Cryptanalysis of the Repaired Public-key Encryption Scheme Based on the Polynomial Reconstruction Problem

Jean-Sébastien Coron
Gemplus Card International
34 rue Guynemer
Issy-les-Moulineaux, F-92447, France
jean-sebastien.coron@gemplus.com

Abstract. At Eurocrypt 2003, Augot and Finiasz proposed a new public-key encryption scheme based on the polynomial reconstruction problem [1]. The scheme was subsequently broken in [5], who showed that given the public-key and a ciphertext, one could recover the corresponding plaintext in polynomial time. Recently, Augot, Finiasz and Loidreau published on the IACR eprint archive a reparation [2] of the cryptosystem. The reparation is based on the trace operator, and is resistant against the previous attack. However, we describe a new cryptanalysis of the repaired scheme. Given the public-key and a ciphertext, we can still recover the corresponding plaintext in polynomial time. Our technique is a variant of the Berlekamp-Welsh algorithm, and works very well in practice, as for the proposed parameters, we recover the plaintext in less than 8 minutes on a single PC.

Key-Words: Cryptanalysis, Augot and Finiasz cryptosystem, Polynomial Reconstruction Problem, Reed-Solomon codes.

1 Introduction

We describe a cryptanalysis of a public-key encryption scheme recently published by Augot, Finiasz and Loidreau [2]. The scheme is based on the polynomial reconstruction (PR) problem [10], which is the following:

Problem 1 (Polynomial Reconstruction). Given n, k, ω and $(x_i, y_i)_{i=1\ldots n}$, output any polynomial p such that $\deg p < k$ and $p(x_i) = y_i$ for at least $n - \omega$ values of i.

This problem has an equivalent formulation in terms of the decoding of Reed-Solomon error-correcting codes [11]. The problem can be solved in polynomial time when the number of errors ω is such that $\omega \le (n-k)/2$, using the Berlekamp-Welsh algorithm [3]. This has been improved to $\omega \le n - \sqrt{kn}$ by Guruswami and Sudan [7].

When the number of errors is larger, no polynomial time algorithm is known for the PR problem. Therefore, some cryptosystem have been constructed based on the hardness of the PR problem; for example, an oblivious polynomial evaluation scheme [10], and a semantically secure symmetric cipher [8].

At Eurocrypt 2003, Augot and Finiasz proposed a new public-key encryption scheme based on the polynomial reconstruction problem [1]. A security level exponential in terms of the parameters was conjectured. However, the scheme was subsequently broken in [5], who showed that given the public-key and a ciphertext, one could recover the corresponding plaintext in polynomial time. Recently, Augot, Finiasz and Loidreau published on the IACR eprint archive a reparation [2] of the cryptosystem. The reparation is based on the trace operator, and is resistant against the previous attack. However, we describe a new cryptanalysis of the repaired scheme. Given the public-key and a ciphertext, we can still recover the corresponding plaintext in polynomial time. Our technique is again a variant of the Berlekamp-Welsh algorithm [3], and works very well in practice, as for the proposed parameters, we recover the plaintext in less than 8 minutes on a single PC.

2 The Original Cryptosystem

In this section, we recall the original cryptosystem proposed by Augot and Finiasz at Eurocrypt 2003. As in [1], we first recall some basic definitions of Reed-Solomon codes.

2.1 Reed-Solomon Codes

Let F_q be the finite field with q elements and let $x_1, \cdots, x_n$ be n distinct elements of F_q. We denote by ev the following map:

$$ev : \begin{cases} F_q[X] \rightarrow F_q^n \\ p(X) \ \ \rightarrow (p(x_1), \ldots, p(x_n)) \end{cases}$$

Definition 1. *The Reed-Solomon code of dimension k and length n over F_q is the following set of n-tuples (codewords):*

$$RS_k = \{ev(f); f \in F_q[X], \deg f < k\}$$

where $F_q[X]$ is the set of univariate polynomials with coefficients in F_q.

The *weight* of a word $c \in F_q^n$ is the number of non-zero coordinates in c. The *Hamming distance* between two words x and y is the weight of $x - y$. Formally, the problem of decoding Reed-Solomon code is the following:

Problem 2 (Reed-Solomon decoding). Given a Reed-Solomon code RS_k of length n, ω an integer and a word $y \in F_q^n$, find any codeword in RS_k at distance less than ω of y.

The smallest weight of non-zero codewords in RS_k is $n - k + 1$. Therefore, when $\omega \leq (n - k)/2$, the solution to Reed-Solomon decoding is guaranteed to be unique. It is easy to see that the Polynomial Reconstruction problem and the Reed-Solomon decoding problem are equivalent. Both problems can be solved in polynomial time when $w \leq (n - k)/2$, using the Berlekamp-Welsh algorithm [3].

2.2 The original Cryptosystem

In the following, we briefly review Augot and Finiasz public-key cryptosystem [1].

