IET Information Security

Research Article

IET Journals
The Institution of Engineering and Technology

Cryptanalysis of McEliece cryptosystem variants based on quasi-cyclic low-density parity check codes

ISSN 1751-8709
Received on 20th February 2015
Revised on 25th September 2015
Accepted on 6th October 2015
doi: 10.1049/iet-ifs.2015.0064
www.ietdl.org

Masoumeh Koochak Shooshtari[1] ✉, *Mahmoud Ahmadian-Attari*[1], *Thomas Johansson*[2], *Mohammad Reza Aref*[3]

[1]*Faculty of Electrical Engineering, K.N. Toosi University of Technology, Tehran 16315-1355, Iran*
[2]*Department of Electrical and Information Technology, Lund University, Lund 221 00, Sweden*
[3]*Department of Electrical Engineering, Sharif University of Technology, Tehran 11365-11155, Iran*
✉ *E-mail: m-koochak@ee.kntu.ac.ir*

Abstract: One of the approaches to modify the McEliece cryptosystem to overcome its large key size is replacing binary Goppa codes with a new structured code. However, this modification makes such cryptosystems encounter some new attacks. There are a few modified McEliece cryptosystem variants which are known to be secure. One of them is the cryptosystem introduced by Baldi *et al.* which uses quasi-cyclic low-density parity check (QC-LDPC) codes. This cryptosystem is still unbroken as no efficient attack has been reported against it since 2008. In this study, an attack has been applied to this cryptosystem which is feasible when the code length is a multiple of a power of 2. Also an important weakness of this kind of cryptosystem has been pointed out, namely utilising a too low-weight intentional error vector. The authors have established a new security level for this cryptosystem which is applicable to other McEliece-like cryptosystems using QC-LDPC codes. This security level for instance is $2^{9.18}$ times lower than previous ones in the case of $n = 4 \times 4096$ when only one ciphertext is available. The gain of the attack in this study can be increased if more than one ciphertext is available.

1 Introduction

Code-based public key cryptography is a highly studied area of cryptography which relies on hard problems in coding theory. The first cryptosystem in this field which was proposed by McEliece [1] uses binary Goppa codes. Although the McEliece cryptosystem allowed fast encryption and decryption, at the time of the proposal, it did not attract too much attention, mainly because of its large public key size in comparison with the cryptosystems based on number theory, such as RSA [2]. Since it was proven that cryptosystems based on number theory will not tolerate quantum computer attacks [3], researchers have tried to find secure cryptosystems for a post-quantum computer age.

Code-based cryptography is one of the most prominent candidates for post-quantum cryptography among the different options examined by cryptography researchers. Much work has been done to overcome the large public key size by applying new codes instead of the binary Goppa codes in McEliece original cryptosystem. Examples are generalised Reed–Solomon codes [4], Reed Muller codes [5], low-density parity check (LDPC) codes [6], quasi-cyclic (QC) codes [7, 8], convolutional codes [9], QC-LDPC codes [10–16], quasi-dyadic codes [17], non-binary Goppa codes [18, 19] and moderate-density parity check (MDPC) codes [20].

McEliece-like public key cryptosystems are subject to two kinds of attacks, namely decoding attacks and structural attacks. The goal of a structural attack is to find the private key through the public key whereas in a decoding attack, an adversary who knows the ciphertext tries to find the plaintext. The security level of cryptosystems (and the complexity of an attack) is typically measured by the average number of binary operations which are required to break the cryptosystem, called attack work factor. In McEliece-like cryptosystems, the attack with the lowest work factor is typically a decoding attack. Thus, usually this kind of attack determines the security level of the cryptosystems. So far, the original McEliece cryptosystem is regarded to be immune against structural attacks. However, the McElice-like cryptosystems having modified codes are often threatened by such an attack. Much work has been done to investigate their security level and most of them were actually broken. Some of this cryptanalysis work can be found in [21–26].

Among McEliece-like cryptosystems that tries to overcome the large public key size problem, only a few have remained unbroken. Among these cryptosystems, the cryptosystem which uses QC-LDPC codes has the minimum key size and seems to be efficient to use.

Using the LDPC codes family in the McEliece cryptosystem was first discussed in [6] in a disappointing manner as it argued that the sparsity of parity check matrix could not be helpful for reducing the key size from a security viewpoint. However, Baldi *et al.* proposed working with QC-LDPC codes instead of LDPC codes in the McEliece cryptosystem in order to exploit the good features of the LDPC family, such as large error correction capability and fast decoding. To do so, they added an idea to exploit the QC structure in order to reduce the public key size [10]. Then, they altered the structure of their cryptosystem without making any changes in the applied code by replacing the permutation matrix used for obtaining the public key with a general transformation matrix. Therefore, they made the whole cryptosystem immune against decoding attacks using the dual code [11]. Shortly thereafter, this proposal was exposed to a structural attack in [22] due to its sparse transformations. By some modifications, Baldi *et al.* proposed a new version of their cryptosystem which has ever since remained unbroken [12]. Although the cryptosysytems based on LDPC and MDPC codes have some similarities, the approaches introduced in [26] for the cryptanalysis of the earlier version of the cryptosystem based on MDPC codes [27] are not applicable to the attack of cryptosystem variants based on QC-LDPC codes. Actually, the approaches used in [26] are based on knowing both parity check and generator matrices which is not the case for [12].

1.1 Our contribution

In this paper, we focus on cryptosystems using QC-LDPC codes and, in particular, the last unbroken one [12]. The extended version of

[12] in [14] includes an updated security analysis. We present a decoding attack by using special squaring technique which is applicable when the code length is a multiple of a power of 2. Our attack is much more efficient than the previously known methods. Moreover, our approach can be easily applied to other cryptosystems based on irregular codes, such as cryptosystems introduced in [16, 15]. These kinds of cryptosystems have been immune to attacks since their publication, but we demonstrate that they have some flaws in their nature. It should be noted that using a code with odd circulant block length (p) can avoid our attack, while the complexity of encryption and decryption are increased as it is impossible to apply Winograd algorithm to reduce their complexity [14]. This approach is similar to what has been done in [20].

1.2 Outline

The paper is organised as follows. We recall some basic facts about coding theory and QC-LDPC codes in Section 2. Next, an overview of the public key cryptosystems based on QC-LDPC is given in Section 3. Our new attack is introduced in Section 4, and Section 5 summarises and concludes the paper.

2 Preliminaries

This section briefly gives adequate background on coding theory.

Definition 1: A $[n, k]$ binary linear code C of length n and dimension k is a k-dimensional subspace of $\mathbb{F}_2^n$, which can be represented by two matrices; a $k \times n$ generator matrix $\boldsymbol{G}$, such that $C = \{m\boldsymbol{G}, m \in \mathbb{F}_2^k\}$ or by a $(n-k) \times n$ parity check matrix $\boldsymbol{H}$, such that $C = \{c \in \mathbb{F}_2^n, c\boldsymbol{H}^T = 0\}$, where c is a codeword of C.

Definition 2: The Hamming weight of a binary codeword or vector $\boldsymbol{x}$ is the number of non-zero coordinates in the codeword or vector which is denoted by $w_H(x)$. The minimum distance, d, of a linear code C is the smallest Hamming weight of a non-zero codeword in C or the minimum Hamming distance between any two different codewords.

Definition 3: A circulant matrix $\boldsymbol{M}$ over $\mathbb{F}_2[x]$ is a $p \times p$ matrix obtained by cyclically right shifting of the first row as follows

$$\boldsymbol{M} = \begin{bmatrix} m_0 & m_1 & \dots & m_{p-1} \\ m_{p-1} & m_0 & \dots & m_{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ m_1 & m_2 & \dots & m_0 \end{bmatrix}.$$

A circulant matrix is completely described by only its first row. It can be equivalently described as a polynomial $m(x) = \sum_{i=0}^{p-1} m_i x^i \in \mathbb{F}_2[x]/(x^p + 1)$ or simply as a vector $\boldsymbol{m} = [m_0, m_1, \dots, m_{p-1}]$.

Proposition 1: Let C_p be the set of all $p \times p$ circulant matrices over $\mathbb{F}_2[x]$. Then an isomorphism exists between the rings $(\mathrm{C}_p, +, \cdot)$ and $(\mathbb{F}_2[x]/(x^p + 1), +, \cdot)$.

Definition 4: A QC code of length n and dimension k is a linear code with its generator matrix or parity check matrix composed by $k_0 \times n_0$ size-p circulant sub-matrices where $k = k_0 \times p$ and $n = n_0 \times p$.

