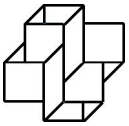

Laboratório Nacional de Computação Científica

Programa de Pós Graduação em Modelagem Computacional

Criptografia Pós-Quântica

Por

Juan del Carmen Grados Vasquez

PETRÓPOLIS, RJ - BRASIL

SETEMBRO DE 2016

CRIPTOGRAFIA PÓS-QUÂNTICA

Juan del Carmen Grados Vasquez

TESE SUBMETIDA AO CORPO DOCENTE DO LABORATÓRIO NACIONAL DE COMPUTAÇÃO CIENTÍFICA COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÃªNCIAS EM MODELAGEM COMPUTACIONAL

Aprovada por:

_____

Prof. Portugal Renato, D.Sc

(Presidente)

_____

Prof. Clemente Augusto Souza Tanajura, Ph.D

_____

Prof. Eduardo LÃºcio Mendes Garcia, D.Sc.

_____

Prof. Fulano de Tal, D.Sc.

_____

Prof. Ciclano de Tal, Ph.D

_____

Prof. Beltrano de Tal, Ph.D

PETRÓPOLIS, RJ - BRASIL
SETEMBRO DE 2016

Grados Vasquez, Juan del Carmen

XXXX criptografia pós-quântica / Juan del Carmen Grados Vasquez. Petropólis, RJ. : Laboratório Nacional de Computação Científica, 2016.

xx, yy p. : il.; 29 cm

Orientador: Portugal Renato

Tese (D.Sc.) – Laboratório Nacional de Computação Científica, 2016.

1. HashBased Cryptography. 2. Multivariate. 3. Coding Theory. 4. palava chave. I. Renato, Portugal. II. LNCC/MCT. III. Título.

CDD XXX.XXX

e.pí.gra.fe

s. f. 1. Sentença ou divisa posta no frontispício de um livro ou capítulo, no começo de um discurso ou de uma composição poética. 2.Inscrição posta em lugar visível de um edifício.

(Fonte: Dic. Aurélio.)

To my wife and daugther

for her love

Agradecimentos

O autor manifesta reconhecimentos às pessoas e instituições que colaboraram para a execução de seu trabalho.

Resumo da Tese apresentada ao LNCC/MCT como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

CRIPTOGRAFIA PÓS-QUÂNTICA

Juan del Carmen Grados Vasquez

Setembro , 2016

Orientador: Portugal Renato, D.Sc

Aqui entra o resumo de seu trabalho em português.

Abstract of Thesis presented to LNCC/MCT as a partial fulfillment of the requirements for the degree of Doctor of Sciences (D.Sc.)

POSTQUANTUM CRYPTOGRAPHY

Juan del Carmen Grados Vasquez

September, 2016

Advisor: Portugal Renato, D.Sc

Here, you type your abstract in english

Contents

List of Figures

Figure

List of Tables

Table

Lista de Siglas e Abreviaturas

- 1ME4: cristal da cruzipaína 1, obtido do PDB
- 1ME4m: modelo da cruzipaína 2, gerado a partir do cristal 1ME4
- 24-SMT: 24-metiltransferase
- 3D: Tridimensional
- 9PAP: cristal da papaína, obtido do PDB
- $\mathring{A}$: Angstroms
- AK: Arginina cinase
- ALA: Alanina
- ARG: Arginina
- ASN: Asparagina
- ASP: Ácido aspártico
- BATS: *BLAST Automatic Targeting for Structures*
- BLAST: *Basic Local Alignment Search Tool*
- C_α: Carbono-α
- CDS: Sequência codificante
- Cl^-: Íon Cloro
- CSO: S-Hidroxicisteína
- CYP51: Citocromo-P450 em *T. cruzi*
- CYS: Cisteína
- DM: Dinâmica Molecular
- DNA (ou ADN): Ácido Desoxirribonucléico
- EC: Classificação Enzimática

- ED_{50}: Dose Média Efetiva
- FTase: Farnesiltransferase
- GLN: Glutamina
- GLU: Ácido glutâmico
- gp = GP: Glicoproteína
- GPI: Glicosilfosfatidilinositol
- GR: glutationa redutase
- GRSI: *Gap Relative Strengh Index*
- *H. sapiens* = *h* = HSA: *Homo sapiens*
- HIS: Histidina
- IC_{50}: Concentração Média Inibitória
- ID_{50}: Dose Média Inibitória
- K_i: Constante de dissociação
- KEGG: *Kyoto Encyclopedia of Genes and Genomes*
- LC_{50}: Concentração Média Letal
- LEU: Leucina
- LIT: *Liver Infusion Tryptose*
- LVI: *Lenght Variation Index*
- LYS: Lisina
- MET: Metionina
- MIC: Concentração Mínima Inibitória
- Na^+: Íon Sódio
- NADH: estado reduzido da Nicotinamida Adenina Dinucleotídeo (NAD)
- NC-IUBMB: *Nomenclature Committee of the International Union of Biochemistry and Molecular Biology*
- PDB: *Protein Data Bank*
- PDF: Função Densidade de Probabilidade
- PHE: Fenilalanina

- PRO: Prolina
- RMN: Ressonância Magnética Nuclear
- RMSD: Desvio Médio Quadrático
- Rx: Raios-X
- SAS: Superfície Acessível ao Solvente
- SER: Serina
- SQS: Esqualeno sintase
- S-S: Ligação dissulfídica
- *T. cruzi* = *Tc* = TCR: *Trypanosoma cruzi*
- TR: Tripanotiona redutase
- TRP: Triptofano
- TYR: Tirosina
- TS: Trans-sialidase
- $T[SH]_2$: Tripanotiona (N^1,N^8-bis(glutationil)spermidina) redutase
- UNG: Uracil-DNA glicosilase
- VAL: Valina

Chapter 1

Introduction

1.1 Contribution of this thesis

The original todo note withouth changed colours.
Here's another line.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Is this correct?

I'm unsure about also!

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Change this!

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

This can help me in chapter seven!

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

This really needs to be improved!

What was I thinking?!

The following section needs to be rewritten!

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Notes

Part

Background

Chapter 2

Hash Based Cryptography

Chapter 3

Coding Based Cryptography

Chapter 4

SAT Problem

In our thesis, we focused the security of multivariate cryptography from the analysis of the boolean satisfiability problem, which is also known as the SAT problem. Thus, before to develop multivariate cryptography, we focus in the concepts of the SAT problem and we start our discussion by defining itself. After, we will describe a special case of the SAT problem which is widely used in multivariate cryptography.

The SAT problem is defined as follows. First, one is given a logical expression in the five operators of predicate calculus ($\wedge, \vee, \neg, \Rightarrow, \iff$) but not the existential or universal quantifiers ($\exists, \forall$). Then, one is asked if the logical expression has a assignment for each of its variables, such that the logical expression evaluates to true. Unlike, the problem if the expression has the value false, for every one its possible assignments, is called UNSAT. (Bard, 2009, p. 200)

Davis e Putnam (1958) proposed a method for solve the SAT problem turning it to logical expressions in conjunctive normal form CNF. A formula is in CNF form if it is a conjunction of clauses, and each clause is a disjunction of literals. In follow, we define these concepts more formally.

Definition 4.0.1. *The language of Boolean formulae consists of Boolean variables, whose values are True or False; Boolean operators such as negation (not), conjunction ($\wedge$), disjunction ($\vee$), implication ($\Rightarrow$), equivalence ($\iff$); and parentheses.*

Definition 4.0.2. *A literal is either an atomic formula or the negation of an*

atomic formula.

Definition 4.0.3. *A clause is a disjunction of $n > 0$ literals*

The most study and promises SAT solvers are based on the work of Davis and Putnam. In this thesis, for the cryptanalysis against multivariate quadratic scheme we use a kind of solvers called Clause Driven Clause Learn SAT (CDCL-SAT). As we will see, these SAT solver are inspired on the Davis and Putnam method and uses as input a logical expression in CNF.

When the maximum number of literals presented in each clause of a CNF logical expression is k, then the respective SAT problem is known as k-CNF-SAT problem. Since clause sets with only one literal per clause are easy to satisfy, the computer scientist, and particularly in this thesis, we are interested in slightly larger classes. Exactly what is the clause size at which the problem turns from polynomial to hard? This transition occurs when each clause contains three literals, the so-called 3-CNF-SAT problem. Follow there is a formalization of the 3-CNF-SAT problem.

3-CNF-SAT Problem. Let $B = \{b_1, \cdots, b_n\}$ be a set of Boolean variables, let $L = \left\{b_1, \overline{b_1}, \cdots, b_n, \overline{b_n}\right\}$ be the corresponding set of literals, let $c_i \in (L \cup L^2 \cup L^3)$ be clauses of at most 3 literals, and let $C = \{c_1, \cdots, c_m\}$ be a set of these clauses. Then the corresponding 3-CNF-SAT problem is to determine if there is an assignment $A \in \{0,1\}^n$ for B such that all c_i are true and hence C is satisfied.

