

Enhancing the Security of the McEliece Public-Key Cryptosystem

HUNG-MIN SUN

*Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan 701, R.O.C.
E-mail: hmsun@mail.ncku.edu.tw*

Berson pointed out that the McEliece public-key cryptosystem suffers from two weaknesses: (1) failure to protect any message which is encrypted more than once; and (2) failure to protect any messages which have a known linear relationship with one another. In this paper, we propose some variants of the McEliece scheme to prevent Berson's attacks. In addition, we also propose some secure and efficient variants of the McEliece scheme in order to improve the information rate. On the other hand, designing a public-key cryptosystem which is theoretically secure has become more and more important since the introduction of notions of public-key encryption security by Bellare, Desai, Pointcheval, and Rogaway. In this paper, we also propose a variant of the McEliece scheme that is non-malleable under the adaptive chosen ciphertext attack.

Keywords: cryptography, public-key cryptosystem, McEliece scheme, error-correcting codes, goppa codes, information rate

1. INTRODUCTION

McEliece proposed the first public-key cryptosystem (*the McEliece scheme*) based on algebraic coding theory in 1978 [1]. The idea behind this public-key cryptosystem is based on the fact that the decoding problem of an arbitrary linear code is an NP-hard problem [2]. Compared with other public-key cryptosystems [3, 4] which involve modular exponentiation, the McEliece scheme has the advantage of high-speed encryption and decryption. In addition, the McEliece scheme employs probabilistic encryption [5, 6], which is better than other types of deterministic encryption [4, 7] in preventing the elimination of any information leaked through public-key cryptography. So far, the McEliece scheme is still not widely used. This is because the information rate of this scheme is low (close to 0.5), and it requires large binary matrices for the secret key and public key. Some methods [8-10] have been proposed to improve the information rate of the McEliece scheme. These methods use the added error vector to carry additional information. Some information bits are mapped into an error vector to be added to a codeword. Once the error vector can be identified, the additional information can be recovered. By using these methods, the information rate can be increased up to around 0.8 or higher. For the large key problem, Sun and Hwang [11] proposed the use of a short sequence of bits (called seed-key) to specify the secret key. Thus, each user only needs to keep a short key, e.g., 64-bit sequence. However, the problem of a large public key has still not been solved.

Received October 17, 1998; revised May 15, 1999; accepted June 23, 1999.
Communicated by Chi Sung Laih.

In the past, many researchers [12-17] attempted to break the McEliece scheme. None of these were successful in the general case. Among them, Korzhik and Turkin [15] claimed that they had broken the McEliece scheme. However, most cryptographers don't believe their result is effective because of a lack of obvious evidence needed to confirm the time bound they obtained. At Crypto '97, Berson [18] showed that the McEliece scheme suffers from two weaknesses: (1) failure to protect any message which is encrypted more than once and (2) failure to protect any messages which have a known linear relationship with one another. Although these weaknesses don't make it possible to break the McEliece scheme immediately (i.e., the private key can't be recovered), it is possible for an attacker to perform an action such that these weaknesses appear. For example, an attacker may introduce some errors into the ciphertext, which is sent from the sender to the receiver, such that the receiver cannot decrypt the ciphertext correctly. If the receiver thinks this is caused by faults in the encryption phase, he will ask the sender to send it again (encrypt the message and send the ciphertext again). Thus, the weakness (1) will appear.

In this paper, we propose some variants of the McEliece public-key cryptosystem to prevent Berson's attacks. These variants do not reduce the information rate of the original scheme. In addition, to improve the information rate, we also propose some variants of the McEliece scheme that can prevent Berson-like attacks. On the other hand, designing a public-key cryptosystem that is theoretically secure has become more and more important since the introduction of notions of public-key encryption security by Bellare, Desai, Pointcheval, and Rogaway [19]. In this paper, we also propose a variant of the McEliece scheme that is non-malleable under the adaptive chosen ciphertext attack. This paper is organized as follows. In section 2, we provide some background information. In section 3, we present some variants of the McEliece public-key cryptosystem to prevent Berson's attacks. In section 4, we propose more variants of the McEliece public-key cryptosystem that can prevent Berson-like attacks and improve the information rate. In section 5, we propose a variant of the McEliece scheme that is non-malleable under the adaptive chosen ciphertext attack. Finally, we conclude this paper in section 6.

2. PRELIMINARIES

2.1 The McEliece Public-Key Cryptosystem

Secret key: S is a random $(k \times k)$ nonsingular matrix over $GF(2)$, called the scrambling matrix,

G is a $(k \times n)$ generator matrix of a binary Goppa code G with the capability of correcting an n -bit random error vector of weight less than or equal to t , and P is a random $(n \times n)$ permutation matrix.

Public key: $G' = S G P$

Encryption: $c = mG' + e$, where m is a k -bit message, c is an n -bit ciphertext, and e is an n -bit random error vector of weight t .

Decryption: The receiver first calculates $c' = cP^{-1} = mSG + eP^{-1}$, where P^{-1} is the inverse of P . Because the weight of eP^{-1} is the same as the weight of e , the receiver uses the decoding algorithm of the original code G to obtain $m' = mS$. Finally, the receiver recovers m by computing $m = m'S^{-1}$, where S is the inverse of S .

In the original version of the McEliece scheme, the parameters k , n , and t were suggested to be 524, 1024, and 50, respectively. Many works [12,13, 20, 21] were to study the optimal values of these parameters such that a cryptanalyst must take the highest cost to break this system. Optimizations were suggested where if $n = 1024$, then k ranges from 524 to 654, and t ranges from 37 to 50. In this paper, we use the parameter sizes of the original version without loss of generality.