The main property of QC codes is that each cyclic shift of a codeword by p positions is also a codeword. Let $\boldsymbol{B}$ be a generator matrix or a parity check matrix of a QC code which is composed by size-p circulant matrices $\boldsymbol{B}_{i,j} \in \mathrm{C}_p$ such that

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_{1,1} & \dots & \boldsymbol{B}_{1,n_0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{B}_{k_0,1} & \dots & \boldsymbol{B}_{k_0,n_0} \end{bmatrix}.$$

It is easy to verify that this matrix preserves its QC property by matrix addition and matrix multiplication with another matrix with the QC property. Therefore, we can establish an identification between $\boldsymbol{B}$ and a polynomial $k_0 \times n_0$ matrix $\boldsymbol{B}(x)$ with coefficients in $\mathbb{F}_2[x]/(x^p + 1)$ by mapping each sub-matrix $\boldsymbol{B}_{i,j}$ onto the polynomial $b_{i,j}(x)$ which defines it.

Definition 5: A LDPC code is a linear code which has a sparse parity check matrix. Its error correction capability depends on the sparsity of $\boldsymbol{H}$ and it allows a very low complexity decoding process.

3 McEliece cryptosystem using QC-LDPC codes

Here is a very brief overview of the QC-LDPC version of McEliece as described in [13].

3.1 Public key generation

(i) Choose a t-error-correcting $[n, k, d_v]$ code from the QC-LDPC family with parity check matrix $\boldsymbol{H} \in \mathbb{F}_2^{(n-k)\times n}$ in which $n = n_0 \times p$, $k = k_0 \times p$ and $k_0 = n_0 - 1$, such that

$$\boldsymbol{H} = \left[\boldsymbol{H}_0 | \boldsymbol{H}_1 |, \dots, | \boldsymbol{H}_{n_0-1}\right],$$

where each $\boldsymbol{H}_i$ is a circulant $p \times p$ matrix with weight d_v in each row or column. Moreover, $\boldsymbol{H}_{n_0} - 1$ must be non-singular.

(ii) Obtain the systematic generator matrix $\boldsymbol{G}' \in \mathbb{F}_2^{k\times n}$ of the code, which is $\boldsymbol{G}' = \left[\boldsymbol{I}_{k\times k} | \boldsymbol{P}_{k\times(n-k)}\right]$, where

$$\boldsymbol{P} = \begin{bmatrix} \left(\boldsymbol{H}_{n_0-1}^{-1}\boldsymbol{H}_0\right)^{\mathrm{T}} \\ \left(\boldsymbol{H}_{n_0-1}^{-1}\boldsymbol{H}_1\right)^{\mathrm{T}} \\ \vdots \\ \left(\boldsymbol{H}_{n_0-1}^{-1}\boldsymbol{H}_{n_0-2}\right)^{\mathrm{T}} \end{bmatrix}.$$

(iii) Generate the public key $\boldsymbol{G}$ from $\boldsymbol{G}'$ as

$$\boldsymbol{G} = \boldsymbol{S}^{-1} \cdot \boldsymbol{G}' \cdot \boldsymbol{Q}^{-1},$$

where $\boldsymbol{Q}$ is a sparse $n \times n$ non-singular matrix with row and column weight $m > 1$ and $\boldsymbol{S}$ is a $k \times k$ dense non-singular scrambling matrix. Both $\boldsymbol{Q}$ and $\boldsymbol{S}$ have a QC structure.

3.2 Encryption

Let $\boldsymbol{m} \in \mathbb{F}_2^k$ be the plaintext. Multiply $\boldsymbol{m}$ with the public key $\boldsymbol{G}$ and add t' random errors, that is, $\boldsymbol{y} = \boldsymbol{m}\boldsymbol{G} + \boldsymbol{e}$, where $w_H(\boldsymbol{e}) = t' \leq t/m$.

3.3 Decryption

Let $\boldsymbol{y} \in \mathbb{F}_2^n$ be the ciphertext. Given the secret low-weight parity check matrix $\boldsymbol{H}$ together with matrices $\boldsymbol{Q}$ and $\boldsymbol{S}$, and multiplying $\boldsymbol{y}$ by $\boldsymbol{Q}$ from right, then a low complexity decoding procedure is used to obtain $\boldsymbol{m}\boldsymbol{S}^{-1}$. At last, multiply S from right to find the plaintext $\boldsymbol{m}$. More details can be found in [13].

4 Presenting a new attack

In what follows, our decoding attack on the scheme from Section 3 is described, where we intend to obtain the error vector $\boldsymbol{e}$ from an observed ciphertext $\boldsymbol{y}$. In our new approach, we will use a form of information set decoding (ISD) algorithm, but only after applying a special squaring technique that will be explained later.

One standard way of finding $\boldsymbol{e}$ in a received codeword (or ciphertext) is searching for the minimum weight codewords in the given code extended by the received codeword, that is, the code described by the generator matrix $\begin{bmatrix}\boldsymbol{G}\\ \boldsymbol{y}\end{bmatrix}$. Such an approach, then, uses an ISD algorithm to search for the minimum weight codeword which is equivalent to finding $\boldsymbol{e}$. The complexity of this attack can be determined through the complexity of the Stern algorithm [28] or its improved versions [29–32].

In QC-LDPC code-based cryptosystems, each block-wise cyclically shifted version of the ciphertext $\boldsymbol{y}$ is still a valid ciphertext. This QC structure can help us to speed up the attack procedure by a factor $1/\sqrt{M}$, where M is the number of possible shifted versions of each received codeword [33, 20].

The main idea that needs to be described in our new attack is a special squaring technique which is combined with a reconstructing step for 'partial' polynomials. In the squaring technique, as introduced in [26], we have decreased both the dimension and the weight of each of the polynomials simultaneously. Squaring each term of an arbitrary polynomial in $\mathbb{F}_2[x]/(x^p+1)$, p even, leads to a polynomial having terms of only even degrees. In general, squaring d times, the degree of each term will be a multiple of 2^d if and only if p is a multiple of 2^d. If so, we can omit all positions that are not a multiple of 2^d. In this case, the dimension is decreased by a factor of 2^d. In addition, some collisions between non-zero elements can occur and, thus, the weight of the squared polynomial can be decreased after squaring. In the reconstruction procedure, we aim to reconstruct the original polynomial from its squared versions. The complexity of this technique depends on both its dimension and the number of times squaring is applied to the polynomial.

In our attack, we apply the squaring technique to the ciphertext (received word) $\boldsymbol{y}$ and we hope to find the squared low-weight codeword in a lower dimension faster. Then, we will be able to find the original low-weight codeword from the squared low-weight codewords in the reconstruction procedure.

4.1 Squaring step

An operation referred to as a special squaring is defined for both QC non-square matrices and long vectors over $\mathbb{F}_2[x]/(x^p+1)$, p even. It should be noted that long polynomials must be divided into length p parts before squaring. For notational convenience, let $e(x)$, $y(x)$ and $g(x)$ represent $\boldsymbol{e}$, $\boldsymbol{y}$ and $\boldsymbol{G}$, respectively.

4.1.1 Vector squaring

Let $\boldsymbol{e}$ be a binary vector with a length of $n=n_0p$. The squaring procedure over $\mathbb{F}_2[x]/(x^p+1)$ divides $\boldsymbol{e}$ into n_0 size-p vectors. By considering Proposition 1, we have

$$\boldsymbol{e}=\begin{bmatrix}\boldsymbol{e}_1 & \boldsymbol{e}_2 & \dots & \boldsymbol{e}_{n_0}\end{bmatrix}$$
$$\Updownarrow$$
$$e(x)=\begin{bmatrix}e_1(x) & e_2(x) & \dots & e_{n_0}(x)\end{bmatrix}.$$

Then, each part is squared separately as follows

$$e^2(x)=\begin{bmatrix}e_1^2(x)\bmod(x^p+1) & e_2^2(x)\bmod(x^p+1) & \dots & e_{n_0}^2(x)\bmod(x^p+1)\end{bmatrix}.$$

To complete the special squaring, all of the indices that are not a multiple of 2 in each $e_i^2(x)\bmod(x^p+1)$ part should be omitted so that the dimension of each part and, thus, that of the whole vector, is decreased by a factor of 2. This can be considered as a variable substitution where x^2 is replaced by a new variable x. After that, it can be transformed into a vectorial form and this reduced form of $e^2(x)$ is called $\boldsymbol{e}_{\mathrm{SP}}^2$,

$$\boldsymbol{e}_{\mathrm{SP}}^2=\begin{bmatrix}\boldsymbol{e}_{1,\mathrm{SP}}^2 & \boldsymbol{e}_{2,\mathrm{SP}}^2 \dots & \boldsymbol{e}_{n_0,\mathrm{SP}}^2\end{bmatrix}.$$

