Hidden Irreducible Polynomials : A cryptosystem based on Multivariate Public Key Cryptography

Borja Gómez
kub0x@elhacker.net

October 9, 2019

Abstract

Asymmetric schemes are moving towards a new series of cryptosystems based on known open problems that until the day guarantee security from the point that are not solvable under determined properties. In this paper you can read a novel research done mostly on the field of Multivariate Public Key Cryptography that focus the interest on sharing a pre-master key between Alice and Bob using quadratic multivariate polynomials as the public key. What does this scheme somehow special is that it uses a private construction involving polynomial factorization that allows Alice to recover the secret sent by Bob.

1 Introduction

The main goal of this paper is to show that Alice and Bob can establish a secure connection by sharing material that Alice can only recover. The attacker, as always, must solve a problem or a series of these to break their communication. In this case, he is going to work on factoring quadratic multivariate non-linear polynomials over a field.

2 Motivation

Multivariate Public Key Cryptography has been around since the '88 when the scheme of Matsumoto-Imai[1] was announced. It was a fashionable and new scheme that provided new ideas of creating asymmetric key cryptosystems using polynomials over finite fields. The trapdoor consisted on computing $X^{q^{\rho}+1}$ to hide the original polynomial. Patarin[2] shown that it can be broken by after only $m^2n^4 \log n$ computations. New attempts[3] were made as FLASH, Oil-Vinegar, Rainbow and others. This work has been done under the perspective of the techniques used in these cryptosystems along with properties of finite fields.

3 Background

It is better to explain some concepts and ideas to put the reader in context. Algebraically this scheme uses mainly polynomial factorization over a field of q elements, but it understanding Finite Fields is important as well. A Finite Field F_{p^n} is an algebraic structure that satisfies ring and group axioms and it's constructed as the polynomial quotient ring $F_{p^n} = Z_p[x]/f(x))$ where f is an irreducible polynomial of degree n over F_p. It's a common structure found in plenty cryptographic schemes both symmetric and asymmetric. Every element of a finite field has an inverse and that's a condition that eases algebraic constructions, besides, the order of the multiplicative group F_q^* is known to be $q^n - 1$ where $q = p^n$, thus exponentiation is affordable and invertible/reversable when the exponent is coprime to the order.

In addition, the vector space F_q^n is identified with the finite field F_{q^n} allowing to represent an element as a vector or as a polynomial. This way an element can be easily transformed using linear transformations over F_q then switching back to F_{q^n} and so on.

4 Scheme construction

A few remarks before describing the algebraic construction of the scheme. The main concept is that Alice multiplies two polynomials $(p \cdot q)(x) = r(x)$, transforms the resulting polynomial by $\phi^{-1}(T\phi(r(x)) = P$ and sends it to Bob as her public key. Bob chooses two irreducible polynomials of particular degree, inputs the combined coordinate tuple of both irred. polynomials in Alice's public key and outputs $P(X)$: the integer representation of Alice's public key under values $(y_1, \cdots, y_{2(k+1)})$. Now Alice inverts by $\phi^{-1}(T^{-1}\phi(P)) = r(x)$. Thus Alice factors $r(x)$ into the product of both $p(x), q(x)$ over F_q recovering the private coordinate tuple used by Bob. But why? The key to understanding the factorization technique is to review the following properties:

A polynomial $f(x) \in F_{q^n}$ of degree k has polynomial factors of degree d_i that sum up to k when these factors are multiplied. This is, the product two irreducible polynomials $(p.q)(x) = r$ has these irreducible polynomials in its factorization $\iff$ $Deg(p(x)) + Deg(q(x)) \leq n - 1$. When $Deg(p(x)) + Deg(q(x)) > n - 1$ the factorization of $r(x)$ is different from $(p.q)(x)$ since we are working in F_{q^n} thus reducing the product by f, this is a very important condition and the scheme works only when the sum of degree is less than n. On Cryptography these properties are useful since there exists a map that embed two polynomial elements into one where inverting the trapdoor is possible under factoring $r(x)$ into $(p(x), q(x)))$.