The worse-case estimates for solving SAT on a CNF problem are commonly given in terms of the numbers of clauses c, number of variables v, the total length of all clauses L, and the Strahler number of the tree-like resolution proof. In section we will review several methods for trying to formalize the hardness of solving SAT problems.

Chapter 5

Multivariate Cryptography

5.1 Basic Notation

In the context of multivariate cryptography, we mostly deal with systems of multivariate quadratic polynomials functions over finite fields. In this work, by $\mathbb{F}_q$, or GF(q) we denote a finite field consisting of q elements, and by $\mathbb{E}$ a extension field of $\mathbb{F}_q$. By $\mathbb{F}_q^n$ we shall denote the vector space of n-tuples over $\mathbb{F}_q$. Elements of $\mathbb{F}_q^n$ are denoted by bold characters.

A multivariate quadratic system of polynomials functions with n variables and m equations has the following form

$$p^{(k)}(x_1, x_2, \cdots, x_n) = \sum_{1 \leq i \leq j \leq n} \gamma_{i,j}^{(k)} x_i x_j + \sum_{i=1}^{n} \beta_i^{(k)} x_i + \alpha^{(k)} \tag{5.1}$$

where $1 \leq k \leq m$ and $\gamma_{i,j}^{(k)}, \beta^{(k)}, \alpha^{(k)} \in \mathbb{F}_q$. If $m = n$ the system is called determined, if $m > n$ the system is called overdetermined and if $m < n$ the system is called undetermined. We call $\gamma_{i,j}^{(k)}, \beta^{(k)}, \alpha^{(k)}$ coefficients and $x_i x_j, x_i$ monomials. The product of a coefficient with a monomial is called a term. We call a multivariate quadratic system P homogeneous if it not contains linear, nor constants terms.

A multivariate quadratic system can be expressed as a set

$$P := \{p^{(1)}, p^{(2)}, p^{(3)} \cdots p^{(m)}\}$$

of polynomial functions or in equivalent way as a quadratic map $P\colon \mathbb{F}_q^n \to \mathbb{F}_q^m$

$$P = \begin{pmatrix} p^{(1)}(x_1, x_2, \cdots, x_n) \\ p^{(2)}(x_1, x_2, \cdots, x_n) \\ \vdots \\ p^{(m)}(x_1, x_2, \cdots, x_n) \end{pmatrix}. \tag{5.2}$$

Matrix Representation. It is possible to write each polynomial function of (5.2) $p^{(k)}$ as a matrix-vector product using the upper triangular matrix

$$P^{(k)} = \begin{bmatrix} \gamma_{1,1}^{(k)} & \gamma_{1,2}^{(k)} & \gamma_{1,3}^{(k)} & \cdots & \gamma_{1,n}^{(k)} & \beta_1^{(k)} \\ 0 & \gamma_{2,2}^{(k)} & \gamma_{2,3}^{(k)} & \cdots & \gamma_{2,n}^{(k)} & \beta_2^{(k)} \\ 0 & 0 & \gamma_{3,3}^{(k)} & \cdots & \gamma_{3,n}^{(k)} & \beta_3^{(k)} \\ \vdots & & \ddots & & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \gamma_{n,n}^{(k)} & \beta_n^{(k)} \\ 0 & 0 & \cdots & 0 & 0 & \alpha^k \end{bmatrix} \tag{5.3}$$

and the vector $(x_1, x_2, x_3, \cdots, x_n, 1)$ in the following form

$$p^{(k)} = (x_1, x_2, x_3, \cdots, x_n, 1) P^{(k)} (x_1, x_2, x_3, \cdots, x_n, 1)^T \tag{5.4}$$

Although, this construction seems strange this will be useful at the chapter **??** when, we talked about Attacks on Multivariate Quadratic Schemes.

5.2 The Standard (Bipolar) Construction

The standard and most popular construction of multivariate cryptography consists of select a map $P\colon \mathbb{F}_q^n \to \mathbb{F}_q^m$ of polynomials functions of degree two as (5.2), such that it is easy to find pre-images. Besides, chooses two affine transformations $S\colon \mathbb{F}_q^m \to \mathbb{F}_q^m$ and $T\colon \mathbb{F}_q^n \to \mathbb{F}_q^n$. These objects are used to construct encryption and signature schemes and we will see that for encryption is necessary that $m \geq n$ and for signature is necessary that $m \leq n$. The public key is the composition $\bar{P} = S \circ P \circ T$ and the private key consists of S, P and T.

Encryption Schemes

Encryption. To encrypt a message $\boldsymbol{m}$, it is only necessary compute $\boldsymbol{c} = \bar{P}(\boldsymbol{m})$. The ciphertext of the message $\boldsymbol{m}$ is $\boldsymbol{c}$.

Decryption. To decrypt the ciphertext $\boldsymbol{c}$ is necessary compute $\boldsymbol{y} = S^{-1}(\boldsymbol{c})$, $\boldsymbol{y'} = \bar{P}^{-1}(\boldsymbol{y})$ and the original message is $\boldsymbol{m} = T^{-1}(\boldsymbol{y'})$. Note that P^{-1} do not denote the inverse functions, which might not even exist, but some pre-image. In other words, sometimes is possible to exist another pre-images $\boldsymbol{m_1}, \boldsymbol{m_2}, \cdots \boldsymbol{m_k}$ such that $\bar{P}(\boldsymbol{m_1}) = \bar{P}(\boldsymbol{m_2}) = \cdots = \bar{P}(\boldsymbol{m_k}) = \bar{P}(\boldsymbol{m})$. This problem has been discussed by Patarin (1996) and he suggest to use error correcting codes or hash function to solve it.

We say that for encryption is necessary that $m \geq n$ because in this case with high probability there is a unique pre-image of $\boldsymbol{y}$ under $\bar{P}$.

Signature Schemes

Signature Generation. To sign a document d, we calculate $\boldsymbol{h} = H(d)$ where H is a hash function. Then we compute $\boldsymbol{y} = S^{-1}(\boldsymbol{h})$, $\boldsymbol{y'} = \bar{P}^{-1}(\boldsymbol{y})$ and the signature of $\boldsymbol{h}$ is $\boldsymbol{z} = T^{-1}(\boldsymbol{y'})$. Again, note that P^{-1} do not denote the inverse functions, which might not even exist, but any pre-image.

We say that for signature $n \geq m$ because thus we can be sure that such a pre-image exists. Thus, every message has a signature.

Signature Verification. To verify the signature $\boldsymbol{z}$ of the document d, we compute $P(\boldsymbol{z})$ if the result equal to $\boldsymbol{h}$ the signature is accepted, otherwise it is rejected.

Schneier (1995) argued that the use of hash functions in cryptography schemes is that in practical implementations, public-key algorithms are often too inefficient to sign long documents. Thus, to save time, digital signature protocols are often implemented with hash functions. Also, hash functions are used to decrease any pattern or any known structure of a message to sign.

The security of bipolar constructions is based on the MQ-Problem and IP-Problem problems. The MQ-Problem is proved NP-Complete but not the IP-

problem, this fact difficult to construct provable security schemes based on multivariate quadratic systems.

5.2.1 Example

We shall give an example of a standard construction namely Random Singular Simultaneous Equation (R(S)SE) which was proposed by Kasahara e Sakai (2004). The reason to chose this is that its simplicity for construction.

Key Generation. The map P in R(S)SE has L layers of polynomials functions. Let i be a i^{th} layer. Each layer has r_i new variables and m_i equations. The r_i values are such that $\sum_{i=1}^{L} r_i = n$ and the m_i values are such that $\sum_{i=1}^{L} r_i = m$. In Figure 5.1, we depict the form of the map P for the R(S)SE scheme.

The public key $\bar{P}$ is constructed by using two affine transformations $S\colon \mathbb{F}_q^m \to \mathbb{F}_q^m$ and $T\colon \mathbb{F}_q^n \to \mathbb{F}_q^n$. These two transformations are operated with $\bar{P}$ in the following way $\bar{P} = S \circ P \circ T$. The private key consists of S, P and T.

Encryption. To encrypt a message $\boldsymbol{m}$ we only has to evaluate this message in the map $\bar{P}$, i.e. the ciphertext $\boldsymbol{c} = \bar{P}(\boldsymbol{m})$.

