A Forgery Attack on a Code-based Signature Scheme

Ali Babaei
Department of Electrical Engineering
Sharif University of Technology
Tehran, Iran
ali.babaei199@sharif.edu

Taraneh Eghlidos
Electronics Research Institute
Sharif University of Technology
Tehran, Iran
teghlidos@sharif.edu

***Abstract*—With the advent of quantum computers, the security of cryptographic primitives, including digital signature schemes, has been compromised. To deal with this issue, some signature schemes have been introduced to resist against these computers. These schemes are known as post-quantum signature schemes. One group of these schemes is based on the hard problems of coding theory, called code-based cryptographic schemes. Several code-based signature schemes are inspired by the McEliece encryption scheme using three non-singular, parity-check, and permutation matrices as the only components of the private keys, and their product as the public key. In this paper, we focus on the analysis of a class of such signature schemes. For this purpose, we first prove that the linear relationships between the columns of the parity-check/generator matrix appear in the public key matrix, and by exploiting this feature we perform a forgery attack on one of the signature schemes of this class as an evidence. The complexity of this attack is of $\mathcal{O}(n^4)$.**

***Index Terms*—code-based signature, code-based cryptography, post-quantum cryptography, scrambler matrix, parity-check matrix, permutation matrix, generator matrix**

I. Introduction

Over the past three decades, public key encryption has played an important role in our global communication infrastructure. These networks support a large number of applications such as mobile technology, e-commerce, social networking, and cloud computing that are important to our economy and security. In such a world, the ability of individuals, businesses, and governments to communicate securely is of utmost importance. Many critical communication protocols are based on three main cryptographic functions: public key encryption, digital signature, and key exchange mechanism [1]. Currently, these functions are implemented using the Diffie-Hellman key exchange scheme, the RSA encryption scheme, and elliptic curve-based encryption schemes. The security of these schemes depends on the difficulty of certain number theory problems, such as the decomposition of integers or the problem of discrete logarithms in different groups.

In 1994, Peter Shor proved that quantum computers can solve integer factorization and discrete logarithm problems. As a result, the security of all public key encryption schemes based on such problems is compromised [2]. A powerful quantum computer would therefore compromise many forms of modern communication, from key exchange to encryption and digital authentication. Therefore, it is necessary to introduce schemes that are resistant to quantum computers and maintain their security in the era of quantum computers.

With the advent of quantum computers, the security of cryptographic primitives such as encryption schemes, digital signature schemes, and key exchange has been compromised. Therefore, the American National Institute of Standards and Technology (NIST) issued a call for proposals for post-quantum cryptographic schemes in 2016. Following this call, various schemes for encryption and signature have been introduced.

One of the schemes that is of interest in this paper is the McEliece encryption scheme [3]. The McEliece scheme is a code-based encryption scheme that was first introduced by McEliece in 1978 and has withstood various attacks so far. For a long time after McEliece's public key encryption scheme, it was believed that it was not possible to provide a code-based signature scheme, until 2001, when Sendrier et al. [4] introduced the first code-based signature scheme which was later called CFS.

Later on, other code-based signature schemes such as code-based group signatures [5], code-based ring signatures [6], code-based one-time signatures [7], [8], code-based undeniable signatures [9] and code-based full-time signatures [10]–[14] have been introduced.

Some of the above mentioned schemes, for example [13] and [14] use a common McEliece-like pattern in their key generation algorithm. In the McEliece encryption scheme, the public key is the product of three non-singular matrices S, generator matrix G, and permutation matrix P. Each of the matrices S, G, and P is also considered as a private key. In this paper, we show that signatures using this approach are not secure and can be forged.

This paper is organized as follows: in Section 2 we introduce the required preliminaries for the code-based forgery attack. We propose our technique to forge a signature in Section 3. By utilizing the forgery technique we apply it to a code-based signature in Section 4. Section 5 concludes the paper.

