A new code-based digital signature based on the McEliece cryptosystem

Farshid Haidary Makoui, Thomas Aaron Gulliver, and Mohammad Dakhilalian

2023

Original citation:

Makoui, F. H., Gulliver, T. A., & Dakhilalian, M. (2023). A new code-based digital signature based on the McEliece cryptosystem. *IET Communications, 17*(10), 1199–1207. https://doi.org/10.1049/cmu2.12607

Received: 17 December 2022 | Revised: 23 February 2023 | Accepted: 16 March 2023

IET Communications

DOI: 10.1049/cmu2.12607

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

A new code-based digital signature based on the McEliece cryptosystem

Farshid Haidary Makoui[1] | **Thomas Aaron Gulliver**[1] | **Mohammad Dakhilalian**[2]

[1]Department of Electrical and Computer Engineering, University of Victoria, Victoria, British Columbia, Canada

[2]Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran

Correspondence
Farshid Haidary Makoui, Department of Electrical and Computer Engineering, University of Victoria, Victoria, British Columbia, Canada.
Email: makoui@uvic.ca

Abstract

Digital signature schemes are used for the authentication and verification of signatures. The Courtois–Finiasz–Sendrier (CFS) digital signature is a well-known code-based digital signature scheme based on the Niederreiter cryptosystem. However, it is not widely used due to the computation time of the signing algorithm. Most code-based digital signature schemes are based on the Niederreiter cryptosystem. This paper proposes a new code-based digital signature that is based on the McEliece cryptosystem. Key generation, signing, and verification algorithms are presented. The key generation algorithm constructs a public key using random inverse matrices. The signing algorithm has lower complexity and requires less computation time than the CFS scheme to sign a document. The verification algorithm is able to detect forgeries. It is shown that the proposed scheme is secure against public key structural attacks.

1 | INTRODUCTION

In 1994, Shor proposed an algorithm indicating that quantum attacks are a serious threat to cryptographic primitives [1]. Post-quantum cryptography [2] is the development of cryptographic mechanisms [3–5] that are secure against quantum attacks. Code-based cryptographic primitives [6] have been shown to be resistant to quantum attacks. The first code-based cryptosystem was introduced by McEliece and is known as the McEliece cryptosystem [7]. The security of this cryptosystem is based on the hardness of the decoding and code distinguishability problems [8, 9]. The inability to distinguish between a scrambled parity check matrix and a random matrix is an NP-problem [9, 10], so decoding a linear code without knowledge of its algebraic structure is also an NP-problem [11].

Another code-based cryptosystem is the Niederreiter cryptosystem [12] and this has been used in code-based digital signatures [13]. However, existing Niederreiter digital signatures are not widely used due to the computation time required for signing. A drawback of code-based cryptosystems is that the number of valid codewords is smaller than the size of the vector space [8]. Thus, a signature may not be decodable [14]. This increases the computation time for signing as the algorithm must find a valid signature that can be verified.

Many applications and services use digital signatures, such as web authentication [15], data integrity, message authentication [16], digital certificates [17], blockchain [18], and Bitcoin [19]. This paper proposes a McEliece-based digital signature scheme that employs the entire vector space so that a valid digital signature can be generated with a higher success rate and lower computation time than the Courtois–Finiasz–Sendrier (CFS) scheme. The proposed scheme has key generation, signing, and verification algorithms. The key generation algorithm constructs a public key using random inverse matrices. This is used by the signing and verification algorithms to sign a document that can be verified by a receiver and detect forgeries.

2 | CODE BASED CRYPTOSYSTEMS

In 1978, the McEliece cryptosystem was introduced as the first code-based cryptosystem [7]. In 1986, the dual version of the McEliece cryptosystem, called the Niederreiter cryptosystem, was introduced. The Niederreiter algorithm is faster and so it is used for digital signatures such as in the CFS scheme.

2.1 | The McEliece cryptosystem

In the McEliece cryptosystem, the plaintext bits are first scrambled, and then the corresponding codeword is permuted. Up to t bits are flipped, where t is 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 these three matrices. The encryption and decryption algorithms of this system are given below.

The systematic form of the generator matrix and parity check matrix of a linear code are $G_{k\times n} = (I_k | P'_{k\times(n-k)})$ and $H_{(n-k)\times n} = (P'^T | I_{n-k})$, respectively, 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 generator matrix G along with 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 length n so that $w(\boldsymbol{e}) \leq t$, where t is the error correcting capability of the code and $w(\cdot)$ denotes the Hamming weight. The cipher text is

$$\boldsymbol{c}' = \boldsymbol{c} + \boldsymbol{e} = \boldsymbol{m}SGP + \boldsymbol{e} \tag{1}$$

The decryption algorithm is as follows.

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

$$\boldsymbol{c}'P^{-1} = (\boldsymbol{m}SGP + \boldsymbol{e})P^{-1} = \boldsymbol{m}SG + \boldsymbol{e}P^{-1} \tag{2}$$

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

The Niederreiter cryptosystem can be considered the dual of the McEliece cryptosystem [12]. It is based on the hardness of the syndrome decoding problem. Similar to the McEliece cryptosystem, a code $C(n, k)$ is chosen with a parity check matrix H along with an $(n-k) \times (n-k)$ scrambling matrix S and an $n \times n$ permutation matrix P. The Niederreiter cryptosystem is a public key encryption scheme where the public key is the product of S, H, and P. Thus, the secret key is $sk = (S, H, P)$ and the public key is $pk = SHP$. The encryption algorithm for the Niederreiter cryptosystem is as follows.

