Fast and Efficient Code-Based Digital Signature with Dual Inverse Matrix

Farshid, Haidary Makoui[1], T. Aaron, Gulliver[1], and Mohammad Dakhilalian[2]

[1]*Department of Electrical and Computer Engineering, University of Victoria, Victoria, B.C., Canada. email: makoui@uvic.ca, agullive@ece.uvic.ca*

[2]*Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran. email: mdalian@iut.ac.ir*

Abstract

Digital signatures ensure legitimate access through identity authentication. It is also used to build blocks in blockchains and to authenticate transactions. The Courtois-Finiasz-Sendrier (CFS) digital signature is a well-known code-based digital signature scheme based on the Niederreiter cryptosystem. The CFS signature, however, is not widely used due to the long processing time required by its signing algorithm. Most code-based digital signature schemes are based on Niederreiter. The present paper proposes a new code-based digital signature based on the McEliece cryptosystem. The proposed McEliece code-based scheme also gives less complexity and a higher success rate. The scheme provides an efficient code-based algorithm to sign a document in a shorter processing time. The scheme is also secure against public key structural attacks. Key generation, signing, and verification algorithms are presented. The key generation algorithm constructs three-tuple public keys using a dual inverse matrix. The proposed signing scheme is the first code-based digital signature based on McEliece with the lower processing time required to construct a valid digital signature. The proposed signing algorithm also constructs smaller signatures. In addition, the verification algorithm checks the integrity value to avoid any forgery before final verification.

1 Introduction

The first code-based cryptosystem was introduced by McEliece [1] and the second is the Niederreiter cryptosystem [2], which is mainly used in code-based digital signatures [3]. The Shor algorithm indicates that quantum attacks are a serious threat to cryptographic primitives [4]. Code-based cryptographic primitives [5] have been shown to be resistant to quantum attacks. Recent technological developments have intensified this threat and therefore, the National Institute of Standards and Technology has considered various proposals in the post-quantum era. Post-quantum cryptography [6] is the development of cryptographic mechanisms [7–9] to secure systems against quantum attacks. The security of the cryptosystem is based on the hardness of the decoding and code distinguishability problems [10, 11]. The inability to distinguish between the scrambled parity check matrix and other random ones is an NP-problem [11, 12], so is decoding a linear code without the knowledge of its algebraic structure [13].

The existing Niederreiter digital signatures are not widely used due to their long signing process time. The drawback of code-based cryptosystems not used in digital signatures is that the number of valid codewords is smaller than the vector space [14]. Thus the signed document may not necessarily be decodable [15]. This increases the signing process time as the algorithm tries to find a valid signature that the receivers can verify.

This paper proposes a McEliece-based digital signature scheme that covers the entire vector space so that a valid code-based digital signature can be generated with a higher success rate and lower processing time without searching the entire vector space for a decodable syndrome, like the CFS signature. The proposed algorithm is a code-based digital signature scheme with key generation, signing, and verification algorithms. The key generation algorithm constructs a public key and a private key. These keys are used by signing and verification algorithms to sign a chosen document that can successfully be verified on the receiver's side with forgery detection capability.

The main obstacle preventing code-based signatures from being widely used is that the ciphertexts do not cover the entire vector space [14]. As a consequence, on average it takes $t!$ executions of the CFS code-based signatuer to find a valid signature [3]. This paper propose a digital signature scheme that covers the entire vector space, so a valid code-based digital signature can be generated with a higher success rate and lower processing time.

2 Review of the Linear Block Code

This section briefly deals with linear block codes. In modern communication systems, redundant bits added to an information sequence are considered able to detect and correct errors introduced by a noisy channel. In a communication system, the channel encoder assigns a binary sequence codeword $\boldsymbol{c} = (c_1, c_2, ..., c_n)$ to a message $\boldsymbol{m} = (m_1, m_2, ..., m_k)$. For a k-tuple message $\boldsymbol{m}$, there would be 2^k distinct messages and codewords. The set of all 2^k codewords is referred to as a $C(n,k)$ block code. The length of a $C(n,k)$ block code is shown by n and k denoting dimension where $k \leq n$.

The channel encoder adds redundancy in the binary information sequence to the transmitted codewords, so each codeword has $n-k$ redundant bits more than the message associated with it. These redundant bits are used by the channel decoder at the receiver's end to detect and correct errors having occurred over a noisy channel.

A $C(n,k)$ code is linear when its codewords form a k-dimensional vector subspace of the n-tuple vector space. Therefore, there are k linearly independent codewords $\boldsymbol{g}_1, \boldsymbol{g}_2, ..., \boldsymbol{g}_k$ that are settled as the rows of the generator matrix. The systematic form of generator matrix G in linear code is given by

$$G_{k\times n} = (I_k | P_{k\times(n-k)}). \tag{1}$$

Linear algebra shows that for any $C(n,k)$ block code there is a dual code represented by $C^{\perp}$, which is an $n-k$ dimensional vector space. Matrices G and H are the generator matrix and its dual space of the $C(n,k)$ block code. Matrix H, also called the parity check matrix, is an $(n-k)\times n$ matrix such that $GH^T = \mathbf{0}$ where H^T denotes the transposition matrix of H. A systematic parity check matrix has the form

$$H_{(n-k)\times n} = (P^T_{(n-k)\times k} | I_{n-k}). \tag{2}$$

2.1 The McEliece Cryptosystem

In the McEliece cryptosystem, bits of plaintext are scrambled, the corresponding codeword is permuted, and up to t bits are flipped where t denotes the error correcting capability of the code. This is a public key cryptosystem where the public key is the product of a non-singular $k\times k$ scrambling matrix, a $k\times n$ generator matrix of the code, and an $n\times n$ permutation matrix. The secret key consists of three matrices. The encryption and decryption algorithms of the system are given below.