Parameters: q is the size of F_q, n is the length of the Reed-Solomon code, k its dimension, W is the weight of a large error, so that the PR problem for n, k, W is believed to be hard, *i.e.* we must have:

$$W > \frac{n-k}{2}$$

ω is the weight of a small error, for which the PR problem with $n - W$ coordinates is easy:

$$\omega \leq \frac{n-W-k}{2} \tag{1}$$

It is recommended in [1] to take $n = 1024$, $k = 900$, $\omega = 25$, $W = 74$ and $q = 2^{80}$.

Key generation: Generate a unitary polynomial p of degree $k - 1$, and a random n-dimensional vector E of weight W. Compute the codeword $c = ev(p)$ of RS_k. The public key is $z = c + E$, while the private key is (p, E).

Encryption: Let m a message of length $k - 1$ over the alphabet F_q. The message m is seen as a polynomial $m(X) = m_0 + m_1 X + \ldots + m_{k-1} X^{k-2}$ of degree at most $k - 2$. Generate a random $\alpha \in F_q$ and a random error e of weight ω. The ciphertext y is then:

$$y = ev(m) + \alpha \times (c + E) + e$$

Decryption: One considers only the positions where $E_i = 0$ and define the shortened code of length $n - W$, which is also a Reed-Solomon code of dimension k, which we denote $\overline{RS}_k$. Let $\overline{y}, \overline{ev}(m), \overline{c}, \overline{e}$ be the shortened $y, ev(m), c, e$. One must solve the equation:

$$\overline{y} = \overline{ev}(m) + \alpha \times \overline{c} + \overline{e}$$

We have $\overline{ev}(m) + \alpha \times \overline{c} \in \overline{RS}_k$, and from (1), the weight of the small error $\overline{e}$ is less than the error correction capacity of $\overline{RS}_k$; therefore, using the Berlekamp-Welsh algorithm, one can recover the unique polynomial r of degree $k - 1$ such that:

$$ev(r) = \overline{ev}(m) + \alpha \times \overline{c}$$

which gives

$$r = m + \alpha \cdot p$$

Since $\deg(m) \leq k - 2$ and p is a unitary polynomial of degree $k - 1$, the field element α is the leading coefficient of r. Therefore one can recover m as:

$$m = r - \alpha \cdot p$$

3 Attack on the Original Cryptosystem

In this section, we recall [5]'s attack on the original cryptosystem. The attack is a variant of the Berlekamp-Welsh algorithm for solving the PR problem (see [6]). Let n, k, W, ω and q be the parameters of the system. Let (p, E) be the private key and $z = ev(p) + E$ be the public-key. Let m be the plaintext encoded as a polynomial of degree less than $k-2$. Let e be an error vector of weight ω, and α be a field element. Let

$$y = ev(m) + \alpha \times z + e \tag{2}$$

be the corresponding ciphertext.

Theorem 1. *Given the public-key z and the ciphertext y, one can recover the corresponding plaintext m in polynomial time.*

Proof. Let y_i, z_i and e_i be the components of the words y, z and e. Given y and z, one must solve the following set of equations:

$$\exists e, m, \alpha,\ \ y_i = m(x_i) + \alpha \cdot z_i + e_i \ \text{ for all } 1 \le i \le n \tag{3}$$

where the weight of e is less than ω. Note that from the definition of the cryptosystem, there is a unique solution.

Consider the following set of equations:

$$\exists V, m, \alpha,\ \begin{cases} \deg(V) \le \omega,\ V \ne 0,\ \deg(m) \le k-2 \\ \forall i,\quad V(x_i) \cdot (y_i - \alpha \cdot z_i) = V(x_i) \cdot m(x_i) \end{cases} \tag{4}$$

Any solution V, m, α of (4) gives a solution to (3). Namely, the fact that $V \ne 0$ and $\deg V \le \omega$ implies that V can be equal to zero at most ω times. Therefore, letting $e_i = y_i - m(x_i) - \alpha \cdot z_i$, the weight of e is less than ω.

Conversely, any solution to (3) gives a solution to (4). Namely, one can take $V(X) = \prod_{i \in B}(X - x_i)$ with $B = \{i | e_i \ne 0\}$. The problem of solving (3) can thus be reduced to finding V, m, α satisfying (4). Consider now the following set of equations:

$$\exists V, N, \lambda,\ \begin{cases} \deg(V) \le \omega,\ V \ne 0,\ \deg(N) \le k + \omega - 1 \\ \forall i,\quad V(x_i) \cdot (y_i - \lambda \cdot z_i) = N(x_i) \end{cases} \tag{5}$$

The system (5) is a linearized version of (4), in which one has replaced the product $V(x_i) \cdot m(x_i)$ by $N(x_i)$. It is easy to see that any solution of (4) gives a solution to (5), as one can take $\lambda = \alpha$ and $N = m \cdot V$. However, the converse is not necessarily true.

