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 Lansade
Eva Palandjian

Encadrant:
Gilles Zemor

Février, 2020

Contents

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 DETAILLER
- 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_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 $i \in 1, n/2$, 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. □

$q = 2$ –> calcul du hull –> 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 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 1.6. (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 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 syndrôme $s \in \mathbb{F}_q^{n-k}$, et renvoyant $e \in \mathbb{F}_q^n$ de poids ω tel que $eH^T = s$.

De plus, pour *presque toutes* 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 ω et de longueur n et s est pris uniformément dans $\mathbb{F}_q^{n-k}$.
2. sans fuite d'information *en moyenne* :
   $\rho(\texttt{InvertAlg}(s, T), e) \in \text{negl}(\lambda)$ 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.

$\texttt{Sign}^{sk}(s)$:
  e $\leftarrow$ `InvertAlg`(s,T)
  `renvoie` e

$\texttt{Verify}^{pk}(s, e')$:
  `Si` $e'H^T = s$ `et` $|e'| = \omega$
    `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 $\mathbf{H}$ 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 syndrôme, 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'ambiguité sur la longueur. On rappelle que l'agorithme `InvertAlg` cherche à inverser la fonction syndrôme :

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

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 :

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

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 :

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

Preuve. TODO □

Nous voulons donc un moyen d'inverser la fonction syndrôme pour $\omega \in \{\omega^-_{UV}, \omega^+_{UV}\}$ avec ω^-_{UV} et ω^+_{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 :

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

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 :

$$\begin{array}{rccc} \varphi^{-1}_{\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}} & : & \mathbb{F}_q^{n/2} \times \mathbb{F}_q^{n/2} & \to & \mathbb{F}_q^{n/2} \times \mathbb{F}_q^{n/2} \\ & & (\mathbf{x}, \mathbf{y}) & \mapsto & (\mathbf{d}.\mathbf{x} - \mathbf{b}.\mathbf{y}, -\mathbf{c}.\mathbf{x} + \mathbf{a}.\mathbf{y}) \end{array}$$

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

$$(\mathbf{e_U}, \mathbf{e_V}) = \varphi^{-1}_{\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}}(\mathbf{e}).$$

Proposition 1.7. Inverser $f_{\omega,\mathbf{H}}$ pour un certain $\mathbf{s} \in F_q^{n-k}$ est équivalent à trouver $\mathbf{e} \in \mathbb{F}_q^n$ tel que:

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

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 on aura :

$\texttt{InvertAlg}(\mathbf{s}, \mathbf{T})$:
$(\mathbf{s}_U, \mathbf{s}_V) = s$
$\mathbf{e}_U = \texttt{DECODE\_U}(\mathbf{s}_U)$
$\mathbf{e}_V = \texttt{DECODE\_V}(\mathbf{s}_V)$
`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 $(U, U+V)$-généréralisés 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 :

$\texttt{InvertAlg}(\mathbf{s}, \mathbf{T})$:
$(\mathbf{s}_U, \mathbf{s}_V) = s$
$\mathbf{e}_V = \texttt{DECODE\_V}(\mathbf{s}_V)$
$\mathbf{e}_U = \texttt{DECODE\_U}(\mathbf{s}_U, \mathbf{e}_V)$
`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 nécéssaire 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 $\mathbf{e} = (\mathbf{e_U}, \mathbf{e_U} + \mathbf{e_V})$

$$|\mathbf{e_V}| = \#\left\{1 \leq i \leq n/2 \mid e_i \neq e_{i+n/2}\right\}$$

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 suffisemment de signatures, il pourra donc en analysant la fréquence des $\mathbf{e}_i \neq \mathbf{e}_j$ retrouver cette permutation. Il est donc nécéssaire 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 a 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

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

Avant d'expliciter nos algorithmes, il est nécéssaire 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_ω, il suffit de choisir `DECODE_V` de façon à ce que ses sorties soient uniforment 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 (H_V, H_V, φ, s) :
    e_V ← DECODE_V (H_V, s^V)
    Faire
        e_U ← DECODE_U (H_U, s^U, φ, e_V)
        e ← φ(e_U, e_V)
    Tant que rand([0,1]) ≤ r(|e_V|), m_1(e))
    retourner 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 definition 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 :

`Init`:
  $(pk, sk) \leftarrow \texttt{Gen}(1^\lambda)$
  $\mathbf{H}_{pk} \leftarrow pk$
  $(\varphi, \mathbf{H}_U, \mathbf{H}_V) \leftarrow sk$
  `renvoie` $\mathbf{H}_{pk}$

`Sign`(s):
  $\mathbf{e} \leftarrow \mathcal{D}_{\varphi,\mathbf{H}_U,\mathbf{H}_V}(s)$
  `renvoie` $\mathbf{e}$

`Fin`(s, e):
  `renvoie` $(\mathbf{e}\mathbf{H}_{pk}^T = s) \wedge (|\mathbf{e}| = \omega)$

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, ω, N), où N est un entier.

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

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

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

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

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 $G_0, G_1, ..., G_N$ 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 :

$$\forall i \in 0, ..., N-1, |\mathbb{P}(G_i) - \mathbb{P}(G_i+1)| \in negl(\lambda) \Longrightarrow |\mathbb{P}(G_0) - \mathbb{P}(G_N)| \in negl(\lambda)$$

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

Théorème 3.3. (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 :

$$Succ_{Wave}^{EUF-CMA}(t, N_{sign}) \leq 2Succ^{DOOM(n,q,k,N)}(t) + SMT$$

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

G_1 : Changer le jeu si lors d'un appelle à la fonction sign, deux r identiques sont choisis pour un même message m. S0 + P(F) négligeable

- G_2 : On crée une liste suffisemment grande de Lm, des sel, tous différent. Dans la fonction hash, si r est dans la liste, alors on choisit un e unifirme dans Sw, on renvoit son syndrome. e est stocké et au même couple (r, m) le même syndrome sera renvoyé. Si r n'est pas la dans liste, on renvoit un des syndromes de doom
  Dans la fonction sign, on prend toujours le r suivant dans Lm. (ne pose pas de pb grace au passage à G_1 Ainsi dans sign, on ne signera jamais les syndromes de DOOM. On s'assure donc que le fait d'avoir un oracle à sign dans UEF-CMA ne change pas la sécu face à DOOM.

- G_3 : Ici, on enlève l'utilisation de la trappe uniquement pour les syndromes non-DOOM, càd dans la fonction sign. On n'appelle plus la fonction de décodage D. A la place, on écupère le e stocké

- G_4 : On remplace Hpk par H0. Pas de pb car plus de trappe

- G_5 : On verifie que r n'est pas dans Lm, ainsi on est bien sûr que A a signé un message de DOOM ET qu'il connait nécessairement la trappe.

□

3.1.3 Et la fonction de hachage ?

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