Bachelor's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

Anojan Skanthamany

Multivariate Public Key Cryptography

Bachelor's thesis in Mathematical Sciences
Supervisor: Kristian Gjøsteen
June 2022

NTNU
Kunnskap for en bedre verden

Anojan Skanthamany

Multivariate Public Key Cryptography

Bachelor's thesis in Mathematical Sciences
Supervisor: Kristian Gjøsteen
June 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

NTNU
Kunnskap for en bedre verden

Abstract

In this thesis, the goal is to introduce the reader to a few multivariate public key cryptography systems. We will go through three such systems. These are Matsumoto Imai, Oil and Vinegar, and Rainbow. We look at the construction, a few examples, and attacks against these systems.

Contents

Chapter 1

Introduction

The thesis will introduce multivariate public key cryptography (MPKC), where the goal is to take the reader through a few systems/signature schemes. These are Matsumoto-Imai, Balanced and Unbalanced Oil-Vinegar, and Rainbow.

Chapter 3 will be about the Matsumoto Imai cryptosystem. We will first look into this system's construction and the signature scheme. Followed by an example, and then close the chapter by looking at the attack that was done by Patarin, using linearization equations.

We will continue to chapter 4 with an introduction to the Oil and Vinegar Signature Scheme. We will again look at the construction of this scheme. Then take a look at two examples, one for the Balanced scheme and one for the Unbalanced scheme. Followed by the attack on this scheme by Kipnis and Shamir, using invariant subspaces.

In chapter 5, we will look at the construction of the Rainbow Signature Scheme, followed by an example, and end the chapter with a brief look at some of the attacks against Rainbow.

Cryptography has played a significant role in the security of modern-day communication and is used in many applications, such as computer passwords, banking transaction cards, etc.

As the world and technology progress, we want the security around us to be better. However, here is where some problems occur.

Most commonly used cryptosystems are systems like Diffie-Hellman and RSA. Both these systems are based on mathematical problems that are difficult to break, where Diffie-Hellman is based on the difficulty of the known discrete-log problem over a large prime field. In contrast, RSA is based on the difficulty of factoring a large integer into a product of prime numbers.

These are secure schemes for the time being. However, as stated earlier, the evolution of technology might change this. Our "normal" computers may not be able to break DH/RSA given a reasonable time frame; however, quantum computers may be able to break these systems within a reasonable amount

of time. Hence there need to be cryptosystems that can stand against these quantum computers.

These last years, a wave of systems has been introduced and categorized as Multivariate public-key cryptosystems. Whose intention is to use multivariate quadratic polynomials (over a field k) to make it difficult for quantum computers to break.

Acknowledgement

This bachelor thesis was written in the spring semester of 2022 for my final semester at NTNU. I want to thank my supervisor Kristian Gjøsteen for all the help, explanations, and feedback that I have received. I want to especially thank Kristian for finding such an interesting topic to work on.

Chapter 2

Background

Definition 1 A public key encryption scheme (PKE) consists of three algorithms $(\mathcal{K}, \mathcal{E}, \mathcal{D})$.

1. Key generation algorithm $\mathcal{K}$; It does not take in anything. However, it returns an encryption key and a decryption key, written as ek and dk, respectively. There is a message set M_{ek} which is associated with each encryption key.

2. Encryption algorithm $\mathcal{E}(ek, m)$; Takes in an encryption key ek and message $m \in M_{ek}$ and returns a ciphertext c.

3. Decryption algorithm $\mathcal{D}(dk, c)$; Takes in a decryption key dk and a ciphertext c and returns a message m. In case of a decryption failure, the symbol $\perp$ will be shown.

Definition 2 The correctness of a public key encryption scheme is defined as $\forall m \in M_{ek}, (ek, dk) \longleftarrow \mathcal{K}$ we have the following

$$\mathcal{D}(dk, \mathcal{E}(ek, m)) = m$$

Note that in PKE, the encryption key ek is the public key, while the decryption key dk is the secret key. In other words, $(ek, dk) = (pk, sk)$ where pk and sk stand for public key and secret key, respectively.

Definition 3 A digital signature scheme consists of three algorithms $(\mathcal{K}, \mathcal{S}, \mathcal{V})$.

1. Key generation algorithm $\mathcal{K}$; It does not take in anything. However, it returns a signature and verification key, written as sk and vk, respectively. There is a message set M_{sk} or M_{vk} which is associated with each key.

2. Signing algorithm $\mathcal{S}(sk, m)$; Takes in a signing key sk and message $m \in M_{sk}$ and returns a signature σ.

3. Verification algorithm $\mathcal{V}(vk, m, \sigma)$; Takes in a verification key vk, a message $m \in M_{vk}$ and the signature σ. It then returns 0 or 1. If 1 is shown, we look at it as a valid signature, and if 0 is shown, then it is an invalid signature.

Definition 4 The correctness of the digital signature scheme is defined as the following.

$$\mathcal{V}(vk, m, \mathcal{S}(sk, m)) = 1$$

Chapter 3

Matsumoto-Imai

Matsumoto and Imai were one of the first to come forward with a new idea in 1988 for MPKCs. Matsumoto and Imai's new way of thinking brought much attention when they proposed C* (or MI) in Eurocrypt88. The idea was based on finding invertible maps on a field K (instead of in vector space k^n), then using this map as an invertible map over k^n.

Because of this, their system MI showed high efficiency and potential for practical use. MI was even offered as a candidate for security standards for the government of Japan. Nevertheless, Jacques Patarin was able to break MI before the final selections, where he used an algebraic attack that uses linearisation equations.

This would typically be the end of a cryptosystem. However, MI had a vital role in the field because of a new mathematical idea later explored and extended. There have been new variants of the MI system with much potential. Sflash is one of them. One can look at Matsumoto Imai as a catalyst for a new way of handling MPKCs.

In this chapter, we will look at the construction of the Matsumoto Imai system and a brief look at the signature scheme. Followed by an example and the attack based on linearization equations done by Patarin.

3.1 Construction

Let k be a finite field of characteristic two, that is $k = GF(2)$ and cardinality q. Let $g(x) \in k[x]$ be any irreducible polynomial of degree n. We define the field K as $k[x]/p(x)$, a degree n extension of k.

Definition 5 Let $\phi : K \longrightarrow \mathrm{k}^n$ be the standard k-linear isomorphism between K and k^n, then its function is given by:

$$\phi(a_0 + a_1 x + ... + a_{n-1} x^{n-1}) = (a_0, a_1, ..., a_{n-1})$$

Hence,

$$\phi^{-1}(a_0, a_1, ..., a_{n-1}) = a_0 + a_1 x + ... + a_{n-1}x^{n-1} = \sum_{i=0}^{n-1} a_i x^i$$

Now we have to pick θ, which has to satisfy these two conditions

$$0 < \theta < n$$

and,

$$gcd(q^\theta + 1, q^n - 1) = 1.$$

Now that θ is chosen, and the conditions are satisfied, then we can define our map F over K, which is invertible because of the conditions of θ.

$$F(X) = X^{1+q^\theta}$$

Assume that we have an integer t such that:

$$t(1 + q^\theta) \equiv 1 mod(q^n - 1) \tag{3.1}$$

Then F^{-1} is given as:

$$F^{-1}(X) = X^t$$

Proposition 1 The inverse of $F(X)$ is X^t.

Proof

We know from number theory that

$$t(q^\theta + 1) \equiv 1 mod(q^n - 1)$$
$$= t(q^\theta + 1) = k(q^n - 1) + 1.$$

Therefore,

$$X^{t(q^\theta+1)} = X^{k(q^n-1)+1} = X^{k(q^n-1)}X.$$

We know that the degree of the extended field is q^n. Using this information together with Fermat's Little Theorem we have,

$$X^{t(q^\theta+1)} = X^{k(q^n-1)+1} = X^{k(q^n-1)}X = X.\square$$

Because $X^{k(q^n-1)} = 1$. $\square$

Let $\tilde{F}$ be the map over k^n defined by:

$$\tilde{F}(x_1, ..., x_n) = \phi \circ F \circ \phi^{-1}(x_1, ..., x_n) = (\tilde{f}_1, ..., \tilde{f}_n)$$

where $\tilde{f}_1, ..., \tilde{f}_n \in k[x_1, ..., x_n]$.

Now, let L_1 and L_2 be two invertible affine transformations over k^n.

$$\overline{F}(x_1, ..., x_n) = L_1 \circ \tilde{F} \circ L_2(x_1, ..., x_n) = (\overline{f}_1, ..., \overline{f}_n)$$

where $\overline{f}_1, ..., \overline{f}_n \in k[x_1, ..., x_n]$. This is for $i = 1, ..., n$. This encryption can be done by anyone since the map $\overline{F}$ is the public key.

Key Generation The key generation of the MI construction returns a public and a private key.

The public key is the field k, including its additive and multiplicative structure. It also includes the map $\overline{F}(x_1, ..., x_n)$.

The private key consists of the two invertible affine transformations L_1 and L_2. It can also include θ; however, it is unimportant. Because $0 < \theta < n$, and n usually is not large, therefore hiding θ will not be necessary.

Encryption of MI

Given a plaintext $p = (p_1, ..., p_n)$, the encrypted plaintext (ciphertext) will be

$$(c_1, ..., c_n) = \overline{f_i}(p_1, .., p_n)$$

for $i = 1, ..., n$.

Decryption of MI

Since $(c_1, ..., c_n) = \overline{f_i}(p_1, .., p_n)$ for $i = 1, ..., n$ we can decrypt the ciphertext like this:

$$\overline{F}^{-1}(c_1, ..., c_n) = L_2^{-1} \circ \tilde{F}^{-1} \circ L_1^{-1}(c_1, ..., c_n)$$

$$= L_2^{-1} \circ \phi \circ F^{-1} \circ \phi^{-1} \circ L_1^{-1}(c_1, ..., c_n).$$

Usually the components of $\overline{F}$ are of high degree; hence it is usual to decrypt the ciphertext $(c_1, ..., c_n)$ like this.

1. Compute $(c'_1, ..., c'_n) = L_1^{-1}(c_1, ..., c_n)$
2. Compute $(c''_1, ..., c''_n) = \phi \circ F^{-1} \circ \phi^{-1}(c'_1, ..., c'_n)$
3. Compute $(p_1, ..., p_n) = L_2^{-1}(c''_1, ..., c''_n)$

Even though the person who decrypts the ciphertext only knows (L_1, L_2), he still can find F^{-1}. Because t will be known by solving (3.1)

Correctness of MI

We want to show the correctness of this MI scheme.

Recall from Definition 2 that the correctness of a public key encryption is defined as:

$$\mathcal{D}(dk, \mathcal{E}(ek, m)) = m$$

From the MI construction we know that the encryption key (or public key) is $\overline{F} = (\overline{f}_1, .., \overline{f}_n)$.

This means,

$$\overline{F}(m) = L_1 \circ \tilde{F} \circ L_2(m) = L_1 \circ \phi \circ F \circ \phi^{-1} \circ L_2(m)$$

and the decryption key (secret key) is (L_1, L_2) and the decryption algorithm is:

$$L_2^{-1} \circ \phi \circ F^{-1} \circ \phi^{-1} \circ L^{-1}(c)$$

where, the ciphertext $c = (c_1, ..., c_n)$.

From the definition of correctness, we have

$$\begin{aligned} & L_2^{-1} \circ \phi \circ^{-1} \circ \phi^{-1} \circ L_1^{-1}(L_1 \circ \phi \circ F \circ \phi^{-1} \circ L_2(m)) \\ &= L_2^{-1} \circ \phi \circ F^{-1} \circ \phi^{-1} \circ \phi \circ F \circ \phi^{-1} \circ L_2(m) \\ &= L_2^{-1} \circ \phi \circ F^{-1} \circ F \circ \phi^{-1} \circ L_2(m) \\ &= L_2^{-1} \circ \phi \circ \phi^{-1} \circ L_2(m) \\ &= L_2^{-1} \circ L_2(m) \\ &= m. \end{aligned}$$

Hence the following theorem is proved:

Theorem 1 The Matsumoto Imai cryptosystem is correct.

