Cryptanalysis of a 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. We describe a cryptanalysis of a public-key encryption scheme based on the polynomial reconstruction problem . Given the public-key and a ciphertext, we recover the corresponding plaintext in polynomial time. Therefore, the scheme is not one-way. Our technique is a variant of the Berlekamp-Welsh algorithm.

1 Introduction

We describe a cryptanalysis of a public-key encryption scheme recently proposed by Augot and Finiasz [1]. The scheme is based on the polynomial reconstruction (PR) problem [6], 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 [7]. The problem can be solved in polynomial time when the number of errors ω is such that $\omega \leq (n-k)/2$, using the Berlekamp-Welsh algorithm [2]. This has been improved to $\omega \leq n - \sqrt{kn}$ by Guruswami and Sudan [4].

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 [6], and a semantically secure symmetric cipher [5].

A new public-key cryptosystem based on the PR problem has been recently introduced in [1] by Augot and Finiasz. A security level exponential in terms of the parameters was conjectured. However, we provide a complete cryptanalysis of the cryptosystem: given the public key pk and a ciphertext c, we recover the corresponding plaintext m in polynomial time. Therefore, the scheme is not one-way and can not be used in any application. Our technique is a variant of the Berlekamp-Welsh algorithm [2] for solving the PR problem.

2 The proposed cryptosystem

2.1 Reed-Solomon codes

As in [1], we briefly recall some basic definitions of Reed-Solomon codes. Let F_q be the finite field with q elements and let $x_1, \cdots, x_n$ be n distincts elements of F_q. We denote by ev the following map:

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

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.

The Polynomial Reconstruction problem and the Reed-Solomon decoding problem are actually completely equivalent. Both problems can be solved in polynomial time when $w \leq (n-k)/2$, using the Berlekamp-Welsh algorithm [2].

2.2 The new 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 error 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 $\deg(m) \le k-2$ 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 Our attack

Our attack is a variant of the Berlekamp-Welsh algorithm for solving the PR problem (see [3]). 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, we 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, we can take $V(X) = \prod_{i \in B}(X - x_i)$ with $B = \{i | e_i \neq 0\}$. The problem of solving (3) can thus be reduced to finding V, m, α satisfying (4). Consider now the following set of equations:

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

It is easy to see that any solution of (4) gives a solution to (5), as we 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:

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

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

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

The matrix $M(\lambda)$ is a rectangular matrix with n lines and $k + 2\omega + 1$ columns; from (1) we have $n > k + 2\omega + 1$. 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 us take $\lambda = 0$. Using Gaussian elimination, we compute the rank of the matrix $M(0)$. We distinguish 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, we 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.

Then, using any polynomial time finite-field factoring algorithm such as [9], 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, we recover 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 $\omega+k+1$ points; consequently, the two polynomials must be equal. This means that we can recover m by performing a polynomial division:

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

Therefore, we 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 $M(0).Y=0$, which gives 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:

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

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]. □

4 Practical experiments

In appendix, we illustrate the attack for small parameters. We have also implemented our attack using Shoup's NTL library [8]. The attack works well in practice. For the recommended parameters ($n=1024$, $k=900$, $\omega=25$, $W=74$, $q=2^{80}$), it takes roughly 30 minutes on a single PC to recover the plaintext from the ciphertext and the public-key.

[1] In this second case, we can also recover the private key (p,E). However, we never came across that case in practice.

5 Conclusion

We have broken the proposed cryptosystem. Our attack recovers the plaintext from the ciphertext and the public-key in polynomial time. We insist that only the ciphertext and the public-key are required; in particular, our attack is not a chosen-ciphertext attack, as we do not require a decryption oracle. 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.
2. E.R. Berlekamp and L.R. Welch, *Error correction for algebraic block codes*. US Patent 4 633 470, 1986.
3. P. Gemmell and M. Sudan, *Highly resilient correctors for multivariate polynomials*, Information Processing Letters, 43(4): 169–174, September 1992.
4. V. Guruswami and M. Sudan, *Improved deconding of Reed-Solomon and Algebraic-Geometric codes*, IEEE Transactions on Information Theory, 45:1757-1767, 1999.
5. 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.
6. M. Naor and B. Pinkas, *Oblivious transfer and polynomial evaluation*. In ACM, editor, STOC 99, pp 245-254, 1999.
7. I.S. Reed and G. Solomon, *Polynomial codes over certain finite fields*, J. SIAM, 8:300-304, 1960.
8. V. Shoup, *NTL: A Library for doing Number Theory (version 5.3.1)*, available at `www.shoup.net`.
9. 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. We take $n = 8$, $k = 3$, $\omega = 1$, $W = 3$. We work modulo $q = 11$. We take $x_i = i$ for $i = 1, \ldots, 8$. We take:

$$\begin{aligned} p(x) &= x^2 + 5x + 3 \\ E &= (0, 0, 4, 0, 7, 6, 0, 0) \end{aligned}$$

for the private key. The public-key is:

$$z = ev(p) + E = (9, 6, 9, 6, 5, 9, 10, 8)$$

Let the message m be $m(x) = 8x + 2$. Let $\alpha = 7$ and $e = (0, 5, 0, 0, 0, 0, 0, 0)$. The ciphertext y is:

$$y = ev(m) + \alpha \times z + e = (7, 10, 1, 10, 0, 3, 7, 1)$$

The matrix $M(\lambda)$ is then:

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

The determinant $f(\lambda)$ of the matrix $M'(\lambda)$ obtained by taking the first 6 lines of $M(\lambda)$ is equal to:

$$f(\lambda) = \det M'(\lambda) = 3\lambda^2 + 5\lambda + 5$$

which factors modulo $q = 11$ into:

$$f(\lambda) = 3 \cdot (\lambda - 6) \cdot (\lambda - 7)$$

For $\lambda = 7$, we obtain $V(x) = 7x + 8$ and $N(x) = x^2 + x + 5$, which gives modulo $q = 11$:

$$m(x) = N(x)/V(x) = 8x + 2$$