On the One-Wayness Against Chosen-Plaintext Attacks of the Loidreau's Modified McEliece PKC

Kazukuni Kobara and Hideki Imai, *Fellow, IEEE*

***Abstract*—McEliece public-key cryptosystem (PKC) is one of a few alternatives for the current PKCs that are mostly based on either the integer factoring problem (IFP) or the discrete logarithm problem (DLP) that would be solved in polynomial time after the emergence of quantum computers. The security of the McEliece PKC is based on the decoding problem and it is known that it satisfies, with an appropriate conversion, the strongest security notion, i.e., INDistinguishability of encryption [10] against adaptively Chosen-Ciphertext Attacks (IND-CCA2), in the random oracle model under the assumption that the underlying primitive McEliece PKC satisfies a weak security notion of One-Wayness against Chosen-Plaintext Attacks (OW-CPA). OW-CPA is said to be satisfied if it is infeasible for chosen plaintext attacks to recover the whole plaintext of an arbitrarily given ciphertext.**

Currently, the primitive McEliece PKC satisfies OW-CPA if a parameter $n \geq 2048$ with optimum t and k is chosen since the binary work factor for $(n, k, t) = (2048, 1278, 70)$ to break it with the best CPA is around 2^{106}, which is infeasible even if world-wide computational power is used. While the binary work factor for the next smaller parameter $n = 1024$ is in a gray level of 2^{62}, it will be improved by applying Loidreau's modification that employs Frobenius automorphism in Goppa codes. In this paper, we carefully investigate the one-wayness of the Loidreau's modified McEliece PKC against ever known CPAs and new CPAs we propose, and then show that it certainly improves the one-wayness against ever known CPAs but it is vulnerable against our new CPAs. Thus, it is rather harmful to apply the new modification to the McEliece PKC.

***Index Terms*—Chosen-ciphertext attack, Goppa code, IND-CCA, McEliece PKC, one-wayness, public-key cryptosystem.**

I. Introduction

SINCE the concept of public-key cryptosystem (PKC) was introduced by Diffie and Hellman [9], 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 due to the cryptographers' 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, Diffie-Hellman, elliptic curve cryptosystems, and so on. They are all based on either integer factoring problem (IFP) or discrete logarithm problem (DLP). This situation would cause a serious problem if someone would discovers one practical algorithm that breaks both IFP and DLP in polynomial time. Actually, Shor has already found a (probabilistic) polynomial-time algorithm in [24], even though it requires a quantum computer that is impractical so far. Who can prove that other practical algorithms will never be found? 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 [21] is based on the nearest codeword problem (NCP), a problem where a word $x \in F_q^n$ and a generator matrix $G' \in F_q^{k \times n}$ with no visible structure are given and then find the nearest codeword $y \in G'$. While NCP is proven to be NP-hard in [5] and solving NCP does imply breaking McEliece PKC, breaking McEliece PKC is not as hard as solving NCP due to the following reasons: 1) the McEliece PKC is a special case of NCP where error weight is guaranteed to be a certain value and 2) adversaries of cryptosystems may use decryption oracles or may know partial information on the plaintext in advance.

It is, however, proven in the random oracle model [4] that the McEliece PKC with an appropriate conversion [15] satisfies the strongest security notion, i.e., INDistinguishability of encryption [10] against adaptively Chosen-Ciphertext Attacks (IND-CCA2), as long as the underlying primitive McEliece PKC, i.e., the McEliece PKC without any conversion, satisfies a weak security notion One-Wayness against Chosen-Plaintext Attacks (OW-CPA).[1]

OW-CPA is said to be satisfied if all the known chosen plaintext attacks cannot recover the whole plaintext of an arbitrarily given ciphertext within a practical time. Currently, the primitive McEliece (and Niederreiter) PKCs satisfy OW-CPA if a parameter $n \geq 2048$ with optimum t and k are chosen since the binary work factor for $(n, k, t) = (2048, 1278, 70)$ to break the one-wayness with the best CPA is around 2^{106}, which is infeasible even if world-wide computational power is used [1], [2], [17]. For $n = 1024$, the binary work factor is unfortunately in a gray level of 2^{62}.

In [19], Loidreau proposed a remarkable improvement of the McEliece PKC. It employs Frobenius automorphism in Goppa codes and then improves the binary work factor for breaking OW-CPA of the next smaller parameter $n = 1024$. If his idea works correctly, we may use the parameter $n = 1024$ safely. This makes the cryptosystem more compact. Even though his modification does not improve the immunity against attacks using decryption oracles [11], [12], [27] and partial knowledge

Manuscript received September 29, 2002; revised May 30, 2003. The material in this paper was presented in part at the 5th International Workshop on Practice and Theory in Public Key Cryptosystems, Paris, France, February 2002.

The authors are with the Institute of Industrial Science, The University of Tokyo, Tokyo, Japan. (e-mail: kobara@iis.u-tokyo.ac.jp; imai@iis.u-tokyo.ac.jp).

Communicated by T. Johansson, Associate Editor for Complexity and Cryptography.

Digital Object Identifier 10.1109/TIT.2003.820016

[1]Similar proof can also be given to the Niederreiter PKC using OAEP ++ [14] as its appropriate conversion scheme.

on the target plaintext [6], [13], it does not a matter since all of them can be prevented by applying an appropriate conversion [15].

In this paper, we investigate the one-wayness of the Loidreau's modified McEliece PKC against ever known CPAs [7], [16] and new CPAs we propose. Then, we show that his modification certainly improves the one-wayness against the "ever known" CPAs to a safe level of 2^{91} and 2^{86}, respectively, but our "new" CPAs reduce it to 2^{42}, which is a dangerous level. Thus, it is rather harmful to apply the new modification to the McEliece PKC.

This paper is organized as follows: in Sections II and III, we describe both the McEliece PKC and the ever known CPAs on the one-wayness of it, respectively. Then, in Section IV, we review the modified cryptosystem proposed by Loidreau [19]. Finally, in Section V, we show our new CPAs which weaken the one-wayness of the modified cryptosystem.