The systematic form of the generator matrix G and the parity check matrix in linear code is given by $G_{k\times n} = (I_k|P'_{k\times(n-k)})$ and $H_{(n-k)\times n} = (P'^T|I_{n-k})$, where I_k is the $k\times k$ identity matrix and P' is an $k\times(n-k)$ matrix.

In the McEliece cryptosystem a code $C(n,k)$ is chosen with a generator matrix G, a scrambling matrix S, and a permutation matrix P. The public key is $pk = SGP$ and the secret key is $sk = (S,G,P)$. The encryption algorithm of the McEliece cryptosystem is as follows.

1. For a plaintext $\boldsymbol{m}$ of length k, Alice uses Bob's public key to encode it via $\boldsymbol{c} = \boldsymbol{m}SGP$.

2. Next, she flips some of the bits of $\boldsymbol{c}$ by selecting a random vector $\boldsymbol{e}$ of the length n so that $w(\boldsymbol{e}) \leq t$, where t is the error correcting capability of the code. The ciphertext is

$$\boldsymbol{c'} = \boldsymbol{c} + \boldsymbol{e} = \boldsymbol{m}SGP + \boldsymbol{e}. \tag{3}$$

The decryption algorithm is as follows.

1. For a ciphertext $\boldsymbol{c'}$, find P^{-1} using the secret key. Then multiply $\boldsymbol{c'}$ by P^{-1} to obtain

$$\boldsymbol{c'}P^{-1} = (\boldsymbol{m}SGP + \boldsymbol{e})P^{-1} = \boldsymbol{m}SG + \boldsymbol{e}P^{-1}. \tag{4}$$

2. As P is a permutation matrix, $P^{-1} = P^T$ is also a permutation matrix. Therefore, $\boldsymbol{e}P^{-1}$ is a vector with the same weight as $\boldsymbol{e}$. Thus $\boldsymbol{c'}P^{-1}$ can be decoded to obtain $\boldsymbol{m}S$.

3. Multiply $\boldsymbol{m}S$ by S^{-1} to obtain the plaintext $\boldsymbol{m}$.

2.2 The Niederreiter Cryptosystem

Niederreiter, in 1986, introduced a code-based encryption scheme that can be considered the dual of the McEliece cryptosystem [2]. It is based on the hardness of the syndrome-decoding problem. The Niederreiter cryptosystem, similar to the McEliece cryptosystem, is a public key encryption scheme where the public key is the product of a non-singular $k\times n$ scrambling matrix, an $(n-k)\times n$ parity check matrix of a code $C(n,k)$, and an $n\times n$ permutation matrix.

Similar to the McEliece cryptosystem, a code $C(n,k)$ is denoted by its parity check matrix H, an $(n-k)\times(n-k)$ scrambling matrix S, and an $n\times n$ permutation matrix P. The secret key consists of the three tuples of the three mentioned matrices, $sk = (S,H,P)$, and the public key is $pk = SHP$.

The encryption algorithm for the Niederreiter cryptosystem is as follows.

1. For a plaintext $\boldsymbol{m}$ of the length n and the weight t, the corresponding ciphertext is
$$\boldsymbol{c'} = SHP\boldsymbol{m}^T. \tag{5}$$

The decryption algorithm for the Niederreiter cryptosystem is as follows.

1. For ciphertext $\boldsymbol{c'}$ find $S^{-1}\boldsymbol{c'} = HP\boldsymbol{m}^T$.
2. Use syndrome decoding to obtain $P\boldsymbol{m}^T$.
3. $P^{-1} \times P\boldsymbol{m}^T$ gives the plaintext $\boldsymbol{m}$.

2.3 CFS digital signature scheme

The CFS (Courtois-Finiasz-Sendrier) signature scheme is a well-known code-based digital signature scheme based on the Niederreiter cryptosystem [2]. In this scheme, a document is hashed to compress its size to n bits, where n is the length of the code used in the cryptosystem. The CFS scheme considers the hashed document a ciphertext. The signature algorithm, verification algorithm, security, and drawback of the CFS signature are given below.

The signing algorithm of the CFS signature scheme has four steps.

1. For a document $\boldsymbol{doc}$, hash it using a hash function $h()$ to find $h(\boldsymbol{doc})$ and set $i = 0$.
2. Find $h(h(\boldsymbol{doc})|i)$, where $|$ denotes concatenation of $h(\boldsymbol{doc})$ and i.
3. Decrypt $h(h(\boldsymbol{doc})|i)$ using the decryption algorithm of the Niederreiter cryptosystem to find $\boldsymbol{sig}$. If this step fails, increase i by 1 and repeat step 2.
4. Output $(\boldsymbol{sig}, i)$ as the signature of the document $\boldsymbol{doc}$.

