Attack on A New Public Key Cryptosystem from ISC'02 (LNCS 2433)

Fangguo Zhang[1] Shengli Liu[2] and Kwangjo Kim[1]

[1] International Research center for Information Security (IRIS)
Information and Communications University(ICU),
58-4 Hwaam-dong Yusong-ku, Taejon, 305-732 KOREA
{zhfg, kkj}@icu.ac.kr

[2] Dept. of Computer Science and Engineering, Shanghai Jiao Tong University,
Shanghai 200030, P.R.China
liu-sl@cs.sjtu.edu.cn

Abstract. In ISC 2002, J. Zheng [8] proposed a new public key cryptosystem whose security is based upon the algebraic problem of reducing a high degree matrix to its canonical form by similarity transformations. In this paper, we show that factoring a polynomial over a finite field can be used to break down Zheng's public key cryptosystem. The complexity of our attack is polynomial time. In other word, the underlying problem of Zheng's public key cryptosystem is not a "hard" problem.

Key words: Public key cryptography, Attack, Polynomial, Matrix, Finite field.

1 Introduction

Since the public key cryptography had been introduced by Diffie and Hellman in 1976 [3], many cryptosystems have been put forth but only few of them have survived. In such a public-key system it must be computationally infeasible to deduce the decryption key from the public key, even when general information about the system and how it operates is known. Finding a new cryptosystem that overcomes deficiencies of existing ones is a challenging task of paramount importance.

Recently, a new public key cryptosystem for constrained hardware was proposed by J. Zheng in [8]. The public cryptosystem was claimed to be "self-sufficient" with a good speed, and expect to be the most efficient cryptosystems ever proposed. The underlying "hard" problem is reducing a high degree matrix to its canonical form by similarity transformations over a finite field. This problem is equivalent to solving a univariate polynomial with the same degree as the matrix. In this paper, we will show that this problem can be solved within polynomial time, *i.e.*, Zheng's public key cryptosystem is insecure.

2 Zheng's Public Key Cryptosystem

First of all, we briefly describe Zheng's public key cryptosystem. $x \in_R \mathcal{X}$ denotes the element x is randomly chosen from $\mathcal{X}$.

System parameter: p, r,and b_2 are system parameters, where p is a prime number, r is an integer chosen to be $r > 4$, and b_2 a r-dimension vector defined over $GF(p)$.

Secret key: Randomly chose $\lambda_1, \lambda_2, \ldots, \lambda_r$, from $GF(p)$, and chose a invertible matrix $H = (h_1, h_2, \ldots, h_r)$, where $h_i, 1 \le i \le r$ is randomly chosen from $GF(p)$.

Public key: The public key are A, a $r \times r$ matrix, and b_1, a r-dimension vector, all defined on $GF(p)$, where

$$A = H \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_r \end{pmatrix} H^{-1} (\mod p).$$

That means that $\lambda_1, \lambda_2, \ldots, \lambda_r$ be distinct eigenvalues of A, and $h_1, h_2, \ldots, h_r$ be the corresponding eigenvectors. b_1 is located in a subspaces spanned by some eigenvectors of A, it can be determined by

$$b_1 = (\alpha_1 h_1 + \alpha_1 h_1 + \cdots + \alpha_m h_m)(\mod p),$$

$1 < m < r$.

Encryption: Let $(z_1, z_2, \ldots, z_r) \in GF(p)^r$ be the plaintext, here z_i is a r-dimension vector. Choose $k_i \in_R GF(p)$, and

$$Y = (k_1 A^{r-1} + k_2 A^{r-2} + \cdots + k_r I)(\mod p).$$

Let $d = Y^2 b_1 + Y b_2$ (here b_2 is selected as a system parameter), and the ciphertext is given by (C, d), where

$$C = Y(z_1, z_2, \ldots, z_r).$$

Decryption: Let

$$(\delta_1, \delta_2, \ldots, \delta_r) = H^{-1} d(\mod p),$$

$$(\alpha_1, \alpha_2, \ldots, \alpha_r) = H^{-1} b_1(\mod p),$$

$$(\beta_1, \beta_2, \ldots, \beta_r) = H^{-1} b_2(\mod p).$$

Note that Y has the same eigenvector set as A, so we can suppose that

$$Y = H \begin{pmatrix} \mu_1 & & & \\ & \mu_2 & & \\ & & \ddots & \\ & & & \mu_r \end{pmatrix} H^{-1} (\mod p). \tag{1}$$