An obvious attack on the McEliece scheme is to guess 524 positions of c that are not distorted by e , and to then find m from $c^* = mG^*$ if G^* is invertible, where c^* and G^* are restrictions on these positions of c and G' . Because there exist 50 errors embedded in 1024

positions, we need $\frac{\binom{1024}{524}}{\binom{974}{524}} \approx 1.37 \times 10^{16}$ guesses to succeed.

2.2 Berson’s Attacks on the McEliece Scheme

Berson proposed two attacks on the McEliece scheme, called the message-resend attack and related-message attack. We restate these two attacks in the following.

Message-Resend Attack:

We assume that a message m is encrypted twice because of some accident or a special action of a cryptanalyst. Then, the cryptanalyst knows that $c_1 = mG' + e_1$ and $c_2 = mG' + e_2$, where $e_1 \neq e_2$ (which is called the message-resend condition). Therefore, $c_1 + c_2 = e_1 + e_2$. It is noted that the weight of $e_1 + e_2$ is even and at most 100 because the weight of each error vector added in the McEliece scheme is 50. According to Berson’s analysis, the expected Hamming weight of $e_1 + e_2$ is about 95.1 if a message-resend condition occurs. If the underlying messages are different, then the expected Hamming weight of $c_1 + c_2$ is 512. Therefore, it is easy to detect the occurrence of a message-resend condition and the weight of $e_1 + e_2$ by observing the Hamming weight of $c_1 + c_2$. If the weight of $e_1 + e_2$ is 94, we need to guess 524 positions of $c_1(c_2)$ that are not distorted by $e_1(e_2)$ from 930 possible positions

with 3 wrong positions. The probability that we can get a correct guess is $\frac{\binom{927}{524}}{\binom{930}{524}} \approx 0.0828$.

This means that the cryptanalyst needs only about 12 guesses to succeed. Similarly, if the weight of $e_1 + e_2$ is 96, then only about 5 guesses are required for the cryptanalyst to succeed.

Note that the main reason why Berson’s attack succeeds is that by observing the value of $c_1 + c_2$, it is possible to obtain more *information* about the positions in which errors will probably occur.

Related-Message Attack:

We assume that two messages m_1 and m_2 are encrypted, and that the cryptanalyst knows a linear relation, e.g., the value $m_1 + m_2$, between these two messages. Then, the cryptanalyst knows that $c_1 = m_1G' + e_1$ and $c_2 = m_2G' + e_2$, where $m_1 \neq m_2$ and $e_1 \neq e_2$.

Therefore, $c_1 + c_2 = m_1G' + e_1 + m_2G' + e_2 = (m_1 + m_2)G' + (e_1 + e_2)$. Because the value $m_1 + m_2$ is known previously, $(m_1 + m_2)G'$ can be computed. Hence, $c_1 + c_2 + (m_1 + m_2)G' = e_1 + e_2$. Similar to the case of the message-resend attack, the number of guesses required to succeed is small. Basically, the message-resend attack is a special case of the related-message attack, where the linear relation between the messages is $m_1 + m_2 = 0$. To overcome these weaknesses, Berson [18] suggested spreading randomness through the plaintext in some complicated fashion; e.g., Bellare and Rogaway's OAEP [22] et seq. are instructive. Thus, the linear relation between the messages can not be found through some action of a cryptanalyst. Of course, any such scheme extracts a penalty in information rate. In the following sections, we will propose some variants of the McEliece scheme to prevent Berson's attacks. Two of them have the same information rate as the original McEliece scheme, and two of them have higher information rates than the original scheme.

2.3 Notations of Encryption Scheme Security

In the following, we first introduce the security notions in the public-key encryption. These notions were organized by Bellare, Desai, Pointcheval, and Rogaway recently [19].

Various notions of security for public key encryption were proposed to evaluate the strength of a public key cryptosystem in the past. Bellare, Desai, Pointcheval, and Rogaway [19] organized definitions of secure encryption by considering separately the various possible *goals* and the various possible *attack* models, and then obtaining each definition as a pair of a particular goal and a particular attack model. They considered two different goals: *indistinguishability of encryptions*, due to Goldwasser and Micali [23], and *non-malleability* (NM), due to Dolev, Dwork and Naor [24]. Indistinguishability (IND) formalizes an adversary's inability to learn any information about the plaintext m underlying a challenge ciphertext. This captures a strong notion of privacy. Non-malleability (NM) formalizes an adversary's inability, given a challenge ciphertext c , to get a different ciphertext c' such that the corresponding plaintexts m and m' are *meaningfully related*, e.g. $m' = 2m$. This captures a sense in which ciphertexts can be tamper-proof. They considered three different attack models: *chosen plaintext attack* (CPA), *non-adaptive chosen ciphertext attack* (CCA1) due to Naor and Yung [25], and *adaptive chosen ciphertext attack* (CCA2) due to Rackoff and Simon [26]. Under CPA, the adversary is given the public-key and is able to obtain ciphertexts of plaintexts of his choice. Under CCA1, the adversary is given the public-key and is able to get access to an oracle for the decryption function before the challenge ciphertext c is given. Under CCA2, the adversary is given the public-key and is able to get access to an oracle for the decryption function at any time. Mixing and Matching the goals {IND, NM} and the attack models {CPA, CCA1, CCA2} in any combination, six notions of security can be obtained. These are IND-CPA, IND-CCA1, IND-CCA2, NM-CPA, NM-CCA1, and NM-CCA2. In Fig. 1, we show the relations among each notion of the security according to [19]. Note that for $A, B \in \{\text{IND-CPA, IND-CCA1, IND-CCA2, NM-CPA, NM-CCA1, NM-CCA2}\}$ " $A \rightarrow B$ " denotes that an encryption scheme being secure in the sense of A is also secure in the sense of B . This also implies that an encryption scheme being insecure in the sense of B is also insecure in the sense of A .