II. McEliece Public-Key Cryptosystem

A. Cryptosystem

The McEliece PKC consists of the following three algorithms.

Key generation:

Generate the following three matrices G, S, and 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 nonsingular matrix
- P $n \times n$ random permutation matrix.

Then output the following key pair:

Secret key: (S, P) and $\Psi(\ )$
Public key: (G', t)
where $G' = SGP$.

Encryption:

Let $\oplus$ denote the bitwise EXCLUSIVE–OR operation and z denote a random binary error vector of length n having t 1's. For a given message msg that is represented as a binary vector of length k, the corresponding ciphertext c is given by

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

Decryption:

Let P^{-1} denote the inverse of P. For a given ciphertext c, the corresponding plaintext msg is given as follows. At first, apply the decoding algorithm $\Psi()$ to cP^{-1}. Since

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

and the Hamming weight of $z \cdot P^{-1}$ is t, $\Psi()$ outputs $msg \cdot S$. The plaintext msg is given by $msg := (msg \cdot S)S^{-1}$.

B. Underlying Codes

The underlying codes of the McEliece PKC (and the Niederreiter PKC) must have the following two properties.

1) There exists an efficient algorithm to correct up to t errors.
2) The structure allowing correction up to t errors turns to be invisible by applying the secret matrices S and P.

It is important to notice is that the codes having the first property do not always satisfy the second property, and it is known that some codes do not satisfy the second property.

For example, Sidel'nikov and Shestakov have shown in [25] that the structure of (n, k)-GRS codes over GF(q) is visible with $O(n^4)$ operations[2] if S is a $k \times n$ nonsingular matrix over GF(q) and P is a permutation of the n coordinates over GF(q). Sendrier has shown in [23] that the permutation of randomly permuted concatenated codes can be recovered with a smaller complexity than that of breaking OW-CPA using the property of concatenated codes, i.e., the small-weight codes of the duals are closely related to the duals of their inner codes. Loidreau and Sendrier have shown in [20] that Support Splitting Algorithm (SSA)[3] [22] can determine the subfield over which the underlying Goppa polynomial is defined (if Goppa codes are used), and then if the cardinality of the remaining candidate polynomials is small enough to enumerate, it can determine the polynomial.

All these results suggest to us that the underlying codes should be chosen carefully. So far, it has been safe to use Goppa codes with large cardinality, say more than 2^{80}, after excluding the Goppa polynomials over small subfields.

III. Known Chosen Plaintext Attacks on One-Wayness of McEliece PKC

The following two kinds of attacks are known as the CPAs on the one-wayness of the McEliece PKC. They are summarized as follows.

A. Generalized Information-Set-Decoding Attack

The basic idea of this attack was proposed by Adams and Meijer in [3]. And then its improvement was proposed by Lee and Brickell in [16].

Their attacks work as follows: 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:

$$c_k = msg \cdot G'_k \oplus z_k. \tag{3}$$

If $z_k = 0$ and G'_k is nonsingular, msg can be recovered [3] by

$$msg = (c_k \oplus z_k)G'^{-1}_k. \tag{4}$$

Even if $z_k \neq 0$, msg can be obtained by guessing z_k among a small set $\{z_k | \mathrm{wt}(z_k) \leq j\}$ for a small j. The correctness of the recovered plaintext msg is verifiable by checking whether the Hamming weight of

$$c \oplus msg \cdot G' = c \oplus c_k G'^{-1}_k \cdot G' \oplus z_k G'^{-1}_k \cdot G' \tag{5}$$

is t or not.

[2]Precisely, $O(s^4+sn)$ arithmetical operations over GF(q) where $s = n-k-1$.

[3]SSA is a tool for testing whether two given codes are permutation equivalent, i.e., whether one code is obtained from the other by permuting the coordinates, and if so it recovers a permutation between them efficiently as long as the dimension of the hull (the intersection of the code with its dual) is small.

The corresponding algorithm is given 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 $\widehat{G'}_k := G'^{-1}_k G'$. 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) **(Process for the case of** $\mathrm{wt}(z'_k) = 0$**)** Calculate $\hat{z} := c \oplus c_k \widehat{G'}_k$. If $\mathrm{wt}(\hat{z}) = t$, output $msg := c_k G'^{-1}_k$.
   2.2) **(Process for the case of** $1 \le \mathrm{wt}(z'_k) \le j$**)** For i_1 from 1 to j do the following:
   i) For all the $\binom{n}{i_1}$ patterns of z'_k s.t. $\mathrm{wt}(z'_k) = i_1$, do the following:
   A) Generate a new pattern z'_k. If $\mathrm{wt}(\hat{z} \oplus z'_k \widehat{G'}_k) = t$, 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 $\widehat{G'}_k := G'^{-1}_k G'$ 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 an identity matrix. It can be obtained by the Gaussian elimination with the work factor of

$$\sum_{i=1}^{k} \frac{(k-1)(n-i+1)}{4} = \frac{k(k-1)(2n+1-k)}{8} \tag{6}$$

bit operations.

In Step 2.1 and Step A, one does not need to calculate the whole n coordinates of $c \oplus c_k \widehat{G'}_k$ and $\hat{z} \oplus z'_k \widehat{G'}_k$, 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 $c \oplus c_k \widehat{G'}_k$ in Step 2.1 is $t \cdot k/2$, and that of $\hat{z} \oplus z'_k \widehat{G'}_k$ in Step A is $t \cdot i_1$. Accordingly, the work factor for Step 2.2 is

$$V_j = \sum_{i_1=1}^{j} t \cdot i_1 \cdot \binom{k}{i_1}. \tag{7}$$

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

$$\frac{(k-1)(n-k)}{4}. \tag{8}$$

Since Step 2 is repeated around T_j times where

$$T_j = \frac{\binom{n}{k}}{\sum_{i=0}^{j} \binom{t}{i} \binom{n-t}{k-i}} \tag{9}$$

the total work factor is given by

$$W_j \approx \left\{ \frac{(k-1)(n-k)}{4} + \frac{t \cdot k}{2} + V_j \right\} \cdot T_j. \tag{10}$$

When n is given, both k and t should be chosen so that (10) should be the highest for the optimum j, which is chosen by adversaries so that (10) should be the lowest. 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.

