient $G_U \in \mathbb{F}_q^{k_U \times n/2}$ (respectivement $G_V \in \mathbb{F}_q^{k_V \times n/2}$) et $H_U \in \mathbb{F}_q^{(n/2 - k_U) \times n/2}$ (respectivement $H_V \in \mathbb{F}_q^{(n/2 - k_V) \times n/2}$) les matrices génératrices et de parité des codes U et V . Soient A, B, C, D de $\mathbb{F}_q^{n \times n}$ les matrices diagonales de diagonale respectives les vecteurs a, b, c et d .

Alors la matrice de $\mathbb{F}_q^{(k_U + k_V) \times n}$:

$$G := \left(\begin{array}{c|c} G_U A & G_U C \\ \hline G_V B & G_V D \end{array} \right)$$

et la matrice $\mathbb{F}_q^{(n - k_U - k_V) \times n}$:

$$H := \left(\begin{array}{c|c} H_U D & -H_U B \\ \hline -H_V C & H_V A \end{array} \right)$$

sont des matrices génératrices et de parité du code UV .

Preuve. Remarquons d'abord que G engendre bien le code UV . Remarquons aussi que

$$G = \left(\begin{array}{c|c} G_U A & G_U C \\ \hline G_V B & G_V D \end{array} \right) = \left(\begin{array}{c|c} G_U & 0 \\ \hline 0 & G_V \end{array} \right) \left(\begin{array}{c|c} A & C \\ \hline B & D \end{array} \right)$$

Par définition des matrices G_V et G_U , la matrice $\left(\begin{array}{c|c} G_U & 0 \\ \hline 0 & G_V \end{array} \right)$ est de rang $k_U + k_V$. De plus les matrices A, B, C, D étant diagonales, le déterminant de la matrice $\left(\begin{array}{c|c} A & C \\ \hline B & D \end{array} \right)$ est le produit des $(a_i d_i - b_i c_i)$ pour $i \in \llbracket 1, n/2 \rrbracket$, et donc non-nul par définition des vecteurs a, b, c, d . On a donc bien $k = k_U + k_V$. On remarque aussi que $GH^T = 0$ et que H est de rang plein par le même raisonnement que précédemment, ce qui conclut la preuve. \square

1.2 Le principe de signature

Notre schéma de signature utilisera donc les codes $(U, U + V)$ -généralisés et la fonction syndrome comme fonction à sens unique, sous l'hypothèse de la difficulté de résoudre le problème du décodage.

Nous allons définir la notion de fonctions GPV en moyenne (GPVM). Pour cela, introduisons d'abord la notion de distance statistique.

Définition 1.7. Soient X et Y deux variables aléatoires à valeurs dans le même espace ϵ . Soient \mathcal{D}_X et \mathcal{D}_Y leurs distributions respectives. On définit la distance statistique entre ces deux distributions comme :

$$\rho(\mathcal{D}_X, \mathcal{D}_Y) := \frac{1}{2} \sum_{x \in \epsilon} |\mathcal{D}_X(x) - \mathcal{D}_Y(x)|.$$

Définition 1.8. (Fonctions GPVM). On appelle fonction GPV en moyenne une paire d'algorithmes (**Trapdoor**, **InvertAlg**) ainsi qu'un triplet de fonctions $(n(\lambda), k(\lambda), \omega(\lambda))$ en fonction d'un paramètre de sécurité λ , tels que :

- **Trapdoor** est un algorithme probabiliste et polynomial en 1^λ et renvoyant le couple (H, T) où $H \in \mathbb{F}_q^{(n-k) \times n}$ de rang $n - k$ et T est la trappe associée.
- **InvertAlg** est un algorithme probabiliste et polynomial prenant en entrée la trappe T et un syndrome $s \in \mathbb{F}_q^{n-k}$, et renvoyant $e \in \mathbb{F}_q^n$ de poids ω tel que $eH^T = s$.

De plus, pour presque toutes matrices H renvoyée par **Trapdoor**, la fonction est :

1. bien distribuée :
 $\rho(eH^T, s) \in \text{negl}(\lambda)$ où e est pris uniformément dans l'ensemble des mots de poids ω et de longueur n et s est pris uniformément dans \mathbb{F}_q^{n-k} .
2. sans fuite d'information en moyenne :
 $\rho(\text{InvertAlg}(s, T), e) \in \text{negl}(\lambda)$ où e est pris uniformément dans l'ensemble des mots de poids ω et de longueur n et s est pris uniformément dans \mathbb{F}_q^{n-k} .

3. À sens unique sans la trappe :

Pour tout algorithme probabiliste polynomial \mathcal{A} , on a

$$\mathbb{P}(\mathcal{A}(H, s) = e \mid eH^T = s) \in \text{negl}(\lambda).$$

C'est une définition relaxée des fonctions GPV.

Nous pouvons maintenant définir notre système de signature.

<p>Sign^{sk}(s): $e \leftarrow \text{InvertAlg}(s, T)$ renvoie e</p>	<p>Verify^{pk}(s, e'): Si $e'H^T = s$ et $e' = \omega$ renvoie 1 renvoie 0</p>
---	--

EXPLIQUER POURQUOI LE SYSTÈME FONCTIONNE.

1.3 Le décodage avec trappe

En partant de l'hypothèse que la matrice de parité \mathbf{H} du code $(U, U + V)$ -généralisé ressemble à une matrice aléatoire, la difficulté de créer une fausse signature sans connaître la trappe \mathbf{T} est exactement celle de résoudre le problème du décodage d'un code aléatoire, que l'on sait difficile. Nous allons expliciter dans cette section l'algorithme d'inversion de la fonction syndrome connaissant la trappe, et discuter de sa difficulté en fonction du poids ω de \mathbf{e} .

Notons $\mathcal{S}_{\omega, n}$ l'ensemble des mots de poids ω et de longueur n . On notera dans la suite \mathcal{S}_{ω} s'il n'y a pas d'ambiguïté sur la longueur. On rappelle que l'algorithme **InvertAlg** cherche à inverser la fonction syndrome :

$$\begin{aligned} f_{\omega, \mathbf{H}} : \mathcal{S}_{\omega, n} &\rightarrow \mathbb{F}_q^{n-k} \\ \mathbf{e} &\mapsto \mathbf{eH}^T \end{aligned}$$

On rappelle que la fonction $f_{\omega, \mathbf{H}}$ avec $\mathbf{H} \in \mathbb{F}_q^{(n-k) \times n}$ s'inverse génériquement si $\omega \in \{\omega_{easy}^-, \omega_{easy}^+\}$, où :

$$\omega_{easy}^- := \frac{q-1}{q}(n-k) \quad \text{et} \quad \omega_{easy}^+ := k + \frac{q-1}{q}(n-k).$$

