Cryptanalysis of Imai and Matsumoto Scheme B Asymmetric Cryptosystem

Amr Youssef[1] and Guang Gong[2]

[1] Center for Applied Cryptographic Research,
Department of Combinatorics & Optimization,
University of Waterloo, Waterloo, Ontario N2L 3G1, Canada,
[2] Center for Applied Cryptographic Research,
Department of Electrical and Computer Engineering,
University of Waterloo, Waterloo, Ontario N2L 3G1, Canada,
{a2youssef,ggong}@cacr.math.uwaterloo.ca

Abstract. Imai and Matsumoto introduced alternative algebraic methods for constructing public key cryptosystems. An obvious advantage of theses public key cryptosystems is that the private side computations can be made very efficient with a simple hardware. Almost all of these proposals and variants of them were broken. However, scheme "B" in [3] is still unbroken. In this paper we show some statistical weaknesses of this scheme. In particular, we show that trying to minimize the size of the public key facilitates a cryptanalytic attack that enables the cryptanalyst to decrypt, with high probability of success, a given ciphertext by performing a very limited number of encryption operations using the public encryption function.

Keywords: Public-key cryptosystems , cryptanalysis, Imai and Matsumoto asymmetric cryptosystems

1 Introduction

Public key cryptosystems based on integer factorization and discrete log problem, such as RSA and ElGamal [7], need to perform a large amount of arithmetic operations, so they are not very efficient compared to symmetric key cryptosystems such as DES. Imai and Matsumoto [3] [6] and Matsumoto *et. al.* [5] introduced alternative algebraic methods for constructing public key cryptosystems. An obvious advantage of theses public key cryptosystems is that the private side computations (decrypting and signing) can be made very efficient with a simple hardware. Almost all of these proposals and variants of them were broken (see [1], [2], [8], [9] [10] [11] [12]). However, as noted in [2], scheme "B" in [3], which was originally proposed by Matsumoto *et. al.* in [5] is still unbroken. In this paper we introduce a piecewise affine approximation attack on this scheme and show that it is insecure.

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 214–222, 2001.

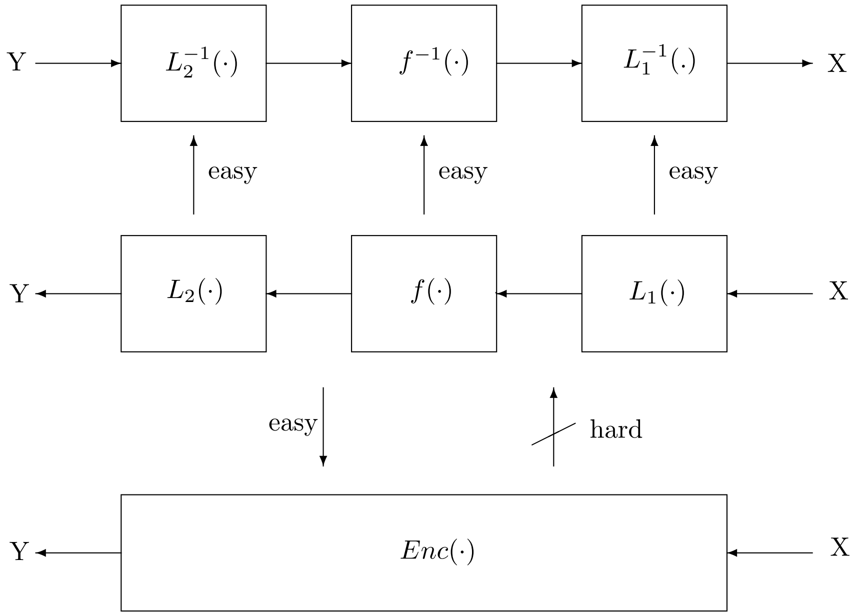

Fig. 1. The basic idea of Imai and Matsumoto Scheme B

2 Description of Scheme "B" in [3]

For a given block length n, the encryption function of Scheme "B" in [3] is composed of

$$L_1 \circ f \circ L_2 \tag{1}$$

where L_1 and L_2 are two secret bijective linear mappings over $GF(2)^n$ and

$$f(x) = \begin{cases} (x + c - 1)\text{mod}(2^n - 1) + 1, & x \neq 0 \\ 0, & x = 0, \end{cases} \tag{2}$$

