Multivariate public key cryptography with polynomial composition

Emile HAUTEFEUILLE
emile.hautefeuille@polytechnique.edu
École Polytechnique, France

2021

Abstract

This paper presents a new public key cryptography scheme using multivariate polynomials over a finite field. Each multivariate polynomial from the public key is obtained by secretly and repeatedly composing affine transformations with series of quadratic polynomials (in a single variable). The main drawback of this scheme is the length of the public key.

1 Introduction

Since the publication of the Diffie–Hellman key exchange [1] and the RSA cryptosystem [2], public key cryptography has been a major field of cryptography with thousands of applications.

A new threat has appeared with the emergence of quantum computing, leading to the development of post-quantum cryptography, aiming to replace current standards (RSA and Elliptic-curve protocols). Many directions have been explored: lattice-based cryptography, code-based cryptography, supersingular isogeny cryptography, multivariate cryptography...

The scheme presented in this article is part of multivariate cryptography: it is based on the supposedly hard computational problem of finding the roots of a system of multivariate polynomials over a finite field. Famous schemes at the origin of multivariate cryptography are due to Jacques Patarin, including Hidden Field Equations [3] and Unbalanced Oil and Vinegar (UOV) [4] (with Louis Goubin). Most of the multivariate public key schemes, like UOV, are based on the same idea: the secret key consists of two affine transformations and quadratic multivariate polynomials with a trapdoor, hidden by composing both types of functions. The resulting public key is a set of quadratic multivariate polynomials.

This paper proposes a new public key scheme using a different approach. This time, the multivariate polynomials from the public key have a degree greater than 2. This means that the public keys generated by this protocol are quite long. As a consequence, this new scheme does not claim to overpass the best known protocols in multivariate cryptography.

In this protocol, the public multivariate polynomials are generated by iterating the following process:

- Apply an affine transformation.
- Apply a quadratic polynomial (in a single variable) to each output of the affine transformation.

A python implementation of the scheme is available here: github.com/mi10e3/pkc-quadratic-composition

2 Preliminaries

2.1 Public-key cryptography

A public key cryptography protocol generates two keys: a secret key and a public key. The secret key must be kept private, while the public key can be shared publicly. Everyone can use the public key to encrypt a message. Only the owner of the secret key must be able to reverse the operation and decrypt the message.

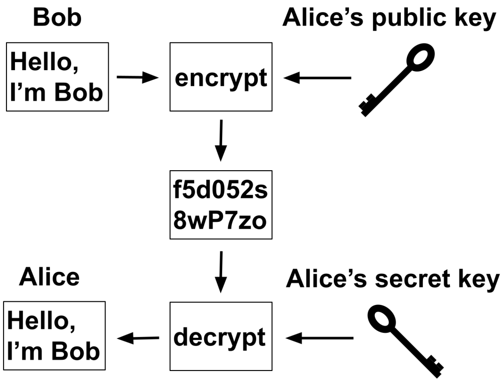

Figure 1: Example of communication using public key cryptography: only Alice can decrypt the message sent by Bob thanks to her secret key.

2.2 Context and notations

For what follows, let p be an odd prime number. We will place ourselves over $\mathbb{F}_p$, the finite field of p elements.

We will use the ring of polynomials in i variables $x_1, ..., x_i$. However, this scheme also involves polynomials in one variable. For the sake of clarity, the number of variables in the polynomials will always be specified.

We will consider that two multivariate polynomials in $\mathbb{F}_p$ are equal when they share the same coefficients (even though two multivariate polynomials, $q_1\colon (\mathbb{F}_p)^i \to \mathbb{F}_p$ and $q_2\colon (\mathbb{F}_p)^i \to \mathbb{F}_p$, may take the same values, $\forall \bar{x} \in (\mathbb{F}_p)^i, q_1(\bar{x}) = q_2(\bar{x})$, without sharing the same coefficients).

If f is a function from X to Y,

- we denote by $f[U]$ the image of a set $U \subseteq X$ under f: $f[U] = \{f(x) : x \in U\}$
- we denote by $f^{-1}[V]$ the preimage of a set $V \subseteq Y$ under f: $f^{-1}[V] = \{x \in X : f(x) \in V\}$

If J is a finite set, we denote by $card(J)$ the number of its elements.

When $a \in \mathbb{F}_p$ is a quadratic residue (modulo p), we denote by $\sqrt{a}$ one of its square roots: $\sqrt{a} = x \in \mathbb{F}_p$ with $x^2 = a$. If $a \neq 0$, there are two possible values for $\sqrt{a}$. We use this symbol when either of the values can be used interchangeably.

2.3 Compute square roots in $\mathbb{F}_p$

Checking if $a \in \mathbb{F}_p$ is a quadratic residue (modulo p) can be done quickly with Euler's criterion:

$$a^{\frac{p-1}{2}} = \begin{cases} 1 \pmod{p} \text{ when } a \text{ is a quadratic residue} \\ -1 \pmod{p} \text{ otherwise} \end{cases}$$

This test requires $O(log(p))$ modular multiplications with fast exponentiation.

The computation of a square root in $\mathbb{F}_p$ is also quick:

- If $p = 3 \pmod 4$, $\sqrt{a} = \pm\, a^{\frac{p+1}{4}}$ (again $O(log(p))$ modular multiplications with fast exponentiation)
- Otherwise $p = 1 \pmod 4$, and the Tonelli–Shanks algorithm [5] will compute a square root of a in $O(log(p) + l^2)$ modular multiplications [6], where l is defined by $p - 1 = 2^l h$, with h odd. For general values of p, l is negligible compared to $log(p)$, and the Tonelli–Shanks algorithm requires also $O(log(p))$ modular multiplications.

2.4 Solve quadratic equations in $\mathbb{F}_p$

To solve a quadratic equation in $\mathbb{F}_p$: $ax^2 + bx + c = 0$, we do as if we were in $\mathbb{R}$: we compute the discriminant: $\Delta = b^2 - 4ac$. If Δ is a quadratic residue, we compute the 2 solutions with the well known formula: $x = (-b \pm \sqrt{\Delta})/2a$, otherwise the equation has no solution. The computational complexity lies in the square root, which involves $O(log(p))$ modular multiplications (as mentioned in section 2.3).

With y in $\mathbb{F}_p$, the equation (with unknown x) $ax^2 + bx + c = y$ has:

- 1 solution when $\Delta = 0 \Leftrightarrow b^2 - 4a(c - y) = 0 \Leftrightarrow y = c - \frac{b^2}{4a}$: it happens only for one value of y
- 2 solutions for n different values of y
- 0 solutions for $p - n - 1$ different values of y

As there are p different values for x, we have $1 + 2n + 0 \times (p - n - 1) = p$ which leads to $n = \frac{p-1}{2}$.

As a result, a random quadratic equation (with a single unknown) in $\mathbb{F}_p$ has 1 solution with probability $1/p$, 2 solutions with probability $(p-1)/(2p)$ and no solutions with probability $(p-1)/(2p)$. When p is big, with a reasonable approximation, a random quadratic equation (with a single unknown) in $\mathbb{F}_p$ has no solutions with probability 1/2 and 2 solutions with probability 1/2.