II. Preliminaries

In this section, we introduce the notations and definitions used in this paper. In this paper, vectors are shown in bold small letters and matrices in bold capital letters. We denote the binary field by F_2. The generator matrix of a linear code $C(n,k)$ and its corresponding parity-check matrix are denoted by $\boldsymbol{G} \in F_2^{k\times n}$ and $\boldsymbol{H} \in F_2^{(n-k)\times n}$ respectively, where n is the length and k is the dimension of the code. If the received vector $\boldsymbol{r}$ differs from the transmitted codeword $\boldsymbol{c}$, then $\boldsymbol{r} \oplus \boldsymbol{c} = \boldsymbol{e}$ and $\boldsymbol{H} \times \boldsymbol{e}^T = \boldsymbol{s}$, where $\boldsymbol{s}$ is called a syndrome of the error vector $\boldsymbol{c}$.

Definition 1 (Computational Syndrome Decoding (CSD) Problem) [15]: Given a matrix $\boldsymbol{H} \in \mathbb{F}_2^{(n-k)\times n}$ a vector $\mathbf{u} \in \mathbb{F}_2^{n-k}$ and an integer $w > 0$ find $\mathbf{x} \in \mathbb{F}_2^n$ of Hamming weight $\leq w$ such that $\mathbf{H} \cdot \mathbf{x}^T = \mathbf{u}$.

Definition 2 (Decisional Syndrome Decoding (DSD) Problem) [15]: Given a matrix $\mathbf{H} \in \mathbb{F}_2^{(n-k)\times n}$, an integer $w > 0$, a random word $\mathbf{x} \in \mathbb{F}_2^n$ of weight w and a random syndrome s_2 of size $n-k$. The DSD problem is defined as distinguishing between random syndrome $\boldsymbol{s_2}$ and the syndrome $\boldsymbol{s_1} = \mathbf{H} \cdot \mathbf{x}^T$ associated with a small weight vector $\mathbf{x}$.

III. The proposed forgery attack

In this section, we propose an attack against a class of signature schemes inspired by the McEliece encryption scheme [3]. The private keys are composed of an $(n-k) \times (n-k)$ non-singular matrix $\boldsymbol{S}$, an $(n-k) \times n$ parity-check matrix $\boldsymbol{H}$, and an $n \times n$ permutation matrix $\boldsymbol{P}$. The product of these three matrices forms the public key matrix $\boldsymbol{H'} = \boldsymbol{SHP}$.

The attack method is based on the fact that $\boldsymbol{H}$ and $\boldsymbol{H'}$ are equivalent. The existing linear relations between the columns of the parity-check matrix $\boldsymbol{H}$ appear in the $\boldsymbol{H'}$ matrix. By linear relation, we mean the linear independence or linear dependence of the columns of the underlying matrix. Based on this idea, one can generate matrices that appear to be different from the private keys, yet their product is equal to the public key.

We note that if more than one set of private keys corresponds to a public key, the signature scheme is prone to a forgery attack. Therefore, an attacker can exploit this weakness to generate a valid signature. In what follows we analyze a signature scheme that has the same flaw.

A. Constructing fake private keys

As we stated earlier, the main idea of the attack is based on the fact that the linear relations between the columns of an arbitrary matrix $\boldsymbol{H}$ appear in the product $\boldsymbol{SH}$, where $\boldsymbol{S}$ is a non-singular matrix. This means that any two or more columns of $\boldsymbol{H}$ that are linearly independent (or dependent) impose that the corresponding columns of $\boldsymbol{SH}$ have the same relation. We prove this statement by the following theorem.

Theorem 1: Assume that $\boldsymbol{S}$ is a non-singular matrix and $\boldsymbol{H}$ is an arbitrary matrix. Then the linear relations between the columns of the matrix $\boldsymbol{H}$ also appear in the corresponding columns of the matrix $\boldsymbol{SH}$.

