

# Fast Cryptanalysis of the Matsumoto-Imai Public Key Scheme

*P. Delsarte*

Philips Research Laboratory,  
Avenue Van Becelaere, 2  
B-1170 Brussels, Belgium

*Y. Desmedt*

Katholieke Universiteit Leuven,  
Laboratorium ESAT, Kardinaal Mercierlaan, 94  
B-3030 Heverlee, Belgium

*A. Odlyzko*

AT&T Bell Laboratories  
Murray Hill, New Jersey 07974, U.S.A.

*P. Piret*

Philips Research Laboratory,  
Avenue Van Becelaere, 2  
B-1170 Brussels, Belgium

## ABSTRACT

The Matsumoto-Imai public key scheme was developed to provide very fast signatures. It is based on substitution polynomials over  $GF(2^m)$ . This paper shows in two ways that the Matsumoto-Imai public key scheme is very easy to break. In the faster of the two attacks the time to cryptanalyze the scheme is about proportional to the binary length of the public key. This shows that Matsumoto and Imai greatly overestimated the security of their scheme.

# Fast Cryptanalysis of the Matsumoto-Imai Public Key Scheme

*P. Delsarte*

Philips Research Laboratory,  
Avenue Van Becelaere, 2  
B-1170 Brussels, Belgium

*Y. Desmedt*

Katholieke Universiteit Leuven,  
Laboratorium ESAT, Kardinaal Mercierlaan, 94  
B-3030 Heverlee, Belgium

*A. Odlyzko*

AT&T Bell Laboratories  
Murray Hill, New Jersey 07974, U.S.A.

*P. Piret*

Philips Research Laboratory,  
Avenue Van Becelaere, 2  
B-1170 Brussels, Belgium

## 1. INTRODUCTION

Several attempts have been made to use the fields  $GF(2^m)$  [1] in cryptography. The motivation is that these fields allow very fast computation and are very easy to implement in hardware [2]. However, many such attempts quickly yielded to cryptanalytic attacks. For example, an extension of the RSA scheme to the fields  $GF(2^m)$  [3] was immediately broken [4,10]. The security of the fields  $GF(2^m)$  in public key distribution systems [5] was also overestimated [6]. Cryptanalysis is possible there if the dimension  $m$  of the field  $GF(2^m)$  is less than 1000 [6].

The Matsumoto-Imai public key scheme [7] also uses the fields  $GF(2^m)$ . It allows generation of signatures much faster than the RSA scheme. Moreover the scheme is very easy to implement. However, in this paper we give two efficient algorithms to cryptanalyze the Matsumoto-Imai public key scheme.

First the details of the Matsumoto-Imai scheme are presented, based on our interpretation of [7]. Then an overview of the first and second cryptanalytic attack are given. Both attacks use the public knowledge of the construction algorithm for public keys, and find secret parameters used in the

construction of the public key. These algorithms are then presented in detail.

## 2. THE MATSUMOTO-IMAI PUBLIC KEY SCHEME

The Matsumoto-Imai [7] enciphering is defined over  $GF(2^m)$ , the message space. The public key is a substitution polynomial [1]:  $E(X) = \sum_{i=0}^{2^m-2} e_i X^i$ . For a message  $Y$ , which belongs to  $GF(2^m)$ , the ciphertext is  $E(Y)$ . In order to have a "short" public key and to be able to encipher rapidly, most of the  $e_i$  must be zero. To that end,  $E(X)$  is constructed as  $E(X) \equiv a(b+X^\alpha)^\beta$  modulo  $(X^{2^m}+X)$ , where the Hamming weight [2] of  $\beta$  is  $r$ . One can then easily prove that only  $2^r$  coefficients  $e_i$  will be non-zero. If  $r$  is small (e.g., 14, as suggested in [7]), the public key is not too long.  $E(X)$  in expanded form is made public, while  $a, b, \alpha$ , and  $\beta$  are kept secret in order to be able to decipher fast. In order to specify  $E(X)$ , the field  $GF(2^m)$  has to be made public also, and  $r$  can be deduced from the number of non-zero coefficients. Therefore we have:

*Remark: One can consider that  $m$  and  $r$  are given, so these values do not have to be deduced.*

## 3. MAIN PRINCIPLES FOR THE CRYPTANALYSIS OF THE MATSUMOTO-IMAI PUBLIC KEY SCHEME

In order to allow a unique deciphering, the system designer has to chose  $\gcd(\alpha, 2^m-1) = \gcd(\beta, 2^m-1) = 1$ , and so from now on we will assume these conditions hold. The following theorems help to explain the cryptanalysis.