1. For a plaintext $\boldsymbol{m}$ of length n and weight t, the corresponding cipher text is

$$\boldsymbol{c}' = SHP\boldsymbol{m}^T. \tag{3}$$

The decryption algorithm is as follows.

1. For cipher text $\boldsymbol{c}'$ 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 signature scheme is a well-known code-based digital signature scheme based on the Niederreiter cryptosystem [12]. In this scheme, a document is first 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 as the cipher text. The signing algorithm, verification algorithm, security, and drawbacks of this scheme are given below. The signing algorithm has four steps.

1. For a document $\boldsymbol{doc}$, hash it using a hash function $h(\cdot)$ to find $h(\boldsymbol{doc})$ and set $i = 0$.
2. Find $h(h(\boldsymbol{doc})|i)$, where | denotes concatenation.
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

$$h(h(\boldsymbol{doc})|i) = SHP\boldsymbol{sig}^T, \tag{4}$$

otherwise, the signature is not valid.

2.4 | CFS performance analysis

The complexity of signing is the main reason code-based signatures are not employed in practical applications. The signing success rate is the probability of successful decoding in step 3 of the CFS signing algorithm. This rate is less than 1 because the cipher texts do not cover the entire vector space, so a signature may not be obtained [8, 14, 20]. In the CFS signature scheme, an integer i is appended to $\boldsymbol{doc}$ iteratively to ensure a signature is obtained. This increases the complexity and the computation time. It has been shown that, on average, it will take $t!$ executions of the CFS signature algorithm to obtain a signature, so the success rate is $\frac{1}{t!}$ [13]. In the next section, a code-based digital signature scheme is proposed

which has a success rate of 1 and has less complexity than the CFS scheme.

3 | PROPOSED CODE-BASED DIGITAL SIGNATURE

The proposed code-based digital signature scheme has key generation, signing, and verification algorithms. A code-based cryptosystem is used in the public key infrastructure of this scheme. It uses a hash function and public keys to construct a signature. First, a document is hashed to compress its size to n bits where n is the length of the code used in the cryptosystem. The proposed code-based digital signature algorithms are as follows

- Key generation: $(pk, sk) \leftarrow Gen(\lambda)$ where λ denotes the key generation scheme.
- Document/message signing: $\sigma \leftarrow Sign(sk, pk, \boldsymbol{doc})$ where σ and $\boldsymbol{doc}$ denote the signature and document, respectively.
- Signature verification: $Ver(\sigma, pk, \boldsymbol{doc}) \in 0, 1$.

The following matrices are used in the proposed public key algorithm.

- G, a generator matrix of size $k \times n$.
- H, a parity check matrix of size $(n-k) \times n$.
- S, a non-singular scrambling matrix of size $k \times k$.
- P, a permutation matrix of size $n \times n$.
- L, a non-singular matrix of size $(n-k) \times (n-k)$.

The following subsection presents an algorithm to randomly construct an inverse matrix.

3.1 | Random inverse matrix construction

The parity check matrix has $n-k$ columns, each of which can have 2^k different values, so the number of matrices that satisfy $HH^{-1} = I_{n-k}$ is $2^{k\times(n-k)}$ [21]. The proposed construction of a random matrix H^{-1} is as follows.

H^{-1} can be divided into two parts, A_1 and A_2, where A_1 consists of rows 1 to k and A_2 consists of rows $k+1$ to n

$$H^{-1}_{n\times(n-k)} = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,(n-k)} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,(n-k)} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{k,1} & a_{k,2} & a_{k,3} & \cdots & a_{k,(n-k)} \\ --- & --- & --- & --- & --- \\ a_{(k+1),1} & a_{(k+1),2} & a_{(k+1),3} & \cdots & a_{(k+1),(n-k)} \\ a_{(k+2),1} & a_{(k+2),2} & a_{(k+2),3} & \cdots & a_{(k+2),(n-k)} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{n,1} & a_{n,2} & a_{n,3} & \cdots & a_{n,(n-k)} \end{pmatrix} = \begin{pmatrix} A_1 \\ - \\ A_2 \end{pmatrix}. \quad (5)$$

H^{-1} can be obtained by selecting a random A_1 and constructing the corresponding matrix A_2. The elements of A_2 are

$$A_2 = \begin{pmatrix} a_{(k+1),1} & a_{(k+1),2} & a_{(k+1),3} & \cdots & a_{(k+1),(n-k)} \\ a_{(k+2),1} & a_{(k+2),2} & a_{(k+2),3} & \cdots & a_{(k+2),(n-k)} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{n,1} & a_{n,2} & a_{n,3} & \cdots & a_{n,(n-k)} \end{pmatrix}, \quad (6)$$