Preuve. TODO

□

On rappelle aussi que la fonction $f_{\omega, \mathbf{H}}$ admet un inverse pour toute entrée $s \in \mathbb{F}_q^{n-k}$ si $\omega \in \{\omega^-, \omega^+\}$, où :

$$\omega^- := \min \left\{ \omega \in \llbracket 0, n \rrbracket, \binom{\omega}{n} (q-1)^\omega \geq q^{n-k} \right\}$$

$$\omega^+ := \max \left\{ \omega \in \llbracket 0, n \rrbracket, \binom{\omega}{n} (q-1)^\omega \geq q^{n-k} \right\}$$

Preuve. TODO □

Nous voulons donc un moyen d'inverser la fonction syndrome pour $\omega \in \llbracket \omega_{UV}^-, \omega_{UV}^+ \rrbracket$ avec ω_{UV}^- et ω_{UV}^+ tels que :

$$\llbracket \omega_{easy}^-, \omega_{easy}^+ \rrbracket \subsetneq \llbracket \omega_{UV}^-, \omega_{UV}^+ \rrbracket \subset \llbracket \omega^-, \omega^+ \rrbracket$$

Afin d'expliciter le décodage, introduisons la fonction :

$$\begin{aligned} \varphi_{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}} : \mathbb{F}_q^{n/2} \times \mathbb{F}_q^{n/2} &\rightarrow \mathbb{F}_q^{n/2} \times \mathbb{F}_q^{n/2} \\ (\mathbf{x}, \mathbf{y}) &\mapsto (\mathbf{a} \cdot \mathbf{x} + \mathbf{b} \cdot \mathbf{y}, \mathbf{c} \cdot \mathbf{x} + \mathbf{d} \cdot \mathbf{y}) \end{aligned}$$

Si cette fonction respecte les conditions sur les vecteurs $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}$ définies dans la définition 1.5, on dit qu'elle est UV-normalisée. Dans ce cas on peut vérifier qu'elle est bijective d'inverse :

$$\begin{aligned} \varphi_{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}}^{-1} : \mathbb{F}_q^{n/2} \times \mathbb{F}_q^{n/2} &\rightarrow \mathbb{F}_q^{n/2} \times \mathbb{F}_q^{n/2} \\ (\mathbf{x}, \mathbf{y}) &\mapsto (\mathbf{d} \cdot \mathbf{x} - \mathbf{b} \cdot \mathbf{y}, -\mathbf{c} \cdot \mathbf{x} + \mathbf{a} \cdot \mathbf{y}) \end{aligned}$$

Ainsi, pour chaque vecteur \mathbf{e} de \mathbb{F}_q^n , on peut associer deux vecteurs \mathbf{e}_U et \mathbf{e}_V de $\mathbb{F}_q^{n/2}$ tels que

$$(\mathbf{e}_U, \mathbf{e}_V) = \varphi_{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}}^{-1}(\mathbf{e}).$$

Proposition 1.9. *Inverser $f_{\omega, \mathbf{H}}$ pour un certain $\mathbf{s} \in \mathbb{F}_q^{n-k}$ est équivalent à trouver $\mathbf{e} \in \mathbb{F}_q^n$ tel que:*

$$\mathbf{e}_U \mathbf{H}_U^T = \mathbf{s}^U \quad \text{et} \quad \mathbf{e}_V \mathbf{H}_V^T = \mathbf{s}^V$$

où $\mathbf{s} = (\mathbf{s}^U, \mathbf{s}^V)$ avec $\mathbf{s}^U \in \mathbb{F}_q^{n/2-k_U}$ et $\mathbf{s}^V \in \mathbb{F}_q^{n/2-k_V}$.

Preuve. TODO □

Ainsi, on aura :

```

InvertAlg(s, T) :
  (sU, sV) = s
  eU = DECODE_U(sU)
  eV = DECODE_V(sV)
  renvoie φa,b,c,d(eU, eV)

```

Si l'on choisit un algorithme générique pour DECODE_U et DECODE_V, alors nous obtiendrons un vecteur **e** de poids ω in $\{\omega_{easy}^-, \omega_{easy}^+\}$. Non allons montrer comment utiliser les propriétés des codes $(U, U+V)$ -généralisés pour permettre un décodage hors de cet intervalle.

Remarque 1.10. *Pour tout $\mathbf{e} = \varphi_{\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}}(\mathbf{e}_U, \mathbf{e}_V)$, on a pour tout $i \in \{1, n/2\}$:*

$$\begin{cases} a_i \mathbf{e}_U(i) + b_i \mathbf{e}_V(i) &= \mathbf{e}(i) \\ c_i \mathbf{e}_U(i) + d_i \mathbf{e}_V(i) &= \mathbf{e}(i + n/2) \end{cases}$$

*Choisir la valeur de \mathbf{e}_U en fonction de la valeur de \mathbf{e}_V nous permettras donc d'influer sur le poids de **e**. On aura alors :*

```

InvertAlg(s, T) :
  (sU, sV) = s
  eV = DECODE_V(sV)
  eU = DECODE_U(sU, eV)
  renvoie φa,b,c,d(eU, eV)

```

Proposition 1.11. *Soit \mathbf{e}_V une sortie de DECODE_V. Soit DECODE_U un algorithme prenant en entrée s_U et e_V et renvoyant e_U tel que $\mathbf{e}_U \mathbf{H}_U^T = \mathbf{s}_U^T$ et tel que pour k_U positions de e_U*

$$\begin{cases} a_i \mathbf{e}_U(i) + b_i \mathbf{e}_V(i) &\neq 0 \\ c_i \mathbf{e}_U(i) + d_i \mathbf{e}_V(i) &\neq 0 \end{cases}$$

Alors $\mathbf{e} = \varphi_{\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}}(\mathbf{e}_U, \mathbf{e}_V)$ a au moins $2k_U$ coordonnées non nulles. De plus les $n - k_U$ autres coordonnées sont uniformément distribuées sur \mathbb{F}_q .