*Theorem 1: If the public key is constructed as mentioned above and  $\beta$  is written as*

$$\beta = \sum_{j=1}^r 2^{u_j}, \quad \text{with } 0 \leq u_j < m, \quad (1)$$

*then the exponents of  $X$  with non-zero coefficients can be expressed as*

$$\alpha \sum_{j=1}^r z_j 2^{u_j} \pmod{2^m-1}, \quad \text{with } z_j = 0 \text{ or } 1, \quad (2)$$

*and their corresponding coefficients as*

$$ab^k \text{ with } k \equiv \sum_{j=1}^r (1-z_j)2^{u_j} \pmod{2^m-1}. \quad (3)$$

*Proof:* Using the construction algorithm for public keys and (1), we have  $E(X) = a \prod_{j=1}^r (b + X^{\alpha_j})^{2^{u_j}}$ . Since the characteristic of  $GF(2^m)$  is 2,  $E(X) = a \prod_{j=1}^r (b^{2^{u_j}} + X^{\alpha_j 2^{u_j}})$ , and using  $X^{2^m} \equiv X \pmod{X^{2^m} + X}$  we obtain (2) and (3). ■

*Corollary 1:* At least  $m$  different  $(a, b, \alpha, \beta)$  determine the same enciphering key.

*Proof:* Choose  $\alpha' \equiv 2^h \alpha$  and  $\beta' \equiv 2^{m-h} \beta \pmod{2^m-1}$  and use the proof of Theorem 1. ■

It is sufficient to find any one of these equivalent  $(a, b, \alpha, \beta)$  in order to break the scheme. To simplify the description, all these equivalent keys will be called the secret key. We will sometimes suppose in the paper that  $u_1 = 0$ , which by Corollary 1 entails no loss of generality.

*Corollary 2:* If  $u_1 = 0$  then  $b = (\text{coefficient of } X \text{ to the power } 0)/(\text{coefficient of } X \text{ to the power } \alpha)$  and  $a = (\text{coefficient of } X \text{ to the power } 0)/b^\beta$ .

*Proof:* Can be verified easily using Theorem 1. ■

*Theorem 2:* If  $\gcd(\alpha, 2^m-1) = 1$  and  $\gcd(\beta, 2^m-1) = 1$ , then the list of exponents of  $X$  in  $E(X)$  with nonzero coefficients contains a unique subset of size  $r$  of the form  $\{2^{v_1}\alpha_1, 2^{v_2}\alpha_1, \dots, 2^{v_r}\alpha_1\} \pmod{2^m-1}$ . Taking Corollary 1 into account one has  $\alpha = \alpha_1$  and  $\beta = \sum 2^{v_j}$ .

*Proof:* In view of Theorem 1 this subset is actually present in the list of exponents. Let now  $\gamma$  be any other element of the list, say  $\gamma \equiv (2^{p_1} + 2^{p_2} + \dots + 2^{p_s})\alpha \pmod{2^m-1}$ , where  $\{p_1, p_2, \dots, p_s\}$  is a subset of  $\{u_1, u_2, \dots, u_r\}$  with  $s \geq 2$ . We shall prove that the list contains fewer than  $r$  elements of the form  $2^k \gamma \pmod{2^m-1}$ . First it is clear there cannot be more than  $r$  such elements, because for each  $i$ , each sum  $p_i + k \pmod{m}$  must coincide with one of the integers  $u_1, u_2, \dots, u_r$ . If there were exactly  $r$  elements, then one would necessarily have  $p_1 + k_j \equiv u_{\pi(j)}$  and  $p_2 + k_j \equiv u_{\sigma(j)} \pmod{m}$ , for  $j = 1, 2, \dots, r$ , where  $\pi$  and  $\sigma$  are two permutations on  $\{1, 2, \dots, r\}$ . Taking the binary exponential of these identities and adding the results together (for  $j = 1, 2, \dots, r$ )

one readily obtains  $\beta(2^{p_2 - p_1} - 1) \equiv 0 \pmod{2^m - 1}$ , which is impossible since  $\gcd(\beta, 2^m - 1) = 1$  and  $p_2 \neq p_1$ . ■

If one finds an  $\alpha_1$  which satisfies the above property, then  $\alpha$  will be chosen equal to  $\alpha_1$ . The calculation of  $\beta$  is then trivial; it is in fact obtained at the same time as  $\alpha$ . Once  $\alpha$  and  $\beta$  have been found,  $a$  and  $b$  are calculated using Corollary 2. As a consequence of Theorem 2 and the remark at the beginning of this section, one does not need to check that the obtained  $\alpha$  is correct! Two algorithms will now be presented which find  $\alpha$  and  $\beta$ . The first algorithm uses the calculation of the inverse of elements modulo  $2^m - 1$ . The second one is based on shift operations and sorting algorithms.