For a given λ, the system (5) gives a linear system of n equations in the $k+2\cdot\omega+1$ unknown, which are the coefficients of the polynomials V and N. More precisely, denoting:

$$V(X) = \sum_{i=0}^{\omega} v_i \cdot X^i, \quad N(X) = \sum_{i=0}^{k+\omega-1} n_i \cdot X^i$$

and Y the vector of coordinates:

$$Y = (v_0, \cdots, v_\omega, n_0, \cdots, n_{k+\omega-1})$$

one let $M(\lambda)$ be the matrix of the system:

$$M(\lambda)_{i,j} = \begin{cases} (y_i - \lambda \cdot z_i) \cdot (x_i)^j & \text{if } 0 \le j \le \omega \\ -(x_i)^{j-\omega-1} & \text{if } \omega < j < k + 2\omega + 1 \end{cases}$$

The matrix $M(\lambda)$ is a rectangular matrix with n lines and $k+2\omega+1$ columns; from (1) we have that $n > k+2\omega+1$. The coefficients of $M(\lambda)$ are a function of the public-key and the ciphertext only. The system (5) is then equivalent to:

$$\exists Y, \lambda, \quad M(\lambda).Y = 0, \ Y \neq 0 \tag{6}$$

Let one consider the matrix $M(\lambda)$ with $\lambda = 0$. Using Gaussian elimination, one computes the rank of the matrix $M(0)$. One distinguishes two cases: $rank\ M(0) = k+2\omega+1$, and $rank\ M(0) < k+2\omega+1$.

If $rank\ M(0) = k+2\omega+1$, then there exists a square sub-matrix of $M(0)$ of dimension $k+2\omega+1$ which is invertible. Without loss of generality, one can assume that the matrix obtained by taking the first $k+2\omega+1$ lines of $M(0)$ is invertible. Let $M'(\lambda)$ be the square matrix obtained by taking the first $k+2\omega+1$ lines of $M(\lambda)$. Any solution Y, λ of (6) satisfies:

$$M'(\lambda).Y = 0, \ Y \neq 0$$

which implies that the matrix $M'(\lambda)$ is non-invertible, *i.e.* $\det(M(\lambda)) = 0$. Then, the solution α in system (4) must be a root of the function:

$$f(\lambda) = \mathrm{Det}(M'(\lambda))$$

which is a polynomial of degree at most $\omega+1$. The polynomial f is not identically zero, because $M'(0)$ is invertible, which implies $f(0) \neq 0$. The polynomial f can easily be obtained from the public-key z and the ciphertext y by computing $f(\lambda) = \mathrm{Det}(M'(\lambda))$ for $\omega+2$ distinct values of λ and then using Lagrange interpolation.

The factorization of a polynomial over a finite-field can be done in polynomial time (see for example [13]). Therefore, one obtains a list of at most $\omega+1$ candidates, one of which being the solution α of (4), and equivalently, of (3). For the right candidate α, the vector $y - \alpha \times z$ is equal to $ev(m) + e$, where the weight of e is less than the error correcting capacity of the Reed-Solomon code. Therefore, using Berlekamp-Welsh algorithm, one recovers the plaintext m from $y - \alpha \times z$ in polynomial time.

More precisely, let α, m, e be the solution of (3). Given a solution V, N, λ of (5) with $\lambda = \alpha$, we have for all $1 \leq i \leq k + 2 \cdot \omega + 1$:

$$V(x_i) \cdot (m(x_i) + e_i) = N(x_i)$$

Since the error vector e has a weight at most ω, we have for at least $\omega+k+1$ values of i:

$$V(x_i) \cdot m(x_i) = N(x_i)$$

N and $V \cdot m$ are therefore two polynomials of degree less than $\omega+k-1$ which take the same value on at least $\omega+k+1$ distinct points; consequently, the two polynomials must be equal. This means that one can recover m by performing a polynomial division:

$$m = \frac{N}{V}$$

Therefore, one can recover the plaintext in polynomial time.

Let us now consider the second case, *i.e.* $rank\ M(0) < k+2\omega+1$. Then there exists $Y \neq 0$ such that $M(0).Y = 0$. The vector Y gives the coefficients of two polynomials V and N such that for all $1 \leq i \leq n$:

$$V(x_i) \cdot y_i = N(x_i)$$

From (2) we have $y_i = m(x_i) + \alpha \cdot (p(x_i) + E_i) + e_i$, which gives for all i:

$$V(x_i) \cdot \big((m + \alpha \cdot p)(x_i) + \alpha \cdot E_i + e_i\big) = N(x_i)$$

The weight of E is at most W and the weight of e is at most ω. Moreover, from (1) we have $n \geq k + 2\omega + W$. Therefore, for at least $\omega + k$ values of i, we have:

$$V(x_i) \cdot (m + \alpha \cdot p)(x_i) = N(x_i)$$

As previously, $V \cdot (m + \alpha \cdot p)$ and N are two polynomials of degree less than $k + \omega - 1$ which take the same value on at least $\omega + k$ distinct points; consequently, they must be equal, which gives:

$$m + \alpha \cdot p = \frac{N}{V}$$

Since the polynomial p is unitary and $\deg p = k - 1$ and $\deg m \leq k - 2$, this enables to recover α. Then, as previously, given α, we recover m in polynomial time[1]. □

Kiayias and Yung describe in [9] a modification of Augot and Finiasz cryptosystem, resistant against the previous attack, but which they manage to break in the same paper.