where

$$a_{(k+b),d} = \sum_{i=1}^{k} p_{ib} a_{id}, \ b \neq d,$$

and

$$a_{(k+b),d} = 1 + \sum_{i=1}^{k} p_{ib} a_{id}, \ b = d.$$

In general, this can be expressed as

$$a_{(k+b),d} = 2^{|b-d|} \bmod 2 + \sum_{i=1}^{k} p_{ib} a_{id}. \quad (7)$$

For example, $a_{(k+1),1}$ in A_2 is given by

$$a_{(k+1),1} = 1 + p_{11}a_{11} + p_{21}a_{21} + \cdots + p_{k1}a_{k1}.$$

The construction of A_2 is then as follows. Let $B_1 = P^T_{(n-k)\times k}$ and $B_2 = I_{n-k}$, so

$$\begin{aligned} HH^{-1} &= (B_1|B_2) \times \begin{pmatrix} A_1 \\ - \\ A_2 \end{pmatrix} = I_{n-k}, \\ &= B_1A_1 + B_2A_2 = I_{n-k}. \end{aligned}$$

Then

$$A_2 = B_1A_1 + I_{n-k}, \quad (8)$$

which gives

$$\begin{aligned} HH^{-1} &= (B_1|B_2) \times \begin{pmatrix} A_1 \\ - \\ A_2 \end{pmatrix} \\ &= (B_1|B_2) \times \begin{pmatrix} A_1 \\ \hline B_1A_1 + I_{n-k} \end{pmatrix}, \\ &= B_1A_1 + B_2(B_1A_1 + I_{n-k}) \\ &= B_1A_1 + B_1A_1 + I_{n-k} = I_{n-k}. \end{aligned}$$

3.2 | Key generation

The proposed algorithm generates a public key (pk) and a private key (pr). The public key is shared within the network while the private key is kept secret.

Key Generation Algorithm $Gen(\lambda)$

1. Obtain a generator matrix G and corresponding parity matrix H for $C(n, k)$.
2. Select a random H^{-1} from the $2^{k\times(n-k)}$ choices using a random matrix $\mathcal{A}_1$ and constructing the corresponding matrix $\mathcal{A}_2$ $H^{-1} = \frac{\mathcal{A}_1}{\mathcal{A}_2}$.
3. As in the McEliece cryptosystem, use the generator matrix G, the scrambling matrix S, and the permutation matrix P to mask G $p_1 = G' = SGP$.
4. Use the non-singular random matrix L and P to mask H^{-1} $p_2 = L^{-1}(H^{-1})^T P$.
5. Verification of the digital signatures requires $p_3 = P^{-1}(H^{-1}H)^T P$.
6. Construct a parity check matrix corresponding to $G' = SGP$ $\mathcal{Q} = H'^T = P^{-1}H^T L$, $H' = L^T H(P^{-1})^T$.
7. Public key: $pk \leftarrow (p_1, p_2, p_3)$.
8. Private key: $pr(sk) \leftarrow (S^{-1}, P^{-1}, G, \mathcal{Q})$, where sk denotes the secret key.

Lemma 1. *The public key* $pk = (p_1, p_2, p_3)$ *satisfies the following*

$$(p_1)(p_3) = \mathbf{0} \tag{9}$$

$$(p_2)(p_3) = p_2 \tag{10}$$

$$(p_3)(p_3) = p_3 \tag{11}$$

Proof. For Equation (9)

$$\begin{aligned}(p_1)(p_3) &= (SGP)(P^{-1}(H^{-1}H)^T P)\\ &= SG(H^{-1}H)^T P\\ &= S(GH^T)(H^{-1})^T P\\ &= \mathbf{0}\end{aligned}$$

For Equation (10)

$$\begin{aligned}(p_2)(p_3) &= (L^{-1}(H^{-1})^T P)(P^{-1}(H^{-1}H)^T P)\\ &= L^{-1}(H^{-1})^T(H^{-1}H)^T P\\ &= L^{-1}(H^{-1}HH^{-1})^T P\\ &= L^{-1}(H^{-1})^T P\\ &= p_2\end{aligned}$$

For Equation (11)

$$\begin{aligned}(p_3)(p_3) &= (P^{-1}(H^{-1}H)^T P)(P^{-1}(H^{-1}H)^T P)\\ &= P^{-1}(H^{-1}H)^T(H^{-1}H)^T P\\ &= P^{-1}(H^{-1}HH^{-1}H)^T P\\ &= P^{-1}(H^{-1}H)^T P\\ &= p_3\end{aligned}$$

□

The following lemma provides the relationship between the private and public keys.

Lemma 2. *The public key* $pk = (p_1, p_2, p_3)$ *and the secret key* $(\mathcal{Q})$ *are related as follows*

$$(p_1)(\mathcal{Q}) = \mathbf{0} \tag{12}$$

$$(p_2)(\mathcal{Q}) = \boldsymbol{I} \tag{13}$$

$$(p_3)(\mathcal{Q}) = \mathcal{Q} \tag{14}$$

$$(\mathcal{Q})(p_2) = p_3 \tag{15}$$

Proof. For Equation (12)

$$\begin{aligned}(p_1)(\mathcal{Q}) &= (SGP)(P^{-1}H^T L)\\ &= S(GH^T)L\\ &= \mathbf{0}\end{aligned}$$

For Equation (13)

$$\begin{aligned}(p_2)(\mathcal{Q}) &= (L^{-1}(H^{-1})^T P)(P^{-1}H^T L)\\ &= L^{-1}(H^{-1})^T H^T L\\ &= L^{-1}(HH^{-1})^T L\\ &= \boldsymbol{I}\end{aligned}$$