The verification algorithm of the CFS signature scheme is as follows.

1. For the signature $(\boldsymbol{sig}, i)$ of a document doc, find $h(h(\boldsymbol{doc})|i)$.

2. The signature is valid if

$$h(h(\boldsymbol{doc})|i) = SHP\boldsymbol{sig}^T, \tag{6}$$

otherwise, the signature is not valid.

2.4 CFS performance analysis

Public key code-based cryptosystems are not used in digital signatures mainly because of low probability of successful decoding the valid signature. The success rate is less than 1 as the ciphertexts do not cover the entire vector space, so a signature may not be obtained [14–16]. In the CFS signature scheme, an integer i is appended to $\boldsymbol{doc}$ and changed if decryption of $h(h(\boldsymbol{doc})|i)$ fails to find $\boldsymbol{sig}$. This results in a reduction in speed and an increase in the the complexity and processing time. The probability of a syndrome being decoded is approximately equal to decodable syndrome space over total vector space. Thus, on average, it will take $t!$ executions of the CFS signature algorithm to find a valid signature [3]. Therefore, the probability of success is $\frac{1}{t!}$ [3]. In the next section, a code-based digital signature proposed which has a success rate of 1 and has lower complexity than the CFS scheme.

3 Proposed Code-Based Digital Signature

The proposed code-based digital signature scheme is a probabilistic polynominal time algorithm for key generation, signing and verification. The code-based cryptosystem is used in the public key infrastructure of the proposed code-based digital signature. The dual matrix A is playing the important roles of key generation, document signing and verification algorithms.

Proposed algorithms

- Key Generation: $(pk, pr_k) \leftarrow Gen(\lambda)$.
- Document/Message Signing: $\sigma \leftarrow Sign(pk, pr_k, \boldsymbol{doc})$, where σ and $\boldsymbol{doc}$ denotes the signature and the document, respectively.
- Signature Verification: $Ver(\sigma, pk, \boldsymbol{doc}) \in 0, 1$.

The key generation algorithm constructs a pair of public and private-keys that can be used by encryption and decryption algorithms for confidentiality and authenticity purposes.

3.1 Dual Matrix A

This section proposes a key generation algorithm for both code-based cryptosystems and digital signatures. The proposed algorithm generates a tree-tuple public key based on specific an inverse matrix with dual functionality. Let matrix A denote a dual parity check matrix with the order of $n \times (n-k)$. Matrix A offers two different functionalities as it function as H^T and H^{-1} (parity check transpose and parity check inverse) at the same time.

In ECC codes, the rows of the parity check matrix are orthogonal to the rows of the generator matrix such that $GH^T = \mathbf{0}$. Consider a matrix A such that $HA = I_{n-k}$ and $GA = \mathbf{0}$. Thus, A is an inverse parity check matrix and the transpose of a parity check matrix, so that $GH^T = GA$. Hence A can be constructed using H^T and a permutation matrix P' that satisfies $A = H^T P'$. Then

$$GA = \mathbf{0} \text{ and } GH^T = \mathbf{0} \text{ so } A = H^T P', \text{ and}$$

$$HA = H(H^T P') = (HH^T)P' = I_{n-k},$$

so $P' = (HH^T)^{-1}$, and matrix A can be constructed only if the $(n-k) \times (n-k)$ matrix HH^T is non-singular. Hence a code $C(n,k)$ can have a dual inverse matrix A, when the square matrix HH^T is non-singular. Let p_A denote the total number of possible linear combinations of column vectors of A, then

$$p_A = \prod_{i=0}^{n-k-1} (2^{n-k} - 2^i) \tag{7}$$

3.2 Key Generation

The key generation algorithm provides the public and private keys using the generator matrix G of the code $C(n,k)$ and the dual matrix A, where

$$GA = \mathbf{0} \tag{8}$$

$$HA = I_{n-k} \tag{9}$$

The following matrices are used by the key generation algorithm.

– Matrix G, represents the generator matrix of the order $k \times n$.

- Matrix H, represents the parity check matrix of the order $(n-k) \times n$
- Matrix A, represents the dual matrix of the order $n \times (n-k)$
- Matrix S, represents a scrambling non-singular matrix of the order $k \times k$
- Matrix P, represents a permutation matrix of the order $n \times n$
- Matrix L, represents a non-singular matrix of the order $(n-k) \times (n-k)$

Key Generation Algorithm $Gen(\lambda)$

Given the generator matrix G with non-singular $HH^T \leftarrow C(n,k)$

$A = H^T P' \leftarrow$ Construct $P' = (HH^T)^{-1}$.

Public key: $pk \leftarrow (SGP, L^{-1}HP, P^{-1}AHP)$.

