A New Public-Key Cryptosystem Based on the Problem of Reconstructing p–Polynomials

Cédric Faure and Pierre Loidreau

Ecole Nationale Supérieure de Techniques Avancées
{Cédric.Faure, Pierre.Loidreau}@ensta.fr

Abstract. In this paper we present a new public key cryptosystem whose security relies on the intractability of the problem of reconstructing p–polynomials. This is a cryptosystem inspired from the Augot–Finiasz cryptosystem published at Eurocrypt 2003. Though this system was broken by Coron, we show However, in our case, we show how these attacks can be avoided, thanks to properties of rank metric and p–polynomials. Therefore, public-keys of relatively small size can be proposed (less than 4000 bits).

1 Introduction

At EUROCRYPT 2003, a cryptosystem based on the so-called problem of *polynomial reconstruction* was presented by Augot and Finiasz [1]. However, this system was broken soon after by Coron, by modifying the Welch–Berlekamp decoding algorithm for Reed–Solomon codes. He managed to argue that in most cases, the system could be broken in Polynomial-time by recovering the valid plaintext from the ciphertext [4]. More recently a result by Kiayias and Yung showed that it was not possible to choose an other way of rescuing the system such as adding more errors and then using Sudan list-decoding algorithm [8]. A different attempt was to use properties of the Trace operator to scramble the structure and design a secure cryptosystem. Once again it was shown that this system could be broken [2, 4].

In this paper we design a cryptosystem based on a new problem called *p–polynomials reconstruction problem*. Whereas the classical polynomial reconstruction problem is closely related to the decoding of Reed–Solomon codes, this problem is closely related to the problem of decoding Gabidulin codes [9]. We show how the attacks investigated in the case of the original Augot–Finiasz cryptosystem can be prevented. Namely, the efficiency of the attacks mainly depends on the metric used in the design of the system. We also construct a public-key cryptosystem with a public-key size of at most 4000 bits, and a security to the state of the art attacks.

The outline of the paper is the following. In section 2, we briefly recall the definitions of rank metric and linear polynomials, mostly in order to fix the notations. In section 3, we present a first, simple adaptation of the Augot-Finiasz system, and we show its vulnerability. In section 4, we introduce the trace operator, and we use it to build the cryptosystem which is the main subject of the paper. Lastly, in section 5, we discuss about the security of this second system.

Ø. Ytrehus (Ed.): WWC 2005, LNCS 3969, pp. 304–315, 2006.

2 Gabidulin Codes and p–Polynomials

Let p be a prime number (practically $p = 2$), $q = p^m$, and $GF(q)$ be the field with q elements. p-polynomials (also called *Linearized polynomials*) over $GF(q)$ were widely investigated by Øre in 1933, 1934 [10, 11]. They are polynomials of the form:

$$P(X) = a_k X^{p^k} + \ldots + a_\ell X^{p^\ell} + \ldots + a_1 X^p + a_0 X,$$

where $a_0, \ldots, a_k$ are elements of $GF(q)$. If $a_k \neq 0$, the integer k is called the p-degree of P. From now on we will denote $[\ell] \stackrel{def}{=} p^\ell$. For any vector $\mathbf{x} \in GF(q)^n$, we denote $P(\mathbf{x}) \stackrel{def}{=} (P(x_1), \ldots, P(x_n))$.

Definition 1. *Let $\mathbf{c} = (c_1, \ldots, c_n)$ be a vector of length n over the field $GF(q) = GF(p^m)$. The rank of $\mathbf{c}$, denoted $Rk(\mathbf{c})$ is the rank of the $m \times n$ p-ary matrix obtained by expanding each coordinate of $\mathbf{c}$ over a basis of $GF(p^m)/GF(p)$.*

Using this definition, we can build the rank distance between two words, by computing the rank (which can move from 0 to n) of their difference. Then, we obtain the rank metric over words of length n in $GF(q)$.

The rank metric is used as a substitute of the Hamming metric, for cryptographic goals mostly. Many properties of the rank metric were widely studied, and it is not the goal of this paper to review them all.