First of all let $p(x), q(x) \in F_{q^n}$ be two polynomials of degree k, where $2k < n-1$ and k prime. Express them as:

$$p(x) = \sum_{i=1}^{k+1} y_i x^{i-1}$$
$$q(x) = \sum_{i=k+2}^{2(k+1)} y_i x^{i-k-2}$$

Then let ϕ be the map that sends a polynomial in F_{q^n} to a n-tuple with coefficients in F_q.

$$\phi : F_{q^n} \mapsto F_q^n \quad \phi(f(x)) = (c_0, \cdots, c_{n-1})$$
$$\phi^{-1} : F_q^n \mapsto F_{q^n} \quad \phi(x) = f(x) = \sum_{i=0}^{n-1} c_i x^i$$

The map $\mathscr{F}$ sends two elements of F_{q^n} to another in F_{q^n} under multiplication:

$$\mathscr{F} : F_{q^n}^2 \mapsto F_{q^n} \quad \mathscr{F}(p(x), q(x)) = r(x)$$
$$\mathscr{F}^{-1} : F_{q^n} \mapsto F_{q^n}^2 \quad \mathscr{F}^{-1}(r(x)) = (p(x), q(x))$$

Let $T \in F_q^{2k+1 \times 2k+1}$ be the transformation matrix that combines the resulting multivariate quadratic polynomials in the vector space F_q^n as follows:

$$T : F_q^n \mapsto F_q^n \quad T(x) = T \cdot x$$

Then the whole scheme can be represented as composition of the aforementioned maps:

$$P(p_1(y_1, \cdots, y_{2(k+1)}), \cdots, p_{2k+1}(y_1, \cdots, x_{2(k+1)})) = \phi^{-1} \circ T \circ \phi \circ \mathscr{F}(p(x), q(x))$$

As $p(x), q(x)$ have degree k their tuple representation has $k+1$ variables each. Their multiplication results on the polynomial $r(x)$ of degree $2k$, thus $r(x)$ is a $2k+1$-tuple. After transforming with T, the public key P contains $2k+1$ polynomials on $2(k+1)$ variables. Summarizing, there is **apparently** one more variable than polynomials.

Bob receives the public key P, chooses two irreducible polynomials $p(x), q(x)$ of degree k over F_q that are elements in F_{q^n}. Then he inputs in P the two coefficient list of p, q concatenated, these are $2(k+1)$ variables and it is written as:

$$P(\phi(p(x)||\phi(q(x)))) = P(y_1, \cdots, y_{2(k+1)})$$

It is clear that the variables $y_{k+1} = 1$ and $y_{2(k+1)} = 1$ such that $p(x), q(x)$ both have degree k. Thus from $2(k+1)$ two are reserved to denote degree k, thus there are simply $2k$ variables and $2k+1$ polynomials.

4.1 But what does Bob transmit?

Bob is forced to select two irreducible polynomials over F_q being not able to freely choose whatever $2(k+1)$ variables he wants. Both irreducible polynomials define the pre-master key that can be combined with a salt or nonce to be used into a KDF for obtaining symmetric key material for confidentiality and integrity. As seen later, the attacker will figure out how obtain information only using irreducible polynomials over F_q.

5 Alternative constructions

Comparing it to other schemes you can find that transformations on the private tuple are done before reaching $\mathscr{F}$. In this scheme is crucial that $p(x), q(x)$ are multiplied without initially modifying them, as Bob will select two irred. polynomials over F_q that will end up factorising into both $p(x)$ and $q(x)$. Transforming the variable tuples as seen in other schemes arises a different scenario:

Define variable tuples for two polynomials $p(x), q(x)$ as $y_1 = (y_1, \cdots, y_{k+1})$ and $y_2 = (y_{k+2}, \cdots, y_{2(k+1)})$. Select $S_1, S_2 \in F_q^{n\times n}$ and present the following composition of maps:

$$\phi^{-1} \circ T \circ \phi \circ \mathscr{F}(\phi^{-1}(S_1 \cdot y_1), \phi^{-1}(S_2 \cdot y_2))$$

You end up multiplying two polynomials in $\mathscr{F}$ that **may** not be irreducible thus not recoverable later when inverting $\mathscr{F}$ since transforming polynomials before multiplication changes their structure. The rule is to transform the output of $\mathscr{F}$ once both $p(x), q(x)$ are multiplied. The approach here is to use a linear transformation T, but exponentiation on F_{q^n} is possible too as $\mathscr{F}(p(x), q(x)) = r(x)$ thus $r^e \equiv_f h$ and $h^d \equiv_f r$ when $\gcd(e, q^n - 1) = 1$ and $ed \equiv_{(q^n-1)} 1$.