where c is a secret positive integer whose binary representation has ***small*** Hamming weight, wt(c). The main reason to choose c with a small Hamming weight is to reduce the size of the public key [3]. The encryption of x is given by

$$Enc(x) = L_2(f(L_1(x))).$$

The private key is L_1, L_2 and c. The public key is an AND-XOR array pattern for the m-tuple of m-variate sparse polynomials over $GF(2)$ representing the composite function $Enc(\cdot)$. As mentioned above, small values for $wt(c)$ is required to reduce the public key size. This restriction is the basic motive for our attack. The security of this scheme (see Figure 1) relies on the fact that the transformations L_1, L_2 and f operate on two different algebraic structures ($GF(2)^n$ and the non-negative set of integers $< 2^n$) . Thus the $Enc^{-1}(\cdot)$ is assumed to have a complex representation when considered as a mapping over only one of these two structures. In other words, it is assumed that it is difficult to obtain any simple algebraic description for the function $Enc^{-1}(\cdot)$ given only the AND-XOR array of the function $Enc(\cdot)$. The size of the public and secret key bits and the complexity of the encryption and decryption operations are all $O(n^2)$.

3 Observations

Our attack is based on the following observation

Observation 1 *For a small Hamming weight of* c*, the piecewise affine approximation of the function* f *in equation (2) has small number of affine segments over* $GF(2)^n$ *compared to that of a randomly selected bijective mapping. Moreover, most of the points belong to a small number of segments, i.e., a small number of segments is enough to achieve a good approximation accuracy.*

Example 1. Let $n = 8$ and $c = 3$ with Hamming weight 2. Then for $x \neq 0$, the binary representation of f belong to one of the following piecewise affine functions

$$l_i(x) = x \oplus d_i$$

where $d_i \in \{3, 5, 7, 13, 15, 29, 31, 61, 63, 125, 127, 252, 253, 255\}$. The number of points on each segment is shown in Table 1. It is clear that the approximation accuracy using the first two constants is about 50%. Using the first 8 constants the accuracy increases to about 94%.

Table 1. The affine constants for Example 1

c	5	3	7	13	29	15	61	31	63	125	252	253	127	255
No. of points	64	63	32	32	16	16	8	8	4	4	3	2	2	1

Example 2. Let $n = 16$ and $c = 1056$ with Hamming weight 2, then more than 90% of the points corresponding to the binary representation of the function belong to one of the following the affine segments

$$l_i(x) = x \oplus d_i,$$

where

$$d_i \in \{15456, 2016, 3040, 31776, 3552, 7392, 1504,$$

$$15392, 3296, 7264, 1248, 3168, 7200, 1120, 3104, 1056\}.$$

Table 2 shows the expected number of segments for $n = 8, 10, 12$. The average number of segments for a randomly selected bijective mapping (obtained by our experiments) is around 176, 690 and 2778 for $n = 8, 10$ and 12 respectively. In

Table 2. Average Number of Segments in the piecewise approximation of f

wt(c)	1	2	3	4	5	6	7	8	9	10	11	12
n=8	9	19	28	31.63	28	19	9	1				
n=10	11	27.33	47.5	64.43	71.06	64.43	47.5	27.33	11	1		
n=12	13	37	73.4	114.4	147.17	159.72	147.17	114.4	73.4	37	13	1

all of our experiments with small values of $wt(c)$, there always exists an affine relation satisfied by $\geq 2^{(n-c)} - 1$ points.

Using Observation 1 the Encryption function (and consequently the decryption function) can be divided into M affine segments $Enc_i(\cdot)$, $i = 1, 2, \cdots M$ where $Enc_i(x)$ given by

$$Enc_i(x) = L_2(L_1(x) \oplus d_i) = L_2L_1(x) \oplus b_i,$$

and $b_i = L_2(d_i), i = 1, 2, \cdots M$. Thus for any specific L_1, L_2, c, the input (plaintext) space can be partitioned into M sets such that the ciphertext Y of each set is related to the plaintext X by an affine relation

$$Y = AX \oplus b_i,$$

where $A = L_2 \circ L1$. The expected value of M is small for c with small Hamming weight.