B. Finding Low-Weight-Codeword Attack

This attack uses an algorithm that can find a small codeword of weight t in an arbitrarily given generator matrix.

Since the codeword of weight t of the following $(k+1) \times n$ generator matrix

$$\begin{bmatrix} G' \\ c \end{bmatrix} \tag{11}$$

is the error vector z in $c = msg \cdot G' \oplus z$, this algorithm can find z. Once z is found, msg can be obtained using the information-set decoding to $c \oplus z$.

The basic idea to find a small codeword efficiently was proposed by Stern in [26], and then it was improved by Canteaut *et al.* in [7], [8]. The corresponding algorithm is given 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 out of the $(k+1) \times n$ matrix of [11], and then apply Gaussian elimination so that the chosen $k+1$ columns should make an identity matrix. Let $\widehat{G'}_{k+1}$ denote the resulting $(k+1) \times n$ matrix. Let I and J denote a set of the indexes of the $k+1$ chosen columns and that of the remaining columns, respectively. Let M denote the $(k+1) \times (n-k-1)$ submatrix of $\widehat{G'}_{k+1}$ corresponding to J.
2) Do the following until a code word z of weight t is found:
   2.1) Split I into two subsets I_1 and I_2 at random where $|I_1| = \lfloor (k+1)/2 \rfloor$, $|I_2| = \lceil (k+1)/2 \rceil$ and $|X|$ denotes the cardinality of the set X. According to I_1 and I_2, split M into two submatrices M_1 and M_1 respectively, i.e., if I_1 includes an index i that is the i_1th member of I, then the i_1-th row of M is included in M_1.
   2.2) Select a ρ-element subset J_ρ out of J at random, and then let $M_{1|J_\rho}$ and

$M_{2|J_\rho}$ denote the corresponding ρ columns of M_1 and M_2, respectively.

2.3) $i := 0$ For all the $\binom{|I_1|}{p}$ combinations of p rows chosen out of the $|I_1|$ rows of $M_{1|J_\rho}$, do the following:
  i) Generate a new combination $\mathcal{P}_{1,i}$.
  ii) Sum up the chosen p rows of $M_{1|J_\rho}$ in F_2. Let $\Lambda_{1,i|J_\rho}$ denote the result.
  iii) Store both $\mathcal{P}_{1,i}$ and $\Lambda_{1,i|J_\rho}$ in a hash table with 2^ρ entries using $\Lambda_{1,i|J_\rho}$ as an index.
  iv) $i := i + 1$

2.4) $j := 0$ For all the $\binom{|I_2|}{p}$ combinations of p rows chosen out of the $|I_2|$ rows of $M_{2|J_\rho}$, do the following:
  i) Generate a new combination $\mathcal{P}_{2,j}$.
  ii) Sum up the chosen p rows of $M_{2|J_\rho}$ in F_2. Let $\Lambda_{2,j|J_\rho}$ denote the result.
  iii) Store both $\mathcal{P}_{2,j}$ and $\Lambda_{2,j|J_\rho}$ in a hash table with 2^ρ entries using $\Lambda_{2,j|J_\rho}$ as an index.
  iv) $j := j + 1$

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 $\mathrm{wt}(\Lambda_{1,i|J} \oplus \Lambda_{2,j|J}) = t - 2p$ 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, let z denote the codeword, and then go to Step 3.

2.6) Replace one coordinate in I with a coordinate in J, and then update $\widehat{G'}_{k+1}$ using Gaussian elimination so that the chosen $k+1$ columns in I should make an identity matrix.

3) Apply the information-set decoding to $c \oplus z$, and recover the corresponding message msg.

Under the assumption that each iteration is independent, one needs to repeat Step 2 around $T_{p,\rho}$ times[4] where

$$T_{p,\rho} = \frac{\binom{k+1-2p}{\frac{k+1}{2}-p}\binom{2p}{p}\binom{n-k-1+2p-t}{\rho}}{\binom{k+1}{\frac{k+1}{2}}\binom{n-k-1}{\rho}} \cdot \frac{\binom{n-t}{k+1-2p}\binom{t}{2p}}{\binom{n}{k+1}}. \tag{12}$$

In Steps 2.1–2.4, 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

$$\Omega_1(p,\rho) = p \cdot \rho \cdot \binom{(k+1)/2}{p}. \tag{13}$$

In Step 2.5, around $((k+1)/2p)^2/2^\rho$ pairs of $(\mathcal{P}_{1,i}, \mathcal{P}_{2,j})$ satisfy $\Lambda_{1,i|J_\rho} \oplus \Lambda_{2,j|J_\rho} = 0$, and for each pair one needs to check the weight of $\Lambda_{1,i|J} \oplus \Lambda_{2,j|J}$. Using the same idea in GISD, one can detect that $\mathrm{wt}(\Lambda_{1,i|J} \oplus \Lambda_{2,j|J}) \neq t - 2p$ by calculating the weight of around $2(t-2p)$ coordinates in $J \backslash J_\rho$. Thus, the binary work factor for Step 2.5 is around

$$\Omega_2(p,\rho) = 2(t-2p) \cdot p \cdot \frac{\binom{(k+1)/2}{p}^2}{2^\rho}. \tag{14}$$

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

$$\Omega_3(p,\rho) = \frac{k(n-k-1)}{4} \tag{15}$$

and the total binary work factor is given by

$$W_{p,\rho} \approx (\Omega_1(p,\rho) + \Omega_2(p,\rho) + \Omega_3(p,\rho)) \cdot T_{p,\rho}. \tag{16}$$

For $n = 2^{10}$,

$$\min_{p,\rho}(\max_{k,t}(W_{p,\rho})) \approx 2^{62}$$

which can be achieved when $(p,\rho) = (2,19)$, $t = 35$ to 44, and $k = n - m \cdot t = 674$ to 584, respectively. For $n = 2^{11}$

$$\min_{p,\rho}(\max_{k,t}(W_{p,\rho})) \approx 2^{106}$$

which can be achieved when $(p,\rho) = (2,22)$, $t = 62$ to 80, and $k = n - m \cdot t = 1366$ to 1168, respectively.