2.5 Invert affine transformations in $\mathbb{F}_p$

Let T be an affine transformation from $(\mathbb{F}_p)^u$ to $(\mathbb{F}_p)^v$ (with $u \geq 1$ and $v \geq 1$), that is to say a function of the form:

$$\begin{aligned} T\colon (\mathbb{F}_p)^u &\to (\mathbb{F}_p)^v \\ (x_1, ..., x_u) &\mapsto (\lambda^{i,0} + \sum_{j=1}^{u} \lambda^{i,j} x_j)_{1 \leq i \leq v} \end{aligned}$$

Let r be the rank of the matrix $(\lambda^{i,j})_{1 \leq i \leq v\,,\, 1 \leq j \leq u}$. The image of T is an r-dimensional affine subspace of $(\mathbb{F}_p)^v$, containing p^r elements. For $\bar{y} = (y_1, .., y_v) \in T[(\mathbb{F}_p)^u]$, $T^{-1}[\{\bar{y}\}]$ is a $(u - r)$-dimensional affine subspace of $(\mathbb{F}_p)^u$, containing p^{u-r} elements (according to the rank–nullity theorem). As a result, if $\bar{Y}$ is a random variable following the discrete uniform distribution over $(\mathbb{F}_p)^v$:

$$T^{-1}[\{\bar{Y}\}] = \begin{cases} \text{a set of } p^{u-r} \text{ elements of } (\mathbb{F}_p)^u & \text{with probability } p^r/p^v \\ \emptyset & \text{with probability } 1 - p^r/p^v \end{cases}$$

In this protocol, the domain of an affine transformation is always smaller than its codomain: $u \leq v$. Moreover, we only consider full rank affine transformations: $r = min(u, v) = u$. As a result, in this protocol, when T is an affine transformation from $(\mathbb{F}_p)^u$ to $(\mathbb{F}_p)^v$, and $\bar{Y}$ is a random variable following the discrete uniform distribution over $(\mathbb{F}_p)^v$:

$$T^{-1}[\{\bar{Y}\}] = \begin{cases} \text{a singleton } \{\tau\} \text{ with } \tau \in (\mathbb{F}_p)^u & \text{with probability } p^u/p^v \\ \emptyset & \text{with probability } 1 - p^u/p^v \end{cases}$$

In practice, computing $T^{-1}[\{\bar{Y}\}]$ requires $O(u^2 v)$ modular multiplications with Gaussian elimination.

3 The scheme

3.1 Parameters of the scheme

This protocol is defined by the following parameters:

- p, an odd prime number
- $m \geq 3$
- $(a_1, a_2, ..., a_m) \in \mathbb{N}^m$ with $2 \leq a_1 \leq a_2 \leq ... \leq a_m$

All calculations are made modulo p.

The message we want to encrypt will be represented by a_1 elements of $\mathbb{F}_p$: $(x_1, ..., x_{a_1})$. The encrypted message will be represented by a_m elements of $\mathbb{F}_p$: $(y_1, ..., y_{a_m})$. Encryption can be seen as a flow: the initial a_1-variable message will successively pass through layers of $a_2, a_3, ...$ and finally a_m variables.

3.2 Generate the secret key

The secret key is obtained by repeatedly choosing an affine transformation and a series of univariate quadratic polynomials. There is one univariate quadratic polynomial for each output of the affine transformations.

The secret key consists of $T_1, ..., T_{m-1}$ and $Q_1, ..., Q_{m-2}$, defined as follows:

- For $k \in \{1, ..., m-1\}$, T_k is an affine transformation from $(\mathbb{F}_p)^{a_k}$ to $(\mathbb{F}_p)^{a_{k+1}}$

$$\begin{aligned} T_k \colon (\mathbb{F}_p)^{a_k} &\to (\mathbb{F}_p)^{a_{k+1}} \\ (x_1, ..., x_{a_k}) &\mapsto (\lambda_k^{i,0} + \sum_{j=1}^{a_k} \lambda_k^{i,j} x_j)_{1 \leq i \leq a_{k+1}} \end{aligned}$$

- For $k \in \{1, ..., m-2\}$, Q_k applies a different quadratic polynomial (in one variable) to each of its a_{k+1} inputs.

$$\begin{aligned} Q_k \colon (\mathbb{F}_p)^{a_{k+1}} &\to (\mathbb{F}_p)^{a_{k+1}} \\ (x_1, ..., x_{a_{k+1}}) &\mapsto (\alpha_k^i {x_i}^2 + \beta_k^i x_i + \gamma_k^i)_{1 \leq i \leq a_{k+1}} \end{aligned}$$

Thus, the secret key is created by randomly choosing (in $\mathbb{F}_p$) the following values:

- $\lambda_k^{i,j}$ for $k \in \{1, ..., m-1\}, i \in \{1, ..., a_{k+1}\}$ and $j \in \{0, 1, ..., a_k\}$
- $\alpha_k^i, \beta_k^i, \gamma_k^i$ for $k \in \{1, ..., m-2\}$ and $i \in \{1, ..., a_{k+1}\}$

Finally, we require the affine transformations to have full rank.

3.3 Generate the public key

The public key is obtained by alternatively composing the affine transformations $(T_k)_{1 \leq k \leq m-1}$ with the series of univariate quadratic polynomials $(Q_k)_{1 \leq k \leq m-2}$:

$$T_{m-1} \circ Q_{m-2} \circ T_{m-2} \circ Q_{m-3} \circ ... \circ Q_2 \circ T_2 \circ Q_1 \circ T_1(X_1, ..., X_{a_1})$$

The resulting function has a_m outputs. Each output M_i $(1 \leq i \leq a_m)$ is a multivariate polynomial with variables $(X_1, ..., X_{a_1})$ over $\mathbb{F}_p$. The degree of each multivariate polynomial is 2^{m-2}.

The multivariate polynomials of the public key are given in their expanded form:

$$(M_i(X_1, ..., X_{a_1}))_{1 \leq i \leq a_m} = \Big(\sum_{u_1 + ... + u_{a_1} \leq 2^{m-2}} \eta_i^{(u_1, ..., u_{a_1})} \times X_1^{u_1} \times ... \times X_{a_1}^{u_{a_1}} \Big)_{1 \leq i \leq a_m}$$

3.4 Encrypt a message

Let's consider a message $(x_1, ..., x_{a_1}) \in (\mathbb{F}_p)^{a_1}$ and a public key $(M_1, ..., M_{a_m})$.

For each $i \in \{1, ..., a_m\}$, we evaluate the multivariate polynomial M_i with the message: $y_i = M_i(x_1, ..., x_{a_1})$

The encrypted message is $(y_1, ..., y_{a_m}) \in (\mathbb{F}_p)^{a_m}$.

3.5 Decrypt a message

Both types of function of the secret key are easily invertible:

- For $k \in \{1, ..., m-1\}$, the affine transformation $T_k : (\mathbb{F}_p)^{a_k} \to (\mathbb{F}_p)^{a_{k+1}}$ is easily invertible with Gaussian elimination. Furthermore, T_k is injective because it has full rank and $a_k \leq a_{k+1}$. This injectivity is important to keep the number of collisions under control (see section 4.4 for details).
- For $k \in \{1, ..., m-2\}$, $Q_k : (\mathbb{F}_p)^{a_{k+1}} \to (\mathbb{F}_p)^{a_{k+1}}$ is also easily invertible:

  For $(y_1, ..., y_{a_{k+1}})$ fixed in $(\mathbb{F}_p)^{a_{k+1}}$, we are looking for $(x_1, ..., x_{a_{k+1}}) \in (\mathbb{F}_p)^{a_{k+1}}$ such that:

  $$Q_k(x_1, ..., x_{a_{k+1}}) = (y_1, ..., y_{a_{k+1}}) \Leftrightarrow \forall i \in \{1, ..., a_{k+1}\},\ \ \alpha_k^i {x_i}^2 + \beta_k^i x_i + \gamma_k^i = y_i$$

  As mentioned in section 2.4, for $i \in \{1, ..., a_{k+1}\}$, we can easily compute Λ_i, the set of solutions of the equation $\alpha_k^i {x_i}^2 + \beta_k^i x_i + \gamma_k^i = y_i$ (with unknown x_i). Λ_i contains $0, 1$ or 2 elements of $\mathbb{F}_p$. The set of solutions of the equation $Q_k(x_1, ..., x_{a_{k+1}}) = (y_1, ..., y_{a_{k+1}})$ (with unknowns $x_1, ..., x_{a_{k+1}}$) is $\Lambda_1 \times \Lambda_2 \times ... \times \Lambda_{a_{k+1}}$. This set is empty if one Λ_i is empty, that is to say if one of the previous quadratic equations has no solutions.

