New Chosen-Plaintext Attacks on the One-Wayness of the Modified McEliece PKC Proposed at Asiacrypt 2000

Kazukuni Kobara and Hideki Imai

Institute of Industrial Science, The University of Tokyo
4-6-1, Komaba, Meguro-ku, Tokyo, 153-8505 Japan
Tel: +81-3-5452-6232
FAX: +81-3-5452-6631
{kobara,imai}@iis.u-tokyo.ac.jp

Abstract. McEliece PKC (Public-Key Cryptosystem), whose security is based on the decoding problem, is one of a few alternatives for the current PKCs that are mostly based on either IFP (Integer Factoring Problem) or DLP (Discrete Logarithm Problem), which would be solved in polynomial-time after the emergence of quantum computers. It is known that the McEliece PKC with an appropriate conversion satisfies (in the random oracle model) the strongest security notion IND-CCA2 (IN-Distinguishability of encryption against adaptively Chosen-Ciphertext Attacks) under the assumption that breaking OW-CPA (One-Wayness against Chosen-Plaintext Attacks) of the underlying McEliece PKC, i.e. the McEliece PKC with no conversion, is infeasible. Breaking OW-CPA of it is still infeasible if an appropriate parameter, $n \geq 2048$ with optimum t and k, is chosen since the binary work factor to break it with the best CPA is around 2^{106} for $(n, k, t) = (2048, 1278, 70)$. The aim of the modification at Asiacrypt 2000 is to improve it of the next smaller parameter $n = 1024$ to a safe level 2^{88} from an almost dangerous level 2^{62}. If his idea works correctly, we can use the more compact system safely. In this paper, we carefully review the modification at Asiacrypt 2000, and then show that the one-wayness of it is vulnerable against our new CPAs.

1 Introduction

Since the concept of public-key cryptosystem (PKC) was introduced by Diffie and Hellman [5], many researchers have proposed numerous PKCs based on various problems, such as integer factoring, discrete logarithm, decoding a large linear code, knapsack, inverting polynomial equations, lattice and so on. While some of them are still alive, most of them were broken by cryptographers due to their intensive cryptanalysis. As a result, almost all of the current secure systems on the market employ only a small class of PKCs, such as RSA and elliptic curve cryptosystems, which are all based on either integer factoring problem (IFP) or discrete logarithm problem (DLP). This situation would cause a serious problem

D. Naccache and P. Paillier (Eds.): PKC 2002, LNCS 2274, pp. 237–251, 2002.

after someone discovers one practical algorithm which breaks both IFP and DLP in polynomial-time. Who can prove that such an algorithm will never be found? Actually, Shor has already found a (probabilistic) polynomial-time algorithm in [17], even though it requires a quantum computer that is impractical so far. In order to prepare for such unfortunate situations, we need to find another secure scheme relying on neither IFP nor DLP.

The McEliece PKC, proposed by R.J. McEliece in [15], is one of a few alternatives for the PKCs based on IFP or DLP. It is based on the decoding problem of a large linear code with no visible structure which is conjectured to be an NP-complete problem.[1] While several attacks [1,3,4,7,11,14,19] are known on the McEliece PKC, all of them can be prevented by either enlarging the parameter size or applying an appropriate conversion to it [10].

The McEliece PKC with an appropriate conversion in [10] satisfies (in the random oracle model [2]) the strongest security notion IND-CCA2 (INDistinguishability of encryption [6] against adaptively Chosen-Ciphertext Attacks) under the assumption that it is infeasible to break OW-CPA (One-Wayness against Chosen-Plaintext Attacks) of the underlying McEliece PKC. OW-CPA is said to be broken if one can recover the whole plaintext of an arbitrarily given ciphertext using neither partial knowledge on the plaintext nor decryption oracles. It is still infeasible to break OW-CPA of the McEliece PKC if an appropriate parameter, $n \geq 2048$ with optimum t and k, is chosen since the binary work factor to break it with the best CPA [4] is around 2^{106} for $(n, k, t) = (2048, 1278, 70)$.

At Asiacrypt 2000, a modification of the McEliece PKC was proposed by P. Loidreau. While his modification does not improve the immunity against attacks using decryption oracles, such as the malleability attack [8,19] and the reaction attack [7], or attacks using partial knowledge on the target plaintext, such as the related-message attack [3], the message-resend attack [3] and the known-partial-plaintext attack [9], it does not matter since all of them can be prevented with a conversion. The aim of his modification is to improve the binary work factor for breaking OW-CPA of the next smaller parameter $n = 1024$ to a safe level 2^{88} from an almost dangerous level 2^{62}. If his idea works correctly, we can use the more compact system safely.

In this paper, we carefully review the modification at Asiacrypt 2000 to see whether it truely improves OW-CPA or not. Then we show that it is vulnerable against our "new" CPAs on OW (even though the modification certainly enhances OW against "ever known" CPAs). Our attacks exploit only the modified structure of it and thus cannot be applied to the original (unmodified) McEliece. This means the OW-CPA of the the original (unmodified) McEliece PKC is still infeasible as long as an secure parameter is chosen.

This paper is organized as follows: in Section 2 and 3, we describe both the McEliece PKC and the ever known CPAs on OW of it, respectively. Then, in Section 4, we review the modified cryptosystem proposed by Loidreau [13].

[1] The complete decoding problem of an arbitrary linear code is proven to be NP-complete in [20].

Finally, in Section 5, we show our new CPAs which weaken the one-wayness of the modified cryptosystem.

