

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

<b>Introduction</b>	<b>2</b>
<b>1 Le schéma de signature Wave</b>	<b>2</b>
1.1 La famille de codes $(U, U+V)$ -généralisés . . . . .	2
1.2 Le principe de signature . . . . .	4
1.3 Le décodage avec trappe . . . . .	6
1.4 Implémentation et choix de paramètres . . . . .	9
<b>2 Uniformisation des signatures et syndromes</b>	<b>9</b>
2.1 Une fuite d'information . . . . .	9
2.2 La méthode du rejet . . . . .	10
2.3 Choix des algorithmes de décodage . . . . .	12
2.4 Estimation du nombre de rejet . . . . .	12
2.5 Une famille de fonctions uniformément distribuée . . . . .	13
<b>3 Sécurité du schéma</b>	<b>13</b>
3.1 Sécurité EUF-CMA . . . . .	13
3.1.1 Définitions . . . . .	13
3.1.2 Réduction au problème DOOM . . . . .	14
3.1.3 Preuve formelle de la réduction . . . . .	14
3.2 Indistinguabilité des codes $(U, U+V)$ -généralisés . . . . .	17
<b>Conclusion</b>	<b>17</b>

# Introduction

- passage au post-quantique
  - appel d'offre NIST
  - > tableau : aucun code correcteur en signatures
  - dur de trouver l'ensemble des syndromes facilement décodable
  - dur de créer une fonction de hachage qui envoie  $m$  dans l'ensemble des syndromes possibles
  - problème du décodage NP-complet
  - mot  $y$  de syndrome  $s$  est associé à un unique mot de code  $c$  le plus proche de  $y$
- quand on chiffre -> ne pose pas de problème  
quand on signe -> pose un problème car il est dur de trouver un syndrome de cette sorte
- la solution Wave est d'enlever la restriction au mot le plus proche  
Nous allons détailler le schéma de signature Wave et détailler sa sécurité.

## 1 Le schéma de signature Wave

Pour répondre aux problèmes :

- Des codes UV-généralisés
- Des fonctions GPV en moyenne
- Un schéma de signature de type hash et signe utilisant ces codes

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

Définition des codes  $(U, U+V)$ -généralisés:

- Comment les créer FAIT
- Choix des paramètres  $a, b, c, d$  À DÉTAILLER
- Liens entre les matrices des codes  $U$  et  $V$  et du code  $UV$  FAIT
- Les dimensions et différents paramètres EN COURS
- Calcul du hull ==>  $q > 2$  TODO
- ...?

**Définition 1.1.** Soient  $U$  et  $V$  deux codes de même longueur  $n/2$  et de dimension respectives  $k_u$  et  $k_v$ . Un code  $(U, U + V)$  est un code de longueur  $n$  et de dimension  $k = k_u + k_v$  et tel que :

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

**Définition 1.2.** (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  $i \in 1, n/2$  :

$$a_i c_i \neq 0$$

$$a_i d_i - b_i c_i \neq 0$$

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  $y_i$ .

**Remarque 1.3.** Dans la suite, on prend  $a, b, c, d$  tels que

$$a_i d_i - b_i c_i = 1 \text{ pour tout } i \in 1, n/2.$$

**Proposition 1.4.** 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 diagonales 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

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

Par définition des matrices  $G_U$  et  $G_V$ , 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  $i \in 1, n/2$ , 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.  $\square$

$q = 2 \rightarrow$  calcul du hull  $\rightarrow$  fuite d'info.

On pose donc  $q = 3$  pour toute la suite du rapport.

## 1.2 Le principe de signature

Un schéma hash et signe utilisant la fonction syndrome comme fonction à sens unique :

- Définition des fonctions GPVM, un couple (Trapdoor, InvertAlg) où trapdoor est un algo poly proba renvoyant une matrice de parité et la trappe associée, et où InvertAlg est un algo poly proba prenant en entrée la trappe et renvoyant l'inverse de la fonction syndrome.

De plus, ces fonctions sont (1) bien distribuées, (2) sans fuite d'info en moyenne, (3) sens unique sans la trappe

- Le schéma : un algo signe et un algo verify.