On a alors

$$\mathbb{E}(|\mathbf{e}|) = \frac{q-1}{q}n + \frac{2k_U}{q}$$

et on peut alors espérer obtenir en temps polynomial des erreurs de poids:

$$\omega_{UV}^+ = \begin{cases} \frac{q-1}{q}n + \frac{2k}{q} & \text{si } k \leq n/2 \\ n & \text{sinon} \end{cases}$$

Preuve. TODO □

Proposition 1.12. *Soit \mathbf{e}_V une sortie de `DECODE_V`. Soit `DECODE_U` un algorithme prenant en entrée \mathbf{s}_U et \mathbf{e}_V et renvoyant \mathbf{e}_U tel que $\mathbf{e}_U \mathbf{H}_U^T = \mathbf{s}^U$ et tel que pour k_U positions de \mathbf{e}_U*

$$\begin{cases} a_i \mathbf{e}_U(i) + b_i \mathbf{e}_V(i) & = & 0 \\ c_i \mathbf{e}_U(i) + d_i \mathbf{e}_V(i) & = & 0 \end{cases} \quad (3)$$

On peut alors espérer obtenir en temps polynomial des erreurs de poids:

$$\omega_{UV}^- = \begin{cases} \frac{q-1}{q}(n - 2k) & \text{si } k \leq n/(2q) \\ \frac{2(q-1)^2}{(2q-1)q}(n - k) & \text{sinon} \end{cases} \quad (4)$$

Preuve. Il n'existe de solution au système (3) que si $\mathbf{e}_V(i) = 0$ car pour tout i on a $a_i d_i - b_i c_i \neq 0$. De ce fait, à l'inverse du cas où nous souhaitons des erreurs de gros poids, l'ensemble d'indices où l'on peut gagner deux fois est réduit à $n/2 - |\mathbf{e}_V|$. De ce fait le poids minimal que nous pouvons espérer pour \mathbf{e}_V est $|\mathbf{e}_V|_{\min} := \frac{q-1}{q}(n/2 - k_V)$. Ainsi :

- Si $k_U \leq n/2 - |\mathbf{e}_V|_{\min}$, nous pouvons obtenir des erreurs e telles que :
 - $2k_U$ coordonnées sont nulles.
 - Les autres coordonnées sont uniformément distribuées.
- Sinon, nous pouvons obtenir des erreurs e telles que :
 - $2(n/2 - |\mathbf{e}_V|_{\min})$ sont nulles.
 - $k_U - (n/2 - |\mathbf{e}_V|_{\min})$ autres coordonnées sont nulles tandis que $k_U - (n/2 - |\mathbf{e}_V|_{\min})$ sont non nulles.
 - Les autres coordonnées sont uniformément distribuées.

PREUVE À DÉTAILLER □

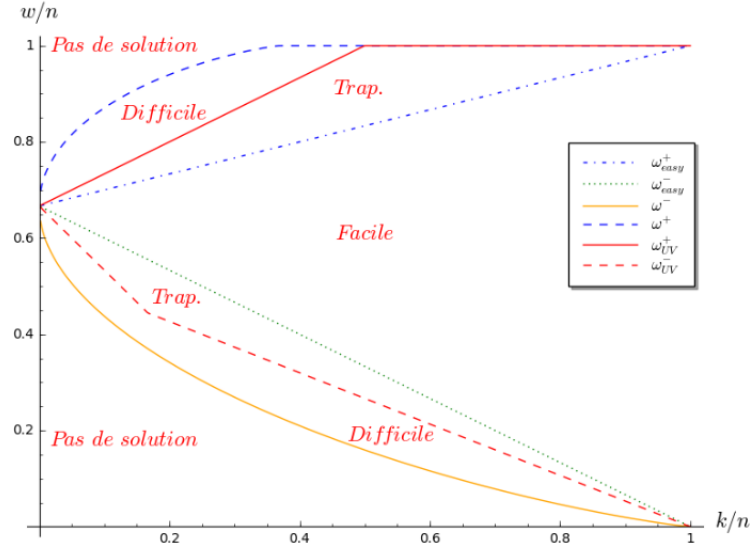


Figure 2: Comparaison des distances w/n avec et sans trappe en fonction du rendement.

On récapitule les différents cas dans la figure 2.

La connaissance de la trappe apporte donc bien un avantage puisqu'elle permet un décodage pour des erreurs de poids ne permettant pas de décodage générique.

Remarque 1.13. *Fonction de hachage À FAIRE*

1.4 Implémentation et choix de paramètres

TODO

2 Uniformisation des signatures et syndromes

2.1 Une fuite d'information

Afin d'assurer la sécurité du système, il est nécessaire que les $\mathbf{e} \in f_{w,\mathbf{H}}^{-1}(\mathbf{s})$ ne révèlent pas d'information sur la structure du code $(U, U+V)$ -généralisé utilisé.

Or, si la sortie \mathbf{e}_V de `DECODE_V` n'est pas uniforme, alors des corrélations entre les coordonnées \mathbf{e}_i et $\mathbf{e}_{i+n/2}$ du vecteur \mathbf{e} .

Par exemple, prenons le cas où $q = 3$, et où pour tout $i \in \{1, n/2\}$, $a_i = c_i = d_i = 1$ et $b_i = 0$, et où `DECODE_V` est l'algorithme de Prange.

On a alors pour tout $\mathbf{e} = (\mathbf{e}_U, \mathbf{e}_U + \mathbf{e}_V)$

$$|\mathbf{e}_V| = \# \{1 \leq i \leq n/2 \mid e_i \neq e_{i+n/2}\}$$

Proposition 2.1. *Si le vecteur \mathbf{e}_V est obtenu par l'algorithme de Prange, alors il est de poids moyen $\frac{2}{3}(\frac{n}{2} - k_V)$.*

Preuve. TODO □

Alors, pour tout $i \in \{1, n/2\}$, on a :

$$\mathbb{P}(\mathbf{e}_i \neq \mathbf{e}_{i+n/2}) = \frac{2}{3(n/2)}(n/2 - k_V)(1 + o(1))$$

PREUVE

En revanche, pour les autres paires (i, j) , on a :

$$\mathbb{P}(\mathbf{e}_i \neq \mathbf{e}_j) = \frac{4wn - 3w^2 - w}{n(n-1)}$$

PREUVE

Ces deux probabilités n'ont donc aucune raison d'être égales. On a donc une fuite d'information. En effet, dans la pratique et afin de cacher la structure, on effectue une permutation sur les coordonnées de \mathbf{e} lors de la signature. Si un attaquant récupère suffisamment de signatures, il pourra donc en analysant la fréquence des $\mathbf{e}_i \neq \mathbf{e}_j$ retrouver cette permutation. Il est donc nécessaire pour la sécurité du schéma de s'assurer de l'uniformité des sorties de l'algorithme `sign`.

2.2 La méthode du rejet

Afin de s'assurer un \mathbf{e} uniforme dans son ensemble, nous allons :

- choisir \mathbf{e}_V de façon à ce qu'il soit uniforme dans son ensemble
- mettre des conditions de rejet sur \mathbf{e}_U en fonction du poids de \mathbf{e}_V afin de supprimer le biais sur l'ensemble

$$m_1(x) := \# \{1 \leq i \leq n/2 ; |(x_i, x_{i+n/2})| = 1\}$$

Avant d'expliciter nos algorithmes, il est nécessaire d'introduire quelques notations et définitions.