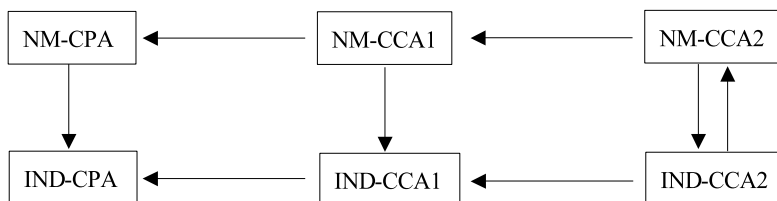


Fig. 1. The relations among each notion of the security.

Note that from Fig. 1, we understand the strongest security notions in the public-key encryption are NM-CCA2 and IND-CCA2, which have been shown to be equivalent in [19].

3. SOME VARIANTS OF THE MCELIECE SCHEME

In this section, we will propose some variants of the McEliece scheme. These variants can protect the McEliece scheme from the message-resend attack and the related-message attack. In addition, these variants will not reduce the information rate. The public key and the secret key in these variants are the same as those in the original McEliece scheme.

Variant I:

Encryption: $c = (m + h(e))G' + e$, where e is an n -bit random error vector of weight t , and h is a one-way hash function with an input e and an output of a k -bit vector. It is necessary to consider how to apply a well-known one-way hash function, e.g., MD5 [27], as the function h .

Decryption: First, $m + h(e)$ can be obtained by using the decryption algorithm in the original scheme. (The error vector can also be found in the decoding process.) Secondly, the receiver computes $m = (m + h(e)) + h(e)$.

Security: Let m_1 and m_2 be two messages. If $m_1 = m_2$, then $c_1 + c_2 = (h(e_1) + h(e_2))G' + e_1 + e_2$. The value $(h(e_1) + h(e_2))G'$ is unknown because of a lack knowledge about $h(e_1)$ and $h(e_2)$. We cannot obtain more information about the positions in which the error occurs. Thus, the message-resend attack fails. If the value $m_1 + m_2$ is known, then $c_1 + c_2 = (m_1 + m_2 + h(e_1) + h(e_2))G' + e_1 + e_2$. Although the value $m_1 + m_2$ is known, $(m_1 + m_2 + h(e_1) + h(e_2))G'$ will not be known because of a lack of the knowledge about $h(e_1)$ and $h(e_2)$. We are not able to obtain any information about the positions in which the error occurs. Thus, the related-message attack cannot succeed.

Variant II:

Encryption: $c = f(m, e)G' + e$, where e is an n -bit random error vector of weight t , and f is a trapdoor one-way function [28] with two inputs (m and e) and an output of a k -bit vector. Here, f must satisfy the property that given $f(m, e)$, it is computationally infeasible to find m and e , but it is easy to compute m given $f(m, e)$ and e . For example, DES [29], which has two inputs (message and key) and an output (ciphertext), can be one candidate. If DES is applied, it is necessary to consider how to implement it as the function f because DES has

a 56-bit key, a 64-bit message, and a 64-bit ciphertext, while f needs an n -bit e , a k -bit m , and a k -bit output. Generally speaking, $k = 524$ and $n = 1024$. We can use a 56-bit hash value $h(e)$ to represent the secret key of DES and then encrypt m block by block (64 bits/per block). Because 524 is not a multiple of 64, the last block is only 12 bits in size, say (m_{11}, \dots, m_{112}) . Assume that the ciphertext corresponding to the first block is $(c_{f1}, \dots, c_{f52}, c_{f53}, \dots, c_{f64})$. Hence, we can use $(c_{f1}, \dots, c_{f52}, m_{11}, \dots, m_{112})$ as the last block (plaintext). Thus, we no longer need to keep the information (c_{f1}, \dots, c_{f52}) as the ciphertext. Therefore, the total amount of ciphertext is still 524 bits, which is the same as that of plaintext. This technique is usually referred as *ciphertext stealing* [28, 30].

- Decryption:** First, $f(m, e)$ can be recovered by using the decryption algorithm in the original scheme. (The receiver keeps the error vector in the decoding process.) Secondly, the receiver computes m by inverting the function f .
- Security:** If $m_1 = m_2$, then $c_1 + c_2 = (f(m_1, e_1) + f(m_2, e_2))G' + e_1 + e_2$. The value $(f(m_1, e_1) + f(m_2, e_2))G'$ is unknown because of a lack of knowledge about $f(m_1, e_1)$ and $f(m_2, e_2)$. We cannot obtain any information about the positions in which the error occurs. Thus, the message-resend attack fails. If the value $m_1 + m_2$ is known, we cannot erase the item $(f(m_1, e_1) + f(m_2, e_2))G'$. Therefore, this scheme is also secure against the related-message attack.

4. MORE WAYS TO IMPROVE THE INFORMATION RATE

In the past, some researchers [8-10] studied how to improve the information rate of the McEliece scheme. They used the added error vector to carry additional information. Thus, the information rate of the McEliece scheme could be increased. Here, we will formally present their ideas as Variant III.