Notre schéma de signature utilisera donc les codes  $(U, U + V)$ -généralisés et la fonction syndrôme 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 1.5.** Soient  $X$  et  $Y$  deux variables aléatoires à valeurs dans le même espace  $\epsilon$ . 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 1.6.** (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  $\lambda$ , tels que :

- **Trapdoor** est un algorithme probabiliste et polynomial en  $1^\lambda$  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  $\omega$  tel que  $eH^T = s$ .

De plus, pour *presque toutes* matrice  $H$  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  $\omega$  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  $\omega$  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.

**Sign**<sup>sk</sup>(s):

e ← InvertAlg(s,T)  
renvoie e

**Verify**<sup>pk</sup>(s,e'):

Si e'H<sup>T</sup> = s et |e'| = ω  
renvoie 1  
renvoie 0

### 1.3 Le décodage avec trappe

Détail de l'algorithme invertAlg avec utilisation de la trappe:

- Conditions sur le poids de e:
  - > facile
  - > facile avec trappe
  - > difficile
- Inverser le syndrome sur le code UV <==> inverser le syndrome sur U et sur V et prendre son image par Phi.
- Prendre un ev par un algo de décodage quelconque, utiliser les propriétés du code UV pour en déduire un eu, vérifier le poids de e, recommencer.
- Différences gros poids et petits poids

En partant de l'hypothèse que la matrice de parité **H** du code (U, U + V)-généralisé ressemble à une matrice aléatoire, la difficulté de créer une fausse signature sans connaître la trappe **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, et discuter sa difficulté en fonction du poids ω de e.

Notons  $\mathcal{S}_{\omega,n}$  l'ensemble des mots de poids ω et de longueur n. On notera  $\mathcal{S}_\omega$  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 :

$$\omega^- := ??? \quad \text{et} \quad \omega^+ := ???.$$

*Preuve.* TODO □

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

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

INSERER SCHEMA !!

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 1.2, 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 1.7.** 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**( $\mathbf{s}, \mathbf{T}$ ) :  
 $(\mathbf{s}_U, \mathbf{s}_V) = \mathbf{s}$   
 $\mathbf{e}_U = \text{DECODE\_U}(\mathbf{s}_U)$   
 $\mathbf{e}_V = \text{DECODE\_V}(\mathbf{s}_V)$   
renvoie  $\varphi_{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}}(\mathbf{e}_U, \mathbf{e}_V)$

Si l'on choisit un algorithme générique pour **DECODE\_U** et **DECODE\_V**, alors nous obtiendrons un vecteur  $\mathbf{e}$  de poids  $\omega$  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 1.8.** 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  $\mathbf{e}$ . On aura alors :

**InvertAlg**( $\mathbf{s}, \mathbf{T}$ ) :  
 $(\mathbf{s}_U, \mathbf{s}_V) = \mathbf{s}$   
 $\mathbf{e}_V = \text{DECODE\_V}(\mathbf{s}_V)$   
 $\mathbf{e}_U = \text{DECODE\_U}(\mathbf{s}_U, \mathbf{e}_V)$   
renvoie  $\varphi_{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}}(\mathbf{e}_U, \mathbf{e}_V)$

**Proposition 1.9.** 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  $\mathbf{e}_U$

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

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

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

*Preuve.* TODO

□

**Proposition 1.10.** 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  $\mathbf{e}_U$

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

On a alors

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

*Preuve.* TODO

□

On récapitule les différents cas dans la figure REF.

INSERER GRAPHIQUE

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 1.11.** Fonction de hash

## 1.4 Implémentation et choix de paramètres

TODO

Détail de `DECODE_U` et `DECODE_V` + choix des paramètres et choix d'implémentation + résultats

## 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 :

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

PREUVE

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

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

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

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

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.* TODO □

Ainsi, pour que  $\mathbf{e}$  soit uniformément distribué sur  $S_\omega$ , il suffit de choisir **DECODE\_V** 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 **DECODE\_U** de façon à ce que  $m_1(\mathbf{e})$  conditionnée à  $|\mathbf{e}_V|$  soit distribué comme  $m_1(\mathbf{e}^{unif})$  conditionnée à  $|\mathbf{e}_V^{unif}|$ .