Remark 1. Probabilistic interpolation attacks [4] based on Sudan's algorithm [13], which operates over $GF(2^n)$, cannot be used to recover these affine segments since the affine function over $GF(2)^n$ will have a high degree when considered as a function over $GF(2^n)$ [14].

In the following, we will describe the basic step in the attack. We use a differential-like attack to group pairs that belong to the same segment. Figure 2 shows the algorithm that enables us to do so. We pick random triples R_1, R_2 and R_3 and test for the condition

$$Enc(R_1) \oplus Enc(R_2) \oplus Enc(R_3) \oplus Enc((R_3 \oplus (R_1 \oplus R_2))) = 0.$$

This condition is satisfied if $R1, R2, R3$ and $R1 \oplus R2 \oplus R3$ are all on one affine segment. Since there is no guarantee that $R_3, (R_3 \oplus (R_1 \oplus R_2))$ will belong to segment S_i even if R_1 and R_2 do, we repeat the test for different values of R_3 ($Trials$ in Figure 2). We decide that R_1 and R_2 belong to the same segment if the equation above is satisfied for a large number of times ($Threshold$ in Figure 2). To prevent the algorithm from accepting wrong pairs we may increase the value of $Trials$ and make the value of $Threshold$ very close to $Trials$. However, very large values for $Trials$ increases the number of plaintext-ciphertext pairs required to break the algorithm. Throughout our experiments, we set $Threshold = Trials$.

One can prove that the plaintext that belong to the same linear segment are not linearly independent and hence the matrix A cannot be uniquely determined

```
$R_1 = Random()$
$R_2 = Random()$
$pass = 0$
$\delta_x = R_1 \oplus R_2$
for $i = 1$ to $i = Trials$
{
    $R_3 = Random()$
    $R_4 = R_3 \oplus \delta_x$
    $\delta_y = Enc(R_1) \oplus Enc(R_2) \oplus Enc(R_3) \oplus Enc(R_4)$
    if $(\delta_y = 0)$ increment pass
}
if(pass $\geq Threshold$) Declare $R_1$ and $R_2 \in$ same set
```

Fig. 2. The Basic Step in the Attack

by collecting plaintext-ciphertext pairs on one segment. In fact, our experiments show that the matrix A cannot be uniquely determined by any reasonable number of queries to the encryption function. So our attack doesn't try to find such unique solution for A.

4 The Attack

Let x_1, x_2 be on the same affine segment S_i. Then

$$Enc(x_1) \oplus Enc(x_2) = Ax_1 \oplus b_i \oplus Ax_2 \oplus b_i = A(x_1 \oplus x_2)$$

which is independent of the segment they belong to and depends only on the difference $(x_1 \oplus x_2)$. Our attack proceeds as follows:

1. Use the basic step in Figure 2 to pick any two plaintext points $(x, x \oplus \delta_x)$ that are on the same segment. Collect enough number of (δ_x, δ_y) pairs for linearly independent δ_x's.
2. Solve for the matrix B that satisfy the linear relation

   $$\delta_x = B \times \delta_y$$

   The coverage of the attack (i.e., the probability of being able to decrypt a random ciphertext) increases exponentially with the number of pairs collected in step 1. (*Remark.* Note that $(L_2 L_1)^{-1}$ is not the only valid solution for B).

After determining this linear relation between δx's and δ_y's we can decrypt any given ciphertext u as follows:

1. Pick random x and assume that it is on the same segment with $Dec(u)$.

2. Calculate
$$TrialDec(u) = x \oplus B \times (u \oplus Enc(x)) \tag{3}$$
3. Using the public encryption function, verify if $Enc(TrialDec(u)) = u$.

If yes, then we have found $Dec(u)$. If no, then pick a different x and repeat the steps above.

Relation 3 holds if x and $Dec(u)$ belong to the same segment and this happens with high probability because we have a small number of segments.