Private key: $pr_k \leftarrow (S^{-1}, P^{-1}, G, P^{-1}AL)$.

where the generator matrix G and its parity check matrix H are blinded by using a non-singular random scrambling matrix S and a random permutation matrix P, and the non-singular random matrix L and a random permutation matrix P are used to blind the dual matrix A. $P^{-1}AHP$ is used by the verification algorithm to verify the signed digital signatures.

Theorem 1
The public key $L^{-1}HP$ has many inverses, and the probability of constructing a particular inverse of $L^{-1}HP$ is trivial.

Proof **:** The parity check matrix is a full rank matrix and it is not unique [17]. The inverse of its matrix has $n-k$ columns, each of which can have 2^k different values, so the number of its valid inverse matrices is $2^{k\times(n-k)}$ [17]. Therefore, the public key $L^{-1}HP$ is a full rank matrix, hence, the probability of constructing a particular inverse of the public key $L^{-1}HP$ would be equal to $\frac{1}{2^{k\times(n-k)}}$ which is trivial.

3.3 The signing algorithm

The signing algorithm of the proposed signature scheme uses both public keys to sign a document as follows.

Signing Algorithm $Sign(pk, pr_k, \boldsymbol{doc})$

```
1 - Use the hash function to compress the size of the document to $n$ bits.
    $h(\boldsymbol{doc}) \leftarrow$ hash document $\boldsymbol{doc}$,
    $h(h(\boldsymbol{doc})) \leftarrow$ hash the $h(\boldsymbol{doc})$

2 - let $\boldsymbol{s}$ denote an $n-k$ bit vectors, such that
    $\boldsymbol{s} \leftarrow h(\boldsymbol{doc})(P^{-1}AL)$

3 - Constructs a codeword $\boldsymbol{c}$ using the $h(\boldsymbol{doc})$ and vector $\boldsymbol{s}$.
    $\boldsymbol{sig}SGP \leftarrow h(\boldsymbol{doc}) + \boldsymbol{s}(L^{-1}HP)$

4 - Decode the codeword $\boldsymbol{c}$ to obtain $\boldsymbol{sig}$.
    $\boldsymbol{sig}SG \leftarrow (\boldsymbol{sig}SGP)(P^{-1})$
    $\boldsymbol{sig}S \leftarrow$ decode $\boldsymbol{sig}SG$
    $\boldsymbol{sig} \leftarrow (\boldsymbol{sig}S)(S^{-1})$

5 - Use $h(h(\boldsymbol{doc}))$ and private $pr_k$ to construct the $\boldsymbol{d}$ vector, (where $\boldsymbol{d}$ denotes another $n-k$ bit
    vector)
    $\boldsymbol{d} \leftarrow h(h(\boldsymbol{doc}))(P^{-1}AL) + \boldsymbol{s}$.

6 - Output $\sigma = (\boldsymbol{sig}, \boldsymbol{d})$.
    Transmit $(\boldsymbol{sig}, \boldsymbol{d})$ and document $\boldsymbol{doc}$ to the receiver for signature verification.
```

Theorem 2
The $h(\boldsymbol{doc}) + \boldsymbol{s}(L^{-1}HP)$ is a valid codeword of the code $C(n,k)$ with a generator matrix $G' = SGP$.

Proof **:** Matrix S and P are full rank as they are non-singular matrices. Therefore, the rank of SGP equals k, and the rank of $P^{-1}AL$ is equal to $n-k$, respectively. Thus the row vectors of SGP and the column vectors of $P^{-1}AL$ are orthogonal, therefore, $P^{-1}AL$ generates the nullspace for every code that is spanned by SGP. Hence, the transpose of $P^{-1}AL$ is a parity check matrix of SGP.

For a codeword $\boldsymbol{c} \in C(n,k)$, $\boldsymbol{c}H^{'T} = \mathbf{0}$ with the generator matrix $G' = SGP$ of order $k \times n$ and $H'^{T} = P^{-1}AL$.

$$\boldsymbol{c} = \boldsymbol{sig}SGP = h(\boldsymbol{doc}) + \boldsymbol{s}(L^{-1}HP)$$

The vector $\boldsymbol{s}$ is equal to $h(\boldsymbol{doc})(P^{-1}AL)$

$$\boldsymbol{sig}SGP = h(\boldsymbol{doc}) + h(\boldsymbol{doc})(P^{-1}AL)(L^{-1}HP)$$

$$\boldsymbol{sig}SGP = h(\boldsymbol{doc}) + h(\boldsymbol{doc})(P^{-1}AHP)$$

Therefore $\boldsymbol{c}H'^{T} = [\boldsymbol{sig}SGP][P^{-1}AL]$

$$\boldsymbol{c}H'^{T} = h(\boldsymbol{doc})(P^{-1}AL) + h(\boldsymbol{doc})(P^{-1}AHP)(P^{-1}AL)$$

$$= h(\boldsymbol{doc})(P^{-1}AL) + h(\boldsymbol{doc})(P^{-1}AL)$$

$$= \mathbf{0}$$

3.4 The verification algorithm

The verification algorithm of the proposed code-based digital signature scheme is as follows.