Variant III:

- Encryption:** Let $m = (m_a, m_b)$ be the message.
 $c = m_a G' + e$, where $e = g(m_b)$, g is an invertible function which maps m_b into an n -bit error vector of weight t . Some good candidates of the function g can be found in [8-10].
- Decryption:** First, m_a can be recovered by using the decryption algorithm in the original scheme. In the meantime, the value $g(m_b)$ can also be obtained. Then, the receiver computes $m_b = g^{-1}(g(m_b))$, where g^{-1} is the inverse of g .
- Information rate:** By using this method, the information rate can be improved from 0.51 to 0.79 if $k = 524$, $n = 1024$, and $t = 50$ (additional 284-bit information is carried), and from 0.63 to 0.87 if $k = 654$, $n = 1024$, and $t = 37$ (additional 225-bit information is carried).
- Security:** Basically, the idea behind this variant is the same as that behind the original McEliece scheme. The main difference between them is in the randomness of the error vector. The error vector of the former is not truly random but is dependent on the probability distribution of m_b . To provide better security, it is suggested that data compression technique be applied before encryption. Note that this variant is a type of deterministic encryption.

Let $m_1 = (m_{1a}, m_{1b})$ and $m_2 = (m_{2a}, m_{2b})$ be two encrypted messages. Because each message in this variant contains two parts, we can extend the linear relation between two messages to many cases. In Table 1, we show the possible weaknesses of these cases. We will try to explain these cases in the following.

Case III.A: If m_{1a} is known previously, then $g(m_{1b}) = c_1 + m_{1a}G'$. Thus, $m_{1b} = g^{-1}(g(m_{1b}))$.

Case III.B: If m_{1b} is known previously, then we know that $m_{1a}G' = c_1 + g(m_{1b})$. It is easy to compute m_{1a} by finding $m_{1a}G^* = (c_1 + g(m_{1b}))^*$, where $(c_1 + g(m_{1b}))^*$ and G^* are restrictions on some positions of $c_1 + g(m_{1b})$ and G' such that G^* is invertible.

Case III.C: If $m_{1a} = m_{2a}$ and $m_{1b} = m_{2b}$ are known previously, then $c_1 = c_2$. That is, $c_1 + c_2 = 0$. We cannot obtain any information about the positions in which the error occurs.

Case III.D: If $m_{1a} = m_{2a}$ and $m_{1b} \neq m_{2b}$ are known previously, then $e_1 \neq e_2$. Thus, $c_1 + c_2 = (m_{1a} + m_{2a})G' + e_1 + e_2 = e_1 + e_2$. Therefore, we can obtain information about the positions in which the errors occur. Thus, m_{1a} , m_{1b} , m_{2a} , and m_{2b} can be known.

Case III.E: If $m_{1a} \neq m_{2a}$ and $m_{1b} = m_{2b}$ are known previously, then $(m_{1a} + m_{2a})G' = c_1 + c_2$. Similar to Case III.B, it is easy to compute $m_{1a} + m_{2a}$.

Case III.F: This is similar to Case III.E except that $m_{1a} + m_{2a}$ is known previously.

Case III.G: If the values $m_{1a} + m_{2a}$ and $m_{1b} \neq m_{2b}$ are known previously, then $c_1 + c_2 = (m_{1a} + m_{2a})G' + e_1 + e_2$. Because the value $m_{1a} + m_{2a}$ is known, $(m_{1a} + m_{2a})G'$ can be computed. Hence, $c_1 + c_2 + (m_{1a} + m_{2a})G' = e_1 + e_2$. Therefore, we can obtain information about the positions in which the errors occur. Thus, m_{1a} , m_{1b} , m_{2a} , and m_{2b} can be known.

From Table 1, it is clear that there are still many weaknesses in Variant III. To overcome these weaknesses and improve the information rate of the McEliece scheme, we propose two variants of the McEliece scheme in the following.

Table 1. The possible weaknesses in Variant III.

	Information Known Previously	Information Leaked
Case III.A	m_{1a} (or m_{2a})	m_{1b} (or m_{2b})
Case III.B	m_{1b} (or m_{2b})	m_{1a} (or m_{2a})
Case III.C	$m_{1a} = m_{2a}, m_{1b} = m_{2b}$	None
Case III.D	$m_{1a} = m_{2a}, m_{1b} \neq m_{2b}$	$m_{1a}, m_{1b}, m_{2a}, m_{2b}$
Case III.E	$m_{1a} \neq m_{2a}, m_{1b} = m_{2b}$	$m_{1a} + m_{2a}$
Case III.F	$m_{1a} + m_{2a}, m_{1b} = m_{2b}$	None
Case III.G	$m_{1a} + m_{2a}, m_{1b} \neq m_{2b}$	$m_{1a}, m_{1b}, m_{2a}, m_{2b}$

Variation IV:

Encryption: Let $m = (m_a, m_b)$ be the message.

$c = (m_a + h(e))G' + e$, where $e = g(r|m_b)$, r is a q -bit random vector, and h and g are the same functions as those in Variation I and Variation III, respectively. Here, we need the function g to satisfy the following property. Let E be the set of 2^n all possible strings of n binary digits, let E_{m_b} be the set of all possible outputs of $g(r|m_b)$ given m_b , let x_i be the i -th item in E_{m_b} and let $d_i = \min_{j, j \neq i} \{dist.(x_i, x_j)\}$. If we regard E as an n -dimensional Hamming space, then we require that E_{m_b} be uniformly distributed (located) in E . That is, we expect E_{m_b} to have an approximately optimal value of $\bar{d}_i (= \frac{\sum d_i}{2^q})$. The schemes proposed in [8-10] are good candidates for the function g .