We now assume $k \leq n \leq m$. Let $\mathbf{g} \in GF(q)^n$ a vector of rank n. The Gabidulin code of length n, dimension k and generating vector $\mathbf{g}$ is the set of words obtained by the evaluation of a p-polynomial of degree at most $k-1$ over $\mathbf{g}$:

$$Gab_k(\mathbf{g}) \stackrel{def}{=} \{(P(g_1), \ldots, P(g_n)) = P(\mathbf{g}) \ , \ \deg_p(P) \leq k-1\}.$$

By this construction, it can be shown that the code $Gab_k(\mathbf{g})$ has minimum rank distance $n-k$. Hence it is an optimal code for rank metric [6]. Moreover there exist polynomial-time decoding algorithms that can correct up to the error-correcting capability of the code [7, 9, 13, 14].

3 An Augot–Finiasz Type Cryptosystem

In this section we show how to design a cryptosystem similar to the Augot–Finiasz cryptosystem. We also show that Coron's attacks can equally be adapted, although their complexity is not any more linear in the weight of the error-vector, but exponential.

3.1 Construction of the System

Parameters:
Let $n \leq m$, and $k < n$, be integers and let $\mathbf{g} = (g_1, \ldots, g_n) \in GF(q)^n$ be a vector of linearly independent elements over $GF(p)$.

Key Generation:
The conceiver of the system picks randomly a p–polynomial P over $GF(q)$ of p–degree $k-1$. He chooses randomly a vector $\mathbf{E}$ of length n and of rank $W > (n-k)/2$.

- The public key is $\mathbf{K} = P(\mathbf{g}) + \mathbf{E}$.
- The secret key consists of the pair $(P, \mathbf{E})$.

Encryption:
The message $\mathbf{m} = (m_0, \ldots, m_{k-2}) \in GF(q)^{k-1}$ can be transformed into the p-polynomial $m(x) = \sum_{i=0}^{k-2} m_i x^{[i]}$. The sender chooses randomly $\alpha \in GF(q)$, and an error-vector $\mathbf{e}$ of rank $\omega \leq (n-k-W)/2$. The ciphertext $\mathbf{y}$ is:

$$\mathbf{y} = m(\mathbf{g}) + \alpha\mathbf{K} + \mathbf{e}.$$

Decryption:
The receiver projects $\mathbf{y}$ on the subspace of $GF(q)^n$ of dimension $n-W$ that is orthogonal to the vector space generated by the coordinates of $\mathbf{E}$. Since he knows $\mathbf{E}$, he equally knows or can easily compute a non-singular p-ary matrix $\mathbf{R}$ such that $\mathbf{ER}$'s $n-W$ first positions are equal to 0. Hence since $\mathbf{R}$ is p-linear, then $P(\mathbf{g})\mathbf{R} = P(\mathbf{gR})$, and

$$\mathbf{yR} = m(\mathbf{g})\mathbf{R} + \alpha P(\mathbf{g})\mathbf{R} + \alpha\mathbf{ER} + \mathbf{eR} = (m + \alpha P)(\mathbf{gR}) + \alpha\mathbf{ER} + \mathbf{eR}.$$

Let $\widetilde{\mathbf{yR}}$ be the vector of length $n-W$ obtained by removing the last W positions of $\mathbf{yR}$. We obtain that

$$\widetilde{\mathbf{yR}} = (m + \alpha P)(\widetilde{\mathbf{gR}}) + \widetilde{\mathbf{eR}}.$$

But $\mathrm{Rk}(\widetilde{\mathbf{eR}}) \leq \mathrm{Rk}(\mathbf{eR}) \leq \mathrm{Rk}(\mathbf{e}) \leq w \leq \frac{n-W-k}{2}$. Therefore, since $(m + \alpha P)$ has p-degree less than k,

By decoding $\widetilde{\mathbf{yR}}$ in the Gabidulin code $Gab_k(\mathbf{g}R)$ one recovers $Q = m + \alpha P$. Since P has p–degree exactly $k-1$, and since m has degree at most $k-2$, the field element αP_{k-1} is the leading coefficient of Q. Thus the receiver gets α and finally recovers $m = Q - \alpha P$.