As a consequence, we can decrypt a message by recursively inverting the functions of the secret key: Let's consider an encrypted message $(y_1, ..., y_{a_m}) \in (\mathbb{F}_p)^{a_m}$ and a secret key corresponding to the public key used for encryption: $T_1, ..., T_{m-1}, Q_1, ..., Q_{m-2}$. We are looking for $(x_1, ..., x_{a_1}) \in (\mathbb{F}_p)^{a_1}$ such that:

$$\begin{aligned} T_{m-1} \circ Q_{m-2} \circ T_{m-2} \circ ... \circ Q_1 \circ T_1(x_1, ..., x_{a_1}) \ &= \ (y_1, ..., y_{a_m}) \\ \Leftrightarrow Q_{m-2} \circ T_{m-2} \circ ... \circ Q_1 \circ T_1(x_1, ..., x_{a_1}) \ &\in \ T_{m-1}^{-1}[\{(y_1, ..., y_{a_m})\}] \\ \Leftrightarrow T_{m-2} \circ ... \circ Q_1 \circ T_1(x_1, ..., x_{a_1}) \ &\in \ Q_{m-2}^{-1} \circ T_{m-1}^{-1}[\{(y_1, ..., y_{a_m})\}] \\ &... \\ \Leftrightarrow (x_1, ..., x_{a_1}) \ &\in \ T_1^{-1} \circ Q_1^{-1} \circ ... \circ T_{m-2}^{-1} \circ Q_{m-2}^{-1} \circ T_{m-1}^{-1}[\{(y_1, ..., y_{a_m})\}] \end{aligned}$$

The decryption may find more than one possible message (more than one possible value for $(x_1, ..., x_{a_1})$). Section 4.4 shows that the risk of collision is drastically reduced when $a_1 < a_2$.

4 Practical aspects

4.1 Size of the keys

For $k \in \{1, ..., m-1\}$, the affine transformation T_k has $a_{k+1}(a_k + 1)$ parameters. For $k \in \{1, ..., m-2\}$, the a_{k+1} quadratic polynomials (in one variable) of Q_k contain $3\,a_{k+1}$ parameters. In total, the number of parameters (elements of $\mathbb{F}_p$) that define the secret key is:

$$\sum_{k=1}^{m-1} a_{k+1}(a_k + 1) + \sum_{k=1}^{m-2} 3\,a_{k+1} \ = a_m(a_{m-1} + 1) + \sum_{k=1}^{m-2} a_{k+1}(a_k + 4)$$

The main drawback of this protocol is the length of the public key. Each multivariate polynomial M_i $(1 \leq i \leq a_m)$ from the public has a degree of 2^{m-2}, with a_1 input variables, resulting in $\binom{2^{m-2}+a_1}{a_1}$ monomials. As a result, the public key consists of $a_m \times \binom{2^{m-2}+a_1}{a_1}$ elements of $\mathbb{F}_p$.

4.2 Complexity of encryption

The encryption of $(x_1, ..., x_{a_1}) \in (\mathbb{F}_p)^{a_1}$ requires $O(a_m \times \binom{2^{m-2}+a_1}{a_1})$ modular multiplications and additions.

Proof. The encryption of $(x_1, ..., x_{a_1})$ consists of the evaluation of a_m multivariate polynomials over $\mathbb{F}_p$. A classical approach for a fast evaluation is:

- Evaluate first all the monomials without considering the coefficients of the polynomial (compute $X^3Y^2Z^5$, not $134.X^3Y^2Z^5$).
- To do so, start from the monomials of low degree: degrees 0 and 1 require no multiplications.
- Compute the monomials (without coefficient) of degree $k+1$ from those of degree k (stored in memory), as it only involves one multiplication: to compute $X^3Y^2Z^5$, multiply X by $X^2Y^2Z^5$ (already calculated).

This process requires $\binom{2^{m-2}+a_1}{a_1} - a_1 - 1$ multiplications (each polynomial has a_1 variables and a degree of 2^{m-2}, resulting in $\binom{2^{m-2}+a_1}{a_1}$ monomials). If we now consider the polynomial's coefficients, we obtain a total of $2 \times \binom{2^{m-2}+a_1}{a_1} - a_1 - 2$ multiplications.

Finally, $\binom{2^{m-2}+a_1}{a_1} - 1$ additions are needed to get the result of the evaluation.

4.3 Complexity of decryption

In a similar way to encryption, the number of additions needed during decryption is close to the number of multiplications. For simplicity's sake, the complexity estimation will be limited to multiplications, which are the most expensive. The proof of the following result is quite technical and is let at the end of this article (section 8.2):

The total number of modular multiplications required to decrypt a message is on average:

$$O(a_{m-1}^2 a_m + \sum_{k=1}^{m-2} \left(log(p)\, a_{k+1} + 2^{a_{k+1}} a_k^2 a_{k+1} + (log(p) + a_k^2 a_{k+1}) \times \sum_{v=k+2}^{m-1} p^{a_{k+1}-a_v}(2^{a_v} - 1) \right))$$

4.4 Implementation suggestions

Because of the length of the public key, we cannot afford too many consecutive compositions: a reasonable value for m is probably between 3 and 10.

a_1 should share a similar order of magnitude, for the same reason.

Choosing a big value for p seems natural to compensate for the small number of variables representing the message (simply to prevent brute force attacks). Note that this protocol can work very well with $p \approx 10^{1000}$, or even more.

With the values suggested above, the probability of message collision can be drastically reduced by using $a_1 < a_2$. With a large value for p, choosing $a_2 = a_1 + 1$ is enough to guarantee a probability of collision almost equal to zero.

Proof. The equation ($\star$) (Section 8.2, Part 2), applied to $k = 1$, gives the average number of solutions at the end of the decryption: when $(y_1, ..., y_{a_m})$ is a random encrypted message, the average number of messages $(x_1, ..., x_{a_1})$ whose encryption gives $(y_1, ..., y_{a_m})$ is:

$$1 + \sum_{v=2}^{m-1} p^{a_1-a_v}(2^{a_v} - 1)$$

This value must be very close to 1 to avoid collisions. This requires $a_1 < a_2$, as shown below:

- This is a necessary condition. Because otherwise, we would have $a_1 = a_2$ (reminder: we always have $a_1 \leq a_2 \leq ... \leq a_m$), leading to: $1 + \sum_{v=2}^{m-1} p^{a_1 - a_v}(2^{a_v} - 1) \geq 1 + p^{a_1 - a_2}(2^{a_2} - 1) = 2^{a_1} \geq 4$.
- This is a sufficient condition. If $a_1 < a_2$, then $1 \leq a_2 - a_1$ and $p^{a_1 - a_v} \leq 1/p$ for all $v \in \{2, ..., m-1\}$. As a result,

$$1 + \sum_{v=2}^{m-1} p^{a_1 - a_v}(2^{a_v} - 1) \leq 1 + \sum_{v=2}^{m-1} (2^{a_v} - 1)/p \leq 1 + (m-2)(2^{a_m} - 1)/p \leq 1 + m2^{a_m}/p$$

In practice, a_m does not exceed a few dozen, as well as for m, while p is reasonably above 10^{100}. As a result, $m2^{a_m}/p \ll 1$, meaning the probability of collision is very low.

