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

- probleme 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 -> pb 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 -> ok quand on signe -> pas ok car dur de trouver un syndrome de cet sorte du coup la solution de machinx est d'enlever la restriction au mot le plus proche -> Wave innovation Nous allons détailler le schéma de signature Wave et détailler sa sécurité.

1 Le schéma de signature Wave

Pour répondre aux problèmes :
- Des fonctions GPV en moyenne - Un schéma de signature utilisant les codes (U,U+V)-généralisés

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

Définition des codes (U,U+V)-généralisés:
- Comment les créer
- Choix des paramètres a,b,c,d
- Liens entre les matrices des codes U et V et du code UV
- Les dimensions et différents paramètres
- Calcul du hull ==> q > 2
- ...?

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.

1.3 Le décodage avec trappe

Détail de l'algorithme invertAlg avec utilisation de la trappe:
- Les différents ω
- 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

1.4 Implémentation et choix de paramètres

TODO

2 Fuite d'information -> rejet

2.1 Une fuite d'information

- Malheureusement, fuite d'information en raison des correspondance entre e[i] et e[i+n/2] !
- Calcul de proba
- Pourquoi c'est problématique

2.2 La méthode du rejet

Idée générale : On attent pour que les sorties aient l'air uniforme (soient suffisemment proche de l'uniforme):
- On choisit ev de façon a ce qu'il soit uniforme dans son ensemble
- On met des conditions de rejets sur eu en fonction de ev pour que eu ait l'air uniforme

- On obtient un e qui a l'air uniforme

2.3 Estimation du nombre de rejet

TODO

2.4 Une famille de fonctions uniformément distribuée

On a donc le point (2) de la def des fonctions GPV qui est ok. On va montrer le point (1), à savoir nos fonctions sont bien distribuées avec les codes (U,U+V)-généralisés

3 Sécurité du schéma

3.1 Sécurité EUF-CMA

3.1.1 Définition sécurité EUF-CMA

3.1.2 Réduction au problème DOOM

3.1.3 Et la fonction de hachage ?

3.2 Distinction d'une matrice de parité d'un code (U,U+V) et d'une matrice aléatoire

réduction à un pb NP-complet ne pas oublier de rappeler que normalement on ajoute S et P pour masquer la forme de la matrice.

Conclusion