One should note that deriving an accurate theoretical estimate for the number of encryption operations required to achieve certain coverage is difficult because the $Enc(\cdot)$ function doesn't behave like a random function. Table 3 and Table 4 show the result of some of our experiments for $wt(c) = 1, 2$ and $n = 16, 18, 20$. The tables show the number of queries (to the public encryption function) that are required to successfully decrypt more than 50% of a random sample of 100 ciphertext. Increasing the coverage close to 99% requires a slight increase in the number of collected pairs. For example, for $n = 20, wt(c) = 1$, only a total of 866 and 900 encryption operations were required to increase the coverage to 92% and 98% respectively. Let the fraction P denote the number of chosen plaintext-ciphertext pairs required to achieve certain coverage. Then , our experimental results show that, on average and for a fixed small Hamming weight of c, $P/2^n$ decreases dramatically with n.

Table 3. Experimental Results for $wt(c) = 1$

n	Number of Encryption Operations	Coverage
16	813	56%
18	670	66%
20	630	64%

Table 4. Experimental Results for $wt(c) = 2$

n	Number of Encryption Operations	Coverage
16	2418	61%
18	2605	51%
20	3525	61%

5 Conclusion

For some selections of the algorithm parameter c, the encryption and decryption operations in Scheme B proposed by *Imai et. al.* can be approximated by a piecewise affine function over $GF(2)^n$ with small number of affine segments. Trying to minimize the size of the public key by using a very small Hamming weight for the algorithm parameter c reduces the number of theses affine segments and may compromise the security of the algorithm. It should be noted that avoiding such selections for c, while may increase the size of the public key, makes the algorithm totally secure against our attack and the security of this scheme still remains an open problem.

References

1. E. Biham *Cryptanalysis of Patarin's 2-Round Public Key System with S Boxes (2R)*, Advances in Cryptology, Proceedings of EUROCRYPT'2000, LNCS 1807, pp. 408-416, Springer-Verlag, 2000.
2. Y. Feng, L. Yan and D. Duo *Cryptanalysis of "2R" Schemes*, Advances in Cryptology, Proceedings of CRYPTO'99, LNCS1666 , pp. 315-325, Springer-Verlag, 1999.
3. H. Imai and T. Matsumoto, *Algebraic methods for constructing asymmetric cryptosystems*, Proceedings of Algebraic Algorithms and error-correcting codes (AAECC-3), Springer-Verlag, LNCS 229 , pp. 108-119
4. T. Jakobsen, *Cryptanalysis of Block Ciphers with Probabilistic Non-linear Relations of L ow Degree*, Proceedings of CRYPTO'99, LNCS 1462, pp. 213-222, 1999.
5. T. Matsumoto, H. Imai, H. Harashima and H. Miyakawa *A high-speed asymmetric cryptosystem with obscure public-keys*, Paper of Technical group, TGAL84-83, Mar. 1985. (in Japanese)
6. T. Matsumoto and H. Imai *Public quadratic polynomial tuples for efficient signature verification and message encryption*, Advances in Cryptology, Proceedings of EUROCRYPT'88, LNCS330 , pp. 419-453, Springer-Verlag, 1988.
7. A J. Menezes, P. C. van Oorschot and S A. Vanstone *Handbook of Applied Cryptographic Research*, CRC Press, 1996.
8. J. Patarin, L. Goubin, and N. Courtois, *C*-+ and HM: Variations around two schemes of T. Matsumoto and H. Imai*, Advances in Cryptology, Proceedings of ASIACRYPT'98, Springer-Verlag, LNCS 1514, pp. 35-49, 1998.
9. J. Patarin and L. Goubin, *Asymmetric Cryptography with S-Boxes*, Proceedings of ICICS'97, Springer-Verlag, LNCS 1334, pp. 369-380, 1997.
10. J. Patarin, *Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt'88*, Advances in Cryptology, Proceedings of CRYPTO '95, Springer-Verlag, LNCS 963, pp. 248-261, 1995.
11. J. Patarin, *Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): two new families of Asymmetric Algorithms*, Advances in Cryptology, Proceedings of EUROCRYPT'96, Springer-Verlag, LNCS 1070, pp. 33-48, 1996
12. J. Patarin and L. Goubin, *Trapdoor one-way permutations and multivariate polynomials*, Advances in Cryptology, Proceedings of ICICS'97, Springer-Veralg, LNCS 1334, pp. 356-368, 1997.
13. M. Sudan, *Decoding Reed Solomon Codes beyond the error-correction bound*, Journal of Complexity, Vol. 13, no 1, pp180-193, March, 1997.
14. A. M. Youssef and G. Gong, On the interpolation attacks on block ciphers, Proceedings of *Fast Software Encryption 2000* , Springer-Veralg, LNCS 1978, pp. 109-120, 2001 .