MI signature scheme

Until now, we looked at MI as an encryption scheme, but for the sake of the other schemes that we will look at later, we want to introduce the algorithm of the MI signature scheme as well. However, it is not very different from the encryption scheme.

Key Generation The key generation is the same as in the encryption scheme. Hence, the public and private keys are the same as above.

Note In the signature scheme, we use sk as the *signature key* and vk as the *verification key* but this is the same as the private and public key, respectively.

Signature Generation Let $m = (m_1, ..., m_n)$ be a the document (or the hash value of the document) that needs to be signed.

The way the document gets signed is by doing the following.

$$\sigma = (\sigma_1, ..., \sigma_n) = \overline{F}^{-1}(m).$$

To get the signature we need to calculate $\overline{F}^{-1}(m)$. From the decryption algorithm of MI, we have

$$\overline{F}^{-1}(m) = L_2^{-1} \circ \phi \circ F^{-1} \circ \phi^{-1} \circ L_1^{-1}(m).$$

Signature Verification

The recipient has to check if

$$\overline{F}(\sigma) = m$$

where, $\sigma = (\sigma_1, ..., \sigma_n)$ and $m = (m_1, ..., m_n)$

Correctness of MI signature scheme

We want to check the correctness of the MI signature scheme just as we did with the encryption scheme.

Correctness is defined as

$$\mathcal{V}(vk, m, \mathcal{S}(sk, m)) = 1.$$

Both the verification algorithm and signature algorithm are known, as shown above. Hence we end up with the following.

$$\begin{aligned} & L_1 \circ \phi \circ F \circ \phi^{-1} \circ L_2(L_2^{-1} \circ \phi \circ F^{-1} \circ \phi^{-1} \circ L_1^{-1}(m)) \\ &= L_1 \circ \phi \circ F \circ \phi^{-1} \circ \phi \circ F^{-1} \circ \phi^{-1} \circ L_1^{-1}(m) \\ &= L_1 \circ \phi \circ F \circ F^{-1} \circ \phi^{-1} \circ L_1^{-1}(m) \\ &= L_1 \circ \phi \circ \phi^{-1} \circ L_1^{-1}(m) \\ &= L_1 \circ L_1^{-1}(m) \\ &= m. \end{aligned}$$

The verification algorithm $\mathcal{V}$ will therefore return 1.

Hence we have proved the following theorem:

Theorem 2 The Matsumoto Imai signature scheme is correct.

3.2 Example

Now we will look at a small example. Let $k = GF(2)$, with $q = 2$ elements. Let K be degree 5 extension. Hence value of $n = 5$ and we choose $\theta = 4$ since $1 < \theta < n$.

The field elements are $\{0, 1\}$. Now let $g(x) = x^5 + x^2 + 1$.

$x^5 + x^2 + 1$ is an irreducible polynomial in $k[x]$.

The map F is given by

$$F(X) = X^{2^4+1}.$$

Now, we find t by solving

$$(1 + 2^4)t \equiv 1 mod(2^5 - 1)$$
$$17t \equiv 1 mod(31).$$

Solving this, gives $t = 11$.

Now we can find the inverse map as well, that is

$$F^{-1}(X) = X^t = X^{11}.$$

Now we need to define L_1 and L_2 such that we can hide F. We want L_1 and L_2 be invertible maps.

Let L_1 and L_2 be given by:

$$L_1 = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} L_2 = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

.

Assume we have a plaintext

$$p = [1, 0, 0, 1, 1]^T.$$

We need to multiply our plaintext with L_2.

This gives us

$$L_2(p) = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}.$$

Now we apply ϕ^{-1} to $L_2(p)$

$$\phi^{-1}(L_2(p)) = 0 + (1)x + (1)x^2 + (0)x^3 + (1)x^4.$$

Here the coefficient in front of the x^i's is the a_i's, which was shown in the definition of ϕ.

We can now apply F to $\phi^{-1}(L_2(p))$.

$\mathrm{F}(\phi^{-1}(L_2(p)) = (\phi^{-1} \circ L_2(p))^{q^\theta+1}$
$= (\phi^{-1} \circ L_2(p))^{2^4} \cdot (\phi^{-1} \circ (L_2(p))$
$= (\mathrm{x} + \mathrm{x}^2 + x^4)^{16} \cdot (x + x^2 + x^4)$.

With the help of the irreducible polynomial $g(x) = x^5 + x^2 + 1$ and a computer program, we can simplify this large expression, which yields

$$F(\phi^{-1}(L_2(p)) = 1 + x + x^2.$$

Now we use the function ϕ.

$$\phi(F(\phi^{-1}(L_2(p)))) = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

Multiply the vector with L_1.

$$L_1(\phi(F(\phi^{-1}(L_2(p))))) = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}.$$

This is the ciphertext that corresponds to the plaintext $\mathrm{p} = [1, 0, 0, 1, 1]^T$.

We want to decrypt this ciphertext to show that we obtain the plaintext after the decryption.

To decrypt the ciphertext $c = [0, 0, 1, 0, 0]^T$ we do the following.

$$L_2^{-1}(\phi(F^{-1}(\phi^{-1}(L_1^{-1}(c))))).$$

We start by taking composing the inverse of L_1 with the ciphertext c.

$$L_1^{-1}(c) = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}.$$

Further, calculate $\phi^{-1}(L_1^{-1}(c))$.

$$\phi^{-1}((L_1^{-1}(c)) = 1 + x + x^2.$$

Now, we use the map F^{-1}, which gives

$$F^{-1}(\phi^{-1}(L_1^{-1}(c))) = (\phi^{-1}(L_1^{-1}(c)))^t = (\phi^{-1} \circ (L_1^{-1}(c)))^{11}$$
$$= (1 + x + x^2)^{11} = x + x^2 + x^4.$$

Now we have found $F^{-1}(\phi^{-1}(L_1^{-1}(c)))$. We will use the function ϕ to obtain

$$\phi(F^{-1}(\phi^{-1}(L_1^{-1}(c)))) = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}.$$

Now multiply this with L_2^{-1}.

$$L_2^{-1}(\phi(F^{-1}(\phi^{-1}(L_1^{-1}(c))))) = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}.$$

This is the plaintext we started with before encrypting it.

3.3 Linearization equation attack

This section will look at Patarin's attack against the Matsumoto Imai cryptosystem. Under we will define a linearization equation (LE).

Definition 6 Let $\overline{F} = \{\overline{f}_1, ..., \overline{f}_m\}$ be the public key of a multivariate public key cryptosystem. A linearization equation for $\overline{F}$ is any polynomial equation in $k[p_1, ..., p_n, c_1, ..., c_m]$ of the form

$$\sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_{ij} p_i c_j + \sum_{i=1}^{n} \beta_i p_i + \sum_{j=1}^{m} \gamma_j c_j + \delta$$

such that, when we substitute the plaintext/ciphertext pair (p, c) in the linearization equations, we get the zero. Also, when substituting a ciphertext $(c_1, ..., c_1)$ into the linearization equation, we get a linear equation in the plaintext variables $p_1, ..., p_n$. This means substituting a random ciphertext into c_j, yields a linear equation with only the p_i's as variables.

The question is, how do we use these linearization equations to break the Matsumoto Imai system?

It is done by doing the following.

First we need to compute $(m+1)(n+1)$ plaintext/ciphertext pairs $(p_1, c_1), ...,$ $(p_{(m+1)(n+1)}, c_{(m+1)(n+1)})$ We compute these pairs by choosing some random plaintexts p_i and then compute $\overline{F}(p_i) = c_i$, for $i = 1, ..., (m+1)(n+1)$.

After computing $(m+1)(n+1)$ plaintext/ciphertext pairs, then substitute the pairs, into the linearization equations, such that we obtain a linear system with the coefficients $\alpha_{ij}, \beta_i, \gamma_j, \delta$. Then we have bilinear equations $b_1, ..., b_k$ (where $b_1, ..., b_k$ are all linearly independent) with the variables $p_1, ..., p_n$ and $c_1, ..., c_m$.

Then we substitute the challenge ciphertext c^* into the equations $b_1, ..., b_k$ and then we obtain linear equations in the plaintext variables $p_1, ..., p_n$.

Patarin proposed the linearization equations attack against the Matsumoto Imai cryptosystem. Remember in the standard Matsumoto Imai system, $m = n$ and,

$$Y = F(X) = X^{q^\theta+1},$$

for $X, Y \in K$.

Raise both sides to the power of $q^\theta - 1$ this gives

$$\begin{aligned} Y^{q^\theta-1} &= (X^{q^\theta+1})^{q^\theta-1} \\ Y^{q^\theta-1} &= X^{(q^\theta+1)(q^\theta-1)} \\ Y^{q^\theta-1} &= X^{(2q^\theta-1)}. \end{aligned}$$

Now multiply both sides by XY, this yields

$$XY^{q\theta} = X^{q^{2\theta}}Y,$$

this is the same as

$$XY^{q^\theta} - X^{q^{2\theta}}Y = 0.$$

We now define $R(X, Y) \in K[X, Y]$ by

$$R(X, Y) = XY^{q\theta} - X^{q^{2\theta}}Y = 0,$$

and we define $\tilde{R}$ as

$$\tilde{R} = \phi \circ R \circ (\phi^{-1} \times \phi^{-1}).$$

Looking at $\tilde{R}$, when substituting a plaintext/ciphertext pair, $\tilde{R}$ will equal zero because of the Frobenius isomorphism. We have the case that $X \longrightarrow X^{q^\theta}$ and $Y \longrightarrow Y^{q^{2\theta}}$ is linear. When substituting one of the pairs, we will achieve n linear equations with degree 1. This is both for the values in X and Y. Hence this demonstrates that $\tilde{R}$ consists of a set of $\mathcal{L}$ linear equations. Note that $\mathcal{L}$ is a vector space since it is closed under addition and multiplication.

We have shown that there exists a set of linear equations, from $\tilde{R}$. Now the question is, how many linearly independent equations do we get after substituting a ciphertext in $\tilde{R}$.

Lemma 1 For a fixed $\tilde{Y} \in K$ there exists at most, $q^{gcd(\theta,n)}$ different values of $X \in K$ such that

$$R(X, \tilde{Y}) = 0$$

.

Proof

We have the following.

$$X\tilde{Y}^{q\theta} = X^{q^{2\theta}}\tilde{Y}$$

this gives for $\tilde{Y} \neq 0$

$$\tilde{Y}^{q^{\theta}-1} = X^{q^{2\theta}-1}.$$

And this has at most $gcd(q^{2\theta} - 1, q^n - 1)$ different solutions for X in K, now,

$$gcd(q^{2\theta} - 1, q^n - 1) = gcd((q^{\theta} - 1)(q^{\theta} + 1), q^n - 1)$$
$$= gcd(q^{\theta} - 1, q^n - 1)gcd(q^{\theta} + 1, q^n - 1) = gcd(q^{\theta} - 1, q^n - 1).$$

This is because of the condition we had in the construction that $gcd(q^{\theta}+1, q^n - 1) = 1$.

This means $X\tilde{Y}^{q\theta} = X^{q^{2\theta}}\tilde{Y}$ has at most $gcd(q^{\theta} - 1, q^n - 1) + 1$ (we have $+1$ because of the trivial solution).

However,

$$gcd(q^{\theta} - 1, q^n - 1) = q^{gcd(\theta,n)} - 1$$

.

Meaning, number of solutions is at most $q^{gcd(\theta,n)}$. □

This means when substituting a ciphertext c^* we get at most $q^{gcd(\theta,n)}$, this forms a space of a dimension less than $q^{gcd(\theta,n)}$, hence there will be $n - gcd(\theta, n)$ linearly independent equations.

However, we are interested in seeing how many linearly independent equations we receive in the plaintext variables when we substitute the challenge ciphertext.

