Cryptanalysis of Rational Multivariate Public Key Cryptosystems

Jintai Ding, John Wagner
Department of Mathematical Sciences,
University of Cincinnati,
Cincinnati, OH, 45220, USA

Abstract

In 1989, Tsujii, Fujioka, and Hirayama proposed a family of multivariate public key cryptosystems, where the public key is given as a set of multivariate rational functions of degree 4[5]. These cryptosystems are constructed via composition of two quadratic rational maps. In this paper, we present the cryptanalysis of this family of cryptosystems. The key point of our attack is to transform a problem of decomposition of two rational maps into a problem of decomposition of two polynomial maps. We develop a new improved 2R decomposition method and other new techniques, which allows us to find an equivalent decomposition of the rational maps to break the system completely. For the example suggested for practical applications, it is extremely fast to perform the computation to derive an equivalent private key, and it requires only a few seconds on a standard PC.

Key Words: multivariate public key cryptosystems, rational polynomials, map decomposition

1 Introduction

Multivariate public key cryptosystems have undergone very fast development in the last 20 years. They are considered one of the promising families of alternatives for post-quantum cryptography, which are cryptosytems that could resist attacks by the quantum computers of the future [1]. Though most people think that Diffie and Fell wrote the first paper on the multivariate public key cryptosystems[3], Tsujii, Kurosawa and etc actually did similar work at the same time[7]. Though this family of cryptosystems is almost 20 years old, it is not so well known. It actually included several methods rediscovered later, which is partially due to the fact that they were written in Japanese and were published inside Japan. Recently it is pointed out by Tsujii[6] that there is not yet any successful attack on the degree 4 rational multivariate public key cryptosystem designed at that time (1989)[5].

This family of multivariate public key cryptosystem is very different from most of the known cryptosystems, namely the public key functions are rational functions instead of polynomial functions and the total degree of the polynomials components are of degree 4 instead of degree 2. The public key can be presented as:

$$P(x_1, .., x_n) = (P_1(x_1, .., x_n)/P_{n+1}(x_1, .., x_n), \cdots, P_n(x_1, .., x_n)/P_{n+1}(x_1, .., x_n)),$$

where $P_i(x_1, .., x_n)$ are degree 4 polynomials over a finite field k. We call this family of cryptosystems rational multivariate public key cryptosystems (RMPKCs).

The construction of this family of cryptosystems relies on three basic methods. The first one is called the core transformation, which is essentially an invertible rational map with two variables.

The second one is called the sequential solution method, which is essentially invertible rational triangular maps. This ideas was used later in the name of tractable rational maps in [8], but the authors [8] were not aware of the work of Tsujii's group. The last one is the method of composition of nonlinear maps, which was also used later by Goubin and Patarin [4] again without knowing the works of Tsujii's group. The public key therefore has following expression:

$$P = L_3 \circ G \circ L_2 \circ F \circ L_1,$$

where $\circ$ stands for map composition and L_i are invertible affine maps. G and F are degree two rational maps:

$$F = (F_1/F_{n+1}, \cdots, F_n/F_{n+1};) \;\; G = (G_1/G_{n+1}, \cdots, G_n/G_{n+1}),$$

where F_i and G_i are quadratic polynomials and F and G utilize both the core transformation and the triangular method.

The designers of this family of cryptosystem also employed two very interesting ideas to reduce the public key size, which is a key constraint with the potential to render a multivariate public key cryptosystem application less efficient. The first idea is to use functions of a small number of variables over a relatively large field. Since the the public key size is $\mathcal{O}(n^4)$, using fewer variables greatly reduces the public key size.

The second idea is to build a public key using a field k, then use an extension field of k, say K, as the field from which the plaintext is defined. If $|k|^e = |K|$, then the public key size required is only $\frac{1}{e}$ as large as if K were used to define the public key. Mathematically, the public key lies in the function ring over k^n, a subring of the function ring over K^n. Encryption and decryption occur using the larger function ring. This idea was used later in Sflash Version-1[10].

In 1989, the designers proposed a practical application using k of size 2^8, K of size 2^{32} and $n = 5$. This application encrypts blocks of 20 bytes using a 756 byte public key. This family of cryptosystems seems to be very interesting and worthy of further exploration.

As we mentioned before, there is a related cryptosystem called 2R by Patarin, which is very similar except that F and G are replaced by 2 quadratic polynomial maps, but this cryptosystem is broken by a decomposition method using partial derivatives[9]. It is clear this method cannot be directly used on RMPKCs because of more complicated expressions for derivatives of rational functions.

Our new method begins by viewing separately the denominator and the numerators of the public key as polynomial functions. We would like to decompose these quartic polynomials into quadratic components. We will use these quadratics to reconstruct the given public key polynomials, but we first have to transform them so that the reconstruction is done is a way that we have a complete alternate private key for the cryptosystem. This alternate private key gives us the ability to invert ciphertext just as easily as the owner of the original private key.

To see how we accomplish this, let's refer to the polynomial expressions in the denominator and the numerators of the public key as $p_i = g_i \circ (f_1, \ldots, f_{n+1})$. We first find $\mathcal{S} = \mathit{Span}\ \{\ f_j : 1 \leq j \leq n+1\ \}$. From $\mathcal{S}$, we carefully choose a basis that will enable us to invert the resulting rational maps when we reconstruct the public key. After choosing this basis, it is easy to find each g_i. We will have to transform in a similar way the components of $\mathit{Span}\ \{\ g_j : 1 \leq j \leq n+1\ \}$.

We would like to emphasize that our attack is not just application of known methods. In particular, the design of these RMPKCs create two especially interesting challenges for us. The first challenge is to find $\mathit{Span}\ \{\ f_j : 1 \leq j \leq n+1\ \}$, and it turns out that the 2R decomposition method alone can not fiund this space by just applying the partial derivative attack directly to the quartic polynomials p_i. Mathematically, our new idea is to use subplanes of our function space, and the computational means that to do this is very simple: we merely set some of the variables equal to zero. By combining results from three or more of such subplanes, we successfully identify

Span $\{ f_j : 1 \le j \le n+1 \}$. This new extension of 2R decompostion is very different from that in[2].

The second challenge comes from the use of a common denominator in both F and G. We must identify each of these two denominators exactly (up to a scaling factor). This step is necessary to complete the reconstruction of the public key. To find the exact denominator of F, we capitalize on a weakness in the design of the core transformation of G. This weakness results in a portion (subspace) of *Span* $\{ p_j : 1 \le j \le n+1 \}$ in which the polynomial elements have the denominator of F as a factor. We find it using linear algebra techniques. Finding the exact denominator of G comes to us automatically as we solve for the g_i's in the equations $p_i = g_i \circ (f_1, \dots, f_{n+1})$.