Appendix: A Detailed Example

In this section we give a detailed example for our attack on a toy version with $n = 20, C = 4096, wt(C) = 1$. L_1 and L_2 are given by

$$\begin{bmatrix}
1&0&0&0&0&1&1&1&1&1&1&1&0&1&1&0&1&0&0&0\\
0&0&1&0&0&0&1&0&1&0&1&1&0&0&0&1&0&1&0&0\\
0&1&1&1&0&1&1&1&1&1&0&1&1&1&0&1&1&0&1&1\\
0&1&0&0&0&0&1&0&0&0&0&1&1&1&1&0&1&1&0&1\\
0&0&1&1&1&1&0&0&1&0&1&0&0&0&1&1&1&1&0&1\\
1&1&0&1&1&1&1&0&0&0&0&0&0&0&0&0&1&0&1&1\\
0&1&1&1&0&0&0&0&1&0&1&1&1&1&1&0&0&0&0&0\\
0&0&0&1&1&1&0&1&1&1&0&1&0&0&0&1&1&1&1&0\\
1&1&1&0&0&1&0&0&0&0&0&0&0&1&1&0&0&1&1&1\\
1&1&0&0&1&0&1&0&0&0&0&1&1&1&1&0&0&0&1&0\\
0&0&0&0&1&1&1&0&1&1&0&1&0&1&1&0&1&0&0&1\\
0&1&1&1&1&0&1&0&1&0&0&0&0&0&1&1&1&0&1&0\\
1&0&1&0&0&0&0&1&0&1&0&1&0&1&1&0&0&1&1&1\\
0&0&1&0&0&1&1&1&0&1&0&0&0&1&0&0&1&1&0&1\\
0&0&0&0&1&0&1&0&0&0&0&1&0&0&0&0&0&0&1&0\\
1&0&0&0&0&1&0&0&0&1&1&0&1&1&0&0&1&1&0&1\\
1&0&1&1&1&0&1&0&1&0&1&0&0&0&1&1&0&0&1&1\\
1&1&0&0&1&1&0&1&0&1&0&1&0&0&0&1&1&1&0&1\\
0&0&0&1&1&0&0&0&0&0&1&1&0&0&0&1&0&1&1&1\\
0&0&1&0&1&1&0&1&1&0&1&0&1&0&1&0&1&0&0&1
\end{bmatrix},
\begin{bmatrix}
1&1&1&0&1&1&1&0&1&1&0&1&0&1&1&0&0&0&1&0\\
1&0&0&0&1&1&0&1&0&0&1&1&1&1&1&1&1&1&0&0\\
1&1&0&0&0&1&0&0&1&1&0&1&0&0&0&1&1&1&1&1\\
0&1&1&1&0&1&1&1&0&0&0&0&0&1&1&1&0&0&1&1\\
1&0&1&1&1&0&0&0&0&1&1&1&1&0&1&0&1&0&0&0\\
0&0&0&1&0&1&1&0&1&0&0&0&1&0&1&0&0&0&1&0\\
0&0&1&1&1&0&1&0&0&1&0&0&1&1&0&0&1&0&1&0\\
1&1&0&1&1&0&1&0&0&0&0&1&1&1&1&0&0&1&1&1\\
1&0&0&1&0&0&0&1&0&1&0&1&0&0&0&1&0&1&1&1\\
1&1&1&1&1&1&0&0&1&0&0&0&1&0&1&0&0&1&0&1\\
0&1&1&0&0&1&0&1&0&0&0&0&1&0&0&0&0&1&1&1\\
1&0&1&0&0&0&1&0&0&1&0&1&1&1&0&0&1&1&1&1\\
1&0&1&0&1&1&0&0&0&1&0&1&0&0&1&0&0&1&1&1\\
0&1&0&0&1&0&1&1&0&0&0&1&0&0&0&1&1&0&0&1\\
1&0&0&0&0&1&0&1&0&0&1&0&0&0&1&1&0&1&1&1\\
0&1&0&0&1&0&0&0&1&0&1&1&1&1&0&0&1&0&0&1\\
1&0&0&1&0&1&0&0&0&1&1&1&1&0&1&0&0&1&0&1\\
0&1&1&1&1&1&0&1&0&0&0&0&1&0&1&0&0&1&0&0\\
1&1&1&1&1&1&1&0&0&0&0&0&0&0&1&1&0&1&0&0\\
0&1&1&1&0&1&0&1&1&1&1&1&1&1&0&0&0&0&1&0
\end{bmatrix}$$