Then $\mu_i, i = 1, 2, \ldots, r$, can be solved from the following equations,

$$\delta_i = (\alpha_i \mu_i^2 + \beta_i \mu_i)(\mod p), i = 1, 2, \ldots, m,$$

$$\delta_i = (\beta_i \mu_i)(\mod p), i = m+1, m+2, \ldots, r.$$

When the eigenvalues of Y is determined, the plaintext $(z_1, z_2, \ldots, z_r)$ can be recovered by

$$z_i = Y^{-1} c_i = H \begin{pmatrix} \mu_1^{-1} & & & \\ & \mu_2^{-1} & & \\ & & \ddots & \\ & & & \mu_r^{-1} \end{pmatrix} H^{-1} c_i,$$

$i = 1, 2, \ldots, r$, and c_i is the ith column of C.

3 Attack and Analysis

As for a public key cryptosystem, it is very important that private key cannot be easily derived from public key. Here "easily" means no polynomial-time algorithm exists for this problem. However, in this section, we will show that the private keys $(\lambda_1, \lambda_2, \ldots, \lambda_r)$ and H can be easily obtained from the public key A.

Given a $r \times r$ matrix A, to find its r eigenvalues is equivalent to find the r solutions of the following equation

$$f(\lambda) = det(\lambda I - A)(\mod p) = 0, \tag{2}$$

where $det(\cdot)$ means the determinant of a matrix.

The left side of Equation (2) is a univariate polynomial of degree r defined over $GF(p)$. It is the characteristic polynomial of matrix A. So it is split completely over $GF(p)$. This means finding the solutions of $f(\lambda) = det(\lambda I - A)(\mod p) = 0$ is equivalent to factor the univariate polynomial of degree r over $GF(p)$.

In fact, the problem of factoring a polynomial of degree r over a finite field $GF(p)$ can be solved with $O(r^{1.815} \log p)$ arithmetic operations in $GF(p)$ according to [6]. As early as 1970, Berlecamp already proposed a random polynomial-time algorithm for such a problem. If multiplying two $r \times r$ matrices needs $O(r^w)$ arithmetic operations, then Berlecamp's algorithm needs $O(r^w + r^{1+o(a)} \log p)$ arithmetic operations [1]. It should be noted that the result of Coppersmith and Winograd's paper [2] in 1990 shows that $2 < w \leq 3$.

Let $f(x) \in GF(p)[x]$ has degree r. Factoring $f(x)$ goes with three steps according to [6]:

Step 1:Square-free factorization The input is a polynomial $f(x) \in GF(p)[x]$. The output is $f_1(x), f_2(x), \ldots, f_n(x)$ such that $f(x) = f_1(x) \cdot f_2(x)^2 \ldots f_n(x)^n$. (Here $f_i(x)$ are all square-free, *i.e.*, there is no polynomial $g(x)$ with degree ≥ 1 such that $g(x)^2$ divides $f(x)$)

Step 2: Distinct-degree factorization The input is $f(x) \in GF(p)[x]$of degree r. The output is $f^{[1]}, f^{[2]}, \ldots, f^{[n]} \in GF(q)[x]$, where $f^{[d]}, 1 \leq d \leq n$, is the product of the monic irreducible factors of $f(x)$ of degree d.

Step 3: Equal-degree factorization The input is a polynomial $f(x) \in GF(q)[x]$ of degree r and an integer d such that $f(x)$ is the product of distinct monic irreducible polynomials, each of degree d. The output is the set of irreducible factors of $f(x)$.