The paper is arranged as follows. In Section 2, we will present the specifics of the cryptosystems we will attack. In Section 3, we will present the details of the cryptanalysis of this family of cryptosystems; we will include our experimental results and relevant information on computational complexity. In the last section, we will summarize our learnings.

2 The RMPKC Cryptosystem

In this section, we will present the design of the rational multivariate public key cryptosystem[5].

Let k be a finite field and k^n the n-dimensional vector space over k.

1. **The public key**

   The public key is given as a set of rational degree 4 functions:

   $$P(x_1, ...x_n) = (P_1(x_1, \dots, x_n)/P_{n+1}(x_1, \dots, x_n), \cdots, P_n(x_1, \dots, x_n)/P_{n+1}(x_1, \dots, x_n)),$$

   where each P_i is a degree 4 polynomial over k. P is constructed as the composition of the five maps:

   $$P = L_3 \circ G \circ L_2 \circ F \circ L_1 = (P_1/P_{n+1}, \cdots, P_n/P_{n+1}).$$

   Here L_1, L_2, L_3 are invertible, linear transformations over k^n. Both F and G are quadratic rational maps, i.e. each consists of n quadratic rational functions, $k^n \to k$.

   $F = (F_1/F_{n+1}, \cdots, F_n/F_{n+1})$ and $G = (G_1/G_{n+1}, \cdots, G_n/G_{n+1})$,, where for $1 \le i \le n+1$, F_i and G_i are quadratic polynomials in $(x_1, \dots, x_n)$. The details of the construction of F and G are provided below in the section explaining the private key. F and G are constructed identically, with different choices of random parameters.

   Note the denominators used in both rational maps are the same in the two nonlinear map respectively. G_{n+1} is the common denominator for G; it enables the public key to consist of exactly $n+1$ polynomials. F_{n+1} is the common denominator for F; it enables the composition of degree 2 rational functions to result in a degree 4 rational function, rather than that of higher degree.

   To see how this works, we'll introduce a division function, $\phi : k^{n+1} \longrightarrow k^n$ with $\phi(x_1, \dots, x_{n+1}) = (\frac{x_1}{x_{n+1}}, \cdots, \frac{x_n}{x_{n+1}})$. Also let $\bar{F}, \bar{G} : k^n \longrightarrow k^{n+1}$ each be quadratic polynomials that satisfy

   $$\phi \circ \bar{G} \ = \ L_3 \circ G \ \text{ and } \ \phi \circ \bar{F} \ = \ L_2 \circ F \circ L_1$$

   resulting in

   $$P = \phi \circ \bar{G} \circ \phi \circ \bar{F} \ = \ \phi \circ (\bar{G} \circ \phi) \circ \bar{F}.$$

   Now let $\tilde{G}$ be the homogenization of $\bar{G}$, i.e. $\tilde{G} : k^{n+1} \to k^{n+1}$ where

   $$\forall \ 1 \le i \le n+1, \tilde{G}_i(v_1, \dots, v_{n+1}) = v_{n+1}^2 \bar{G}_i(\tfrac{v_1}{v_{n+1}}, \cdots, \tfrac{v_n}{v_{n+1}}) = v_{n+1}^2 \bar{G}_i \circ \phi(v_1, \dots, v_{n+1}).$$

Note that $\tilde{G} \neq \bar{G} \circ \phi$, but $\phi \circ \tilde{G} = \phi \circ \bar{G} \circ \phi$. So $P = \phi \circ \tilde{G} \circ \bar{F}$ where $\tilde{G}$ and $\bar{F}$ are quadratic polynomials. The public key, then, contains the ordered list of n+1 quartic polynomials $(P_1, \ldots, P_{n+1})$ where $\forall\ 1 \leq i \leq n+1,\ P_i(x_1, \ldots, x_n) = \tilde{G}_i \circ \bar{F}(x_1, \ldots, x_n)$.

2. **Encryption**

   Given a plaintext $X = (X'_1, \cdots, X'_n) \in k^n$ one computes the ciphertext $Y' = (Y'_1, \cdots, Y'_n) \in k^n$ as

$$(Y'_1, \cdots, Y'_n) = (P_1(X'_1, \ldots, X'_n)/P_{n+1}(X'_1, \ldots, X'_n), \cdots, P_n(X'_1, \ldots, X'_n)/P_{n+1}(X'_1, \ldots, X'_n)).$$

3. **The private key**

   The private key is the set of the five maps F, G, L_1, L_2, L_3 and the key to invert the non-linear maps F and G. The map P can illustrated as:

$$k^n \xrightarrow{L_1} k^n \xrightarrow{F} k^n \xrightarrow{L_2} k^n \xrightarrow{G} k^n \xrightarrow{L_3} k^n.$$

The design principles of the quadratic rational components, F and G, are identical, except that they use different choices for the random parameters involved. A two-part construction is used. The first part is what the designers call a core transformation. The second part is called the sequential part, since inversion is accomplished sequentially. Its structure can be seen as triangular.

The core tranformation is applied only to the last two components, namely $C = (\frac{F_{n-1}}{F_{n+1}}, \frac{F_n}{F_{n+1}})$, which can be viewed as a map $k^2 \longrightarrow k^2$. To construct F_{n-1}, F_n, F_{n+1}, we first randomly choose 12 elements in k: $\alpha_1, \ldots, \alpha_6$ and $\beta_1, \ldots, \beta_6$. C has an inverse which is given by:

$$C^{-1}(y_{n-1}, y_n) = (\ \frac{\alpha_1 y_{n-1} + \alpha_2 y_n + \alpha_3}{\alpha_4 y_{n-1} + \alpha_5 y_n + \alpha_6},\ \frac{\beta_1 y_{n-1} + \beta_2 y_n + \beta_3}{\beta_4 y_{n-1} + \beta_5 y_n + \beta_6}\ ).$$

Then F_{n-1}, F_n and F_{n+1} are defined as follows:

$$\forall\ n-1 \leq i \leq n+1,\quad F_i(x_{n-1}, x_n) = \tau_{i,1} x_{n-1} x_n + \tau_{i,2} x_{n-1} + \tau_{i,3} x_n + \tau_{i,4}$$

where the $\tau_{i,j}$ is defined as follows:

$$\begin{array}{lll}
\tau_{n-1,1} = \alpha_6\beta_5 - \alpha_5\beta_6 & \tau_{n,1} = \alpha_6\beta_4 - \alpha_4\beta_6 & \tau_{n+1,1} = \alpha_5\beta_4 - \alpha_4\beta_5 \\
\tau_{n-1,2} = \alpha_3\beta_5 - \alpha_5\beta_3 & \tau_{n,2} = \alpha_3\beta_4 - \alpha_4\beta_3 & \tau_{n+1,2} = \alpha_1\beta_4 - \alpha_4\beta_1 \\
\tau_{n-1,3} = \alpha_6\beta_2 - \alpha_2\beta_6 & \tau_{n,3} = \alpha_6\beta_1 - \alpha_1\beta_6 & \tau_{n+1,3} = \alpha_5\beta_2 - \alpha_2\beta_5 \\
\tau_{n-1,4} = \alpha_3\beta_2 - \alpha_2\beta_3 & \tau_{n,4} = \alpha_3\beta_1 - \alpha_1\beta_3 & \tau_{n+1,4} = \alpha_1\beta_2 - \alpha_2\beta_1
\end{array}$$

The rest of the components are given in a triangular form:

$$\forall 1 \leq i \leq n-2,\ F_i(x_1, \ldots, x_n) = a_i(x_{i+1}, \ldots, x_n) x_i\ +\ b_i((x_{i+1}, \ldots, x_n),$$

where the a_i's are randomly chosen linear polynomials and the b_i's are randomly chosen quadratic polynomials.