4 The Repaired Cryptosystem

In this section, we describe the repaired cryptosystem published in [2]. The new cryptosystem is resistant against the previous attack. The reparation is based on working in the subfield of a given field, and using the trace operator. Following [2], we recall these notions in the next section.

4.1 Subfields and Trace Operator

We consider the finite field $\mathrm{GF}(q^u)$, where q is the power of a prime integer. The finite field $\mathrm{GF}(q)$ is a subfield of $\mathrm{GF}(q^u)$. The finite field $\mathrm{GF}(q^u)$ can be viewed as a u-dimensional vector space over $\mathrm{GF}(q)$. Let $\gamma_1, \ldots, \gamma_u$ be a basis of $\mathrm{GF}(q^u)$ over $\mathrm{GF}(q)$, then every element $\alpha \in \mathrm{GF}(q^u)$ can be uniquely written $\alpha = \sum_{i=1}^{u} \alpha_i \gamma_i$, where $\alpha_i \in \mathrm{GF}(q)$.

Definition 2. *The trace operator of $GF(q^u)$ into $GF(q)$ is defined by:*

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

[1] In this second case, we can also recover the private key (p, E). It has been shown in [9] that this second case happens with negligible probability.

The trace operator is a GF(q)-linear mapping (and not GF(q^u)-linear) of GF(q^u) into GF(q). For any basis $\gamma_1, \ldots, \gamma_u$ of GF(q^u), there exists a unique dual basis $\gamma_1^*, \ldots, \gamma_u^*$ with respect to the Trace operator. The dual basis is such that:

$$\mathrm{Tr}(\gamma_i \gamma_j^*) = 1 \text{ if } i = j, \text{ and } 0 \text{ otherwise}$$

The dual basis can be efficiently computed. We extend the trace operator to vectors:

$$\mathrm{Tr}(c_1, \ldots, c_n) = (\mathrm{Tr}(c_1), \ldots, \mathrm{Tr}(c_n))$$

and to polynomials: for any polynomial $p \in \mathrm{GF}(q^u)[X]$, $p(x) = \sum_{i=0}^{k} p_i x^i$, we define the polynomial $\mathrm{Tr}(p) \in \mathrm{GF}(q)[X]$ as:

$$\mathrm{Tr}(p)(x) = \sum_{i=0}^{k} \mathrm{Tr}(p_i) x^i$$

Let $x_1, \cdots, x_n$ be n distinct elements of $\mathrm{GF}(q) \in \mathrm{GF}(q^u)$. As in section 2.1 we denote by ev the following map:

$$ev : \begin{cases} \mathrm{GF}(q^u)[X] \to \mathrm{GF}(q^u)^n \\ p(X) \qquad\quad \to (p(x_1), \ldots, p(x_n)) \end{cases}$$

Proposition 1. *For all $p \in GF(q^u)[X]$, we have $Tr(ev(p)) = ev(Tr(p))$*

Proof. The j-th component of $\mathrm{Tr}(ev(p))$ is

$$\mathrm{Tr}(p(x_j)) = \mathrm{Tr}(\sum_{i=0}^{k} p_i \cdot x_j^i)$$

From the GF(q)-linearity of the Trace operator and the fact that $x_j \in \mathrm{GF}(q)$, we obtain:

$$\mathrm{Tr}(p(x_j)) = \sum_{i=0}^{k} \mathrm{Tr}(p_i) x_j^i$$

which is the j-th component of $ev(\mathrm{Tr}(p))$. □

As in section 2.1, we define the Reed-Solomon code of dimension k and length n over GF(q^u) as the following set of n-tuples (codewords):

$$RS_k = \{ev(f); f \in \mathrm{GF}(q^u)[X], \deg f < k\}$$

4.2 The Repaired Cryptosystem

In this section, we recall the repaired cryptosystem [2].