4.1.2 Matrix squaring: Let $\boldsymbol{G}\in\mathbb{F}_2^{k\times n}[x]$ be a QC matrix containing $k_0\times n_0$, $p\times p$ circulant sub-matrices of the form

$$\boldsymbol{G}=\begin{bmatrix}\boldsymbol{G}_{1,1} & \dots & \boldsymbol{G}_{1,n_0}\\ \vdots & \ddots & \vdots\\ \boldsymbol{G}_{k_0,1} & \dots & \boldsymbol{G}_{k_0,n_0}\end{bmatrix}.$$

Considering Proposition 1, we can show $\boldsymbol{G}$ in the polynomial matrix form as

$$\boldsymbol{G}(x)=\begin{bmatrix}g_{1,1}(x) & \dots & g_{1,n_0}(x)\\ \vdots & \ddots & \vdots\\ g_{k_0,1}(x) & \dots & g_{k_0,n_0}(x)\end{bmatrix}.$$

Then, the matrix squaring of $\boldsymbol{G}(x)$ is defined as

$$\boldsymbol{G}^2(x)=\begin{bmatrix}g_{1,1}^2(x)\bmod(x^p+1) & \dots & g_{1,n_0}^2(x)\bmod(x^p+1)\\ \vdots & \ddots & \vdots\\ g_{k_0,1}^2(x)\bmod(x^p+1) & \dots & g_{k_0,n_0}^2(x)\bmod(x^p+1)\end{bmatrix}.$$

As for vectors, all of the positions that are not a multiple of 2 in each polynomial $g_{i,j}^2(x)\bmod(x^p+1)$ part should be omitted so the dimension of each part and, thus, that of the whole matrix is decreased by a factor of 2. After that, it can be transformed to QC form; this reduced form of $\boldsymbol{G}^2$ is called $\boldsymbol{G}_{\mathrm{SP}}^2$.

With this definition, we can square all QC matrices including the non-square ones for p even.

Applying squaring q times and omitting positions which are not a multiple of 2^q result in a decrease in the dimension of polynomials by the factor of 2^q and we can obtain $\boldsymbol{e}_{\mathrm{SP}}^{2^q}$ with a length of $n/2^q$ and $\boldsymbol{G}_{\mathrm{SP}}^{2^q}\in\mathbb{F}_2^{k/2^q\times n/2^q}[x]$. Moreover, the polynomial weight is reduced only when $2^q|p$ because more squaring actually has no effect on weight and dimension. For the sake of simplicity, we have omitted SP in the notation, so $\boldsymbol{e}^{2^q}$ and $\boldsymbol{G}^{2^q}$ denote $\boldsymbol{e}_{\mathrm{SP}}^{2^q}$ and $\boldsymbol{G}_{\mathrm{SP}}^{2^q}$ for $q\in N$ in what follows.

4.2 Searching for low-weight codewords

ISD algorithms are a kind of decoding algorithm which can be used to search for low-weight codewords in random linear codes. In case of ISD algorithm, it is noteworthy that an early efficient algorithm is Stern's algorithm [28]. Some improved versions of Stern algorithm have been proposed in [29, 31, 32], but for better comparison with [14], similarly [29] has been considered in our attack. To find a single codeword of weight w in a code of length n and dimension k, the complexity which is denoted as $\mathbf{WF}(n, k, w)$, is calculated as $\mathbf{WF}(n, k, w)=N/P_w$, where the number of binary operations for each iteration is

$$N=\frac{1}{2}(n-k)^2(n+k)+\left(0.5k-g+1+2\binom{k/2}{g}\right)l+\frac{(w-2g+1)g\binom{k/2}{g}^2}{2^{l-2}},$$

IET Inf. Secur., 2016, Vol. 10, Iss. 4, pp. 194–202

and the probability of finding a single codeword of weight w is

$$P_w = \frac{\binom{k/2}{g}^2 \binom{n-k-l}{w-2g}}{\binom{n}{w}}.$$

Here, l and g are two parameters whose size is determined in such a way that **WF** is minimised.

As the code here has QC structure and each cyclic shift of size p in a codeword is also a valid codeword, the number of possible shifts which is equal to $n-k$ is considered in calculating the whole complexity. This feature can reduce the complexity by a factor of $\sqrt{n-k}$ [33, 20]. Thus, the ISD complexity can be calculated as $\mathbf{WF}^{\mathrm{QC}}(n, k, w) = N/\sqrt{n-k}P_w$. Note that as this algorithm is used after the special squaring step is taken, this algorithm is applied to $n/2^q$ and $k/2^q$ parameters.

4.3 Reconstructing the polynomials

In each iteration of the squaring procedure, non-zero elements can collide and become zero elements. This procedure causes loss of information and that information must be reconstructed. In this section, we will show how the reconstruction procedure is performed. Let $e^2(x)$ of Hamming weight w' be the squared polynomial of $e(x)$ of Hamming weight w. If no collisions have occurred in the squaring procedure, then $w = w'$. However, for each collision, the weight reduction is equal to 2, therefore, for α collisions the weight reduction is $w - w' = 2\alpha$.

After the special squaring of $e(x)$, there will be some one-positions and zero-positions in $e^2(x)$. In $e^2(x)$, if the coefficient of x^i is 1, it is called one-position; otherwise, that index is referred to as a zero-position. For each x^i over $\mathbb{F}_2[x]/(x^p+1)$ (including one and zero-positions) in $e^2(x)$, there are two monomials $x^{i/2}$, $x^{(i+p)/2}$ in $e(x)$ which can lead to x^i in $e^2(x)$. If there is a one-position in $e^2(x)$, it means that one of the two monomials, $x^{i/2}$, $x^{(i+p)/2}$, exists in $e(x)$. If we have a zero-position in that index, it means that either both of them or none of them may exist in $e(x)$.

To obtain $e(x)$ from $e^2(x)$, we should focus on one and zero-positions in $e^2(x)$. The solutions to an arbitrary polynomial $e(x)$ when $e^2(x)$ is known can be illustrated as in Fig. 1.

In this example, exactly one of $x^{i_1/2}$, $x^{(i_1+p)/2}$ must exist in $e(x)$ and either none (most likely for sparse polynomials) or both $x^{j_1/2}$, $x^{(j_1+p)/2}$ must exist in $e(x)$.

In case no collisions have occurred in the squaring procedure, then, recovering $e(x)$ is easy as it is sufficient to select between two sets of incidences: $x^{i_1}/2$ and $x^{(i_1}+p)/2$, which is done by solving a system of linear equations that will be explained later. However, in case of the occurrence of any collisions, first, the location of collisions should be recovered. One approach is checking all collisions in each step of the reconstruction and solving the system of linear equations for each of the steps. This easy approach is useful when we have a few collisions. The second approach is recovering the collisions by means of ISD algorithm. Since $e^2(x)$ of weight $w' = w - 2\alpha$ has been recovered, the one-positions show the location of a set of $2w'$ positions which contains, at least, w' elements of w, which is the Hamming weight of $e(x)$. To recover the location of collisions, the extended code (by the received codeword) can be punctured in those positions and the problem is just finding a low-weight codeword of weight 2α in a code of length $n-2w'$ while the dimension remains unchanged, which is equivalent to finding the locations of collisions. When we have several collisions, the second approach is more efficient.

Let us assume that we have a known vector $\boldsymbol{e}^2$ and we want to find $\boldsymbol{e}$ such that $\boldsymbol{e}\boldsymbol{H}^{\mathrm{T}} = \boldsymbol{s}$. As we work in $\mathbb{F}_2[x]/(x^p+1)$, we have

$$\begin{bmatrix} \boldsymbol{e}_1 & \boldsymbol{e}_2 & \dots & \boldsymbol{e}_{n_0} \end{bmatrix} \begin{bmatrix} \boldsymbol{H}_1 & \boldsymbol{H}_2 & \dots & \boldsymbol{H}_{n_0} \end{bmatrix}^{\mathrm{T}} = \boldsymbol{s},$$

or equivalently

$$\boldsymbol{e}_1\boldsymbol{H}_1^{\mathrm{T}} + \boldsymbol{e}_2\boldsymbol{H}_2^{\mathrm{T}} + \cdots + \boldsymbol{e}_{n_0}\boldsymbol{H}_{n_0}^{\mathrm{T}} = \boldsymbol{s}.$$