#### 4. CRYPTANALYSIS USING INVERSE CALCULATION

##### 4.1 The Principles Of The Algorithm

Exponents of  $X$  with non-zero coefficients will be written as  $i_k$ , with  $1 \leq k \leq 2^r$ . For each  $k$ ,  $1 \leq k \leq 2^r$ , we test whether  $\alpha = i_k$ . If  $\gcd(i_k, 2^m - 1)$  is not equal to one, a wrong choice for  $\alpha$  was made. If  $\gcd(i_k, 2^m - 1) = 1$  then several techniques can be used to find  $\beta$ . In one of them the cryptanalyst first calculates

$$f_\ell \equiv i_\ell i_k^{-1} \pmod{2^m - 1}, \quad 1 \leq \ell \leq 2^r. \quad (4)$$

In view of Theorem 2, if  $r$  values of  $f_\ell$  are powers of 2, then  $i_k$  is  $\alpha$ , and  $\beta$  is the sum of these  $r$  values of  $f_\ell$ . If no such  $r$  values are found, continue the exhaustive search. Because  $r$  is small this exhaustive search is fast.

##### 4.2 Speed Evaluation Of The Algorithm

The number of elementary steps (such as additions and shifts) used in the above cryptanalytic algorithm will be analyzed. First the complexity of each step will be obtained; next this value will be multiplied by the number of times each step is executed.

The calculation of the  $\gcd(i_k, 2^m - 1)$  and the calculation of  $i_k^{-1}$ , if it exists, can be done at once. This requires  $O(m)$  steps [8] (subtractions or shifts). This means in total  $O(m2^r)$  steps during the

exhaustive search for  $\alpha$ . The calculation of (4) requires in practice  $O(m^2)$  [8] elementary steps (additions and small multiplications). For larger values of  $m$  better algorithms (e.g., using the FFT) can be used [8]. This means for the exhaustive search that (4) is executed in worst case in  $O(m^2 2^{2r})$  steps, while on average it takes  $O(m^2 2^{2r}/r)$  steps. The calculation of  $\beta$  requires for each trial  $O(\log_2 m)$  steps. We conclude that the cryptanalysis requires  $O(m^2 2^{2r}/r)$  steps on average, and  $O(m^2 2^{2r})$  in the worst case. The next algorithm has an improved speed performance.

## 5. CRYPTANALYSIS USING SHIFT OPERATIONS

### 5.1 The Principles Of The Algorithm

In the second method of attack we partition the exponents of  $X$  with nonzero coefficients into sets  $S_p$ . Two different exponents  $i_k$  and  $i_l$  of  $X$ , with non-zero coefficients, will belong to the same set  $S_p$  if and only if for some  $s$ ,  $i_k \equiv 2^s i_l \pmod{2^m - 1}$ . In other words,  $i_k$  and  $i_l$  belong to the same set  $S_p$  if one can obtain  $i_k$  from  $i_l$  by a suitable rotation of its binary representation. The cryptanalysis consists of determining all different sets  $S_p$  for all exponents of  $X$  with non-zero coefficients. Using Theorem 2 exactly one  $S_p$ , which we call  $S_r$ , will contain  $r$  elements. Using Corollary 2, any element of  $S_r$  can be chosen as  $\alpha$ . Identifying the required rotation operations for going from  $\alpha$  to obtain the other elements of  $S_r$ , we obtain  $\beta$ . We now describe how the above ideas can be carried out. The speed of the algorithm will be discussed later.