Verification Algorithm $Ver(\sigma, pk, \boldsymbol{doc})$

1− Use the hash function $h()$ to hash the received document to construct $h(\boldsymbol{doc})$, $h(h(\boldsymbol{doc}))$ and assign variable a such that,
$a \leftarrow \boldsymbol{sig}SGP$

2− Use the public key and $\boldsymbol{d}$ value to compute $v_1 = \boldsymbol{s}(L^{-1}HP)$, where v_1 denotes an n bit vector.
$v_1 \leftarrow \boldsymbol{s}(L^{-1}HP) = h(h(\boldsymbol{doc}))(P^{-1}AHP) + \boldsymbol{d}(L^{-1}HP)$

$$\boldsymbol{d} = h(h(\boldsymbol{doc}))(P^{-1}AL) + \boldsymbol{s}$$

$$\boldsymbol{d}(L^{-1}HP) = (h(h(\boldsymbol{doc}))(P^{-1}AL) + \boldsymbol{s})(L^{-1}HP)$$

$$\boldsymbol{d}(L^{-1}HP) = h(h(\boldsymbol{doc}))(P^{-1}AL)(L^{-1}HP) + \boldsymbol{s}(L^{-1}HP)$$

Note that, $(P^{-1}AL)(L^{-1}HP) = P^{-1}AHP$, therefore,

$$v_1 = \boldsymbol{s}(L^{-1}HP) = h(h(\boldsymbol{doc}))(P^{-1}AHP) + \boldsymbol{d}(L^{-1}HP) \tag{10}$$

3− Use the public key $(P^{-1}AHP)$ to compute $v_2 = \boldsymbol{s}(L^{-1}HP)$, where v_2 denotes another n bit vector.
$v_2 \leftarrow \boldsymbol{s}(L^{-1}HP) = h(\boldsymbol{doc})(P^{-1}AHP)$

$$\boldsymbol{sig}SGP = h(\boldsymbol{doc}) + \boldsymbol{s}(L^{-1}HP)$$

$$\boldsymbol{s}(L^{-1}HP) = \boldsymbol{sig}(SGP) + h(\boldsymbol{doc})$$

$$\boldsymbol{s}(L^{-1}HP)(P^{-1}AHP) = \boldsymbol{sig}(SGP)(P^{-1}AHP) + h(\boldsymbol{doc})(P^{-1}AHP)$$

Note that, $(SGP)(P^{-1}AHP) = \mathbf{0}$ and $(L^{-1}HP)(P^{-1}AHP) = L^{-1}HP$, therefore,

$$v_2 = \boldsymbol{s}(L^{-1}HP) = h(\boldsymbol{doc})(P^{-1}AHP) \tag{11}$$

4− Check the integrity condition if

$$v_1 = v_2$$

otherwise, the verification fails.

5− Having $v_1 = \boldsymbol{s}(L^{-1}HP)$ and given $h(\boldsymbol{doc})$, compute $\boldsymbol{c}$.
$\boldsymbol{c} \leftarrow h(\boldsymbol{doc}) + \boldsymbol{s}(L^{-1}HP)$

6− The verification will be successful if the second condition is valid as follows.

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

otherwise, the verification fails.

Any changes or modification by an adversary should be detected by the verification algorithm. The integrity condition protects the accuracy of the transmitted signature by comparing the v_1 and v_2 variables. Hence v_1 is not dependent on the signature $\boldsymbol{sig}$ and v_2 does not depend on the private key, although the integrity condition is met when $v_1 = v_2$.

3.5 Proposed digital signature example

Let $n = 12$ and $k = 5$ be the indexes of the following generator matrix G,

$$G = (I_k|P_{k\times(n-k)}) = \left(\begin{array}{c|ccccccc} & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ I_k & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right),$$

its corresponding parity check matrix H, and its constructed dual inverse matrix A.

$$H_{(n-k)\times n} = \left(\begin{array}{ccccc|c} 1 & 0 & 0 & 0 & 1 & \\ 0 & 1 & 0 & 1 & 1 & \\ 0 & 0 & 1 & 0 & 0 & \\ 1 & 0 & 1 & 1 & 1 & I_{n-k} \\ 0 & 1 & 1 & 0 & 0 & \\ 1 & 0 & 1 & 1 & 0 & \\ 1 & 0 & 1 & 0 & 1 & \end{array}\right), A_{n\times(n-k)} = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

A non-singular matrix L of order $n-k$ and inverse matrix S^{-1} of order $k\times k$.

$$L_{(n-k)\times(n-k)} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}, S_{k\times k} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

A permutation matrix P of order $n\times n$.

$$P_{n\times n} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

1. Alice hashes a document $\boldsymbol{doc}$ by a hash function $h()$ to find $h(\boldsymbol{doc})$ to size the document to n bits as the lenght of the code here. $h(\boldsymbol{doc}) = 100110010001$ and $h(h(\boldsymbol{doc})) = 110001110111$.