2 McEliece Public-Key Cryptosystem

2.1 Cryptosystems

The McEliece PKC consists of the following key generation, encryption and decryption systems:

Key generation: One generates the following three matrices G,S,P:

G**:** $k \times n$ generator matrix of a binary Goppa code that can correct up to t errors, and for which an efficient decoding algorithm $\Psi()$ is known. The parameter t is given by $\lfloor \frac{d_{min}-1}{2} \rfloor$ where d_{min} denotes the minimum Hamming distance of the code.

S**:** $k \times k$ random binary non-singular matrix

P**:** $n \times n$ random permutation matrix.

Then, computes the $k \times n$ matrix $G' = SGP$.

Secret key: (S, P) and $\Psi()$

Public key: (G', t)

Encryption: The ciphertext c of a given message msg is calculated as follows:

$$c = msg \cdot G' \oplus z \tag{1}$$

where msg a binary vector of length k, and z denotes a random binary error vector of length n having t 1's.

Decryption: First, one calculates cP^{-1} where

$$c \cdot P^{-1} = (msg \cdot S)G \oplus z \cdot P^{-1} \tag{2}$$

and P^{-1} denotes the inverse of P. Second, applies the decoding algorithm $\Psi()$ to cP^{-1}. Since the Hamming weight of $z \cdot P^{-1}$ is t, $\Psi()$ can correct it:

$$msg \cdot S = \Psi(c \cdot P^{-1}). \tag{3}$$

Now, the plaintext msg of c can be obtained by

$$msg = (msg \cdot S)S^{-1}. \tag{4}$$

2.2 Underlying Codes

As the underlying codes of the McEliece PKC, we recommend to employ Goppa codes but other codes, such as Reed-Solomon codes, BCH codes and so on. The difference between them is whether the weight distribution is determined by the public parameters n and k. While the Goppa codes have a variety of the weight distributions according to the underlying Goppa polynomials even if both n and k are fixed, both the Reed-Solomon codes and BCH codes have only the fixed weight distribution depending on both n and k. In other words, when both n and k are given from a public matrix G', one can know the underlying code G if codes like Reed-Solomon or BCH are used. Then once G is found, an adversary can reveal the permutation P between G and G' using the SSA(Support Splitting Algorithm) [16] that can reveal the permutation P between the codes having the same weight distribution. The other secret matrix S can be revealed using G and $G'P^{-1}$ with a simple linear algebra.

The next case we have to avoid is that the candidates for G is small enough to enumerate. In this case, the following attack is possible [14]. An adversary picks up a candidate for G and then sees whether it has the same weight distribution as G'. If it has, it is the correct G. The following processes to obtain both P and S are the same as the previous attack.

The former attack can be avoided using a Goppa code, and then the latter attack can be avoided using a Goppa code where the cardinality of the Goppa polynomials is too large to enumerate.

3 Known Chosen Plaintext Attacks on One-Wayness of McEliece PKC

Since the aim of the modification at Asiacrypt 2000 is to enhance the immunity against CPAs on OW, we focus on them. Note that the attack on the public key [14] can be avoided if one choose the underlying Goppa polynomial out of a large set enough to avoid the exhaustive search, and also the other attacks either abusing decryption oracles, such as the malleability attack [8,19] and the reaction attack [7], or abusing partial knowledge on the target plaintext, such as the related-message attack [3], the message-resend attack [3] and the known-partial-plaintext attack [9], can be avoided by applying an appropriate conversion in [10].

Only the following two attacks are known as the CPAs on OW of the McEliece PKC. They can be summarized as follows.

3.1 Generalized Information-Set-Decoding Attack

Let G'_k denote k independent columns picked out of G', and then let c_k and z_k denote the corresponding k coordinates of c and z, respectively. They have the following relationship

$$c_k = msg \cdot G'_k \oplus z_k. \tag{5}$$

If $z_k = 0$ and G'_k is non-singular, msg can be recovered [1] by

$$msg = (c_k \oplus z_k)G'^{-1}_k. \tag{6}$$

Even if $z_k \neq 0$, msg can be obtained by guessing z_k among small Hamming weights [11], i.e. $Hw(z_k) \leq j$ for small j. The correctness of the recovered plaintext msg is verifiable by checking whether the Hamming weight of

$$c \oplus msg \cdot G' = c \oplus c_k G'^{-1}_k \cdot G' \oplus z_k G'^{-1}_k \cdot G' \tag{7}$$

is t or not.

The corresponding algorithm is summarized as follows:

Algorithm 1 (GISD)

Input: a ciphertext c, a public key (G', t) and an attack parameter $j \in Z$.
Output: a plaintext msg.

```
1. Choose $k$ independent columns out of $G'$, and then calculate $\hat{G'}_k := G'^{-1}_k G'$.
   Let $I$ denote the set of the indexes of the $k$ chosen columns, and then $J$
   denote the set of the remaining columns.
2. Do the following until $msg$ is found:
   2.1 Calculate $\hat{z} := c \oplus c_k \hat{G'}_k$. If $Hw(\hat{z}) = t$, output $msg := c_k G'^{-1}_k$.
   2.2 For $i_1$ from 1 to $j$ do the following:
       i. For $i_2$ from 1 to $\binom{n}{i_1}$ do the following:
          A. Choose a new $z'_k$, such that $Hw(z'_k) = i_1$.
          B. If $Hw(\hat{z} \oplus z'_k \hat{G'}_k) = t$, output $msg := (c_k \oplus z'_k) G'^{-1}_k$.
   2.3 Replace one coordinate in $I$ with a coordinate in $J$, and then renew the
       $\hat{G'}_k := G'^{-1}_k G'$ using Gaussian elimination.
```