**Proposition 2.5.** Soit l'algorithme :

```

DECODE_UV ( $\mathbf{H}_V, \mathbf{H}_V, \varphi, \mathbf{s}$ ) :
   $\mathbf{e}_V \leftarrow \text{DECODE\_V}(\mathbf{H}_V, \mathbf{s}^V)$ 
  Faire
     $\mathbf{e}_U \leftarrow \text{DECODE\_U}(\mathbf{H}_U, \mathbf{s}^U, \varphi, \mathbf{e}_V)$ 
     $\mathbf{e} \leftarrow \varphi(\mathbf{e}_U, \mathbf{e}_V)$ 
  Tant que  $\text{rand}([0, 1]) \leq r(|\mathbf{e}_V|), m_1(\mathbf{e})$ 
  retourner  $\mathbf{e}$ 

```

Où :

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

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

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

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

Alors si `DECODE_V` est uniforme en poids et si `DECODE_U` est  $m_1$ -uniforme, on a  $\mathbf{e} \sim \mathbf{e}^{unif}$ .

*Preuve.* TODO

□

## 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

TODO

## 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 syndrômes est uniformément distribuée avec les codes  $(U, U+V)$ -généralisés

## 3 Sécurité du schéma

Deux problèmes :

- Distinction d'une matrice de parité d'un code  $(U, U+V)$ -généralisé permutée d'une matrice aléatoire
- Sécurité EUF-CMA du système si  $H$  ressemble à 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.

**Définition 3.1.** (Modèle de sécurité EUF-CMA). On définit 3 algorithmes :

<b>Init:</b> $(pk, sk) \leftarrow \text{Gen}(1^\lambda)$ $\mathbf{H}_{pk} \leftarrow pk$ $(\varphi, \mathbf{H}_U, \mathbf{H}_V) \leftarrow sk$ <b>renvoie</b> $\mathbf{H}_{pk}$	<b>Sign(<math>s</math>):</b> $\mathbf{e} \leftarrow \mathcal{D}_{\varphi, \mathbf{H}_U, \mathbf{H}_V}(s)$ <b>renvoie</b> $\mathbf{e}$
	<b>Fin(<math>s, e</math>):</b> <b>renvoie</b> $(\mathbf{e}\mathbf{H}_{pk}^T = s) \wedge ( \mathbf{e}  = \omega)$

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

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

**Définition 3.2.** (Le problème DOOM). Soient des paramètres  $(n, q, k, \omega, N)$ , 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 :

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

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

La réduction à ce problème est naturelle pour un schéma de signature puisque JHFEKHZFKHFKZEHFILZEJHFKLZEJFHJ.

EXPLICATION INFORMELLE DE LA REDUCTION, POURQUOI ELLE VA MARCHER

### 3.1.3 Preuve formelle de la réduction

Pour faire une preuve formelle de cette réduction, nous allons 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

considérer 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  $\lambda$  est un paramètre de sécurité. Autrement dit, 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  $R$ .

EVENTUELLEMENT METTRE LES 3 conditions mais je ne pense pas que ça soit nécessaire.

**Théorème 3.3.** (Réduction de sécurité).

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

$$\begin{aligned} \text{Succ}_{\text{Wave}}^{\text{EUF-CMA}}(t, N_{\text{sign}}) &\leq 2\text{Succ}^{\text{DOOM}(n,q,k,N)}(t) + \rho(\mathcal{D}_{\text{rand}}, \mathcal{D}_{\text{pub}})(t) + \\ &\quad f(\mathcal{U}_\omega, \mathcal{D}_\omega^{\mathbf{H}^{pk}}) + g(\epsilon) + c + \frac{N_{\text{hash}}}{2} \sqrt{\epsilon} + \frac{1}{2^\lambda} \end{aligned}$$

*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  $\lambda_0 = \lambda + 2 \log_2(N_{\text{sign}})$ , 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  $S_\omega$ . 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  $\epsilon$  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  $\omega$  et des différence de distribution entre  $\mathcal{U}_\omega$  et  $\mathcal{D}_\omega^{\mathbf{H}_{pk}}$ , où  $\mathcal{U}_\omega$  est la distribution uniforme sur  $S_\omega$  et où  $\mathcal{U}_\omega$  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