Decryption. To decrypt the ciphertext $\boldsymbol{c}$, we first apply the inverse of the affine transformation T in $\boldsymbol{c}$. i.e. $\boldsymbol{y} = T^{-1}(\boldsymbol{c})$. After that, we must, to find the pre-image $\boldsymbol{x}$ of the system $\boldsymbol{y} = P(\boldsymbol{x})$. To calculate the pre-image we need to construct L sub-systems. The left side of each sub-systems has the form as each layer of Figure 5.1. To calculate the right side of each sub-system we need to split $\boldsymbol{y}$ in L layers $\boldsymbol{y}_1, \boldsymbol{y}_1, \cdots \boldsymbol{y}_L$. Each layer $\boldsymbol{y}_1, \boldsymbol{y}_1, \cdots \boldsymbol{y}_L$ has $m_1, m_2, \cdots, m_L$ elements respectively. For the Layer 1, we must compute an vector of variables $\boldsymbol{x}_1$ for the given vector $\boldsymbol{y}_1$ such that $\boldsymbol{x}_1$ has r_1 entries. To compute $\boldsymbol{x}_1$, we need to use brute force, that is, to compute $\boldsymbol{x}_1$ it is necessary to make q^{r_1} attempts. After one, we compute a vector of variables $\boldsymbol{x}_2$ for the given vector $\boldsymbol{y}_2$ such that $\boldsymbol{x}_2$ has r_2 entries. Again, to compute $\boldsymbol{x}_2$ we must, to use brute force, that is, to compute $\boldsymbol{x}_2$ it is necessary to make q^{r_2} attempts. These steps are repeated until we calculate the last vector

$$P = \left(\begin{array}{ll} \text{Layer 1} \left\{ \begin{array}{l} p^{(1)}(x_1, x_2, \cdots, x_{r_1}) \\ \quad \vdots \\ p^{(m_1)}(x_1, x_2, \cdots, x_{r_1}) \end{array} \right. \\ \text{Layer 2} \left\{ \begin{array}{l} p^{(m_1+1)}(x_1, x_2, \cdots, x_{r_1}, x_{r_1+1}, x_{r_1+2}, \cdots, x_{r_2}) \\ \quad \vdots \\ p^{(m_1+m_2)}(x_1, x_2, \cdots, x_{r_1}, x_{r_1+1}, x_{r_1+2}, \cdots, x_{r_2}) \end{array} \right. \\ \qquad\qquad \vdots \\ \text{Layer L} \left\{ \begin{array}{l} p^{(m_{L-1}+1)}(x_1, \cdots, x_{r_1}, x_{r_1+1}, \cdots, x_{r_2}, \cdots, x_{n_{L-1}+1}, \cdots, x_{n_{L-1}+n_L}) \\ \quad \vdots \\ p^{(m_{L-1}+m_L)}(x_1, \cdots, x_{r_1}, x_{r_1+1}, \cdots, x_{r_2}, \cdots, x_{n_{L-1}+1}, \cdots, x_{n_{L-1}+n_L}) \end{array} \right. \end{array} \right)$$

Figure 5.1: map P of (R(S)SE)

of variables $\boldsymbol{x}_L$ with r_L variables. Note, that following the construction of R(S)SE, we are finding r_i new variables in each step. Putting all $\boldsymbol{x}_i$ together in x we retrieve the original message setting $\boldsymbol{m} = S^{-1}(\boldsymbol{x})$. Note, that in order to obtain a practical scheme the value of r must be small, otherwise a legitimate user has a workload to decrypt.

5.3 The MQ Problem

As we said, the security of multivariate quadratic schemes relies on the computational hardness of solving large multivariate systems. This problem, called here, the MQ-problem is enunciated as follow.

MQ-Problem. Given a system $P = \left(p^{(1)}), \cdots, p^{(m)}\right)$ of m non-linear polynomials functions in the variables $x_1, \cdots, x_n$. Is there some $\overline{x_1}, \cdots, \overline{x_n}$ such that $p^{(1)}(x_1, \cdots, x_n) = 0, \cdots, p^{(m)}(x_1, \cdots, x_n) = 0$?

Even for the case of non-linear polynomial functions of degree 2, this problem is proved NP-complete. To show that, it is necessary to reduce the 3-CNF-SAT Problem to the MQ-Problem. A proof of this reduction can be found in the thesis of Wolf (2005).

5.4 Attacks against Multivariate Schemes

There are several attacks which are possible to apply against multivariate schemes. Wolf (2005) classify these attacks in two types, depending which of the following problems them trying to solve.

Inversion Problem. Given $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{F}^n \times \mathbb{F}^m$ a pair message/ciphertext or signature/message and $\bar{P}$ a public key for a multivariate scheme. Recover $\boldsymbol{x}$ for given $\boldsymbol{y}$ and public key $\bar{P}\colon \mathbb{F}_q^n \to \mathbb{F}_q^m$.

Note that the Inversion Problem is the same of the MQ-Problem.

Key Recovery Problem. Recover the private key (S, P, T) for a given public key $\bar{P}$.

The Inversion Problem plays a important role for the goals of our thesis. Remember that one of our goals is select a set of multivariate schemes together with the known best attack of each one of them. After that, we will fix several security levels against the known best attack of each one. At each level we will compare them to determine who is more resistant to inversion attack using SAT solvers.

We will dealing with the known best attack of each multivariate scheme that are still unbroken in the chapter 6. In this section, we will describe some general inversion attacks which works against any multivariate schemes.

5.4.1 SAT Solving Attack

A approach for trying to solve the Inversion Problem is to use SAT solvers. The inversion problem is equivalent to the 3-CNF-SAT problem enunciated in the chapter 4. Therefore, it is possible to translate the a system of multivariate polynomials functions into a logical expression in CNF. This 3-CNF-SAT instance is submitted to a SAT solver. [BARD] shows techniques for to make this translating.

Below, we will describe how to convert instances of the Inversion Problem to the 3-CNF-SAT Problem, using mainly the techniques of [BARD] as reference.

5.4.2 Fault Attacks

[France paper]

5.4.3 Groebner Basis Attack

In this section we will review and to make a theoretical study of the number of equations that determined, undetermined and overdetermined random multivariate quadratic systems should have, to resists attacks from the HybridF5 Algorithm, which is the fastest algorithm to solve them. Basically, this algorithm consists in guess some variables of a multivariate nonlinear equation before to solve it with the F5 algorithm. Guessing variables reduces the complexity of the F5 algorithm, but also increases the number of its runs. The complexity of solving a system of m quadratic equations in n variables over a finite field with q elements by the HybridF5 algorithm is given by

$$\text{complexity}_{\text{HF5}}(q,m,n) = \left(\min_{k\geq 0} q^k \cdot O\left(m \cdot \binom{n-k+d_{\text{reg}}-1}{d_{\text{reg}}}\right)\right)^{\omega} \tag{5.5}$$

where, for random systems, the degree of regularity d_{reg} is given as the lowest integer D for which the coefficient of t^D in $\dfrac{(1-t^2)^m}{(1-t)^n}$ is less or equal to 0 and $2 < \omega \leq 3$ denotes the linear algebra constant of solving a linear system. Unfortunately, the details of the F5 algorithm are not publicly known, which makes it difficult to estimate the concrete running time the algorithm needs to solve a multivariate system. To derive secure parameters for multivariate schemes from formula (5.5), we follow the approach of the thesis of Petzoldt (2013). He set $\omega = 2$ and get by

$$\text{lower bound}_{\text{HF5}}(q,m,n) = \left(\min_{k\geq 0} q^k \cdot O\left(m \cdot \binom{n-k+d_{\text{reg}}-1}{d_{\text{reg}}}\right)\right)^{2} \tag{5.6}$$

a lower bound of the complexity of the HybridF5 algorithm. If this lower bound is larger than the security level, we can be sure that an attack against the multivariate quadratic system with the HybridF5 algorithm is infeasible.

The number of variables n and equations m of this review and study is focused on the selected schemes. The results of this study helps to calculate the parameters of the different selected multivariate quadratic schemes to resist attacks of the HybridF5 algorithm. Different sets of parameters, for each multivariate quadratic scheme, generate several security levels. We will use these sets of parameters to estimate the running time that the SAT solver takes, when it is used to attack multivariate quadratic schemes. All this analysis is in the chapter.

When the system is only slightly undetermined ($m < n < 2m$), then the best strategy to solve it is by fixing $n - m$ of the variables and trying to solve the generated determined system with the HybridF5 algorithm. One can expect that the determined system will have exactly one solution. So the complexity of solving an undetermined system with m equations in $n(m < n < 2m)$ variables is the same as that of a determined system with m equations. As was developed by Petzoldt (2013) is his Phd thesis this strategy is applied to attack Rainbow and UOV schemes. If the system is highly underdetermined system, namely $n \geq m \cdot (m + 1)$ Kipnis et al. (1999) found that it can be solved in polynomial time. Thomae e Wolf (2012) revisiting the Kipnis and Shamir attack found that, when $n = \omega \cdot m$ holds, then the undetermined system of m equations in n variables is hard to solve as a determined system in $m' = m - \lfloor \omega \rfloor + 1$ equations. The strategy of fixing $n - m$ of the variables together the improved of Thomae e Wolf (2012) are taken into account by Petzoldt (2013) when analyse UOV.

For the case of overdetermined system Kipnis et al. (1999) found that if the number of equations and variables is such that $m \geq n(n - 1)/2$ then the system can be solved in polynomial time. In this thesis we have selected the ABC scheme proposed (Tao et al., 2013) which has $m = 2n$ equations.