Parameters: A finite field $\mathrm{GF}(q^u)$, an integer n as the length of the Reed-Solomon code, k its dimension, W is the weight of a large error, ω is the weight of a small error, for which the PR problem with $n-W$ coordinates is easy:

$$\omega \leq \frac{n-W-k}{2} \tag{7}$$

The authors of the repaired cryptosystem recommend in [2] to take $q = 2^{20}$, $u = 4$, $n = 2048$, $k = 1400$, $W = 546$ and $\omega = 49$. [2]

Key generation: Generate a random polynomial p of degree $k-1$ over $\mathrm{GF}(q^u)$, such that the u coefficients $p_{k-1}, \ldots, p_{k-u}$ form a basis of $\mathrm{GF}(q^u)$ over $\mathrm{GF}(q)$. Compute $c = ev(p) \in RS_k$. Generate a random n-dimensional vector E of weight W with coefficients in $\mathrm{GF}(q^u)$. The public-key is the vector $K = c + E$ over $\mathrm{GF}(q^u)$. The private key is (p, E).

Encryption: Let m a message of length $k-u$ over the alphabet $\mathrm{GF}(q)$. The message m is seen as a polynomial $m(X) = m_0 + m_1 X + \ldots + m_{k-1} X^{k-u-1}$ in $\mathrm{GF}(q)[X]$. Generate a random $\alpha \in \mathrm{GF}(q^u)$ and a random vector e of weight ω over $\mathrm{GF}(q)$. The ciphertext y is then:

$$y = ev(m) + \mathrm{Tr}(\alpha \cdot K) + e$$

Decryption: One considers only the positions where $E_i = 0$ and define the shortened code of length $n - W$, which is also a Reed-Solomon code of dimension k, which we denote $\overline{RS}_k$. Let $\overline{y}, \overline{c}, \overline{e}$ be the shortened y, c, e and let $\overline{ev}$ be the shortened map ev. One must solve the equation:

$$\overline{y} = \overline{ev}(m) + \mathrm{Tr}(\alpha \cdot \overline{c}) + \overline{e}$$

Using proposition 1, we have:

$$\mathrm{Tr}(\alpha \cdot \overline{c}) = \mathrm{Tr}(\alpha \cdot \overline{ev}(p)) = \mathrm{Tr}(\overline{ev}(\alpha p)) = \overline{ev}(\mathrm{Tr}(\alpha p))$$

Thus $\overline{ev}(m) + \mathrm{Tr}(\alpha \cdot \overline{c}) = \overline{ev}(m + \mathrm{Tr}(\alpha p)) \in \overline{RS}_k$, and from (7), the weight of the small error $\overline{e}$ is less than the error correction capacity of $\overline{RS}_k$; therefore, using the Berlekamp-Welsh algorithm, one can recover the polynomial $q = m + \mathrm{Tr}(\alpha p)$.

Letting $q = \sum_{i=0}^{k-1} q_i x^i$, since $\deg(m) \leq k - u - 1$, we have $q_i = \mathrm{Tr}(\alpha p_i)$ for $i = k-u, \ldots, k-1$. This gives the u coordinates of α in the dual basis of $p_{k-u}, \ldots, p_{k-1}$, from which we derive α. From α one recovers m as $m = q - \mathrm{Tr}(\alpha p)$.

5 The Attack

In this section, we describe an attack that breaks the repaired cryptosystem. Given the public key and a ciphertext, we recover the plaintext in polynomial time. As the

[2] Actually, the authors of [2] forgot to clearly specify k, but they state that with these parameters, "a plaintext consists of $k - u$ elements in $\mathrm{GF}(2^{20})$, that is 27920 bits", from which we infer that $k = 27920/20 + 4 = 1400$

attack of section 3, it is a variant of the Berlekamp-Welsh algorithm. As opposed to the attack of section 3, the attack is heuristic, but it works very well in practice.

Let $\mathrm{GF}(q^u)$, n, k, W, ω be the parameters of the system. Let (p, E) be the private key and $K = ev(p) + E$ be the public-key. Let m be the plaintext encoded as a polynomial of degree less than $k - u - 1$. Let e be an error vector of weight ω, and $\alpha \in \mathrm{GF}(q^u)$. Let

$$y = ev(m) + \mathrm{Tr}(\alpha \cdot K) + e$$

be the corresponding ciphertext.

Let $\gamma_1, \ldots, \gamma_u$ be a basis of $\mathrm{GF}(q^u)$ over $\mathrm{GF}(q)$. We write $\alpha = \sum_{t=1}^{u} \alpha_t \cdot \gamma_t$ where $\alpha_t \in \mathrm{GF}(q)$. We have

$$\mathrm{Tr}(\alpha \cdot K) = \sum_{t=1}^{u} \alpha_t \mathrm{Tr}(\gamma_t \cdot K)$$

For $t = 1, \ldots, u$, we define

$$K_t = \mathrm{Tr}(\gamma_t \cdot K)$$

Note that the u vectors K_t are vectors over $\mathrm{GF}(q)$ which can be computed from the public-key K. Finally the ciphertext can be written as:

$$y = ev(m) + \sum_{t=1}^{u} \alpha_t \cdot K_t + e \tag{8}$$

Note that in equation (8), all computation is done in the subfield $\mathrm{GF}(q)$. Let $y_i, K_{t,i}$ and e_i be the components of the vectors y, K_t and e. Given y and K_t, one must solve the following set of equations:

$$\exists e, m, \alpha_1, \ldots, \alpha_u,\ y_i = m(x_i) + \sum_{t=1}^{u} \alpha_t \cdot K_{t,i} + e_i \text{ for all } 1 \le i \le n \tag{9}$$

where the weight of e is ω. Note that from the definition of the cryptosystem, there is a unique solution.