3.2 Investigation on the Security of the System

The security of the system relies on the fact that given the public-key

$$\mathbf{K} = P(\mathbf{g}) + \mathbf{E},$$

it is not computationally feasible to recover $(P, \mathbf{E})$. Since any other possible candidate $(R, \mathbf{F})$ such that $\mathbf{K} = R(\mathbf{g}) + \mathbf{F}$ would lead to wrong decoding, an attacker would have to list all possibilities and try them one by one. It is a kind of list decoding of Gabidulin codes up to the rank W. We can show that

such a list-decoding is strictly equivalent to finding all p–polynomial V with $deg_p(V) \leq W$ and f with $deg_p(f) < k$ such that

$$\forall i = 1, \ldots, n \quad V(K_i) = V(f(g_i)). \tag{1}$$

This means that we would have to solve the following problem

Reconstruction($\mathbf{K}, \mathbf{g} = (g_1, \ldots, g_n), k, W$).
Find the set (V, f) where V is a non-zero p–polynomial of p-degree $\leq W$ and where f is a p–polynomial of p-degree $< k$, such that $V(g_i) = V[f(g_i)]$, for all $i = 1, \ldots, n$.

In rank metric we have no equivalent of the Johnson bound, but our simulation results tend to show that this problem is hard to solve. Actually, we used the MAGMA computational algebra system, which is optimized for fast computations over finite fields. Over small sizes ($p = 2, n = 3$), it is possible to compute the exhaustive list of solutions to the Reconstruction problem. But the size of this list increases dramatically (seemingly exponentionnaly) as soon as W is larger than the error-correcting capability of $Gab_k(\mathbf{g})$. Therefore solving this problem can be considered as being hard, as it implies the manipulation of an oversized list.

The problem on which the security of the Augot-Finiasz cryptosystem relies is also a hard problem. The system was nevertheless broken by Coron who showed that one could generally recover plaintexts by finding roots of a polynomial of degree $\omega + 1$,[4]. This new system does not suffer from the weakness shown by Coron for the original system. Namely, by following the same idea, an attacker would have to find the roots of a polynomial (not a p-polynomial) of degree $(q^{\omega+1} - 1)/(q - 1)$. Therefore this system would provide a much better security than the original one.

Our system suffers from an other weakness, which was shown in [5]. Given a received ciphertext $\mathbf{y}$ an attacker wants to find $m, \alpha, \mathbf{e}$ such that:

$$\mathbf{y} = m(\mathbf{g}) + \alpha\mathbf{K} + \mathbf{e},$$

with $\deg_p(m) \leq k - 2$ and $\mathrm{Rk}(\mathbf{e}) \leq \frac{n-k-W}{2}$.

Using a kind of Welch–Berlekamp technique, as in [4], solving this system is equivalent to finding m, α, V such that:

$$V(\mathbf{y}) = V \circ m(\mathbf{g}) + V(\alpha\mathbf{K}), \tag{2}$$

with $\deg_p(m) \leq k - 2$ and $\deg_p(V) \leq \omega$.

Instead of solving this system, we study a more general system. We search N, V, V' such that:

$$V(\mathbf{y}) = N(\mathbf{g}) + V'(\mathbf{K}),$$

with $\deg_p(N) \leq k + \omega - 2$, $\deg_p(V) \leq \omega$, and $\deg_p(V') \leq \omega$. Considering the coefficients of N, V, V' as unknowns, it is a linear system with n equations and $k + 3\omega + 1$ unknowns. Therefore, it can be solved in polynomial time. Since there is by construction at least one solution to the system, and since we can show

that $k + 3\omega + 1 < n$, this implies that the matrix of the system is degenerate, and the solution space often is of small dimension, typically 1. If it is the case we can find a solution to system (2).

4 System Using the Trace Operator

In this section we were inspired by the attempt to repair the original system by means of the trace operator which was introduced in [2]. In Hamming metric, Coron showed that this could not work. In rank metric however, such attacks cannot be adapted.