We estimate the binary work factor of the above GISD attack as follows. In Step 1, $G'^{-1}_k G'$ is the $k \times n$ matrix where the chosen k columns make the identity matrix. It can be obtained by the Gaussian elimination with the work factor of

$$\sum_{i=1}^{k} \frac{(k-1)(n-i+1)}{4} = \frac{k(k-1)(2n+1-k)}{8} \tag{8}$$

bit operations. When one checks the Hamming weight in Step 2.1 and Step B, he/she does not need to calculate the whole n coordinates of $c \oplus c_k \hat{G'}_k$ in Step 2.1 and $\hat{z} \oplus z'_k \hat{G'}_k$ in Step B, respectively, since he/she can know whether their weight exceeds t or not with around $2t$ coordinates in J provided that wrong cases have the average weight of $n/2$. Thus the binary work factor for calculating the $2t$ coordinates of $c \oplus c_k \hat{G'}_k$ in Step 2.1 is $t \cdot k/2$, and that of $\hat{z} \oplus z'_k \hat{G'}_k$ in Step B is $t \cdot i_1$. Accordingly, the work factor for Step 2.2 is

$$V_j = \sum_{i_1=1}^{j} t \cdot i_1 \cdot \binom{k}{i_1}. \tag{9}$$

In Step 2.3, one needs to update $\hat{G'}_k = G'^{-1}_k G'$ whose binary work factor is

$$\frac{(k-1)(n-k)}{4}. \tag{10}$$

Since Step 2 is repeated around T_j times where:

$$T_j = \frac{\binom{n}{k}}{\sum_{i=0}^{j} \binom{t}{i} \binom{n-t}{k-i}}, \tag{11}$$

the total work factor is given by

$$W_j \approx \left\{ \frac{(k-1)(n-k)}{4} + \frac{t \cdot k}{2} + V_j \right\} \cdot T_j \tag{12}$$

When n is given, designers of the cryptosystem can optimize both k and t to make (12) higher, and then attackers can optimize the attack parameter j to make it lower. For $n = 2^{10}$, $\min_j(\max_{k,t}(W_j)) \approx 2^{67}$, which can be achieved when $j = 1$, $t = 38$ to 40 and $k = n - m \cdot t = 644$ to 624, respectively. For $n = 2^{11}$, $\min_j(\max_{k,t}(W_j)) \approx 2^{113}$, which can be achieved when $j = 1$, $t = 63$ to 78 and $k = n - m \cdot t = 1355$ to 1190, respectively.

3.2 Finding-Low-Weight-Codeword Attack

This attack uses an algorithm which accepts both an arbitrary generator matrix and a positive integer w, and then finds out a codeword of weight w [18,4]. Since the codeword of weight t of the following $(k + 1) \times n$ generator matrix

$$\begin{bmatrix} G' \\ c \end{bmatrix} \tag{13}$$

is the error vector z where $c = msg \cdot G' \oplus z$, this algorithm can be used to recover msg from given c and G'.

This algorithm is summarized as follows:

Algorithm 2 (FLWC)

Input: a ciphertext c, a public key (G', t) and attack parameters $(p, \rho) \in Z \times Z$.
Output: a plaintext msg.

```
1. Choose $k+1$ independent columns from (13) and then apply Gaussian elimination to obtain a $(k+1) \times n$ matrix where chosen $k+1$ columns make the identity matrix. Let $I$ denote a set of the indexes of the $k+1$ chosen coordinates, and $J$ denote those of the remaining coordinates.
2. Do the following until a code word $z$ of weight $t$ is found:
   2.1 Split $I$ in two subsets $I_1$ and $I_2$ at random where $|I_1| = \lfloor (k+1)/2 \rfloor$ and $|I_2| = \lceil (k+1)/2 \rceil$. The rows of the $(k+1) \times (n-k-1)$ matrix $M$ corresponding to $J$ are also split in two parts, a $(\lfloor (k+1)/2 \rfloor) \times (n-k-1)$ matrix $M_1$ and a $(\lceil (k+1)/2 \rceil) \times (n-k-1)$ matrix $M_2$ according to $I_1$ and $I_2$, respectively, i.e. if $I_1$ includes $i$-th coordinate, the $i$-th row of $M$ is included in $M_1$.
   2.2 Select a $\rho$-element subset $J_\rho$ of $J$ at random.
   2.3 For $i$ from 1 to $\binom{|I_1|}{p}$ do the following:
        i. Select a new set of $p$ rows of the matrix $M_1$. Let $\mathcal{P}_{1,i}$ denote the set.
       ii. Sum up the chosen $p$ rows of $M_1$ in $Z_2$. Let $\Lambda_{1,i|J_\rho}$ denote the chosen $\rho$ coordinates of the result.
      iii. Store both $\mathcal{P}_{1,i}$ and $\Lambda_{1,i|J_\rho}$ in a hash table with $2^\rho$ entries using $\Lambda_{1,i|J_\rho}$ as an index.
   2.4 For $j$ from 1 to $\binom{|I_2|}{p}$ do the following:
         i. Select a new set of $p$ rows of the matrix $M_2$. Let $\mathcal{P}_{2,j}$ denote the set.
        ii. Sum up the chosen $p$ rows of $M_2$ in $F_2$. Let $\Lambda_{2,j|J_\rho}$ denote the chosen
            $\rho$ coordinates of the result.
       iii. Store both $\mathcal{P}_{2,j}$ and $\Lambda_{2,j|J_\rho}$ in a hash table with $2^\rho$ entries using
            $\Lambda_{2,j|J_\rho}$ as an index.
   2.5 Using the hash table, find all pairs of sets $(\mathcal{P}_{1,i}, \mathcal{P}_{2,j})$ such that $\Lambda_{1,i|J_\rho} =$
       $\Lambda_{2,j|J_\rho}$ and check whether $Hw(\Lambda_{1,i|J} \oplus \Lambda_{2,j|J}) = t - 2p$ where $\Lambda_{1,i|J}$ and
       $\Lambda_{2,j|J}$ denote the sums of the $p$ rows of $M_1$ and $M_2$ corresponding to
       $\mathcal{P}_{1,i}$ and $\mathcal{P}_{2,j}$, respectively. If found, output the code word.
   2.6 Replace one coordinate in $I$ with a coordinate in $J$, and then make the
       chosen $k+1$ columns be the identity matrix using Gaussian elimination.
3. Apply the information-set decoding to $c \oplus z$, and then recover the corre-
   sponding message $msg$.
```