Decryption: First, $m_a' = m_a + h(e)$ can be obtained by using the decryption algorithm in the original scheme. In the meantime, e can also be obtained. Secondly, the receiver computes $r|m_b = g^{-1}(e)$, where g^{-1} is the inverse of g , and then discards the r part. Thus, m_b is obtained. Finally, m_a can be computed by means of $m_a = m_a' + h(e)$.

Information rate: By using this method, the information rate can be improved from 0.51 to 0.79 if $k = 524$, $n = 1024$, $t = 50$, and $q = 0$; from 0.51 to 0.73 if $k = 524$, $n = 1024$, $t = 50$, and $q = 64$; from 0.63 to 0.87 if $k = 654$, $n = 1024$, $t = 37$, and $q = 0$; and from 0.63 to 0.8 if $k = 654$, $n = 1024$, $t = 37$, and $q = 64$.

Security: We will discuss the security of this variation with the parameters $q = 0$ and $q = 64$, respectively.

Parameter $q = 0$:

In Table 2, we show the possible weaknesses in Variation IV with parameter $q = 0$. Some cases are explained in the following.

Case IV.A: Assume that m_{1a} is known previously. $(m_{1a} + h(g(m_{1b}))G'$ cannot be removed from c_1 because $h(g(m_{1b}))$ is unknown.

Case IV.B: If m_{1b} is known previously, then we know that $(m_{1a} + h(g(m_{1b}))G' = c_1 + g(m_{1b})$. Similar to Case III.B, it is easy to compute $m_{1a} + h(g(m_{1b}))$ and, hence, m_{1a} .

Case IV.C: This is similar to Case III.C.

Table 2. The possible weaknesses in Variation IV with parameter $q = 0$.

	Information Known Previously	Information Leaked
Case IV.A	m_{1a} (or m_{2a})	None
Case IV.B	m_{1b} (or m_{2b})	m_{1a} (or m_{2a})
Case IV.C	$m_{1a} = m_{2a}$, $m_{1b} = m_{2b}$	None
Case IV.D	$m_{1a} = m_{2a}$, $m_{1b} \neq m_{2b}$	None
Case IV.E	$m_{1a} \neq m_{2a}$, $m_{1b} = m_{2b}$	$m_{1a} + m_{1b}$
Case IV.F	$m_{1a} + m_{2a}$, $m_{1b} = m_{2b}$	None
Case IV.G	$m_{1a} + m_{2a}$, $m_{1b} \neq m_{2b}$	None

Case IV.D: If $m_{1a} = m_{2a}$ and $m_{1b} \neq m_{2b}$ are known previously, then $c_1 + c_2 = (h(g(m_{1b}) + h(g(m_{2b})))G' + e_1 + e_2$. We cannot remove $(h(g(m_{1b}) + h(g(m_{2b})))G'$ from $c_1 + c_2$. Therefore, we cannot obtain any information about the positions in which the errors occur.

Case IV.E: This is similar to Case III.E.

Case IV.F: This is similar to Case III.F.

Case IV.G: If the values $m_{1a} + m_{2a}$ and $m_{1b} \neq m_{2b}$ are known previously, then $c_1 + c_2 = (m_{1a} + m_{2a} + h(g(m_{1b}) + h(g(m_{2b})))G' + e_1 + e_2$. Because the value $m_{1a} + m_{2a}$ is known, $(m_{1a} + m_{2a})G'$ can be computed. Hence, $c_1 + c_2 + (m_{1a} + m_{2a})G' = h(g(m_{1b}) + h(g(m_{2b})))G' + e_1 + e_2$. However, we cannot remove $(h(g(m_{1b}) + h(g(m_{2b})))G'$ from $c_1 + c_2 + (m_{1a} + m_{2a})G'$.

Parameter $q = 64$:

In Table 3, we show the possible weaknesses in Variant IV with parameter $q = 64$. Some cases are explained in the following.

Table 3. Thhe possible weaknesses in Variant IV with parameter $q = 64$.

	Information Known Previously	Information Leaked
Case IV.R.A	m_{1a} (or m_{2a})	None
Case IV.R.B	m_{1b} (or m_{2b})	None
Case IV.R.C	$m_{1a} = m_{2a}, m_{1b} = m_{2b}$	None
Case IV.R.D	$m_{1a} = m_{2a}, m_{1b} \neq m_{2b}$	None
Case IV.R.E	$m_{1a} \neq m_{2a}, m_{1b} = m_{2b}$	None
Case IV.R.F	$m_{1a} + m_{2a}, m_{1b} = m_{2b}$	None
Case IV.R.G	$m_{1a} + m_{2a}, m_{1b} \neq m_{2b}$	None

Case IV.R.A: This is similar to Case IV.A.

Case IV.R.B: Assume that m_{1b} is known previously. Because r_1 is an unknown 64-bit random vector, the probability that we can get a correct guess of the value $g(r_1||m_{1b})$ is only $\frac{1}{2^{64}}$. Therefore, we cannot remove $g(r_1||m_{1b})$ from c_1 . Another possible attack is to guess k positions of c that are not distorted by e . Because $E_{m_{1b}}$ is uniformly distributed in E , a cryptanalyst cannot identify which positions have better chances.

Case IV.R.C: If $m_{1a} = m_{2a}$ and $m_{1b} = m_{2b}$ are known previously, then $c_1 + c_2 = (h(g(r_1||m_{1b}) + h(g(r_2||m_{2b})))G' + g(r_1||m_{1b}) + g(r_2||m_{2b}))$. We cannot remove $(h(g(r_1||m_{1b}) + h(g(r_2||m_{2b})))G'$ from $c_1 + c_2$ because m_{1a}, m_{2a}, m_{1b} , and m_{2b} are unknown.