IV. Loidreau's Modified McEliece PKC

In [19], Loidreau proposed a modified version of the McEliece PKC using the Frobenius automorphism of the underlying Goppa code. The point of his modification is that it can improve the difficulty of breaking OW-CPA without increasing n. We will explain the underlying principles first.

A. Frobenius Automorphism Group of Goppa Codes

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

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

$$\forall c = (c_{\alpha_1}, \ldots, c_{\alpha_n}) \in \Gamma(L,g) \tag{17}$$

$$\sigma(c) = (c_{\sigma'(\alpha_1)}, \ldots, c_{\sigma'(\alpha_n)}) \in \Gamma(L,g) \tag{18}$$

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

B. Orbits Generated by the Frobenius Automorphism

The action of the Frobenius automorphism creates some orbits in the field. For an extension field $F_{(2^s)^{s_1}}$ of F_{2^s} of prime degree s_1, the action $\sigma' : x \mapsto x^{2^s}$ creates $N_{s_1} = \{(2^s)^{s_1} - 2^s\}/s_1$ orbits of length s_1 and $N_1 = 2^s$ orbits of length 1. For $i \in \{1, \ldots, N_{s_1}\}$, let $\mathcal{Z}_i$ describe the N_{s_1} orbits of length s_1 and $\mathcal{Z}_0$ denote a set of N_1 orbits of length 1.

[4]Precise expected number of iterations is estimated in [7], [8].

For example, let $\{z_{\alpha_1}, z_{\alpha_2}, \ldots, z_{\alpha_n}\}$ be a binary word of length n, α denote a generator of $F_{2^{10}}$, $s_1 = 5$, and $s = 2$. Then

$$\mathcal{Z}_0 = \{z_0, z_{\alpha^{341}}, z_{\alpha^{341\times 2}}, z_{\alpha^{341\times 3}}\}$$

is a set of $N_1 = 4$ orbits of length 1. Note that $\sigma'(0) = 0^4 = 0$ and

$$\sigma'(\alpha^{341\times i}) = \alpha^{341\times i\times 4} = \alpha^{341\times i}, \qquad \text{for } i \in \{1,2,3\}.$$

The orbits $\mathcal{Z}_1$ and $\mathcal{Z}_2$ may be

$$\{z_{\alpha^1}, z_{\alpha^4}, z_{\alpha^{16}}, z_{\alpha^{64}}, z_{\alpha^{256}}\}$$

and

$$\{z_{\alpha^2}, z_{\alpha^{2\times 4}}, z_{\alpha^{2\times 16}}, z_{\alpha^{2\times 64}}, z_{\alpha^{2\times 254}}\}$$

respectively. There are $N_5 = 204$ orbits of length 5 in total. Note that $\sigma'(\alpha^{256}) = \alpha^{256\times 4} = \alpha^1$ and $\sigma'(\alpha^{2\times 256}) = \alpha^{2\times 256\times 4} = \alpha^2$.

After reordering the labeling L, a word $\{z_{\alpha_1}, z_{\alpha_2}, \ldots, z_{\alpha_n}\}$ can be rewritten in the following form:

$$z = \{\mathcal{Z}_1, \mathcal{Z}_2, \ldots, \mathcal{Z}_{N_{s_1}}, \mathcal{Z}_0\}. \tag{19}$$

For the reordered coordinates, the action of σ on the word z is given as follows:

$$\sigma(z) = \{\sigma(\mathcal{Z}_1), \ldots, \sigma(\mathcal{Z}_{N_{s_1}}), \mathcal{Z}_0\} \tag{20}$$

where $\sigma(\mathcal{Z}_i)$ is a simple left cyclic shift in $\mathcal{Z}_i$. For $s_1 = 5$, $\mathcal{Z}_{i_1} = \{1,1,1,0,0\}$ and $\mathcal{Z}_{i_2} = \{1,1,0,1,0\}$, $\sigma^l(\mathcal{Z}_{i_1})$ and $\sigma^l(\mathcal{Z}_{i_2})$ for $l \in \{0, \ldots, s_1 - 1\}$ are given as follows:

$$\begin{aligned}
\mathcal{Z}_{i_1} &= \{1,1,1,0,0\}\\
\sigma(\mathcal{Z}_{i_1}) &= \{1,1,0,0,1\}\\
\sigma^2(\mathcal{Z}_{i_1}) &= \{1,0,0,1,1\}\\
\sigma^3(\mathcal{Z}_{i_1}) &= \{0,0,1,1,1\}\\
\sigma^4(\mathcal{Z}_{i_1}) &= \{0,1,1,1,0\}\\
\\
\mathcal{Z}_{i_2} &= \{1,1,0,1,0\}\\
\sigma(\mathcal{Z}_{i_2}) &= \{1,0,1,0,1\}\\
\sigma^2(\mathcal{Z}_{i_2}) &= \{0,1,0,1,1\}\\
\sigma^3(\mathcal{Z}_{i_2}) &= \{1,0,1,1,0\}\\
\sigma^4(\mathcal{Z}_{i_2}) &= \{0,1,1,0,1\}.
\end{aligned} \tag{21}$$

C. t-Tower Decodable Vector

Instead of random error vectors of weight t, the Loidreau's modified cryptosystem uses t-tower decodable vectors defined as follows.

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

Larger weight:
$\mathrm{wt}(z') > t$ where $\mathrm{wt}(x)$ denotes the Hamming weight of x.

Reducibility:
There exists a linear combination $f()$ s.t. $\mathrm{wt}(z) \le t$ where

$$z = f(z') = \sum_{i=0}^{m/s-1} b_i \cdot \sigma^i(z'), \qquad b_i \in F_2. \tag{22}$$

Recoverability:
z' is recoverable from the reduced z in (22).