Definition 2. *The Trace operator from $GF(q^u)$ to $GF(q)$ is defined by :*

$$\forall x \in GF(q^u), Tr(x) = x + x^q + x^{q^2} + \ldots + x^{q^{u-1}}$$

We can extend this definition to vectors :

$$Tr(\mathbf{x}) \stackrel{def}{=} (Tr(x_1), \ldots, Tr(x_n))$$

and to linearized polynomials over $GF(q^u)$:

$$Tr(\sum_{i=0}^{k} p_i X^{[i]}) \stackrel{def}{=} \sum_{i=0}^{k} Tr(p_i) X^{[i]}$$

We will require the following proposition :

Proposition 1. *If $(g_1, \ldots, g_n) \in GF(q)^n$, and P is a p-polynomial over $GF(q^u)$, then $Tr(P(\mathbf{g})) = (Tr(P))(\mathbf{g})$.*

Proof.
Let $j \in [1, n]$. Since $g_j \in GF(q), \forall x \in GF(q^u)$, we have :
$Tr(xg_j) = g_j Tr(x)$ (by $GF(q)$-linearity).
So $Tr(P(g_j)) = Tr(\sum_{i=0}^{k} p_i g_j^{[i]}) = \sum_{i=0}^{k} g_j^{[i]} Tr(p_i) = Tr(P)(g_j)$.
Hence $Tr(P(\mathbf{g})) = (Tr(P))(\mathbf{g})$.

This leads to the design of a cryptosystem based on the trace operator.

4.1 Design of the System

Parameters

- We consider $\mathbf{g} = (g_1, \ldots, g_n)$ a vector formed of elements of $GF(q)$ that are linearly independent over $GF(p)$;
- An extension field $GF(q^u)$ of $GF(q)$;
- An integer k;
- An integer $W > \frac{n-k}{2}$. This implies that the linearized reconstruction problem is difficult as it was discussed in section 3.2.

Key generation
We generate randomly a p–polynomial P with coefficients in $GF(q^u)$, of p-degree $k-1$, such that the coefficients $p_{k-1}, \ldots, p_{k-u}$ form a basis of $GF(q^u)$ over $GF(q)$.

We also generate an error vector $\mathbf{E}$, with coefficients in $GF(q^u)$, of rank W.

- The public key is $\mathbf{K} = Tr(P(\mathbf{g})) + \mathbf{E} \in GF(q^u)^n$.
- The secret key is the pair $(P, \mathbf{E})$.

Encryption
Let $m \in GF(q)$ be a plaintext written as a p–polynomial of degree at most $k-u-1$, that is $m(X) = m_0 X + m_1 X^p + \ldots + m_{k-u-1} X^{p^{k-u-1}}$. We generate randomly $\alpha \in GF(q^u)$ and $\mathbf{e}$ an error vector in $GF(q)$ of rank $\omega \leq (n-W-k)/2$. The ciphertext is:

$$\mathbf{y} = m(\mathbf{g}) + Tr(\alpha \mathbf{K}) + \mathbf{e} \in GF(q)^n$$

The encryption can be done in $O(nk)$ multiplications in $GF(q)$.

Decryption
Without loss of generality we can assume that the receiver knows an invertible p-ary matrix $\mathbf{R}$, such that the first $n-W$ columns of $\mathbf{ER}$ are equal to zero.

Since p-polynomials and the trace operator are $GF(p)$-linear transformations, we have that $Tr(\alpha P(\mathbf{g}))\mathbf{R} = Tr(\alpha P(\mathbf{g}R))$. Since $\mathbf{g}$ has coefficients in $GF(q)$, the vector $\mathbf{gR}$ has also coefficients in $GF(q)$ and by proposition 1, this implies that $Tr(\alpha P(\mathbf{gR})) = Tr(\alpha P)(\mathbf{gR})$. Therefore:

$$\mathbf{yR} = m(\mathbf{g})\mathbf{R} + Tr(\alpha P(\mathbf{g}))\mathbf{R} + Tr(\alpha \mathbf{ER}) + \mathbf{eR},$$

$$\Leftrightarrow \mathbf{yR} = (m + Tr(\alpha P))(\mathbf{gR}) + Tr(\alpha \mathbf{ER}) + \mathbf{eR}.$$

Let $\widetilde{\mathbf{yR}}$ be the vector obtained by removing the last W positions of $\mathbf{yR}$. We have:

$$\widetilde{\mathbf{yR}} = (m + Tr(\alpha P))(\widetilde{\mathbf{gR}}) + \widetilde{\mathbf{eR}}.$$

But $\mathrm{Rk}(\widetilde{\mathbf{eR}}) \leq \mathrm{Rk}(\mathbf{eR}) \leq \mathrm{Rk}(\mathbf{e}) \leq w \leq \frac{n-W-k}{2}$.