Theorem 3 Let $\overline{F}$ be a Matsumoto Imai public key, after substituting the challenge ciphertext $\hat{c} \in k^n\ 0\}$, in the linearization equations from $\overline{F}$, we get at least

$$n - gcd(n, \theta) \geq \frac{2n}{3}$$

linearly independent linear equations, in the plaintext variable, $p_1, ..., p_n$.

Proof

To prove this, we have to introduce some notations.

$\mathcal{L}$ is the space of linearization equations derived from the map $\tilde{F}$.

$\overline{\mathcal{L}}$ is the space of linearization equations derived from the public key map $\overline{F}$.

$\mathcal{L}_{c^*}$ is the space of linear equations, with plaintext variables, that we derive after substituting the ciphertext c^* into the linearization equations from $\mathcal{L}$.

$\overline{\mathcal{L}}_{c^*}$ is the space of linear equations, with plaintext variables, that we derive after substituting the ciphertext c^* into the linearization equations from $\overline{\mathcal{L}}$.

Remark: There are a few different notations used in this sub chapter. However, these are the difference the reader should know p, c are generally speaking plaintext, ciphertext. p^*, c^* some chosen plaintext,ciphertext. $\hat{c}$ challenger ciphertext, i.e., the ciphertext whose plaintext we want to find.

As we can see, the linearization equations derived from $\tilde{R}$ are contained in $\mathcal{L}$. We know that the dimension of $\mathcal{L}_{c^*}$ is $n - gcd(n, \theta)$ because of Lemma 1, where we showed the substitution of a fixed ciphertext for the linearization equations in $\mathcal{L}$. Now our task is to show that.

$$dim(\overline{\mathcal{L}}_{c^*}) = n - gcd(n, \theta) \geq \frac{2n}{3}.$$

We will prove the theorem by using three Lemmas.

Lemma 2

$$dim(\overline{\mathcal{L}}) = dim(\mathcal{L})$$

.

Proof

We want to find a bijection between these two spaces, that is, a bijection between $\overline{\mathcal{L}}, \mathcal{L}$. This bijection will show the equality between the spaces, i.e., show that the dimension of the spaces is the same. We start by assuming one of the transformations is the identity, while write the other transformation using coefficients and variables (because we want to have it in polynomial form), then we start with one of the sets $\tilde{F} = (\tilde{f}_1, ..., \tilde{f}_n), \overline{F} = (\overline{f}_1, ..., \overline{f}_n)$, and do linearization equations for this set, using the form in (Definition 6). After substituting the transformation, we want to rearrange the linearization equation such that it can be passed on to the other set. Doing this for both sets will lead to a bijection, showing that the dimensions are the same.

First, assume L_2 is the identity matrix. This gives

$$\overline{f}_i(p_1, ..., p_n) = \sum_{j=1}^{n} s_{ij} \tilde{f}_i(p_1, ..., p_n) + s_{i0}.$$

Now, let

$$r = \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_{ij} p_i c_j + \sum_{i=1}^{n} \beta_i p_i + \sum_{j=1}^{m} \gamma_j c_j + \delta$$

be the linearization equation for the public polynomials $(\overline{f}_1, ..., \overline{f}_n)$, hence when substituting plaintext/ciphertext pair $(p^*, \overline{F}(p^*))$, we get 0. Meaning,

$$0 = \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_{ij} p_i^* \overline{f}_j(p^*) + \sum_{i=1}^{n} \beta_i p_i^* + \sum_{j=1}^{m} \gamma_j \overline{f}_j(p^*) + \delta$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_{ij} p_i^* (\sum_{k=1}^{n} s_{jk} \tilde{f}_k(p^*) + s_{j0}) + \sum_{i=1}^{n} b_i p_i^* + \sum_{j=1}^{m} \gamma_j (\sum_{k=1}^{n} s_{jk} \tilde{f}_k(p^*) + s_{j0}) + \delta$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha'_{ij} p_i^* \tilde{f}_j(p^*) + \sum_{i=1}^{n} \beta'_i p_i^* + \sum_{j=1}^{m} \gamma'_j \tilde{f}_j(p^*) + \delta'.$$

The second equality shows the substitution, while the third equality shows the rearrangement, we now have linearization equations for the other set, in this case, the set of $(\tilde{f}_1, ..., \tilde{f}_n)$.

The same goes for $\tilde{F} = L_1^{-1} \circ \overline{F}$, meaning when we start with linearization equations in the set $\tilde{F} = (\tilde{f}_1, ..., \tilde{f}_n)$, we can get linearization equations for $\tilde{F} = L_1^{-1} \circ \overline{F}$.

Hence there is an isomorphism between the spaces $\tilde{F}$ and $\overline{F}$ this implies both spaces have the same dimension.

Now, assume L_1 is the identity matrix. This leaves us with $\overline{F} = \tilde{F} \circ L_2$.

Let,

$$\overline{p}_i = L_2(p)_i = \sum_{j=1}^{n} t_{ij} p_j + t_{i0}$$

this means,

$$\overline{f}_i(p_1, ..., p_n) = \tilde{f}_i(\overline{p}_1, ..., \overline{p}_n)$$

.

We do the same as above by having r as the linearization equation for polynomials $\tilde{f}_i$.

$$0 = r(p^*, \tilde{F}(p^*)) = \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_{ij} p_i^* \tilde{f}_j(p^*) + \sum_{i=1}^{n} \beta_i p_i^* + \sum_{j=1}^{m} \gamma_j \tilde{f}_j(p^*) + \delta.$$

The invertible change of variables above, amounts to a perturbation of k^n, which gives

$$0 = \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_{ij}\overline{p_i^*}\tilde{f}_j(\overline{p^*}) + \sum_{i=1}^{n}\beta_i\overline{p_i^*} + \sum_{j=1}^{m}\gamma_j\tilde{f}_j\overline{p^*}) + \delta.$$

Because of $\overline{f}_i(p_1, ..., p_n) = \tilde{f}_i(\overline{p}_1, ..., \overline{p}_n)$ we have,

$$0 = \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_{ij}\overline{p_i^*}\overline{f}_j(p^*) + \sum_{i=1}^{n}\beta_i\overline{p_i^*} + \sum_{j=1}^{m}\gamma_j\overline{f}_j(p^*) + \delta.$$

Now we get the following, because of $\overline{p}_i = L_2(p)_i = \sum_{j=1}^{n} t_{ij}p_j + t_{i0}$.

$$0 = \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha'_{ij}p_i^*\overline{f}_j(p^*) + \sum_{i=1}^{n}\beta_i p_i^* + \sum_{j=1}^{m}\gamma_j\overline{f}_j(p^*) + \delta.$$

This an linearization equation for the public polynomials.

The same way goes for $\tilde{F} = \overline{F} \circ L_2^{-1}$. $\square$

Lemma 3

$$\mathcal{L}_{c^*} = \overline{\mathcal{L}}_{c^*}.$$

From Lemma 2, we saw that there is a bijection between $\mathcal{L}$ and $\overline{\mathcal{L}}$, this means there is an bijection between $\mathcal{L}_{c^*}$ and $\overline{\mathcal{L}}_{c^*}$. $\square$

We know from Lemma 1, that the dimension of $\mathcal{L}_{c^*} = n - gcd(n, \theta)$, from Lemma 3 we saw that $dim(\mathcal{L}_{c^*}) = dim(\overline{\mathcal{L}}_{w^*})$, now we need to show the lower bound for the dimension.

Lemma 4 For the Matsumoto Imai cryptosystem, we have

$$n - gcd(n, \theta) \geq \frac{2n}{3}$$

.

Proof

We want to prove that $gcd(n, \theta)$ has to be less or equal to $\frac{n}{3}$. This is done by showing that it cannot be either n or $n/2$.

From the construction of Matsumoto Imai system, we defined θ as $1 < \theta < n$ meaning $\theta \neq n$.

If $gcd(n, \theta) = \frac{n}{2}$ then θ has to be $\frac{n}{2}$, however if $\theta = \frac{n}{2}$, then

$$gcd(q^{n-1}, q^{\theta+1}) = gcd(q^{(n/2+1)(n/2-1)}, q^{(n/2+1)}) = q^{(n/2+1)} > 1.$$

Which is a contradiction, since θ was chosen such that $gcd(q^{n-1}, q^{\theta+1}) = 1$.

Therefore, $gcd(n, \theta) \leq \frac{n}{3}$ $\square$

Hence, with the help of Lemma $2 - 4$, Theorem 3 is proved. $\square$

Chapter 4

Oil and Vinegar Signature Scheme

As mentioned in the chapter of Matsumoto Imai, the role of their cryptosystem had much effect on future systems. One of these systems was the Oil and Vinegar signature scheme, which Patarin proposed in 1997. He converted the linearization equation attack on MI into the Oil Vinegar signature scheme.

The system is based on hiding quadratic equations consisting of "Oil" variables and "Vinegar" variables over a finite field k, using linear secret functions.

A difference between Oil Vinegar and Matsumoto Imai is that the private polynomials are random in Oil Vinegar but not in Matsumoto Imai. Hence in Oil Vinegar, the map of private polynomials F will not be composed with random invertible matrices on both sides, just one. Both operate with quadratic polynomials; however, in Matsumoto Imai, we did a field extension, but this is not the case in the Oil Vinegar scheme. Also, Matsumoto Imai has both an encryption system and a signature scheme, while Oil Vinegar only has a signature scheme. This is because of the algorithm for finding the inverse of the map F.

In this chapter, we will study the construction of the Oil Vinegar signature scheme, followed by an example both for the balanced and unbalanced cases, and end the chapter by looking at the attack by Kipnis and Shamir using invariant subspaces.

4.1 Construction of Oil and Vinegar

Definition 7 An Oil-Vinegar polynomial is any polynomial with a total degree of two $f \in k[x_1, .., x_o, \tilde{x}_1, .., \tilde{x}_v]$ of the form.

$$f = \sum_{i=1}^{o}\sum_{j=1}^{v} a_{ij}x_i\tilde{x}_j + \sum_{i=1}^{v}\sum_{j=1}^{v} b_{ij}\tilde{x}_i\tilde{x}_j + \sum_{i=1}^{o} c_i x_i + \sum_{j=1}^{v} d_j\tilde{x}_j + e,$$

where $a_{ij}, b_{ij}, c_i, d_j, e \in k$.

Definition 8 Let $F : k^n \longrightarrow k^o$ be a polynomial map of the form.

$$F(x_1, ...x_o, \tilde{x}_1, ..., \tilde{x}_v) = (f_1, ..., f_o),$$

where the $f_1, ..., f_o \in k[x_1, ..., x_o, \tilde{x}_1, ..., \tilde{x}_v]$ are Oil and Vinegar polynomials. Then F is called an Oil-Vinegar map.

The main idea of the Oil-Vinegar map F is that we want to invert this map for a fixed vector (for instance $m = (m_1, ..., m_o) \in k^o$), we do this by choosing randomly (or if we are given) a vector $(\tilde{x}'_1, ..., \tilde{x}'_v)$ that we assign to the Vinegar variables. This will give us a set of linear equations of just Oil variables given by

$$F(x_1, ..., x_n, \tilde{x}_1, ..., \tilde{x}_n) = (m_1, ..., m_o).$$

The inverse is of $(m_1, ..., m_o)$ under F is given by

$$F^{-1}(m_1, ..., m_o) = (x^*_1, ...x^*_o).$$

Let us take a closer look at the Oil-Vinegar map F.

Assume again that we have $(m_1, ..., m_o) \in k^o$ which is a fixed vector and the Vinegar variables $(\tilde{x}'_1, ..., \tilde{x}'_v) \in k^v$, which we will give some value, leaving us with $(x^*_1, ..., x^*_o)$ that satisfies

$$F(x^*_1, ..., x^*_o, \tilde{x}'_1, ..., \tilde{x}'_v) = (m_1, ..., m_o).$$

The inverse is given as

$$F^{-1}(m_1, ..., m_o) = (x^*_1, ..., x^*_o).$$

Notice that the notation of F^{-1} does not show that we depend on the value of $(\tilde{x}'_1, ..., \tilde{x}'_v) \in k^v$, however we will only be concerned whether or not $F^{-1}(m_1, ..., m_o)$ exists for a given value of $(\tilde{x}'_1, ..., \tilde{x}'_v)$.