In [19], $s_1 = 5$ is used, and the corresponding 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 $u = \lfloor t/2 \rfloor$ orbits out of the N_5 orbits (of length 5).
3) Flip 3 bits each at random in the chosen u orbits.

It is obvious that the generated z' satisfies the larger-weight property since $\mathrm{wt}(z') = 3 \cdot \lfloor t/2 \rfloor > t$. The reducibility is satisfied using $f_1()$ or $f_2()$, s.t.

$$z = f_1(z') = z' + \sigma(z') + \sigma^2(z') \tag{23}$$

$$z = f_2(z') = z' + \sigma^2(z') + \sigma^3(z'). \tag{24}$$

The reason why either $f_1()$ or $f_2()$ can satisfy the reducibility can be explained as follows. All the binary vectors of length 5 and weight 3 can be expressed in either $\sigma^l(\mathcal{Z}_{i_1})$ or $\sigma^l(\mathcal{Z}_{i_2})$ in (21) for certain l's. By using $f_1()$ or $f_2()$, respectively, all of them are transformed into the following vectors:

$$\begin{aligned}
f_1\left(\sigma^l(\mathcal{Z}_{i_1})\right) &= \sigma^l\left(\mathcal{Z}_{i_1} + \sigma(\mathcal{Z}_{i_1}) + \sigma^2(\mathcal{Z}_{i_1})\right)\\
&= \sigma^l(\{1,0,1,1,0\})\\
f_1\left(\sigma^l(\mathcal{Z}_{i_2})\right) &= \sigma^l\left(\mathcal{Z}_{i_2} + \sigma(\mathcal{Z}_{i_2}) + \sigma^2(\mathcal{Z}_{i_2})\right)\\
&= \sigma^l(\{0,0,1,0,0\})\\
f_2\left(\sigma^l(\mathcal{Z}_{i_1})\right) &= \sigma^l\left(\mathcal{Z}_{i_1} + \sigma^2(\mathcal{Z}_{i_1}) + \sigma^3(\mathcal{Z}_{i_1})\right)\\
&= \sigma^l(\{0,1,0,0,0\})\\
f_2\left(\sigma^l(\mathcal{Z}_{i_2})\right) &= \sigma^l\left(\mathcal{Z}_{i_2} + \sigma^2(\mathcal{Z}_{i_2}) + \sigma^3(\mathcal{Z}_{i_2})\right)\\
&= \sigma^l(\{0,0,1,1,1\})
\end{aligned} \tag{25}$$

where $\sigma^l()$ denotes l-bit left cyclic shift. Let u_1 and u_2 denote the number of subvectors that are in the form of $\sigma^l(\mathcal{Z}_{i_1})$ and $\sigma^{l'}(\mathcal{Z}_{i_2})$, respectively, in z'. Since $u_1 + u_2 = u = \lfloor t/2 \rfloor$

$$\begin{aligned}
\min\left(f_1(z'), f_2(z')\right) &= \min(3u_1 + u_2, u_1 + 3u_2)\\
&= \min\left(2u_1 + \lfloor t/2 \rfloor, \lfloor t/2 \rfloor + 2u_2\right)\\
&\le 2 \cdot \lfloor t/2 \rfloor\\
&\le t.
\end{aligned} \tag{26}$$

Thus, the weight of z' is reduced within t using either $f_1()$ or $f_2()$.

The recoverability holds since both $f_1()$ and $f_2()$ are one-to-one mappings and z' is uniquely determined from z.

D. Loidreau's Modified Cryptosystem

Loidreau's modified cryptosystem uses $s_1 = 5$ and a Goppa polynomial g (of degree t) in which all the coefficients are from the 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 hide the structure of g. Employing

TABLE I
CARDINALITIES OF SYSTEM PARAMETERS

Cryptosystem	Permutations	Goppa polynomials	Error vectors	Orders of orbits
McEliece PKC [21]	$n!$	$\frac{(2^m)^{(t+1)}}{t}$	$\binom{n}{t}$	-
Loidreau's modified cryptosystem [19]	$(5!)^{N_5} \cdot N_5!$	$\frac{(2^m)^{(t_1+1)}}{t_1} \cdot \frac{(2^s)^t}{t}$	$10^{\lfloor t/2 \rfloor} \cdot \binom{N_5}{\lfloor t/2 \rfloor}$	$(4!)^{N_5}$

TABLE II
CARDINALITIES OF SYSTEM PARAMETERS FOR $(n, k, t, t_1, s, N_5) = (1024,624,40,2,2,204)$

Cryptosystem	Permutations	Goppa polynomials	Error vectors	Orders of orbits
McEliece PKC [21]	2^{8769}	2^{405}	2^{240}	-
Loidreau's modified cryptosystem [19]	2^{2685}	2^{104}	2^{157}	2^{935}

TABLE III
BINARY WORK FACTORS TO BREAK OW-CPA FOR $(n, k, t, t_1, s, N_5) = (1024,624,40,2,2,204)$

Systems \ Attacks	Ever known attacks		Our new attacks	
	GISD[16]	FLWC[7]	Attack I	Attack II
McEliece PKC [21]	2^{67}	2^{62}	-	-
Loidreau's modified cryptosystem [19]	2^{91}	2^{86}	2^{60}	2^{42}

g_1 makes it possible to enlarge the cardinality of gg_1 and thus reduces the risk of the enumeration (exhaustive search) attack in [20].

The cardinality of gg_1 is approximately given by

$$\{(2^{5s})^{(t_1+1)}/t_1\} \cdot \{(2^s)^t/t\}$$

since the number of irreducible monic polynomials of degree x over F_{2^y} is around $(2^y)^x/x$ [18]. The decoding algorithm of $\Gamma(L, gg_1)$ is the same as that of $\Gamma(L, g)$ since $\Gamma(L, gg_1)$ is a subcode of $\Gamma(L, g)$. The information length of $\Gamma(L, gg_1)$ is given by $k - t_1 m$ bits.