5.4.3.1 Reviewing the number of equations to solve undetermined systems

Because the strategy of fixing $n - m$ is better when the HybridF5 algorithm is used against undetermined systems, i.e translate the undetermined system to determined system, Petzoldt (2013) make a theoretical and experimental study of the use of HybridF5 algorithm to solve random determined multivariate quadratic systems. With this study they calculates secure parameters for UOV and Rainbow.

After the fixing strategy, he calculates the number of variables for guessing to build overdetermined systems such that HybridF5 algorithm has better performance. He found:

- for determined systems over $\mathbb{F}_{16}$ it is the best strategy to guess 5 ($m \leq 29$), 6 ($30 \leq m \leq 40$) or 7 ($m \geq 41$) variables,
- for determined systems over $\mathbb{F}_{31}$ it is the best strategy to guess 2 ($m \leq 27$), 3 ($28 \leq m \leq 35$) or 4 ($m \geq 36$) variables and
- for determined systems over $\mathbb{F}_{256}$ it is the best strategy to guess 1 ($m \leq 32$) or 2 ($m \geq 33$) variables.

With these guessing variables and evaluating equation (5.6) for $m \in \{20, \cdots, 60\}$, he found that the bit complexity of solving a determined random system of m multivariate quadratic equations using the HybridF5 algorithm is roughly

$$2.30 \cdot m + 14.4 \tag{5.7}$$

for systems over $\mathbb{F}_{16}$,

$$2.50 \cdot m + 13.0 \tag{5.8}$$

for systems over $\mathbb{F}_{31}$ and

$$2.65 \cdot m + 12.1 \tag{5.9}$$

for systems over $\mathbb{F}_{256}$.

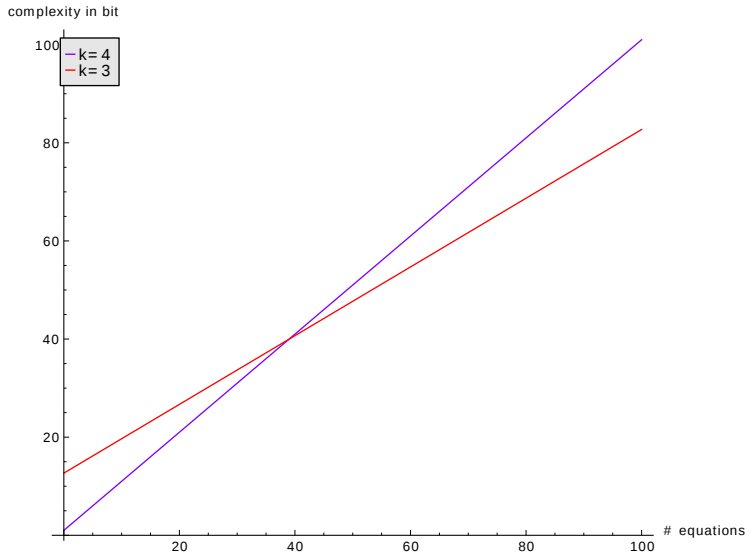

Figure 5.2: Complexity of solving overdetermined random systems over $\mathbb{F}_2$ with HybridF5

Besides, the theoretical equations (5.7), (5.8) and (5.9) he carried out a large number of computer experiments. For these experiments he used MAGMA v. 2.13-10 which contains an efficient implementation of Faugeres F4 algorithm. Although MAGMA v. 2.13-10 was already released in 2008, it is still one of the fastest publicly available solvers for multivariate nonlinear systems and in fact often faster than newer versions of MAGMA. The results of the experiments are very similar to the result of together with the theoretical study.

5.4.3.2 Calculating the number of equations to solve overdetermined systems

Next, we are going to compute the number of equations to protect overdetermined random multivariate quadratic systems for the case $m = 2 \cdot n$, using the complexity of the HybridF5 algorithm.

As Petzoldt (2013), we use the equation (5.6) to compute the complexity of solving overdetermined random systems over $\mathbb{F}_2$, $\mathbb{F}_4$, $\mathbb{F}_{16}$ and $\mathbb{F}_{256}$ with the HybridF5 algorithm when guessing different number of variables. Figures 5.2, 5.3, 5.4 and 5.5 show the results.

A natural question in this context is to choose the best tradeoff, i.e. for

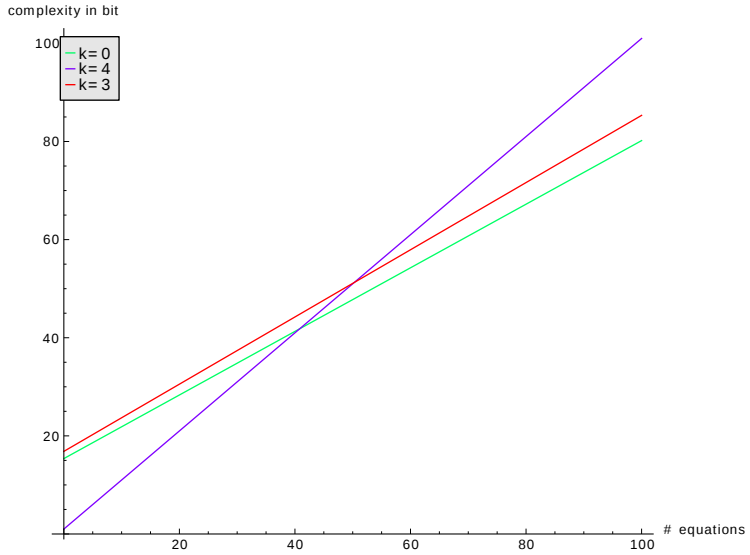

Figure 5.3: Complexity of solving overdetermined random systems over $\mathbb{F}_4$ with HybridF5

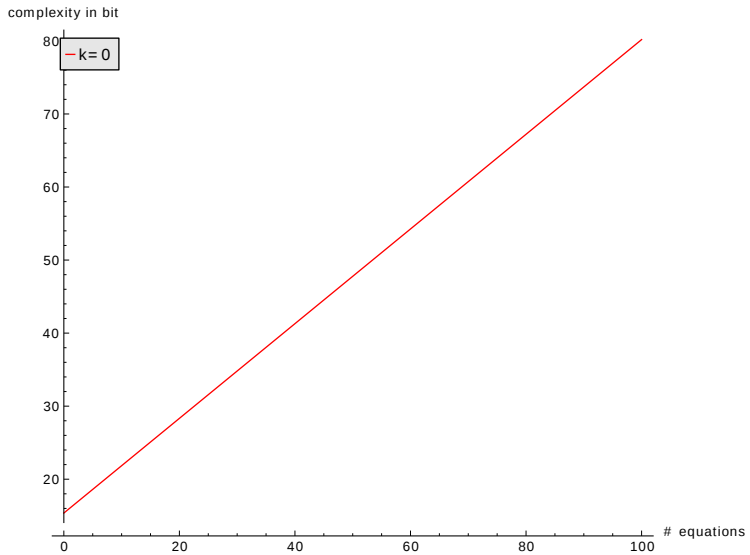

Figure 5.4: Complexity of solving overdetermined random systems over $\mathbb{F}_{16}$ with HybridF5

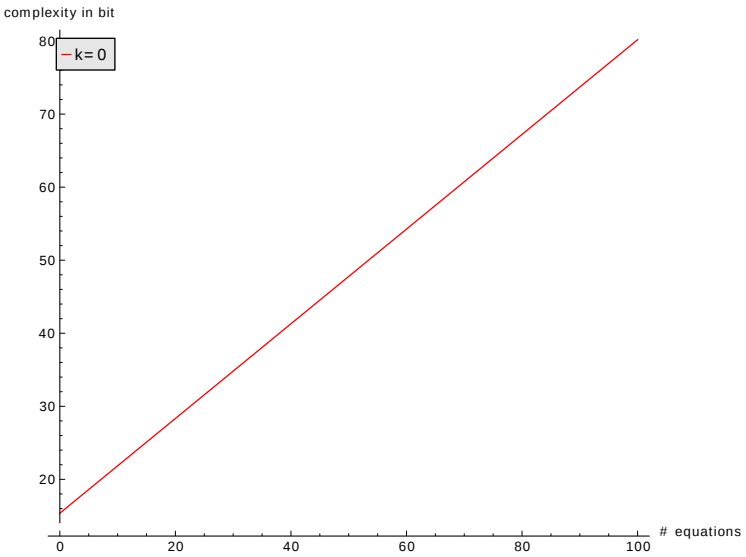

Figure 5.5: Complexity of solving overdetermined random systems over $\mathbb{F}_{256}$ with HybridF5

which number $k \geq 0$ the complexity of the algorithm as shown by equation (5.6) gets minimal. Guessing variables to build **overdetermined systems** reduces the complexity of each run of the F5 algorithm, but also increases the number of this runs. We found

- for overdetermined systems over $\mathbb{F}_2$ it is the best strategy to guess 7 ($m \leq 41$), 6 ($m \geq 42$),
- for overdetermined systems over $\mathbb{F}_4$ it is the best strategy to guess 4 ($m \leq 40$), 0 ($m \geq 41$),
- for overdetermined systems over $\mathbb{F}_{16}$ and $\mathbb{F}_{256}$ it is the best strategy to guess 0 ($m > 0$).