Under the assumption that each iteration is independent, one needs to repeat Step 2 around $T_{p,\rho}$ times where

$$T_{p,\rho} = \frac{\binom{k-2p}{\frac{k}{2}-p}\binom{2p}{p}\binom{-k+n+2p-t}{\rho}}{\binom{k}{\frac{k}{2}}\binom{-k+n}{\rho}} \cdot \frac{\binom{n-t}{k-2p}\binom{t}{2p}}{\binom{n}{k}}. \tag{14}$$

In Step 2.1 to 2.4, one needs to compute both $\Lambda_{1,i|J_\rho}$ and $\Lambda_{2,j|J_\rho}$ for about $\binom{(k+1)/2}{p}$ combinations, respectively, whose binary work factor is around

$$\Omega_1(p,\rho) = p \cdot \rho \cdot \binom{(k+1)/2}{p}. \tag{15}$$

In Step 2.5, around $\binom{(k+1)/2}{p}^2/2^\rho$ pairs of $(\mathcal{P}_{1,i}, \mathcal{P}_{2,j})$ satisfies $\Lambda_{1,i|J_\rho} \oplus \Lambda_{2,j|J_\rho} = 0$, and for each pair one needs to check the weight of $\Lambda_{1,i|J} \oplus \Lambda_{2,j|J}$. In the same way as Algorithm 1, one can know that $Hw(\Lambda_{1,i|J} \oplus \Lambda_{2,j|J}) \neq t - 2p$ by calculating the weight of around $2(t-2p)$ coordinates in J. Thus the binary work factor for Step 2.5 is around

$$\Omega_2(p,\rho) = 2(t-2p) \cdot p \cdot \frac{\binom{(k+1)/2}{p}^2}{2^\rho}. \tag{16}$$

The binary work factor for updating the generator matrix in Step 2.6 is

$$\Omega_3(p,\rho) = \frac{k(n-k-1)}{4}. \tag{17}$$

Thus the total binary work factor is given by

$$W_{p,\rho} \approx (\Omega_1(p,\rho) + \Omega_2(p,\rho) + \Omega_3(p,\rho)) \cdot T_{p,\rho}. \tag{18}$$

For $n = 2^{10}$, $\min_{p,\rho}(\max_{k,t}(W_{p,\rho})) \approx 2^{62}$, which can be achieved when $(p,\rho) = (2,19)$, $t = 36$ to 43 and $k = n - m \cdot t = 664$ to 594, respectively. For $n = 2^{11}$, $\min_{p,\rho}(\max_{k,t}(W_{p,\rho})) \approx 2^{106}$, which can be achieved when $(p,\rho) = (2,22)$, $t = 63$ to 79 and $k = n - m \cdot t = 1355$ to 1179, respectively.

4 Loidreau's Modification at Asiacrypt 2000

The aim of the modification at Asiacrypt 2000 [13] is to improve the difficulty of breaking OW-CPA without increasing n. It uses some linear transformation $f()$ such that $f(\mathcal{C}) = \mathcal{C}$ (C being the Goppa code of the PKC). Instead of choosing an error vector of small weight, it uses an error vector z' such that $f(z')$ has small weight. This way, the error vector itself can have higher Hamming weight, and it is harder to find it via the usual search methods.

In this section, we review the underlying principles and the modified cryptosystem more precisely.

4.1 Frobenius Automorphism Group of Goppa Codes

Let us consider the Goppa code $\varGamma(L, g)$ over F_{2^m} where $L = (\alpha_1, \cdots, \alpha_n)$ contains all the elements in F_{2^m}.

If all the coefficients of the Goppa polynomial g is in a subfield F_{2^s} of F_{2^m}, then the code $\varGamma(L, g)$ is invariant under the action of the Frobenius automorphism. That is, a Frobenius mapped word $\sigma(c)$ of a code word c of $\varGamma(L, g)$ is also a code word of $\varGamma(L, g)$:

$$\forall c = (c_{\alpha_1}, \cdots, c_{\alpha_n}) \in \varGamma(L, g), \qquad \sigma(c) = (c_{\sigma(\alpha_1)}, \cdots, c_{\sigma(\alpha_n)}) \in \varGamma(L, g) \quad (19)$$

where $\sigma : x \mapsto x^{2^s}$.