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 units of the orbits of length 5 are open to the public as a part of the public key (but the order of the orbit in each unit is kept secret). Note that the units of the orbits can be opened without increasing the public key size as follows. Instead of permuting all the columns in L as P, one permutes units first, and then permutes the columns in each unit. Since the units of the orbits are the same as L, anyone can know them. While this reduces the cardinality of the permutations P, it still maintains a large amount (see Tables I and II, respectively).
- The order of orbit in each unit is kept in secret.
- t-tower decodable vectors z' are used instead of the random error vectors of weight t.

The corresponding decryption process is given as follows

Algorithm 4 (Decryption of the Loidreau's modified McEliece PKC)

Input: a ciphertext c.
Output: a plaintext msg or an error message.

1) Apply $f_1()$ and $f_2()$ to the given ciphertext c, respectively.
2) Decode $f_1(c)$ and $f_2(c)$ using the decoding algorithm for $\Gamma(L, g)$.
3) If either $f_1(c)$ or $f_2(c)$ is decoded:
   3.1) Using the decoded error vector z, reconstruct the corresponding t-tower decodable vector z'.
   3.2) Apply the information-set decoding to $c \oplus z'$, and then recover the corresponding message msg.
4) If neither $f_1(c)$ nor $f_2(c)$ are decoded, output an error message.

In Step 2, at least one of $f_1(c)$ and $f_2(c)$ can be decoded when a proper ciphertext is given, since the Hamming weight of the error vector of at least one of them is smaller than or equal to t.

E. One-Wayness of the Modified Cryptosystem Against Ever Known Chosen-Plaintext Attacks

Since the Loidreau's modification enlarges the Hamming weight of the error vector t to $3\lfloor t/2 \rfloor$, the binary work factors of the ever known CPAs on the one-wayness are improved. For example, for $n = 2^{10}$ and $t_1 = 2$, the binary work factor against the GISD attack is improved to $\min_j(\max_{k',t}(W_j)) \approx 2^{91}$, which can be achieved when $j = 1$, $t = 35$ to 43, and the corresponding $k' = n - m \cdot (t + t_1) = 654$ to 574. That against the FLWC attack is improved to $\min_{p,\rho}(\max_{k',t}(W_{p,\rho})) \approx 2^{86}$, which can be achieved when $(p, \rho) = (2,19)$, $t = 35$ to 44, and the corresponding $k' = n - m \cdot (t + t_1) = 654$ to 564. These results are summarized in Table III.

Even though his modification employs a new secret, i.e., the order of orbit in all the orbit unit, and also it reduces the cardinality of 1) permutations P, 2) Goppa polynomials gg_1, and 3) error vectors z', respectively, they still preserve enough amounts to avoid exhaustive search for them (see Tables I and II). Thus, guessing one of them does not work.

V. Our New Chosen-Plaintext Attacks on the Modified Cryptosystem

A. Attack I

Before we explain our new Attack I, recall that the Loidreau's modified cryptosystem uses the following equations:

$$f_1(c) = msg' \cdot G' \oplus f_1(z') \tag{27}$$

$$f_2(c) = msg'' \cdot G' \oplus f_2(z') \tag{28}$$

which hold for certain messages msg' and msg''. The point of these equations is that only those who know the "correct" orbit order in all the orbit units can use them, but anyone who does not know it cannot use them. If one applies them in the "wrong" orbit order in some orbit units, they look as if a noise were added, and thus the Hamming weight of the corresponding error vector increases.

On the other hand, our Attack I applies $f_1()$ and $f_2()$ to c regardless of its orbit order, and then applies them to all of the rows in G', as well. Let $f_1'()$ and $f_2'()$ denote the same transformations as $f_1()$ and $f_2()$ except for the underlying orbit order, i.e., while $f_1()$ and $f_2()$ use the right orbit order, $f_1'()$ and $f_2'()$ use a random orbit order in each orbit unit. Due to the linearity of $f_1'(\,)$ and $f_2'(\,)$, the following equations hold:

$$f_1'(c) = msg \cdot f_1'(G') + f_1'(z') \tag{29}$$

and

$$f_2'(c) = msg \cdot f_2'(G') + f_2'(z'). \tag{30}$$

Since both GISD and FLWC are the generic decoding algorithms for arbitrary linear codes, one can apply them to both $f_1'(c)$ and $f_2'(c)$ without knowing the underlying algebraic structure of $f_1'(G')$ and $f_2'(G')$. Moreover, the Hamming weight of either $f_1'(z')$ or $f_2'(z')$ is no larger than t, and thus one can find it with smaller complexity than finding z' (of weight around $3\lfloor t/2 \rfloor$) in c.

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 to 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[5] both on the pair of $f_1'(c)$ and $f_1'(G')$, and on the pair of $f_2'(c)$ and $f_2'(G')$ respectively, and then find out a message msg.

If $f_1'(G')$ and $f_2'(G')$ have a lot of nonzero codewords of weight smaller than or equal to t, FLWC may capture wrong codewords. The probability is, however, negligibly small since the number of such codewords is expected to follow

$$\#\{f_i'(c) : 1 \le \mathrm{wt}(f_i'(c)) \le t\} \approx 2^{k'} \frac{\left(\sum_{i=1}^{t} \binom{n}{i}\right)}{2^n} \tag{31}$$

for $i \in \{1, 2\}$ if $f_1'(G')$ and $f_2'(G')$ act as random functions. Note that

$$\#\{f_i'(c) : 1 \le \mathrm{wt}(f_i'(c)) \le t\} \approx 5.6 \times 10^{-55}$$

for $(n, k', t) = (1024, 604, 40)$. Precisely, $f_1'(\,)$ and $f_2'(\,)$ are not necessarily random functions, but they modify the Hamming weight in each orbit: 5 to 5, 4 to 2, 3 to $\{3$ or $1\}$, 2 to $\{4$ or $2\}$, and 1 to 3, respectively. This guarantees that the minimum weight of $f_1'(G')$ and $f_2'(G')$ can be no smaller than $d/3$ and the number of nonzero codewords of weight smaller than or equal to t is negligibly small.