Evaluating (5.6) for $m \in \{8, \cdots, 108\}$, we find that the bit complexity of solving a overdetermined random system of m multivariate quadratic equations and n variables using the HybridF5 algorithm is roughly

$$m + 1 \tag{5.10}$$

for systems over $\mathbb{F}_2$ and $\mathbb{F}_4$,

$$0.65 \cdot m + 15.36 \tag{5.11}$$

for systems over $\mathbb{F}_{16}$,

$$0.65 \cdot m + 15.36 \tag{5.12}$$

for systems over $\mathbb{F}_{256}$.

Chapter 6

Some Multivariate Schemes

In this Chapter we will review some multivariate schemes and its known best attacks. We will analyse the security of these multivariate schemes at chapter **??**T solvers.

6.1 UOV Scheme

The Unbalanced Oil Vinegar (UOV) scheme was proposed by Kipnis et al. (1999) and it is a generalisation of the original Oil and Vinegar scheme proposed by Patarin (1997). This scheme has not the usual standard construction, later we will explain this fact.

The polynomial map P of UOV is defined has the following form:

$$P = \begin{pmatrix} p^{(1)}(\overset{v}{x_1}, \cdots, \overset{v}{x_{n-m}}, \overset{o}{x_{n-m+1}} \cdots, \overset{o}{x_n}) \\ \vdots \\ p^{(m)}(\overset{v}{x_1}, \cdots, \overset{v}{x_{n-m}}, \overset{o}{x_{n-m+1}} \cdots, \overset{o}{x_n}) \end{pmatrix}. \tag{6.1}$$

In this context, n is the number of variables and m the number of equations of the polynomial map P. The variables x_i for $1 \leq i \leq n-m$ are called of vinegar and are represented with the symbol v over these variables. The variables x_i where $n-m+1 \leq i \leq n$ are called of oil and are represented with the symbol o over

these variables. The polynomials functions $p^{(i)}$ are defined as follow

$$p^{(i)} = \sum_{j=1}^{n-m}\sum_{k=1}^{n-m} \gamma_{j,k}^{i} \overset{v}{x}_j \overset{v}{x}_k + \sum_{j=1}^{n-m}\sum_{k=n-m+1}^{n} \gamma_{j,k}^{i} \overset{v}{x}_j \overset{o}{x}_k + \sum_{k=1}^{n-m} \beta_k^i \overset{v}{x}_k + \sum_{k=n-m+1}^{n} \beta_k^i \overset{o}{x}_k + \alpha^i.$$

Note that the vinegar variables are combined quadratically, whereas the oil variables are only combined with vinegar variables. Hence, replacing the vinegar variables with values in $\mathbb{F}_q$ is possible to obtain a linear system of m equations in $n-m$ oil variables. That linear system can be resolved using the Gaussian elimination method easily. Besides, note that because we have the term $n-m$ this scheme can be used only for signature.

Key Generation. The UOV scheme ignore the usual linear transformation $S \in \mathbb{F}_q^m \to \mathbb{F}_q^m$ of the standard construction. This omission is because in UOV scheme all equations have the same shape, then is not necessary to hide some special structure.

Therefore, the key generation processes works as follow. Select a invertible map $T \in \mathbb{F}_q^n \to \mathbb{F}_q^n$ and compute $\bar{P} = P \circ T$. Then the public key is $\bar{P}$ and the private key are P and T.

Signature Generation To sign a document d we use a hash function H and compute $\boldsymbol{h} = H(d)$. After that, we compute $\boldsymbol{y} = P^{-1}(\boldsymbol{h})$. Hence, the signature of the document d is $\boldsymbol{x} = T^{-1}(\boldsymbol{y})$. Note that P^{-1} not means it be an invert map of P but, one that find pre-images of P.

Verification To verify that $\boldsymbol{x}$ is the signature of the document d, we compute $\boldsymbol{h} = H(d)$ and after one we compute $\boldsymbol{h}' = \bar{P}(\boldsymbol{x})$. If $h' = h$ then the signature is verified, otherwise rejected.

6.1.1 Known Best Attack against the UOV scheme

According [thesis Albrecht] the relevant attacks are the direct attack, the UOV attack of Kipnis and Shamir and the UOV Reconciliation attack. There is another attack proposed by called The Rainbow Band Separation attack is actually an extension of the UOV Reconciliation attack which uses the special structure of

Rainbow; it is not suitable for UOV.

All three attacks (UOV, UOV Reconciliation and Rainbow Band Separation) find an equivalent private key, i.e UOV/Rainbow private maps which compose to the same public key. However, in the case of UOV, this private key consists only of a UOV central map and one linear transformation, whereas in the case of Rainbow you have, besides the central map, two linear transformations. Therefore, the UOV and the UOV Reconciliation attack output two, whereas the Rainbow central map outputs three maps. Therefore the first two attacks can only be used against UOV, while the third one is used against Rainbow.

Regarding the efficiency of the attacks: The UOV Reconciliation attack has hardly effect on the parameter choice of UOV. The reason for this is that it requires the solution of a multivariate quadratic system in o equations (just as a direct attack). It is therefore not more efficient than a direct attack against UOV.

The UOV attack has a complexity of $O(q^{v-o})$, where q is the cardinality of the underlying field. To defend UOV against this attack, the difference between v and o should not be too small. If $v = o$, the attack runs in polynomial time and the scheme offers no security at all. To be on the conservative side, on therefore chooses the parameters of UOV to be $v >= 2o$.

To find good parameters for UOV, one first selects the number of equations in such a way that a direct attack against the scheme is infeasible (for $q = 256$, this leads to $o = 28$). After this, to be secure against the UOV attack, one chooses $v = 2o$. The number of variables in the scheme is therefore given by $n = o + v = 3o$.

6.2 Rainbow Scheme

6.3 STS Scheme

6.4 ABC Scheme

This scheme has a standard construction and was proposed by Tao et al. (2013). In the original decryption process failures occur with non-negligible prob-

ability. These failures has been addressed by Ding et al. (2014), Tao et al. (2015) and by Petzoldt et al. (2016). Here we will explain the ABC scheme with the modification of Petzoldt et al. (2016) to solve the cited problem.

The main idea of this scheme is create matrices with high rank and use some Simple Matrix Multiplication to construct the map of polynomial functions P. Let $\mathbb{F}_q$ be a finite field. Let s be a integer. Let $n = s^2$ a number of variables and $m = 2n$ a number of equations of the map. Let A be a matrix with dimensions $s \times s$ such that its entries are indeterminates belonging to $\mathbb{F}_q$. Let B and C be matrices with dimensions $s \times s$ such that its entries are multivariate polynomials of degree one . Define $E_1 = AB$ and $E_2 = AC$. Let $p^{((i-1)s+j)}$ and $p^{(s^2+(i-1)s+j)}$ be respectively the (i, j) element of E_1 and E_2 respectively, where $(i, j = 1, \cdots, s)$. Then each polynomial function of the polynomial map P is define by the entries of E_1 and E_2 in the following way.

$$P = \begin{pmatrix} p^{(1)}(x_1, x_2, \cdots, x_n) \\ \vdots \\ p^{(s^2)}(x_1, x_2, \cdots, x_n) \\ p^{(s^2+1)}(x_1, x_2, \cdots, x_n) \\ \vdots \\ p^{(m)}(x_1, x_2, \cdots, x_n) \end{pmatrix}. \tag{6.2}$$

Key Generation. Choose two invertible $s \times s$ matrices T_1 and T_2 and define $T = T_1 \otimes T_2$. Then, the public key, that is, the map $\bar{P}$, is constructed by using two affine transformations $S \colon \mathbb{F}_q^m \to \mathbb{F}_q^m$ and $T \colon \mathbb{F}_q^n \to \mathbb{F}_q^n$ in the following way $\bar{P} = S \circ P \circ T$ and the private key consists of T_1, T_2, B, C, and S.

Encryption. To encrypt a message $\boldsymbol{m} \in \mathbb{F}^*$ there are 3 steps:

- Split the message in blocks of size n, namely $m_1, m_2, m_3, \cdots$.
- After one, verify if the associate matrix A of m_1 is invertible. If that matrix is invertible then we evaluate this message in the map $\bar{P}$, i.e. compute the ciphertext $\boldsymbol{c}_1 = \bar{P}(\boldsymbol{m}_1)$. Otherwise, Add some constant character to the

block m_1 at random position and verify if the new associated matrix has inverse.