For Equation (14)

$$\begin{aligned}(p_3)(\mathcal{Q}) &= (P^{-1}(H^{-1}H)^T P)(P^{-1}H^T L)\\ &= P^{-1}(H^{-1}H)^T H^T L\\ &= P^{-1}(HH^{-1}H)^T L\\ &= P^{-1}(H)^T L\\ &= \mathcal{Q}\end{aligned}$$

For Equation (15)

$$\begin{aligned}(\mathcal{Q})(p_2) &= (P^{-1}H^T L)(L^{-1}(H^{-1})^T P)\\ &= P^{-1}H^T(H^{-1})^T P\\ &= P^{-1}(H^{-1}H)^T P\\ &= p_3\end{aligned}$$

□

3.3 | Signing algorithm

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

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

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 $h(\boldsymbol{doc})$.
2. Let $\boldsymbol{s}$ denote an $n-k$ bit vector such that
   $\boldsymbol{s} \leftarrow h(\boldsymbol{doc})(Q)$.
3. Construct a codeword $\boldsymbol{c}$ using $h(\boldsymbol{doc})$ and $\boldsymbol{s}$
   $\boldsymbol{sig}SGP \leftarrow h(\boldsymbol{doc}) + \boldsymbol{s}(p_2)$.
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 the private key sk to construct the $n-k$ bit vector $\boldsymbol{d}$
   $\boldsymbol{d} \leftarrow h(h(\boldsymbol{doc}))(Q) + \boldsymbol{s}$.
6. Output $\sigma = (\boldsymbol{sig}, \boldsymbol{d})$ and send $(\boldsymbol{sig}, \boldsymbol{d})$ and the document $\boldsymbol{doc}$ to the receiver for signature verification.

Theorem 1. *$h(\boldsymbol{doc}) + \boldsymbol{s}(p_2)$ is a valid codeword of the code $C(n, k)$ with generator matrix $G' = SGP$.*

Proof. Matrices S and P have full rank as they are non-singular matrices. Therefore, the rank of SGP is k, and the rank of $P^{-1}H^TL$ is $n-k$. Further, the row vectors of SGP and the column vectors of $P^{-1}H^TL$ are orthogonal. Therefore, $P^{-1}H^TL$ generates the nullspace of the space spanned by SGP. Hence, the transpose of $P^{-1}H^TL$ is a parity check matrix for the code generated by SGP.

For a codeword $\boldsymbol{c} \in C(n, k)$, $\boldsymbol{c}H'^T = \boldsymbol{0}$ where $G' = SGP = p_1$ is the generator matrix and $H'^T = P^{-1}H^TL = Q$ is the parity check matrix

$$\boldsymbol{c} = \boldsymbol{sig}SGP = h(\boldsymbol{doc}) + \boldsymbol{s}(p_2).$$

The vector $\boldsymbol{s}$ is equal to $h(\boldsymbol{doc})(Q)$ so

$$\begin{aligned} \boldsymbol{sig}SGP &= h(\boldsymbol{doc}) + h(\boldsymbol{doc})(Q)(p_2) \\ \boldsymbol{sig}SGP &= h(\boldsymbol{doc}) + h(\boldsymbol{doc})(p_3). \end{aligned}$$

Therefore, $\boldsymbol{c}H'^T = (\boldsymbol{sig}SGP)(Q)$ and

$$\begin{aligned} \boldsymbol{c}H'^T &= h(\boldsymbol{doc})(Q) + h(\boldsymbol{doc})(p_3)(Q) \\ &= h(\boldsymbol{doc})(Q) + h(\boldsymbol{doc})(Q) \\ &= \boldsymbol{0}. \end{aligned}$$

□

3.4 | 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 $h(\cdot)$ to hash the received document to construct $h(\boldsymbol{doc})$ and $h(h(\boldsymbol{doc}))$, and assign
   $a \leftarrow \boldsymbol{sig}SGP$.
2. Use the public key (p_2, p_3) and $\boldsymbol{d}$ to compute $v_1 = \boldsymbol{s}(p_2)$ which is an n bit vector
   $v_1 \leftarrow \boldsymbol{s}(p_2) = h(h(\boldsymbol{doc}))(p_3) + \boldsymbol{d}(p_2)$.
3. Use the public key (p_3) to compute $v_2 = \boldsymbol{s}(p_2)$ which is an n bit vector
   $v_2 \leftarrow \boldsymbol{s}(p_2) = h(\boldsymbol{doc})(p_3)$.
4. If the condition
   $$v_1 = v_2,$$
   is not true, verification fails.
5. Use $v_1 = \boldsymbol{s}(p_2)$ and $h(\boldsymbol{doc})$ to compute
   $\boldsymbol{c} \leftarrow h(\boldsymbol{doc}) + \boldsymbol{s}(p_2)$.
6. Verification is successful if
   $$a = \boldsymbol{c},$$
   otherwise it fails.