Another remark is that redefining $\mathscr{F}$ as $\mathscr{F}(p(x), p(x)) = p^2(x) = r(x)$ gives $2k+1$ quadratic polynomials on $k+1$ variables when $q \neq 2$. Thus after applying the transformation T the system $P(X)$ results to be over-determined. It is another approach but the number of equations approximately doubles the size comparing to the number of variables. The original scheme provides almost the same number of variables and equations.

6 Example

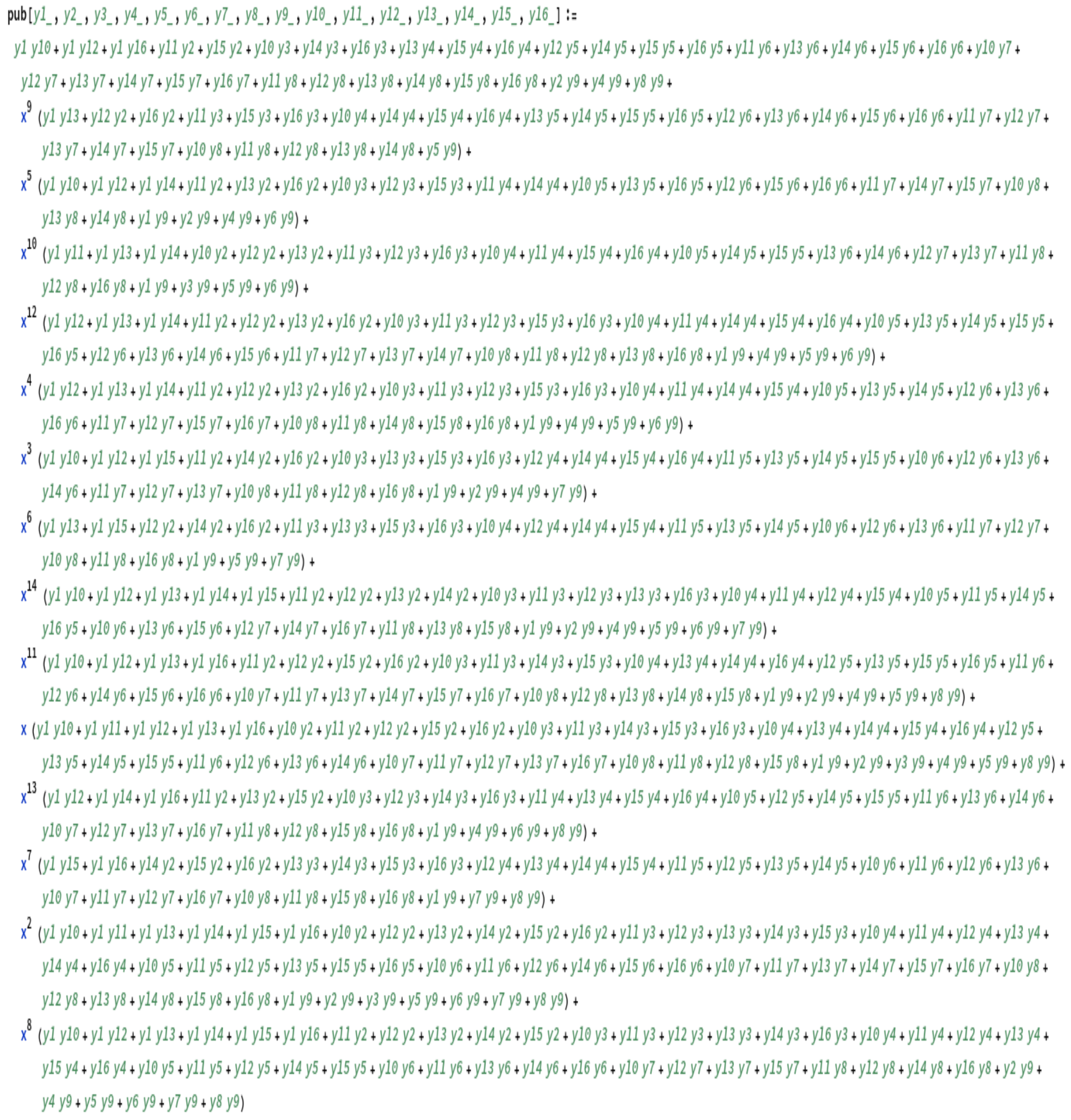
Take $q = 2, n = 17, k = 7$. Let's construct the public key, as said before consider $p(x) = \sum_{i=1}^{k+1} y_i x^{i-1} \quad q(x) = \sum_{i=k+2}^{2(k+1)} y_i x^{i-k-2}$. Multiply both polynomials as seen in $\mathscr{F}(p(x), q(x)) = r$. Then think of $r(x)$ as a column vector ordered by the powers of x:

$$y1y9+ \tag{1}$$
$$x(y1y10 + y2y9)+ \tag{2}$$
$$x^2(y1y11 + y10y2 + y3y9)+ \tag{3}$$
$$x^3(y1y12 + y11y2 + y10y3 + y4y9)+ \tag{4}$$
$$x^4(y1y13 + y12y2 + y11y3 + y10y4 + y5y9)+ \tag{5}$$
$$x^5(y1y14 + y13y2 + y12y3 + y11y4 + y10y5 + y6y9)+ \tag{6}$$
$$x^6(y1y15 + y14y2 + y13y3 + y12y4 + y11y5 + y10y6 + y7y9)+ \tag{7}$$
$$x^7(y1y16 + y15y2 + y14y3 + y13y4 + y12y5 + y11y6 + y10y7 + y8y9) \tag{8}$$
$$x^8(y16y2 + y15y3 + y14y4 + y13y5 + y12y6 + y11y7 + y10y8)+ \tag{9}$$
$$x^9(y16y3 + y15y4 + y14y5 + y13y6 + y12y7 + y11y8)+ \tag{10}$$
$$x^{10}(y16y4 + y15y5 + y14y6 + y13y7 + y12y8)+ \tag{11}$$
$$x^{11}(y16y5 + y15y6 + y14y7 + y13y8)+ \tag{12}$$
$$x^{12}(y16y6 + y15y7 + y14y8)+ \tag{13}$$
$$x^{13}(y16y7 + y15y8)+ \tag{14}$$
$$x^{14}y16y8 \tag{15}$$

Now let's transform this polynomial $r(x)$ using T. Define T and remove powers of x from the column vector (as it would be linear algebra):

$$T \in F_q^{15\times15} = \begin{pmatrix}
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\
1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0
\end{pmatrix}$$