Hence, by decoding $\widetilde{\mathbf{yR}}$ in the code $Gab_k(\widetilde{\mathbf{gR}})$, one recovers the linear polynomial $Q = m + Tr(\alpha P)$. Since the p-degree of m is at most $k-u-1$, and since $p_{k-1}, \ldots, p_{k-u}$ form a basis of $GF(q^u)$ over $GF(q)$, the receiver recovers α, and then $m = Q - Tr(\alpha P)$.

Using precomputation, the complexity of the decryption phase is then $O(\omega^2(k+\omega) + (k+\omega^2) + u^2 + kn)$ multiplications in $GF(q)$.

5 Security of the System

In the case of Hamming metric, Coron, showed that the cryptosystems could be broken in polynomial-time. It was even shown by Kiayias and Yung that it

was illusionary to try to repair the system by adding more errors and then using Sudan algorithm rather than classical decoding algorithms for Reed-Solomon codes [8].

Here, one attack can be translated directly from Coron's approach implying that the parameters have to be carefully chosen, but the second one is too much related with properties of Hamming metric to be adapted to rank metric.

Let $\gamma_1, \ldots, \gamma_u$ be a basis of $GF(q^u)$ over $GF(q)$. If we write $\alpha = \sum_{t=1}^{u} \alpha_t \gamma_t$, we have: $Tr(\alpha \mathbf{K}) = \sum_{t=1}^{u} \alpha_t Tr(\gamma_t \mathbf{K})$. Let $\mathbf{K}_t = Tr(\gamma_t \mathbf{K})$, for $t = 1 \ldots u$.

The vectors $\mathbf{K}_t$ are vectors in $GF(q)$ easily computable from the public key $\mathbf{K}$. Knowing $\mathbf{y}, \mathbf{g}, \mathbf{K}_1, \ldots, \mathbf{K}_u$, recovering the plaintext consists of solving:

$$\exists \mathbf{e}, m, \alpha_1, \ldots, \alpha_u, \begin{cases} \mathbf{y} = m(\mathbf{g}) + \sum_{t=1}^{u} \alpha_t \mathbf{K}_t + \mathbf{e}, \\ \deg_p(m) \leq k - u - 1, Rk(\mathbf{e}) \leq \omega. \end{cases} \tag{3}$$

The rest of this section consists of investigating three ways of solving system (3).

5.1 Decoding Attacks

The decoding attack is a cipher-text only attack. The ciphertext can be seen under the form

$$\mathbf{y} = \mathbf{cG} + \mathbf{e},$$

where $\mathbf{c} = (m_0, \ldots, m_{k-u-1}, \alpha_1, \ldots, \alpha_u)$, and

$$\mathbf{G} = \begin{pmatrix} g_1 & \cdots & g_n \\ \vdots & \ddots & \vdots \\ g_1^{[k-u-1]} & \cdots & g_n^{[k-u-1]} \\ & \mathbf{K}_1 & \\ & \vdots & \\ & \mathbf{K}_u & \end{pmatrix}$$

Hence it can be reduced to a problem of decoding the linear code of dimension k generated by $\mathbf{G}$ up to the rank distance ω. To do this we do not know a better decoder than a general purpose decoder. The most efficient one was designed by Ourivski and Johannson and works in $O((q\omega^3)p^{(k+1)(\omega-1)})$, operations in the base field, [12].

5.2 Attack by Linearization

This attack is made on the ciphertext. The attacker writes the encryption equation, and uses it to obtain a linear system over $GF(q)$.

Proposition 2. *Assuming $k+(u+2)\omega \leq n$, an attacker can recover the plaintext from a given ciphertext in polynomial time with very high probability. If $n > k+(u+2)\omega$, then the same attack can be made in $O(n^3 q^{n-k-(u+2)\omega})$ multiplications with very high probability.*

Proof.
We consider another kind of Welch–Berlekamp approach. Namely, solving system (3) is equivalent to solving:

$$\exists m, \alpha_1, \ldots, \alpha_u, V, \begin{cases} V(\mathbf{y}) = V \circ m(\mathbf{g}) + \sum_{t=1}^{u} \alpha_t V(\mathbf{K}_t), \\ \deg_p(m) \leq k-u-1, \deg_p(V) \leq \omega \end{cases}$$