5 Security

The security of this protocol lies in the difficulty of finding the roots of the system of multivariate polynomials used to encrypt messages. However, these polynomials are not fully random, as they are obtained by successively composing affine transformations and series of quadratic polynomials (in one variable). The most promising attack is probably to try to recover a secret key from a public key. If we consider the coefficients of the secret keys $((\lambda_k^{i,j}), (\alpha_k^i), (\beta_k^i)$ and (γ_k^i), see section 3.2) as unknowns, and if we develop $T_{m-1} \circ Q_{m-2} \circ ... \circ Q_1 \circ T_1(X_1, ..., X_{a_1})$, we get a_m multivariate polynomials with variables $X_1, ..., X_{a_1}$. Each coefficient of these multivariate polynomials is also a multivariate polynomial with variables $(\lambda_k^{i,j}), (\alpha_k^i), (\beta_k^i)$ and (γ_k^i). If we identify these coefficients with the ones of the public key, we obtain a system of multivariate polynomials. This system contains:

- $a_m \times \binom{2^{m-2}+a_1}{a_1}$ multivariate polynomials, because the public key is composed of a_m multivariate polynomials, each with $\binom{2^{m-2}+a_1}{a_1}$ coefficients (see section 4.1).
- $a_m(a_{m-1} + 1) + \sum_{k=1}^{m-2} a_{k+1}(a_k + 4)$ unknowns, corresponding to the the number of values in $(\lambda_k^{i,j})$, (α_k^i), (β_k^i) and (γ_k^i) (see section 4.1).

An efficient attack exists when the affine transformations (T_k) between the (Q_k) are removed (see section 8.1). Although the use of these intermediate affine transformations makes the secret key more complex, further work needs to be conducted to assess the security of this protocol, to understand how it behaves against some classical attacks (Gröbner basis for instance).

6 Conclusion

A new public key protocol in multivariate cryptography has been introduced. It is based on the ease of solving quadratic equations (in one variable) modulo p. Due to the size of the public key, this scheme is not intended to compete with the best known protocols in multivariate cryptography. The choice of the right parameters in order to guarantee security remains unclear. Finally, it would be interesting to assess to what extent the idea of composition with univariate quadratic polynomials can be used alongside another existing multivariate cryptosystem, to reinforce its security.

7 Acknowledgements

I am grateful to Ali Deniz Özkan (*École Polytechnique*) for his advice.

The protocol presented in this paper is an improvement of a previous version [7] based on the same ideas. The previous version of the protocol was broken by Charles Bouillaguet (*LIP6, Sorbonne University*). A detailed description of his attack is given in section 8.1.

8 Appendix

8.1 An attack showing the importance of the intermediate affine transformations

This attack was proposed by Charles Bouillaguet (*LIP6, Sorbonne University*), on a previous version [7] of this protocol.

In the previous version, we only applied one secret affine transformation, at the beginning of the scheme. By keeping the previous notation for Q and T, the public key was obtained as follows:

$$Q_m \circ Q_{m-1} \circ ... \circ Q_2 \circ Q_1 \circ \mathbf{T}(X_1, ..., X_i)$$

Without the affine transformations between the (Q_k), the secret quadratic polynomials (in one variable) of each line were composed together, independently of the other lines. As a result, each multivariate polynomial of the public key was obtained by secretly composing:

$$(\alpha_m X^2 + \beta_m X + \gamma_m) \circ (\alpha_{m-1} X^2 + \beta_{m-1} X + \gamma_{m-1}) \circ ... \circ (\alpha_1 X^2 + \beta_1 X + \gamma_1) \circ (\lambda_0 + \lambda_1 X_1 + ... + \lambda_i X_i)$$

The attack recovers a secret key from a public key, using the collisions between the keys: each public key was the consequence of numerous secret keys, allowing to restrict considerably the search for the parameters of the secret key.

At the beginning of the attack, we are looking for values of (α), (β), (γ) and (λ) such that the development of the secret compositions corresponds to a given multivariate polynomial from the public key:

$$\begin{gathered}\sum_{u_1+...+u_i \,\leq\, 2^m} \overbrace{\eta^{(u_1,...,u_i)}}^{\text{publicly available}} \times X_1^{u_1} \times ... \times X_i^{u_i} \;= \\ (\alpha_m X^2 + \beta_m X + \gamma_m) \circ ... \circ (\alpha_1 X^2 + \beta_1 X + \gamma_1) \circ (\lambda_0 + \lambda_1 X_1 + ... + \lambda_i X_i)\end{gathered} \tag{1}$$

Part 1: $\lambda_0, \gamma_1, \gamma_2, ..., \gamma_m$ can be fixed

If q is a given value in $\mathbb{F}_p$, every quadratic polynomial (in one variable) can be expressed in the form:

$$aX^2 + bX + c = a(X - q)^2 + b'(X - q) + c'$$

(By taking $b' = b + 2aq$ and $c' = c + aq^2 + bq$).

Let's now consider the first composition in the secret key:

$$\begin{aligned}&(\alpha_1 X^2 + \beta_1 X + \gamma_1) \circ (\lambda_0 + \lambda_1 X_1 + ... + \lambda_i X_i)\\ =\;& (\alpha_1 (X - \lambda_0)^2 + \beta'_1 (X - \lambda_0) + \gamma'_1) \circ (\lambda_0 + \lambda_1 X_1 + ... + \lambda_i X_i)\\ =\;& (\alpha_1 X^2 + \beta'_1 X + \gamma'_1) \circ (\lambda_1 X_1 + ... + \lambda_i X_i)\end{aligned}$$

In the secret key, replacing β_1 by β'_1, γ_1 by γ'_1 and λ_0 by 0 will not change the resulting public multivariate polynomial.

Without loss of generality, we can thus assume that $\lambda_0 = 0$.

We can apply this trick for the next composition: with appropriate values for β_2 and γ_2, we can assume that $\gamma_1 = 0$.

Iterating this process shows that, without loss of generality, we can assume that $\lambda_0, \gamma_1, \gamma_2, ..., \gamma_{m-1}$ are all equal to 0.

Finally, γ_m can be easily deduced because it becomes the only value constituting the constant coefficient in the multivariate polynomial in the second line of (1), leading to $\gamma_m = \eta^{(0,\cdots,0)}$, publicly available.

Part 2: $\alpha_1, ..., \alpha_m$ can be also be fixed

Let's u be a fixed quadratic non-residue (use Euler's criterion (section 2.3) to determine one).

We consider the coefficient of $X_1^{2^m}$ in both multivariate polynomials of (1):

$$\eta^{(2^m,0,\ldots,0)} = \alpha_m \times \alpha_{m-1}^2 \times \alpha_{m-2}^4 \times ... \times \alpha_1^{2^{m-1}} \times \lambda_1^{2^m} \tag{2}$$

- Suppose first that $\eta^{(2^m,0,\ldots,0)}$ is a quadratic residue.

  Then $\alpha_m = \left(\sqrt{\eta^{(2^m,0,\ldots,0)}}/(\alpha_{m-1} \times \alpha_{m-2}^2 \times ... \times \alpha_1^{2^{m-2}} \times \lambda_1^{2^{m-1}})\right)^2$ is also a quadratic residue.

  As a consequence:

$$\begin{aligned}&(\alpha_m X^2 + \beta_m X + \gamma_m) \circ (\alpha_{m-1} X^2 + \beta_{m-1} X + \gamma_{m-1})\\ &= (X^2 + \frac{\beta_m}{\sqrt{\alpha_m}} X + \gamma_m) \circ (\sqrt{\alpha_m}\alpha_{m-1} X^2 + \sqrt{\alpha_m}\beta_{m-1} X + \sqrt{\alpha_m}\gamma_{m-1})\end{aligned}$$

- Otherwise, $\eta^{(2^m,0,\ldots,0)}$ is not a quadratic residue.