Let V, $R_1, \ldots, R_u$ be polynomials of degree at most ω, with $V \neq 0$. Let N be a polynomial of degree at most $\omega + k - u - 1$. Consider the following set of equations, where the unknown are the polynomials V, $R_1, \ldots, R_u$ and N:

$$\forall i \in [1, n], \quad V(x_i) \cdot y_i = N(x_i) + \sum_{t=1}^{u} K_{t,i} \cdot R_t(x_i) \tag{10}$$

It is clear that given a solution to system (9), one can obtain a non-zero solution to system (10). Namely, one can take $V(X) = \prod_{i \in B}(X - x_i)$ with $B = \{i | e_i \neq 0\}$, and $R_t = \alpha_t \cdot V$ for $t = 1, \ldots, u$ and $N = m \cdot V$. This shows that the system (10) has at least a non-zero solution.

The system (10) gives a homogeneous linear system of n equations in the $k + (u + 2) \cdot \omega + 1$ unknowns, which are the coefficients of the polynomials V, $R_1, \ldots, R_u$ and N. Let M be the matrix of the corresponding system. The matrix has $k + (u + 2) \cdot \omega + 1$

columns and n rows and can be computed from the ciphertext and the public-key. In the following, we assume that:

$$n \geq k + (u+2) \cdot \omega \tag{11}$$

This inequality is valid for the proposed parameters. Since the system (10) has at least a non-zero solution, the matrix cannot be of maximum rank, therefore $rank\ M \leq k + (u+2) \cdot \omega$.

In the following, we assume that $rank\ M = k+(u+2)\cdot\omega$. This is the only assumption that we make for our cryptanalysis. It seems that in practice, this assumption is always satisfied. In this case, the kernel of M is a linear space of dimension 1. We have already seen that $V(X) = \prod_{i \in B}(X - x_i)$ with $B = \{i | e_i \neq 0\}$, and $R_t = \alpha_t \cdot V$ for $t = 1, \ldots, u$ and $N = m \cdot V$ is a solution to the system (10), and so $(V, R_1, \ldots, R_t, N)$ generates the kernel of M.

Therefore, if we compute by Gaussian elimination an element $(V', R'_1, \ldots, R'_u, N')$ in $\ker M$, we must have that $V' = \lambda \cdot V$, $R'_t = \lambda R_t$ for $t = 1, \ldots, u$ and $N' = \lambda \cdot N$ for some $\lambda \in \mathrm{GF}(q)$ with $\lambda \neq 0$. Therefore, we have $N' = \lambda \cdot N = \lambda \cdot m \cdot V = m \cdot V'$ and we can recover m by doing a polynomial division:

$$m = \frac{N'}{V'}$$

To summarize, assuming that $rank\ M = k+(u+1)\cdot\omega$, we recover the plaintext from the public-key and the ciphertext in polynomial time.

6 Practical Experiments

In appendix, we illustrate the attack for small parameters, in a simplified setting. We have also implemented our attack using Shoup's NTL library [12], against the full cryptosystem. The attack works well in practice. For the recommended parameters, it takes roughly 8 minutes on a single PC to recover the plaintext from the ciphertext and the public-key.

7 Discussion

In this section, we try to see if it is possible to modify the parameters of the scheme in order to resist to the previous attack. The only condition on the parameters for the attack to work is inequality (11). Therefore, one may try to increase k, u or ω while keeping n constant. In the following, we show that this is not possible. Namely, we describe another attack on the repaired cryptosystem that recovers the private-key from the public-key. The attack does not work for the recommended parameters, but applies for large u.

The attack is the following. Let $K = ev(p) + E$ be the public-key with n components, where $\deg p = k-1$ and the weight of E is W. The Berlekamp-Welsh algorithm for recovering p from K is the following: it looks for two polynomials V and N such that $\deg V = W$, $\deg N = k + W - 1$ and $V \neq 0$, such that:

$$\forall i \in [1, n], V(x_i) \cdot K_i = N(x_i)$$

This gives a homogeneous linear system of n equations in $k + 2 \cdot W + 1$ unknown. This system has a non-zero solution as we can take $V(X) = \prod_{i \in B}(X - x_i)$ with $B = \{i | E_i \neq 0\}$ and $N = p \cdot V$. Letting V, N be any non-zero solution, we have for at least $n - W$ values of i:

$$V(x_i) \cdot p(x_i) = N(x_i)$$

Therefore, if $n - W > k + W - 1$, which gives:

$$n \geq k + 2 \cdot W \tag{12}$$

the polynomials $V \cdot p$ and N must be equal, which enables to recover p as $p = N/V$.