Due to even dimensions of $\boldsymbol{e}$ and also $\boldsymbol{e}_i$ (p is even), we can write it as $\begin{bmatrix} \boldsymbol{x}_i & \boldsymbol{x}_i + \boldsymbol{a}_i \end{bmatrix}$ where $\boldsymbol{x}_i$ and $\boldsymbol{a}_i$ are binary vectors of size $p/2$: $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \dots, x_{ip/2}]$ and $\boldsymbol{a}_i = [a_{i1}, a_{i2}, \dots, a_{ip/2}]$. $\boldsymbol{x}_i$s are unknown but $\boldsymbol{a}_i$s are known vectors which can be recovered from $\boldsymbol{e}_i^2$. $\boldsymbol{a}_i$ is non-zero in the positions whose corresponding polynomial term is in the one-position of $e_i^2(x)$ and zero in the remaining positions. Moreover, $\boldsymbol{s}$ is divided into two vectors of size $p/2$: $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$. Since each sub-block of $\boldsymbol{H}^{\mathrm{T}}$ has a circulant structure of size $p \times p$, it is possible to show each circulant $\boldsymbol{H}_i^{\mathrm{T}}$ as

$$\boldsymbol{H}_i^{\mathrm{T}} = \begin{bmatrix} \boldsymbol{E}_{1,i} & \boldsymbol{E}_{2,i} \\ \boldsymbol{E}_{2,i} & \boldsymbol{E}_{1,i} \end{bmatrix},$$

where each $\boldsymbol{E}_{j,i}$, $1 \le j \le 2$, $1 \le i \le n_0$ is a binary matrix of size $p/2 \times p/2$, we have

$$\sum_{i=1}^{n_0} \boldsymbol{e}_i\boldsymbol{H}_i^{\mathrm{T}} = \sum_{i=1}^{n_0} \begin{bmatrix} \boldsymbol{x}_i & \boldsymbol{x}_i + \boldsymbol{a}_i \end{bmatrix} \begin{bmatrix} \boldsymbol{E}_{1,i} & \boldsymbol{E}_{2,i} \\ \boldsymbol{E}_{2,i} & \boldsymbol{E}_{1,i} \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \end{bmatrix} = \boldsymbol{s}. \quad (1)$$

We can use half of the equations in (1) as follows

$$\sum_{i=1}^{n_0} \boldsymbol{x}_i(\boldsymbol{E}_{1,i} + \boldsymbol{E}_{2,i}) = \boldsymbol{s}_1 + \sum_{i=1}^{n_0} \boldsymbol{a}_i\boldsymbol{E}_{2,i}.$$

First, let us assume that no collisions have occurred. Let $\mathcal{I}_i$ be a set of all one-position's locations in $\boldsymbol{e}_i^2$ and $\phi_{\mathcal{I}_i(\boldsymbol{x})}$ contain the rows of $\boldsymbol{x}$ with indices in $\mathcal{I}_i$ ($\boldsymbol{x}$ can be vector or matrix). Then, we have a system of linear equations as follows

$$\begin{bmatrix} \phi_{\mathcal{I}_1}(\boldsymbol{x}_1^{\mathrm{T}}) \\ \phi_{\mathcal{I}_2}(\boldsymbol{x}_2^{\mathrm{T}}) \\ \vdots \\ \phi_{\mathcal{I}_{n_0}}(\boldsymbol{x}_{n_0}^{\mathrm{T}}) \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \phi_{\mathcal{I}_1}(\boldsymbol{E}_{1,1} + \boldsymbol{E}_{2,1}) \\ \phi_{\mathcal{I}_2}(\boldsymbol{E}_{1,2} + \boldsymbol{E}_{2,2}) \\ \vdots \\ \phi_{\mathcal{I}_{n_0}}(\boldsymbol{E}_{1,n_0} + \boldsymbol{E}_{2,n_0}) \end{bmatrix} = \boldsymbol{s}_1 + \sum_{i=1}^{n_0} \boldsymbol{a}_i\boldsymbol{E}_{2,i}, \quad (2)$$

where $\boldsymbol{s}_1$, $\boldsymbol{a}_i$s and $\boldsymbol{E}_{j,i}$s are known and constant. By solving (2), we can find $\boldsymbol{x}_i$, $1 \le i \le n_0$.

In the average case, we expect to have a few collisions. Whenever a collision has occurred, it means that two non-zero elements have collided leading to a zero element. In each step of reconstruction, by using ISD algorithm to find a low-weight codeword of weight 2α in the punctured extended code, that is, $[n-2(w-2\alpha), k]$, we are able to find the binary collision vector $\boldsymbol{u} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_{n_0}]$ of size n that contains 2α ones, if α collisions have happened in squaring $\boldsymbol{e}$. After finding $\boldsymbol{u}$, we can extract each sub-vector $\boldsymbol{u}_i = [\boldsymbol{v}_i, \boldsymbol{v}_i]$ which is symmetric over $\mathbb{F}_2[x]/(x^p+1)$. We should consider $\boldsymbol{u}_i$ in addition to each $\boldsymbol{e}_i$ in (1). Then, we can consider the indices of non-zero positions of $\boldsymbol{u}_i$ and move their combinations of corresponding rows in $\boldsymbol{H}_i^{\mathrm{T}}$ for each $\boldsymbol{e}_i$, that is, $\boldsymbol{u}_i\boldsymbol{H}_i^{\mathrm{T}}$, to the right hand side. In this case, the number of unknowns remains constant if we consider each of the collision variable assignments as fixed. As we are interested in solving half of the equations, we can solve

$$\begin{bmatrix} \phi_{\mathcal{I}_1}(\boldsymbol{x}_1^{\mathrm{T}}) \\ \phi_{\mathcal{I}_2}(\boldsymbol{x}_2^{\mathrm{T}}) \\ \vdots \\ \phi_{\mathcal{I}_{n_0}}(\boldsymbol{x}_{n_0}^{\mathrm{T}}) \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \phi_{\mathcal{I}_1}(\boldsymbol{E}_{1,1} + \boldsymbol{E}_{2,1}) \\ \phi_{\mathcal{I}_2}(\boldsymbol{E}_{1,2} + \boldsymbol{E}_{2,2}) \\ \vdots \\ \phi_{\mathcal{I}_{n_0}}(\boldsymbol{E}_{1,n_0} + \boldsymbol{E}_{2,n_0}) \end{bmatrix} = \boldsymbol{s}_1 + \sum_{i=1}^{n_0} \boldsymbol{a}_i\boldsymbol{E}_{2,i} + \sum_{i=1}^{n_0} \boldsymbol{v}_i(\boldsymbol{E}_{1,i} + \boldsymbol{E}_{2,i}), \quad (3)$$

Fig. 1 *Reconstruction of* e(x) *from* $e^2(x)$

in case of the occurrence of collision and find $\boldsymbol{x}_i$, $1 \le i \le n_0$. Since the dimension of this linear equation system is equal to $w_{\mathrm{H}}(\boldsymbol{e}^{2^q}) \times (n-k)/2^q$, it is over-defined and solving it can yield a single solution (of course, if it does have any solution).

4.4 Complexity of reconstruction

The complexity of reconstruction procedure in each step (constructing $\boldsymbol{e}^{2^{q-1}}$ from $\boldsymbol{e}^{2^q}$) is formulated as follows

$$\boldsymbol{C}_{\mathrm{Recons}} \simeq \sum_{\alpha} \mathbf{WF}(n/2^{q-1} - 2w_{\mathrm{H}}(\boldsymbol{e}^{2^q}), k/2^{q-1}, 2\alpha) + ((n-k)/2^q)^3 + \alpha \times (n-k)/2^q,$$

where $\mathbf{WF}(n/2^{q-1} - 2w_{\mathrm{H}}(\boldsymbol{e}^{2^q}), k/2^{q-1}, 2\alpha)$ is the cost of searching for the collision vector of weight 2α when α collisions have occurred in the qth step of squaring. $((n-k)/2^q)^3$ is the cost of Gaussian elimination to solve the linear equations and $\alpha \times (n-k)/2^q$ is the cost of considering collisions.

$\boldsymbol{C}_{\mathrm{Recons}}$ is calculated according to the second approach which was explained in Section 4.3. In fact in this approach, after finding the collision vector, it is needed to solve the system of linear equations (3) only once. However, by following the first approach, the number of collision vector patterns, $N_q(\alpha) = \binom{n_0(n-k)/2^q}{\alpha}$, should be enumerated and the system of linear equations for each one should be solved. In case we have several collisions, this easy approach is not efficient.

It is possible to find the collision vector totally in one step. Since $e^{2^q}(x)$ of weight $w - 2\alpha$ has been recovered, its one-positions show the location of a set of $2^q(w-2\alpha)$ positions which contains, at least, $w - 2\alpha$ elements of w in $\boldsymbol{e}$. To recover the locations of collisions, the extended code (by the received codeword) can be punctured in those positions and the problem is just finding a low-weight codeword of weight 2α in a code with parameters $[n - 2^q(w-2\alpha), k]$. Thus, the complexity of ISD is $\mathbf{WF}(n - 2^q(w-2\alpha), k, 2\alpha)$. After finding the whole collision vector, the number of collisions that have occurred in each step should be determined so that the system of linear equations (3) in each step can be solved.