From here, we choose the map F and then hide it. This is done by using an invertible and affine map $L : k^n \longrightarrow k^n$, which is of the form

$$(x_1, ..., x_o, \tilde{x}_1, ..., \tilde{x}_v) = L(z_1, ..., z_n).$$

Then this is composed with the Oil Vinegar map F.

That leaves us with the map $\overline{F} : k^n \longrightarrow k^o$ defined by

$$\overline{F} = F \circ L = (\overline{f}_1, ..., \overline{f}_o).$$

Key Generation The key generation returns a public key (the verification key) and a secret key (the signature key).

The public key consists of the field k, including the additive and multiplicative structure. Also the map $\overline{F} = F \circ L$.

The private key consists of the invertible affine transformation $L : k^n \longrightarrow k^n$ and the Oil Vinegar map F.

Signature Generation:

Let $m = (m_1, ..., m_o) \in k^o$ be a document (or the hash of a document) that needs to be signed. The signer has to first compute

$$(x_1^*, ..., x_o^*) = F^{-1}(m_1, ..., m_n)$$

for some random choice of $(\tilde{x}_1', ..., \tilde{x}_v') \in k^v$.

This is the same as solving the linear system

$$F(x_1, ..., x_o, \tilde{x}_1, ..., \tilde{x}_v) = (m_1, ..., m_o).$$

(Because, as we mentioned above, we choose random values for the Vinegar variables, leaving us with the linear equations with only Oil variables).

The signer now computes the signature of $(m_1, ..., m_o)$ as

$$\sigma = (\sigma_1, ..., \sigma_n) = L^{-1}(x_1^*, ..., x_o^*, \tilde{x}_1, ..., \tilde{x}_v).$$

Signature Verification:

To check if $\sigma = (\sigma_1, ..., \sigma_n)$ is a valid signature for the message $(m_1, ..., m_o')$ the recipient simply see if

$$\overline{F}(\sigma_1, .., \sigma_n) = (m_1, ..., m_o).$$

Correctness of the Oil Vinegar signature scheme

We want to look at the correctness of the Oil Vinegar signature scheme. From Definition 4 we have that correctness of a signature scheme is

$$\mathcal{V}(vk, m, \mathcal{S}(sk, m)) = 1.$$

Where vk and sk are the verification key and signature key, respectively.

The verification key vk is $\overline{F} = F \circ L$, and the signature key sk is F and L. Since the signature $\sigma = L^{-1} \circ F^{-1}(m)$, we simply need to show

$$\begin{aligned} & F \circ L(L^{-1} \circ F^{-1}(m)) \\ &= F \circ F^{-1}(m) \\ &= m. \end{aligned}$$

The verification algorithm $\mathcal{V}$ will return 1.

Hence, the following theorem is proved:

Theorem 4 The Oil and Vinegar Signature Scheme is correct.

4.2 Example o = v

We will look at a small example of the Oil Vinegar signature scheme.

We will use $GF(2)$ like we did in the MI example. Let $n = 6$ since this is a scheme where $o = v$, we have that $o = v = 3$. Let

$$x = [x_1, x_2, x_3, \tilde{x}_1, \tilde{x}_2, \tilde{x}_3].$$

Since $o = 3$, there will be three random polynomials f_1, f_2, f_3 of the form shown in Definition 7. The polynomials are shown below.

$$f_1 = x_1\tilde{x}_1 + x_1\tilde{x}_2 + x_2\tilde{x}_1 + x_2\tilde{x}_2 + x_3\tilde{x}_2 + x_3\tilde{x}_3 + \tilde{x}_1\tilde{x}_3 + \tilde{x}_2^2 + \tilde{x}_3^2.$$
$$f_2 = x_1\tilde{x}_1 + x_1\tilde{x}_3 + x_2\tilde{x}_2 + x_2\tilde{x}_3 + x_3\tilde{x}_1 + \tilde{x}_1^2 + \tilde{x}_1\tilde{x}_2 + \tilde{x}_1\tilde{x}_3 + \tilde{x}_2^2 + \tilde{x}_2\tilde{x}_3 + \tilde{x}_3^2.$$
$$f_3 = x_1\tilde{x}_3 + x_2\tilde{x}_2 + x_2\tilde{x}_3 + x_3\tilde{x}_3 + \tilde{x}_1\tilde{x}_1 + \tilde{x}_3^2.$$

Now we want to write these functions in bilinear form $f_i = x^T Q_i x$ for $i = 1, 2, 3$.

$$\mathbf{Q_1} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{Q_2} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{Q_3} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Let L be an invertible linear transformation in matrix form given by

$x = Lz$. Where $z = [z_1, z_2, ..., z_6]^T$.

$$\mathbf{L} = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \mathbf{L^{-1}} = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

$$\overline{f}_i(z_1, ..., z_n) = f_i(x_1, .., x_o, \tilde{x}_1, ..., \tilde{x}_v)$$
$$z^T \overline{Q}_i z = x^T Q_i x$$
$$z^T \overline{Q}_i z = (Lz)^T Q_i (Lz)$$
$$z^T \overline{Q}_i z = z^T (L^T Q_i L) z$$

This means that we can calculate

$$\overline{f}_i = z^T \overline{Q}_i z = z^T (L^T Q_i L) z$$

for $i = 1, 2, 3$.

The set of these new polynomials is the public key for this scheme.

$$\overline{f}_1 = z_1^2 + z_2 z_1 + z_2^2 + z_2 z_4 + z_2 z_5 + z_3^2 + z_3 z_4 + z_5 z_4 + z_5^2$$
$$\overline{f}_2 = z_2 z_3 + z_2 z_5 + z_3 z_1 + z_3^2 + z_4 z_1 + z_4^2 + z_6 z_1 + z_6 z_3 + z_6 z_4 + z_6 z_5$$
$$\overline{f}_3 = z_1 z_2 + z_2^2 + z_4 z_1 + z_5 z_1 + z_5 z_2 + z_5 z_3 + z_5^2 + z_6 z_3 + z_6 z_4.$$

Now that we have our public key, which consists of $\overline{F} = (\overline{f}_1, \overline{f}_2, \overline{f}_3)$, then use this to get our signature σ, and then verify it.

Let $m = (m_1, m_2, m_3) = (0, 1, 1)$ and the signature is $\sigma = (\sigma_1, .., \sigma_6)$.

We start by choosing random values for the vinegar variables

$$(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3) = (1, 1, 0)$$

so that we can find a valid signature σ . In this case a valid signature means that we find a solution for the linear system $f_i(x_1, x_2, x_3, 1, 1, 0) = m_i$. If the system doesn't have any solutions, we will try again but with different values for the vinegar variables $(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3)$.

We substitute the vinegar variables in $F = (f_1, f_2, f_3)$ this leaves us with

$$f_1(x_1, x_2, x_3, 1, 1, 0) = x_3 + 1$$
$$f_2(x_1, x_2, x_3, 1, 1, 0) = x_1 + x_2 + x_3 + 1$$
$$f_3(x_1, x_2, x_3, 1, 1, 0) = x_2 + 1.$$

Now we write $f_i(x_1, x_2, x_3, 1, 1, 0) = x_i'$, hence leaving us with

$$x_3 + 1 = 0$$
$$x_1 + x_2 + x_3 + 1 = 1$$
$$x_2 + 1 = 1$$

After solving the linear system, we get the following.

$$(x_1, x_2, x_3) = (1, 0, 1).$$

(Remember we are working in $GF(2)$, so we have $x_3 = 1$ for instance).

Before sending our signature σ and our message m, we want to check that there are not any mistakes by verifying if

$$F(1, 0, 1, 1, 1, 0) = (0, 1, 1).$$

After that, we find our signature σ that is

$$\sigma = (\sigma_1, .., \sigma_6) = L^{-1}(1, 0, 1, 1, 1, 0) = (0, 1, 0, 1, 0, 0).$$

We send the pair (σ, m) to the verifier, and the signature gets verified if

$$\overline{F}(0, 1, 0, 1, 0, 0) = (0, 1, 1).$$

4.3 Example o = 2v (UOV)

The Oil-Vinegar schemes can be divided into three groups. These are balanced Oil-Vinegar, unbalanced Oil-Vinegar, and Rainbow (multilayer construction that uses Oil-Vinegar at each layer). We have talked about the general construction of the Oil-Vinegar scheme (this goes for both balanced and unbalanced cases), and in the next chapter, we will talk about Rainbow. However, there will be a brief introduction of the unbalanced Oil and Vinegar scheme, with an example.

In UOV $o \neq v$, i.e., the amount of Oil variables differs from that of Vinegar variables.

The construction of the scheme is quite similar to the balanced OV construction. We have in total $n = o + v$ variables. A map L, that is invertible and maps from $k^n \longrightarrow k^n$. The public key consists of o - polynomials, just as in the balanced Oil Vinegar scheme. We will look at an example where $o = 2v$.

Let $n = 6$, this means $o = 2, v = 4$. We use $GF(2)$ as we have done earlier, where the elements are $\{0, 1\}$.

Let

$$x = [x_1, x_2, \tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \tilde{x}_4].$$

Where x_1, x_2 are Oil variables, while $\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \tilde{x}_4$ are Vinegar variables.

Again we define two polynomials with the form from Definition 6.

$$f_1 = x_1\tilde{x}_2 + x_1\tilde{x}_4 + x_2\tilde{x}_1 + x_2\tilde{x}_3 + x_2\tilde{x}_4 + \tilde{x}_1^2 + \tilde{x}_1\tilde{x}_2 + \tilde{x}_1\tilde{x}_3 + \tilde{x}_2^2 + \tilde{x}_2\tilde{x}_3 + \tilde{x}_3\tilde{x}_4 + \tilde{x}_4^2.$$

$$f_2 = x_1\tilde{x}_1 + x_1\tilde{x}_3 + x_2\tilde{x}_4 + \tilde{x}_1\tilde{x}_2 + \tilde{x}_1\tilde{x}_3 + \tilde{x}_2^2 + \tilde{x}_3^2 + \tilde{x}_3\tilde{x}_4 + \tilde{x}_4\tilde{x}_1 + \tilde{x}_4^2.$$

Now we want to write these functions in bilinear form $f_i = x^T Q_i x$

$$\mathbf{Q_1} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{Q_2} = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

We will use the same invertible map L as in the example above.

Further, we calculate $\overline{f}_i = z^T \overline{Q}_i z = z^T (L^T Q_i L) z$ for $i = 1, 2$. This is just the same process as we did in the example above.

The set $\overline{F}$ of these new polynomials is the public key for this scheme.

$$\overline{f}_1 = z_1z_2 + z_2z_3 + z_3z_6 + z_4z_5 + z_4z_6 + z_5z_4 + z_5z_6.$$

$$\overline{f}_2 = z_1z_2 + z_2^2 + z_3z_1 + z_3z_2 + z_3z_5 + z_4z_2 + z_5z_2 + z_6z_4 + z_6z_5.$$

Again assume we want to send a message $m = (m_1, m_2) = (1, 0)$ and the signature is $\sigma = (\sigma_1, .., \sigma_6)$. Now that there are only two Oil variables, $= (m_1, m_2)$ rather than $m = (m_1, m_2, m_3)$ which we had in our example above.

We start by choosing random values for the vinegar variables so that we can find a valid signature σ.

$$(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \tilde{x}_4) = (1, 0, 1, 1).$$

Substitute the vinegar variables in $\overline{F}$ this leaves us with

$$f_1(x_1, x_2, 1, 0, 1, 1) = x_1 + x_2,$$

$$f_2(x_1, x_2, 1, 0, 1, 1) = x_2 + 1.$$

Now we set $f_i(x_1, x_2, 1, 0, 1, 1) = m_i$.

$$x_1 + x_2 = 1,$$

$$x_2 + 1 = 0.$$

It is easily seen that $x_1 = 0$ and $x_2 = 1$.