- We repeat the last step until the associated matrix has inverse and for all blocks of the message $\boldsymbol{m}$.

Decryption. The decryption process of the ciphertext $\boldsymbol{c} \in \mathbb{F}^n$ consists of four steps:

1 Compute $\boldsymbol{x} = S^{-1}(\boldsymbol{c})$. The elements of the vector $\boldsymbol{x} \in \mathbb{F}^m$ are written into matrices $\bar{E}_1$ and $\bar{E}_2$ as follows

$$E_1 = \begin{pmatrix} x_1 & \cdots & x_s \\ \vdots & & \vdots \\ x_{(s-1)s+1} & \cdots & x_n \end{pmatrix}, E_2 = \begin{pmatrix} x_{n+1} & \cdots & x_{n+s} \\ \vdots & & \vdots \\ x_{n+(s-1)s+1} & \cdots & x_m \end{pmatrix}. \quad (6.3)$$

2 After one, it is necessary to compute the pre-image $\boldsymbol{y}$ of the system $\boldsymbol{x} = P(\boldsymbol{y})$. To do this, there are four cases, namely:

- If $\bar{E}_1$ is invertible, we consider the equation $B\bar{E}_1^{-1}\bar{E}_2 - C = 0$. Thus, we gets n linear equations in the n variables $y_1, \cdots, y_n$.
- If $\bar{E}_1$ is not invertible, but $\bar{E}_2$, we consider $C\bar{E}_2^{-1}\bar{E}_1 - B = 0$. Thus, we gets n linear equations in the n variables $y_1, \cdots, y_n$.
- If neither $\bar{E}_1$ nor $\bar{E}_2$ are not invertible but $\bar{A} = A(\boldsymbol{y})$ is invertible, we consider the relations $\bar{A}^{-1}\bar{E}_1 - B = 0$ and $\bar{A}^{-1}\bar{E}_2 - C = 0$. We interpret the elements of A^{-1} as new variables $w_1, \cdots, w_n$ and therefore we gets m linear equations in m variables $w_1, \cdots, w_n, y_1, \cdots y_n$.

3 Finally, we compute the plaintext by $\boldsymbol{m} = T^{-1}(y1, \cdots, yn)$.

4 After having found the encrypted plaintext m, we remove all the appearances of the constant character to get the original message.

6.4.1 Known Best Attack against the ABC scheme

There are several attacks that it is possible to apply against several multivariate schemes. Among them, there is the Minrank attack proposed by against the multivariate proposed by and called HFE.

Moody et al. (2014) developed a attack, which combined with the Minrank attack is the known best attack against the ABC scheme.

Below, we will explain first the Minrank attack against the ABC scheme and after the strategy proposed by Moody et al. (2014).

Minrank Attack Against the ABC Scheme

An Asymptotically Optimal Structural Attack on the ABC Scheme

Part

Contributions

Chapter 7

Hash-Coding Based Cryptography

Chapter 8

Direct Attack on MQ using CDCL SAT solvers

In this chapter, we are going to review the best known structural attacks against the ABC, UOV and Rainbow multivariate schemes. These known best structural attacks to each specific scheme together the HybridF5 attack given secure parameters which we are going to use to construct several instances of these schemes and attack them using CDCL SAT solvers.

Specifically, we used a massively parallel portfolio SAT solver called HordeSAT which used MPI. As we said, the SAT solvers in the portfolio can be instances of a single solver with different configuration settings. Additionally the solvers can exchange information usually in the form of clauses.

8.1 UOV

Taking into account the attack of Thomae e Wolf (2012) and using the equations (5.7),(5.8) and (5.9) Petzoldt (2013) has calculated the secure parameters for the Rainbow and UOV schemes. Because of the attack of Thomae e Wolf (2012) reduce the number of equations in the public systems by 2 before applying an algorithm like HybridF5, Petzoldt (2013) increased the number of equations by 2. Besides, to defend the scheme against the UOV attack of Kipnis et al. (1999), they recommend choose $v = 2 \cdot o$. Thus, in the Table 8.1 are shown the proposed parameter sets for the UOV scheme for different levels of security and different

underlying fields.

Solving UOV intances over $\mathbb{F}_{16}$ using CDCL SAT solvers

8.2 ABC

Following the model showed in Petzoldt (2013) we have calculated the secure parameters for the ABC scheme proposed by Tao et al. (2015). Taking into account that the attack of Thomae e Wolf (2012) reduce the number of equations in the public systems by 2 before applying an algorithm like HybridF5, we increased the number of equations by 2. Also, the known best structural attack against this scheme was proposed by Moody et al. (2014) and its complexity is

$$s^6 \cdot q^{s+4} \tag{8.1}$$

for even characteristic and

$$s^6 \cdot q^{s+2} \tag{8.2}$$

for odd characteristic. Hereafter, we are going to name StructuralABC the attack proposed by Moody et al. (2014).

We use the equations (5.6) and (8.1) to make a comparison between the complexity of the HybridF5 and StructuralABC attacks against instances of the ABC cryptosystem over $\mathbb{F}_2$, $\mathbb{F}_4$, $\mathbb{F}_{16}$ and $\mathbb{F}_{256}$. Figures 8.1, 8.2, 8.3 and 8.4 show the results when the equations (5.6) and (8.1) are evaluated for $m \in \{8, \cdots, 108\}$.[1] .

Clearly figure 8.4 shows that the best attack is the HybridF5 for those values of m. However, in figures 8.1, 8.2 and 8.3 are showed that there is not a absolute best attack for those chosen values of m. Thus, to be more precise and determine which is the intersection point that define at which interval the best attack is HybridF5 or StructuralABC, we have to re-write the equations (5.10), (5.11) and (5.12) for the critic intervals. Evaluating the equation (5.6) for the critic values, we find that the bit complexity of the HybridF5 attack against instances of the

[1] for the case of the HybridF5 was chose the minimum k - note by Juan

ABC scheme with m multivariate quadratic equations and n variables is roughly

$$0.72 \cdot m + 3.45 \tag{8.3}$$

for system over $\mathbb{F}_2$,

$$1.28 \cdot m + 2.45 \tag{8.4}$$

for system over $\mathbb{F}_4$,

$$0.77 \cdot m + 11.62 \tag{8.5}$$

for system over $\mathbb{F}_{16}$,

On other hand, evaluating (8.1) for $m \in \{8, \cdots, 108\}$, we find that the bit complexity of the StructuralABC attack against instances of the ABC scheme with m multivariate quadratic equations and n variables is roughly

$$0.15 \cdot m + 14.56 \tag{8.6}$$

for system over $\mathbb{F}_2$,

$$0.19 \cdot m + 20.9 \tag{8.7}$$

for system over $\mathbb{F}_4$,

$$0.29 \cdot m + 33.42 \tag{8.8}$$

for system over $\mathbb{F}_{16}$,

$$0.5 \cdot m + 58.47 \tag{8.9}$$

for system over $\mathbb{F}_{256}$.

Thus, according this comparison, we find

- for instances of the ABC scheme over $\mathbb{F}_2$ the known best attack is the HybridF5 attack, if $m \leq 19$ and the StructuralABC if $m \geq 20$,
- for instances of the ABC scheme over $\mathbb{F}_4$ the known best attack is the HybridF5 attack, if $m \leq 17$ and the StructuralABC if $m \geq 18$,

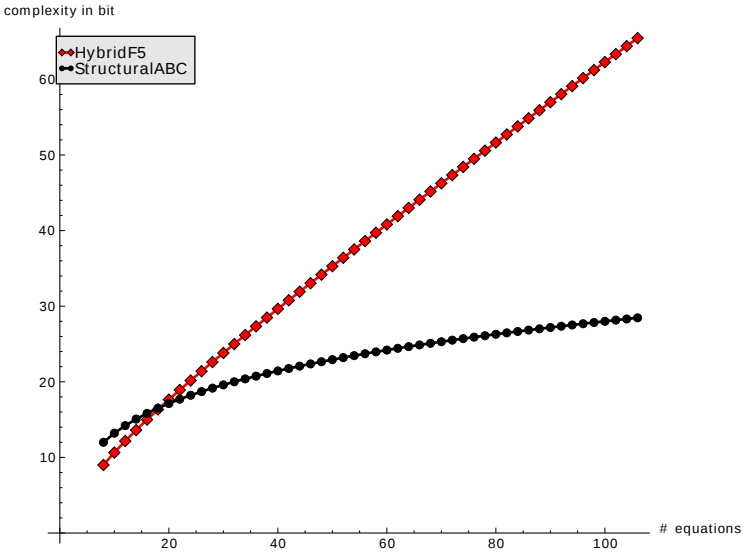

Figure 8.1: Complexity of the HybridF5 and StructuralABC attacks against instances of the ABC cryptosystem over $\mathbb{F}_2$.