As in the attack of section 5, from K we derive u vectors K_t for $t = 1, \ldots, u$ such that:

$$K_t = \mathrm{Tr}(\gamma_t \cdot K)$$

where $\gamma_1, \ldots, \gamma_u$ is a basis of $\mathrm{GF}(q^u)$ over $\mathrm{GF}(q)$. Then we have:

$$K_t = \mathrm{Tr}(\gamma_t \cdot (ev(p) + E)) = ev(\mathrm{Tr}(\gamma_t \cdot p)) + \mathrm{Tr}(\gamma_t \cdot E)$$

Letting $p_t = \mathrm{Tr}(\gamma_t \cdot p)$ and $E_t = \mathrm{Tr}(\gamma_t \cdot E)$, we can write:

$$\forall t \in [1, u], K_t = ev(p_t) + E_t$$

Therefore, we obtain a set of u vectors K_t which are evaluation of a polynomial p_t plus some error E_t. The key observation is that the errors occur is the same positions in all vectors E_t. This enables to derive the following improved attack: we look for a polynomial $V \neq 0$, $\deg V \leq W$ and polynomials $N_1, \ldots, N_u$, $\deg N_t \leq k + W - 1$ such that:

$$\forall i \in [1, n], \begin{cases} V(x_i) \cdot K_{1,i} = N_1(x_i) \\ \qquad \cdots \\ V(x_i) \cdot K_{u,i} = N_u(x_i) \end{cases}$$

We can take the same polynomial V for each $t \in [1, u]$ because the errors are in the same positions for all E_t. This gives a system of $u \cdot n$ equations in the $u \cdot k + (u+1) \cdot W + 1$ unknowns. Let M be the corresponding matrix. It has $u \cdot n$ rows and $u \cdot k + (u+1) \cdot W + 1$ columns. We assume that:

$$u \cdot n \geq u \cdot k + (u + 1) \cdot W \tag{13}$$

The system has a non-zero solution. Therefore, the matrix cannot be of maximum rank, therefore $rank\ M \leq u \cdot k + (u + 1) \cdot W$. In the following, we assume that $rank\ M = u \cdot k + (u + 1) \cdot W$. This makes our attack heuristic, but the heuristic works well in practice. In this case, as in section 5, the kernel of M is a linear space of dimension one, and given a solution $(V, N_1, \ldots, N_u)$, one can recover the polynomials p_t as $p_t = N_t/V$. A similar approach was already used in [4] for the decoding of interleaved Reed-Solomon codes.

The inequality (13) gives the following condition for the attack to work:

$$n \geq k + \frac{u + 1}{u} \cdot W$$

which is an improvement over (12). Note that for the recommended parameters in [2], the attack does not apply. Therefore, to prevent this attack, one must have:

$$n < k + \frac{u+1}{u} \cdot W \tag{14}$$

Then, combining inequality (14) with inequality (7) which is necessary to be able to decrypt, one must have:

$$n \geq k + 2 \cdot (u+1) \cdot \omega$$

which shows that condition (11) of the attack of section 5 is always satisfied. Therefore, there is no set of parameters which make the repaired cryptosystem secure against both attacks.

8 Conclusion

We have broken the repaired cryptosystem of [2]. Our attack recovers the plaintext from the ciphertext and the public-key in polynomial time. Therefore, the cryptosystem does not achieve one-wayness. Moreover, our attack works well in practice, as for the recommended parameters, one recovers the plaintext in a few minutes on a single PC.