3.5 | Digital signature example

Let $n = 12$ and $k = 7$ be the parameters of the code $C(n, k)$ with

$$G_{k\times n} = \left(\begin{array}{c|cccccc} & 0 & 1 & 1 & 1 & 0 \\ & 0 & 0 & 0 & 1 & 1 \\ & 0 & 1 & 0 & 0 & 1 \\ I_k & 0 & 0 & 1 & 1 & 0 \\ & 0 & 1 & 0 & 1 & 0 \\ & 1 & 0 & 0 & 1 & 0 \\ & 1 & 0 & 1 & 0 & 0 \end{array}\right),$$

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

Randomly select a matrix A_1 of size $k \times (n-k)$ and construct the corresponding matrix A_2 of size $(n-k) \times (n-k)$ to obtain

the random inverse parity check matrix

$$H^{-1} = \begin{pmatrix} A_1 \\ - \\ A_2 \end{pmatrix}, A_1 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix},$$

$$A_2 = B_1 A_1 + I_{n-k} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

The matrices S, L, and P are

$$S_{k\times k} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix},$$

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

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

1. Alice hashes a document $\boldsymbol{doc}$ using the hash function $h(\cdot)$ to obtain $h(\boldsymbol{doc})$ of length n bits
   $h(\boldsymbol{doc}) = 100011010010$ and $h(h(\boldsymbol{doc})) = 101001000101$.
2. Construct an $n-k$ bit vector $\boldsymbol{s}$ such that $\boldsymbol{s} = h(\boldsymbol{doc})(Q)$
   $\boldsymbol{s} = 11110$.
3. Construct a codeword $h(\boldsymbol{doc}) + \boldsymbol{s}(p_2)$ of the code $C(n, k)$

$$\begin{aligned} \boldsymbol{c} &= \boldsymbol{sig}SGP = h(\boldsymbol{doc}) + \boldsymbol{s}(p_2) \\ &= 100011010010 + (11110)(p_2) \\ &= 100011010010 + 011111011001 \\ &= 111100001011. \end{aligned}$$

4. Decodes the codeword to obtain $\boldsymbol{sig} = 0000110$.
5. Decode $\boldsymbol{d}$ using sk and $\boldsymbol{s}$

$$\begin{aligned} \boldsymbol{d} &= h(h(\boldsymbol{doc}))(Q) + \boldsymbol{s} \\ &= (101001000101)(Q) + 11110 \\ &= 11011 + 11110 \\ &= 00101. \end{aligned}$$

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

Bob verifies the signature as follows.

1. Use the hash function $h(\cdot)$ to hash the received document to construct $h(\boldsymbol{doc})$ and $h(h(\boldsymbol{doc}))$, and assign $a = \boldsymbol{sig}SGP$.
   $h(\boldsymbol{doc}) = 100011010010$
   $h(h(\boldsymbol{doc})) = 101001000101$
   $a = \boldsymbol{sig}SGP = (0000110)(p_1)$
   $a = 111100001011$.
2. Use Alice's public key (p_2, p_3) and $\boldsymbol{d}$ to compute

$$\begin{aligned} v_1 &= h(h(\boldsymbol{doc}))(p_3) + \boldsymbol{d}(p_2) \\ &= 101001000101(p_3) + 00101(p_2) \\ &= 101001000101 + 110110011100 \\ &= 011111011001. \end{aligned}$$

3. Use Alice's public key (p_3) to compute

$$\begin{aligned} \boldsymbol{s}(p_2) &= h(\boldsymbol{doc})(p_3) \\ &= 100011010010(p_3) \\ &= 011111011001. \end{aligned}$$

4. Bob checks the condition $v_1 = v_2$. If it is not true, verification fails.
5. Having $v_1 = \boldsymbol{s}(p_2)$ and given $h(\boldsymbol{doc})$, Bob computes

$$\begin{aligned} \boldsymbol{c} &= h(\boldsymbol{doc}) + \boldsymbol{s}(p_2) \\ &= 100011010010 + 011111011001 \\ &= 111100001011. \end{aligned}$$

6. Verification is successful if $a = \boldsymbol{c}$.

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

4 | PERFORMANCE ANALYSIS

As mentioned above, the computation time and low success rate are the main obstacles to the widespread use of code-based signatures. On average, the CFS code-based signature algorithm will have to be executed $t!$ times to obtain a valid signature.

TABLE 1 Code-based signature success rate and complexity.

Scheme	Error correction capability (t)	Success rate	Signature generation complexity	Verification complexity
Proposed	Any	1	$O(n^2)$	$O(n^2)$ (with integrity check)
CFS	50	$(50!)^{-1}$	$O(t! \times n^2)$	$O(n^2)$
Modified CFS	10	$(10!)^{-1}$	$O(t! \times n^2)$	$O(n^2)$

Abbreviation: CFS, Courtois–Finiasz–Sendrier.

TABLE 2 Signature length and public key size comparison (bytes).

Signature scheme	Success rate	Public key size	Signature size
Bliss-I [23]	0.63	2048	5734
Bliss-II [23]	0.14	2048	5120
Bliss-III [23]	0.36	3072	6144
Bliss-IV [23]	0.19	3072	6656
qTeslaIII [24, 25]	1	3103	2908
Dilithium [25] 1024 × 768	1	1188	2098
Falcon 1024 [24, 25]	1	1792	1260
Uniform 1024 [24, 25]	1	2099	2099
Proposed algorithm $n = 256$	1	16384	32
Proposed algorithm $n = 512$	1	32768	64

Modified CFS schemes have been developed with smaller values of t to reduce $t!$ [22]. Table 1 compares the success rate of several code-based signature schemes and the proposed scheme in terms of signature generation and verification complexity. The approximate number of operations for signing and verification with the proposed scheme is $3n^2$, including the integrity check.