Notation 2.2. *On notera :*

- \mathbf{e}^{unif} la variable aléatoire tirée uniformément dans l'ensemble $S_{w,n}$
- \mathbf{e}_V^{unif} la variable aléatoire tirée uniformément dans les mots de $\mathbb{F}_q^{n/2}$
- \mathbf{e}_U^{unif} la variable aléatoire tirée uniformément dans les mots de $\mathbb{F}_q^{n/2}$ conditionné au vecteur \mathbf{e}_V^{unif}

Définition 2.3. *(uniforme en poids et m_1 -uniforme)*

- **DECODE_V** est dit uniforme en poids si ces sorties \mathbf{e}_V sont telles que $\mathbb{P}(\mathbf{e}_V)$ n'est fonction que du poids de \mathbf{e}_V quand \mathbf{s}^V est tiré uniformément dans son ensemble.
- **DECODE_U** est dit m_1 -uniforme si ces sorties \mathbf{e}_U sont telles que $\mathbb{P}(\mathbf{e}_U \mid \mathbf{e}_V)$ n'est fonction que du poids de \mathbf{e}_V et de $m_1(\varphi(\mathbf{e}_U, \mathbf{e}_V))$.

Lemme 2.4. *Soit \mathbf{e} la sortie de **InvertAlg** avec \mathbf{s}_U et \mathbf{s}_V choisis uniformément dans leurs ensembles. Soit **DECODE_V** uniforme en poids et **DECODE_U** m_1 -uniforme. Si pour tout y et z*

$$|\mathbf{e}_V| \sim |\mathbf{e}_V^{unif}| \quad \text{et} \quad \mathbb{P}(m_1(\mathbf{e}) = z \mid |\mathbf{e}_V| = y) = \mathbb{P}(m_1(\mathbf{e}^{unif}) = z \mid |\mathbf{e}_V^{unif}| = y)$$

Alors

$$\mathbf{e} \sim \mathbf{e}_V^{unif}.$$

Preuve. Nous allons montrer qu'avec les hypothèses précédentes nous avons
 $\forall x \in S_\omega \quad \mathbb{P}(\mathbf{e} = x) = \mathbb{P}(\mathbf{e}^{unif} = x)$.
 Soit $x \in S_\omega$:

$$\begin{aligned} \mathbb{P}(\mathbf{e} = x) &= \mathbb{P}(\mathbf{e}_U = x_U, \mathbf{e}_V = x_V) \\ &= \mathbb{P}(\mathbf{e}_U = x_U | \mathbf{e}_V = x_V) \mathbb{P}(\mathbf{e}_V = x_V) \end{aligned}$$

Notre but étant de faire apparaître les expressions énoncées lors du lemme, nous allons exprimer ces deux probabilités en fonction de $|x_V| = y$ et de $m_1(x) = z$.

$$\begin{aligned} \mathbb{P}(\mathbf{e}_V = x_V) &= \mathbb{P}(\mathbf{e}_V = x_V, |\mathbf{e}_V| = y) \\ &= \mathbb{P}(\mathbf{e}_V = x_V | |\mathbf{e}_V| = y) \mathbb{P}(|\mathbf{e}_V| = y) \\ &= \frac{\mathbb{P}(|\mathbf{e}_V| = y)}{n(y)} \end{aligned}$$

avec $n(y) := \# \left\{ \mathbf{e} \in \mathbb{F}_3^{n/2} \mid |e| = y \right\}$. De la même façon, nous obtenons :

$$\begin{aligned} \mathbb{P}(\mathbf{e}_U = x_U | |\mathbf{e}_V| = y) &= \mathbb{P}(\mathbf{e}_U = x_U, m_1(\mathbf{e}) = z \mid |\mathbf{e}_V| = y) \\ &= \mathbb{P}(\mathbf{e}_U = x_U \mid m_1(\mathbf{e}) = z, |\mathbf{e}_V| = y) \mathbb{P}(m_1(\mathbf{e}) = z \mid |\mathbf{e}_V| = y) \\ &= \frac{\mathbb{P}(m_1(\mathbf{e}) = z \mid |\mathbf{e}_V| = y)}{n(y, z)} \end{aligned}$$

avec $n(y, z) := \# \left\{ \mathbf{e} \in \mathbb{F}_3^{n/2} \mid m_1(\mathbf{e}) = z \text{ et } |\mathbf{e}_V| = y \right\}$.

Ainsi nous obtenons

$$\begin{aligned} \mathbb{P}(\mathbf{e} = x) &= \frac{\mathbb{P}(m_1(\mathbf{e}) = z \mid |\mathbf{e}_V| = y)}{n(y, z)} \frac{\mathbb{P}(|\mathbf{e}_V| = y)}{n(y)} \\ &= \frac{\mathbb{P}(m_1(\mathbf{e}^{unif}) = z \mid |\mathbf{e}_V^{unif}| = y)}{n(y, z)} \frac{\mathbb{P}(|\mathbf{e}_V^{unif}| = y)}{n(y)} \\ &\quad \text{par les hypohèses} \\ &= \mathbb{P}(\mathbf{e}_U^{unif} = x_U | \mathbf{e}_V^{unif} = x_V) \mathbb{P}(\mathbf{e}_V^{unif} = x_V) \\ &= \mathbb{P}(\mathbf{e}^{unif} = x) \end{aligned}$$

□

Ainsi, pour que \mathbf{e} soit uniformément distribué sur S_ω , il suffit de choisir **DecodeV** de façon à ce que ses sorties soient uniformes sur $\mathbb{F}_q^{n/2}$ puis d'ajouter

une condition de rejet sur les sorties de **DecodeU** de façon à ce que $m_1(\mathbf{e})$ conditionnée à $|\mathbf{e}_V|$ soit distribué comme $m_1(\mathbf{e}^{unif})$ conditionnée à $|\mathbf{e}_V^{unif}|$.
On peut alors introduire l'algorithme suivant :

Algorithme 1 DecodeUV($\varphi, \mathbf{s}, \mathbf{H}_V, \mathbf{H}_U$)

Entrées: $\varphi, \mathbf{s} \in \mathbb{F}_q^{n-k}$ un syndrome, $\mathbf{H}_V \in \mathbb{F}_q^{(\frac{n}{2}-k_V) \times \frac{n}{2}}$, $\mathbf{H}_U \in \mathbb{F}_q^{(\frac{n}{2}-k_U) \times \frac{n}{2}}$

Sortie: $\mathbf{e} = \varphi(e_U, e_V)$ avec $\mathbf{e}_U \mathbf{H}_U^T = \mathbf{s}^U$ et $\mathbf{e}_V \mathbf{H}_V^T = \mathbf{s}^V$

- 1: $\mathbf{e}_V \leftarrow \text{DecodeV}(\mathbf{s}^V, \mathbf{H}_V)$
 - 2: **Faire**
 - 3: $\mathbf{e}_U \leftarrow \text{DecodeU}(\varphi, \mathbf{e}_V, \mathbf{s}^U, \mathbf{H}_U)$
 - 4: $\mathbf{e} \leftarrow \varphi(\mathbf{e}_U, \mathbf{e}_V)$
 - 5: **Tant que** $\text{rand}([0, 1]) > \mathbf{r}_U(|\mathbf{e}_V|, m_1(\mathbf{e}))$
 - 6: **Retourne** \mathbf{e}
-