4.2 Orbits Generated by Frobenius Automorphism

The action of the Frobenius automorphism makes some orbits in the field. For simplicity, we consider the field extension $F_{2^{5s}}$ of F_{2^s}, and the corresponding Frobenius automorphism $\sigma : x \mapsto x^{2^s}$. The action of the Frobenius automorphism to $F_{2^{5s}}$ makes $N_5 = (2^{5s} - 2^s)/5$ orbits of size 5 and 2^s orbits of size 1. In other words, a word $\{z_{\alpha_1}, z_{\alpha_2}, \cdots, z_{\alpha_n}\}$ can be rewritten in the following form after reordering its labeling L:

$$z = \{Z_1, Z_2, \cdots, Z_{N_5}, Z_0\} \quad (20)$$

where $Z_i = \{z_{\alpha_j}, z_{\sigma(\alpha_j)}, z_{\sigma^2(\alpha_j)}, z_{\sigma^3(\alpha_j)}, z_{\sigma^4(\alpha_j)}\}$ for $i \in \{1, \cdots, N_5\}$ denotes an orbit of length 5 generated by α_j, and then Z_0 denotes a sub-vector of length 2^s corresponding to the 2^s orbits of length 1 generated by 2^s distinct elements in F_{2^s}.

For the reordered coordinate, the action of the Frobenius automorphism σ on a word z is given as follows:

$$\sigma(z) = \{\sigma(Z_1), \cdots, \sigma(Z_{N_5}), Z_0\} \quad (21)$$

where $\sigma(Z_i)$ is a left cyclic shift in Z_i, e.g. for $Z_{i_1} = \{1, 1, 1, 0, 0\}$ and $Z_{i_2} = \{1, 1, 0, 1, 0\}$, $\sigma^l(Z_{i_1})$ and $\sigma^l(Z_{i_2})$ for $l \in Z_5$ are listed as follows:

$$\begin{array}{rlrl} Z_{i_1} &= \{1,1,1,0,0\}, & Z_{i_2} &= \{1,1,0,1,0\}, \\ \sigma(Z_{i_1}) &= \{1,1,0,0,1\}, & \sigma(Z_{i_2}) &= \{1,0,1,0,1\}, \\ \sigma^2(Z_{i_1}) &= \{1,0,0,1,1\}, & \sigma^2(Z_{i_2}) &= \{0,1,0,1,1\}, \\ \sigma^3(Z_{i_1}) &= \{0,0,1,1,1\}, & \sigma^3(Z_{i_2}) &= \{1,0,1,1,0\}, \\ \sigma^4(Z_{i_1}) &= \{0,1,1,1,0\}, & \sigma^4(Z_{i_2}) &= \{0,1,1,0,1\}. \end{array} \qquad (22)$$

4.3 *t*-Tower Decodable Vector

In the Loidreau's modified cryptosystem, t-tower decodable vectors are used instead of random error vectors of weight t.

The definition of a t-tower decodable vector is given as follows:

Definition 1 (t-Tower Decodable Vector) *t-tower decodable vector z' is a word of length n satisfying the following three conditions:*

Larger-weight: $Hw(z') > t$.
Reducibility: *There exists a linear combination $f()$ such that $Hw(z) \leq t$ where*

$$z = f(z') = \sum_{i=0}^{m/s-1} b_i \cdot \sigma^i(z'), \qquad b_i \in F_2. \qquad (23)$$

Recoverability: *z' is uniquely recoverable from the above z.*

In [13], t-tower decodable vector z' is generated as follows:

Algorithm 3 (Generation of a t-Tower Decodable Vector)
Output: a t-tower decodable vector z'.

1. Set all the coordinates of z' to 0.
2. Choose randomly $p = \lfloor t/2 \rfloor$ orbits out of the N_5 orbits of length 5.
3. Flip 3 bits each at random in the chosen p orbits.

The following $f_1()$ or $f_2()$ where

$$z = f_1(z') = z' + \sigma(z') + \sigma^2(z'), \qquad (24)$$
$$z = f_2(z') = z' + \sigma^2(z') + \sigma^3(z') \qquad (25)$$

reduces the weight of z' within t since $\sigma^l(Z_{i_1})$ or $\sigma^l(Z_{i_2})$ in (22) cover all the patterns of a vector of length 5 and weight 3, and then they are transformed into the following patterns:

$$\begin{aligned} f_1(\sigma^l(Z_{i_1})) &= \sigma^l(Z_{i_1} + \sigma(Z_{i_1}) + \sigma^2(Z_{i_1})) = \sigma^l(\{1,0,1,1,0\}), \\ f_1(\sigma^l(Z_{i_2})) &= \sigma^l(Z_{i_2} + \sigma(Z_{i_2}) + \sigma^2(Z_{i_2})) = \sigma^l(\{0,0,1,0,0\}), \\ f_2(\sigma^l(Z_{i_1})) &= \sigma^l(Z_{i_1} + \sigma^2(Z_{i_1}) + \sigma^3(Z_{i_1})) = \sigma^l(\{0,1,0,0,0\}), \\ f_2(\sigma^l(Z_{i_2})) &= \sigma^l(Z_{i_2} + \sigma^2(Z_{i_2}) + \sigma^3(Z_{i_2})) = \sigma^l(\{0,0,1,1,1\}) \end{aligned} \qquad (26)$$

where $\sigma^l()$ denotes a l-bit left cyclic shift. More formally, let p_1 and p_2 denote the number of $\sigma^l(Z_{i_1})$ for any l and $\sigma^l(Z_{i_2})$ for any l in z', respectively. Since $p_1 + p_2 = p = \lfloor t/2 \rfloor$ and

$$\begin{aligned} \min(f_1(z'), f_2(z')) &= \min(3p_1 + p_2, p_1 + 3p_2) \\ &= \min(2p_1 + \lfloor t/2 \rfloor, \lfloor t/2 \rfloor + 2p_2) \\ &\leq 2 \cdot \lfloor t/2 \rfloor \\ &\leq t, \end{aligned} \tag{27}$$

one can reduce the weight of z' within t using either $f_1()$ or $f_2()$.