The public key is $P = T.\phi(r(x))$ resulting in the following polynomial over F_{q^n}



```
pub[y1_, y2_, y3_, y4_, y5_, y6_, y7_, y8_, y9_, y10_, y11_, y12_, y13_, y14_, y15_, y16_] :=
 y1 y10 + y1 y12 + y1 y16 + y11 y2 + y15 y2 + y10 y3 + y14 y3 + y16 y3 + y13 y4 + y15 y4 + y16 y4 + y12 y5 + y14 y5 + y15 y5 + y16 y5 + y11 y6 + y13 y6 + y14 y6 + y15 y6 + y16 y6 + y10 y7 +
  y12 y7 + y13 y7 + y14 y7 + y15 y7 + y16 y7 + y11 y8 + y12 y8 + y13 y8 + y14 y8 + y15 y8 + y16 y8 + y2 y9 + y4 y9 + y8 y9 +
  x^9 (y1 y13 + y12 y2 + y16 y2 + y11 y3 + y15 y3 + y16 y3 + y10 y4 + y14 y4 + y15 y4 + y16 y4 + y13 y5 + y14 y5 + y15 y5 + y16 y5 + y12 y6 + y13 y6 + y14 y6 + y15 y6 + y16 y6 + y11 y7 + y12 y7 +
     y13 y7 + y14 y7 + y15 y7 + y10 y8 + y11 y8 + y12 y8 + y13 y8 + y14 y8 + y5 y9) +
  x^5 (y1 y10 + y1 y12 + y1 y14 + y11 y2 + y13 y2 + y16 y2 + y10 y3 + y12 y3 + y15 y3 + y11 y4 + y14 y4 + y10 y5 + y13 y5 + y16 y5 + y12 y6 + y15 y6 + y16 y6 + y11 y7 + y14 y7 + y15 y7 + y10 y8 +
     y13 y8 + y14 y8 + y1 y9 + y2 y9 + y4 y9 + y6 y9) +
  x^10 (y1 y11 + y1 y13 + y1 y14 + y10 y2 + y12 y2 + y13 y2 + y11 y3 + y12 y3 + y16 y3 + y10 y4 + y11 y4 + y15 y4 + y16 y4 + y10 y5 + y14 y5 + y15 y5 + y13 y6 + y14 y6 + y12 y7 + y13 y7 + y11 y8 +
     y12 y8 + y16 y8 + y1 y9 + y3 y9 + y5 y9 + y6 y9) +
  x^12 (y1 y12 + y1 y13 + y1 y14 + y11 y2 + y12 y2 + y13 y2 + y16 y2 + y10 y3 + y11 y3 + y12 y3 + y15 y3 + y16 y3 + y10 y4 + y11 y4 + y14 y4 + y15 y4 + y16 y4 + y10 y5 + y13 y5 + y14 y5 + y15 y5 +
     y16 y5 + y12 y6 + y13 y6 + y14 y6 + y15 y6 + y11 y7 + y12 y7 + y13 y7 + y14 y7 + y10 y8 + y11 y8 + y12 y8 + y13 y8 + y16 y8 + y1 y9 + y4 y9 + y5 y9 + y6 y9) +
  x^4 (y1 y12 + y1 y13 + y1 y14 + y11 y2 + y12 y2 + y13 y2 + y16 y2 + y10 y3 + y11 y3 + y12 y3 + y15 y3 + y16 y3 + y10 y4 + y11 y4 + y14 y4 + y15 y4 + y10 y5 + y13 y5 + y14 y5 + y12 y6 + y13 y6 +
     y16 y6 + y11 y7 + y12 y7 + y15 y7 + y16 y7 + y10 y8 + y11 y8 + y14 y8 + y15 y8 + y16 y8 + y1 y9 + y4 y9 + y5 y9 + y6 y9) +
  x^3 (y1 y10 + y1 y12 + y1 y15 + y11 y2 + y14 y2 + y16 y2 + y10 y3 + y13 y3 + y15 y3 + y16 y3 + y12 y4 + y14 y4 + y15 y4 + y16 y4 + y11 y5 + y13 y5 + y14 y5 + y15 y5 + y10 y6 + y12 y6 + y13 y6 +
     y14 y6 + y11 y7 + y12 y7 + y13 y7 + y10 y8 + y11 y8 + y12 y8 + y16 y8 + y1 y9 + y2 y9 + y4 y9 + y7 y9) +
  x^6 (y1 y13 + y1 y15 + y12 y2 + y14 y2 + y16 y2 + y11 y3 + y13 y3 + y15 y3 + y16 y3 + y10 y4 + y12 y4 + y14 y4 + y15 y4 + y11 y5 + y13 y5 + y14 y5 + y10 y6 + y12 y6 + y13 y6 + y11 y7 + y12 y7 +
     y10 y8 + y11 y8 + y16 y8 + y1 y9 + y5 y9 + y7 y9) +