To handle the complexity of reconstruction, we prefer to find the collision vector by applying ISD algorithm in each step and doing the reconstruction step-by-step.

Now, we can summarise the new attack in the next subsection.

4.5 Attack procedure

(i) Square the public key $\boldsymbol{G}$ and the received ciphertext $\boldsymbol{y}$ for q times with the special squaring technique that has been explained in Section 4.1.

(ii) Apply the ISD algorithm to the matrix $\begin{bmatrix} \boldsymbol{G}^{2^q} \\ \boldsymbol{y}^{2^q} \end{bmatrix}$ and find the minimum weight codeword which will be $\boldsymbol{e}^{2^q}$.

(iii) Calculate $\boldsymbol{H}$ from $\boldsymbol{G}$ in systematic form and the syndrome $\boldsymbol{s} = \boldsymbol{y} \cdot \boldsymbol{H}^{\mathrm{T}}$.

(iv) Find the expected weight of $\boldsymbol{e}^{2^q}$ for different q and its probability of occurrence as the weight of $\boldsymbol{e}$, $w = t'$ is known.

(v) Use $\boldsymbol{e}^{2^q} \cdot \boldsymbol{H}^{2^q} = \boldsymbol{s}^{2^q}$ for reconstructing $\boldsymbol{e}$ by iterating the reconstruction procedure explained in Section 4.3; that is, find $\boldsymbol{e}^{2^{q-1}}$ from the given $\boldsymbol{e}^{2^q}$, for the expected weights while assuming α collisions in each step.

Example 1: Let $h_0(x) = 1 + x^2 + x^3$ and $h_1(x) = x + x^2 + x^7$ in $\mathbb{F}_2[x]/(x^8 + 1)$ be the polynomial generators of $\boldsymbol{H} = [\boldsymbol{H}_0|\boldsymbol{H}_1]$ which corresponds to $\boldsymbol{G} = [\boldsymbol{I}_8|\boldsymbol{P}]$, with $p(x) = x^7$, as public key. Also, let us assume that $y = mG + e = [10000111, 10011101]$ is the ciphertext and we want to find $\boldsymbol{e}$ of weight $w = 3$ by our attack, assuming no collision has occurred.

First, by squaring $\boldsymbol{G}$ and $\boldsymbol{y}$, we have

$$\boldsymbol{G}^2_{\mathrm{SP}} = \left[\begin{array}{c|c} & \begin{matrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{matrix} \\ I_4 & \end{array}\right], \quad \boldsymbol{y}^2_{\mathrm{SP}} = [1111, 1100].$$

Then, by applying ISD algorithm, we can find $\boldsymbol{e}^2_{\mathrm{SP}} = [0100, 0011]$ as the minimum weight codeword. From $\boldsymbol{e}^2_{\mathrm{SP}}$, we can derive three values: (i) $\boldsymbol{e}^2 = [00100000, 00001010]$, (ii) $\boldsymbol{a}_1 = [0100]$ and $\boldsymbol{a}_2 = [0011]$, (iii) $\mathcal{I}_1 = \{2\}$ and $\mathcal{I}_2 = \{3, 4\}$. It is easy to determine two sets of possible values for $\boldsymbol{e}$ from $\boldsymbol{e}^2_1 = x^2$ and $\boldsymbol{e}^2_2 = x^4 + x^6$ by the approach which was depicted in Fig. 1. Two sets are as follows

$$\boldsymbol{e}_{\mathrm{Set1}} = \{x, x^2, x^3\}$$

$$\boldsymbol{e}_{\mathrm{Set2}} = \{x^5, x^6, x^7\}$$

In the next step, $\boldsymbol{H} = [\boldsymbol{P}^{\mathrm{T}}|\boldsymbol{I}_8]$ and the syndrome $\boldsymbol{s} = \boldsymbol{y} \cdot \boldsymbol{H}^{\mathrm{T}} = [\boldsymbol{s}_1, \boldsymbol{s}_2] = [1001, 0010]$ can be calculated. From $\boldsymbol{H}^{\mathrm{T}}$, we derive $\boldsymbol{E}_{1,1}$, $\boldsymbol{E}_{2,1}$, $\boldsymbol{E}_{1,2}$, $\boldsymbol{E}_{2,2}$. Now, the system of linear equations is as follows

$$\boldsymbol{x}_1 \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \boldsymbol{x}_2 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

By applying $\phi_{\mathcal{I}_1}$ and $\phi_{\mathcal{I}_2}$ according to (2), we get the following equation system

$$\begin{bmatrix} \boldsymbol{x}_{11} \\ \boldsymbol{x}_{21} \\ \boldsymbol{x}_{22} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

Having solved the equation system, we get $\boldsymbol{x}_{11} = 1$, $\boldsymbol{x}_{21} = 0$, $\boldsymbol{x}_{22} = 1$. Owing to the structure of $\boldsymbol{e}_i = \begin{bmatrix} \boldsymbol{x}_i & \boldsymbol{x}_i + \boldsymbol{a}_i \end{bmatrix}$, if $x_{ii} = 1$, it means that element of $\boldsymbol{e}$ should be chosen from the first set, that is, $\boldsymbol{e}_{\mathrm{Set1}}$ and if $x_{ii} = 0$, that element is chosen from $\boldsymbol{e}_{\mathrm{Set2}}$. So we can find $e_1(x) = x$ and $e_2(x) = x^3 + x^6$. The error vector is $\boldsymbol{e} = [01000000, 00010010]$ and by $\boldsymbol{e}\boldsymbol{H}^{\mathrm{T}} = \boldsymbol{s}$, the correctness of $\boldsymbol{e}$ is verified.

Example 2: Let us assume that in Example 1, $\boldsymbol{y} = [10100101, 10011101]$ is given as ciphertext and we want to find the error vector $\boldsymbol{e}$ of weight $w = 5$. We should do all the steps, similar to what we have done in Example 1. $\boldsymbol{e}^2_{\mathrm{SP}} = [0100, 0011]$ and $\boldsymbol{s} = [1101, 0110]$ are recovered. As the Hamming weight of $\boldsymbol{e}^2_{\mathrm{SP}}$ is less than $w = 5$ and the difference is 2, it means that one collision has occurred. In this case, we should consider this collision in the reconstruction step.

IET Inf. Secur., 2016, Vol. 10, Iss. 4, pp. 194–202

To recover the collision vector, first, the punctured extended code should be constructed. To do so, we remove two columns corresponding to x and x^5 in the first block and four columns corresponding to x^2, x^3, x^6 and x^7 in the second block of $\begin{bmatrix} \boldsymbol{G} \\ \boldsymbol{y} \end{bmatrix}$. Applying the ISD algorithm, we can find a codeword of weight 2 as the collision vector: [001000100, 00000000] that shows one collision has occurred in the first block, that is, $v_1 = [0010]$. So the system of linear equations according to (3) is as follows

$$\begin{bmatrix} \boldsymbol{x}_{11} \\ \boldsymbol{x}_{21} \\ \boldsymbol{x}_{22} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

It is noteworthy that (by finding the collisions correctly), this equation system is the same as Example 1 with the same solution, so the discovered $\boldsymbol{e}$ is given by

$$\begin{aligned} \boldsymbol{e} &= [01000000, 00010010] + [00100010, 00000000] \\ &= [01100010, 00010010]. \end{aligned}$$

Note that for the sake of simplicity, $\boldsymbol{Q}$ and $\boldsymbol{S}$ have been omitted in the public key of our examples, but nothing is changed in case they are used.

4.6 Cryptanalysis and complexity of attack

The complexity of the attack is the sum of the reconstruction steps and the ISD algorithm complexity. In fact, by selecting the number of special squarings q, a tradeoff between these two complexity numbers can be used. The attack can be handled in such a way that ISD algorithm determines the attack complexity.

***Selecting q*:** q must be selected in such a way that the complexity of the ISD algorithm is reduced by decreasing the dimension while the complexity of the reconstruction is kept within an acceptable range around the highest ISD algorithm complexity. Note that q is upper bounded by the highest power of 2 which divides p, that is, $2^q | p$.

Success probability of the attack: As explained before, we know that by applying the ISD algorithm, we can find the low-weight codeword which is $\boldsymbol{e}^{2^q}$. In the reconstruction step, if all possible search-weights are checked, it can be guaranteed that $\boldsymbol{e}$ can be found, but sometimes, some of the search-weights occur with a very low probability while the ISD or the reconstructing complexity for those weights is expensive. So these rare cases can be omitted and the probability of having the expected search-weights determines the success probability of the attack. The probability of having a certain weight is equivalent to having a certain number of collisions as the occurrence of a collision causes the Hamming weight of the polynomial to be subtracted by two.