respectively. By setting $Threshold = Trials = 8$ (See Figure 2) we were able to collect the following 19 (δ_x, δ_y) pairs:

$$\begin{array}{llll}
(624503, 241984) & (776771, 695001) & (327753, 131087) & (55169, 514545) \\
(202272, 445310) & (602355, 656872) & (917362, 320210) & (58440, 623796) \\
(974042, 35345) & (715678, 214754) & (383370, 531929) & (204095, 609811) \\
(653178, 824812) & 108979, 97871) & (174443, 861123) & (469759, 1002664) \\
(741723, 572238) & (671505, 841867) & 928012, 475934) &
\end{array}$$

by performing 1736 encryption operations. We chose δ_x's to be linearly independent. Using the above pairs, we formed the following linear relation for any points on the same segment:

$$\delta_x = B\delta_y,$$

where B is given by

$$\begin{bmatrix}
1\oplus t_1 & 1 & t_1 & 1\oplus t_1 & 0 & 1\oplus t_1 & t & 1 & 1\oplus t_1 & t_1 & 1 & t_1 & 1 & 0 & 0 & 1\oplus t_1 & 0 & t & 1\oplus t_1 & t_1 \\
1\oplus t_2 & 0 & 1\oplus t_2 & 1\oplus t_2 & 0 & 1\oplus t_2 & t_2 & 1 & 1\oplus t_2 & 1\oplus t_2 & 0 & t_2 & 1 & 1 & 1 & 1\oplus t_2 & 0 & 1\oplus t_2 & t_2 & t_2 \\
1\oplus t_3 & 1 & t_3 & t_3 & 1 & 1\oplus t_3 & t_3 & 1 & 1\oplus t_3 & t_3 & 1 & t_3 & 1 & 1 & 1 & t_3 & 0 & t_3 & 1\oplus t_3 & t_3 \\
t_4 & 1 & t & t & 0 & t_4 & t_4 & 0 & t_4 & 1\oplus t_4 & 0 & t_4 & 1 & 0 & 0 & t_4 & 1 & t_4 & t_4 & t_4 \\
t_5 & 1 & 1\oplus t_5 & t & 1 & 1\oplus t_5 & t_5 & 1 & 1\oplus t_5 & t & 0 & 1\oplus t_5 & 0 & 0 & 1 & 1\oplus t_5 & 1 & t_5 & t_5 & t_5 \\
1\oplus t_6 & 0 & 1\oplus t_6 & 1\oplus t_6 & 0 & t_6 & t_6 & 1 & 1\oplus t_6 & 1\oplus t_6 & 1 & t_6 & 1 & 1 & 1 & t_6 & 1 & 1\oplus t_6 & 1\oplus t_6 & t_6 \\
t_7 & 1 & 1\oplus t_7 & 1\oplus t_7 & 0 & 1\oplus t_7 & 1\oplus t_7 & 0 & 1\oplus t_7 & t_7 & 1 & t_7 & 0 & 1 & 1 & 1\oplus t_7 & 1 & t_7 & t_7 & t_7 \\
1\oplus t_8 & 0 & 1\oplus t_8 & t_8 & 0 & t_8 & t_8 & 1 & 1\oplus t_8 & t_8 & 1 & 1\oplus t_8 & 0 & 0 & 1 & 1\oplus t_8 & 0 & t_8 & 1\oplus t_8 & t_8 \\
t_9 & 1 & t & 1\oplus t_9 & 0 & 1\oplus t_9 & 1\oplus t_9 & 1 & t_9 & 1\oplus t_9 & 0 & 1\oplus t_9 & 1 & 1 & 0 & t_9 & 0 & t_9 & t_9 & t_9 \\