4. **Decryption**

   To decrypt, we need to invert the map P, which is done as follows:

$$P^{-1}(Y'_1, \dots, Y'_n) = L_1^{-1} \circ F^{-1} \circ L_2^{-1} \circ G^{-1} \circ L_3^{-1}(Y'_1, \dots, Y'_n) = (X'_1, \dots, X'_n).$$

   The holder of the private key has the means to find the inverse of each of L_3, G, L_2, F, L_1. Performing the calculations in order yields $(X'_1, \dots, X'_n)$. Inversion of the linear transformations is obvious.

   To invert the map F is to find the solution of equation: $F(x_1, ..., x_n) = (y'_1, ..., y'_n)$ for a given vector $(y'_1, ..., y'_n)$. We first use the inverse of C to calculate $(x'_{n-1}, x'_n) = C^{-1}(y'_{n-1}, y'_n)$. Then we plug the resulting values into the third last component function of F. This gives us the following linear equation in x_{n-2}:

$$y'_{n-2} = \frac{F_{n-2}(x_{n-2}, x'_{n-1}, x'_n)}{F_{n+1}(x'_{n-1}, x'_n)} = \frac{a_{n-2}(x'_{n-1}, x'_n) * x_{n-2} + b_{n-2}(x'_{n-1}, x'_n)}{\tau_{n-2,1} x'_{n-1} x'_n + \tau_{n-2,2} x'_{n-1} + \tau_{n-2,3} x'_n + \tau_{n-2,4}}$$

   yielding

$$x'_{n-2} = \frac{y'_{n-2} * (\tau_{n-2,1} x'_{n-1} x'_n + \tau_{n-2,2} x'_{n-1} + \tau_{n-2,3} x'_n + \tau_{n-2,4}) - b_{n-2}(x'_{n-1}, x'_n)}{a_{n-2}(x'_{n-1}, x'_n)}.$$

   After obtaining x'_{n-2}, we can plug known values into the fourth last component function of F and derive x'_{n-3}. This sequential solution method is continued to find the rest of $(x'_1, \dots, x'_n)$ which gives us a solution for $F(x_1, ..., x_n) = (y'_1, ..., y'_n)$.

   Inversion of G is performed in the exact same manner as F.

   Note that in the inversion process, division is required in the calculation of each of the components of $(x'_1, \dots, x'_n)$. In each case, the expression for the divisor is linear in terms of known values of input variables $(x'_{i+1}, \dots, x'_n)$ and the given values of output variables $(y'_i, \dots, y'_n)$. In both cases, the probability of valid division is approximately $\frac{q-1}{q}$. The probability of successfully inverting both F and G, and thus P, therefore, is approximately $(\frac{q-1}{q})^{2n}$.

3 Cryptanalysis of RMPKC

Our attack can be viewed as the decomposition of maps. The cryptanalysis of RMPKC is performed as follows: given P, the composition of $L_3 \circ G \circ L_2 \circ F \circ L_1$, generate a new set of maps L'_3, G', L'_2, F', and L'_1 such that

$$L_3 \circ G \circ L_2 \circ F \circ L_1 = L'_3 \circ G' \circ L'_2 \circ F' \circ L'_1,$$

and G' and F' can be inverted in the same way as G and F, with the keys to inversion obtained during the process. This new set of maps can be viewed as a private key equivalent to the original one, thus can be used to defeat the RMPKC cryptosystem.

To decompose RMPKC, we will use the partial derivative method, which takes the composition of two homogeneous quadratic polynomial maps forming a homogeneous quartic map, and decomposes it into quadratic maps which, when composed together, form the original quartic map[9]. Consider $g \circ f$ where $g = (\ (g_1(x_1, \dots, x_m), \cdots, g_m(x_1, \dots, x_m)\ ),\ f = (\ (f_1(x_1, \dots, x_m), \cdots, f_m(x_1, \dots, x_m)\ )$

and each of the g_i's and the f_i's are homogeneous quadratic polynomials. The first step is to find $\mathcal{F} = Span\ \{\ f_i : 1 \leq i \leq m\ \}$, a vector space over k.

Once found, one can select linearly independent quadratics from it, say $(f'_1, \ldots, f'_m)$. Then by solving a set of linear equations, one can find $(g'_1, \ldots, g'_m)$ such that $\forall\ 1 \leq i \leq m,\ g'_i \circ f' = g_i \circ f$ where $f' = (f'_1, \ldots, f'_m)$.

The critical step of this process is finding $\mathcal{F}$. The following definitions are needed:

$$D = Span\ \{\ \frac{\partial}{\partial x_j} g_i \circ f(x_1, \ldots, x_m) : 1 \leq i, j \leq m\ \}$$

$$\Lambda =\ \{\ x_j f : 1 \leq j \leq m, f \in \mathcal{F}\ \}.$$

$$R = \{\ \theta : \forall\ 1 \leq i \leq m,\ x_i \theta \in D\ \}.$$

When each of the f_i's and g_i's are homogeneous quadratic polynomials, $D \subseteq \Lambda$. This is true basically because

$$\tfrac{\partial}{\partial x_j}(g_i \circ f) = \sum_{r=1}^{m} \tfrac{\partial}{\partial w_r} g_i(f) \times \tfrac{\partial}{\partial x_j} f_r(x_1, \ldots, x_m)$$

where $\frac{\partial}{\partial w_r} g_i(f)$ is linear in the f's and $\frac{\partial}{\partial x_j} f(x_1, \ldots, x_m)$ is linear in the $(x_1, \ldots, x_m)$.

We calculate D and R from $g \circ f$. If $D = \Lambda$, then $R = \mathcal{F}$ and this step is complete. When $D \subset \Lambda$, $R \subset \mathcal{F}$. Why $R \subseteq \mathcal{F}$ and $D = \Lambda \iff R = \mathcal{F}$ should be fairly easy to see.