Using the corrected vector z, one can uniquely recover the corresponding t-tower decodable vector z' since both $f_1()$ and $f_2()$ are one-to-one mappings.

4.4 Loidreau's Modified Cryptosystem

As we have seen in the previous subsections, Loidreau's modified cryptosystem [13] uses the field extension $F_{2^{5s}}$ of F_{2^s}, i.e. it employs a Goppa polynomial g (of degree t) over a subfield F_{2^s} of $F_{2^{5s}}$ to enable the Frobenius automorphism. It also employs a hiding polynomial g_1 over $F_{2^{5s}}$ of degree t_1 (which has no roots in L) to enlarge the cardinality of the underlying polynomial gg_1.

The cardinality of gg_1 is approximately given by $\{(2^{5s})^{t_1}/t_1\} \cdot \{(2^s)^t/t\}$ since the number of irreducible monic polynomials of degree x over F_{2^y} is around $(2^y)^x/x$ [12].

$\Gamma(L, gg_1)$ can be decoded using the decoding algorithm for $\Gamma(L, g)$ since $\Gamma(L, gg_1)$ is a subcode of $\Gamma(L, g)$. The plaintext size of $\Gamma(L, gg_1)$ is $k - t_1$. Both the key generation process and the encryption process are the same as the (unmodified) McEliece PKC except the following points:

- All the N_5 orbits of length 5 are open to the public as a part of a public key (but the order in each orbit is kept in secret). Note that the orbits can be opened without increasing the public key size by permuting orbits and then by permuting columns in each orbit as P. Since the units of orbits are the same as L, one can know them. While it reduces the cardinality of the permutations P, it still maintains a large amount (see Table 1 and 2, respectively).
- Both $f_1()$ and $f_2()$ are kept in secret (which is the same that the order in each orbit is kept in secret).
- t-tower decodable vectors z' are used as error vectors instead of random vectors of weight t.

The decryption process is described as follows:

Algorithm 4 (Decoding for the modified cryptosystem)

Input: a ciphertext c.

Output: a corresponding plaintext msg.

1. Apply $f_1()$ and $f_2()$ to the given ciphertext c, respectively.

Table 1. Cardinalities of System Parameters

Cryptosystem	Permutations	Goppa polynomials	Error vectors	Orders in orbits
McEliece PKC [15]	$n!$	$\frac{(2^m)^t}{t}$	$\binom{n}{t}$	–
Loidreau's modified version [13]	$(5!)^{N_5} \cdot N_5!$	$\frac{(2^m)^{t_1}}{t_1} \cdot \frac{(2^s)^t}{t}$	$10^{\lfloor t/2 \rfloor} \cdot \binom{N_5}{\lfloor t/2 \rfloor}$	$(4!)^{N_5}$

2. Decode $f_1(c)$ and $f_2(c)$ using the decoding algorithm for $\Gamma(L, g)$. At least one of them can be corrected since the Hamming weight of the error vector of at least one of them is smaller than or equal to t.
3. Using the corrected error vector z, reconstruct the corresponding t-tower decodable vector z'.
4. Apply the information-set decoding to $c \oplus z'$, and then recover the corresponding message msg.

If both of $f_1(c)$ and $f_2(c)$ are corrected in Step 2, decrypt two messages and then discard one using the redundancy in the plaintexts. This means the modified cryptosystem requires a redundancy in a plaintext.

4.5 One-Wayness of Modified Cryptosystem against Ever Known Chosen-Plaintext Attacks

Since the Loidreau's modification enlarges the Hamming weight of the error vectors to $3\lfloor t/2 \rfloor$ from t, the binary work factors to break the one-wayness with the ever known CPAs, i.e. both the GISD and the FLWC attacks, are improved (see Table 3).

The Loidreau's modification employs new secrets $f_1()$ and $f_2()$ (or equivalently the order in each orbit). Also it reduces the cardinality of permutations P, Goppa polynomials gg_1 and error vectors z', respectively. If at least one of the cardinalities is small enough to enumerate, the cryptosystem will be broken by guessing it. Fortunately, all of them still preserve a sufficient amount enough to avoid exhaustive search for them (see Table 1 and 2).

5 Our New Chosen-Plaintext Attacks on the Modified Cryptosystem

In this section, we show our new chosen-plaintext attacks on the one-wayness of the Loidreau's modified cryptosystem.

Our attacks use the fact that 1's in a t-tower error vector z' is not uniformly distributed. (This means that our new attacks are not applicable to the (unmodified) McEliece PKC since 1's in its error vector is uniformly distributed.)