Case IV.R.D: This is similar to Case IV.D.

Case IV.R.E: If $m_{1a} \neq m_{2a}$ and $m_{1b} = m_{2b}$ are known previously, then $c_1 + c_2 = (m_{1a} + m_{2a} + h(g(r_1||m_{1b}) + h(g(r_2||m_{2b})))G' + g(r_1||m_{1b}) + g(r_2||m_{2b}))$. Because r_1 is a 64-bit random vector, the probability that $r_1 = r_2$ (hence, $g(r_1||m_{1b}) = g(r_2||m_{2b})$) is equal to $\frac{1}{2^{64}}$, which is in significantly small. Therefore, neither $(m_{1a} + m_{2a} + h(g(r_1||m_{1b}) + h(g(r_2||m_{2b})))G'$ nor $g(r_1||m_{1b}) + g(r_2||m_{2b})$ can be removed from $c_1 + c_2$.

Case IV.R.F: If the values $m_{1a} + m_{1b}$ and $m_{2a} = m_{2b}$ are known previously, then $c_1 + c_2 + (m_{1a} + m_{2a})G' = h(g(r_1||m_{1b}) + h(g(r_2||m_{2b}))G' + g(r_1||m_{1b}) + g(r_2||m_{2b})$. Neither $(h(g(r_1||m_{1b}) + h(g(r_2||m_{2b}))G'$ nor $g(r_1||m_{1b}) + g(r_2||m_{2b})$ can be removed from $c_1 + c_2 + (m_{1a} + m_{2a})G'$.

Case IV.R.G: This is similar to Case IV.G.

Variant V:

Encryption: Let $m = (m_a, m_b)$ be the message.

$c = f(m_a, e)G' + e$, where $e = g(r||m_b)$, and r is a q -bit random vector, f and g are the same functions as those in Variant II and Variant III, respectively. Here the function g must satisfy the same property as that in Variant IV.

Decryption: First, $m_a' = f(m_a, e)$ can be obtained by using the decryption algorithm in the original scheme. In the meantime, e can also be recovered. Secondly, the receiver computes $r||m_b = g^{-1}(e)$, where g^{-1} is the inverse of g and then discards the part r . Thus, m_b is obtained. Finally, m_a can be computed by $m_a = f^{-1}(m_a', e)$, where f^{-1} is the inverse of f .

Information rate: The same as in Variant IV.

Security: We will discuss the security of this variant with parameters $q = 0$ and $q = 64$, respectively.

Table 4. The possible weaknesses in Variant V with parameter $q = 0$.

	Information Known Previously	Information Leaked
Case V.A	m_{1a} (or m_{2a})	None
Case V.B	m_{1b} (or m_{2b})	m_{1a} (or m_{2a})
Case V.C	$m_{1a} = m_{2a}, m_{1b} = m_{2b}$	None
Case V.D	$m_{1a} = m_{2a}, m_{1b} \neq m_{2b}$	None
Case V.E	$m_{1a} \neq m_{2a}, m_{1b} = m_{2b}$	None
Case V.F	$m_{1a} + m_{2a}, m_{1b} = m_{2b}$	None
Case V.G	$m_{1a} + m_{2a}, m_{1b} \neq m_{2b}$	None

Parameter $q = 0$:

In Table 4, we will show the possible weaknesses in Variant V with parameter $q = 0$. Some cases are explained in the following.

Case V.A: This is similar to Case IV.A.

Case V.B: If m_{1b} is known previously, then we know $f(m_{1a}, g(m_{1b}))G' = c_1 + g(m_{1b})$. Similar to Case III.B, it is easy to compute $f(m_{1a}, g(m_{1b}))$ and, hence, $m_{1a} = f^{-1}(f(m_{1a}, g(m_{1b})), g(m_{1b}))$.

Case V.C: This is similar to Case III.C.

Case V.D: If $m_{1a} = m_{2a}$ and $m_{1b} \neq m_{2b}$ are known previously, then $c_1 + c_2 = (f(m_{1a}, g(m_{1b})) + f(m_{2a}, g(m_{2b}))G' + e_1 + e_2$. We cannot erase $(f(m_{1a}, g(m_{1b})) + f(m_{2a}, g(m_{2b}))G'$ from $c_1 + c_2$.

Case V.E: If $m_{1a} \neq m_{1b}$ and $m_{2a} = m_{2b}$ are known previously, then $c_1 + c_2 = (f(m_{1a}, g(m_{1b})) + f(m_{2a}, g(m_{2b})))G' + f(m_{2a}, g(m_{2b}))$. We can only obtain the value $f(m_{1a}, g(m_{1b})) + f(m_{2a}, g(m_{2b}))$.

Case V.F: This is similar to Case V.E.

Case V.G: This is similar to Case V.D.

Table 5. The possible weaknesses in Variant V with parameter $q = 64$.

	Information Known Previously	Information Leaked
Case V.R.A	m_{1a} (or m_{2a})	None
Case V.R.B	m_{1b} (or m_{2b})	None
Case V.R.C	$m_{1a} = m_{2a}, m_{1b} = m_{2b}$	None
Case V.R.D	$m_{1a} = m_{2a}, m_{1b} \neq m_{2b}$	None
Case V.R.E	$m_{1a} \neq m_{2a}, m_{1b} = m_{2b}$	None
Case V.R.F	$m_{1a} + m_{2a}, m_{1b} = m_{2b}$	None
Case V.R.G	$m_{1a} + m_{2a}, m_{1b} \neq m_{2b}$	None

Parameter $q = 64$:

In Table 5, we show the possible weaknesses in Variant V with parameter $q = 64$. Some cases are explained in the following.