  x^14 (y1 y10 + y1 y12 + y1 y13 + y1 y14 + y1 y15 + y11 y2 + y12 y2 + y13 y2 + y14 y2 + y10 y3 + y11 y3 + y12 y3 + y13 y3 + y16 y3 + y10 y4 + y11 y4 + y12 y4 + y15 y4 + y10 y5 + y11 y5 + y14 y5 +
     y16 y5 + y10 y6 + y13 y6 + y15 y6 + y12 y7 + y14 y7 + y16 y7 + y11 y8 + y13 y8 + y15 y8 + y1 y9 + y2 y9 + y4 y9 + y5 y9 + y6 y9 + y7 y9) +
  x^11 (y1 y10 + y1 y12 + y1 y13 + y1 y16 + y11 y2 + y12 y2 + y15 y2 + y16 y2 + y10 y3 + y11 y3 + y14 y3 + y15 y3 + y10 y4 + y13 y4 + y14 y4 + y16 y4 + y12 y5 + y13 y5 + y15 y5 + y16 y5 + y11 y6 +
     y12 y6 + y14 y6 + y15 y6 + y16 y6 + y10 y7 + y11 y7 + y13 y7 + y14 y7 + y15 y7 + y16 y7 + y10 y8 + y12 y8 + y13 y8 + y14 y8 + y15 y8 + y1 y9 + y2 y9 + y4 y9 + y5 y9 + y8 y9) +
  x (y1 y10 + y1 y11 + y1 y12 + y1 y13 + y1 y16 + y10 y2 + y11 y2 + y12 y2 + y15 y2 + y16 y2 + y10 y3 + y11 y3 + y14 y3 + y15 y3 + y16 y3 + y10 y4 + y13 y4 + y14 y4 + y15 y4 + y16 y4 + y12 y5 +
     y13 y5 + y14 y5 + y15 y5 + y11 y6 + y12 y6 + y13 y6 + y14 y6 + y10 y7 + y11 y7 + y12 y7 + y13 y7 + y16 y7 + y10 y8 + y11 y8 + y12 y8 + y15 y8 + y1 y9 + y2 y9 + y3 y9 + y4 y9 + y5 y9 + y8 y9) +
  x^13 (y1 y12 + y1 y14 + y1 y16 + y11 y2 + y13 y2 + y15 y2 + y10 y3 + y12 y3 + y14 y3 + y16 y3 + y11 y4 + y13 y4 + y15 y4 + y16 y4 + y10 y5 + y12 y5 + y14 y5 + y15 y5 + y11 y6 + y13 y6 + y14 y6 +
     y10 y7 + y12 y7 + y13 y7 + y16 y7 + y11 y8 + y12 y8 + y15 y8 + y16 y8 + y1 y9 + y4 y9 + y6 y9 + y8 y9) +
  x^7 (y1 y15 + y1 y16 + y14 y2 + y15 y2 + y16 y2 + y13 y3 + y14 y3 + y15 y3 + y16 y3 + y12 y4 + y13 y4 + y14 y4 + y15 y4 + y11 y5 + y12 y5 + y13 y5 + y14 y5 + y10 y6 + y11 y6 + y12 y6 + y13 y6 +
     y10 y7 + y11 y7 + y12 y7 + y16 y7 + y10 y8 + y11 y8 + y15 y8 + y16 y8 + y1 y9 + y7 y9 + y8 y9) +
  x^2 (y1 y10 + y1 y11 + y1 y13 + y1 y14 + y1 y15 + y1 y16 + y10 y2 + y12 y2 + y13 y2 + y14 y2 + y15 y2 + y16 y2 + y11 y3 + y12 y3 + y13 y3 + y14 y3 + y15 y3 + y10 y4 + y11 y4 + y12 y4 + y13 y4 +
     y14 y4 + y16 y4 + y10 y5 + y11 y5 + y12 y5 + y13 y5 + y15 y5 + y16 y5 + y10 y6 + y11 y6 + y12 y6 + y14 y6 + y15 y6 + y16 y6 + y10 y7 + y11 y7 + y13 y7 + y14 y7 + y15 y7 + y16 y7 + y10 y8 +
     y12 y8 + y13 y8 + y14 y8 + y15 y8 + y16 y8 + y1 y9 + y2 y9 + y3 y9 + y5 y9 + y6 y9 + y7 y9 + y8 y9) +
  x^8 (y1 y10 + y1 y12 + y1 y13 + y1 y14 + y1 y15 + y1 y16 + y11 y2 + y12 y2 + y13 y2 + y14 y2 + y15 y2 + y10 y3 + y11 y3 + y12 y3 + y13 y3 + y14 y3 + y16 y3 + y10 y4 + y11 y4 + y12 y4 + y13 y4 +
     y15 y4 + y16 y4 + y10 y5 + y11 y5 + y12 y5 + y14 y5 + y15 y5 + y10 y6 + y11 y6 + y13 y6 + y14 y6 + y16 y6 + y10 y7 + y12 y7 + y13 y7 + y15 y7 + y11 y8 + y12 y8 + y14 y8 + y16 y8 + y2 y9 +
     y4 y9 + y5 y9 + y6 y9 + y7 y9 + y8 y9)
```