Proof. We assume that $\mathbf{h}_i$ and $\mathbf{h}_j$ are two linearly independent columns of $\boldsymbol{H}$. Then,

$$\alpha \boldsymbol{h_i} + \beta \boldsymbol{h_j} = 0 \implies \alpha, \beta = 0. \tag{1}$$

and we define,

$\langle \boldsymbol{a}, \boldsymbol{b} \rangle \triangleq \sum_{i=1}^{n} a_i b_i, \quad a, b \in \mathbb{F}_2^n$

The columns corresponding to $\boldsymbol{h_i}$ and $\boldsymbol{h_j}$ in $\boldsymbol{SH}$ are:

$$\boldsymbol{SH} = \begin{pmatrix} \cdots \langle \boldsymbol{S_1}, \boldsymbol{h_i} \rangle & \cdots & \langle \boldsymbol{S_1}, \boldsymbol{h_j} \rangle \cdots \\ \vdots & \ddots & \vdots \\ \cdots \langle \boldsymbol{S_n}, \boldsymbol{h_i} \rangle & \cdots & \langle \boldsymbol{S_n}, \boldsymbol{h_j} \rangle \cdots \end{pmatrix}$$

where $\boldsymbol{S_i}$ is the i-th row of the matrix $\boldsymbol{S}$.

Now we assume that the two columns i and j are not linear independent in $\boldsymbol{SH}$. For this purpose, we consider a linear combination of these two columns. The sum of the i-th and j-th column can be written in the following form:

$$\begin{pmatrix} s_{11} & \cdots & s_{1n} \\ \vdots & \ddots & \vdots \\ s_{n1} & \cdots & s_{nn} \end{pmatrix} \begin{pmatrix} h_{1i} + h_{1j} \\ \vdots \\ h_{ni} + h_{nj} \end{pmatrix}$$

where h_{ki} and h_{kj} are the k-th entries corresponding to vectors $\boldsymbol{h_i}$ and $\boldsymbol{h_j}$. Given that the vectors $\boldsymbol{h_i}$ and $\boldsymbol{h_j}$ in $\boldsymbol{H}$ are linearly independent, their linear combination cannot be zero unless the sum of the columns in $\boldsymbol{S}$ corresponding to the non-zero elements of $\boldsymbol{h_i} + \boldsymbol{h_j}$ is zero. This means that those columns of $\boldsymbol{S}$ are linearly dependent which contradics the non-singularity of $\boldsymbol{S}$. Therefore, the corresponding columns of $\boldsymbol{SH}$ associated with $\boldsymbol{h_i}$ and $\boldsymbol{h_j}$ must also be linearly independent.

Next we assume that $\boldsymbol{h_i}$ and $\boldsymbol{h_j}$ are linearly dependent columns of $\boldsymbol{H}$. That is, without loss of generality, we assume that $\boldsymbol{h_j}$ is a multiple of $\boldsymbol{h_i}$ which means $\boldsymbol{h_j} = \alpha.\boldsymbol{h_i}, \alpha \in F_2$. Now, we demonstrate that if two columns of $\boldsymbol{H}$ are linearly dependent, the corresponding columns in $\boldsymbol{SH}$ is also linearly dependent. By substituting $\boldsymbol{h_j}$ with $\alpha \boldsymbol{h_i}$, we have:

$$\begin{pmatrix} \cdots \langle \boldsymbol{S_1}, \boldsymbol{h_i} \rangle & \cdots & \langle \boldsymbol{S_1}, \alpha \boldsymbol{h_i} \rangle \cdots \\ \vdots & \ddots & \vdots \\ \cdots \langle \boldsymbol{S_n}, \boldsymbol{h_i} \rangle & \cdots & \langle \boldsymbol{S_n}, \alpha \boldsymbol{h_i} \rangle \cdots \end{pmatrix} =$$

$$\begin{pmatrix} \cdots \langle \boldsymbol{S_1}, \boldsymbol{h_i} \rangle & \cdots & \alpha \langle \boldsymbol{S_1}, \boldsymbol{h_i} \rangle \cdots \\ \vdots & \ddots & \vdots \\ \cdots \langle \boldsymbol{S_n}, \boldsymbol{h_i} \rangle & \cdots & \alpha \langle \boldsymbol{S_n}, \boldsymbol{h_i} \rangle \cdots \end{pmatrix}$$

Therefore, we conclude that each entry in the j-th column of $\boldsymbol{SH}$ is α times the corresponding entry of its i-th column.

As a result, if two columns of $\boldsymbol{H}$ are linearly independent (dependent), the corresponding columns in SH are also linearly independent (dependent). ■

Using Theorem 1, the forger can find fake matrices $\boldsymbol{H_f}$ and $\boldsymbol{S_f}$ such that $\boldsymbol{H'} = \boldsymbol{S_f H_f}$. We describe the method of finding these two matrices below.