Case V.R.A: This is similar to Case IV.R.A.

Case V.R.B: This is similar to Case IV.R.B.

Case V.R.C: If $m_{1a} = m_{2a}$ and $m_{1b} = m_{2b}$ are known previously, then $c_1 + c_2 = (f(m_{1a}, g(r_1||m_{1b})) + f(m_{2a}, g(r_2||m_{2b})))G' + g(r_1||m_{1b}) + g(r_2||m_{2b})$. We can remove neither $(f(m_{1a}, g(r_1||m_{1b})) + f(m_{2a}, g(r_2||m_{2b})))G'$ nor $g(r_1||m_{1b}) + g(r_2||m_{2b})$ from $c_1 + c_2$.

Case V.R.D: This is similar to Case V.D.

Case V.R.E: This is similar to Case IV.R.E.

Case V.R.F: This is similar to Case V.R.C.

Case V.R.G: This is similar to Case V.G.

5. NON-MALLEABLE VARIANT AGAINST ADAPTIVE CHOSEN CIPHERTEXT ATTACK

In this section, we propose a variant of the McEliece public-key cryptosystem. The proposed variant satisfies the non-malleability property under the adaptive chosen ciphertext attack model. This variant is the following.

Encryption: Let $m = (m_a, m_b)$ be the message.

$c = (m_a || h(m_a, e))G' + e$, where $e = g(r||m_b)$, r is a q -bit random vector (q is a security parameter and is larger than 64), h and g are the same functions as those in Variant I and Variant III respectively. Here the function g must satisfy the same property as that in Variant IV.

Decryption: First $m_a || h(m_a, e)$ can be obtained by using the decryption algorithm in the original scheme. In the meantime, e can also be recovered. Secondly the decryption algorithm computes $r || m_b = g^{-1}(e)$, where g^{-1} is the inverse of g , and then discards the part r . Thus, m_b is obtained. At last, the decryption algorithm computes $h(m_a, e)$ and then compares it with the part in $m_a || h(m_a, e)$. If they match each other, then the decryption algorithm accepts and outputs $m = (m_a, m_b)$ as a valid message; otherwise, outputs ‘reject’.

Security: We discuss the security of this variant. Under the adaptive chosen ciphertext attack model, we assume that an adversary is given the public-key and is able to get access to an oracle for the decryption function at any time. We observe whether given a challenge ciphertext c , the adversary can get a different ciphertext c' such that the corresponding plaintexts m and m' are meaningfully related.

Here we first consider the case when the adversary tries to product c' directly from c . It is clear that any change made by the adversary with respect to m_a , e.g., $m_a' = m_a + 1$, will lead to the change with respect to $h(m_a, e)$. Because both m_a and e are unknown to the adversary, he cannot obtain the change with respect to $h(m_a, e)$. Note that if he can obtain the change with respect to $h(m_a, e)$, he can compute c' by $c' = c + dG'$, where d is the difference of $m_a || h(m_a, e)$ and $m_a' || h(m_a', e)$. Similarly, any change made by the adversary with respect to m_b will lead to the change with respect to $e = g(r || m_b)$ and $h(m_a, e)$. Because r, m_a and m_b are unknown to the adversary, he cannot obtain the change with respect to $g(r || m_b)$ and $h(m_a, e)$.

Secondly, we consider the case when the adversary tries to product a ciphertext c' that is ‘valid’ but different from c . Here ‘valid’ means that the decryption oracle will output something. Note that if he can do so, he may send c' into the decryption oracle in order to obtain some information on the plaintext m . It is clear that any random modification to c will lead to non-existence of decoding due to the property of Goppa code. So, the only possible approach is to change few bits (e.g., 2 bits) in c , (we assume the result is c'). We expect that this change will leads to c' being decodable for the Goppa code. In fact, the occurrence probability isn’t too small. The adversary need only try a number of times to get such a c' (decodable). Note that such a c' will be $(m_a || h(m_a, e))G' + e'$, where e and e' differ in few bits. Unfortunately, the decryption oracle will output ‘reject’ because $h(m_a, e')$ is different from $h(m_a, e)$.

6. CONCLUSIONS

In this paper, we first proposed two variants, Variant I and Variant II, of the McEliece scheme, which can protect against both the message-resend attack and the related-message attack. These two variants are probabilistic encryptions and have the same information rate as the original McEliece scheme. To improve the information rate and to protect against Berson-like attacks, we have also proposed two other variants, Variant IV and Variant V, of the McEliece scheme. In these two variants, if the parameter q is equal to 0, then they are two types of deterministic encryption and can improve the information rate from 0.51 to 0.79 if $k = 524$, $n = 1024$, and $t = 50$; or from 0.63 to 0.87 if $k = 654$, $n = 1024$, and $t = 37$. If the parameter q is equal to 64, then they are two kinds of probabilistic encryption and can improve the information rate from 0.51 to 0.73 if $k = 524$, $n = 1024$, and $t = 50$; or from 0.

63 to 0.8 if $k = 654$, $n = 1024$, $t = 37$. Finally, we propose a more secure variant of the McEliece scheme according to the notions of public-key encryption security. The proposed variant satisfies the non-malleability property under the adaptive chosen ciphertext attack model.