Table 2. Cardinalities of System Parameters for $(n, k, t, t_1, s, N_5) = (1024, 624, 40, 9, 2, 204)$

Cryptosystem	Permutations	Goppa polynomials	Error vectors	Orders in orbits
McEliece PKC [15]	2^{8769}	2^{375}	2^{240}	–
Loidreau's modified version [13]	2^{2685}	2^{162}	2^{157}	2^{935}

5.1 Attack I

This attack applies $f_1()$ and $f_2()$ to both the ciphertext and all the rows in the public generator matrix G', respectively. This gives

$$f_1(c) = msg \cdot f_1(G') + f_1(z') \quad \text{and} \tag{28}$$

$$f_2(c) = msg \cdot f_2(G') + f_2(z') \tag{29}$$

respectively [2]. One can view $f_1(G')$ and $f_2(G')$ as generator matrices, and do the usual search for $f_1(z')$ or $f_2(z')$.

Since either $f_1(z')$ or $f_2(z')$ has low weight and both GISD and FLWC are the generic decoding algorithms for an arbitrary linear code, *msg* of (28) and (29) can be decoded without knowing the algebraic structure of $f_1'(G')$ and $f_2'(G')$.

The corresponding algorithm is given as follows:

Algorithm 5 (Attack I)

Input: a ciphertext c, a public key (G', t) and attack parameters $(p, \rho) \in Z \times Z$.
Output: a plaintext *msg*.

1. Apply $f_1'()$ and $f_2'()$ to the given c and all the rows of G' respectively, and then obtain $f_1'(c)$, $f_1'(G')$, $f_2'(c)$ and $f_2'(G')$.
2. Execute the FLWC attack[3] on the pair of $f_1'(G')$ and $f_1'(c)$ and that of $f_2'(G')$ and $f_2'(c)$ respectively to find the code word of weight less than or equal to t. If found, it recovers *msg*.

For $t_1 = 9$, $\min_{p,\rho}(\max_{k',t}(W_j)) \approx 2^{61}$ that can be achieved when $(p, \rho) = (2, 19)$, $t = 38$ to 41 and $k' = n - m \cdot t - t_1 = 635$ to 605 respectively [4].

[2] The Frobenius automorphism gives $f_1(c) = msg' \cdot G' \oplus f_1(z')$ for a certain msg'.

[3] One can use the GISD attack, too.

[4] This binary work factor is smaller than that of the (unmodified) McEliece PKC since the Loidreau's modification employs a subcode of $\Gamma(L, g)$, i.e. $k' = k - t_1$. Then t_1 should be chosen so that the cardinality of Goppa polynomials should be large enough to avoid collisions among users.

Table 3. Binary work factors to break OW-CPA for $(n, k, t, t_1, s_1, N_5) = (1024, 624, 40, 9, 2, 204)$

\ Attacks Systems \	Ever known attacks		Our new attacks	
	GISD[11]	FLWC[4]	Attack I	Attack II
McEliece PKC [15]	2^{67}	2^{62}	–	–
Loidreau's modified cryptosystem [13]	2^{94}	2^{88}	2^{61}	2^{42}

5.2 Attack II

For the error vector z' in the modified cryptosystem, it is proposed (roughly) to split the coordinates into N_5 orbits of five, then choose $p = \lfloor t/2 \rfloor$ such orbits, and in each of them to choose three positions for the non-zero bits. Under this strategy, one can again apply the usual search method, this time on orbits, rather than individual positions.

We found that it is not so difficult to choose $\lceil (k' - 2^s)/5 \rceil$ orbits of all zeros or almost all zeros out of the N_5 orbits. Note that $\lceil (k' - 2^s)/5 \rceil$ corresponds with more than or equal to k coordinates. Once such k coordinates are found, one can decrypt a given ciphertext using the information-set decoding.

The corresponding algorithm is given as follows:

Algorithm 6 (Attack II)

Input: a ciphertext c, a public key (G', t) and an attack parameter $j \in Z$.
Output: a plaintext msg.

```
1. Choose $k'$ independent columns out of $G'$ with which chosen $\lceil (k' - 2^s)/5 \rceil$ orbits of length 5 and $2^s$ orbits of length 1 correspond. Then calculate $\hat{G'}_{k'} := G'^{-1}_{k'} G'$. Let $I$ denote a set of the indexes of the $k'$ chosen columns, and then $J$ denote a set of the remaining columns.
2. Do the following until $msg$ is found:
   2.1 Calculate $\hat{z} := c \oplus c_{k'} \hat{G'}_{k'}$. If $Hw(\hat{z}) = 3\lfloor t/2 \rfloor$, output $msg := c_{k'} G'^{-1}_{k'}$.
   2.2 For $i_1$ from 1 to $j$ do the following:
       i. For $i_2$ from 1 to $\binom{\lceil (k'-2^s)/5 \rceil}{i_1}$ do the following:
          A. Choose a new $z'_{k'}$ that contains $i_1$ non-zero orbits.
          B. If $Hw(\hat{z} \oplus z'_{k'} \hat{G'}_{k'}) = 3\lfloor t/2 \rfloor$, output $msg := (c_{k'} \oplus z'_{k'}) G'^{-1}_{k'}$.
   2.3 Replace one orbit in $I$ with a new orbit in $J$, and then renew the $\hat{G'}_{k'} := G'^{-1}_{k'} G'$ using Gaussian elimination.
```

This algorithm repeats Step 2 around T_j times where:

$$T_j = \frac{\binom{N_5}{\lceil (k'-2^s)/5 \rceil}}{\sum_{i=0}^{j} \binom{\lfloor t/2 \rfloor}{i} \binom{N_5 - \lfloor t/2 \rfloor}{\lceil (k'-2^s)/5 \rceil - i}}. \tag{30}$$