First, the forger selects a uniformly random $(n-k) \times (n-k)$ submatrix $\boldsymbol{H_f^1}$ and considers the public key as

$$\boldsymbol{H'} = [\boldsymbol{H'_1} \mid \boldsymbol{H'_2}],$$

where $\boldsymbol{H'_1}$ is the $(n-k) \times (n-k)$ submatrix which is the $(n-k)$ linearly independent columns of $\boldsymbol{H'}$, and $\boldsymbol{H'_2}$ is the $(n-k) \times k$ submatrix, which is obtained by the remaining k

columns of $\boldsymbol{H'}$. The forger can obtain a fake matrix $\boldsymbol{S_f}$ by solving the system of equations $\boldsymbol{S_f} \times \boldsymbol{H_f^1} = \boldsymbol{H_1'}$, consisting of $(n-k)^2$ equations and $(n-k)^2$ variables. Then the remaining k columns of H_f are obtained by solving the system of equations

$$\boldsymbol{S_f} \times \boldsymbol{H_2^f} = \boldsymbol{H_2'}$$

including $(n-k) \times k$ equations and $(n-k) \times k$ variables.

From the following relations, we conclude that $\boldsymbol{S_f}$ has a non-zero determinant.

$$\boldsymbol{S H_f^1} = \boldsymbol{H_1'}$$

$$\det(\boldsymbol{H_f^1}) \neq 0$$

$$\det(\boldsymbol{H_1'}) \neq 0$$

Finally, we can use $\boldsymbol{S_f}$ to obtain the remaining k columns of $\boldsymbol{H_f}$.

Theorem 2: The linear relations between the columns of $\boldsymbol{SH} = \boldsymbol{H'}$ appear between the columns of $\boldsymbol{H_f}$.

Proof. Suppose that the columns i and j of $\boldsymbol{H'}$ are linearly dependent. In this case, the i-th and j-th column of the matrix $\boldsymbol{H_f}$ are

$$\boldsymbol{h_i^f} = \boldsymbol{S_f^{-1}} \times \boldsymbol{h_i'}$$

$$\boldsymbol{h_j^f} = \boldsymbol{S_f^{-1}} \times \boldsymbol{h_j'}$$

where $\boldsymbol{h_i^f}$ and $\boldsymbol{h_i'}$ denote the i-th columns of the matrices $\boldsymbol{H_f}$ and $\boldsymbol{H'}$, respectively. The same statement is true for the j-th columns of $\boldsymbol{H_f}$ and $\boldsymbol{H'}$.

It follows from the above that the linear dependency (independency) of each two columns of $\boldsymbol{H_f}$ is inherited from that of the corresponding columns of $\boldsymbol{H'}$. ■

An easier way to obtain $\boldsymbol{H_f}$ is to consider it as the reduced row echelon form of $\boldsymbol{H'}$, because of the fact that the linear relations between the columns of $\boldsymbol{H'}$ appear in the reduced row echelon form of $\boldsymbol{H'}$.

IV. Forgery attack on a code-based signature

In this section, we first introduce a code-based signature scheme by Haidary et.al [14]. Then we use our technique mentioned earlier to forge the signature scheme.

A. The signature scheme

The signature scheme consists of three algorithms: key generation algorithm, signature generation, and verification.

Key Generation Algorithm. This algorithm consists of the following matrices:

- A $k \times n$ generator matrix $\boldsymbol{G}$.
- An $(n-k) \times n$ parity check matrix $\boldsymbol{H}$.
- An $n \times (n-k)$ dual matrix $\boldsymbol{A}$.
- A $k \times k$ scrambler matrix $\boldsymbol{S}$.
- An $n \times n$ permutation matrix $\boldsymbol{P}$.
- An $(n-k) \times (n-k)$ non-singular matrix $\boldsymbol{L}$.