By linearizing the equations, we now obtain the following system:

$$\exists V, R_1, \ldots, R_u, N, \begin{cases} V(\mathbf{y}) = N(\mathbf{g}) + \sum_{t=1}^{u} R_t(\mathbf{K}_t), \\ \deg_p(V) \leq \omega, \deg_p(R_t) \leq \omega, \\ \deg_p(N) \leq k+\omega-u-1 \end{cases}$$

If one writes the unknown coefficients of polynomials $V, R_1, \ldots, R_u, N$ in a vectorial form, one has to solve :

$$M \begin{pmatrix} V \\ R_1 \\ \vdots \\ R_u \\ N \end{pmatrix} = 0$$

where :

$$M = \left(\begin{array}{cc|ccc|c|ccc} y_1 & y_1^{[\omega]} & -K_{11} & -K_{11}^{[\omega]} & \ldots & -K_{u1} & -K_{u1}^{[\omega]} & -g_1 & -g_1^{[k+\omega]-u-1} \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots \\ y_j & y_j^{[\omega]} & -K_{1j} & -K_{1j}^{[\omega]} & \ldots & -K_{uj} & -K_{uj}^{[\omega]} & -g_j & -g_j^{[k+\omega]-u-1} \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots \\ y_n & y_n^{[\omega]} & -K_{1n} & -K_{1n}^{[\omega]} & \ldots & -K_{un} & -K_{un}^{[\omega]} & -g_n & -g_n^{[k+\omega]-u-1} \end{array}\right)$$

using the notation $[i] = p^i$.

M is $n \times (k+(u+2)\omega+1)$-matrix in $GF(q)$, we can compute its kernel in polynomial time to find all the solutions for polynomials $V, R_1, \ldots, R_u, N$.

Because there exists a non-trivial solution, the kernel of M is of dimension at least 1. Aside from this condition, practical experiments lead us to think that M is very likely to have maximum rank. Thus, we simply deduce the dimension of $\ker(M)$ from the size of M, and we almost always have : $\dim(\ker(M)) = \max(1, k+(u+2)\omega+1-n)$.

From the space of solutions for $V, R_1, \ldots, R_u, N$, we now have to extract a suitable non-trivial solution for $m, \alpha_1, \ldots, \alpha_u, V$, and then this m is the plaintext.

If $k + (u+2)\omega \leq n$, then with high probability $\dim(\ker(M)) = 1$, and thus, any solution of the linearized system allows us to find m by a mere left Euclidian division, the whole attack made in polynomial time.

If this is not the case, one has to check each direction of the solution space, which leads to an attack in roughly $O(n^3 q^{n-k-(u+2)\omega})$ multiplications, [5].

5.3 Algebraic Attacks

This part is devoted to finding the secret element α. Once one recovers the element $\alpha \in GF(q^u)$, it is trivial to recover the plaintext, by computing $\mathbf{y} - Tr(\alpha \mathbf{K}) = m(\mathbf{g}) + \mathbf{e}$ and decoding this vector in the Gabidulin code.

To decrypt we have to solve the following system

$$y_i - Tr(\alpha K_i) = m(g_i) + e_i, \quad \forall i = 1, \ldots, n,$$

where the unknowns are $\mathbf{e} = (e_1, \ldots, e_n)$ of rank ω, $\alpha \in GF(q^u)$, and the p-polynomial m. Solving this system is still equivalent to solving

$$V(y_i - Tr(\alpha K_i)) = V \circ m(g_i), \quad \forall i = 1, \ldots, n, \tag{4}$$

where the unknowns are m, α and the coefficients of a p-polynomial V of degree ω. Now to solve this system we consider the system

$$V(y_i - Tr(\alpha K_i)) = N(g_i), \quad \forall i = 1, \ldots, n \tag{5}$$

where the unknowns are the element α, and the coefficients of two p-polynomials V of degree ω and N of degree $k + \omega - u - 1$. If (α, V, m) is a solution of (4), then $(\alpha, V, N = V \circ m)$ is a solution of (5). There are two manners to solve this system:

Univariate case
Let us define the following $n \times (k + 2\omega - u + 1)$ matrix

$$M(x) = \left(\begin{array}{ccc|ccc} y_1 - Tr(xK_1) & \cdots & (y_1 - Tr(xK_1))^{[\omega]} & g_1 & \cdots & g_1^{[k+\omega-u-1]} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ y_n - Tr(xK_n) & \cdots & (y_n - Tr(xK_n))^{[\omega]} & g_n & \cdots & g_n^{[k+\omega-u-1]} \end{array}\right),$$

where $[i] = p^i$.