The main competition for the proposed scheme is lattice-based signatures which are considered quantum secure. However, the signatures obtained are large [24]. The proposed code-based scheme can reduce the signature length and the computation time for signing. Table 2 compares the signature length of several lattice-based signature schemes and the proposed scheme [23–25]. This shows that the proposed code-based scheme has smaller signatures and requires less computation time for signing. The size of the signature is a main factor in the choice of a digital signature scheme. Further, the proposed algorithm generates signatures of lengths 32 and 64 bytes compared to Bliss-IV with a signature of length 6556 bytes and a low success rate of 0.19 [23], and qTesla-III with a signature of length 2848 bytes [24, 25]. Speed is a critical factor in many digital signature applications such as online banking, e-commerce, and blockchains, for example, Bitcoin and Ethereum.

5 | SECURITY ANALYSIS

This section presents an analysis of the correctness, integrity, and probability of attack success against the proposed scheme. To protect the integrity of the transmitted signature, the verification algorithm should detect if the signature has been changed after generation or if a different secret key has been used to construct the signature. The proposed verification algorithm constructs v_1 and v_2 as follows

$$v_1 \leftarrow \boldsymbol{s}(p_2) = h(h(\boldsymbol{doc}))(p_3) + \boldsymbol{d}(p_2)$$

$$\begin{aligned} \boldsymbol{d} &= h(h(\boldsymbol{doc}))(Q) + \boldsymbol{s} \\ \boldsymbol{d}(p_2) &= (h(h(\boldsymbol{doc}))(Q) + \boldsymbol{s})(p_2) \\ \boldsymbol{d}(p_2) &= h(h(\boldsymbol{doc}))(Q)(p_2) + \boldsymbol{s}(p_2). \end{aligned}$$

From Equation (15), $(Q)(p_2) = p_3$, therefore

$$v_1 = \boldsymbol{s}(p_2) = h(h(\boldsymbol{doc}))(p_3) + \boldsymbol{d}(p_2). \tag{16}$$

$$v_2 \leftarrow \boldsymbol{s}(p_2) = h(\boldsymbol{doc})(p_3)$$

$$\begin{aligned} \boldsymbol{sig}SGP &= h(\boldsymbol{doc}) + \boldsymbol{s}(p_2) \\ \boldsymbol{s}(p_2) &= \boldsymbol{sig}(p_1) + h(\boldsymbol{doc}) \\ \boldsymbol{s}(p_2)(p_3) &= \boldsymbol{sig}(p_1)(p_3) + h(\boldsymbol{doc})(p_3). \end{aligned}$$

From Equations (9) and (10), $(p_1)(p_3) = \mathbf{0}$ and $(p_2)(p_3) = p_2$, so

$$v_2 = \boldsymbol{s}(p_2) = h(\boldsymbol{doc})(p_3). \tag{17}$$

Thus, the same value of $\boldsymbol{s}(p_2)$ is obtained using two different approaches. In addition, v_1 does not depend on the signature $\boldsymbol{sig}$ and v_2 does not depend on the secret key although the integrity condition is met when $v_1 = v_2$. If an adversary uses a private key other than the correct one or forges $\boldsymbol{sig}$, then the condition $v_1 = v_2$ will not be met so verification fails. Hence, the integrity of the signature is ensured by the condition $v_1 = v_2$.

There are several attacks that an adversary can use in an attempt to break a signature scheme [3]. Public key and forgery attacks are common attacks against digital signatures. A public key attack (called a structural attack), is typically more successful than a forgery attack. In this attack, an adversary tries to discover the secret key, so the security of the public key is critical. To increase the security of the public key, the proposed algorithm masks the generator and parity check matrices. Encrypted vectors $\boldsymbol{s}$ and $\boldsymbol{d}$ are used which increases the security. Therefore an adversary cannot break the scheme and forge a signature that can be verified through generic, directed, or adaptive chosen-message attacks [28].

The unforgeability attack is as follows [26].

1. The challenger uses the given code $C(k, n)$ to generate a public key (pk) and secret key (sk), and then shares (pk) with an adversary.

2. The adversary (a polynomial-time probabilistic machine), provides chosen messages (documents) $(doc_1, \dots, doc_q)$, and the challenger provides valid signatures $(\sigma_1, \dots, \sigma_q)$ in response.
3. The challenger provides algorithm access to the adversary.
4. The adversary forges a new message and a signature (doc^*, σ^*) and sends it to the challenger for verification. Note that (doc^*) does not belong to the previously chosen messages $(doc_1, \dots, doc_q)$.
5. The adversary wins if the verification is successful $V(doc^*, \sigma^*) = 1$.

An algorithm is secure when the probability of success is sufficiently low (negligible)

$$Pr[(\mathcal{A}dv, \gamma) = 1] < \epsilon(\gamma),$$

where $\mathcal{A}dv$ denotes the adversary and γ denotes the security parameter [26, 27].