Application of the partial derivative attack to RMPKC requires some additional work. As we saw in the explanation of the public key, we have access to $n+1$ polynomials of the form $P_i = \tilde{G}_i \circ \bar{F}(x_1, \ldots, x_n)$ where $\tilde{G}_i$ is a homogeneous quadratic polynomial and $\bar{F}$ consists of non-homogeneous quadratic polynomials. Our first step is to homogenize each of the P_i's, which effectively homogenizes each of the $\bar{F}_i$'s, yielding the following:

$$\tilde{P}_i(x_1, \ldots, x_{n+1}) = \tilde{G}_i \circ \tilde{F}(x_1, \ldots, x_{n+1})$$

where each of the $\tilde{P}_i$'s are homogeneous quartic polynomials and each of the $\tilde{G}_i$'s and $\tilde{F}_i$'s are homogeneous quadratic polynomials.

Then we begin the partial derivative attack, by calculating D from $\tilde{G}_i \circ \tilde{F}(x_1, \ldots, x_{n+1})$. We never get $D = \Lambda$, due to the triangular structure of G and the use of k which has characteristic 2. We are able to recover $\mathcal{F}$ by applying the attack with a new method of projection of our functions to subplanes; the details will be provided in the section that follows. After finding $\mathcal{F}$, we de-homogenize the space by setting $x_{n+1} = 1$.

The second challenge that the specifics of RMPKC present to the partial derivative attack is the challenge to select the polynomials $F'_1, \ldots, F'_{n+1}$ from $\mathcal{F}|_{x_{n+1}=1}$ in such a way that they may be easily inverted. The procedure we use to find such $F'_1, \ldots, F'_{n+1}$ is described below. The process results in a linear transformation L'_1 and a quadratic rational map F', which inverts in the same manner as F for the holder of the private key.

Then to continue the partial derivative attack we can find the g_i's that satisfy $P_i = g_i \circ F'$; but these g_i's would not invert easily. So we define $\mathcal{G}' = Span\ \{\ g_i : 1 \leq i \leq n+1\ \}$ and select polynomials from $\mathcal{G}'$ which we can invert. This process generates linear transformations L'_2 and L'_3, and quadratic rational map G', which inverts in the same manner as G in the private key. Then we have $P = L'_3 \circ G' \circ L'_2 \circ F' \circ L'_1$, an alternative private key, thus breaking the RMPKC.

We organize our attack into four phases. The sections that follow will present an explanation in further detail of each phase.

1. Find $\mathcal{F} = Span\ \{\ \tilde{F}_i\ : 1 \leq i \leq n+1\ \}$.

2. Determine F' and L'_1.
3. Find $\mathcal{G}' = Span\ \{\ g'_i \mid g'_i \circ F' \circ L'_1 = P_i : 1 \leq i \leq n+1\ \}$.
4. Determine G', L'_2, and L'_3.

3.1 Phase I: Find $\mathcal{F} = Span\ \{\ \tilde{F}_i\ : 1 \leq i \leq n+1\ \}$

We start with the public key, $P = \tilde{G} \circ \bar{F} = (P_1, \ldots, P_{n+1})$ and homogenize by creating $\tilde{P} = (\tilde{P}_1, \ldots, \tilde{P}_{n+1})$ using $\forall\ 1 \leq i \leq n+1,\ \tilde{P}_i(x_1, \ldots, x_{n+1}) = x_{n+1}^4 P_i(\frac{x_1}{x_{n+1}}, \cdots, \frac{x_n}{x_{n+1}})$. This gives us $\tilde{P} = \tilde{G} \circ \tilde{F}$ where $\tilde{F} = (\tilde{F}_1, \ldots, \tilde{F}_{n+1})$ and $\forall\ 1 \leq i \leq n+1,\ \tilde{F}_i(x_1, \ldots, x_{n+1}) = x_{n+1}^2 \bar{F}_i(\frac{x_1}{x_{n+1}}, \cdots, \frac{x_n}{x_{n+1}})$.

To proceed we need to define $H_i\ \forall\ i \in\ \{\ 1, 2, 3\ \}$ as the set of all homogeneous polynomials in $k[x_1, \ldots, x_{n+1}]$ of degree i. Each H_i is a vector space over k as well as a subset of $k[x_1, \ldots, x_{n+1}]$. For notational simplification, we will use context to distinguish between these uses of H_i.

We now define D, R, and Λ for $\tilde{G} \circ \tilde{F}$. Recall that we calculate D and R from $\tilde{P}$.

$$D = Span\ \{\ \tfrac{\partial}{\partial x_j} \tilde{G}_i \circ \tilde{F}(x_1, \ldots, x_{n+1}) :\ 1 \leq i, j \leq n+1\ \} \subset \mathcal{H}_3$$

$$\Lambda =\ \{\ x_j f : 1 \leq j \leq n+1, f \in \mathcal{F}\ \} \subset \mathcal{H}_3$$

$$R =\ \{\ f \in \mathcal{H}_2 : \forall 1 \leq i \leq n+1,\ x_i f \in D\ \}.$$

Since the polynomials of $\tilde{G}$ and $\tilde{F}$ are homogeneous quadratics, we are guaranteed $D \subseteq \Lambda$ and $R \subseteq \mathcal{F}$. We also have $D = \Lambda \iff R = \mathcal{F}$. Because of the structure of the original polynomials in G and the use of a field of characteristic 2, we will always find $D \subset \Lambda$ and therefore $R \subset \mathcal{F}$. So we use the following definitions of Γ and γ to help explain how to see what is happening with individual f's in $\mathcal{F}$, why they do not find themselves in R, and how we are going to eventually find them with our alternative approach.

$$\Gamma(f) =\ \{\ \theta \in \mathcal{H}_1 : \theta f \in D\ \}\ \text{ and } \gamma(f) = dim(\ \Gamma(f)\ ).$$

Clearly, $f \in R \iff \gamma(f) = n+1$. We always get $\gamma(f) \leq n+1$, and $Min\ \{\ \gamma(f) : f \in \mathcal{F}\ \}$ describes how far away from obtaining $R = \mathcal{F}$ for any given application of RMPKC. For $n = 5$ and $n = 6$, we find $Min\ \{\ \gamma(f) : f \in \mathcal{F}\ \}\ = n$ almost every time. For $n = 7$ we usually get $Min\ \{\ \gamma(f) : f \in \mathcal{F}\ \}\ = n-1$. And for $n \geq 8$ we most likely get $Min\ \{\ \gamma(f) : f \in \mathcal{F}\ \}\ = n-2$. Our alternative approach works most simply for $Min\ \{\ \gamma(f) : f \in \mathcal{F}\ \}\ = n$. We will describe this now in detail; then briefly show how we accomplish this for $Min\ \{\ \gamma(f) : f \in \mathcal{F}\ \}\ < n$. We again start with the key definitions, valid $\forall\ 1 \leq s \leq n+1$; and we have access to each D_s and R_s.