Avec :

$$r(s, t) := \frac{1}{M(t)} \frac{q^{unif}(s, t)}{q(s, t)}$$

$$q(s, t) := \mathbb{P}(m_1(\mathbf{e}) = s \mid |\mathbf{e}_V| = t)$$

$$q^{unif}(s, t) := \mathbb{P}(m_1(\mathbf{e}^{unif}) = s \mid |\mathbf{e}_V^{unif}| = t)$$

$$M(t) := \max_{0 \leq s \leq t} \frac{q^{unif}(s, t)}{q(s, t)}$$

On peut alors énoncer la proposition suivante :

Proposition 2.5. *Si **DecodeV** est uniforme en poids et si **DecodeU** est m_1 -uniforme, alors on a $\mathbf{e} \sim \mathbf{e}^{unif}$.*

Preuve. Soit \mathbf{e}'_U le vecteur obtenu à la sortie de la boucle dans **DECODE_UV**.
On remarque que pour tout i, j :

$$0 \leq r(i, j) = \frac{1}{M(i, j)} \frac{q^{unif}(i, j)}{q(i, j)} = \left(\inf_{0 \leq k \leq l} \frac{q(k, l)}{q^{unif}(k, l)} \right) \frac{q^{unif}(i, j)}{q(i, j)} \leq 1$$

Donc les coordonnées de r sont bien des probabilités. Notons pour tout $i, j \in \llbracket 1, n \rrbracket$:

$$q'(i, j) := \mathbb{P}(m_1(\mathbf{e}'_U) = i \mid |\mathbf{e}'_V| = j)$$

On a alors

$$\begin{aligned} q'(i, j) &= \frac{r(i, j)q(i, j)}{\sum_{0 \leq k \leq l} r(k, l)q(k, l)} \\ &= \frac{q^{unif}(i, j)}{M(i, j) \sum_{0 \leq k \leq l} \frac{1}{M(k, l)} q^{unif}(l)} \\ &= q^{unif}(i, j) \end{aligned} \tag{5}$$

La première ligne vient de l'indépendance du tirage du réel $\text{rand}([0, 1])$ du poids de \mathbf{e}_V et de $m_1(\mathbf{e})$. La dernière ligne vient du fait que la somme sur i, j des $q^{unif}(i, j)$ vaut 1. De plus la sortie de l'algorithme `DECODE_V` est uniforme, ce qui conclut la preuve. \square

2.3 Choix des algorithmes de décodage

description explicite de `DECODE_V`

description explicite de `DECODE_U`

Application de la méthode du rejet selon ces choix et choix des distributions.

2.4 Estimation du nombre de rejet

Algorithme 2 `DecodeU`($\varphi, \mathbf{e}_V, \mathbf{s}^U, \mathbf{H}_U$)

- 1: $t \leftarrow |\mathbf{e}_V|$
 - 2: $k_0 \leftarrow \mathcal{D}_U^t$
 - 3: **Faire**
 - 4: $\mathcal{I} \leftarrow$ ensemble d'information de $\langle \mathbf{H} \rangle^\perp$
 - 5: $\mathcal{J} \subset \mathcal{I}$ de taille $k - d$ tel que $|\mathbf{e}_V|_{\mathcal{J}} = k_0$
 - 6: $x_U \leftarrow \{x \in \mathbb{F}_3^{n/2} \mid \forall j \in \mathcal{J}, x_j \notin \{-\frac{b_i}{a_i} \mathbf{e}_{V_i}, -\frac{d_i}{c_i} \mathbf{e}_{V_i}\}\}$
 - 7: $\mathbf{e}_U \leftarrow \text{PRANGE}(\mathbf{H}_U, \mathbf{s}^U, \mathcal{I}, x_U)$
 - 8: **Tant que** $|\varphi(\mathbf{e}_U, \mathbf{e}_V)| \neq \omega$
 - 9: **Retourne** \mathbf{e}_U
-

EXPLIQUER POURQUOI ON AJOUTE LE d .

Avec $d \in [0, k_U]$, qui permettra de montrer que notre sortie est uniforme.

Algorithme 3 DecodeV(\mathbf{s}^V)

- 1: c mot aléatoire du code V
 - 2: $\mathbf{s} \leftarrow$ le syndrome \mathbf{s}^V paddé avec des zéros
 - 3: $\mathbf{e}_V \leftarrow \mathbf{s} + c$
 - 4: **Retourne** \mathbf{e}_V
-

Définition 2.6. (*Bon ensemble*). Soient $d \leq k \leq n$, $\mathbf{H} \in \mathbb{F}_3^{(n-k) \times n}$ et $\varepsilon \subseteq \llbracket 1, n \rrbracket$ de taille $k - d$. On dit que ε est un bon ensemble pour \mathbf{H} si \mathbf{H}_{ε} est de rang plein. Sinon, on dit que ε est un mauvais ensemble.

Nous voulons compter le nombre de rejets moyen, pour cela nous allons comparer \mathbf{e} la sortie de **DecodeUV** avec \mathbf{e}^{unif} une erreur aléatoire uniforme de poids ω .

Nous savons doré et déjà que la sortie de **DecodeV** est uniforme, nous allons donc étudier la sortie de **DecodeU**. Nous introduisons pour cela **VarDecodeU** qui fonctionne de la même façon que **DecodeU** quand \mathcal{J} est un bon ensemble pour \mathbf{H} et qui renvoie une erreur aléatoire selon la distribution \mathcal{D}_U^t sur \mathcal{J} dans le cas contraire. Il n'y a donc aucune raison que la sortie soit une solution du problème de décodage lorsque \mathcal{J} n'est pas un bon ensemble pour \mathbf{H} . Nous pouvons facilement voir que **VarDecodeU** est m_1 -uniforme.

La sortie de l'algorithme **DecodeUV** utilisant **VarDecodeU** est alors uniforme, on la note \mathbf{e}^{unif} .

Définition 2.7.

- $J_{x_V, l}^{unif}$ et un ensemble choisi uniformément tel que $J_{x_V, l}^{unif} \subseteq \llbracket 1, n/2 \rrbracket$, il est de cardinal $k_U - d$ et $\#J_{x_V, l}^{unif} \cap \text{Supp}(x_V) = k_0$.
- $J_{x_V, l}^{\mathbf{H}_U}$ est défini de la même façon avec une contrainte supplémentaire : il fait parti des bons ensembles pour \mathbf{H}_U .

Pour pouvoir compter le nombre de rejets, nous allons avoir besoin des lemmes suivants dont les preuves sont en annexes.