Consider the steps above. The adversary uses its secret key (sk_2) to sign a document and outputs $(\boldsymbol{sig}, \boldsymbol{d})$ to be verified by the challenger. The challenger uses the verification algorithm and reaches step 4

$$v_1 = v_2,$$

$$h(\boldsymbol{doc})(p_3) = h(h(\boldsymbol{doc}))(p_3) + \boldsymbol{d}(p_2).$$

The left side $(h(\boldsymbol{doc})(p_3))$ is independent of the adversary's secret key (sk_2) while $\boldsymbol{d}$ on the right side was constructed using the adversary's secret key during the signing process. Then

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

and therefore

$$h(\boldsymbol{doc})(p_3) = h(h(\boldsymbol{doc}))(p_3) + (h(h(\boldsymbol{doc}))(sk_2) + h(\boldsymbol{doc})(sk_2))(p_2)$$

$$(h(\boldsymbol{doc}) + h(h(\boldsymbol{doc})))(p_3) = (h(h(\boldsymbol{doc})) + h(\boldsymbol{doc}))(sk_2)(p_2)$$

$$(p_3) = (sk_2)(p_2).$$

From Equation (16) in Lemma 2, this is true if and only if $(sk_2) = (sk)$.

Suppose an adversary selects $(L^{-1}(\mathcal{A}^{-1})^T P)^{-1}$ as the secret key. Then

$$\begin{aligned}(sk_2)(p_2) &= (L^{-1}(\mathcal{A}^{-1})^T P)^{-1}(L^{-1}(H^{-1})^T P)\\ &= (P^{-1}((\mathcal{A}^{-1})^{-1})^T L)(L^{-1}(H^{-1})^T P)\\ &= P^{-1}(H^{-1}((\mathcal{A}^{-1})^{-1}))^T P,\end{aligned}$$

so if $(\mathcal{A}^{-1})^{-1} = \mathcal{A} = H$, $(sk_2)(p_2)$ would be equal to (p_3) and the condition for the signed document is satisfied. In this case, the adversary can forge a signature with respect to the given public key and so is the winner.

$$Pr[(\mathcal{A}dv, \gamma) = 1 \leftarrow Pr[sk_2 = sk] \leftarrow Pr[(\mathcal{A}^{-1})^{-1} = \mathcal{A}]] < \epsilon(\gamma)$$

The matrix L is a square matrix so the inverse L^{-1} has the same size. However, the parity check matrix H is a full rank non-square matrix of size $(n - k) \times n$, so $\mathcal{A}$ should be a full rank matrix with the same dimensions. As noted in Section 3.1, the inverse of H is not unique. Hence, the probability that $(H^{-1})^{-1} = H$ is negligible, so the proposed scheme is secure against structural attacks [3].

Theorem 2. *The inverse of the matrix $L^{-1}(\mathcal{A}^{-1})^T P$ is not unique and the probability of constructing a particular inverse of $L^{-1}(\mathcal{A}^{-1})^T P$ is negligible.*

Proof. The matrix $(\mathcal{A}^{-1})^T$ has full rank and size $(n - k) \times n$. Therefore, the public key $L^{-1}(\mathcal{A}^{-1})^T P$ is a full rank matrix, and its inverse has $n - k$ columns, each of which can have 2^k different values. The number of valid inverse matrices is then $2^{k\times(n-k)}$ [21]. Hence, the probability of constructing a particular inverse of the public key $L^{-1}(\mathcal{A}^{-1})^T P$ is equal to $\frac{1}{2^{k\times(n-k)}}$ which is negligible for reasonable values of n and k. □

As a result of Theorem 2, the probability of an adversary constructing a secret key using the public key is $2^{-(k\times(n-k))}$. Therefore, the probability of an adversary signing a document that can be verified is negligible

$$Pr[(\mathcal{A}dv, \gamma) = 1] < \frac{1}{2^{k\times(n-k)}}.$$

Hence, the probability of the adversary forging the signature by accessing the algorithm is also negligible, so the proposed algorithm is secure.

6 | CONCLUSION

The CFS digital signature scheme was examined and its drawbacks described. An approach to overcoming these drawbacks was proposed. It was also shown that code-based digital signature schemes require significant computation time because the cipher texts are only part of the vector space. In particular, the CFS scheme requires $t!$ executions on average, where t is the error correction capability of the code, to obtain a valid signature. The proposed code-based digital signature scheme provides a practical solution to these drawbacks. It was shown to be secure in that an adversary cannot forge a signature that can be verified and it is secure against a structural public key attack. Further, the probability of constructing the secret key from the public key was shown to be negligible. Moreover, results were presented which show that the proposed scheme requires less computation time for signing than other code-based digital signature schemes.

AUTHOR CONTRIBUTIONS

Farshid Haidary Makoui: Conceptualization, data curation, formal analysis, investigation, methodology, resources, software, validation, visualization, writing - original draft. Thomas Aaron Gulliver: Conceptualization, data curation, formal

analysis, funding acquisition, project administration, resources, software, supervision, validation, writing - review and editing. Mohammad Dakhilalian: Conceptualization, data curation, formal analysis, project administration, resources, software, supervision, validation, writing - review and editing.

CONFLICT OF INTEREST

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

There is no data associated with this research.