$$\mathcal{F}_s = Span\ \{\ f(x_1, \ldots, x_{s-1}, 0, x_{s+1}, \ldots, x_{n+1}) : \forall\ f \in \mathcal{F}\ \}\ .$$

$$D_s = Span\ \{\ \frac{\partial}{\partial x_j} \tilde{G}_i \circ \tilde{F}(x_1, \ldots, x_{s-1}, 0, x_{s+1}, \ldots, x_{n+1}) :\ 1 \leq i, j \leq n+1\ \}\ .$$

$$\Lambda_s =\ \{\ x_i f : 1 \leq i \leq n+1 (i \neq s), f \in \mathcal{F}_s\ \}\ .$$

$$R_s =\ \{\ f \in \mathcal{H}_2 : \forall\ 1 \leq i \leq n+1 (i \neq s),\ x_i f \in D_s\ \}\ .$$

$$\Gamma_s(f) = \{\ \theta \in \mathcal{H}_1 : \theta f \in D_s\ \}\ , \qquad \gamma_s(f) = dim(\ \Gamma_s(f)\ ).$$

Now we always get $D_s \subseteq \Lambda_s$, $R_s \subseteq \mathcal{F}_s$, and $D_s = \Lambda_s \iff R_s = \mathcal{F}_s \iff Min \; \{ \; \gamma_s(f) : f \in \mathcal{F}_s \; \} \; = n$. Fortunately for this attack, with high probability, $\gamma_s(f) = Min \; \{ \; \gamma(f), n \; \}$. This is a crucial point. At this time, we do not have a mathematical explanation for why it is so; our experiments confirm it with consistent results. Once we get $\forall \; 1 \leq s \leq n+1, R_s = \mathcal{F}_s$, finding $\mathcal{F}$ is easy.

Let $R_s^+ = R_s + Span \; \{ \; x_s x_i : 1 \leq i \leq n+1 \; \}$. When $R_s = \mathcal{F}_s$, $\mathcal{F} \subset R_s^+$. Furthermore, if $\forall \; 1 \leq s \leq n+1, R_s = \mathcal{F}_s$, then $\mathcal{F} = \bigcap_{s=1}^{n+1} R_s^+$, completing the task of finding $\mathcal{F}$.

For the cases of $Min \; \{ \; \gamma(f) : f \in \mathcal{F} \; \} \; < n$, we expand our alternative approach one or more levels further. Notice above the spaces R_s^+, which are created by setting $x_s = 0$, finding D_s and R_s, then adding $Span \; \{ \; x_s x_i : 1 \leq i \leq n+1 \; \}$. For $n = 7$, when we have $Min \; \{ \; \gamma(f) : f \in \mathcal{F} \; \} \; = n-1$, we use $x_{s_1} = 0 = x_{s_2}$ where $s_1 \neq s_2$. Following the same manner we form D_{s_1,s_2} and R_{s_1,s_2}. Then we let $R_{s_1,s_2}^+ = R_{s_1,s_2} + Span \; \{ \; x_{s_1} x_i : 1 \leq i \leq n+1 \; \} \; + Span \; \{ \; x_{s_2} x_i : 1 \leq i \leq n+1 \; \}$. With consistency, we do get $\mathcal{F} = \bigcap_{\substack{1 \leq s_1, s_2 \leq n+1 \\ s_1 \neq s_2}} R_{s_1,s_2}^+$.

For $n \geq 8$, when we have $Min \; \{ \; \gamma(f) : f \in \mathcal{F} \; \} \; = n-2$, we use $x_{s_1} = 0 = x_{s_2} = 0 = x_{s_3}$ where $s_1 \neq s_2 \neq s_3 \neq s_1$. Following the same manner we form D_{s_1,s_2,s_3} and R_{s_1,s_2,s_3}. Then we let $R_{s_1,s_2,s_3}^+ = R_{s_1,s_2,s_3} + Span \; \{ \; x_{s_1} x_i : 1 \leq i \leq n+1 \; \} + Span \; \{ \; x_{s_2} x_i : 1 \leq i \leq n+1 \; \} + Span \; \{ \; x_{s_3} x_i : 1 \leq i \leq n+1 \; \}$. Again we consistently get $\mathcal{F} = \bigcap_{\substack{1 \leq s_1, s_2, s_3 \leq n+1 \\ s_1 \neq s_2 \neq s_3 \neq s_1}} R_{s_1,s_2,s_3}^+$.

3.2 Phase II: Choose $\mathbf{F'}$ and $\mathbf{L_1'}$

In this phase we will determine the quadratic polynomials of $F' = (F_1'/F_{n+1}', \cdots, F_n'/F_{n+1}',$; and the linear transformation, L_1' such that

$$Span \; \{ \; F_i' \circ L_1' : 1 \leq i \leq n+1 \; \} \; = Span \; \{ \; F_i \circ L_1 : 1 \leq i \leq n+1 \; \} \; ,$$

and F' can be easily inverted just like F.

However, we do need one additional condition on our new map, namely we must have

$$F_{n+1}' \circ L_1' = \lambda F_{n+1} \circ L_1$$

for some $\lambda \in k$. This is necessary in order to find the proper G', which will be determined later, to be chosen so that it too can be inverted in the same manner as G.

Our first step is to determine a core transformation in F'. From the definition in Section 2, we can see that there is a subspace spanned by two linearly independent linear functions in $\mathcal{F}$, which actually lies in the space spanned by F_{n-1}, F_n, F_{n+1}. Therefore F' also contains a subspace that is contained in $Span \; \{ \; \theta_{n-1}', \theta_n', 1 \; \}$ for some $\theta_{n-1}', \theta_n' \in \mathcal{H}_1$. This space can be found easily, and it is clear that we have $Span \; \{ \; \theta_{n-1}', \theta_n' \; \} = Span \; \{ \; L_{1,n-1}, L_{1,n} \; \}$,where $L_{1,n-1}$ and $L_{1,n}$ are the last two components of the linear transformation L_1. Next we find the three-dimensional subspace of $\mathcal{F}$ which forms the core transformation, i.e. let $\mathcal{R} = \mathcal{F} \cap Span \; \{ \; {\theta_{n-1}'}^2, {\theta_n'}^2, \theta_{n-1}'\theta_n', \theta_{n-1}', \theta_n', 1 \; \}$.