Provided α is known, $M(\alpha)$ is the matrix of system (5). Since by construction $(k + 2\omega - u + 1) \leq n$ and since that we know that there is a non-zero solution, the matrix $M(\alpha)$ is not of full rank. Therefore, the determinant of every square submatrix $\widetilde{M}(x)$ of $M(x)$ satisfies the equation

$$Det(\widetilde{M}(\alpha)) = 0.$$

The Trace operator is a polynomial of degree q^{u-1}, therefore any determinant is a polynomial over $GF(q^u)$ of degree at most $q^{u-1}(p^{\omega+1}-1)/(p-1)$. Hence, a way to find the value of α would be to search for common divisors between some of the obtained determinants. However for practical cases, as we will see further, the quantity $q^{u-1}(p^{\omega+1}-1)/(p-1) > 2^{80}$.

Since such an approach is not very practical, we skip to another more interesting case.

Multivariate case
If we set $x = \sum_{j=1}^{u} \gamma_j x_j$, where $\gamma_1, \ldots, \gamma_u$ is a basis of $GF(q^u)/GF(q)$, by setting $K_{i,t} \stackrel{def}{=} Tr(\gamma_t K_i)$ for $t = 1, \ldots, u$ and $i = 1, \ldots, n$, we now define

$$M(x_1, \ldots, x_u) \stackrel{def}{=} \left(\begin{array}{ccc|ccc} y_1 - \sum_{t=1}^{u} K_{1,t}x_t & \cdots & (y_1 - \sum_{t=1}^{u} K_{1,t}x_t)^{[\omega]} & g_1 & \cdots & g_1^{[k+\omega-u-1]} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ y_n - \sum_{t=1}^{u} K_{n,t}x_t & \cdots & (y_n - \sum_{t=1}^{u} K_{n,t}x_t)^{[\omega]} & g_n & \cdots & g_n^{[k+\omega-u-1]} \end{array} \right).$$

Once again if $\alpha \stackrel{def}{=} \sum_{t=1}^{u} \gamma_t \alpha_t$, then $M(\alpha_1, \ldots, \alpha_u)$ is the matrix of the linear system (5). Hence, the determinant of every square submatrix $\widetilde{M}(x_1, \ldots, x_u)$ of M satisfies the equation

$$Det(\widetilde{M}(\alpha_1, \ldots, \alpha_u)) = 0.$$

The determinants $\widetilde{M}(x_1, \ldots, x_u)$ are multivariate polynomials of degree at most $(p^{(\omega+1)}-1)/(p-1)$. in u variables. We can construct up to $\binom{n}{k+2\omega-u+1}$ different determinants by choosing exactly $k+2\omega-u+1$ lines out of n.

We made some simulations in the MAGMA language by using the algorithms finding first Gröbner bases and then solving the equations. Every time we succeeded in computing the Gröbner Basis of the system we obtained only one solution that was exactly the element α of the private key.

Simulation results can be found in Table 1. For these computations we used an OPTERON processor 2,2Ghz with 8Gb of memory.

Table 1. Simulations of attacks made on parameters $n = 36,\ q = 2^{36},\ k = 10,\ W = 14$

Number of variables	Error-Rank	Degree	Magma 2.11-2/F4
$u = 2$	$\omega = 2$	7	0.01s
	$\omega = 3$	15	0.340s
	$\omega = 4$	31	9s
	$\omega = 5$	63	500s
	$\omega = 6$	127	11 hours
$u = 3$	$\omega = 2$	7	0.06s
	$\omega = 3$	15	54s
	$\omega = 4$	31	15 hours

5.4 Discussion About the Choice of Parameters

Now we propose the following set of parameters:

- *Extension field:* $q = 2^{36}$, and $u = 3$ which implies $q^u = 2^{108}$.
- *Length of the code:* $n = 36$.
- *Public-key:* vector of length 36 over $GF(2^{108})$, that is $36 \times 108 = 3888$ bits.