REFERENCES

1. R. J. McEliece, "A public-key cryptosystem based on algebraic coding theory," *DSN Progress Report*, 42-44, 1978, pp. 114-116.
2. E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Transactions on Information Theory*, Vol. 24, No. 5, 1978, pp. 384-386.
3. T. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, Vol. 31, No. 4, 1985, pp. 469-472.
4. R. L. Rivest, A. Shamir, and L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, Vol. 21, No. 2, 1978, pp. 120-126.
5. M. Blum and S. Goldwasser, "An efficient probabilistic public-key encryption scheme which hides all partial information," *Advances in Cryptology-CRYPTO '84*, Lecture notes in computer science (Springer-Verlag), 1985, pp. 289-299.
6. S. Goldwasser and S. Micali, "Probabilistic encryption and how to play mental poker keeping secret all partial information," in *Proceedings of the 14th ACM Symposium on the Theory of Computing*, 1982, pp. 270-299.
7. M. O. Rabin, "Digital signatures and public-key functions as intractable as factorization," MIT Lab. for Computer Science, Technical Report, MIT/LCS/TR-212, 1979.
8. M. C. Lin and H. L. Fu, "Information rate of McEliece's public-key cryptosystem," *Electronics Letters*, Vol. 26, No. 1, 1990, pp. 16-18.
9. C. S. Park, "Improving code rate of McEliece's public-key cryptosystem," *Electronics Letters*, Vol. 25, No. 21, 1989, pp. 1466-1467.
10. N. Sendrier, "Efficient generation of binary words of given weight," *Cryptography and Coding: 5th IMA Conference*, 1995, pp. 184-187.
11. H. M. Sun and T. Hwang, "Key generation of algebraic-code cryptosystems," *Computers and Mathematics with Applications*, Vol. 27, No. 2, 1994, pp. 99-106.
12. C. Adams and H. Meijer, "Security-related comments regarding McEliece's public-key cryptosystem," *Advances in Cryptology-CRYPTO '87*, Lecture notes in computer science, 1988, pp. 224-228.
13. C. Adams and H. Meijer, "Security-related comments regarding McEliece's public-key cryptosystem," *IEEE Transactions on Information Theory*, Vol. 35, No. 2, 1989, pp. 454-455.
14. E. F. Brickell and A. Odlyzko, "Cryptanalysis: a survey of recent results," in *Proceedings of IEEE*, Vol. 76, No. 5, 1988, pp. 153-165.
15. V. I. Korzhik and A. I. Turkin, "Cryptanalysis of McEliece's public-key cryptosystem," *Advances in Cryptology-EUROCRYPT '91*, Lecture notes in computer science, 1991, pp. 68-70.
16. P. J. Lee and E. F. Brickell, "An observation on the security of McEliece's public-key

- cryptosystem,” *Advances in Cryptology-EUROCRYPT '88*, Lecture notes in computer science, 1988, pp. 275-280.
17. J. van Tilburg, “On the McEliece public-key cryptosystem,” *Advances in Cryptology-CRYPTO '88*, Lecture notes in computer science, 1990, pp. 119-131.
 18. T. A. Berson, “Failure of the McEliece public-key cryptosystem under message-resend and related-message attack,” *Advances in Cryptology-CRYPTO '97*, Lecture notes in computer science, 1997, pp. 213-220.
 19. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, “Relations among notions of security for public-key encryption schemes,” *Advances in Cryptology-CRYPTO '98*, Lecture Notes in Computer Science, Vol.1462, 1998, pp. 26-45.
 20. P. J. M. Hin, “Channel-error-correcting privacy cryptosystems,” M.Sc. Thesis, Delft University of Technology, 1986.
 21. F. Jorissen, “A security evaluation of the public-key cipher system proposed by McEliece, used as a combined scheme,” Technical Report, Dept. of Elektrotechniek, Katholieke University Leuven, 1986.
 22. M. Bellare and P. Rogaway, “Optimal asymmetric encryption,” *Advances in Cryptology-CRYPTO '94*, Lecture notes in computer science, 1997, pp. 232-249.
 23. S. Goldwasser and S. Micali, “Probabilistic encryption,” *Journal of Computer and System Science*, Vol. 28, No. 2, 1984, pp. 270-199.
 24. D. Dolev, C. Dwork, and M. Naor, “Non-malleable cryptography,” in *Proceedings of the 23rd Annual Symposium on Theory of Computing, ACM*, 1991, pp. 542-552.
 25. M. Naor and M. Yung, “Public-key cryptosystems provably secure against chosen ciphertext attacks,” in *Proceedings of the 22nd Annual Symposium on Theory of Computing, ACM*, 1990, pp. 427-437.
 26. C. Rackoff and D. Simon, “Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack,” *Advances in Cryptology-CRYPTO '91*, Lecture notes in computer science, Vol. 576, 1992, pp. 433-444.
 27. R. L. Rivest, “The MD5 message digest algorithm,” RFC 1321, 1992.
 28. B. Schneier, *Applied Cryptography*, John Wiley & Sons, 1996.
 29. National Bureau of Standards, NBS FIPS PUB 46, “Data Encryption Standard,” National Bureau of Standards, U.S. Department of Commerce, 1977.
 30. J. Daeman, “Cipher and hash function design,” Ph.D. Thesis, Dept. of Elektrotechniek, Katholieke Universiteit Leuven, 1995.



Hung-Min Sun (孫宏民) received his B.S. degree in applied mathematics from National Chung-Hsing University in 1988, his M.S. degree in applied mathematics from National Cheng-Kung University in 1990, and his Ph.D. degree in computer science and information engineering from National Chiao-Tung University in 1995. He was an associate professor in the Department of Information Management, the Chaoyang University of Technology, from 1995 to 1999. Currently, he is teaching in the Department of Computer Science and Information Engineering, National Cheng Kung University. His research interests include cryptography, information theory, network security, reliability, distributed systems.