By construction, we know not only that $\exists \; R_1, R_2, R_3 \in \mathcal{R}$ such that $\mathcal{R} = Span \; \{ \; R_1, R_2, R_3 \; \}$ and $R_1, R_2 \in Span \; \{ \; \theta_{n-1}', \theta_n', 1 \; \}$ and $R_3 \in Span \; \{ \; {\theta_{n-1}}^2, {\theta_n}^2, \theta_{n-1}\theta_n, 1 \; \}$, but also that $\exists \; \theta_{n-1}, \theta_n \in Span \; \{ \; \theta_{n-1}', \theta_n' \; \}$ where $R_1, R_2 \in Span \; \{ \; \theta_{n-1}, \theta_n, 1 \; \}$ and $R_3 \in Span \; \{ \; \theta_{n-1}\theta_n, 1 \; \}$. Furthermore, R_3 can be chosen so that $R_3 = {\theta_{n-1}'}^2 + a\theta_{n-1}'\theta_n' + b{\theta_n'}^2 + c$. We can find appropriately $\theta_{n-1} = \theta_{n-1}' + s\theta_n'$ and $\theta_n = \theta_{n-1}' + t\theta_n'$ by finding the right values for s and t.

We solve for s and t by equating the quadratic terms of our chosen R_3, i.e. ${\theta'_{n-1}}^2 + a\theta'_{n-1}\theta'_n + b{\theta'_n}^2 = (\theta'_{n-1} + s\theta'_n)(\theta'_{n-1} + t\theta'_n)$. So $s + t = a$ and $st = b$. Thus $s(a - s) = b$, i.e. $s^2 - as + b = 0$. In characteristic 2, this last equation is actually linear and can be solved for s.

This choice of θ_i allows us to calculate an inversion function for the core transformation (described below), just like the inversion function of F. Coincidently, either $\theta_{n-1} = \lambda_1 L_{1,n-1}$ and $\theta_n = \lambda_2 L_{1,n}$ for some $\lambda_1, \lambda_2 \in k$ or $\theta_{n-1} = \lambda_1 L_{1,n}$ and $\theta_n = \lambda_2 L_{1,n-1}$ for some $\lambda_1, \lambda_2 \in k$; but we don't care which nor do we use this result directly.

To get $F'_{n+1} \circ L'_1 = \lambda F_{n+1} \circ L_1$ for some $\lambda \in k$, we choose $f_{n+1} \in \mathcal{R}$ such that $f_{n+1} | \rho$ for some nonzero $\rho \in \mathcal{P} = Span\ \{\ P_i : 1 \leq i \leq n + 1\ \}$. This works to identify $f_{n+1} = \lambda F_{n+1} \circ L_1$ for some $\lambda \in k$ because the quadratic polynomials of G become homogeneous when composed with the rational functions in F, making the linear subspace of the polynomials of G become a subspace divisible by $F_{n+1} \circ L_1$ (the denominator) when composed with $L_2 \circ F \circ L_1$.

We randomly choose $f_{n-1}, f_n \in \mathcal{R}$ such that $\mathcal{R} = Span\ \{\ f_i : n - 1 \leq i \leq n + 1\ \}$.

We then determine $f_1, \dots, f_{n-2}$ and $\theta_1, \dots, \theta_{n-2}$ sequentially, by first choosing f_{n-2} and θ_{n-2}, then working our way to f_1 and θ_1. Our procedure is as follows:

$\forall\ i = (n-2, n-3, \cdots, 2)$ find $\theta_i \notin Span\ \{\ \theta_{i+1}, \dots, \theta_n\ \}$ and $f_i \in \mathcal{F}$ such that $f_i \in Span\ \{\ \theta_j\theta_k : {}_{\substack{i \leq j \leq k \leq n+1 \\ k \neq i}}\ \}\ + Span\ \{\ \theta_j : i \leq j \leq n + 1\ \}\ + 1$.

The last components, f_1 and θ_1, can be chosen randomly as long as $Span\ \{\ f_i : 1 \leq i \leq n{+}1\ \}\ = \mathcal{F}$ and $Span\ \{\ \theta_i : 1 \leq i \leq n + 1\ \}\ = Span\ \{\ x_i : 1 \leq i \leq n\ \}$.

$\theta_1, \dots, \theta_n$ are the components of L'_1. It is easy to calculate $F_1, \dots, F_{n+1}$ such that $\forall\ 1 \leq i \leq n + 1, f_i = F_i \circ L'_1$.

Now that we have determined L_1 and F', we can find the inversion function parameters ($\alpha'_1, \dots, \alpha'_6, \beta'_1, \dots, \beta'_6$) for the core transformation of F' by considering

$$x_{n-1} = \frac{\alpha'_1 \frac{F'_{n-1}(x_{n-1},x_n)}{F'_{n+1}(x_{n-1},x_n)}\ + \alpha'_2 \frac{F'_n(x_{n-1},x_n)}{F'_{n+1}(x_{n-1},x_n)}\ + \alpha'_3}{\alpha'_4 \frac{F'_{n-1}(x_{n-1},x_n)}{F'_{n+1}(x_{n-1},x_n)}\ + \alpha'_5 \frac{F'_n(x_{n-1},x_n)}{F'_{n+1}(x_{n-1},x_n)}\ + \alpha'_6} = \frac{\alpha'_1 F'_{n-1}(x_{n-1},x_n)\ + \alpha'_2 F'_n(x_{n-1},x_n)\ + \alpha'_3 F'_{n+1}(x_{n-1},x_n)}{\alpha'_4 F'_{n-1}(x_{n-1},x_n)\ + \alpha'_5 F'_n(x_{n-1},x_n)\ + \alpha'_6 F'_{n+1}(x_{n-1},x_n)}$$

or equivalently

$$\begin{gathered} x_{n-1}(\alpha'_4 F'_{n-1}(x_{n-1}, x_n)\ + \alpha'_5 F'_n(x_{n-1}, x_n)\ + \alpha'_6 F'_{n+1}(x_{n-1}, x_n)) = \\ \alpha'_1 F'_{n-1}(x_{n-1}, x_n)\ + \alpha'_2 F'_n(x_{n-1}, x_n)\ + \alpha'_3 F'_{n+1}(x_{n-1}, x_n) \end{gathered}$$

We equate the coefficients of the terms $(1, x_{n-1}, x_n, (x_{n-1})^2, x_{n-1}x_n$, and $(x_{n-1})^2 x_n)$ and simultaneously solve for the $\alpha'_1, \dots, \alpha'_6$. In the same manner we find $\beta'_1, \dots, \beta'_6$ by starting with

$$x_n = \frac{\beta'_1 \frac{F'_{n-1}(x_{n-1},x_n)}{F'_{n+1}(x_{n-1},x_n)}\ + \beta'_2 \frac{F'_n(x_{n-1},x_n)}{F'_{n+1}(x_{n-1},x_n)}\ + \beta'_3}{\beta'_4 \frac{F'_{n-1}(x_{n-1},x_n)}{F'_{n+1}(x_{n-1},x_n)}\ + \beta'_5 \frac{F'_n(x_{n-1},x_n)}{F'_{n+1}(x_{n-1},x_n)}\ + \beta'_6} = \frac{\beta'_1 F'_{n-1}(x_{n-1},x_n)\ + \beta'_2 F'_n(x_{n-1},x_n)\ + \beta'_3 F'_{n+1}(x_{n-1},x_n)}{\beta'_4 F'_{n-1}(x_{n-1},x_n)\ + \beta'_5 F'_n(x_{n-1},x_n)\ + \beta'_6 F'_{n+1}(x_{n-1},x_n)}$$

3.2.1 Phase III: Find $\mathcal{G}'$

$\forall\ 1 \leq i \leq n + 1$, find linear combinations of $\{\ (F'_j \circ L'_1)(F'_r \circ L'_1) : 1 \leq j \leq r \leq n + 1\ \}$ which are equal to P_i. The coefficients of these combinations are the coefficients of the homogeneous polynomials $\bar{G}'_i$.