References

1. D. Augot and M. Finiasz, *A Public Key encryption scheme based on the Polynomial Reconstruction problem*, Proceedings of Eurocrypt 2003, LNCS vol. 2656, Springer-Verlag.
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, 30 Sep 2003, `http://eprint.iacr.org/`.
3. E.R. Berlekamp and L.R. Welch, *Error correction for algebraic block codes.* US Patent 4 633 470, 1986.
4. D. Bleichenbacher, A. Kiayias and M. Yung, *Decoding of Interleaved Reed Solomon Codes over Noisy Data*, proceedings of ICALP 2003.
5. J.S. Coron, *Cryptanalysis of a public-key encryption scheme based on the polynomial reconstruction problem*, Cryptology ePrint Archive, Report 2003/036, 5 Mar 2003, `http://eprint.iacr.org/`.
6. P. Gemmell and M. Sudan, *Highly resilient correctors for multivariate polynomials*, Information Processing Letters, 43(4): 169–174, September 1992.
7. V. Guruswami and M. Sudan, *Improved deconding of Reed-Solomon and Algebraic-Geometric codes*, IEEE Transactions on Information Theory, 45:1757-1767, 1999.
8. A. Kiayias and M. Yung, *Cryptographic hardness based on the decoding of Reed-Solomon codes with applications*, Proceedings of ICALP 2002, LNCS 2380, pp 232-243, 2002.
9. A. Kiayias and M. Yung, *Cryptanalysis of the polynomial reconstruction based public-key cryptosystem of Eurocrypt 2003 in the optimal parameter setting*, available at `http://www.cse.uconn.edu/~ akiayias/`.
10. M. Naor and B. Pinkas, *Oblivious transfer and polynomial evaluation.* In ACM, editor, STOC 99, pp 245-254, 1999.
11. I.S. Reed and G. Solomon, *Polynomial codes over certain finite fields*, J. SIAM, 8:300-304, 1960.
12. V. Shoup, *NTL: A Library for doing Number Theory (version 5.3.1)*, publicly available at `www.shoup.net`.
13. V. Shoup, *A fast deterministic algorithm for factoring polynomials over finite fields of small characteristic*, in Proc. 1991 International Symposium on Symbolic and Algebraic Computation, pp. 14-21, 1991.

A A Toy Example

In this section we illustrate the attack for small parameters, in a simplified setting. As in section 5, we let $\gamma_1, \ldots, \gamma_u$ be a basis of $\mathrm{GF}(q^u)$ over $\mathrm{GF}(q)$. Letting K be the public-key, we let:

$$K_t = \mathrm{Tr}(\gamma_t \cdot K) \in \mathrm{GF}(q)^n$$

Given $\alpha \in \mathrm{GF}(q^u)$, we write $\alpha = \sum_{t=1}^{u} \alpha_t \cdot \gamma_t$ where $\alpha_t \in \mathrm{GF}(q)$. The ciphertext can then be written as:

$$y = ev(m) + \sum_{t=1}^{u} \alpha_t \cdot K_t + e \tag{15}$$

Note that the ciphertext is computed using only operations in $\mathrm{GF}(q)$.

To simplify the illustration, we randomly generate the vectors K_t in $\mathrm{GF}(q)$, without generating K as a public-key in $\mathrm{GF}(q^u)$. Then we encrypt using (15) and show how to recover m from y and the vectors K_t. The difference with the normal setting is that the generated K_t may not correspond to a valid public-key K.

We take $n = 8$, $k = 5$, $\omega = 1$, $u = 2$. We work modulo $q = 11$. We take $x_i = i$ for $i = 1, \ldots, 8$. We take $K_1 = (2, 5, 0, 4, 9, 0, 4, 0, 5, 4, 6)$ and $K_2 = (5, 6, 9, 10, 6, 8, 2, 9, 4, 5, 1)$. We take $m = 5 + x + 9x^2$. We take $\alpha_1 = 8$, $\alpha_2 = 3$, and $e = (0, 0, 0, 0, 0, 7, 0, 0, 0, 0, 0)$. The ciphertext is then:

$$y = ev(m) + \alpha_1 \cdot K_1 + \alpha_2 \cdot K_2 + e = (2, 2, 6, 6, 6, 3, 7, 0, 3, 5, 1)$$

The matrix of the linear system is:

$$M = \begin{bmatrix} 2 & 2 & 9 & 9 & 6 & 6 & 10 & 10 & 10 & 10 \\ 2 & 4 & 6 & 1 & 5 & 10 & 10 & 9 & 7 & 3 \\ 6 & 7 & 0 & 0 & 2 & 6 & 10 & 8 & 2 & 6 \\ 6 & 2 & 7 & 6 & 1 & 4 & 10 & 7 & 6 & 2 \\ 6 & 8 & 2 & 10 & 5 & 3 & 10 & 6 & 8 & 7 \\ 3 & 7 & 0 & 0 & 3 & 7 & 10 & 5 & 8 & 4 \\ 7 & 5 & 7 & 5 & 9 & 8 & 10 & 4 & 6 & 9 \\ 0 & 0 & 0 & 0 & 2 & 5 & 10 & 3 & 2 & 5 \\ 3 & 5 & 6 & 10 & 7 & 8 & 10 & 2 & 7 & 8 \\ 5 & 6 & 7 & 4 & 6 & 5 & 10 & 1 & 10 & 1 \\ 1 & 0 & 5 & 0 & 10 & 0 & 10 & 0 & 0 & 0 \end{bmatrix}$$

The kernel of the matrix modulo q is generated by the vector:

$$(3, 5, 2, 7, 9, 4, 4, 6, 10, 1)$$

which gives $V(x) = 3 + 5 \cdot x$ and $N(x) = 4 + 6x + 10x^2 + x^3$ and eventually:

$$m = \frac{N}{V} \bmod q = 5 + x + 9x^2$$