The key generation is performed as follows:

- Compute $\boldsymbol{P'} = (\boldsymbol{H H^T})^{-1}$.
- Compute $\boldsymbol{A} = \boldsymbol{H}^T \boldsymbol{P'}$.
- Like the McEliece cryptosystem we have,

$$\boldsymbol{pk_1} = \boldsymbol{G'} = \boldsymbol{SGP}$$

- Compute $\boldsymbol{pk_2} = \boldsymbol{L^{-1} H P}$.
- For verification, we need $\boldsymbol{pk_3} = \boldsymbol{P^{-1} A H P}$.
- Compute the parity-check matrix $\boldsymbol{H'}$: $\boldsymbol{Q} = \boldsymbol{H'}^T = ((\boldsymbol{AL})^T (\boldsymbol{P}^{-1})^T)^T$

The resulting public and private keys are:

$$\text{pk} = (\boldsymbol{pk}_1, \boldsymbol{pk}_2, \boldsymbol{pk}_3) \quad and \quad \text{pr} = (\boldsymbol{S^{-1}}, \boldsymbol{P^{-1}}, \boldsymbol{G}, \boldsymbol{Q})$$

Furthermore, the following relations hold:

$$\boldsymbol{pk_1}.\boldsymbol{pk_3} = \boldsymbol{0}, \tag{2}$$

$$\boldsymbol{pk_2}.\boldsymbol{pk_3} = \boldsymbol{pk_2}, \tag{3}$$

$$\boldsymbol{pk_3}.\boldsymbol{pk_3} = \boldsymbol{pk_3} \tag{4}$$

Signature generation algorithm.

- Hash a document $\boldsymbol{doc}$, and get the result as n bits $h(\boldsymbol{doc})$, and apply the hash function again on it $h(h(\boldsymbol{doc})) \leftarrow \text{hash}(h(\boldsymbol{doc}))$.
- $\boldsymbol{s}$ is the $n-k$ bit vector $\boldsymbol{s} \leftarrow h(\boldsymbol{doc}) \cdot \boldsymbol{Q}$.
- Obtain:

$$(\boldsymbol{sig})\boldsymbol{SGP} \leftarrow h(\boldsymbol{doc}) + s \cdot \boldsymbol{pk}_2$$

- 4) Apply the decoding function on $\boldsymbol{c}$ to obtain $\boldsymbol{sig}$ and get the result named $\boldsymbol{sig}$,

$$(\boldsymbol{sig})\boldsymbol{SG} \leftarrow ((\boldsymbol{sig})\boldsymbol{SGP})(\boldsymbol{P^{-1}})$$

$$(\boldsymbol{sig})\boldsymbol{S} \leftarrow \boldsymbol{decode}((\boldsymbol{sig})\boldsymbol{SG})$$

$$\boldsymbol{sig} \leftarrow ((\boldsymbol{sig})\boldsymbol{S})(\boldsymbol{S^{-1}})$$

- Compute the vector $\boldsymbol{d}$:

$$\boldsymbol{d} \leftarrow h(h(\boldsymbol{doc}))(\boldsymbol{Q}) + \boldsymbol{s}$$

- The resulting signature is $(\boldsymbol{sig}, \boldsymbol{d})$.

Verification algorithm.

- Hash the received document to compute $h(\boldsymbol{doc})$ and $h(h(\boldsymbol{doc}))$ and compute

$$a \leftarrow (\boldsymbol{sig})\boldsymbol{SGP} \tag{5}$$

- Compute

$$\boldsymbol{v_1} = \boldsymbol{s}(\boldsymbol{pk_2})$$

$$\boldsymbol{d} = h(h(\boldsymbol{doc}))(\boldsymbol{Q}) + \boldsymbol{s}$$

$$\boldsymbol{d}(\boldsymbol{pk_2}) = (h(h(\boldsymbol{doc}))(\boldsymbol{Q}) + \boldsymbol{s})(\boldsymbol{pk_2})$$

$$\boldsymbol{d}(\boldsymbol{pk_2}) = h(h(\boldsymbol{doc}))(\boldsymbol{Q})(\boldsymbol{pk_2}) + \boldsymbol{s}(\boldsymbol{pk_2})$$