Let $\mathcal{G}' = Span\ \{\ \bar{G}'_i : 1 \leq i \leq n + 1\ \}$.

3.3 Phase IV: Choose $\mathbf{G}', \mathbf{L}_2'$ and $\mathbf{L}_3'$

In this phase we will determine the quadratic polynomials of $G' = \begin{pmatrix} G_1'/G_{n+1}' \\ \vdots \\ G_n'/G_{n+1}' \end{pmatrix}$; and the linear transformations, L_2' and L_3' such that $\forall\ 1 \le i \le n+1, P_i = (L_3')_i \circ G' \circ L_2' \circ F' \circ L_1'$, and G' can be easily inverted just like G.

Our first step is to determine a core transformation in G'. We easily find two linearly independent linear vectors in $\mathcal{G}'$, ϕ_{n-1}' and ϕ_n'. Let $\mathcal{U} = Span\ \{\ \phi_{n-1}', \phi_n'\ \}$. That makes $\mathcal{U} = Span\ \{\ L_{2,n-1}, L_{2,n}\ \}$. Next we find the three-dimensional subspace of $\mathcal{G}'$ which forms the core transformation, i.e. let $\mathcal{V} = \mathcal{G}' \cap Span\ \{\ {\phi_{n-1}'}^2, {\phi_n'}^2, \phi_{n-1}'\phi_n', \phi_{n-1}', \phi_n', 1\ \}$.

Now we find ϕ_{n-1} and ϕ_n in $\mathcal{U}$ such that $\forall\ g \in \mathcal{V}, g \in Span\ \{\ \phi_{n-1}\phi_n, \phi_{n-1}, \phi_n, 1\ \}$. This choice of ϕ's allows us to calculate an inversion function for the core transformation, just like the inversion function of G. Coincidently, either $\phi_{n-1} = \lambda_1 L_{2,n-1}$ and $\phi_n = \lambda_2 L_{2,n}$ for some $\lambda_1, \lambda_2 \in k$ or $\phi_{n-1} = \lambda_1 L_{2,n}$ and $\phi_n = \lambda_2 L_{2,n-1}$ for some $\lambda_1, \lambda_2 \in k$; but we don't care which nor do we use this result directly.

Up to this point, our work with G' has been identical to the work with F'. The method to determine G_{n+1}' is the first place where we differ. G_{n+1}' will be the quadratic polynomial in two variables such that $G_{n+1}'(\phi_{n-1}, \phi_n) = \bar{G}_{n+1}'(x_1, \ldots, x_n, 1)$.

Now we randomly choose $g_{n-1}, g_n \in \mathcal{V}$ such that $\mathcal{V} = Span\ \{\ g_i : n-1 \le i \le n+1\ \}$.

We then determine $g_1, \ldots, g_{n-2}$ and $\phi_1, \ldots, \phi_{n-2}$ sequentially, by first choosing g_{n-2} and ϕ_{n-2}, then working our way to g_1 and ϕ_1. Our procedure is as follows:

$\forall\, i = (n-2, n-3, \cdots, 2)$ find $\phi_i \notin Span\ \{\ \phi_{i+1}, \ldots, \phi_n\ \}$ and $g_i \in \mathcal{G}'$ such that $g_i \in Span\ \{\ \phi_j\phi_k : \underset{k\neq i}{i\le j\le k\le n+1}\ \} + Span\ \{\ \phi_j : i \le j \le n+1\ \} + 1$.

The last components, g_1 and ϕ_1, can be chosen randomly as long as $Span\ \{\ g_i : 1 \le i \le n+1\ \} = \mathcal{G}'$ and $Span\ \{\ \phi_i : 1 \le i \le n+1\ \} = Span\ \{\ x_i : 1 \le i \le n\ \}$.

$\phi_1, \ldots, \phi_n$ are the components of L_2'. And again we must differ in our approach to G' from the approach to F'. At this point, we have for $1 \le i \le n, \bar{G}_i$ is a linear combination of $\{\ g_j : 1 \le j \le n+1\ \}$. We need to have $\forall 1 \le i \le n, \bar{G}_i$ is a linear combination of only $\{\ g_j : 1 \le j \le n\ \}$, (excluding g_{n+1}).

To explain how we accomplish this is best done using $(n+1)$ x $(n+1)$ matrices over k. Let χ be the matrix that represents the linear transformation ($k^{n+1} \longrightarrow k^{n+1}$) such that $\begin{pmatrix} & & \\ & \chi & \\ & & \end{pmatrix} \begin{pmatrix} g_1 \circ L_2' \\ \vdots \\ g_{n+1} \circ L_2' \end{pmatrix} = \begin{pmatrix} \bar{G}_1' \\ \vdots \\ \bar{G}_{n+1}' \end{pmatrix}$. χ is in the form $\begin{pmatrix} * & \cdots & * & * \\ \vdots & \ddots & \vdots & \vdots \\ * & \cdots & * & * \\ 0 & \cdots & 0 & * \end{pmatrix}$ but $\begin{pmatrix} * & \cdots & * & 0 \\ \vdots & \ddots & \vdots & \vdots \\ * & \cdots & * & 0 \\ 0 & \cdots & 0 & * \end{pmatrix}$ is the form which we need.

So we find an invertible upper triangular matrix π and an invertible matrix ν of the desired form such that $\nu\chi = \pi$. The zero entries of π provide linear equations to solve for the entries of ν with coefficients from χ, which are known. Now we have $\chi = \nu^{-1}\pi$. So let $G' = \begin{pmatrix} G_1'/G_{n+1}' \\ \vdots \\ G_n'/G_{n+1}' \end{pmatrix}$ where $\begin{pmatrix} G_1' \\ \vdots \\ G_{n+1}' \end{pmatrix} = \pi \begin{pmatrix} g_1 \\ \vdots \\ g_{n+1} \end{pmatrix}$; and let $L_3' = \nu^{-1}$.