1\oplus t_{10} & 0 & 1\oplus t_{10} & 1\oplus t_{10} & 1 & 1\oplus t_{10} & 1\oplus t_{10} & 1 & 1\oplus t_{10} & 1\oplus t_{10} & 0 & 1\oplus t_{10} & 1 & 1 & 0 & t_{10} & 0 & 1\oplus t_{10} & t_{10} & t_{10} \\
1\oplus t_{11} & 0 & 1\oplus t & 1\oplus t_{11} & 1 & t_{11} & 1\oplus t_{11} & 0 & 1\oplus t_{11} & t_{11} & 0 & 1\oplus t_{11} & 0 & 1 & 1 & t & 1 & 1\oplus t_{11} & 1\oplus t_{11} & t_{11} \\
t_{12} & 0 & 1\oplus t_{12} & t_{12} & 1 & 1\oplus t_{12} & t_{12} & 1 & 1\oplus t_{12} & 1\oplus t_{12} & 0 & t_{12} & 0 & 0 & 0 & 1\oplus t_{12} & 1 & 1\oplus t_{12} & 1\oplus t_{12} & t_{12} \\
1\oplus t_{13} & 1 & 1\oplus t_{13} & 1\oplus t_{13} & 1 & 1\oplus t_{13} & t_{13} & 1 & 1\oplus t_{13} & t_{13} & 1 & 1\oplus t_{13} & 0 & 0 & 1 & t_{13} & 1 & t_{13} & 1\oplus t_{13} & t_{13} \\
t_{14} & 1 & 1\oplus t_{14} & t & 0 & 1\oplus t_{14} & 1\oplus t_{14} & 0 & t_{14} & 1\oplus t_{14} & 1 & 1\oplus t_{14} & 1 & 0 & 0 & t_{14} & 0 & t_{14} & t_{14} & t_{14} \\
t_{15} & 0 & t_{15} & t_{15} & 0 & t_{15} & 1\oplus t_{15} & 1 & t_{15} & t_{15} & 0 & 1\oplus t_{15} & 0 & 1 & 1 & t_{15} & 0 & t_{15} & 1\oplus t_{15} & t_{15} \\
t_{16} & 1 & 1\oplus t_{16} & {}_{16}t & 0 & t_{16} & 1\oplus t_{16} & 1 & 1\oplus t_{16} & t_{16} & 1 & 1\oplus t_{16} & 0 & 1 & 1 & t_{16} & 0 & t_{16} & t_{16} & t_{16} \\
1\oplus t_{17} & 1 & 1\oplus t_{17} & t_{17} & 1 & t_{17} & t_{17} & 0 & 1\oplus t_{17} & t & 0 & 1\oplus t_{17} & 1 & 1 & 0 & 1\oplus t_{17} & 1 & t & 1\oplus t_{17} & t_{17} \\
t_{18} & 0 & t_{18} & t_{18} & 1 & 1\oplus t_{18} & 1\oplus t_{18} & 0 & 1\oplus t_{18} & 1\oplus t_{18} & 1 & 1\oplus t_{18} & 1 & 0 & 0 & t_{18} & 1 & t_{18} & 1\oplus t_{18} & t_{18} \\
t_{19} & 1 & t_{19} & 1\oplus t_{19} & 0 & 1\oplus t_{19} & t_{19} & 0 & t_{19} & 1\oplus t_{19} & 1 & t & 1 & 0 & 1 & t_{19} & 0 & 1\oplus t_{19} & t_{19} & t_{19} \\
t_{20} & 0 & t_{20} & 1\oplus t_{20} & 0 & 1\oplus t_{20} & t_{20} & 0 & t_{20} & 1\oplus t_{20} & 0 & 1\oplus t_{20} & 0 & 1 & 0 & 1\oplus t_{20} & 0 & 1\oplus t_{20} & t_{20} & t_{20}
\end{bmatrix}$$

for any $t_i \in \{0,1\}$, $i = 1, 2, \cdots 20$. Using this relation and the pool of already encrypted 1736 plaintext-ciphertext pairs, we were able to decode correctly 99.946% of a random $100,000$ ciphertext.