- for instances of the ABC scheme over $\mathbb{F}_1 6$ the known best attack is the HybridF5 attack, if $m \leq 45$ and the StructuralABC if $m \geq 46$,
- for instances of the ABC scheme over $\mathbb{F}_{256}$ the known best attack is the HybridF5 attack for all considered cases of m.

In the Table **??** are shown the proposed parameter sets for the ABC scheme for different levels of security and different underlying fields.

8.3 Secure Parameters against SAT solving attack

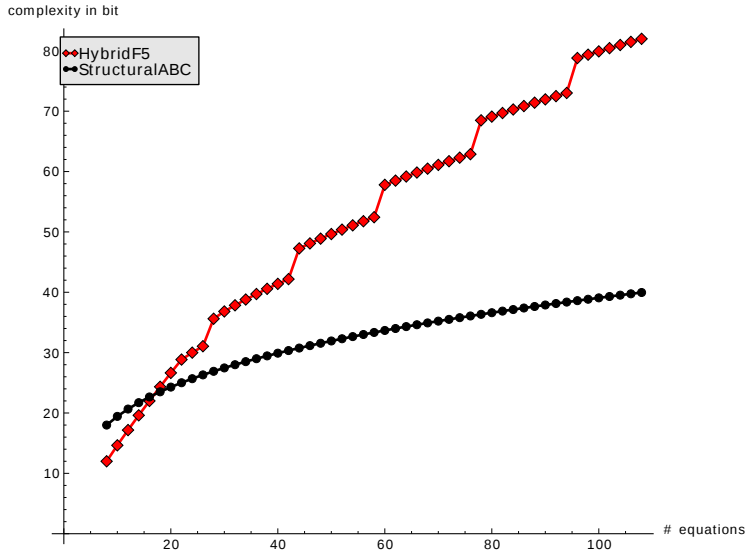

Figure 8.2: Complexity of the HybridF5 and StructuralABC attacks against instances of the ABC cryptosystem over $\mathbb{F}_4$.

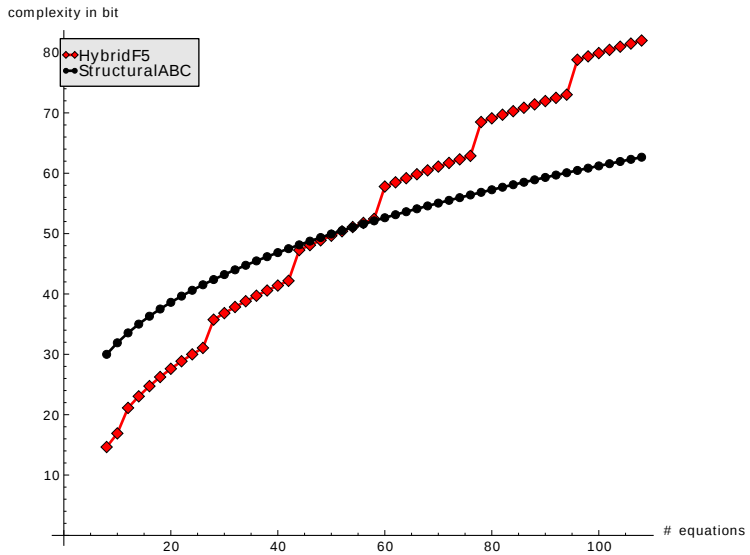

Figure 8.3: Complexity of the HybridF5 and StructuralABC attacks against instances of the ABC cryptosystem over $\mathbb{F}_{16}$.

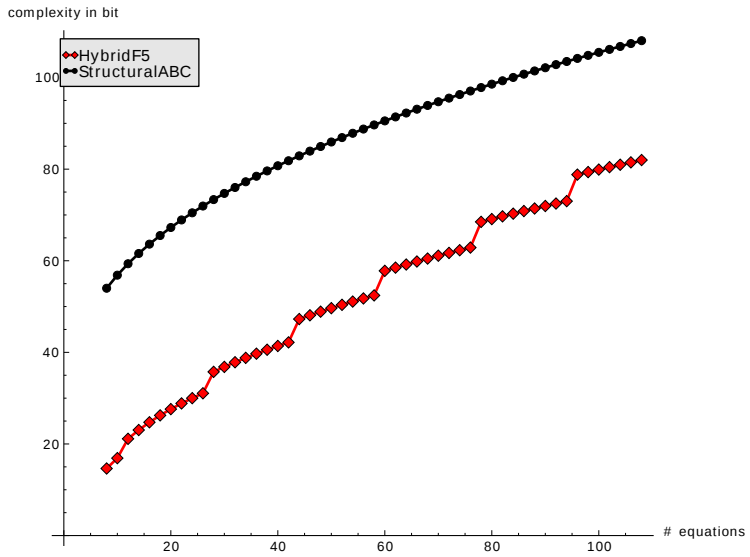

Figure 8.4: Complexity of the HybridF5 and StructuralABC attacks against instances of the ABC cryptosystem over $\mathbb{F}_{256}$.

Table 8.1: UOV parameters and security level for $\mathbb{F}_{16}$, $\mathbb{F}_{31}$ and $\mathbb{F}_{256}$

	$\mathbb{F}_{16}$		$\mathbb{F}_{31}$		$\mathbb{F}_{256}$	
security level (bit)	o	v	o	v	o	v
16	3	6	3	6	3	6
17	3	6	4	8	4	8
18	4	8	4	8	4	8
19	4	8	4	8	5	10
20	4	8	5	10	5	10
21	5	10	5	10	5	10
22	5	10	6	12	6	12
23	6	12	6	12	6	12
24	6	12	6	12	6	12
25	7	14	7	14	7	14
26	7	14	7	14	7	14
27	7	14	8	16	8	16
28	8	16	8	16	8	16
29	8	16	8	16	8	16
30	9	18	9	18	9	18
31	9	18	9	18	9	18
32	10	20	10	20	10	20
33	10	20	10	20	10	20
34	11	22	10	20	10	20
35	11	22	11	22	11	22
36	11	22	11	22	11	22
37	12	24	12	24	11	22
38	12	24	12	24	12	24
39	13	26	12	24	12	24
40	13	26	13	26	13	26
41	14	28	13	26	13	26
42	14	28	14	28	13	26
43	14	28	14	28	14	28
44	15	30	14	28	14	28

Table 8.2: HordeSAT configurations

Core Solvers	Parallel Solved	Avg.	Tot.	Med
solved-lingeling_uov_1x6x4	150	1	1.00	1.00
solved-lingeling_uov_2x6x4	150	3	1.68	1.12
solved-lingeling_uov_4x6x4	150	4	2.15	1.49
solved-lingeling_uov_8x6x4	150	14	6.39	2.83
solved-lingeling_uov_16x6x4	150	13	11.70	2.45
solved-lingeling_uov_32x6x4	150	12	17.46	2.86
solved-lingeling_uov_64x6x4	150	15	20.64	2.36
solved-lingeling_uov_128x6x4	150	19	25.22	2.10
solved-lingeling_uov_256x6x4	150	10	14.41	1.33
solved-lingeling_uov_512x6x4	150	18	21.37	1.36

Table 8.3: My caption

	$\mathbb{F}_2$	$\mathbb{F}_4$	$\mathbb{F}_{16}$	$\mathbb{F}_{256}$
security level(bit)	s	s	s	s
16	$\sqrt{10}$	$\sqrt{7}$	$\sqrt{4}$	$\sqrt{3/2}$
17	$\sqrt{11}$	$\sqrt{7}$	$\sqrt{9/2}$	$\sqrt{5/2}$
18	$\sqrt{13}$	$\sqrt{7}$	$\sqrt{5}$	$\sqrt{3}$
19	$\sqrt{16}$	$\sqrt{8}$	$\sqrt{6}$	$\sqrt{4}$
20	$\sqrt{19}$	$\sqrt{8}$	$\sqrt{13/2}$	$\sqrt{9/2}$
21	$\sqrt{23}$	$\sqrt{8}$	$\sqrt{7}$	$\sqrt{11/2}$
22	$\sqrt{26}$	$\sqrt{9}$	$\sqrt{15/2}$	$\sqrt{6}$
23	$\sqrt{29}$	$\sqrt{9}$	$\sqrt{17/2}$	$\sqrt{7}$
24	$\sqrt{33}$	$\sqrt{10}$	$\sqrt{9}$	$\sqrt{15/2}$
25	$\sqrt{36}$	$\sqrt{12}$	$\sqrt{19/2}$	$\sqrt{17/2}$
26	$\sqrt{39}$	$\sqrt{15}$	$\sqrt{21/2}$	$\sqrt{9}$
27	$\sqrt{43}$	$\sqrt{17}$	$\sqrt{11}$	$\sqrt{10}$
28	$\sqrt{46}$	$\sqrt{20}$	$\sqrt{23/2}$	$\sqrt{21/2}$
29	$\sqrt{49}$	$\sqrt{23}$	$\sqrt{25/2}$	$\sqrt{23/2}$
30	$\sqrt{53}$	$\sqrt{25}$	$\sqrt{13}$	$\sqrt{25/2}$
31	$\sqrt{56}$	$\sqrt{28}$	$\sqrt{27/2}$	$\sqrt{13}$
32	$\sqrt{59}$	$\sqrt{30}$	$\sqrt{14}$	$\sqrt{14}$
33	$\sqrt{63}$	$\sqrt{33}$	$\sqrt{15}$	$\sqrt{29/2}$
34	$\sqrt{66}$	$\sqrt{36}$	$\sqrt{31/2}$	$\sqrt{31/2}$
35	$\sqrt{69}$	$\sqrt{38}$	$\sqrt{16}$	$\sqrt{16}$
36	$\sqrt{73}$	$\sqrt{41}$	$\sqrt{17}$	$\sqrt{17}$
37	$\sqrt{76}$	$\sqrt{44}$	$\sqrt{35/2}$	$\sqrt{35/2}$
38	$\sqrt{79}$	$\sqrt{46}$	$\sqrt{18}$	$\sqrt{37/2}$
39	$\sqrt{83}$	$\sqrt{49}$	$\sqrt{19}$	$\sqrt{19}$
40	$\sqrt{86}$	$\sqrt{52}$	$\sqrt{39/2}$	$\sqrt{20}$
41	$\sqrt{89}$	$\sqrt{54}$	$\sqrt{20}$	$\sqrt{41/2}$
42	$\sqrt{93}$	$\sqrt{57}$	$\sqrt{41/2}$	$\sqrt{43/2}$
43	$\sqrt{96}$	$\sqrt{59}$	$\sqrt{43/2}$	$\sqrt{45/2}$
44	$\sqrt{99}$	$\sqrt{62}$	$\sqrt{22}$	$\sqrt{23}$