Therefore:

$$\boldsymbol{v_1} = \boldsymbol{s}(\boldsymbol{pk_2}) = h(h(\boldsymbol{doc}))(\boldsymbol{pk_3}) + \boldsymbol{d}(\boldsymbol{pk_2})$$

- Compute

$$\boldsymbol{v_2} = \boldsymbol{s}(\boldsymbol{pk_2})$$

$$(\boldsymbol{sig})\boldsymbol{SGP} = h(\boldsymbol{doc}) + \boldsymbol{s}(\boldsymbol{pk_2})$$

Using the public key and the relations (3) we have

$$\boldsymbol{s}(\boldsymbol{pk}_2) = (\boldsymbol{sig})(\boldsymbol{pk}_1) + h(\boldsymbol{doc})$$
$$\boldsymbol{s}(\boldsymbol{pk_2})(\boldsymbol{pk_3}) = (\boldsymbol{sig})(\boldsymbol{pk_1})(\boldsymbol{pk_3}) + h(\boldsymbol{doc})(\boldsymbol{pk_3})$$

therefore,

$$\boldsymbol{v_2} = \boldsymbol{s}(\boldsymbol{pk_2}) = h(\boldsymbol{doc})(\boldsymbol{pk_3})$$

- Check if the following relation holds:

$$\boldsymbol{v_1} = \boldsymbol{v_2}$$

- Compute:

$$\boldsymbol{c} \leftarrow h(\boldsymbol{doc}) + \boldsymbol{s}(\boldsymbol{pk_2}) \tag{6}$$

- Using the relations (5) and (6), check if the verification is successful,

$$\boldsymbol{a} = \boldsymbol{c}$$

B. Forgery attack

The proposed forgery attack consists of two steps. First we obtain the secret key $\boldsymbol{Q}$ and in the second step we compute the fake private keys whose product equals the public key. In this way we can generate a signature which can be validated by the verifier. Let us assume that $\boldsymbol{Q}$, $\boldsymbol{pk_1}$ and $\boldsymbol{pk_2}$ are shown as follows:

$$\boldsymbol{Q} = \begin{pmatrix} q_{11} & \cdots & q_{1(n-k)} \\ \vdots & \ddots & \vdots \\ q_{n1} & \cdots & q_{n(n-k)} \end{pmatrix}$$

$$\boldsymbol{pk_1} = \begin{pmatrix} p^1_{11} & \cdots & p^1_{1n} \\ \vdots & \ddots & \vdots \\ p^1_{k1} & \cdots & p^1_{kn} \end{pmatrix}$$

$$\boldsymbol{pk_2} = \begin{pmatrix} p^2_{11} & \cdots & p^2_{1n} \\ \vdots & \ddots & \vdots \\ p^2_{(n-k)1} & \cdots & p^2_{(n-k)n} \end{pmatrix}$$

As it is shown in [14], the following equations hold between the public keys and the private key $\boldsymbol{Q}$:

$$\boldsymbol{pk_1 Q} = \boldsymbol{0} \tag{7}$$
$$\boldsymbol{pk_2 Q} = \boldsymbol{I} \tag{8}$$
$$\boldsymbol{pk_3 Q} = \boldsymbol{Q} \tag{9}$$
$$\boldsymbol{Q pk_2} = \boldsymbol{pk_3} \tag{10}$$

Therefore, $\boldsymbol{Q}$ can be obtained by solving $(n-k)$ systems of linear equations from (7) and (8). From (7) we have:

$$\boldsymbol{pk_1 Q} = \begin{pmatrix} p^1_{11} & \cdots & p^1_{1n} \\ \vdots & \ddots & \vdots \\ p^1_{k1} & \cdots & p^1_{kn} \end{pmatrix} \begin{pmatrix} q_{11} & \cdots & q_{1(n-k)} \\ \vdots & \ddots & \vdots \\ q_{n1} & \cdots & q_{n(n-k)} \end{pmatrix} = \boldsymbol{0}$$

Therefore, for each column i of the matrix $\boldsymbol{Q}$ we have the following system of k linear equations in n variables,

$$\begin{cases} p^1_{11} q_{1i} + p^1_{12} q_{2i} + \cdots + p^1_{1n} q_{ni} = 0 \\ \vdots \\ p^1_{k1} q_{1i} + p^1_{k2} q_{2i} + \cdots + p^1_{kn} q_{ni} = 0 \end{cases}$$