2. Constructs an $n-k$ bit vector $\boldsymbol{s}$, such that $\boldsymbol{s} = h(\boldsymbol{doc})(Q)$.

$$\boldsymbol{s} = 0101111$$

3. Constructs a codeword $h(\boldsymbol{doc}) + \boldsymbol{s}(L^{-1}HP)$ of the code $C(n,k)$.

$$\begin{aligned} \boldsymbol{c} = \boldsymbol{sig}SGP = h(\boldsymbol{doc}) + \boldsymbol{s}(L^{-1}HP) \\ &= 100110010001 + (0101111)(L^{-1}HP) \\ &= 100110010001 + 000110110010 \\ &= 100000100011 \end{aligned}$$

4. She decodes the codeword to obtain the $\boldsymbol{sig} = 01010$.

5. She decodes $\boldsymbol{d}$, using pr_k and vector $\boldsymbol{s}$.

$$\begin{aligned} \boldsymbol{d} &= h(h(\boldsymbol{doc}))(P^{-1}AL) + \boldsymbol{s} \\ &= (110001110111)(P^{-1}AL) + 0010111 \\ &= 1000110 + 0010111 \\ &= 1101001 \end{aligned}$$

6. She outputs $(\boldsymbol{sig}, \boldsymbol{d})$ along with the document $\boldsymbol{doc}$.

Bob verifies the signature as follows.

1. Use the hash function $h()$ to hash the received document to construct $h(\boldsymbol{doc})$, $h(h(\boldsymbol{doc}))$ and assign $a = \boldsymbol{sig}SGP$.

$$\begin{aligned} &h(\boldsymbol{doc}) = 100110010001, \\ &h(h(\boldsymbol{doc})) = 110001110111, \\ &a = \boldsymbol{sig}SGP = (0000110)(SGP), \\ &a = 100000100011. \end{aligned}$$

2. Use Alice's public key and $\boldsymbol{d}$ value to compute v_1

$$v_1 = h(h(\boldsymbol{doc}))(P^{-1}AHP) + \boldsymbol{d}(L^{-1}HP)$$

$$= 110001110111(P^{-1}AHP) + 1101001(L^{-1}HP)$$

$$= 110111000110 + 110001110100$$

$$= 000110110010$$

3. Use Alice's public key to compute $v_2 = \boldsymbol{s}(L^{-1}HP)$

$$v_2 = h(\boldsymbol{doc})(P^{-1}AHP)$$

$$= 100110010001(P^{-1}AHP)$$

$$= 000110110010$$

3. Use Alice's public key $pk = (P^{-1}AHP)$ to compute $\boldsymbol{s}L^{-1}HP$.

$$h(\boldsymbol{doc})(P^{-1}AHP) = (\boldsymbol{sig}SGP + \boldsymbol{s}L^{-1}HP)(P^{-1}AHP)$$

$$h(\boldsymbol{doc})(P^{-1}AHP) = \mathbf{0} + (\boldsymbol{s}L^{-1}HP)(P^{-1}AHP)$$

$$\boldsymbol{s}(L^{-1}HP) = h(\boldsymbol{doc})(P^{-1}AHP)$$

$$= 100110010001(P^{-1}AHP)$$

$$= 000110110010$$

4. Bob continues to the next level, if the integrity condition is met, $v_1 = v_2$, otherwise the verification failed.

5. Having $v_1 = \boldsymbol{s}(L^{-1}HP)$, and given $h(\boldsymbol{doc})$, Bob computes $\boldsymbol{c}$

$$\boldsymbol{c} = h(\boldsymbol{doc}) + \boldsymbol{s}(L^{-1}HP)$$

$$= 100110010001 + 000110110010$$

$$= 10000010011$$

6. The verification is successful as

$$a = \boldsymbol{c}$$

It is recommended that every time a new scheme signs the same document, the signing algorithm should generate a different digital signature. This can be achieved by simply concatenating an n bit random vector $\boldsymbol{r}$ into the chosen document. However, it increases the size of the signature, and the random vector should output $(\boldsymbol{r})$ along with $(\boldsymbol{sig}, \boldsymbol{d})$.

3.6 Performance and security analysis

On average, the CFS code-based signature and modified CFS schemes will take $t!$ of executions to find a valid signature [18], therefore their success rate is $\frac{1}{t!}$.

The size of the signature and the speed of the signing process are the two main factors of a digital signature algorithm. The proposed code-based scheme constructs small signatures and increases the speed of the signing process. The proposed algorithm provides a signature with 130 bytes ($n = 1024$) compared to Bliss-IV, with a signature size of 6500 bytes and a low success rate of 5.6 [19], and qTesla-III, which can construct a signature with a size of 2848 bytes [20, 21]. Hence, the presented model has shorter signature lengths and can sign documents faster. As mentioned, speed is a critical factor in various applications of digital signature schemes such as online banking, e-commerce, and blockchains (Bitcoin, Ethereum).

There are different types of cryptanalysis attack models that adversaries may use to discover the weaknesses of the algorithm, gain access to the contents of the messages/document, and break the secret codes [22]. The proposed algorithm blinds the generator matrix by interchanging the rows, columns, and their linear combinations using permutation and scrambling non-singular matrices. Therefore the robust and secure algorithm should not verify a forged document signed by an adversary through generic, directed, or adaptive chosen-message attacks [25]. The proposed scheme encrypts vector $\boldsymbol{s}$ and $\boldsymbol{d}$ and ensures that the probability of constructing the private key from the public key is negligible as follows

$Pr[(Adv, \gamma) = 1] < \varepsilon(\gamma)$, where Adv, γ denote adversary and security parameter.