We now have $\begin{pmatrix} \bar{G}'_1 \\ \vdots \\ \bar{G}'_{n+1} \end{pmatrix} = \chi \begin{pmatrix} g_1 \circ L'_2 \\ \vdots \\ g_{n+1} \circ L'_2 \end{pmatrix} = \nu^{-1}\pi \begin{pmatrix} g_1 \circ L'_2 \\ \vdots \\ g_{n+1} \circ L'_2 \end{pmatrix} = L'_3 \begin{pmatrix} G'_1 \circ L'_2 \\ \vdots \\ G'_{n+1} \circ L'_2 \end{pmatrix}$. Furthermore, $P = L'_3 \circ G' \circ L'_2 \circ F' \circ L'_1$ and our decomposition is complete.

We can find the inversion function parameters ($\delta'_1, \ldots, \delta'_6, \gamma'_1, \ldots, \gamma'_6$) for the core transformation of G' in the exact same manner that we found $\alpha'_1, \ldots, \alpha'_6$ and $\beta'_1, \ldots, \beta'_6$ for F'.

In summary, we have created an alternate CQRM cryptosystem using L'_1, F', L'_2, G', and L'_3 such that $L'_3 \circ G' \circ L'_2 \circ F' \circ L'_1 = L_3 \circ G \circ L_2 \circ F \circ L_1$ and both G' and F' are invertible, just like G and F; so cryptanalysis of CQRM is complete.

3.4 Experimental Results and Computational Complexity

The proposal for RMPKC in 1989 suggested an implementation with k of size 2^8 and $n = 5$. Our attack programmed in Magma completes cryptanalysis consistently in less than six seconds running on a personal computer with a Pentium 4 1.5 GHz processor and 256 MB of RAM. We ran several experiments at higher values of n and for larger fields k.

Increasing the size of the field increases the run time of the program linearly. The larger values of n cause a much greater run time and manifest the critical elements of both the public key size of the cryptosystem and the computational complexity of our cryptanalysis. Since the public key is a set of $n + 1$ quartic polynomials, its size is of order $\mathcal{O}(n^4)$.

The following table indicates the public key size, median total run time, and median percent of total run time for each of the four steps, for various values of n as indicated. We used $|k| = 2^{16}$, which seems to be reasonable. A k of size 2^{32} would be quite reasonable as well.

n	Public Key (kBytes)	Total Run Time (sec)	Step 1 Find $\mathcal{F}$ (%)	Step 2 Define L'_1 and F' (%)	Step 3 Find $\mathcal{G}'$ (%)	Step 4 Define L'_2, G' and L'_3 (%)
5	1.5	10.8	11	78	8	3
6	2.9	40.0	9	80	8	2
10	22.0	1949	15	76	8	1
14	91.8	33654	10	80	9	1

Step 2 clearly comprises the bulk of the run time. Finding of the exact denominator of F takes almost all of this time, requiring $\frac{1}{24}(16n^6 + 131n^5 + 440n^4 + 595n^3 + 419n^2 + 114n)$ operations. However, step 1 has computational complexity of $\mathcal{O}(n^7)$ and step 3 has computational complexity of $\mathcal{O}(n^9)$ so eventually at higher values for n step 3 will comprise the bulk of the run time.

Remark. *From the steps above, it is clear our attack is not a simple application of any one existing attack method, let alone, just the Minrank attack alone. The key point is that we need first to accomplish a polynomial map decomposition and then recover a subtle rational map decomposition equivalent to the original one, which requires something much more than the Minrank method.*

One more important point is about the direct algebraic attack, namely from the public key, we can derive a set of polynomial equations once we are given the ciphertext, **but these are degree 4 equations not degree 2 equations**, *whose computation complexity, as we all know, is much higher than the case of degree 2 equations. This is further complicated by the fact that we are working on the field of size of* 2^{32}*, where the field equations can not be used. This is confirmed by our experiments, for example, Magma F4 implementation failed to solve even the cases* $n = 5$ *on an ordinary PC, which was proposed more than 20 years ago.*

4 Conclusion

We develop a new improved 2R decomposition method to break the family of rational multivariate public key cryptosystems proposed by Tsujii, Fujioka, and Hirayama in 1989. We also show that it is polynomial time to break this family of cryptosystems in terms of the number of variables, the critical parameter of the system. We demonstrate in computer experiments that our method is very efficient and we can break the scheme originally suggested for practical applications in only a few seconds on a standard PC. The main contribution of our work is that we develop new techniques to improve the original 2R decomposition such that it can be used successfully to attack a special family of rational maps. Although we defeat the cryptosystems, we still believe that this family of cryptosystems contains some very interesting ideas that may be utilized effectively.

References

[1] International Workshop on Post-Quantum Cryptography. Katholieke Universiteit Leuven, Belgium; 24–26 May 2006. http://postquantum.cr.yp.to

[2] Jean-Charles Faugere, Ludovic Perret: Cryptanalysis of 2R- Schemes. CRYPTO 2006, Page 357-372, LNCS 4117/2006, Springer

[3] Harriet Fell and Whitfield Diffie Analysis of a public key approach based on polynomial substitution Advances in cryptology—CRYPTO '85, LNCS 218, Springer 1986, 340-349

[4] L. Goubin, and J. Patarin. Asymmetric Cryptography with S-Boxes, Extended Version. Available at http://citeseer.ist.psu.edu/patarin97asymmetric.html.

[5] S. Tsujii and A. Fujioka and Y. Hirayama Generalization of the public key cryptosystem based on the difficulty of solving a system of non-linear equations ICICE Transactions (A) J72-A, Vol. 2, 1989, 390-397 English version is appended at http://eprint.iacr.org/2004/336

[6] Shigeo Tsujii and Kohtaro Tadaki and Ryou Fujita, Piece In Hand Concept for Enhancing the Security of Multivariate Type Public Key Cryptosystems: Public Key Without Containing All the Information of Secret Key, Cryptology ePrint Archive, Report 2004/366, http://eprint.iacr.org/2004/366

[7] S. Tsujii and K. Kurosawa and T. Itoh and A. Fujioka and T. Matsumoto A public key cryptosystem based on the difficulty of solving a system of nonlinear equations ICICE Transactions (D) J69-D, Vol. 12, 1986, Page 1963-1970

[8] WANG Lih-Chung, HU Yuh-Hua, FEIPEI LAI, CHOU Chun-Yen, YANG Bo-Yin Tractable rational map signature Public key cryptography - PKC 2005, LNCS 3386, pp. 244-257, 2005

[9] D.F. Ye, K.Y. Lam, Z.D. Dai. Cryptanalysis of 2R Schemes, Advances in Cryptology CRYPTO 1999, Lecture Notes in Computer Science, vol. 1666, Springer-Verlag, pp. 315-325, 1999.

[10] Specifications of SFLASH, NESSIE documentation, available at https://www.cosic.esat.kuleuven.ac.be/nessie/workshop/.