Here we need additional $(n-k)$ linear equations in n variables to obtain the i-th column of the matrix $\boldsymbol{Q}$. For this purpose, we use the relation (8) as follows:

$\boldsymbol{pk_2 Q} = \boldsymbol{I}$

$$\begin{pmatrix} p^2_{11} & \cdots & p^2_{1n} \\ \vdots & \ddots & \vdots \\ p^2_{(n-k)1} & \cdots & p^2_{(n-k)n} \end{pmatrix} \begin{pmatrix} q_{11} & \cdots & q_{1(n-k)} \\ \vdots & \ddots & \vdots \\ q_{n1} & \cdots & q_{n(n-k)} \end{pmatrix} = \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{pmatrix}$$

For the i-th column of Q we have the following system of $n-k$ linear equations in n variables:

$$\begin{cases} p^2_{11} q_{1i} + p^2_{12} q_{2i} + \cdots + p^2_{1n} q_{ni} = 0 \\ \vdots \\ p^2_{i1} q_{1i} + p^2_{i2} q_{2i} + \cdots + p^2_{in} q_{ni} = 1 \\ \vdots \\ p^2_{(n-k)1} q_{1i} + p^2_{(n-k)2} q_{2i} + \cdots + p^2_{(n-k)n} q_{ni} = 0 \end{cases}$$

Since $\boldsymbol{pk_1} = \boldsymbol{SGP}$ and $\boldsymbol{pk_2} = \boldsymbol{L^{-1}HP}$ are equivalent to $\boldsymbol{GP}$ and $\boldsymbol{HP}$ respectively, the rows of $\boldsymbol{pk_1}$ are linearly independent from the rows of $\boldsymbol{pk_2}$ ($\boldsymbol{G}$ and $\boldsymbol{H}$ are dual matrices).

Thus, by repeating these operations for each column of $\boldsymbol{Q}$, we have to solve a system of n linear equations in n variables which have a unique solution for $\boldsymbol{Q}$ because the matrix Q satisfies both equations (7) and (8).

Using the fourth component of the private key, $\boldsymbol{Q}$, the adversary can successfully recover the second component, $\boldsymbol{d}$, of the signature. Next, we can easily proceed to forge the first component of the signature, $\boldsymbol{sig}$, according to section 3.1. For this purpose, we have to compute the fake private keys,

$$\boldsymbol{pk_1} = \boldsymbol{S_f G_f P_1}$$

It is clear that by permutating the columns of the generator matrix of a code, we get an equivalent code [16]. Without loss of generality, we can consider $\boldsymbol{P_f}$ as an identity matrix.

$$\boldsymbol{P_f} = \boldsymbol{I}$$

The fake private key $\boldsymbol{G_f}$ can easily be obtained by the reduced row echelon form of $\boldsymbol{pk_1}$:

$$\boldsymbol{G_f} = rref(\boldsymbol{pk_1})$$

And finally by solving $k \times n$ systems of linear equations, $\boldsymbol{pk_1} = \boldsymbol{S_f G_f}$, the unknown matrix $\boldsymbol{S_f}$ can be obtained.

At this point, we have fake private keys, $\boldsymbol{S_f}$, $\boldsymbol{G_f}$, $\boldsymbol{P_f}$, and the recovered matrix $\boldsymbol{Q}$. Therefore, any forger can forge the first component of the signature, $\boldsymbol{sig}$, using the fake private keys and can obtain the second component of the signature, $\boldsymbol{d}$, using the private key $\boldsymbol{Q}$.

V. Conclusion

In this paper, we have analyzed a code-based signature scheme inspired by the McEliece cryptosystem in terms of forgery attacks. To the best of our knowledge, this is the first forgery attack on McEliece-like signature schemes that we are aware of. We have shown that these kinds of signature schemes are vulnerable to forgery attacks. In this way, an adversary can forge the private keys to generate a signature, which can be validated by the verifier. This attack can be applied to any signature scheme of the same structure. For this purpose, the forger must obtain fake private keys to generate a valid signature. In this manner, the signature is verified by the verifier, because the verification is not only a function of the signature but also a function of the signer's public key. At the same time, it cannot be checked if the signature is generated by the genuine keys or the fake ones. It is worth mentioning that the complexity of this attack is $O(n^4)$, where n is the code length.