Let's analyze if an adversary could form a structural attack by constructing the private key from the public key. The challenger provides full access to an adversary to input any selected document and receive valid signature. Then an adversary uses its private-key (sk_2) to sign a document and output ($\boldsymbol{sig}, \boldsymbol{d}$) to be verified by the challenger. The challenger uses the verification algorithm and reaches step (4) to challenge the first condition.

$$v_1 = v_2$$
$$h(\boldsymbol{doc})(P^{-1}AHP) = h(h(\boldsymbol{doc}))(P^{-1}AHP) + \boldsymbol{d}(L^{-1}HP)$$

The left side of the above equation ($h(\boldsymbol{doc})(P^{-1}AHP)$) is independent of the adversary private key (sk_2), while the $\boldsymbol{d}$ value on the right is constructed by the adversary private key during the signing process.

The $\boldsymbol{d}$ value is equal

$$\boldsymbol{d} = h(h(\boldsymbol{doc}))(sk_2) + h(\boldsymbol{doc})(sk_2)$$

Therefore,

$$h(\boldsymbol{doc})(P^{-1}AHP) = h(h(\boldsymbol{doc}))(P^{-1}AHP) + (h(h(\boldsymbol{doc}))(sk_2) + h(\boldsymbol{doc})(sk_2))(L^{-1}HP)$$

$$(h(\boldsymbol{doc}) + h(h(\boldsymbol{doc})))(P^{-1}AHP) = (h(h(\boldsymbol{doc})) + h(\boldsymbol{doc}))(sk_2)(L^{-1}HP)$$

$$(P^{-1}AHP) = (sk_2)(L^{-1}HP)$$

The above equation would be valid if and only if $(sk_2) = (sk)$.
Lets consider that an adversary selects the $(L^{-1}H''P)^{-1}$ as the private key. Then

$$\begin{aligned}(sk_2)(L^{-1}HP) &= (L^{-1}H''P)^{-1}(L^{-1}HP)\\ &= (H''P)^{-1}(L)(L^{-1})(HP)\\ &= P^{-1}H''^{-1}HP\end{aligned}$$

So if $(H'')^{-1} = A$, then the $(sk_2)(L^{-1}HP)$ would be equal to the third public key $(P^{-1}AHP)$ and the signed document can be verified successfully, with the result that the adversary can forge a signature with respect to the given public key. The algorithm is secure when the probability of success is significantly low and negligible

$$Pr[(Adv, \gamma) = 1] < \varepsilon(\gamma)$$

where Adv denotes the adversary and γ denotes the security parameter, respectively [23, 24]. The matrix L is a square matrix, and the inverse of (L^{-1}) would be equal to L matrix, as this matrix is a non-singular with an order of $(n-k) \times (n-k)$. However, this is not the case with the parity check matrix as matrix H is a full rank and non-square matrix of order $(n-k) \times n$, and therefore, the H'' matrix should be a full rank matrix with the same order of parity check matrix $(n-k) \times n$. Hence the inverse of the H matrix is not unique, and based on $Theorem$ 1, the probability of $(H'')^{-1} = H^{-1}$ would be insignificant, and the proposed algorithms are secure against structural attacks [22]. Therefore, the possibility of constructing an adversary private key from the algorithm's public key would be equal to $2^{-(k\times(n-k))}$. As result, the probability that an adversary signs a verifiable document is negligible.

$$Pr[(Adv, \gamma) = 1 \leftarrow Pr[pr_{k2} = pr_2] \leftarrow Pr[(H'')^{-1} = H^{-1}]] < \varepsilon(\gamma)$$

Hence, the chances that an adversary forges the signature by accessing the signing algorithm are extremely low and negligible, and the proposed digital signature algorithms are secure.

4 Conclusion

The CFS digital signature scheme and its drawbacks have been described in the above section. In addition, the primary approach to overcoming these drawbacks in the form of the CFS signature scheme was elucidated. It was also shown that even in these approaches, the digital signature schemes based on codes are still significantly slow because the ciphertexts only cover part of the vector space. Therefore it requires $t!$ number of executions (t, the error correction capability of the code) of the CFS signature algorithm to find a valid signature. A code-based digital signature scheme was proposed, which provides practical code-based digital signatures. In fact, the proposed schemes require no decoding syndrome search operation, which optimizes the digital signature construction process.

The proposed public key allows the encryption of the value vectors. Therefore, the verification process can examine the integrity and authenticity of the signature. In fact, the proposed schemes is secure, as the probability of an adversary being able to forge a signature that can be verified is trivial. In addition, the proposed algorithms are safe against any structural attack. It has been proven that the probability of constructing the private key from the public key is negligible. Moreover, it was shown how the proposed schemes can considerably increase the speed of the signature algorithm in code-based digital signatures.

References

[1] R. J. McEliece, “A public-key cryptosystem based on algebraic coding theory,” *Jet Propulsion Lab*, DSN Tech. Rep. 42.44, pp. 114-116, 1978.