Remark: As the example above if there isn't a solution for the linear system $f_i(x_1, x_2, 1, 0, 1, 1) = m_i$, then we go back and choose other vinegar variables.

Next, we want to check if there have not been any mistakes, we will check if

$$F(0, 1, 1, 0, 1, 1) = (1, 0).$$

Now we find our signature.

$$\sigma = (\sigma_1, .., \sigma_6) = L^{-1}(0, 1, 1, 0, 1, 1)^T = (1, 1, 0, 0, 0, 1).$$

We send the pair (σ, m) and verify that

$$\overline{F}(1, 1, 0, 0, 0, 1) = (1, 0).$$

4.4 Attack on the Balanced OV Signature Scheme

This section will look at the potent attack that Kipnis and Shamir proposed. This is against the Oil and Vinegar scheme where $n = 2o$, i.e., the balanced case. The goal is to find an equivalent key such that the forger can generate signatures for random messages. To simplify the description of the attack, we assume that the components of map F are homogeneous quadratic polynomials. $\overline{F} = F \circ L$ will be a homogeneous quadratic map too.

Definition 9 We define the Oil subspace $\mathcal{O}$ in k^n to be

$$\mathcal{O} = \{(x_1, ..., x_o, 0, ..., 0) | x_i \in k\}.$$

Definition 10 We define the Vinegar subspace $\mathcal{V}$ in k^n to be

$$\mathcal{V} = \{(0, ..., 0, \tilde{x}_1, ..., \tilde{x}_v) | \tilde{x}_i \in k\}.$$

In the balanced Oil Vinegar case, the subspace of the vectors in k^n in which the second half only contains zeros are in $\mathcal{O}$, and for all vectors in k^n in which the first half only contains zeros are in $\mathcal{V}$.

In this case, we have $(o = v)$ and,

$$2o = 2v = o + v = n.$$

Remark: To arrive at a common understanding of forgery, we will define the term below.

Forgery is a valid signature, which was created without the signing key.

What we want to achieve when attacking the balanced Oil Vinegar scheme is to forge the signature by recovering a key that is equivalent to the original private

key. This can be done with symmetric matrices for the corresponding quadratic form from the different polynomial in the public key.

We first look at the case where k is not of characteristic two. Later the case of characteristic two will be shown.

As described in section 4.1, we have $\overline{F} = F \circ L$ where, $\overline{F} : k^n \longrightarrow k^o$. $\overline{F}$ is the Oil Vinegar mapping with the components $\overline{f}_1, ..., \overline{f}_o \in k[z_1, ..., z_n]$. F and L are the private keys.

We define $z = (z_1, ..., z_n)^T$ as a n-dimensional column vector and $x = (x_1, ..., x_o, \tilde{x}_1, ..., \tilde{x}_o)^T$ is also a n-dimensional column vector, it is n-dimensional because $o + v = n$.

We also have $x = Lz$,

where L is a linear invertible $n \times n$ matrix.

As we defined at the start of this section, we will use $\mathcal{O}$ as the Oil space and $\mathcal{V}$ as the Vinegar space of F.

We want to find an invertible linear map $L' : k^n \longrightarrow k^n$ such that

$$L' \circ L^{-1}(\mathcal{O}) = \mathcal{O}.$$

Then we calculate a new Oil Vinegar map $F' : k^n \longrightarrow k^o$ defined by

$$F' = \overline{F} \circ (L')^{-1}.$$

This can be done since $\overline{F}$ is the public key.

The attacker can now use F' and L' to forge signatures because we have the following.

$$F \circ L = \overline{F} = F' \circ L'.$$

Now we write each quadratic part of each $\overline{f_i}(z_1, ..., z_n)$ as $\overline{q_i}(z_1, ..., z_n)$ for $i = 1, ..., o$. As mentioned above we are now working with k, which is not of characteristics two, hence there exists a unique symmetric matrix $\overline{Q}_i$ of size $n \times n$, that can be used to represent $\overline{q}_i(z_1, ..., z_n)$, the following way

$$z^T \overline{Q}_i z.$$

The same goes for every $f_i(x_1, ..., x_o, \tilde{x}_1, ..., \tilde{x}_o)$, we represent every q_i for $i = 1, ..., o$ as

$$x^T Q_i x.$$

The unique form of Q_i is

$$Q_i = \begin{pmatrix} 0 & B_{i1} \\ B_{1i} & B_{i2} \end{pmatrix}.$$

Hence, the intention is to write the polynomials in bilinear form, where the square matrix is symmetric.

Q_i is of size $n \times n$, and the four submatrices $0, B_{i1}, B_{1i}^T, B_{i2}$ are of size $o \times o$. The upper left $o \times o$ submatrix is a zero matrix because, from the definition of Oil Vinegar polynomials, no Oil variable is multiplied by another Oil variable.

We want to turn our attention to the relation between Q_i and $\overline{Q}_i$.

We know that

$$\overline{f}_i(z_1, ..., z_n) = f_i(x_1, ..., x_o, \tilde{x}_1, ..., \tilde{x}_o)$$
$$z^T \overline{Q}_i z = x^T Q_i x$$
$$z^T \overline{Q}_i z = z^T (L^T Q_i L) z.$$

This means that

$$\overline{Q}_i = L^T Q_i L,$$

and rearranging the equation makes

$$Q_i = (L^{-1})^T \overline{Q}_i L^{-1}.$$

Lemma 5 For any $u_1, u_2 \in \mathcal{O}$

$$u_1^T Q_i u_2 = 0$$

and, for any $w_1, w_2 \in L^{-1}(\mathcal{O})$ we have

$$w_1^T \overline{Q}_i w_2 = 0.$$

Proof: The first equation results from the definition of an Oil and Vinegar polynomial. Q_i is a matrix whose upper left $o \times o$ submatrix only contains zero. When this is multiplied with a column vector whose second half only contains zero, then we get a column vector whose first half only contains zero. Since, $u_1, u_2 \in \mathcal{O}$, then we can write them as $u_1 = (*, 0)^T$ same for u_2.

$$u_1^T Q_i u_2 = \begin{pmatrix} * & 0 \end{pmatrix} \begin{pmatrix} 0 & B_{i1} \\ B_{1i} & B_{i2} \end{pmatrix} \begin{pmatrix} * \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} * & 0 \end{pmatrix} \begin{pmatrix} 0 \\ * \end{pmatrix} = \begin{pmatrix} 0 \end{pmatrix}.$$

The second equation; $w_1 \in L^{-1}(\mathcal{O})$, this means $w_1 = L^{-1} \circ w_1'$, where $w_1' \in \mathcal{O}$, the same goes for w_2. Since $\overline{Q}_i = L^T Q_i L$ we have the following

$$\begin{aligned} & w_1^T \overline{Q}_i w_2 \\ &= (L^{-1} \circ w_1')^T L^T Q_i L (L^{-1} \circ w_2') \\ &= w_1'^T \circ (L^T)^{-1} L^T Q_i L L^{-1} \circ w_2' \\ &= w_1'^T \circ Q_i \circ w_2' = 0. \quad \square \end{aligned}$$

We want to find the pre-image of the Oil subspace under the map L.

Lemma 6

Let $H : k^n \longrightarrow k^n$ be a linear transformation such that

$$H = \begin{pmatrix} 0 & * \\ * & * \end{pmatrix}.$$

1) $H(\mathcal{O}) \subset \mathcal{V}$.

2) If H is invertible, then we have $H(\mathcal{O}) = \mathcal{V}$ and $H^{-1}(\mathcal{V}) = \mathcal{O}$.

Proof

1)Let u_1 be defined as in Lemma 5.

$$H(u_1) = \begin{pmatrix} 0 & * \\ * & * \end{pmatrix} \begin{pmatrix} * \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ * \end{pmatrix} \in \mathcal{V}.$$

From definition of a Vinegar subspace, we can clearly see that $H(\mathcal{O}) \subset \mathcal{V}$.

2)We know from 1) that $H(\mathcal{O}) \subset \mathcal{V}$. However since H is invertible, the image space of $H(\mathcal{O})$ has dimension o, and therefore it has to be $H(\mathcal{O}) = \mathcal{V}$ and $H^{-1}(\mathcal{V}) = \mathcal{O}$. $\square$

Let $\overline{Q}$ be the subspace of matrices spanned by the $\overline{Q}_i$ and let Q be the subspace spanned by Q_i. Let $\overline{W}_1$ and $\overline{W}_2$ be two non-singular elements in $\overline{Q}$, and let W_1 and W_2 be the corresponding matrices in Q.

Because of earlier definitions of $\overline{Q}_i$ and Q_i, we know $\overline{W}_i$ is of the form

$$\overline{W}_i = L^T W_i L = L^T \begin{pmatrix} 0 & W_{i1} \\ W_{i1}^T & W_{i2} \end{pmatrix} L.$$

We achieve this because $\overline{W}_i$ is a linear combination of the $\overline{Q}_i's$.

While its inverse is

$$\overline{W}_i^{-1} = (L^T W_i L)^{-1} = L^{-1} \frac{1}{-W_{i1} W_{i1}^T} \begin{pmatrix} W_{i2} & -W_{i1} \\ -W_{i1}^T & 0 \end{pmatrix} (L^{-1})^T$$
$$= L^{-1} \begin{pmatrix} -W_{i1}^{-1}(W_{i1}^T)^{-1} W_{i2} & (W_{i1}^T)^{-1} \\ W_{i1}^{-1} & 0 \end{pmatrix} (L^{-1})^T.$$

Next we define $\overline{W}_{ij} = \overline{W}_i^{-1} \overline{W}_j$.

$$\overline{W}_i^{-1} \overline{W}_j = ((L^T W_i L)^{-1})(L^T W_j L) = L^{-1} W_i^{-1} (L^T)^{-1} L^T W_j L = L^{-1} \begin{pmatrix} -W_{i1}^{-1}(W_{i1}^T)^{-1} W_{i2} & (W_{i1}^T)^{-1} \\ W_{i1}^{-1} & 0 \end{pmatrix} \begin{pmatrix} W$$

Where,

$V_{11} = (W_{i1}^T)^{-1} W_{j1}^T$,

$V_{12} = -W_{i1}^{-1}(W_{i1}^T)^{-1} W_{i2} W_{j1} + (W_{i1}^T)^{-1} W_{j2}$,

$V_{22} = W_{i1}^{-1} W_{j1}$.

Now we will define what an invariant subspace is.

Definition 11 Let G be a k-vector space and let $D : G \longrightarrow G$ be a linear transformation on G. A linear subspace $C \subset G$ is called an invariant subspace for D if

$$D(c) \in C \forall c \in C.$$

Corollary 1 The Oil subspace, is a common invariant subspace of all matrices $W_{12} = W_1^{-1} W_2$.

Proof

The form of W_2 is

$$W_2 = \begin{pmatrix} 0 & W_{21} \\ W_{21}^T & W_{22} \end{pmatrix}.$$

The exact form as the Q_i matrices we talked about earlier.

While

$$W_1^{-1} = \begin{pmatrix} -W_{22} W_{21}^{T\,-1} W_{21}^{-1} & W_{21}^T \\ W_{21} & 0 \end{pmatrix}.$$

Now, let u_1 be the same as described earlier.

$$W_{12} = W_1^{-1} W_2(\mathcal{O}) = \begin{pmatrix} -W_{22} W_{21}^{T\,-1} W_{21}^{-1} & W_{12}^T \\ W_{12} & 0 \end{pmatrix} \begin{pmatrix} 0 & W_{21} \\ W_{21}^T & W_{22}^T \end{pmatrix} \begin{pmatrix} * \\ 0 \end{pmatrix}.$$

We know from Lemma6, 1) that the first composition of $W_2(u_1)$ gives an element in the Vinegar subspace of the form $w' = (0, *)^T$. Now W_1^{-1} composed with w' will give

$$W_1^{-1}(w')\begin{pmatrix} -W_{22}W_{21}^{T\,-1}W_{21}^{-1} & W_{12}^T \\ W_{12} & 0 \end{pmatrix}\begin{pmatrix} 0 \\ * \end{pmatrix} = \begin{pmatrix} * \\ 0 \end{pmatrix} \in \mathcal{O}.$$

Hence, $\mathcal{O}$ is a common invariant subspace for $W_{12} = W_1^{-1}W_2$ where, W_1, W_2 are linear combinations of the matrices Q_i. $\square$

Theorem 5 The space $L^{-1}(\mathcal{O})$ is a common invariant subspace of all the matrices $\overline{W}_{12} = \overline{W}_1^{-1}\overline{W}_2$.