Step 1 uses $O(r^{1+o(1)} + r \log p)$ (with Yun's algorithm, see [7]), Step 2 uses $O(r^{1.844} \log p)$(see [6]), and Step 3 uses $O(r^{1.688} + r^{1+o(1)} \log p)$ operations over $GF(p)$(see [5]).

As for the case of solving Equation (2), we know that A has r distinct eigenvalues. Therefore, the polynomial $f(\lambda)$ of degree r determined by $det(\lambda I - A)(\mod p) = 0$ has r distinct solutions over $GF(p)$. In other words, $f(x)$ is already a completely split and square-free polynomial over $GF(p)$. Consequently, here we only need to solve the equal-degree factorization problem. In [5], von zur Gathen and Shoup have given an algorithm to solve this problem with complexity of $O(r^{1.688} + r^{1+o(1)} \log p)$.

Above factoring algorithm is probabilistic polynomial time. Thanks to [4], for a completely split and square-free polynomial over $GF(p)$, under the extended Riemann hypothesis (ERH), it can be factored deterministically in polynomial time.

4 Implementation of Attack

With the analysis in the previous section, we attack Zheng's public key cryptosystem as follows:

S1 Input a $r \times r$ matrix A and prime numbers p.
S2 Compute $f(\lambda) = det(\lambda I - A)(\mod p)$.
S3 Factoring $f(\lambda)$ over $GF(p)$.
S4 Get $(\lambda_1, \lambda_2, \ldots, \lambda_r)$.
S5 Solve $(h_1, h_2, \ldots, h_r)$ from $(\lambda_i I - A)h_i = 0$.
S7 Output $H = (h_1, h_2, \ldots, h_r)$ and $(\lambda_1, \lambda_2, \ldots, \lambda_r)$.

We implemented this attack with Maple 7 on PIII 650 MHz, and illustrate the average time of breaking down the cryptosystem in Table 1.

$r \setminus \lvert p \rvert$	$1024(bits)$	$2048(bits)$	$3000(bits)$	$5000(bits)$
5	$34s$	$4m$	$9m$	$43m$
7	$1m$	$7.5m$	$17m$	$1.3h$
10	$1.6m$	$13m$	$38m$	$2.8h$
20	$3.8m$	$24m$	$1h$	$6h$

Table 1. Timing on PIII 650

5 Conclusion

In this letter, we break down a new public key cryptosystem proposed recently in [8] with the known method of factoring polynomial over a Field $GF(p)$. With our attack the private key $(\lambda_1, \lambda_2, \ldots, \lambda_r)$ can be derived from public key A with polynomial time.

References

1. E.R. Berlecamp, *Factoring polynomials over large finite fields*, Proc. 22nd IEEE Symp. Foundations Comp. Sci., pp. 713-735, 1970.
2. D. Coppersmith and S. Winograd, *Matrix multiplication via arithmetic progressions*, J. Symbolic Comput. Vol.9, no.3, pp. 251–280, 1990.
3. W. Diffie and M.Hellman, *New Directions in Cryptography*, IEEE Transactions on Information Theory, IT-22(6), pp.644-654, 1976.
4. S. Gao, *On the deterministic complexity of polynomial factoring*, J. Symbolic Computation 31 (2001), pp.19-36.
5. J. von zur Gathen and V. Shoup, *Computing Frobenius maps and factoring polynomials*, Comput. Complexity, Vol.2, pp. 187-224, 1992.
6. E. Kaltofen and V. Shoup, *Subquadratic-Time Factoring of Polynomials over Finite Fields*, Proceeding 27th Annual ACM Symp. Theory of Computing, ACM Press, pp. 298–406, 1995.
7. D.E. Knuth, *The art of computer programming*, vol. 2, Seminumerical algorithms, Ed. 2, Addison Wesley, Reading, MA, 1981.
8. J. Zheng, *A new public key cryptosystem for constrained hardware*, Proceeding of ISC 2002, LNCS 2433, Springer-Verlag, pp.334-341, 2002.