The binary work factors of Step 2.1, 2.2 and 2.3 are $t \cdot k'/2$, V_j and $5(k'-1)(n-k')/4$ respectively where

$$V_j = \sum_{i_1=1}^{j} t \cdot 3i_1 \cdot \binom{5}{3}^{i_1} \cdot \binom{\lceil (k'-2^s)/5 \rceil}{i_1}. \qquad (31)$$

Thus the total work factor is given by

$$W_j \approx \left\{ \frac{5(k'-1)(n-k')}{4} + \frac{t \cdot k'}{2} + V_j \right\} \cdot T_j. \qquad (32)$$

For $t_1 = 9$, $\min_j(\max_{k',t}(W_j)) \approx 2^{42}$ that can be achieved when $j = 1$, $t = 32$ to 50 and $k' = n - m \cdot t - t_1 = 693$ to 517, respectively.

6 Conclusion

The modified McEliece PKC proposed by Loidreau at Asiacrypt 2000 [13] employs interesting techniques using the Frobenius automorphism in Goppa codes. While it certainly improves the difficulty of breaking one-wayness against the "ever known" CPAs, it is vulnerable against our "new" CPAs, which exploit the modified structure, i.e. the biased 1's in a t-tower error vector. The binary work factor to break the one-wayness of the modified McEliece PKC with our new CPA is 2^{42}, which is feasible with currently available computational power.

Since our new attacks do not weaken the one-wayness of the (unmodified) McEliece PKC, it still satisfies OW-CPA for $n \geq 2048$ with optimum t and k. This means the (unmodified) McEliece PKC with an appropriate conversion still satisfies IND-CCA2.

Acknowledgments

The authors would like to thank anonymous referees for useful comments.

References

1. C. M. Adams and H. Meijer. "Security-Related Comments Regarding McEliece's Public-Key Cryptosystem". In *Proc. of CRYPTO '87, LNCS 293*, pages 224–228. Springer–Verlag, 1988.
2. M. Bellare and P. Rogaway. "Random Oracles are Practical: A Paradigm for Designing Efficient Protocols". In *Proc. of the First ACM CCCS*, pages 62–73, 1993.
3. T. Berson. "Failure of the McEliece Public-Key Cryptosystem Under Message-Resend and Related-Message Attack". In *Proc. of CRYPTO '97, LNCS 1294*, pages 213–220. Springer–Verlag, 1997.
4. A. Canteaut and N. Sendrier. "Cryptoanalysis of the Original McEliece Cryptosystem". In *Proc. of ASIACRYPT '98*, pages 187–199, 1998.

5. W. Diffie and M. Hellman. “New directions in cryptography”. *IEEE Trans. IT*, 22(6):644–654, 1976.
6. S. Goldwasser and S. Micali. “Probabilistic encryption”. *Journal of Computer and System Sciences*, pages 270–299, 1984.
7. C. Hall, I. Goldberg, and B. Schneier. “Reaction Attacks Against Several Public-Key Cryptosystems”. In *Proc. of the 2nd International Conference on Information and Communications Security (ICICS'99), LNCS 1726*, pages 2–12, 1999.
8. K. Kobara and H. Imai. “Countermeasure against Reaction Attacks (in Japanese)”. In *The 2000 Symposium on Cryptography and Information Security : A12*, January 2000.
9. K. Kobara and H. Imai. “Countermeasures against All the Known Attacks to the McEliece PKC”. In *Proc. of 2000 International Symposium on Information Theory and Its Applications*, pages 661–664, November 2000.
10. K. Kobara and H. Imai. “Semantically Secure McEliece Public-Key Cryptosystems –Conversions for McEliece PKC–”. In *Proc. of PKC '01, LNCS 1992*, pages 19–35. Springer–Verlag, 2001.
11. P. J. Lee and E. F. Brickell. “An Observation on the Security of McEliece's Public-Key Cryptosystem”. In *Proc. of EUROCRYPT '88, LNCS 330*, pages 275–280. Springer–Verlag, 1988.
12. R. Lidl and H. Niederreiter. *“Finite Fields”*, page 13. Cambridge University Press, 1983.
13. P. Loidreau. “Strengthening McEliece Cryptosystem”. In *Proc. of ASIACRYPT 2000*, pages 585–598. Springer–Verlag, 2000.
14. P. Loidreau and N. Sendrier. “Some weak keys in McEliece public-key cryptosystem”. In *Proc. of IEEE International Symposium on Information Theory, ISIT '98*, page 382, 1998.
15. R. J. McEliece. “A Public-Key Cryptosystem Based on Algebraic Coding Theory”. In *Deep Space Network Progress Report*, 1978.
16. N. Sendrier. “The Support Splitting Algorithm”. *Rapport de recherche: ISSN0249-6399*, 1999.
17. P.W. Shor. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.
18. J. Stern. “A method for finding codewords of small weight”. In *Proc. of Coding Theory and Applications , LNCS 388*, pages 106–113. Springer–Verlag, 1989.
19. H. M. Sun. “Further Cryptanalysis of the McEliece Public-Key Cryptosystem”. *IEEE Trans. on communication letters*, 4(1):18–19, 2000.
20. A. Vardy. “The Intractability of Computing the Minimum Distance of a Code”. *IEEE Trans. on IT*, 43(6):1757–1766, 1997.