References

[1] Chen, Lily, Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray A. Perlner, and Daniel Smith-Tone. Report on post-quantum cryptography. Vol. 12. Gaithersburg, MD, USA: US Department of Commerce, National Institute of Standards and Technology, 2016.

[2] Shor, Peter W. "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer." SIAM review 41, no. 2 (1999): 303-332.

[3] McEliece, Robert J. "A public-key cryptosystem based on algebraic." Coding Thv 4244 (1978): 114-116.

[4] Courtois, Nicolas T., Matthieu Finiasz, and Nicolas Sendrier. "How to achieve a McEliece-based digital signature scheme." In Advances in Cryptology—ASIACRYPT 2001: 7th International Conference on the Theory and Application of Cryptology and Information Security Gold Coast, Australia, December 9–13, 2001 Proceedings 7, pp. 157-174. Springer Berlin Heidelberg, 2001.

[5] Alamélou, Quentin, Olivier Blazy, Stéphane Cauchie, and Philippe Gaborit. "A code-based group signature scheme." Designs, Codes and Cryptography 82 (2017): 469-493.

[6] Melchor, Carlos Aguilar, Pierre-Louis Cayrel, Philippe Gaborit, and Fabien Laguillaumie. "A new efficient threshold ring signature scheme based on coding theory." IEEE Transactions on Information Theory 57, no. 7 (2011): 4833-4842.

[7] Gaborit, Philippe, and Julien Schrek. "Efficient code-based one-time signature from automorphism groups with syndrome compatibility." In 2012 IEEE International Symposium on Information Theory Proceedings, pp. 1982-1986. IEEE, 2012.

[8] Persichetti, Edoardo. "Efficient one-time signatures from quasi-cyclic codes: A full treatment." Cryptography 2, no. 4 (2018): 30.

[9] Aguilar-Melchor, Carlos, Slim Bettaieb, Philippe Gaborit, and Julien Schrek. "A code-based undeniable signature scheme." In Cryptography and Coding: 14th IMA International Conference, IMACC 2013, Oxford, UK, December 17-19, 2013. Proceedings 14, pp. 99-119. Springer Berlin Heidelberg, 2013.

[10] Debris-Alazard, Thomas, Nicolas Sendrier, and Jean-Pierre Tillich. "Wave: A new code-based signature scheme." (2018).

[11] Biasse, Jean-François, Giacomo Micheli, Edoardo Persichetti, and Paolo Santini. "LESS is more: code-based signatures without syndromes." In Progress in Cryptology-AFRICACRYPT 2020: 12th International Conference on Cryptology in Africa, Cairo, Egypt, July 20–22, 2020, Proceedings 12, pp. 45-65. Springer International Publishing, 2020.

[12] Song, Yongcheng, Xinyi Huang, Yi Mu, Wei Wu, and Huaxiong Wang. "A code-based signature scheme from the Lyubashevsky framework." Theoretical Computer Science 835 (2020): 15-30.

[13] Haidary Makoui, Farshid, Thomas Aaron Gulliver, and Mohammad Dakhilalian. "A new code-based digital signature based on the McEliece cryptosystem." IET Communications 17, no. 10 (2023): 1199-1207.

[14] Haidary Makoui, F., Gulliver, T. A., Dakhilalian, M. (2023). 'Post Quantum Digital Signature Based on the McEliece Cryptosystems with Dual Inverse Matrix', The ISC International Journal of Information Security, 15(3), pp. 101-108. doi: 10.22042/isecure.2023.419559.1026

[15] Blazy, Olivier, Philippe Gaborit, Julien Schrek, and Nicolas Sendrier. "A code-based blind signature." In 2017 IEEE International Symposium on Information Theory (ISIT), pp. 2718-2722. IEEE, 2017.

[16] Hill, Raymond. A first course in coding theory. Oxford University Press, 1986.