Till the moment, Bob has Alice's public key, so he chooses $p(x) = x^7 + x + 1$ and $q(x) = x^7 + x^5 + x^3 + x + 1$ both irreducible over F_2. He puts the coefficient list of both polynomials into the public key $P(X)$:

$P(1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1) = x+x^3+x^7+x^8+x^9+x^{11}+x^{13}+x^{14}$. Bob sends $P(X)$ back to Alice.

Alice does $r(x) = \phi^{-1}T^{-1}\phi(P(X)) = 1+x^2+x^3+x^4+x^5+x^6+x^{10}+x^{12}+x^{14}$.

Eventually, computing the factors of $r(x)$ over F_2 yields $r(x) = (x^7 + x + 1)(x^7 + x^5 + x^3 + x + 1) = p(x)q(x)$.

The whole example is basic and insecure from the point of parametrization and selection. Polynomials $p(x), q(x)$ have been randomly chosen to satisfy irreducibility. Matrix T has been randomly chosen to satisfy $r(T) = 2k + 1$, this is full rank. Moreover the irreducible polynomial set where Bob chooses $p(x), q(x)$ is limited to roughly $2^{4.16993}$ so the attacker recovers the private factors only by comparing the output Bob's $P(X)$ with his own. But this example suffices to demonstrate how the scheme works in both ways, enciphering and deciphering following the map construction. The scheme escalates well under distinct parametrization as seen later in Complexity.

7 Security

The attacker knows that Bob must choose two irreducible polynomials over F_q of degree k. He starts to wonder himself how to locate these two polynomials and what's the complexity behind his enumeration method. As the degree k is prime, the formula for determining how many irreducible polynomials of degree k exits over F_q when k prime is: $\frac{q^k-q}{k}$. Thus a estimation on the selection of parameters can be made:

The recommended parametrization is $q = 2, k = 127$ because there are about $2^{\log_2(\frac{1}{127}(2^{127}-2))} = 2^{120.01}$ irreducible polynomials of degree 127 and allows Bob to encipher $2(k+1)$ variables.. The attacker realizes that this method doesn't fit as it requires a long quantity of time for finding each selected polynomial. Moreover, he looks at the structure of irreducible polynomials and realizes that he can eliminate more variables of the system if all these polynomials fix a concrete coefficient, this is, a concrete coefficient is always present in every irreducible polynomial of degree k over F_q. This is the case of the leading coefficient on x^k that's why in previous sections $y_{k+1}, y_{2(k+1)}$ are eliminated. This attack is not going to be analyzed here, but it must be taken into consideration as would result into variable elimination/reduction.

Now, he puts the eye on the linear transformation $T \in F_q^{2k+1\times 2k+1}$. He knows that every vector can be decomposed into a product of a matrix of coefficients times a column vector this is $Ax = b$. For that he attempts to find out a common expression in every coefficient of b. The common expression turns out to be the basis and the coefficients that multiply these expressions form the matrix of coefficients. This is always possible when the vector coefficients of b are linear multivariate polynomials. But in this scheme, P contains $2(k + 1)$ multivariate quadratic polynomials, thus separating the matrix coefficient from basis depends on finding out a common simplification-reduction for every polynomial in P which it seems to be non-trivial. Other applicable transformations (like exponentiation, conjugation transform) are not discussed here since right now because computing exponentiations on symbolic-agelbraic polynomials is time consuming and results in higher degree polynomials.