Proof

$$\overline{W}_{12} = \overline{W}_1^{-1}\overline{W}_2 = L^{-1}VL.$$

Where V is the matrix

$$\begin{pmatrix} V_{11} & V_{12} \\ 0 & V_{22} \end{pmatrix}.$$

The matrix V has the same form as the matrix W_{12}. We know from Lemma 5 that $\mathcal{O}$ is an invariant subspace of W_{12}. Hence we get

$$\overline{W}_{12}(L^{-1}(\mathcal{O})) = \overline{W}_1^{-1}\overline{W}_2(L^{-1}(\mathcal{O})) = L^{-1}VL(L^{-1}(\mathcal{O})) = L^{-1}V(\mathcal{O}) = L^{-1}(\mathcal{O}). \ \square$$

By using linear algebra, we can be able to find the subspace of $L^{-1}(\mathcal{O})$. After we have found $L^{-1}(\mathcal{O})$, we can use the relevant parts of the transformation L in $\overline{F} = F \circ L$ so that we can forge signatures for random messages. We will be looking at an algorithm for finding the space $L^{-1}(\mathcal{O})$.

The idea is the following we want to find a random linear combination $\overline{W}_1$ and $\overline{W}_2$ of the matrices $\overline{Q}_i$ that represents the bilinear form of the public polynomials. Where $\overline{W}_1, \overline{W}_2 \in \Omega$. We define Ω as the $span(\overline{Q}_1, ..., \overline{Q}_o)$. Then find $\overline{W}_{12} = \overline{W}_1^{-1}\overline{W}_2$.

After finding $\overline{W}_{12}$, the algorithm then finds the minimal invariant subspace (invariant subspace that does not contain any non-trivial invariant subspaces) of this matrix. The subspaces correspond to the irreducible factors of the characteristic polynomial of $\overline{W}_{12}$. However, each of these minimal invariant subspaces may not be a subspace of $L^{-1}(\mathcal{O})$. But this can be checked quickly due to Lemma 5 We keep continuing this process until we have o independent basis vectors for $L^{-1}(\mathcal{O})$.

Therefore the way of breaking the Oil Vinegar scheme is like this.

1. Write down the symmetric matrices $\overline{Q}_i$ that are associated with $\overline{f}_i$ for $i = 1, .., o$, where $\overline{f}_i$ is the $i'th$ public polynomial.

2. Choose $\overline{W}_1, \overline{W}_2 \in \Omega$ Here, $\overline{W}_1$, must be invertible, and calculate $\overline{W}_{12}$ if it is, else find another $\overline{W}_1$.

3. After calculating $\overline{W}_{12}$, find the characteristic polynomial $C(\lambda)$. Go back to step 2, unless $C(\lambda) = C_1(\lambda)^2$, i.e if $C(\lambda)$ only has quadratic factors.

4. Compute $C_1(\overline{W}_{12})$. Find a basis for the null space, and then extend the basis for k^n. I.e., we make a basis of the eigenvectors of dimension o.

In case there aren't o eigenvectors, make another $\overline{W}_{12}$ but with different linear combinations for $\overline{W}_1$ and $\overline{W}_2$. And repeat this until you have o eigenvectors.

Check if the eigenvectors lie in $L^{-1}(\mathcal{O})$, this is done by using 2) in Lemma 5.

After this, extend the basis from k^o to k^n by inserting basis vectors into the columns of L^{*-1}.

5. Use the basis to transform the public polynomials, into the Oil-Vinegar form. By calculating $f_i{}^* = (L^{*-1})^T \circ \overline{Q}_i \circ (L^{*-1})$

Where $\{F^*, L^*\}$, $F^* = (f_1^*, ..., f_o^*)$ are the equivalent private keys.

Now we have the equivalent private keys, which can be used to forge signatures.

The case of characteristic two When we are looking in characteristic two, we cannot make a symmetric matrix of $(L^{-1})^T \overline{Q}_i L^{-1}$. For a matrix $A = (a_{ij})$, if A is symmetric then $a_{ij} + a_{ji} = 0$, leaving a zero coefficient of $x_i \tilde{x}_j$ when working with characteristic two.

Hence we need symmetric matrices $\overline{S}_i$ to be on the form

$$\overline{S}_i = \overline{Q}_i + \overline{Q}_i^T .$$

This is symmetric because $\overline{Q}_i + \overline{Q}_i^T = (\overline{Q}_i + \overline{Q}_i^T)^T$.

The algorithm is very much the same. However, there are a few changes.

$C_1(\lambda)$ will be zero. This is because all the entries in the diagonal are zero. Hence we need to look for a distinctive linear factor $(\lambda - \lambda_1)$ of multiplicity one.

We calculate $\overline{W}_{12}$ the same way we did in the odd case. The eigenspace of W_{12} has dimension two, where one of the eigenvectors must be in $L^{-1}(\mathcal{O})$. We then try out the $(q+1)$ possible eigenvectors. These are the eigenvectors in the set

$$S_{eigv} = \{v_1 + kv_2\} \cup \{v_2\}$$

for $k \in GF(2)$.

Now, to see which eigenvector that is in the wanted invariant subspace, we will go through every vector in $s \in S_{eigv}$, together with arbitrary $\overline{W}_{12} = \overline{W}_1{}^{-1}\overline{W}_2$, and find which vector, that will generate the invariant subspace that we are after.

This is done by computing the image of the space spanned by the eigenvector under the action of $\overline{W}_{12}$, i.e., if we denote the space spanned by s as V_{eigs}, then we are looking for $\overline{W}_{12}(V_{eigs})$. Then set $V_{eigs} = V_{eigs} \cup \overline{W}_{12}(V_{eigs})$.

This step will be repeated $2o - 1$ times, or until the dimension of T is greater than o.

If the dimension is greater than o, then it cannot be the space $L^{-1}(\mathcal{O})$, we need to find new set of eigenvectors, by using another linear combination for $\overline{W}_1^{-1}, \overline{W}_2$. Given that the dimension is o for some s in S_{eigv}, we can extend the basis as we did in the case where k is odd and forge the signatures.

Chapter 5

Rainbow Signature Scheme

In 2004, Jintai Ding and Dieter Schmidt proposed their signature scheme called Rainbow. Rainbow is built on the Oil and Vinegar scheme that Patarin proposed. We saw in the previous chapter that the attack on the Oil Vinegar scheme proved it was not safe. The big difference between Rainbow and Oil Vinegar is that Rainbow is just an Oil Vinegar scheme but with multiple "layers" where each layer represents a set of Oil and Vinegar variables.

In this chapter, we will look at the construction of Rainbow, an example, and a brief look at the MinRank attack against Rainbow.

5.1 Construction of Rainbow

In this chapter, we will look at the construction of the Rainbow Signature Scheme. However, before describing the scheme, we need to understand a few notations that have an essential part of the scheme and the maps included.

Let V be the set $\{1, 2, 3, ..., n\}$. Let $v_1, ..., v_u$ be any set of u integers, where $u \leq n$ and $0 < v_1 < v_2 < ... < v_u = n$ and define $V_l = \{1, 2, ..., v_l\}$ for $l = 1, 2, ...u$ as sets of integers.

$$V_1 \subset V_2 \subset ... \subset V_u = V.$$

This is because $v_i < v_{i+1}$ for $i = 1, ..., u - 1$. Each V_l contains the integers 1 to v_l.

Further, let $o_i = v_{i+1} - v_i$ and $O_i = V_{i+1} - V_i$ for $i = 1, 2..., u - 1$, o_i is the number of elements in O_i.

Now let P_l be the linear space of quadratic polynomials spanned by the polynomials of the form

$$f = \sum_{i \in O_l} \sum_{j \in S_l} a_{ij} x_i x_j + \sum_{i \in S_l} \sum_{j \in S_l} b_{ij} x_i x_j + \sum_{i S_{l+1}} c_i x_i + d.$$

Given that $i \in O_l$ and $j \in V_l$, we say that x_i is an Oil variable and x_j is a Vinegar variable, respectively. These are also called the l^{th} layer Oil variable and l^{th} layer Vinegar variable if $x_i \in O_l$ and $x_j \in V_l$, respectively. This means P_l will be called the l^{th} layer Oil Vinegar polynomial.

Remark: In the Oil Vinegar chapter, we called and Oil variable x_i and a Vinegar variable $\tilde{x}_j$, in these scheme, we will call an Oil variable for x_i and a Vinegar variable for x_j and not $\tilde{x}_j$. This is because the Oil variables are the integers in $O_i = V_{i+1} - V_i$. Writing Vinegar variables as $\tilde{x}_j$ and Oil variables as x_i, can lead to unnecessary confusion.

It is easily seen that $P_i \leq P_j$ for $i \leq j$. This is because P_j which is the linear space of quadratic polynomials of the j^{th} layer Oil (O_j) and Vinegar (V_j) sets. These sets contains more elements than the sets on the i^{th} layer, that is why $P_i \leq P_j$.

Now, let

$$\tilde{F} = (F_1, ..., F_{u-1}),$$

where each F_i for $i = 1, ..., u-1$ is

$$F_i = (f_{i1}, ..., f_i o_i).$$

This means that the map $\tilde{F}$ contains the maps F_i which represents the different layers, that is why we have F_i for $i = 1, ..., u-1$. Now every F_i contains o_i random polynomials.

However, to simplify the notations, we can look at $\tilde{F}$ as

$$\tilde{F} = (f_1, ..., f_{n-v_1}).$$

This means the map $\tilde{F} : k^n \longrightarrow k^{n-v_1}$ contains all the $n - v_1$ random polynomials, that are from all the different layers.

We will take a look at the different layers in Rainbow. The way we see it is that the first layer consists of $x_1, ..., x_{v_1}$ Vinegar variables, together with the Oil variables $x_{v_1+1}, ..., x_{v_2}$.

And the next layer consists of $x_1, ..., x_{v_1}, x_{v_1+1}, ..., x_{v_2}$ Vinegar variables, along with $x_{v_2+1}, ..., x_{v_3}$ which are the Oil Variables of the second layer.

It continues like this until we reach the $u - 1^{th}$ layer.

Now we define the map $\overline{F} : k^n \longrightarrow k^{n-v_1}$.

$$\overline{F} = L_1 \circ \tilde{F} \circ L_2 = (\overline{f}_1, ..., \overline{f}_{n-v_1}),$$

where $L_1 : k^{n-v_1} \longrightarrow k^{n-v_1}$ and $L_2 : k^n \longrightarrow k^n$, are both randomly chosen invertible affine maps.

This is all the information we need to show how the Rainbow Signature Scheme works.

Key Generation

The Key Generation consists of the public and private keys.

The public key consists of the field k and including its additive and multiplicative structure. It also consists of $\overline{F}$ which consists of the polynomials $(\overline{f}_1, ..., \overline{f}_{n-v_1})$.

The private key consists of the maps L_1, L_2 and $\tilde{F}$.