Thus, the work factor of this algorithm is almost the same as that of the FLWC attack except that k is replaced with $k' = n - m \cdot (t + t_1)$ and then it is run twice. For $t_1 = 2$, the binary work factor against this attack is reduced to

$$\min_{p,\rho}(\max_{k',t}(2W_{p,\rho})) \approx 2^{60}$$

that can be achieved when $(p, \rho) = (2, 19)$, $t = 34$ to 45, and the corresponding $k' = n - m \cdot (t + t_1) = 674$ to 544.

B. Attack II

Recall that the GISD attack ties to choose k coordinates of almost zeros out of the n coordinates of the invisible error vector z. While it is difficult to choose $k' = n - m \cdot (t + t_1)$ almost zeros out of the t-tower decodable vector z' whose weight is $3\lfloor t/2 \rfloor$, we found that it is easier to choose $\lceil (k' - 2^s)/5 \rceil$ orbits (which correspond to more than or equal to k' coordinates) of almost zeros out of the N_5 orbits where only $\lfloor t/2 \rfloor$ orbits are nonzeros.

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 $\lceil (k' - 2^s)/5 \rceil$ orbits out of the N_5 orbits. Let I_o denote a set of the indexes of all the coordinates in the chosen $\lceil (k' - 2^s)/5 \rceil$ orbits and of the 2^s coordinates corresponding to the 2^s orbits of length 1. Let J_o denote a set of the remaining coordinates.
2) Choose k' coordinates out of I_o so that the corresponding k' columns in G' should be independent. Let $G'_{k'}$ denote the $k' \times k'$ matrix composed of the chosen k' columns in G'. Let $c_{k'}$ and $z'_{k'}$ denote the corresponding k'-dimensional vectors in c and z', respectively. Calculate $\widehat{G'}_{k'} := G'^{-1}_{k'} G'$.
3) Do the following until msg is found:

[5]The FLWC attack here tries to find out a codeword of weight not only t but also smaller than t.

3.1) **(Process for the case of $\mathrm{wt}(z'_{k'})=0$)** Calculate $\hat{z'} := c \oplus c_{k'}\widehat{G'}_{k'}$, which is the error vector corresponding to the case of $\mathrm{wt}(z'_{k'})=0$. If $\hat{z'}$ is a t-tower decodable vector, output $msg := c_{k'}G'^{-1}_{k'}$.

3.2) **(Process for the case that $z'_{k'}$ contains up to j nonzero orbits)** For i_1 from 1 to j do the following:

i) For all the patterns of $z'_{k'}$ that contains i_1 nonzero orbits, do the following:

A) Generate a new pattern $z'_{k'}$. If $\hat{z} \oplus z'_{k'}\widehat{G'}_{k'}$ is a t-tower decodable vector, output $msg := (c_{k'} \oplus z'_{k'})G'^{-1}_{k'}$.

3.3) Replace one orbit of length 5 in I_o with a new orbit in J_o, and then renew both I and $\widehat{G'}_{k'} := G'^{-1}_{k'}G'$ using Gaussian elimination.

Similarly to the GISD attack, the binary work factor of this algorithm is given as follows.

In Step 3, T_j iterations are expected to find out msg where

$$T_j = \frac{\binom{N_5}{\lceil (k'-2s)/5 \rceil}}{\sum_{i=0}^{j} \binom{\lfloor t/2 \rfloor}{i}\binom{N_5-\lfloor t/2 \rfloor}{\lceil (k'-2^s)/5 \rceil - i}}. \tag{32}$$

In Step 3.1 and Step A, one does not need to calculate the whole n coordinates of $\hat{z'}$ and $\hat{z} \oplus z'_{k'}\widehat{G'}_{k'}$, respectively, to detect that they are not t-tower decodable vectors since their weights exceed $3/2t$ with around $3t$ coordinates in J provided that wrong cases have the average weight of $n/2$.[6] Thus, the binary work factors of Step 3.1 and Step A are given by $3t \cdot k'/4$ and $3t \cdot (3/2)i_1$, respectively.

In Step i), there are at most

$$\binom{5}{3}^{i_1} \cdot \binom{\lceil (k'-2^s)/5 \rceil}{i_1}$$

patterns of $z'_{k'}$, thus, the binary work factor of Step 3.2 is given by

$$V_j \approx \sum_{i_1=1}^{j} 3t \cdot \frac{3}{2} i_1 \cdot \binom{5}{3}^{i_1} \cdot \binom{\lceil (k'-2^s)/5 \rceil}{i_1}. \tag{33}$$

The binary work factor of Step 3.3 is $5(k'-1)(n-k')/4$. Thus, the total work factor is given by

$$W_j \approx \left\{ \frac{5(k'-1)(n-k')}{4} + \frac{3t \cdot k'}{4} + V_j \right\} \cdot T_j. \tag{34}$$

For $t_1 = 2$, the binary work factor against this attack is reduced to $\min_j(\max_{k',t}(W_j)) \approx 2^{42}$ that can be achieved when $j=1$, $t=38$ to 43, and the corresponding $k' = n - m \cdot (t + t_1) = 624$ to 574.

[6]One can also detect that by seeing whether or not the weight of each orbit is either 0 or 3.

VI. How to Avoid Our New Attacks

The work factor of Attack II depends on the number of nonzero orbits u (rather than the total weight of z'). Thus, if we could choose a larger u than $\lfloor t/2 \rfloor$, the immunity against it would be improved. Unfortunately, the theoretical upper bound of u is t and $u = \lfloor t/2 \rfloor$ is the maximum value among the possible t-tower decodable vectors for $n = 1024$. Thus, we cannot avoid this attack by enlarging u for $n = 1024$.

Moreover, even if we could avoid Attack II, we would not avoid Attack I since one can easily find the reduction function $f'()$ for Attack I. Keeping $f()$ in secret does not work since it is not difficult for anyone to find $f'()$ from the orbit length and the weight of nonzero orbits in z', which must be open to the public so that any sender can generate z' to encrypt a message.