Following with the security topic, now is the turn for the attacker to switch to the next area where he can recover sensitive information. He starts to think that he may be able to solve the system of multivariate quadratic polynomials $P^{-1}(Y) = X$. For that he knows that every coefficient of the polynomial of degree $2k$ is a quadratic multivariate polynomial, thus he tries to solve the system of polynomial equations by examining the complexity of the multiple existing methods to solve quadratic multivariate polynomials over F_q. Elaborates a list of candidates resulting in Linearization techniques and Gröbner Bases.

Gröbner Bases have been used extensively for solving systems of polynomial equations but they are not recommended beyond $n = m = 15$[4]. CAS software like Mathematica has a built-in implementation of this method that returns a list of polynomials that allows the user to return the roots of the initial polynomials, but it didn't work with the previous example of $m = 15, n = 16$.

(Re)Linearization[4] is a non-trivial technique that tries to express $P(X)$ as a new system of equations in linear variables. For that it express every term $y_i y_j$ as y_{ij} yielding linear equations and obtaining solutions for these new variables. This scheme uses $m = 255, n = 256$ thus it seems that these attacks cannot be applied at a first glance. But as the scheme construction is public, the attacker could linearize equations in $r(x)$ to measure how did T transformed the polynomial, and maybe, obtain information of T as $r(x)$ would be linear.

8 Complexity

Normally schemes that are not based on discrete logarithm or integer factorization have bigger sizes on their public keys. The public key of this scheme consists of $2k + 1$ equations on $2(k + 2)$ variables or $2k$ variables if eliminating $y_{k+1}, y_{2(k+1)}$.

When multiplying $p(x).q(x) = r(x)$ we see in section 6 that there exits a symmetry on the number of variables per equation, once the half of the table is passed, the number of variables start to decrease. This behaviour defines a deterministic way of computing the algebraic description of $r(x)$ programatically on $k + 1$ iterations. The public key goes after obtaining $r(x)$ by $T\phi(r(x))$ and needs an estimation on iterations-time and space.

As T is a $(2k + 1) \times (2k + 1)$ matrix, multplying by $\phi(r(x))$ consists on $2k + 1$ multiplications and sums per row resulting in $(2k + 1)^2$ sums and multiplications. Since $r(x)$ is a symbolic expression the implementation program will append to the left the integer value of $T_{i,j}$ on the symbolic value $r(x)_j$, this is, $\sum_{i=1}^{2k+1} \sum_{j=1}^{2k+1} T_{i,j} r(x)_j$. It's hard to calculate the size of $P(X)$ as the resulting quadratic symbolic equations depends on the chosen T: every column vector or polynomial equation of $P(X)$ contains distinct symbolic coefficients of $r(x)$.

For Bob, testing an irreducible polynomials over F_q by Rabin's test of irreducibility gives a total of $\mathcal{O}(n^2 \log n \log q)$ field operations.

Furthermore, Alice must recover the irreducible poylnomials chosen by Bob. First she inverts T and recovers $r(x)$. After, she analyzes the complexity of factoring over F_q, which initially was $\mathcal{O}(d^3 \log q)$ which later had been improved to $\mathcal{O}(d \log q)$[5], what turns out to be practical for this scheme.

References

[1] T. Matsumoto and H. Imai, *Public Quadratic Polynomial tuples for efficient signature-verification and message-encryption,*. EUROCRYPT'88, Springer-Verlag !988 pp. 419-453.

[2] Jacques Patarin, *Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt'88*. CRYPTO 1995: Advances in Cryptology — CRYPT0' 95 pp 248-261

[3] Christian Eder, Jean-Charles Faugère and Ludovic Perret *Multivariate Public Key Cryptography or Why is there a rainbow hidden behind fields full of oil and vinegar?*. Seminar on Fundamental Algorithms, University of Kaiserslautern, June 25, 2015 `https://www.mathematik.uni-kl.de/~ederc/download/mpkc.pdf`

[4] Nicolas Courtois, Alexander Klimov, Jacques Patarin and Adi Shamir. *Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations**. `https://www.iacr.org/archive/eurocrypt2000/1807/18070398-new.pdf`

[5] Ali Ayad *A Lecture on The Complexity of Factoring Polynomials over Global Fields*. International Mathematical Forum, 5, 2010, no. 10, 477 - 486 `http://m-hikari.com/imf-2010/9-12-2010/ayadIMF9-12-2010.pdf`