Table 8.4: $\mathbb{F}_{16}$

	$\mathbb{F}_2$	$\mathbb{F}_4$	$\mathbb{F}_{16}$	$\mathbb{F}_{256}$
security level(bit)	s	s	s	s
18	$\sqrt{25/2}$	$\sqrt{-13/2}$	$\sqrt{3}$	$\sqrt{3}$
19	$\sqrt{16}$	$\sqrt{-4}$	$\sqrt{4}$	$\sqrt{4}$
22	$\sqrt{26}$	$\sqrt{4}$	$\sqrt{6}$	$\sqrt{6}$
23	$\sqrt{29}$	$\sqrt{13/2}$	$\sqrt{7}$	$\sqrt{7}$
26	$\sqrt{39}$	$\sqrt{29/2}$	$\sqrt{9}$	$\sqrt{9}$
27	$\sqrt{85/2}$	$\sqrt{17}$	$\sqrt{10}$	$\sqrt{10}$
31	$\sqrt{56}$	$\sqrt{55/2}$	$\sqrt{13}$	$\sqrt{13}$
32	$\sqrt{59}$	$\sqrt{30}$	$\sqrt{14}$	$\sqrt{14}$
35	$\sqrt{69}$	$\sqrt{38}$	$\sqrt{16}$	$\sqrt{16}$
36	$\sqrt{145/2}$	$\sqrt{81/2}$	$\sqrt{17}$	$\sqrt{17}$
39	$\sqrt{165/2}$	$\sqrt{97/2}$	$\sqrt{19}$	$\sqrt{19}$
40	$\sqrt{86}$	$\sqrt{103/2}$	$\sqrt{20}$	$\sqrt{20}$
44	$\sqrt{99}$	$\sqrt{62}$	$\sqrt{23}$	$\sqrt{23}$

Table 8.5: Running Times vs Security Levels for UOV over $\mathbb{F}_{16}$

security level	1x6x4	2x6x4	4x6x4	8x6x4	16x6x4	32x6x4	64x6x4	128x6x4	256x6x4	512x6x4
22	1.9492	1.669	1.2832	1.1602	1.2624	1.4404	1.761	2.1884	3.108	16.3644
23	34.1918	18.7232	10.0598	8.357	6.2682	5.1414	5.5122	5.3082	12.913	7.5914
24	1250.32	741.795	584.909	191.694	102.364	67.0636	55.034	43.4934	73.2016	36.2316

Bibliography

Gregory Bard. **Algebraic cryptanalysis**. Springer Science & Business Media, 2009.

Martin Davis e Hilary Putnam. Reductions of hilbert's tenth problem. **The Journal of Symbolic Logic**, 23(02):183–187, 1958.

Jintai Ding, Albrecht Petzoldt, e Lih-chung Wang. **Post-Quantum Cryptography: 6th International Workshop, PQCrypto 2014, Waterloo, ON, Canada, October 1-3, 2014. Proceedings**. Capítulo The Cubic Simple Matrix Encryption Scheme, páginas 76–87, Springer International Publishing, Cham, 2014. ISBN 978-3-319-11659-4. URL http://dx.doi.org/10.1007/978-3-319-11659-4z_5.

Masao Kasahara e Ryuichi Sakai. A construction of public key cryptosystem for realizing ciphertext of size 100 bit and digital signature scheme. **IEICE Transactions**, 87-A(1):102–109, 2004.

Aviad Kipnis, Jacques Patarin, e Louis Goubin. **Advances in Cryptology — EUROCRYPT '99: International Conference on the Theory and Application of Cryptographic Techniques Prague, Czech Republic, May 2–6, 1999 Proceedings**. Capítulo Unbalanced Oil and Vinegar Signature Schemes, páginas 206–222, Springer Berlin Heidelberg, Berlin, Heidelberg, 1999. ISBN 978-3-540-48910-8. URL http://dx.doi.org/10.1007/3-540-48910-X_15.

Dustin Moody, Ray Perlner, e Daniel Smith-Tone. **Post-Quantum Cryptog-**

raphy: 6th International Workshop, PQCrypto 2014, Waterloo, ON, Canada, October 1-3, 2014. Proceedings. Capítulo An Asymptotically Optimal Structural Attack on the ABC Multivariate Encryption Scheme, páginas 180–196, Springer International Publishing, Cham, 2014. ISBN 978-3-319-11659-4. URL http://dx.doi.org/10.1007/978-3-319-11659-4_11.

Jacques Patarin. **Advances in Cryptology — EUROCRYPT '96: International Conference on the Theory and Application of Cryptographic Techniques Saragossa, Spain, May 12–16, 1996 Proceedings**. Capítulo Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms, páginas 33–48, Springer Berlin Heidelberg, Berlin, Heidelberg, 1996. ISBN 978-3-540-68339-1. URL http://dx.doi.org/10.1007/3-540-68339-9_4.

Jacques Patarin. The oil and vinegar signature scheme. In: **Dagstuhl Workshop on Cryptography**, 1997.

Albrecht Petzoldt. **Selecting and Reducing Key Sizes for Multivariate Cryptography**. Tese de Doutorado, Instituto de Biofísica Carlos Chagas Filho - Universidade Federal do Rio de Janeiro / Brasil, 2013.

Albrecht Petzoldt, Jintai Ding, e Lih-chung Wang. Eliminating decryption failures from the simple matrix encryption scheme. **IACR Cryptology ePrint Archive**, 2016:10, 2016. URL http://eprint.iacr.org/2016/010.

Bruce Schneier. **Applied Cryptography (2Nd Ed.): Protocols, Algorithms, and Source Code in C**. John Wiley & Sons, Inc., New York, NY, USA, 1995. ISBN 0-471-11709-9.

Chengdong Tao, Adama Diene, Shaohua Tang, e Jintai Ding. **Post-Quantum Cryptography: 5th International Workshop, PQCrypto 2013, Limoges, France, June 4-7, 2013. Proceedings**. Capítulo Simple Matrix Scheme for Encryption, páginas 231–242, Springer Berlin Heidelberg, Berlin, Heidel-

berg, 2013. ISBN 978-3-642-38616-9. URL http://dx.doi.org/10.1007/978-3-642-38616-9_16.

Chengdong Tao, Hong Xiang, Albrecht Petzoldt, e Jintai Ding. Simple matrix - a multivariate public key cryptosystem (mpkc) for encryption. **Finite Fields Appl.**, 35(C):352–368, Setembro 2015. ISSN 1071-5797. URL http://dx.doi.org/10.1016/j.ffa.2015.06.001.

Enrico Thomae e Christopher Wolf. **Public Key Cryptography – PKC 2012: 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings**. Capítulo Solving Underdetermined Systems of Multivariate Quadratic Equations Revisited, páginas 156–171, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-30057-8. URL http://dx.doi.org/10.1007/978-3-642-30057-8_10.

Christopher Wolf. **Multivariate Quadratic Polynomials in Public Key Cryptography**. Tese de Doutorado, Katholieke Universiteit Leuven, The address of the publisher, 2005.

Appendix A

A.1 Minha Inspiração

.....

A.1.1 Como realizar minha Tese/Dissertação

.....

Appendix B

Título do Apêndice

B.1 Desenvolvendo a minha inspiração

.....