The only measure to avoid our attacks is to use the usual error vectors z where t 1's are uniformly distributed in z (this means we should use the unmodified McEliece PKC or the unmodified Niederreiter PKC with a conversion such as OAEP++ [14])[7] since our attacks are not applicable to the unbiased error vectors.

VII. Conclusion

We investigated the one-wayness of the Loidreau's modified McEliece PKC against ever known CPAs and our new CPAs. While his modification certainly improves it against the "ever known" CPAs, it is unfortunately vulnerable against our "new" CPAs that exploit the modified structure, i.e., the biased 1's in the t-tower decodable vector.

Since our new attacks are not applicable to the (unmodified) McEliece PKC, the best method currently known to enhance OW-CPA of the McEliece (and the Niederreiter) PKC is simply to increase the parameter n (and then to optimize both t and k) instead of using t-tower decodable vectors employed in the Loidreau's modified McEliece PKC.

Acknowledgment

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

References

[1] (1997) RSA Labs' 64bit RC5 Encryption Challenge. [Online]. Available: http://stats.distributed.net/

[2] (1999) RSA code-breaking contest again won by distributed.net and Electronic Frontier Foundation (EFF). [Online]. Available: http://www.rsasecurity.com/news/pr/990119–1.html

[3] C. M. Adams and H. Meijer, "Security-related comments regarding McEliece's public-key cryptosystem," in *Proc. CRYPTO '87 (Lecture Notes in Computer Science)*. Berlin, Germany: Springer-Verlag, 1988, vol. 293, pp. 224–228.

[4] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *Proc. 1st ACM CCCS*, 1993, pp. 62–73.

[5] E. R. Berlekamp, R. J. McEliece, and H. van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 384–386, May 1978.

[6] T. Berson, "Failure of the McEliece public-key cryptosystem under message-resend and related-message attack," in *Proc. CRYPTO '97 (Lecture Notes in Computer Science)*. Berlin, Germny: Springer-Verlag, 1997, vol. 1294, pp. 213–220.

[7]In the Niederreiter PKC, conversions can also be used to generate randomized error vectors.

[7] A. Canteaut and F. Chabaud, "A new algorithm for finding minimum-weight words in a linear code: Application to McEliece's cryptosystem and to narrow-sense bch codes of length 511," *IEEE Trans. Inform. Theory*, vol. 44, pp. 367–378, Jan. 1998.

[8] A. Canteaut and N. Sendrier, "Cryptoanalysis of the original McEliece cryptosystem," in *Proc. ASIACRYPT '98*, 1998, pp. 187–199.

[9] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 644–654, Nov. 1976.

[10] S. Goldwasser and S. Micali, "Probabilistic encryption," *J. Comput. Syst. Sci.*, pp. 270–299, 1984.

[11] C. Hall, I. Goldberg, and B. Schneier, "Reaction attacks against several public-key cryptosystems," in *Proc. 2nd Int. Conf. Information and Communications Security (ICICS'99) (Lecture Notes in Computer Science)*. Berlin, Germany: Springer-Verlag, 1999, vol. 1726, pp. 2–12.

[12] K. Kobara and H. Imai, "Countermeasure against reaction attacks (in Japanese)," presented at the The 2000 Symposium on Cryptography and Information Security, Jan. 2000. Paper A12.

[13] ——, "Countermeasures against all the known attacks to the McEliece PKC," in *Proc. 2000 Int. Symp. Information Theory and Its Applications*, Nov. 2000, pp. 661–664.

[14] ——, "OAEP++—Another very simple way to fix the bug in OAEP−," in *Proc. 2002 Int. Symp. Information Theory and Its Applications*, 2002, pp. 563–566. Paper S6-4-5.

[15] ——, "Semantically secure McEliece public-key cryptosystem," *IEICE Trans.*, vol. E85-A, no. 1, pp. 74–83, Jan. 2002.

[16] P. J. Lee and E. F. Brickell, "An observation on the security of McEliece's public-key cryptosystem," in *Proc. EUROCRYPT '88 (Lecture Notes in Computer Science)*. Berlin, Germany: Springer-Verlag, 1988, vol. 330, pp. 275–280.

[17] A. K. Lenstra and E. R. Verheul, "Selecting cryptographic key sizes," *J. Cryptology*, vol. 14, no. 4, pp. 255–293, 2001.

[18] R. Lidl and H. Niederreiter, *Finite Fields*. Cambridge, U.K.: Cambridge Univ. Press, 1983, p. 13.

[19] P. Loidreau, "Strengthening McEliece cryptosystem," in *Proc. ASIACRYPT 2000*, 2000, pp. 585–598.

[20] P. Loidreau and N. Sendrier, "Weak keys in McEliece public-key cryptosystem," *IEEE Trans. Inform. Theory*, vol. 47, pp. 1207–1212, Mar. 2001.

[21] R. J. McEliece, "A public-key cryptosystem based on algebraic coding theory," in *Deep Space Network Progress Report*, 1978.

[22] N. Sendrier, "The Support Splitting Algorithm," report, Rapport de Recherche: ISSN0249-6399, 1999.

[23] ——, "On the structure of randomly permuted concatenated code," in *Proc. EUROCODE 94*, Abbeye de la Bussiere sur Ouche, France, Oct. 24–28, 1994, pp. 169–173.

[24] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. Computing*, vol. 26, no. 5, pp. 1484–1509, 1997.

[25] V. M. Sidel'nikov and S. O. Shestakov, "On insecurity of cryptosystems based on generalized reed-solomon codes," *Discr. Math. Applic.*, vol. 2, no. 4, pp. 439–444, 1992.

[26] J. Stern, "A method for finding codewords of small weight," in *Proc. Coding Theory and Applications (Lecture Notes in Computer Science)*. Berlin, Germany: Springer-Verlag, 1989, vol. 388, pp. 106–113.

[27] H. M. Sun, "Further cryptanalysis of the McEliece public-key cryptosystem," *IEEE Commun. Lett.*, vol. 4, pp. 18–19, Jan. 2000.