Signature Generation

For us to sign the document, $m = (m, ..., m_{n-v_1}) \in k^{n-v_1}$ we need to find a solution for the equation

$$\overline{F}(x_1, ...x_n) = L_1 \circ \tilde{F} \circ L_2(x_1, ..., x_n) = m.$$

This is done by following these steps:

1) First, we start by taking the inverse of L_1, namely L_1^{-1} of the message

$$L_1^{-1}(m) = m' = (m'_1, ..., m'_{n-v_1}).$$

2) Now, we need to take the inverse of $\tilde{F}$. It is almost the same procedure as when we inverted F in the Oil Vinegar scheme. However, the difference is that we need to do it multiple times in Rainbow because of the layers.

$$\tilde{F}(x_1, ..., x_n) = m' = (m'_1, ..., m'_{n-v_1})$$

This is the equation we want to solve, and as mentioned, we do it the same way as we did in the Oil Vinegar Scheme. Therefore we start by choosing some random values for the vinegar variables $(x_1, ..., x_{v_1})$. We will use the notation $(\tilde{x}'_1, ..., \tilde{x}'_{v_1})$ for these random values.

3) Substitute these values into the first layer (F_1) of o_1 equations, this gives us

$$F_1(\tilde{x}'_1, ..., \tilde{x}'_{v_1}, x_{v_{n+1}}, ..., x_{v_2}) = (m'_1, ..., m'_{o_1}).$$

This equation represents the substituted randomly chosen Vinegar variables $\tilde{x}'_1, ..., \tilde{x}'_{v_1}$, together with the Oil Variables, $x_{v_{n+1}}, ..., x_{v_2}$. Which equals the first o_1 part of m' i.e $(m'_1, ..., m'_{o_1})$.

Given that we find a solution for the Oil Variables in the first layer (by solving the linear system of equations), we now have a set of variables that we substitute into the second layer. These variables are the Vinegar variables and the Oil Variables from the first layer. The values of the variables $x_{v_1}, ..., x_{v_2}$ will be substituted into the second layer. This produces o_2 linear equations with $x_{v_2+1}, ..., x_{v_3}$ as Oil Variables. Given that there exists a solution, we use these

values together with the earlier ones and substitute them for the Vinegar variables in the next layer, and continue this process until we get to the $u - 1^{th}$ layer.

Which will give us the solution

$$X^* = (x_1^*, ..., x_n^*).$$

Which means

$$\tilde{F}(X^*) = m'.$$

If there is no solution to one of the linear systems in one of the layers, we start from 2) again, however, with new values for the Vinegar variables. Nevertheless, it is expected with a high probability that we will succeed eventually, as long as the number of layers is not too many.

4) Lastly,

$$\sigma = (\sigma_1, ..., \sigma_n) = L_2^{-1}(X^*)$$

σ is the signature of $m = (m_1, ..., m_{n-v_1})$.

Signature verification

To verify the signature σ for $m = (m_1, ..., m_{n-v_1})$ we need to see if

$$\overline{F}(\sigma) = m.$$

The signature is valid if the equation is true. Else the signature is not valid.

Correctness of Rainbow

We will look at the correctness of the Rainbow Signature Scheme the same way we have done earlier with the other schemes.

Remember that the correction of a signature scheme is defined as

$$\mathcal{V}(vk, m, \mathcal{S}(sk, m)) = 1.$$

In Rainbow, the signature key is $L_1, L_2, \tilde{F}$, while the verification key is $\overline{F}$.

$$\overline{F} = L_1 \circ \tilde{F} \circ L_2,$$

where $\tilde{F}$ is a map, that contains the $n - v_1$ polynomials.

Sigma is defined as

$$\sigma = S(sk, m) = L_2^{-1} \circ \tilde{F}^{-1} \circ L_1^{-1}(m)$$

Therefore we have,

$$
\begin{aligned}
& L_1 \circ \tilde{F} \circ L_2(\sigma) \\
= & L_1 \circ \tilde{F} \circ L_2(L_2^{-1} \circ \tilde{F}^{-1} \circ L_1^{-1}(m)) \\
= & L_1 \circ \tilde{F} \circ \tilde{F}^{-1} \circ L_1^{-1}(m) \\
= & L_1 \circ L_1^{-1}(m) \\
= & m.
\end{aligned}
$$

The verification algorithm $\mathcal{V}$ will return 1.

Hence, following the theorem is proved:

Theorem 6 The Rainbow Signature Scheme is correct.

5.2 Example

We will look at an example of the Rainbow Signature Scheme with the following information.

Let $k = GF(2), n = 9$ and $v_u = 4$, where $v_1 = 2, v_2 = 4, v_3 = 7, v_4 = 9$, hence, $o_1 = 2, o_2 = 3, o_3 = 2$. This means the set $V = \{1, 2..., 9\}$. And we will have the following sets

$$V_1 = \{1, 2\}, V_2 = \{1, 2, 3, 4\}, V_3 = \{1, 2, 3, 4, 5, 6, 7\}, V_4 = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

and,

$$O_1 = \{3, 4\}, O_2 = \{5, 6, 7\}, O_3 = \{8, 9\}.$$

The scheme consists of three layers. Each layer consists of o_i polynomials of the form from the construction.

Remember from the construction that $\tilde{F} = (F_1, ..., F_{u-1}) = (f_1, ..., f_{n-v_1})$. We will, now use the map $\tilde{F} = (f_1, ..., f_{n-v_1})$ where each f_i is a Oil Vinegar polynomial.

There are in total $o_1 + o_2 + o_3 = 2 + 3 + 2 = 7$ polynomials.

Polynomials - first layer

$\mathrm{f}_1 = x_1x_3 + x_2x_3 + x_2x_4 + x_1^2 + x_2^2$.

$\mathrm{f}_2 = x_1x_3 + x_2x_3 + x_2x_4 + x_1x_2$.

Polynomials - second layer

$\mathrm{f}_3 = x_1x_5 + x_1x_7 + x_2x_5 + x_2x_6 + x_2x_7$
$+ x_3x_6 + x_4x_5 + x_4x_7 + x_1x_2 + x_2^2 + x_2x_3 + x_3^2 + x_4^2$.

$\mathrm{f}_4 = x_1x_6 + x_2x_5 + x_2x_7 + x_3x_5 + x_3x_7$
$+ x_4x_7 + x_1^2 + x_1x_3 + x_1x_4 + x_2^2$.

$\mathrm{f}_5 = x_1x_6 + x_1x_7 + x_2x_6 + x_3x_7 + x_4x_5 + x_4x_7 + x_1x_3 + x_2x_4 + x_3^2 + x_3x_4 + x_4^2$.

Polynomials - third layer

$f_6 = x_1x_8 + x_2x_8 + x_2x_9 + x_3x_8 + x_4x_8 + x_5x_8 + x_5x_9$
$+ x_6x_9 + x_7x_8 + x_7x_9 + x_1^2 + x_1x_3 + x_1x_4$
$+ x_2^2 + x_2x_5 + x_4^2 + x_4x_7 + x_6x_7 + x_7^2.$

$f_7 = x_1x_9 + x_2x_8 + x_3x_9 + x_5x_9 + x_6x_8 +$
$x_1x_2 + x_2x_3 + x_3^2 + x_3x_5 + x_4^2$
$+ x_5x_6 + x_5x_7 + x_6^2 + x_6x_7.$

These are the private polynomials.

Now we do the same as we did in the Oil Vinegar examples, where we write these polynomials in bilinear form $f_i = x^T Q_i x$.

This will give us the Q_i matrices

$$\mathbf{Q_1} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{Q_2} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{Q_3} = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{Q_4} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{Q_5} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{Q_6} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{Q_7} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

We need to define random invertible maps $L_1 : k^7 \longrightarrow k^7$ and $L_2 : k^9 \longrightarrow k^9$.

$$\mathbf{L_1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{L_2} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Now we compose each Q_i matrix with L_2. This will mix the variables within each polynomial f_i.

Let

$$x = L_2 z.$$

This means the composition between L_2 and Q_i will be

$$f_i' = z^T Q_i' z = z^T (L_2^T Q_i L_2) z.$$

Then each f_i' will be

f'$_1 = z_1^2 + z_1z_5 + z_1z_8 + z_1z_9 + z_2z_3 + z_2z_4 + z_2z_5 + z_2z_6$
$+ z_2z_7 + z_2z_9 + z_3^2 + z_3z_5 +$
$z_3z_8 + z_3z_9 + z_6z_5 + z_6^2 + z_6z_8 + z_6z_9.$

f'$_2 = z_1z_2 + z_1z_5 + z_1z_8 +$
$z_1z_9 + z_2^2 + z_2z_4 + z_2z_5 + z_2z_7 + z_2z_9 + z_3z_5 +$
$z_3z_8 + z_3z_9 + z_6z_5 + z_6z_8 + z_6z_9.$

f'$_3 = z_1z_3 + z_1z_6 + z_1z_7 + z_1z_8 + z_2^2 + z_2z_4 + z_2z_5$
$+ z_2z_6 + z_2z_7 + z_3^2 + z_3z_4 + +z_4^2 + z_4z_8 + z_5z_3 + z_5z_4$
$+ z_5z_9 + z_6z_3 + z_6z_9 + zz_7z_6 + z_7^2$
$+ z_7z_9 + z_8z_3 + z_8z_6 + z_8^2 + z_9^2.$

f'$_4 = z_1^2 + z_1z_2 + z_1z_3 + z_1z_5 + z_1z_9 + z_2z_3 + z_3^2 + z_3z_7$
$+ z_4z_1 + z_4z_2 + z_4z_7 + z_5z_3 + z_5^2 + z_5z_6 + z_6z_3 + z_6z_7$
$+ z_7z_1 + z_7z_5 + z_7^2 + z_7z_9 + z_8z_1 + z_8z_2 + z_8z_3 + z_8z_4$
$+ z_8z_6 + z_8z_7 + z_8z_9 + z_9z_3 + z_9z_5 + z_9z_6.$

f'$_5 = z_1z_3 + z_1z_6 + z_1z_8 + z_2z_6 + z_2z_7 + z_2z_8 + z_2z_9$
$+ z_3z_5 + z_4z_3 + z_4z_5 + z_4z_6 + z_5^2 + z_5z_8 + z_6z_5 + z_8z_7$
$+ z_8^2 + z_9z_5 + z_9z_7 + z_9z_8.$

f'$_6 = z_1^2 + z_1z_6 + z_1z_9 + z_2z_1 + z_2^2 + z_2z_3 + z_2z_9 + z_3^2$
$+ z_3z_6 + z_3z_8 + z_3z_9 + z_4z_2 + z_5z_3 + z_5z_4 + z_5^2 +$
$z_5z_7 + z_5z_8 + z_5z_9 + z_6z_2 + z_6^2 + z_6z_7 + z_7z_1 + z_7z_1 +$
$z_7z_2 + z_7z_8 + z_7z_9 + z_8^2 + z_9z_8.$

$$\begin{aligned}
\text{f'}_7 &= z_1z_2 + z_1z_9 + z_2^2 + z_2z_3 + z_2z_9 + z_3z_4 + z_4z_1 \\
&+ z_4z_2 + z_4^2 + z_4z_8 + z_5z_8 + z_6z_1 + z_6z_3 + z_6z_5 + z_6^2 + z_6z_7 \\
&+ z_6z_9 + z_7z_4 + z_7z_8 + z_7z_9 + z_8z_1 + z_8z_2 + z_8z_5 + z_9z_4 + z_9^2.
\end{aligned}$$

Now we compose the polynomials f_i' with L_1 and we get the following polynomials, which we call $\overline{f}_i$ for $i = 1, ..., n - v_1$.