Lemme 2.8.

$$\rho(\mathbf{e}, \mathbf{e}^{unif}) \leq \sum_{x_V, l} \rho(J_{x_V, l}^{\mathbf{H}_U}, J_{x_V, l}^{unif}) \mathbb{P}(k_0 = l, \mathbf{e}_V = x_V)$$

Lemme 2.9.

$$\mathbb{E} \left(\rho \left(J_{x_V, l}^{unif}, J_{x_V, l}^{\mathbf{H}_U} \right) \right) \leq \frac{3^{-d}}{2} \quad (6)$$

Lemme 2.10. (*Inégalité de Markov*).

Soit Z une variable aléatoire supposée presque sûrement positive ou nulle, alors

$$\forall a > 0 \quad \mathbb{P}(Z > a) \leq \frac{\mathbb{E}(Z)}{a}$$

Nous pouvons maintenant énoncer le théorème suivant :

Théorème 2.11. Soit \mathbf{e} la sortie de `DecodeUV` et \mathbf{e}^{unif} la variable aléatoire uniformément distribuée parmi les mots de poids ω . On a :

$$\mathbb{P} \left(\rho(\mathbf{e}, \mathbf{e}^{unif}) > 3^{-d/2} \right) \leq \frac{3^{-d/2}}{2}$$

Preuve.

$$\begin{aligned} & \mathbb{P}(\rho(\mathbf{e}, \mathbf{e}^{unif}) > 3^{-d/2}) \\ & \leq 3^{d/2} \mathbb{E}(\rho(\mathbf{e}, \mathbf{e}^{unif})) \quad \text{par l'inégalité de Markov} \\ & \leq 3^{d/2} \mathbb{E} \left(\sum_{x_V, l} \rho \left(J_{x_V, l}^{\mathbf{H}_U}, J_{x_V, l}^{unif} \right) \mathbb{P}(k_0 = l, \mathbf{e}_V = x_V) \right) \quad \text{d'après le lemme 2.8} \\ & \leq 3^{d/2} \left(\sum_{x_V, l} \frac{3^{-d}}{2} \mathbb{P}(k_0 = l, \mathbf{e}_V = x_V) \right) \quad \text{par (6)} \\ & = 3^{d/2} \frac{3^{-d}}{2} \left(\sum_{x_V, l} \mathbb{P}(k_0 = l, \mathbf{e}_V = x_V) \right) \\ & = \frac{3^{-d/2}}{2} \end{aligned}$$

□

La probabilité d'avoir un rejet équivaut à la probabilité d'avoir une distance significative entre \mathbf{e} et \mathbf{e}^{unif} . D'après le théorème 2.11, nous voyons donc qu'en faisant varier d , nous serons en mesure d'effectuer très peu de rejets.

2.5 Une famille de fonctions uniformément distribuée

On a donc le point (2) de la définition des fonctions GPV qui est obtenu dans la section précédente. On va montrer le point (1), à savoir, notre famille de fonctions syndromes est uniformément distribuée avec les codes $(U, U+V)$ -généralisés.

TODO

3 Sécurité du schéma

Pour montrer la sécurité du schéma, nous allons dans un premier temps montrer que sous l'hypothèse que la matrice de parité du code considéré est difficile à distinguer d'une matrice aléatoire, le schéma est sûr au sens EUF-CMA.

Nous montrerons ensuite qu'il est effectivement difficile de distinguer notre matrice de parité permutée d'une matrice aléatoire.

3.1 Sécurité EUF-CMA

Nous allons montrer que le schéma est sûr au sens EUF-CMA (Existential Unforgeability under Chosen Message Attacks). Pour cela nous ferons une réduction au problème DOOM.

3.1.1 Définitions

Soit \mathcal{A} un adversaire ayant accès à N_{sign} signatures de son choix. Soit les trois algorithmes suivants :

Algorithme 4 $\text{Init}(\lambda)$

- 1: $(pk, sk) \leftarrow \text{Gen}(1^\lambda)$
 - 2: $\mathbf{H}_{pk} \leftarrow pk$
 - 3: $(\varphi, \mathbf{H}_U, \mathbf{H}_V) \leftarrow sk$
 - 4: **Retourne** \mathbf{H}_{pk}
-

Le jeu EUF-CMA se déroule comme suit. \mathcal{A} fait appel à **Init**. Il peut ensuite faire N_{sign} requêtes à **sign**. Le jeu est dit réussi si \mathcal{A} est capable de

Algorithme 5 $\text{Sign}(s)$

1: $\mathbf{e} \leftarrow \mathcal{D}_{\varphi, \mathbf{H}_U, \mathbf{H}_V}(\mathbf{s})$
2: **Retourne** \mathbf{e}

Algorithme 6 $\text{Fin}((s, e))$

1: **Retourne** $(\mathbf{e}\mathbf{H}_{pk}^T = s) \wedge (|\mathbf{e}| = \omega)$

donner (s, e) accepté par **Fin** et tel que s n'est jamais été demandé à **Sign**.
On définit alors le succès EUF-CMA comme :

$$Succ_{Wave}^{EUF-CMA}(t, N_{sign}) := \max_{\mathcal{A}; |A| \leq t} (\mathbb{P}(\mathcal{A} \text{ réussit le jeu EUF-CMA de Wave})).$$

Le protocole est alors sûr au sens EUF-CMA si ce succès est négligeable.

Nous souhaitons donc montrer que notre système est sûr au sens EUF-CMA. Pour cela, nous allons dans la section suivante majorer ce succès par rapport au succès d'un problème connu, le problème DOOM.

3.1.2 Réduction au problème DOOM

Le problème DOOM. Soient des paramètres (n, q, k, ω, N) , où N est un entier.

I : \mathbf{H} une matrice uniforme de $\mathbb{F}_q^{(n-k) \times n}$ et $(\mathbf{s}_1, \dots, \mathbf{s}_N)$ une liste de N syndromes.

Q : Décoder l'un des syndromes à la distance $w := \lfloor \omega n \rfloor$.

On définit alors le succès de DOOM comme :

$$Succ^{DOOM(n, q, k, N)}(t) := \max_{\mathcal{A}; |A| \leq t} (\mathbb{P}(\mathcal{A}(\mathbf{H}, \mathbf{s}_1, \dots, \mathbf{s}_N) = \mathbf{e} \text{ tel que}$$

$$\mathbf{e}\mathbf{H}^T = \mathbf{s}_j \text{ pour un certain } j \in \{1, \dots, N\})).$$

La réduction à ce problème est naturelle pour un schéma de signature puisque
EXPLICATION INFORMELLE DE LA REDUCTION, POURQUOI ELLE
VA MARCHER

3.1.3 Preuve formelle de la réduction

Afin de faire une preuve formelle de la sécurité EUF-CMA, nous allons changer le jeu en rajoutant une fonction de hachage. L'attaquant \mathcal{A} peut maintenant faire N_{hash} appelle à la fonction de hachage et ainsi obtenir des couples (m,s) . De plus, la fonction de signature prends maintenant en entrée un message quelconque. Elle prend ensuite un aléa r dans $\{0,1\}^{\lambda_0}$. Le tout est alors donné à la fonction de hachage qui renvoie un syndrome valide.