  Then $u \times \alpha_m = \left(\sqrt{u \times \eta^{(2^m,0,\ldots,0)}}/(\alpha_{m-1} \times \alpha_{m-2}^2 \times ... \times \alpha_1^{2^{m-2}} \times \lambda_1^{2^{m-1}})\right)^2$ is a quadratic residue.

  (The product of two quadratic non-residues, $u \times \eta^{(2^m,0,\ldots,0)}$, is always a quadratic residue).

  As a consequence:

$$\begin{aligned}&(\alpha_m X^2 + \beta_m X + \gamma_m) \circ (\alpha_{m-1} X^2 + \beta_{m-1} X + \gamma_{m-1})\\ &= (u^{-1} X^2 + \frac{\beta_m}{\sqrt{u \times \alpha_m}} X + \gamma_m) \circ (\sqrt{u \times \alpha_m}\alpha_{m-1} X^2 + \sqrt{u \times \alpha_m}\beta_{m-1} X + \sqrt{u \times \alpha_m}\gamma_{m-1})\end{aligned}$$

In both cases, $(\alpha_m X^2 + \beta_m X + \gamma_m) \circ (\alpha_{m-1} X^2 + \beta_{m-1} X + \gamma_{m-1})$ admits an equivalent expression where $\alpha_m \in \{1, u\}$. Without loss of generality, we can thus assume that $\alpha_m \in \{1, u\}$. We know which of the two values to choose because we can easily determine whether $\eta^{(2^m,0,\ldots,0)}$, publicly available, is a quadratic residue or not with Euler's criterion (see section 2.3).

Using (2) again, we get: $\sqrt{\eta^{(2^m,0,\ldots,0)}/\alpha_m} = \alpha_{m-1} \times \alpha_{m-2}^2 \times \alpha_{m-3}^4 \times ... \times \alpha_1^{2^{m-2}} \times \lambda_1^{2^{m-1}}$

- Suppose first that $\sqrt{\eta^{(2^m,0,\ldots,0)}/\alpha_m}$, is a quadratic residue.

  Then α_{m-1} is also a quadratic residue, with the same reasoning as above.

  As a consequence:

$$\begin{aligned}&(\alpha_{m-1} X^2 + \beta_{m-1} X + \gamma_{m-1}) \circ (\alpha_{m-2} X^2 + \beta_{m-2} X + \gamma_{m-2})\\ &= (X^2 + \frac{\beta_{m-1}}{\sqrt{\alpha_{m-1}}} X + \gamma_{m-1}) \circ (\sqrt{\alpha_{m-1}}\alpha_{m-2} X^2 + \sqrt{\alpha_{m-1}}\beta_{m-2} X + \sqrt{\alpha_{m-1}}\gamma_{m-2})\end{aligned}$$

- Otherwise, we treat the case where $\sqrt{\eta^{(2^m,0,\ldots,0)}/\alpha_m}$ is not a quadratic residue in the same way we did above, by artificially multiplying by $u^{-1} \times u$.

In both cases, $(\alpha_{m-1} X^2 + \beta_{m-1} X + \gamma_{m-1}) \circ (\alpha_{m-2} X^2 + \beta_{m-2} X + \gamma_{m-2})$ admits an equivalent expression where $\alpha_{m-1} \in \{1, u\}$. Without loss of generality, we can thus assume that $\alpha_{m-1} \in \{1, u\}$. We know which of the two values to choose because we have access to $\sqrt{\eta^{(2^m,0,\ldots,0)}/\alpha_m}$ (α_m was revealed above) and we can thus determine whether it is a quadratic residue or not.

By iterating this process, without loss of generality, we can thus assume that $\alpha_m, \alpha_{m-1}, ..., \alpha_2$ are known values in $\{1, u\}$.

At the end of the iteration, we can fix α_1 with the same trick:

As we did above, we can now deduce whether α_1 is a quadratic residue or not, because we have access to:

$$\sqrt{...\sqrt{\sqrt{\sqrt{\eta^{(2^m,0,...,0)}/\alpha_m}/\alpha_{m-1}}/...}\ /\alpha_2} = \alpha_1 \times \lambda_1^2$$

- If α_1 is a quadratic residue:

$$\begin{aligned}&(\alpha_1 X^2 + \beta_1 X + \gamma_1) \circ (\lambda_0 + \lambda_1 X_1 + ... + \lambda_i X_i)\\ =\ &(X^2 + \frac{\beta_1}{\sqrt{\alpha_1}} X + \gamma_1) \circ (\sqrt{\alpha_1}\lambda_0 + \sqrt{\alpha_1}\lambda_1 X_1 + ... + \sqrt{\alpha_1}\lambda_i X_i)\end{aligned}$$

- Otherwise, we treat the case where α_1 is not a quadratic residue in the same way we did above, by artificially multiplying by $u^{-1} \times u$.

To conclude, without loss of generality, we can assume that all $\alpha_1, ..., \alpha_m$ are known values in $\{1, u\}$.

Part 3: Deducing $\lambda_1, ..., \lambda_i$

Let $b \in \{1, ..., i\}$.

We consider the coefficient of $X_b^{2^m}$ in both multivariate polynomials of (1):

$$\begin{aligned}&\eta^{\overbrace{(0,...,0,2^m,0...,0)}^{2^m\text{is in position b}}} = \alpha_m \times \alpha_{m-1}^2 \times \alpha_{m-2}^4 \times ... \times \alpha_1^{2^{m-1}} \times \lambda_b^{2^m}\\ \Leftrightarrow\ &\lambda_b^{2^m} = \eta^{(0,...,0,2^m,0...,0)}/(\alpha_m \times \alpha_{m-1}^2 \times \alpha_{m-2}^4 \times ... \times \alpha_1^{2^{m-1}})\end{aligned}$$

$\eta^{(0,...,0,2^m,0...,0)}$ is a known value from the public key. As explained in part 2, we can consider that $\alpha_1, ..., \alpha_m$ are known values in $\{1, u\}$. As a consequence, the 2^mth power of λ_b is known. On average, this leaves $m + 1$ possibilities for λ_b.

As a result, we can restrict $(\lambda_1, ..., \lambda_i)$ to a set of about $(m + 1)^i$ possibilities in $(\mathbb{F}_p)^i$. This is very reasonable in practice because both m and i are small values (probably around 5, otherwise the public key is too long).

Part 4: Deducing $\beta_1, ..., \beta_m$

With Part 1, we know $\lambda_0, \gamma_1, \gamma_2, ..., \gamma_m$. With Part 2, we know $\alpha_1, ..., \alpha_m$. Part 3 reduces $\lambda_1, ..., \lambda_i$ to a small set of possibilities. We iterate over these possibilities until we find correct values for $\beta_1, .., \beta_m$:

Given $(\gamma), (\alpha)$ and (λ), we can deduce $\beta_1, .., \beta_m$ as follows:

- For $k \in \{1, ..., m\}$, by considering the coefficient of $X_1^{2^m - 2^{k-1}}$ in (1), we get a multivariate polynomial equation with unknowns $\beta_1, ..., \beta_k$, where β_k is never raised to a power greater than 1. These equations are easily solvable by successively considering $k = 1, k = 2, ..., k = m$.
- Finally, we need to check that the values of $\beta_1, .., \beta_m$ we found match with the public key, that is to say if the the equation (1) is verified.

8.2 Proof of the complexity of decryption

Part 1: Inverting $Q \circ T$

The decryption of a message involves successively inverting functions of the form $\Gamma = Q \circ T$, with:

- $u \leq v$ two natural integers ($u \geq 1$).
- $T\colon (\mathbb{F}_p)^u \to (\mathbb{F}_p)^v$ an affine transformation with full rank.
- $Q\colon (\mathbb{F}_p)^v \to (\mathbb{F}_p)^v$ a function that applies a different (univariate) quadratic polynomial to each of its v inputs: $Q(x_1, ..., x_v) = (q_1(x_1), ..., q_v(x_v))$ with $q_i(x) = \alpha_i x^2 + \beta_i x + \gamma_i$ $(1 \leq i \leq v)$.