First we define a unique representative for each set  $S_p$ . A value  $v_p$  is the representative of the set  $S_p$  if it is the smallest of the  $m$  values obtained by rotating  $0, 1, 2, \dots, m-1$  times an element of the set  $S_p$ . Note that  $v_p$  can be viewed as the value of a function  $v(i)$  defined over the set  $\{0, 1, \dots, 2^m - 2\}$  and satisfying  $v(i_1) = v(i_2)$  if and only if  $i_1 \equiv 2^s i_2 \pmod{2^m - 1}$ , for a certain  $s$ . This function  $v(i)$  will now be used to find  $\alpha$ . We calculate  $v(i_k)$  for all  $2^r$  exponents  $i_k$  of  $X$  with non-zero coefficients. The  $v(i_k)$  and  $i_k$  together are written in lists A and B of  $2^r$  elements, in which each element contains  $m$  bits. There is a unique element  $w$  that appears  $r$  times in list A, and then  $\alpha$  can be chosen as any of the corresponding elements in list B. This search for  $w$  and  $\alpha$  can easily be performed by sorting [9, pp. 2] the list A while simultaneously permuting the list B in

the same way.

## 5.2 Speed Evaluation Of The Algorithm

The calculation of  $v(i)$  requires  $m$  steps. Doing this for all the exponents of  $X$  that appear in  $E(X)$  takes  $m2^r$  steps. The sorting of list A, together with the permutation of list B, requires  $O(r2^r)$  steps in practice [9, pp. 181-198, p. 381].

In total this algorithm requires  $O(m2^r)$  steps even in the worst case!

## 6. CONCLUSIONS

The Matsumoto-Imai public key scheme seems attractive from speed considerations, *but is totally insecure!* Matsumoto and Imai estimated the cryptanalysis of their scheme would require about  $10^{20}$  steps if  $m = 127$  and  $r = 14$ . However using our cryptanalytic attack using inverse calculation, on the same example, one needs only about  $3 \cdot 10^{11}$  steps, which is performable even on a small computer. If one step requires  $10 \mu\text{sec}$  on a small computer, then the attack requires 36 days. If one step asks  $100 \text{ nsec}$  on a fast computer, the attack can be performed in only 8 hours. Using the cryptanalysis based on shift operations, one needs only  $2 \cdot 10^6$  steps. Using the same small and fast computer, this requires 20 sec and 0.2 sec, respectively. On a fast computer, the cryptanalytic attack suggested by Matsumoto and Imai would require  $3 \times 10^5$  years.

*Remark: The second cryptanalytic attack requires about as many steps as the binary length of the public key!*

One could increase the security of the Matsumoto-Imai scheme by increasing  $m$  and  $r$ . However, even disregarding the fact that this might entail impractically large storage requirements, this would not produce an acceptable system. Evaluation of the publicly known function  $E(X)$  takes at least  $2^r$  multiplications in  $GF(2^m)$ , and each such multiplication might be expected to take about  $m$  operation such as the shifts we utilize in our second attack. *Hence the time needed to cryptanalyze the Matsumoto-Imai system is essentially the same as the time needed to use it once!*

## REFERENCES

1. R. Carmichael, *Introduction to the Theory of Groups of Finite Order*, Dover, New York, 1956.
2. E. R. Berlekamp, *Algebraic Coding Theory*, McGraw-Hill, 1968.
3. D. W. Kravitz and I. S. Reed, Extension of RSA Crypto-Structure: A Galois Approach, *Electronics Letters*, vol. 18, no 6, pp. 255-256, 18th March 1982.
4. P. Delsarte and P. Piret, Comment: Extension of RSA Crypto-Structure: A Galois Approach, *Electronics Letters*, vol. 18, no. 13, pp. 582-583, 24th June 1982.
5. K. Yiu and K. Peterson, A Single-Chip VLSI Implementation of the Discrete Exponential Public Key Distribution System, *Proc. Globecom '82 IEEE Global Telecommunications Conference*, vol. 1, pp. 173-179, Miami, USA, 1982.
6. D. Coppersmith, Fast Evaluation of Logarithms in Fields of Characteristic Two, *Research Report*, RC 10187 IBM Yorktown Heights. *IEEE Trans. Inform. Theory*, to appear.
7. T. Matsumoto and H. Imai, A Class of Asymmetric Crypto-Systems based on Polynomials over Finite Rings, *IEEE Intern. Symp. Inform. Theory, St. Jovite, Quebec, Canada*, September 26-30, 1983, Abstracts of Papers, pp. 131-132.
8. D. E. Knuth, *The Art of Computer Programming, Vol. 2, Seminumerical Algorithms*, Addison-Wesley, Reading, Massachusetts, 1981.
9. D. E. Knuth, *The Art of Computer Programming, Vol. 3, Sorting and Searching*, Addison-Wesley, Reading, Massachusetts 1975.
10. J. Gait, Short Cycling in the Kravitz-Reed Public Key Encryption System, *Electronics Letters*, vol. 18, no. 16, pp. 706-707, 5th August 1982.