Since it is possible to incorporate the success probability into the work factor, we have defined the total complexity C_{Tot} as the product of the inverse of success probability ($N_c = 1/P_{\text{Succ}}$, the expected number of ciphertexts needed) and the complexity of the attack for each ciphertext C, that is, $C_{\text{Tot}} = N_c C$.

Next, we give the complexity of the attack for the two sets of the parameters proposed in [14]

- P_1: $n = 4 \times 4096$, $k = 3 \times 4096$, $t' = 25$, $\mathbf{WF} = 2^{60}$,
- P_2: $n = 4 \times 6144$, $k = 3 \times 6144$, $t' = 38$, $\mathbf{WF} = 2^{85}$.

First, by examining different choices for q, we select and do the squaring for $q = 4$ times for both sets of the parameters. Having done the squaring procedure, we obtain the reduced version of both sets of the parameters

- P_1: $n = 4 \times 256$, $k = 3 \times 256$,
- P_2: $n = 4 \times 384$, $k = 3 \times 384$.

Then, we can search for the low-weight codewords that have a high probability of existence after squaring. As the selected q is small, we expect the occurrence of only few collisions. Therefore, it is enough to search for the low-weight codewords only in $w = 25$, $w = 23$, $w = 21$ and $w = 19$ for P_1 as also $w = 38$, $w = 36$, $w = 34$, $w = 32$ and $w = 30$ in the case of P_2.

In Table 1, the probability of having collisions in each step of special squaring of $\boldsymbol{e}$ is shown. $\boldsymbol{e}$ is constructed by random distribution of t' ones in n_0 blocks of size p. This table can help us to adjust the success probability of the attack.

The reconstruction complexity depends on q and the number of collisions that have occurred in each step of the reconstruction (q). Table 2 shows the complexity of the reconstruction in each step. Note that the number of collisions in $\boldsymbol{e}^{2^2}$, for instance, reveals that in the reconstruction of $\boldsymbol{e}^{2^2}$ to obtain $\boldsymbol{e}^2$, the procedure must be capable of reconstructing the same amount of collisions.

Now, we determine the final complexity for the two proposed instances under consideration while assuming $\boldsymbol{e}^{2^4}$ has weight w.

The complexity of applying the new attack for P_1:

- $w = 25$ ($\alpha = 0$)

  ISD algorithm $\Rightarrow \mathbf{WF}^{\text{QC}} = 2^{54.14}(\ell = 26, g = 3)$
  Reconstruction $\Rightarrow$ As there is no collision, the complexity is about 2^{33}.

- $w = 23$ ($\alpha = 1$)

  ISD algorithm $\Rightarrow \mathbf{WF}^{\text{QC}} = 2^{50.43}(\ell = 26, g = 3)$
  Reconstruction $\Rightarrow$ As we have only one collision, the complexity, at most, is about $2^{40.60}$.

- $w = 21$ ($\alpha = 2$)

  ISD algorithm $\Rightarrow \mathbf{WF}^{\text{QC}} = 2^{46.83}(\ell = 26, g = 3)$
  Reconstruction $\Rightarrow$ As we have two collisions, the complexity depends on the number of collisions that occur in each step. If, at most, one collision occurs in each step, then the complexity is, at most, about $2^{40.61}$ but if both of the collisions occur in one step of squaring, then the complexity depends on that step and can be $2^{32.53}$, $2^{35.71}$, $2^{38.80}$ or $2^{41.84}$.

- $w = 19$ ($\alpha = 3$)

  ISD algorithm $\Rightarrow \mathbf{WF}^{\text{QC}} = 2^{43.35}(\ell = 26, g = 3)$
  Reconstruction $\Rightarrow$ As we have three collisions, the complexity depends on the number of collisions that occur in each step. If, at most, one collision occurs in each step, then the complexity is, at most, about $2^{40.61}$. If, at most, two collisions occur in each step, then the complexity is, at most, about $2^{41.85}$. If all three collisions occur in one step of squaring, then the complexity depends on that step and can be $2^{32.94}$, $2^{35.96}$, $2^{38.97}$ or $2^{41.98}$.

It is obvious that by considering all three collisions that can happen (see Table 1), the success probability of the attack is $P_{\text{Succ}} = 100\%$. The complexity of the attack is about

$$C_{\text{Tot}} = C = 2^{54.14} + 2^{33} + 2^{50.43} + 2^{40.60} + 2^{46.83} + 2^{41.84} + 2^{43.35} + 2^{41.98} \simeq 2^{54.26}.$$

At the cost of the success probability, we can decrease the work factor of the attack. We can only consider the case of two

Table 1 Probability of having collision in each step after the special squaring

collision (α)	e^2		e^{2^2}		e^{2^3}		e^{2^4}		e^{2^5}	
	P_{I}	P_{II}	P_{I}	P_{II}	P_{I}	P_{II}	P_{I}	P_{II}	P_{I}	P_{II}
4	—	—	—	—	—	0.01%	—	0.04%	0.19%	0.07%
3	—	—	—	0.02%	0.05%	0.06%	0.18%	0.79%	1.18%	4.28%
at most 2	—	—	—	99.98%	99.95%	99.93%	99.81%	99.16%	98.62%	94.92%
2	0.02%	0.07%	0.02%	0.48%	0.83%	1.64%	3.16%	6.57%	9.53%	16.93%
at most 1	99.98%	99.93%	99.8%	99.4%	99.12%	98.29%	96.65%	92.59%	89.09%	77.99%
1	3.52%	5.2%	6.84%	10.18%	12.92%	19.19%	22.51%	29.43%	33.89%	38.94%
0	96.46%	94.73%	92.96%	89.32%	78.62%	79.1%	74.14%	63.16%	55.20%	39.05%

collisions, so with the success probability of $P_{\mathrm{Succ}} = 3.16\%$, the work factor and total complexity of the attack will be about $C = 2^{46.87}$ and $C_{\mathrm{Tot}} = 2^{51.85}$, respectively.

The complexity of applying the new attack for P_2:

- $w = 38$ ($\alpha = 0$)

  ISD algorithm $\Rightarrow$ $\mathbf{WF}^{\mathrm{QC}} = 2^{79.50}(l = 27, g = 3)$
  Reconstruction $\Rightarrow$As there is no collision, the complexity is about $2^{34.75}$.

- $w = 36$ ($\alpha = 1$)

  ISD algorithm $\Rightarrow$ $\mathbf{WF}^{\mathrm{QC}} = 2^{75.55}(l = 27, g = 3)$
  Reconstruction $\Rightarrow$ As we have only one collision, the complexity is, at most, about $2^{42.35}$.

- $w = 34$ ($\alpha = 2$)

  ISD algorithm $\Rightarrow$ $\mathbf{WF}^{\mathrm{QC}} = 2^{71.65}(l = 27, g = 3)$
  Reconstruction $\Rightarrow$ As mentioned before, the complexity depends on the number of collisions that occur in each step. If, at most, one collision occurs in each step, then the complexity will, at most, be about $2^{42.36}$ but if both of the collisions occur in one step of squaring, the complexity will depend on that step and can be $2^{34.24}$, $2^{37.44}$, $2^{40.54}$ or $2^{43.59}$.

- $w = 32$ ($\alpha = 3$)

  ISD algorithm $\Rightarrow$ $\mathbf{WF}^{\mathrm{QC}} = 2^{67.79}(l = 27, g = 3)$
  Reconstruction $\Rightarrow$ As we have three collisions, the complexity depends on the number of collisions that occur in each step. If, at most, one collision occurs in each step, then the complexity is, at most, about $2^{42.36}$. If, at most, two collisions occur in each step, then the complexity is, at most, about $2^{43.60}$. If all three collisions occur in one step of squaring, then the complexity depends on that step and can be $2^{34.63}$, $2^{37.69}$, $2^{40.71}$ or $2^{43.72}$.

- $w = 30$ ($\alpha = 4$)

  ISD algorithm $\Rightarrow$ $\mathbf{WF}^{\mathrm{QC}} = 2^{63.98}(l = 27, g = 3)$
  Reconstruction $\Rightarrow$ As we have four collisions, the complexity depends on the number of collisions that occur in each step. If, at most, one collision occurs in each step, then the complexity is, at most, about $2^{42.36}$. If, at most, two collisions occur in each step, then the complexity is, at most, about $2^{43.60}$. If, at most, three collisions occur in each step, then the complexity is, at most, about $2^{43.72}$. If all four collisions occur in one step of squaring, then the complexity depends on that step and can be $2^{36.69}$, $2^{39.60}$, $2^{42.56}$ or $2^{45.53}$.