Lemma 1: If $\bar{Y}$ is a random variable following the discrete uniform distribution over $(\mathbb{F}_p)^v$, then:

$$\mathbb{E}(card(\Gamma^{-1}[\{\bar{Y}\}])) = p^{u-v}$$

Proof. Let $\bar{Y} = (Y_1, ..., Y_v)$ be a random variable following the discrete uniform distribution over $(\mathbb{F}_p)^v$.

For $i \in \{1, ..., v\}$, as explained in section 2.4, the equation (with unknown x_i) $q_i(x_i) = Y_i$ has no solutions with probability 1/2, and 2 solutions with probability 1/2.

With probability $1/2^v$, for all $i \in \{1, ..., v\}$, $q_i(x_i) = Y_i$ admits a pair of distinct solutions $\{a_i, b_i\}$, leading to $Q^{-1}[\{\bar{Y}\}] = \{a_1, b_1\} \times ... \times \{a_v, b_v\}$.

Otherwise, with probability $1 - 1/2^v$, there exists $i \in \{1, ..., v\}$ such that the equation $q_i(x_i) = Y_i$ has no solutions, leading to $Q^{-1}[\{\bar{Y}\}] = \emptyset$.

As a result:

$$Q^{-1}[\{\bar{Y}\}] = \begin{cases} \text{a set of } 2^v \text{ elements of } (\mathbb{F}_p)^v : \{c_1, ..., c_{2^v}\} & \text{with probability } 1/2^v \\ \emptyset & \text{with probability } 1 - 1/2^v \end{cases}$$

- Suppose $Q^{-1}[\{\bar{Y}\}] = \emptyset$:

  Then $\Gamma^{-1}[\{\bar{Y}\}] = T^{-1} \circ Q^{-1}[\{\bar{Y}\}] = T^{-1}[\emptyset] = \emptyset$

  Which leads to: $\mathbb{E}(card(\Gamma^{-1}[\{\bar{Y}\}]) \mid Q^{-1}[\{\bar{Y}\}] = \emptyset) = \mathbb{E}(card(\emptyset)) = 0$

- Suppose $Q^{-1}[\{\bar{Y}\}] \neq \emptyset$:

  Therefore, $Q^{-1}[\{\bar{Y}\}] = \{c_1, ..., c_{2^v}\}$ with $c_1, ..., c_{2^v}$ 2^v distinct elements of $(\mathbb{F}_p)^v$.

  Hence, $\Gamma^{-1}[\{\bar{Y}\}] = T^{-1} \circ Q^{-1}[\{\bar{Y}\}] = T^{-1}[\{c_1, ..., c_{2^v}\}] = T^{-1}[\{c_1\}] \bigcup ... \bigcup T^{-1}[\{c_{2^v}\}]$

  The c_i being distinct: $card(\Gamma^{-1}[\{\bar{Y}\}]) = card(T^{-1}[\{c_1\}]) + ... + card(T^{-1}[\{c_{2^v}\}])$

  T is an affine transformation with full rank, from $(\mathbb{F}_p)^u$ to $(\mathbb{F}_p)^v$ $(u \leq v)$. As explained in section 2.5, for $i \in \{1, ..., 2^v\}$, $card(T^{-1}[\{c_i\}])$ is following the Bernoulli distribution of parameter p^{u-v}, which leads to:

  $$\mathbb{E}(card(\Gamma^{-1}[\{\bar{Y}\}]) \mid Q^{-1}[\{\bar{Y}\}] \neq \emptyset) = \mathbb{E}(card(T^{-1}[\{c_1\}])) + ... + \mathbb{E}(card(T^{-1}[\{c_{2^v}\}])) = 2^v p^{u-v}$$

So, with the law of total expectation:

$$\begin{aligned} \mathbb{E}(card(\Gamma^{-1}[\{\bar{Y}\}])) &= \mathbb{P}(Q^{-1}[\{\bar{Y}\}] \neq \emptyset) \times \mathbb{E}(card(\Gamma^{-1}[\{\bar{Y}\}]) \mid Q^{-1}[\{\bar{Y}\}] \neq \emptyset) \\ &\quad + \mathbb{P}(Q^{-1}[\{\bar{Y}\}] = \emptyset) \times \mathbb{E}(card(\Gamma^{-1}[\{\bar{Y}\}]) \mid Q^{-1}[\{\bar{Y}\}] = \emptyset) \\ &= (1/2)^v \times (2^v p^{u-v}) + 0 = p^{u-v} \end{aligned}$$

■

Lemma 2: If $\bar{Y}$ is a random variable following the discrete uniform distribution over $(\mathbb{F}_p)^v$, the average number of modular multiplications needed to compute $\Gamma^{-1}[\{\bar{Y}\}]$ is $O(log(p) + u^2 v)$.

Proof. Let $\bar{Y} = (Y_1, ..., Y_v)$ be a random variable following the discrete uniform distribution over $(\mathbb{F}_p)^v$.

For $i \in \{1, ..., v\}$, as explained in section 2.4, solving the equation (with unknown x_i) $q_i(x_i) = Y_i$ requires $O(log(p))$ modular multiplications, leading to 2 solutions with probability 1/2, and no solutions with probability 1/2.

To compute $Q^{-1}[\{(Y_1, ..., Y_v)\}]$, we begin by solving the equation $\alpha_1 x^2 + \beta_1 x + \gamma_1 = Y_1$. This costs $O(log(p))$ modular multiplications. With probability 1/2, the previous equation has solutions, and we solve the next equation $\alpha_2 x^2 + \beta_2 x + \gamma_2 = Y_2$. Again , this costs $O(log(p))$ modular multiplications.

With probability $(1/2)^2$, both previous equations have solutions, and we solve the next equation... As a result, the average number of modular multiplications required to compute $Q^{-1}[\{(Y_1, ..., Y_v)\}]$ is:

$$O(\sum_{i=1}^{v-1} i\,(1/2)^i log(p) + v\,(1/2)^{v-1} log(p)) = O((2 - 1/2^{v-1})\,log(p)) = O(log(p))$$

As explained in the previous proof, $Q^{-1}[\{(Y_1, ..., Y_v)\}]$ has 2^v solutions with probability $1/2^v$. Otherwise, with probability $1 - 1/2^v$ there are no solutions. For each solution c, we need to compute $T^{-1}[\{c\}]$, which requires $O(u^2 v)$ modular multiplications with Gaussian elimination. As a result, inverting T requires on average a total of $O((1/2^v)\,2^v\,u^2 v) = O(u^2 v)$ modular multiplications.

Considering both Q and T, the average number of modular multiplications required to compute $\Gamma^{-1}[\{\bar{Y}\}]$ is $O(log(p) + u^2 v)$.

■

Lemma 3: If $\bar{Y}$ is a random variable following the discrete uniform distribution over $(\mathbb{F}_p)^v$, then:

$$\mathbb{E}(card(\Gamma^{-1}[\{\bar{Y}\}]) \mid \Gamma^{-1}[\{\bar{Y}\}] \neq \emptyset) = 1 + p^{u-v}(2^v - 1)$$

Proof. Let $\bar{Y} = (Y_1, ..., Y_v)$ be a random variable following the discrete uniform distribution over $(\mathbb{F}_p)^v$. Let's suppose that $\Gamma^{-1}[\{\bar{Y}\}] \neq \emptyset$.