$$\begin{aligned}
\overline{f}_1 &= z_1^2 + z_1z_3 + z_1z_5 + z_1z_6 + z_1z_9 + z_2z_3 + z_2z_4 \\
&+ z_2z_5 + z_2z_8 + z_3^2 + z_3z_8 + z_3z_9 + z_4z_3 + z_4z_5 + \\
&z_4z_6 + z_5^2 + z_5z_8 + z_6^2 + z_6z_8
\end{aligned}$$

$$\begin{aligned}
\overline{f}_2 &= z_1z_2 + z_1z_5 + z_1z_7 + z_1z_8 + z_1z_9 + z_2z_7 + \\
&z_2z_8 + z_3^2 + z_3z_9 + z_4^2 + z_4z_6 + z_4z_8 + z_5z_3 + z_5^2 + \\
&z_5z_8 + z_6z_3 + z_7z_6 + z_7^2 + z_8z_7 + z_9z_8 + z_9^2.
\end{aligned}$$

$$\begin{aligned}
\overline{f}_3 &= z_1^2 + z_1z_3 + z_1z_5 + z_1z_6 + z_1z_7 + z_2^2 \\
&+ z_2z_3 + z_2z_9 + z_3z_4 + z_3z_8 + z_3z_9 + z_5z_4 + z_5z_9 + z_6z_3 \\
&+ z_6z_5 + z_6^2 + z_7z_6 + z_7^2 + z_7z_8 + z_8z_3 + z_8^2 + z_9^2.
\end{aligned}$$

$$\begin{aligned}
\overline{f}_4 &= z_1^2 + z_1z_2 + z_1z_3 + z_1z_5 + z_1z_9 + z_2z_3 + z_3^2 + z_3z_7 \\
&+ z_4z_1 + z_4z_2 + z_4z_7 + z_5z_3 + z_5^2 + z_5z_6 + z_6z_3 + z_6z_7 \\
&+ z_7z_1 + z_7z_5 + z_7^2 + z_7z_9 + z_8z_1 + z_8z_2 + z_8z_3 + z_8z_4 \\
&+ z_8z_6 + z_8z_7 + z_8z_9 + z_9z_3 + z_9z_5 + z_9z_6.
\end{aligned}$$

$$\begin{aligned}
\overline{f}_5 &= z_1^2 + z_1z_6 + z_2^2 + z_2z_3 + z_2z_5 + z_2z_6 + z_3z_7 + \\
&z_4z_1 + z_4z_3 + z_4z_5 + z_4z_6 + z_4z_7 + z_5z_6 + z_5z_8 + z_6z_3 + \\
&z_6z_7 + z_7z_1 + z_7z_5 + z_7^2 + z_8z_1 + z_8z_4 + z_8^2.
\end{aligned}$$

$$\begin{aligned}
\overline{f}_6 &= z_1z_2 + z_1z_3 + z_1z_8 + z_1z_9 + z_3z_6 + z_3z_8 \\
&+ z_3z_9 + z_5z_4 + z_5z_7 + z_6z_5 + z_6z_7 + z_7z_1.
\end{aligned}$$

$$\begin{aligned}
\overline{f}_7 &= z_1^2 + z_1z_2 + z_1z_3 + z_1z_5 + z_1z_7 + z_3z_8 \\
&+ z_3z_9 + z_4z_1 + z_4z_2 + z_5z_4 + z_5z_9 + z_6z_9 + z_7z_4 + z_7^2 \\
&+ z_7z_8 + z_8z_1 + z_8z_2 + z_8^2 + z_9z_4.
\end{aligned}$$

These are the public polynomials.

We want to sign the hashed message $m = (1, 1, 0, 0, 1, 0, 1)$. First, we need to find a signature σ. From the construction, we know how this works.

Start by finding

$$m' = L_1^{-1}(m) = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}.$$

Solving this, gives us $m' = (0, 1, 0, 1, 1, 0, 1)$.

Now we need to find the values of the variables $x_3, ..., x_9$ for some random values of the Vinegar variables x_1, x_2.

Let us have $x_1 = 0$ and $x_2 = 1$. Substituting these values into f_1 and f_2 results to

$$x_3 + x_4 + 1 = 0,$$
$$x_3 + x_4 = 1.$$

Now, this system has infinite amount of solutions, however since we are working in $GF(2)$ there are only two solutions, that is, $x_3 = 0, x_4 = 1$ or $x_3 = 1, x_4 = 0$. We will use the first one.

This means we have the following

$$(x_1, x_2, x_3, x_4) = (0, 1, 0, 1).$$

These will be used as values for the Vinegar variables in the next layer. Substituting these values in f_3, f_4, f_5 yields

$$x_5 + x_6 + x_7 + x_5 + x_7 + 1 + 1 = 0,$$
$$x_5 + x_7 + x_7 + 1 = 0,$$
$$x_6 + x_5 + x_7 + 1 + 1 = 1.$$

Simplifying, and solving this system gives $x_5 = 1, x_6 = 0, x_7 = 1$.

This process will be done once more for the third layer. Substituting the values of $x_1, ..., x_7$ in f_6 and f_7 gives

$$x_8 + x_9 + x_8 + x_8 + x_9 + x_8 + x_9 + 1 + 1 + 1 + 1 + 1 = 0,$$

$$x_8 + x_9 + 1 + 1 = 1.$$

Simplifying and solving this, gives $x_8 = 0, x_9 = 1$.

Hence, we are left with $X^* = (0, 1, 0, 1, 1, 0, 1, 0, 1)$.

The last step for finding the signature σ is to multiply the inverse of L_2 with X^*.

$$\sigma = L_2^{-1}(X^*) = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}.$$

This gives us the signature

$$\sigma = (0, 1, 1, 0, 1, 0, 0, 0, 0).$$

Now, we need to verify the signature, which is done by checking if

$$\overline{F}(\sigma) = m$$

or,

$$\overline{F}(0,1,1,0,1,0,0,0,0) = (1,1,0,0,1,0,1).$$

Remark: This system was mainly done by hand; hence it is important that the reader doesn't expect the public polynomials to be entirely correct. There are a few miscalculations (this is mostly from simplifying the polynomials after composing with L_2 and/or L_1). That is also why we cannot verify the signature. This example intends to show the procedure of the signature scheme; unfortunately, it was with a few miscalculations.

5.3 MinRank attack

In this section we will briefly look at the MinRank attack.

The goal of this attack is to recover (or parts) of the transformations L_1 and L_2. This is done by looking at the rank of the linear combinations of the matrices that corresponds to the public polynomials.

Regarding Rainbow, we want to use the MinRank attack to find a matrix which is a linear combination of the matrices that corresponds to $\overline{f}_i$ for $i = v_1 + 1 \leq i \leq n$, of low rank, which is v_2.

Such a matrix corresponds to a linear combination of the o_1 matrices $L_2^T Q_i L_2, (i \in O_1)$, which are the private polynomials in the first layer of Rainbow.

Let us introduce the MinRank problem.

Given a set of m matrices $M_1, ..., M_m$ of size $n \times n$. We want to achieve a linear combination

$$\mathcal{H} = \sum_{i=1}^{m} \alpha_i M_i.$$

Whose rank is less or equal than r.

For Rainbow, the matrices we are looking at are $\overline{Q}_i's$ that corresponds to $\overline{f}_{v_1+1}, ..., \overline{f}_n$ with rank $r = v_2$.

Now lets look at how we find a low rank matrix, which is a linear combination of the matrices that corresponds to $\overline{f}_{v_1+1}, ..., \overline{f}_n$.

Start by finding a vector $\alpha \in k^m$ and find $\overline{\overline{F}} = \sum_{i=v_1+1}^{n} \alpha_i \overline{f}_i$

Repeat this until a vector α is found such that $Rank(\overline{\overline{F}}) > 1$ and $Rank(\overline{\overline{F}}) < n$

Then randomly choose a vector γ from the kernel of $\overline{\overline{F}}$ then see if,

$Rank(M) \leq v_2$ for $M = \sum_{i=v_1+1}^{n} \gamma_i \overline{f}_i$.

If its not, try with another γ.

By finding, o_1 linearly independent low rank linear combinations of the matrices, $\overline{f}_1, ..., f_n$, we can extract the first Rainbow layer. More layers can be extracted with similar technique. After separating all the Rainbow layers, the attacker can generate signatures the same way as a legitimate user.

Chapter 6

References

[1]https://link.springer.com/content/pdf/10.1007%2F978-0-387-36946-4.pdf

[2] https://link.springer.com/content/pdf/10.1007/3-540-48910-X.pdf

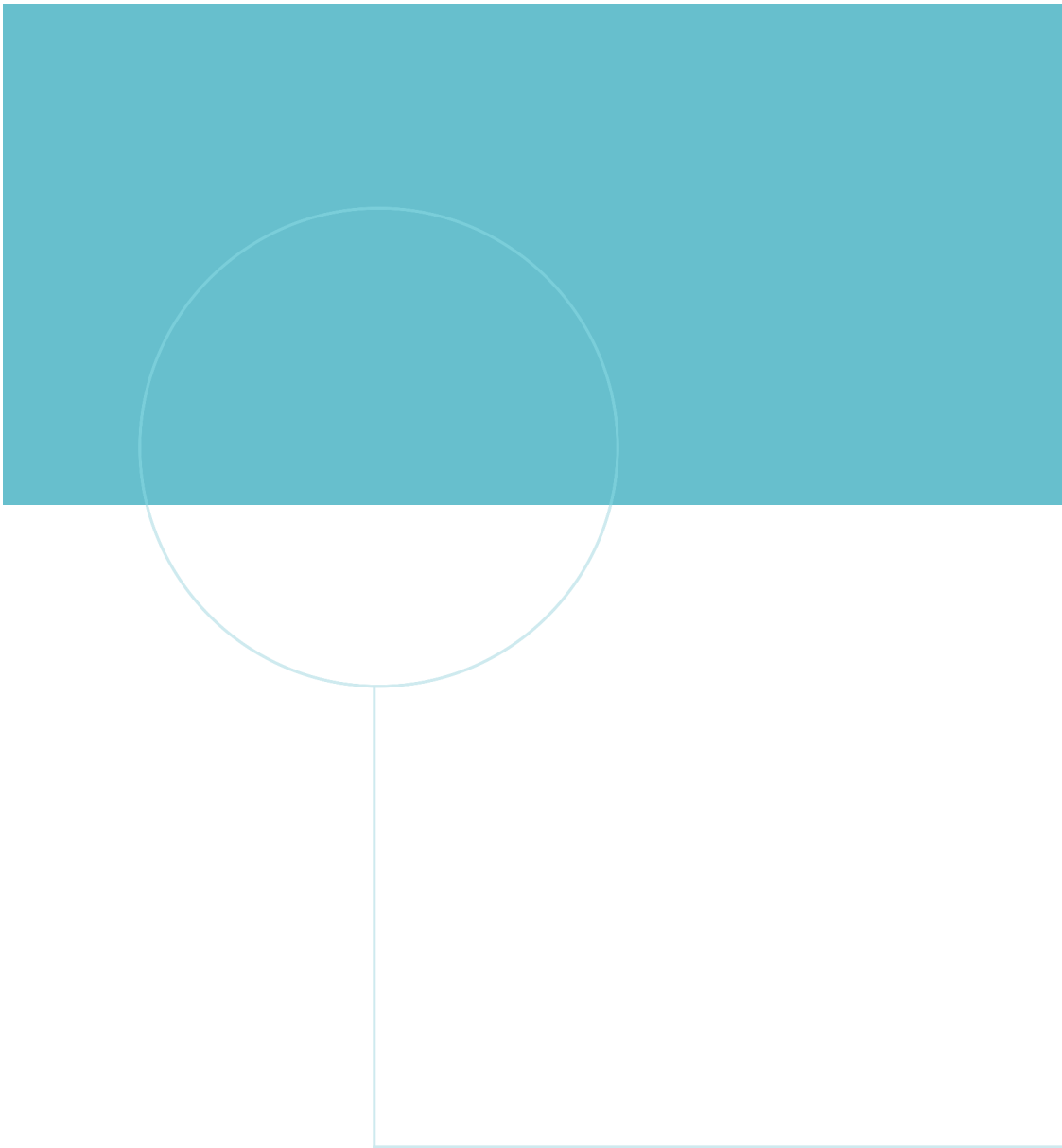
[3] https://wiki.math.ntnu.no/_media/tma4160/pke.pdf

[4] https://wiki.math.ntnu.no/_media/tma4160/signatures.pdf

[5] https://link.springer.com/content/pdf/10.1007/b137093.pdf

[6] https://link.springer.com/content/pdf/10.1007/978-1-0716-0987-3.pdf

[7] https://zaguan.unizar.es/record/87388/files/TAZ-TFG-2019-3124.pdf



NTNU

Kunnskap for en bedre verden