  It is obvious that by considering four collisions (see Table 1), the success probability of attack is $P_{\mathrm{Succ}} = 99.94\%$. Moreover, the complexity of the attack is about

$$C_{\mathrm{Tot}} \simeq C = 2^{79.50} + 2^{34.75} + 2^{75.55} + 2^{42.35} + 2^{71.65} + 2^{43.59} + 2^{67.79} + 2^{43.72} + 2^{63.98} + 2^{45.53} \simeq 2^{79.60}.$$

Similar to the previous instance, at the cost of the success probability of the attack, we can decrease the work factor of the attack. Assume the case of three collisions, with the success probability of $P_{\mathrm{Succ}} = 0.79\%$, then the work factor and the total complexity of the attack will be about $C = 2^{69.79}$ and $C_{\mathrm{Tot}} = 2^{74.77}$, respectively.

In Table 3, we have listed some instances of parameters with more details. The $\mathbf{WF}^{\mathrm{QC}}$ of QC-LDPC McEliece cryptosystem [14] and QC-MDPC McEliece cryptosystem [27] considering the QC structure [33] are calculated according to the formulas given in Section 4.2. To give the gain of our attack, we compare this work factor with that of ours.

Our attack can be seen as two scenarios; in the first one, the target is to attack one precise ciphertext, but in the other, the goal is to attack only one ciphertext out of many ciphertexts. When P_{Succ} is about 100% ($N_{\mathrm{c}} \simeq 1$), it means that only one ciphertext is needed to attack the cryptosystem which is equivalent to the first scenario. In Table 3, it can be seen that even by one ciphertext, we can significantly decrease the work factor of Baldi cryposystem [14]: $2^{9.18}$ for $n = 4\times$

Table 2 Reconstruction complexity in each step

$w_{\mathrm{H}}(\boldsymbol{e}^{2^q})$ $P_{\mathrm{I}}/P_{\mathrm{II}}$	collision (α)	$e^{2^5}(q=5)$		$e^{2^4}(q=4)$		$e^{2^3}(q=3)$		$e^{2^2}(q=2)$		$e^2(q=1)$	
		P_{I}	P_{II}	P_{I}	P_{II}	P_{I}	P_{II}	P_{I}	P_{II}	P_{I}	P_{II}
25/38	0	2^{21}	$2^{22.75}$	2^{24}	$2^{25.75}$	2^{27}	$2^{28.76}$	2^{30}	$2^{31.76}$	2^{33}	$2^{34.75}$
23/36	1	$2^{27.90}$	$2^{29.63}$	$2^{31.29}$	$2^{33.03}$	$2^{34.47}$	$2^{36.22}$	$2^{37.56}$	$2^{39.31}$	$2^{40.60}$	$2^{42.35}$
21/34	1	$2^{27.97}$	$2^{29.67}$	$2^{31.32}$	$2^{33.05}$	$2^{34.49}$	$2^{36.23}$	$2^{37.57}$	$2^{39.31}$	$2^{40.61}$	$2^{42.36}$
	2	$2^{29.14}$	$2^{30.82}$	$2^{32.53}$	$2^{34.24}$	$2^{35.71}$	$2^{37.44}$	$2^{38.80}$	$2^{40.54}$	$2^{41.84}$	$2^{43.59}$
19/32	1	$2^{28.04}$	$2^{29.72}$	$2^{31.35}$	$2^{33.07}$	$2^{34.50}$	$2^{36.24}$	$2^{37.57}$	$2^{39.32}$	$2^{40.61}$	$2^{42.36}$
	2	$2^{29.22}$	$2^{30.87}$	$2^{32.56}$	$2^{34.27}$	$2^{35.73}$	$2^{37.46}$	$2^{38.81}$	$2^{40.55}$	$2^{41.85}$	$2^{43.60}$
	3	$2^{29.91}$	$2^{31.54}$	$2^{32.94}$	$2^{34.63}$	$2^{35.96}$	$2^{37.69}$	$2^{38.97}$	$2^{40.71}$	$2^{41.98}$	$2^{43.72}$
17/30	1	$2^{28.11}$	$2^{29.76}$	$2^{31.39}$	$2^{33.09}$	$2^{34.52}$	$2^{36.25}$	$2^{37.58}$	$2^{39.32}$	$2^{40.61}$	$2^{42.36}$
	2	$2^{29.30}$	$2^{30.92}$	$2^{32.60}$	$2^{34.29}$	$2^{35.74}$	$2^{37.47}$	$2^{38.82}$	$2^{40.56}$	$2^{41.85}$	$2^{43.60}$
	3	$2^{29.94}$	$2^{31.56}$	$2^{32.96}$	$2^{34.65}$	$2^{35.97}$	$2^{37.69}$	$2^{38.99}$	$2^{40.71}$	$2^{41.98}$	$2^{43.72}$
	4	$2^{32.28}$	$2^{33.89}$	$2^{35.01}$	$2^{36.69}$	$2^{37.89}$	$2^{39.60}$	$2^{40.83}$	$2^{42.56}$	$2^{43.79}$	$2^{45.53}$

IET Inf. Secur., 2016, Vol. 10, Iss. 4, pp. 194–202

Table 3 Attack complexity

Ref.	n	t'/t	$\mathbf{WF}^{QC}$	q	α	P_{Succ}, %	N_c	C	C_{Tot}	Gain
[14]	4×4096	25	$2^{63.44}$	4	0, 1, 2, 3	100	1	$2^{54.26}$	$2^{54.26}$	$2^{9.18}$
				4	1	22.51	4.44	$2^{50.50}$	$2^{52.65}$	$2^{10.79}$
				4	2	3.16	31.65	$2^{46.87}$	$2^{51.85}$	$2^{11.59}$
				5	3	1.18	84.75	$2^{42.83}$	$2^{49.24}$	$2^{14.2}$
				6	4	1.29	77.52	$2^{40.01}$	$2^{46.29}$	$2^{17.15}$
[14]	4×6144	38	$2^{88.38}$	4	0, 1, 2, 3, 4	99.94	1.00	$2^{79.60}$	$2^{79.60}$	$2^{8.78}$
				4	1,2	36.0	2.78	$2^{75.64}$	$2^{77.11}$	$2^{11.27}$
				4	3	0.79	126.58	$2^{67.79}$	$2^{74.77}$	$2^{13.61}$
				5	3	4.28	23.36	$2^{68.43}$	$2^{72.98}$	$2^{15.3}$
				5	4	0.7	142.86	$2^{64.26}$	$2^{71.41}$	$2^{16.97}$
				6	6	0.22	454.55	$2^{58.74}$	$2^{67.57}$	$2^{20.81}$
[14]	4×8192	51	$2^{113.47}$	5	3	9.3	10.75	$2^{94.97}$	$2^{98.39}$	$2^{15.08}$
				5	4	2.31	43.29	$2^{90.66}$	$2^{96.10}$	$2^{17.37}$
				6	7	0.38	263.16	$2^{82.57}$	$2^{90.61}$	$2^{22.86}$
				7	9	0.99	101.01	$2^{82.38}$	$2^{89.04}$	$2^{24.43}$
[27]	2×4800	84	$2^{93.44}$	4	6	17.29	5.78	$2^{82.66}$	$2^{85.19}$	$2^{8.25}$
				4	12	0.16	625	$2^{68.72}$	$2^{78.01}$	$2^{15.43}$
				5	16	0.56	178.57	$2^{64.56}$	$2^{72.04}$	$2^{21.04}$
				6	20	5.94	16.84	$2^{61.18}$	$2^{65.25}$	$2^{28.19}$

4096 and $2^{8.78}$ for $n=4\times6144$. If more than one ciphertext is available for cryptanalysis, we are faced with the second scenario of attack and with $P_{Succ}<100\%$ ($N_c>1$), the gain of our attack can be $2^{17.15}$ for $n=4\times4096$, $2^{20.81}$ for $n=4\times6144$ and $2^{24.43}$ for $n=4\times8192$. We have applied our attack to QC-MDPC McEliece cryptosystem [27], as it is shown in Table 3, the work factor can be decreased by a factor of $2^{28.19}$. In comparison with [14], since on average the number of the intended error is more than [14], the number of collisions increases faster during special squaring, so, it is difficult to have a good gain by one or a few ciphertexts.