Then $Q^{-1}[\{\bar{Y}\}] \neq \emptyset$ (because otherwise: $\Gamma^{-1}[\{\bar{Y}\}] = T^{-1} \circ Q^{-1}[\{\bar{Y}\}] = T^{-1}[\emptyset] = \emptyset$).

What follows is similar to the proof of Lemma 1:

$Q^{-1}[\{\bar{Y}\}] = \{c_1, ..., c_{2^v}\}$ with $c_1, ..., c_{2^v}$ 2^v distinct elements of $(\mathbb{F}_p)^v$.

Because $\Gamma^{-1}[\{\bar{Y}\}] \neq \emptyset$, we know for sure that there is $c \in \{c_1, ..., c_{2^v}\}$ such that $T^{-1}[\{c\}] \neq \emptyset$. For this particular value, $card(T^{-1}[\{c\}]) = 1$ (see section 2.5).

For the other $c' \in \{c_1, ..., c_{2^v}\} \setminus \{c\}$, $card(T^{-1}[\{c'\}])$ is following the Bernoulli distribution of parameter p^{u-v}.

As a result:
$$\begin{aligned} card(\Gamma^{-1}[\{\bar{Y}\}]) &= card(T^{-1} \circ Q^{-1}[\{\bar{Y}\}]) \\ &= card(T^{-1}[\{c_1, ..., c_{2^v}\}]) \\ &= card(T^{-1}[\{c_1\}]) + ... + card(T^{-1}[\{c_{2^v}\}]) \\ &= \underbrace{card(T^{-1}[\{c\}])}_{=1} + \sum_{c' \in \{c_1, ..., c_{2^v}\} \setminus \{c\}} card(T^{-1}[\{c'\}]) \end{aligned}$$

Which leads to:

$$\begin{aligned} \mathbb{E}(card(\Gamma^{-1}[\{\bar{Y}\}]) \mid \Gamma^{-1}[\{\bar{Y}\}] \neq \emptyset) &= 1 + \sum_{c' \in \{c_1, ..., c_{2^v}\} \setminus \{c\}} \mathbb{E}(card(T^{-1}[\{c'\}])) \\ &= 1 + \sum_{c' \in \{c_1, ..., c_{2^v}\} \setminus \{c\}} p^{u-v} \\ &= 1 + p^{u-v}(2^v - 1) \end{aligned}$$

■

Lemma 4: If $\bar{Y}$ is a random variable following the discrete uniform distribution over $(\mathbb{F}_p)^v$, the number of modular multiplications required to compute $\Gamma^{-1}[\{\bar{Y}\}]$ given that $\Gamma^{-1}[\{\bar{Y}\}] \neq \emptyset$ is $O(log(p)v + 2^v u^2 v)$.

Proof. Let $\bar{Y} = (Y_1, ..., Y_v)$ be a random variable following the discrete uniform distribution over $(\mathbb{F}_p)^v$. Let's suppose that $\Gamma^{-1}[\{\bar{Y}\}] \neq \emptyset$.

For $i \in \{1, ..., v\}$, as explained in section 2.4, solving the equation (with unknown x_i) $q_i(x_i) = Y_i$ requires $O(log(p))$ modular multiplications, and always leads to two solutions (because otherwise $\Gamma^{-1}[\{\bar{Y}\}] = \emptyset$). As a result, computing $Q^{-1}[\{\bar{Y}\}]$ requires $O(v \times log(p))$ modular multiplications.

As explained in the proof of Lemma 1, $Q^{-1}[\{\bar{Y}\}]$ is a set of 2^v elements of $(\mathbb{F}_p)^v$. For each $c \in Q^{-1}[\{\bar{Y}\}]$, computing $T^{-1}[\{c\}]$ requires $O(u^2 v)$ modular multiplications with Gaussian elimination. As a result, inverting T requires a total of $O(2^v u^2 v)$ modular multiplications.

Considering both Q and T, the number of modular multiplications required to compute $\Gamma^{-1}[\{\bar{Y}\}]$ given that $\Gamma^{-1}[\{\bar{Y}\}] \neq \emptyset$ is $O(log(p)v + 2^v u^2 v)$.

■

Part 2: Average number of possible solutions at each step of the decryption

Suppose Bob chooses a message $\bar{X} = (X_1, ..., X_{a_1})$ represented as a random variable following the discrete uniform distribution over $(\mathbb{F}_p)^{a_1}$. Bob encrypts $\bar{X}$ with Alice's secret key. Let $\bar{Y} = (Y_1, ..., Y_{a_m}) \in (\mathbb{F}_p)^{a_m}$ be the corresponding encrypted message. To decrypt it, Alice uses her secret key $T_1, ..., T_{m-1}, Q_1, ..., Q_{m-2}$ and computes:

$$T_1^{-1} \circ Q_1^{-1} \circ ... \circ T_{m-2}^{-1} \circ Q_{m-2}^{-1} \circ T_{m-1}^{-1}[\{\bar{Y}\}]$$

To simplify the notations, we define: $\Gamma_k = Q_k \circ T_k \colon (\mathbb{F}_p)^{a_k} \to (\mathbb{F}_p)^{a_{k+1}}$ for $k \in \{1..., m-2\}$.

The set Alice must compute to find $\bar{X}$ can now be written as: $\Gamma_1^{-1} \circ .... \circ \Gamma_{m-3}^{-1} \circ \Gamma_{m-2}^{-1} \circ T_{m-1}^{-1}[\{\bar{Y}\}]$

- First, Alice computes $\Lambda_{m-1} = T_{m-1}^{-1}[\{\bar{Y}\}]$.
- Then, she computes $\Lambda_{m-2} = \Gamma_{m-2}^{-1}[\Lambda_{m-1}]$.
- Then, she computes $\Lambda_{m-3} = \Gamma_{m-3}^{-1}[\Lambda_{m-2}]$.
- ...
- Finally, she computes $\Lambda_1 = \Gamma_1^{-1}[\Lambda_2]$.

If one Λ_k was empty, the following $\Lambda_{k-1}, ..., \Lambda_1$ would also be empty, which is impossible because Λ_1 must contain $\bar{X}$. As a result, for all $k \in \{1, ..., m-1\}, \Lambda_k \neq \emptyset$.

Let N_k be the random variable counting the number of elements found in Λ_k: $N_k = card(\Lambda_k)$ for $k \in \{1, ..., m-1\}$.

$T_{m-1}^{-1}[\{\bar{Y}\}] = \{\tau\}$ with $\tau \in (\mathbb{F}_p)^{a_{m-1}}$. τ exists because Λ_{m-1} is not empty and is unique as explained in section 2.5. As a consequence, $N_{m-1} = card(\Lambda_{m-1}) = card(\{\tau\}) = 1$.

Let $k \in \{1, ..., m-2\}$.

Λ_{k+1} is a set of N_{k+1} distinct elements of $(\mathbb{F}_p)^{a_{k+1}}$: $\tau_1, ..., \tau_{N_{k+1}}$.

$\Lambda_k = \Gamma_k^{-1}[\Lambda_{k+1}] = \Gamma_k^{-1}[\{\tau_1, ..., \tau_{N_{k+1}}\}] = \Gamma_k^{-1}[\{\tau_1\}] \bigcup ... \bigcup \Gamma_k^{-1}[\{\tau_{N_{k+1}}\}]$

Because $\Lambda_k \neq \emptyset$, we know for sure that there is $\boldsymbol{\tau} \in \{\tau_1, ..., \tau_{N_{k+1}}\}$ such that $\Gamma_k^{-1}[\{\boldsymbol{\tau}\}] \neq \emptyset$. For this particular value, $\mathbb{E}(card(\Gamma_k^{-1}[\{\boldsymbol{\tau}\}])) = 1 + p^{a_k - a_{k+1}}(2^{a_{k+1}} - 1)$ (see Lemma 3).