[2] H. Niederreiter, “Knapsack-type cryptosystems and algebraic coding theory,” *Problems of Control and Information Theory*, vol. 15, pp. 159-166, 1986.

[3] N. Courtois, M. Finiasz, N. Sendrier, “How to achieve a McEliece-based digital signature scheme,” *Advances in cryptology ASIACRYPT 2001*, Lecture notes in computer science, vol. 2248, Springer, Berlin, pp. 157-174, 2001.

[4] P. W. Shor, “Algorithms for quantum computation: discrete logarithms and factoring,” Proc. *35th Annual Symp on Foundations of Computer Science*, Santa Fe, NM, pp.124–134, 1994.

[5] N. Sendrier, "Code based cryptography: State of art and perspectives," *IEEE Privacy Security*, vol. 15, no. 4, pp. 44-50, 2017.

[6] M. Baldi, "Post-quantum cryptographic schemes based on codes," *Proc. Int. Conf. on High Perf. Computing and Simulation*, Genoa, Italy, pp. 908–910, 2017.

[7] T. N. R. Rao, K. H. Nam, "A private key algebraic coded cryptosystem," *Proc. Crytpology*, pp.35–48, 1986.

[8] R. Hooshmand, M. R. Aref, "Efficient secure channel coding scheme based on low-density lattice codes," *IET Communications.*, vol. 10, no. 11, pp. 1365-1373, 2016.

[9] T. N. R. Rao, "Joint encryption and error correction schemes," *roc. Int. Symp. on Computer Architecture*, Ann Arbor, MI, pp. 240–241, 1984.

[10] P.L. Cayrel, M. Meziani, "Post-quantum Cryptography: Code-Based Signatures," *Advances in Computer Science and Information Technology*, pp.82-99, 2013.

[11] R. Nojima, H. Imai, K. Kobara, K. Morozov, "Semantic security for the McEliece cryptosystem without random oracles," *Design, Codes and Cryptography*, vol. 49 (1-3), pp.289-305, 2008.

[12] P.L. Cayrel, P. Gaborit, M. Girault, "Identity based identification and signature schemes using correcting codes," *International Workshop on Coding and Cryptography*, pp. 69–78, 2007.

[13] E. R. Berlekamp, R. J. McEliece, and H. C. A. Van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Trans. Information Theory*, vol. 24, no. 3, pp. 384–386, 1978.

[14] P. Cayrel, M. Meziani, TH. Kim, H. Adeli, "Post-quantum Cryptography: Code-Based Signatures," *Advances in Computer Science and Information Technology*, vol. 6059, pp. 82-99, 2010.

[15] S.B Xu, J.M Doumen, Henk Van Tilborg, "On the security of digital signature scheme based on error correcting codes," *Designs, codes, and cryptography*, vol. 28 (2), pp. 187-199, 2003.

[16] W. Xinmei, "Digital signature scheme based on error correcting codes," *Electronics Letters*, vol. 26, no. 13, pp. 898–899, 1990.

[17] M. Esmaeili, T.A.Gulliver, “Application of Linear Block Codes in Cryptography,” *University of Victoria department of Electrical and Computer Engineering*, Ph.D. Dissertations. Chapter 5, Security analysis. pp. 45-53, 2019.

[18] Matthieu Finiasz, “Parallel-CFS: Strengthening the CFS McEliece-Based Signature Scheme,” *Selected Areas in Cryptography*, Springer Computer Science eBooks, Berlin, pp. 159-170, 2011.

[19] T. Pöppelmann, L. Ducas, T. Güneysu, “Enhanced Lattice-Based Signatures on Reconfigurable Hardware,” *Cryptographic Hardware and Embedded Systems*, Cryptographic Hardware and Embedded Systems, p.353-370, CHES 2014.

[20] J. Howe, T. Pöppelmann, M. O’neill, E. O’sullivan, T. Güneysu, “Practical Lattice-Based Digital Signature Schemes,” *Practical Lattice-Based Digital Signature Schemes*, ACM Transactions on Embedded Computing Systems, vol.14 (3), p.1-24, 2015.

[21] D. Das, J. Hoffstein, J. Pipher, W. Whyte, Z. Zhang, “Modular lattice signatures, revisited,” *Designs, codes and cryptography*, Springer Science New York, US. vol.88 (3), p.505-532, 2019.

[22] Rao T. N. R., Nam K. H., “A private key algebraic coded cryptosystem,” *Advanced in Crytpology*, CRYPTO’86, p.35–48, 1986.

[23] D. Diemert, K. Gellert, T. Jager, Lin Lyu, “More Efficient Digital Signatures with Tight Multi-user Security,” *Public-Key Cryptography*, p.1-31, 2021.

[24] M. Bellare, C. Namprempre, G. Neven, “Security Proofs for Identity-Based Identification and Signature Schemes,” *Journal of cryptology*, vol.22 (1), p.1-61, 2009.

[25] S. Goldwasser, S. Micali, R. Rivest, “A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks,” *SIAM journal on computing*, vol.17 (2), p.281-308, 1988.