5 Conclusion

In this paper, we have proposed an attack on QC-LDPC variant of the McEliece cryptosystem which ever since its publication has been considered to be immune to attacks. In fact, there has been no report of an efficient attack on it since 2008. We have shown that having a low-weight intentional error vector and a highly structured code can be drawbacks in such cryptosystems and matrix $\boldsymbol{Q}$ is not useful for countering our attack. In our attack, by using a special squaring technique which is applicable where the ciphertext length is a multiple of a power of 2, and by extracting the low-weight error vector, we could find low-weight codewords easier than a general ISD attack. This leads to attacks on the proposed parameter choices with a complexity that is considerably lower than that of the previously known variants. Our approach can be applied to other cryptosystems based on irregular QC-LDPC codes such as [16, 15]. We expect that there is still room for further improvements by using a stronger search algorithm for low-weight codewords.

6 Acknowledgments

The authors thank the anonymous reviewers for extremely useful comments that greatly helped to improve the publication. The authors also thank Dr Sogand Noroozizadeh for her careful proofreading. This work was supported in part by the Ministry of Science, Research and Technology of I. R. Iran, Iranian National Science Foundation (INSF) under grant no. 92.32575 and Iran Telecommunications Research Center (ITRC) grant T/500/19241.

7 References

1 McEliece, R.J.: 'A public-key cryptosystem based on algebraic coding theory'. DSN Progress Report, 1978, pp. 114–116
2 Rivest, R.L., Shamir, A., Adleman, L.M.: 'A method for obtaining digital signature and public key cryptosystems', *Commun. ACM*, 1978, **21**, (2), pp. 120–126
3 Shor, P.W.: 'Algorithms for quantum computation: discrete logarithms and factoring'. 35th Annual Symp. on Foundations of Computer Science, November 1994, pp. 124–134
4 Niederreiter, H.: 'Knapsack-type crytosystems and algebraic coding theory', *Probl. Control Inf. Theory*, 1986, **15**, pp. 159–166
5 Sidelnikov, V.: 'A public-key cryptosystem based on binary Reed-Muller codes', *Discrete Math. Appl.*, 1994, **4**, (3), pp. 191–207
6 Monico, C., Rosenthal, J., Shokrollahi, A.: 'Using low density parity check codes in the McEliece cryptosystem'. Proc. of IEEE Int. Symp. on Information Theory (ISIT 2000), Sorrento, Italy, June 2000, p. 215
7 Gaborit, P.: 'Shorter keys for code based cryptography'. Proc. of Int. Workshop on Coding and Cryptography, Springer Verlag, 2005 (LNCS, **6110**), pp. 81–91
8 Berger, T.P., Cayrel, P.-L., Gaborit, P., *et al.*: 'Reducing key length of the McEliece cryptosystem'. Progress in Cryptology – AFRICACRYPT 2009, Springer Verlag, 2009 (LNCS, **5580**), pp. 77–97
9 Löndahl, C., Johansson, T.: 'A new version of McEliece PKC based on convolutional codes'. Information and Communications Security, Springer Verlag, 2012 (LNCS, **7618**), pp. 461–470
10 Baldi, M., Chiaraluce, F., Garello, R., *et al.*: 'Quasi-cyclic low-density parity-check codes in the McEliece cryptosystem'. Proc. of IEEE Int. Conf. on Communications (ICC 2007), Glasgow, Scotland, June 2007, pp. 951–956
11 Baldi, M., Chiaraluce, F.: 'Cryptanalysis of a new instance of McEliece cryptosystem based on QC-LDPC codes'. Proc. of IEEE Int. Symp. on Information Theory (ISIT 2007), Nice, France, June 2007, pp. 2591–2595
12 Baldi, M., Bodrato, M., Chiaraluce, F.: 'A new analysis of the McEliece cryptosystem based on QC-LDPC codes'. Security and Cryptography for Networks, Springer Verlag, 2008 (LNCS, **5229**), pp. 246–262
13 Baldi, M.: 'LDPC codes in the McEliece cryptosystem: attacks and countermeasures'. NATO Science for Peace and Security Series - D: Information and Communication Security, IOS Press, 23, 2009, pp. 160–174
14 Baldi, M., Bianchi, M., Maturo, N., *et al.*: 'Security and complexity of the McEliece cryptosystem based on quasi-cyclic low-density parity-check codes', *IET Inf. Secur.*, 2013, **7**, (3), pp. 212–220
15 Baldi, M., Bianchi, M., Maturo, N., *et al.*: 'Improving the efficiency of the LDPC code-based McEliece cryptosystem through irregular codes'. Proc. of IEEE Symp. on Computers and Communications (ISCC 2013), Split, Croatia, July 2013
16 Koochak Shooshtari, M., Ahmadian, M., Payandeh, A.: 'Improving the security of McEliece-like public key cryptosystem based on LDPC codes'. Proc. of 11th Int. Conf. on Advanced Communication Technology (ICACT 2009), February 2009, pp. 1050–1053
17 Misoczki, R., Barreto, P.S.L.M.: 'Compact McEliece keys from Goppa codes'. Selected Areas in Cryptography, Springer Verlag, 2009 (LNCS, **5867**), pp. 376–392
18 Bernstein, D.J., Lange, T., Peters, C.: 'Wild McEliece'. Selected Areas in Cryptography, Springer Verlag, 2010 (LNCS, **6544**), pp. 143–158
19 Bernstein, D.J., Lange, T., Peters, C.: 'Wild McEliece incognito'. Post-Quantum Cryptography (PQCrypto 2011), Springer Verlag, 2011 (LNCS, **7071**), pp. 244–254
20 Misoczki, R., Tillich, J-P., Sendrier, N., *et al.*: 'MDPC-McEliece: new McEliece variants from moderate density parity-check codes'. Proc. of IEEE Int. Symp. on Information Theory (ISIT 2013), Istanbul, Turkey, July 2013, pp. 2069–2073
21 Sidelnikov, V.M., Shestakov, S.O.: 'On the insecurity of cryptosystems based on generalized Reed–Solomon codes', *Discrete Math. Appl.*, 1992, **2**, (4), pp. 439–444
22 Otmani, A., Tillich, J.-P., Dallot, L.: 'Cryptanalysis of two McEliece cryptosystems based on quasi-cyclic codes', *Math. Comput. Sci.*, 2010, **3**, (2), pp. 129–140

23 Umana, V.G., Leander, G.: 'Practical key recovery attacks on two McEliece variants'. Proc. of Symbolic Computation and Cryptography Conf., Egham, UK, 2010, pp. 27–44
24 Faugère, J.C., Otmani, A., Perret, L., *et al.*: 'Algebraic cryptanalysis of McEliece variants with compact keys'. Eurocrypt 2010, Springer Verlag, 2010 (*LNCS*, **6110**), pp. 279–298
25 Couvreur, A., Otmani, A., Tillich, J.P.: 'Polynomial time attack on wild McEliece over quadratic extensions'. EUROCRYPT 2014, Springer Verlag, 2014 (*LNCS*, **8441**), pp. 17–39
26 Löndahl, C., Johansson, T., Koochak Shooshtari, M., *et al.*: 'A new attack on McEliece public-key cryptosystems using quasi-cyclic codes of even dimension'. Design, Code and Cryptography, Springer Verlag, 2015, doi: 10.1007/s10623-015-0099-x
27 Misoczki, R., Tillich, J.-P., Sendrier, N., *et al.*: 'MDPC-McEliece: new McEliece variants from moderate density parity-check codes'. ePrint Archive 2012/409, 2012
28 Stern, J.: 'A method for finding codewords of small weight'. Coding Theory and Applications, Springer Verlag, 1989 (*LNCS*, **388**), pp. 106–113
29 Bernstein, D.J., Lange, T., Peters, C.: 'Attacking and defending the McEliece cryptosystem'. Post-Quantum Cryptography (PQCrypto 2008), Springer Verlag, 2008 (*LNCS*, **5299**), pp. 31–46
30 Johansson, T., Löndahl, C.: 'An improvement to Stern's algorithm'. Internal Report, 2011. http://lup.lub.lu.se/record/2204753
31 May, A., Meurer, A., Thomae, E.: 'Decoding random linear codes in $\tilde{\mathcal{O}}(2^{0.054n})$'. ASIACRYPT 2011, Springer Verlag, 2011 (*LNCS*, **7073**), pp. 107–124
32 Becker, A., Joux, A., May, A., *et al.*: 'Decoding random binary linear codes in $2^{n/20}$: how $1 + 1 = 0$ improves information set decoding'. EUROCRYPT 2012, Springer Verlag, 2012 (*LNCS*, **7237**), pp. 520–536
33 Sendrier, N.: 'Decoding one out of many'. Post-Quantum Cryptography (PQCrypto 2011), Springer Verlag, 2011 (*LNCS*, **7071**), pp. 51–67