Nous allons aussi introduire un système de jeux qui nous permettra de réduire la sécurité d'un système à un problème P . Soit \mathcal{A} un attaquant et \mathcal{R} un rival. Soient G_0, G_1, \dots, G_N un ensemble de jeux et soit $\mathbb{P}(G_i)$ la probabilité pour \mathcal{A} de répondre au défi posé par \mathcal{R} pour le jeu G_i . $\mathbb{P}(G_0)$ est alors la probabilité de cassé le système considéré et $\mathbb{P}(G_N)$ la probabilité de répondre au problème P .

L'idée est de changer pas à pas les jeux G_0 à G_N de façon à ce que :

$$\forall i \in 0, \dots, N-1, |\mathbb{P}(G_i) - \mathbb{P}(G_{i+1})| \in \text{negl}(\lambda) \implies |\mathbb{P}(G_0) - \mathbb{P}(G_N)| \in \text{negl}(\lambda)$$

où λ est un paramètre de sécurité. Autrement dis, les changements sur les jeux ne changent qu'à un facteur négligeable près les probabilités de succès de l'attaquant \mathcal{A} .

Il n'est pas possible de changer le comportement de \mathcal{A} puisqu'il est quelconque, en revanche nous pouvons modifier celui de R .

Théorème 3.1. (*Réduction de sécurité*).

Soit N_{sign} le nombre de requêtes faites à l'oracle de signature. Soit λ le paramètre de sécurité et $\lambda_0 = \lambda + 2\log_2(N_{sign})$. On a :

$$\text{Succ}_{Wave}^{\text{EUF-CMA}}(t, N_{sign}) \leq 2\text{Succ}^{\text{DOOM}(n,q,k,N)}(t) + \rho(\mathcal{D}_{rand}, \mathcal{D}_{pub})(t) +$$

$$f(\mathcal{U}_\omega, \mathcal{D}_\omega^{\mathbf{H}^{pk}}) + g(\epsilon) + c + \frac{N_{hash}}{2}\sqrt{\epsilon} + \frac{1}{2^\lambda}$$

Preuve. On rappelle que G_0 correspond à notre jeu pour la sécurité EUF-CMA de Wave.

- G_1 : Le jeu G_1 est identique au jeu G_0 sauf si l'évènement

$$F := \{\text{Un même aléa } r \text{ a été tiré lors de deux requêtes d'un même message à l'oracle de signature}\}.$$

On a alors

$$\mathbb{P}(G_0) \leq \mathbb{P}(G_1) + \mathbb{P}(F)$$

Or pour $\lambda_0 = \lambda + 2\log_2(N_{sign})$, la probabilité que l'évènement F se produise est majorée par $\frac{1}{2^\lambda}$. C'est donc négligeable et le changement est autorisé.

- G_2 : Le passage au jeu G_2 permet d'empêcher \mathcal{A} de faire appel à l'oracle de signature sur les syndrome du problème DOOM. L'idée est de créer une liste suffisamment grande L_m d'aléas tous différents. On modifie alors la fonction **hash** de cette façon :

1. Si **hash** est appelée par la fonction **sign**, alors les aléas seront pris successivement dans L_m et associés à un vecteur erreur $\mathbf{e}_{m,r}$ (stocké) pris uniformément dans S_ω . Elle renvoie alors $\mathbf{s} = \mathbf{e}_{m,r}\mathbf{H}^T$.
2. En revanche si **hash** est appelée hors de la fonction **sign** par \mathcal{A} , alors elle son comportement dépendra de l'aléa. Si r est dans L_m elle se comporte comme si elle avait été appelée par **sign** et renvoie $\mathbf{e}_{m,r}\mathbf{H}^T$. Sinon elle renvoie successivement les syndromes du problème DOOM.

On prend donc dans la fonction **sign** toujours le r suivant de L_m . On a alors changé le jeu en supprimant le cas où deux mêmes r sont tirés lors de la signature. Cela ne pose pas de problème grace au passage à G_1 . Le passage au jeu G_2 permettra ainsi de s'assurer par la suite que \mathcal{A} n'a pas fait d'appel à **sign** sur les syndromes du problème DOOM. On a alors

$$\mathbb{P}(G_1) \leq \mathbb{P}(G_2) + \frac{N_{hash}}{2}\sqrt{\epsilon}$$

où ϵ est une fonction en n qui décroît exponentiellement. C'est donc bien négligeable.

- G_3 : Le jeu G_3 permet à l'oracle de signature de se passer de l'algorithme de décodage, et donc de la trappe T . Il sera nécessaire pour remplacer la matrice du code (U,U+V)-généralisé par la matrice aléatoire de l'instance du problème DOOM. Pour passer au jeu G_3 , on modifie la sortie de **sign**. Au lieu de renvoyer le couple (\mathbf{e}, r) où $\mathbf{e} = D_{\varphi, H_U, H_V}$, on renvoie le couple $(\mathbf{e}_{m,r}, r)$ préalablement stocké.

La différence de succès de dépend que de ω et des différence de distribution

entre \mathcal{U}_ω et $\mathcal{D}_\omega^{\mathbf{H}_{pk}}$, où \mathcal{U}_ω est la distribution uniforme sur S_ω et où \mathcal{U}_ω et $\mathcal{D}_\omega^{\mathbf{H}_{pk}}$ est la distribution des couples (e, r) où r est un aléa uniforme dans $\{0, 1\}^{\lambda_0}$ et e est la sortie de l'algorithme de décodage avec trappe sur une entrée s prise uniformément dans \mathbb{F}_q^{n-k} . On a alors

$$\mathbb{P}(G_2) \leq \mathbb{P}(G_3) + f(\mathcal{U}_\omega, \mathcal{D}_\omega^{\mathbf{H}_{pk}}) + g(\epsilon) + c$$

où f et g sont linéaires et c une certaine constante.