We can reasonably consider that:

- The identically distributed random variables: $card(\Gamma_k^{-1}[\{\tau'\}])$ for $\tau' \in \{\tau_1, ..., \tau_{N_{k+1}}\} \setminus \{\boldsymbol{\tau}\}$, with expectation $p^{a_k - a_{k+1}}$ (see Lemma 1), are independent.
- N_{k+1} is independent of the sequence $(card(\Gamma_k^{-1}[\{\tau'\}]))$

Wald's equation leads to:

$$\begin{aligned}\mathbb{E}(N_k) =& \ \mathbb{E}(card(\Gamma_k^{-1}[\{\boldsymbol{\tau}\}])) + \mathbb{E}\left(\sum_{\tau' \in \{\tau_1,...,\tau_{N_{k+1}}\}\setminus\{\boldsymbol{\tau}\}} card(\Gamma_k^{-1}[\{\tau'\}])\right)\\ =& \ 1 + p^{a_k - a_{k+1}}(2^{a_{k+1}} - 1) + p^{a_k - a_{k+1}}(\mathbb{E}(N_{k+1}) - 1)\end{aligned}$$

As a result, we get: $\underbrace{(\mathbb{E}(N_k) - 1)/p^{a_k}}_{u_k} = (2^{a_{k+1}} - 1)/p^{a_{k+1}} + \underbrace{(\mathbb{E}(N_{k+1}) - 1)/p^{a_{k+1}}}_{u_{k+1}}$

By using $u_k = (2^{a_{k+1}} - 1)/p^{a_{k+1}} + u_{k+1}$, we obtain $u_k = \sum_{v=k+1}^{m-1} (2^{a_v} - 1)/p^{a_v} + u_{m-1}$

And because $u_{m-1} = (\mathbb{E}(N_{m-1}) - 1)/p^{a_{m-1}} = (\mathbb{E}(1) - 1)/p^{a_{m-1}} = 0$, we finally obtain:

$$\mathbb{E}(N_k) = 1 + p^{a_k} u_k = 1 + \sum_{v=k+1}^{m-1} p^{a_k - a_v}(2^{a_v} - 1) \qquad (\star)$$

Part 3: Average number of operations to decrypt a message

Lemma 5: For $k \in \{1, ..., m-2\}$, the average number of modular multiplications to compute $\Lambda_k = \Gamma_k^{-1}[\Lambda_{k+1}]$, given Λ_{k+1}, is:

$$O\left(log(p)\, a_{k+1} + 2^{a_{k+1}} a_k^2 a_{k+1} + (log(p) + a_k^2 a_{k+1}) \times \sum_{v=k+2}^{m-1} p^{a_{k+1} - a_v}(2^{a_v} - 1)\right)$$

Proof. Let $k \in \{1, ..., m-2\}$.

We suppose known Λ_{k+1}, a set of N_{k+1} distinct elements of $(\mathbb{F}_p)^{a_{k+1}}$: $\Lambda_{k+1} = \{\tau_1, ..., \tau_{N_{k+1}}\}$. Computing $\Gamma_k^{-1}[\Lambda_{k+1}]$ requires to compute $\Gamma_k^{-1}[\{\tau_1\}], ..., \Gamma_k^{-1}[\{\tau_{N_{k+1}}\}]$

As explained in section 4.3.2, we know for sure that there is $\boldsymbol{\tau} \in \{\tau_1, ..., \tau_{N_{k+1}}\}$ such that $\Gamma_k^{-1}[\{\boldsymbol{\tau}\}] \neq \emptyset$.

For this particular value, $O(log(p)\, a_{k+1} + 2^{a_{k+1}} a_k^2 a_{k+1})$ modular multiplications are required to compute $\Gamma_k^{-1}[\{\boldsymbol{\tau}\}]$ (see Lemma 4).

For the other values $\tau' \in \{\tau_1, ..., \tau_{N_{k+1}}\} \setminus \{\boldsymbol{\tau}\}$, $O(log(p) + a_k^2 a_{k+1})$ modular multiplications are required to compute $\Gamma_k^{-1}[\{\tau'\}]$ (see Lemma 2).

In total, $O(log(p)\, a_{k+1} + 2^{a_{k+1}} a_k^2 a_{k+1}) + (N_{k+1} - 1)(log(p) + a_k^2 a_{k+1}))$ modular multiplications are required to compute $\Gamma_k^{-1}[\Lambda_{k+1}]$. As a result, the average number of modular multiplications required to compute $\Gamma_k^{-1}[\Lambda_{k+1}]$ is:

$$\begin{aligned}& O(log(p)\, a_{k+1} + 2^{a_{k+1}} a_k^2 a_{k+1} + (\mathbb{E}(N_{k+1}) - 1) \times (log(p) + a_k^2 a_{k+1}))\\ \underset{\text{using } (\star)}{=} & O(log(p)\, a_{k+1} + 2^{a_{k+1}} a_k^2 a_{k+1} + (1 + \sum_{v=k+2}^{m-1} p^{a_{k+1} - a_v}(2^{a_v} - 1) - 1) \times (log(p) + a_k^2 a_{k+1}))\\ =& O(log(p)\, a_{k+1} + 2^{a_{k+1}} a_k^2 a_{k+1} + (log(p) + a_k^2 a_{k+1}) \times \sum_{v=k+2}^{m-1} p^{a_{k+1} - a_v}(2^{a_v} - 1))\end{aligned}$$

■

Decrypting a message requires computing $\Lambda_{m-1} = T_{m-1}^{-1}[\{\bar{Y}\}]$, then $\Lambda_{m-2} = \Gamma_{m-2}^{-1}[\Lambda_{m-1}]$, then $\Lambda_{m-3} = \Gamma_{m-3}^{-1}[\Lambda_{m-2}]$ ...

The first inversion $T_{m-1}^{-1}[\{\bar{Y}\}]$ requires $O(a_{m-1}^2 a_m)$ modular multiplications with Gaussian elimination. The other inversions are treated with Lemma 5.

As a result, the total number of modular multiplications required to decrypt a message is on average:

$$O(a_{m-1}^2 a_m + \sum_{k=1}^{m-2} \left(log(p)\, a_{k+1} + 2^{a_{k+1}} a_k^2 a_{k+1} + (log(p) + a_k^2 a_{k+1}) \times \sum_{v=k+2}^{m-1} p^{a_{k+1}-a_v} (2^{a_v} - 1) \right))$$

Which ends the proof of the complexity of decryption.

9 References

[1] W. Diffie and M. Hellman, "New directions in cryptography," in IEEE Transactions on Information Theory, vol. 22, no. 6, pp. 644-654, November 1976, doi: 10.1109/TIT.1976.1055638

[2] R. L. Rivest, A. Shamir, and L. Adleman. 1978. A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM 21, 2 (Feb. 1978), 120–126.

[3] Patarin J. (1996) Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. In: Maurer U. (eds) Advances in Cryptology — EUROCRYPT '96. EUROCRYPT 1996. Lecture Notes in Computer Science, vol 1070. Springer, Berlin, Heidelberg.

[4] Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced oil and vinegar signature schemes. In Eurocrypt'99, LNCS, volume 1592, pages 206–222. Springer, 1999.

[5] Tonelli, A.: Bemerkung ûber die Aufl ösung quadratischer Congruenzen. Gottinger Nachrichten (1891) 344–346

[6] Volker Diekert; Manfred Kufleitner; Gerhard Rosenberger; Ulrich Hertrampf (24 May 2016). Discrete Algebraic Methods: Arithmetic, Cryptography, Automata and Groups. De Gruyter. pp. 163–165. ISBN 978-3-11-041632-9

[7] Previous version of this protocol (broken by Charles Bouillaguet)