The different attacks we described in the paper give the following results:

- *Decoding attacks*: The best general purpose decoding algorithm was designed by Ourivski and Johannson [12]. The complexity of recovering a vector of rank ω in a code of dimension $k + u$ is equal to $(n\omega)^3 2^{(k+u+1)(\omega-1)} > 2^{91}$ binary operations.
- *Attacks by linearization*: We have to check $q^{(k+(u+2)\omega-n)} = 2^{144}$ solutions of a linear system to recover the plaintext.
- *Algebraic attacks*: Table 1 shows that these parameters are well beyond what is feasible for now. Namely the system to solve consists of $\approx 2^{32}$ cubic equations of degree 127 over $GF(2^{36})$. In the univariate case, one has to compute *gcd's* of polynomials of degree 2^{100}.

An implementation of the system in the MAGMA language on a 1200 MHz processor gives the following average times (1000 tests). The decoding algorithm used is described in [5, 9].

- *Key generation:* 72 ms.
- *Precomputation:* 16 ms.
- *Cipher:* 23 ms.
- *Decipher:* 7.9 ms.

This corresponds to a the transmision of $(k - u)m = 252$ information bits, encapsulated in a message of $nm = 1296$ bits, the useful transmission rate is so about 11 kb/s on this computing speed. Therefore, we can greatly increase the speed of the algorithms by using an efficient language, for example C language. It is also possible to transmit more information by putting some information on the error codeword. On how to do this, see for example [3].

References

1. D. Augot and M. Finiasz. A public key encryption scheme bases on the polynomial reconstruction problem. In *EUROCRYPT 2003*, pages 222–233, 2003.
2. D. Augot, M. Finiasz, and P. Loidreau. Using the trace operator to repair the polynomial reconstruction based cryptosystem presented at eurocrypt 2003. Cryptology ePrint Archive, Report 2003/209, 2003. http://eprint.iacr.org/.
3. T. P. Berger and P.Loidreau. Designing an efficient and secure public-key cryptosystem based on reducible rank codes. In *Proceedings of INDOCRYPT 2004*, 2004.

4. J.-S. Coron. Cryptanalysis of a public-key encryption scheme based on the polynomial reconstruction problem. In F. Bao, R. Deng, and J. Zhou, editors, *7th International Workshop on Theory and Practice in Public Key Cryptography, PKC 2004*, volume 2947, pages 14–28. Springer, 2004.
5. C. Faure. Etude d'un système de chiffrement à clé publique fondé sur le problème de reconstruction de polynômes linéaires. Master's thesis, Université Paris 7, 2004.
6. E. M. Gabidulin. Theory of codes with maximal rank distance. *Problems of Information Transmission*, 21:1–12, July 1985.
7. E. M. Gabidulin. A fast matrix decoding algorithm for rank-error correcting codes. In G. Cohen, S. Litsyn, A. Lobstein, and G. Zémor, editors, *Algebraic coding*, volume 573 of *LNCS*, pages 126–133. Springer-Verlag, 1991.
8. A. Kiayias and M. Yung. Cryptanalyzing the polynomial-reconstruction based public-key system under optimal parameter choice. Cryptology ePrint Archive, Report 2004/217, 2004. http://eprint.iacr.org/.
9. P. Loidreau. Sur la reconstruction des polynômes linéaires : un nouvel algorithme de décodage des codes de Gabidulin. *Comptes Rendus de l'Académie des Sciences : Série I*, 339(10):745–750, 2004.
10. Ö. Øre. On a special class of polynomials. *Transactions of the American Mathematical Society*, 35:559–584, 1933.
11. Ö. Øre. Contribution to the theory of finite fields. *Transactions of the American Mathematical Society*, 36:243–274, 1934.
12. A. Ourivski and T. Johannson. New technique for decoding codes in the rank metric and its cryptography applications. *Problems of Information Transmission*, 38(3):237–246, September 2002.
13. G. Richter and S. Plass. Error and erasure decoding of rank-codes with a modified Berlekamp-Massey algorithm. In *5th Int. ITG Conference on Source and Channel Coding (SCC 04)*, 2004.
14. R. M. Roth. Maximum-Rank array codes and their application to crisscross error correction. *IEEE Transactions on Information Theory*, 37(2):328–336, March 1991.