- G_4 : On peut maintenant remplacer \mathbf{H}_{pk} par \mathbf{H}_0 . Ce changement ne pose pas de problème puisque **sign** n'utilise plus la trappe. En revanche, nous avons créé un distingueur entre la distribution ($:= \mathcal{D}_{rand}$) des matrices prises aléatoirement dans $\mathbb{F}_q(n-k) \times n$ et la distribution ($:= \mathcal{D}_{pub}$) des matrices prises aléatoirement dans l'ensemble des matrices de parité d'un code $(U, U+V)$ -généralisé où U (resp. V) est un $[n/2, k_U]$ -code (resp. $[n/2, k_V]$ -code). On a alors

$$\mathbb{P}(G_3) \leq \mathbb{P}(G_4) + \rho(\mathcal{D}_{rand}, \mathcal{D}_{pub})(t)$$

- G_5 : On change ici la procédure de fin. On rajoute à la vérification la condition $r \notin L_m$. Ainsi on est bien sûr que \mathcal{A} réussit le jeu s'il répond au problème DOOM. Alors la probabilité que \mathcal{A} réussisse G_5 est exactement la probabilité que \mathcal{A} réussisse G_4 et $r \notin L_m$. On a alors

$$\mathbb{P}(G_4) \leq 2\mathbb{P}(G_5) + \rho(\mathcal{D}_{rand}, \mathcal{D}_{pub})(t)$$

où $\mathbb{P}(G_5)$ est exactement la probabilité pour \mathcal{A} de renvoyer $\mathbf{e}_j \in S_\omega$ et tel que $\mathbf{e}_j \mathbf{H}_0^T = \mathbf{s}_j$ pour un certain indice j du problème DOOM. On a donc

$$\mathbb{P}(G_5) \leq Succ_{DOOM}^{n,k,N_{hash},\omega}(t)$$

En rassemblant toutes les inégalités on termine la preuve. (On trouvera le détail des preuves de probabilité en annexe de ce rapport.) \square

3.2 Indistinguabilité des codes $(U, U+V)$ -généralisés

Distinguer une matrice de parité d'un code $(U, U+V)$ -généralisé d'une matrice de parité aléatoire.

Réduction à un problème NP-complet.

Utilisation de S et P pour masquer les propriétés de la matrice.

Conclusion

ANNEXES

Preuve du lemme 2.8. Nous allons avoir besoin dans la suite de la proposition suivante :

Proposition 3.2. *Soient X et Y deux variables aléatoires dans le même espace A et Z une variable aléatoire dans l'espace B indépendante de X et Y . Alors, pour toute fonction f , nous avons :*

$$\rho(f(X, Z), f(Y, Z)) \leq \rho(X, Y)$$

$$\begin{aligned} \rho(\mathbf{e}, \mathbf{e}^{unif}) &= \rho((\mathbf{e}_U, \mathbf{e}_V), (\mathbf{e}_U^{unif}, \mathbf{e}_V^{unif})) \\ &\leq \sum_{x_U, x_V} \left| \mathbb{P}((\mathbf{e}_U, \mathbf{e}_V) = (x_U, x_V)) - \mathbb{P}((\mathbf{e}_U^{unif}, \mathbf{e}_V^{unif}) = (x_U, x_V)) \right| \\ &= \sum_{x_U, x_V} \left| \mathbb{P}(\mathbf{e}_V = x_V) \mathbb{P}(\mathbf{e}_U = x_U | \mathbf{e}_V = x_V) - \mathbb{P}(\mathbf{e}_V^{unif} = x_V) \mathbb{P}(\mathbf{e}_U^{unif} = x_U | \mathbf{e}_V^{unif} = x_V) \right| \\ &= \sum_{x_U, x_V} \mathbb{P}(\mathbf{e}_V = x_V) \left| \mathbb{P}(\mathbf{e}_U = x_U | \mathbf{e}_V = x_V) - \mathbb{P}(\mathbf{e}_U^{unif} = x_U | \mathbf{e}_V^{unif} = x_V) \right| \end{aligned}$$

Étudions le terme $\mathbb{P}(\mathbf{e}_U = x_U | \mathbf{e}_V = x_V) - \mathbb{P}(\mathbf{e}_U^{unif} = x_U | \mathbf{e}_V^{unif} = x_V)$:

$$\begin{aligned} \mathbb{P}(\mathbf{e}_U = x_U | \mathbf{e}_V = x_V) - \mathbb{P}(\mathbf{e}_U^{unif} = x_U | \mathbf{e}_V^{unif} = x_V) &= \\ \sum_l \left[\mathbb{P}(\mathbf{e}_U = x_U | k_0 = l, \mathbf{e}_V = x_V) - \mathbb{P}(\mathbf{e}_U^{unif} = x_U | k_0 = l, \mathbf{e}_V^{unif} = x_V) \right] \mathbb{P}(k_0 = l | \mathbf{e}_V = x_V) \end{aligned}$$

Les ensembles \mathcal{J} étant inclus dans un ensemble d'information, l'aléa interne de `DecodeU` et de `VarDecodeU` ne dépendent pas du choix de $J_{x_V, l}^{unif}$ et $J_{x_V, l}^{\mathbf{H}_U}$. Nous pouvons voir que $\mathbb{P}(\mathbf{e}_U = x_U | k_0 = l, \mathbf{e}_V = x_V)$ ne dépend que de x_U et $J_{x_V, l}^{\mathbf{H}_U}$ et $\mathbb{P}(\mathbf{e}_U^{unif} = x_U | k_0 = l, \mathbf{e}_V^{unif} = x_V)$ ne dépend que de x_U et $J_{x_V, l}^{unif}$. De plus, $J_{x_V, l}^{unif}$ et $J_{x_V, l}^{\mathbf{H}_U}$ vivent dans le même espace. Ainsi d'après la proposition 3.2, nous avons

$$\sum_{x_U} \mathbb{P}(\mathbf{e}_U = x_U | k_0 = l, \mathbf{e}_V = x_V) - \mathbb{P}(\mathbf{e}_U^{unif} = x_U | k_0 = l, \mathbf{e}_V^{unif} = x_V) \leq \rho(J_{x_V, l}^{\mathbf{H}_U}, J_{x_V, l}^{unif})$$

$$\rho \left(J_{x_V, l}^{unif}, J_{x_V, l}^{\mathbf{H}_U} \right) = \frac{N_{x_V, l}}{\binom{|x_V|}{l} \binom{n/2 - |x_V|}{k_U - d - l}} \quad (7)$$

Avec $N_{x_V, l}$ le nombre de mauvais sous-ensembles de $\llbracket 1, n/A \rrbracket$ de taille $k_U - d$ pour \mathbf{H}_U . \square

Preuve de la proposition 1.4. La dimension d'un *hull* étant invariante par permutation nous allons ici montrer que nous avons $\dim(\text{hull}(U, U + V)) = k_U - k_V$ avec probabilité $1 - \mathcal{O}(2^{k_U - k_V})$. Par définition du *hull*, nous avons :

$$\begin{aligned}\text{hull}(U, U + V) &= (U, U + V) \cap (U, U + V)^\perp \\ &= (U, U + V) \cap (V^\perp + U^\perp, V^\perp)\end{aligned}$$

Ainsi $\forall u \in U$ et $\forall v \in V$ tels que $(u, u + v) \in \text{hull}(U, U + V)$, il existe $u_t \in U^\perp$ et $v_t \in V^\perp$ tels que

$$\begin{cases} u = v_t + u_t \\ u + v = v_t \end{cases} \iff \begin{cases} v = u_t \\ u + v = v_t \end{cases}$$

Donc $v \in V \cap U^\perp$. De plus, nous avons $\dim(V) + \dim(U^\perp) = \frac{n}{2} + k_V - k_U < \frac{n}{2}$
SUITE À ÉCRIRE \square