REFERENCES

1. Shor, P.W.: Algorithms for quantum computation: Discrete logarithms and factoring. In: Proceedings of Annual Symposium on Foundations of Computer Science, pp. 124–134. IEEE, Piscataway, NJ (1994)
2. Baldi, M.: Post-quantum cryptographic schemes based on codes. In: Proceedings of the International Conference on High Performance Computing & Simulation, pp. 908–910. IEEE, Piscataway, NJ (2017)
3. Rao, T.N.R., Nam, K.H.: A private key algebraic coded cryptosystem. In: Odlyzko, A.M. (ed.) Advances in Cryptology— CRYPTO' 86. Lecture Notes in Computer Science, vol. 263, pp. 35–48. Springer, Berlin (1986)
4. Hooshmand, R., Aref, M.R.: Efficient secure channel coding scheme based on low-density lattice codes. IET Commun. 10(11), 1365–1373 (2016)
5. Rao, T.N.R.: Joint encryption and error correction schemes. In: Proceedings of the Annual International Symposium on Computer Architecture, pp. 240–241. Association for Computing Machinery, New York, NY (1984)
6. Sendrier, N.: Code based cryptography: State of art and perspectives. IEEE Secur. Privacy 15(4), 44–50 (2017)
7. McEliece, R.J.: A public-key cryptosystem based on algebraic coding theory. Jet Propulsion Lab. DSN Tech. Rep. 42–44, pp. 114–116 (1978)
8. Cayrel, P.L., Meziani, M.: Post-quantum cryptography: Code-based signatures. In: Kim, Th., Adeli, H. (eds.), Advances in Computer Science and Information Technology. Lecture Notes in Computer Science, vol. 6059, pp. 82–99. Springer, Berlin (2010)
9. Nojima, R., Imai, H., Kobara, K., Morozov, K.: Semantic security for the McEliece cryptosystem without random oracles. Des. Codes Cryptogr. 49(1-3), 289–305 (2008)
10. Cayrel, P.L., Gaborit, P., Girault, M.: Identity based identification and signature schemes using correcting codes. In: Proceedings of International Workshop on Coding and Cryptograph, pp. 69–78. IOS Press, Amsterdam (2007)
11. Berlekamp, E.R., McEliece, R.J., Van Tilborg, H.C.A.: On the inherent intractability of certain coding problems. IEEE Trans. Inf. Theory 24(3), 384–386 (1978)
12. Niederreiter, H.: Knapsack-type cryptosystems and algebraic coding theory. Probl. Control Inf. Theory 15(2), 159–166 (1986)
13. Courtois, N., Finiasz, M., Sendrier, N.: How to achieve a McEliece-based digital signature scheme. In: Boyd, C. (ed.) Advances in Cryptology—ASIACRYPT 2001. Lecture Notes in Computer Science, vol. 2248, pp. 157–174. Springer, Berlin (2001)
14. Xu, S.B., Doumen, J.M., Van Tilborg, H.C.A.: On the security of digital signature scheme based on error correcting codes. Des. Codes Cryptogr. 28(2), 187–199 (2003)
15. Wang, Z., Sun, W.: Review of web authentication. J. Phys.: Conf. Ser. 1646, 012009 (2020)
16. Boyd, C., Mathuria, A., Stebila, D.: Protocols for Authentication and Key Establishment, 2nd ed. Springer, Berlin (2019)
17. Tycksen, F.A., Jennings, C.W.: Digital certificate. U.S. Patent. 6,189,097, 12 February 2001
18. Zavalishina, E., Krendelev, S., Volkov, E., Permiashkin, D., Gridin, D.: Public key and digital signature for blockchain technology based on the complexity of solving a system of polynomial equations. In: Arai, K., Kapoor, S., Bhatia, R. (eds.) Intelligent Systems and Applications. Advances in Intelligent Systems and Computing, vol. 868, pp. 1251–1258. Springer, Cham (2018)
19. Zhang, W., Wu, Q., Qin, B., Han, T., Zhang, Y., Chen, X., Li, N.: TTP-free fair exchange of digital signatures with Bitcoin. In: Liu, J., Samarati, P. (eds.) Information Security Practice and Experience. Lecture Notes in Computer Science, vol. 10701, pp. 62–81. Springer, Cham (2017)
20. Xinmei, W.: Digital signature scheme based on error correcting codes. Electron. Lett. 26(13), 898–899 (1990)
21. Esmaeili, M.: Application of linear block codes in cryptography. Dissertation, University of Victoria (2019)
22. Finiasz, M.: Parallel-CFS: Strengthening the CFS McEliece-based signature scheme. In: Proceedings of the International Conference on Selected Areas in Cryptography, pp. 159–170. Springer, Berlin (2011)
23. Pöppelmann, T., Ducas, L., Güneysu, T.: Enhanced lattice-based signatures on reconfigurable hardware. In: Batina, L., Robshaw, M., (eds.) Cryptographic Hardware and Embedded Systems – CHES 2014, Lecture Notes in Computer Science, vol. 8731, pp. 353–370. Springer, Berlin, Heidelberg (2014)
24. Howe, J., Pöppelmann, T., O'Neill, M., O'Sullivan, E., Güneysu, T.: Practical lattice-based digital signature schemes. ACM Trans. Embedded Comput. Syst. 14(3), 1–24 (2015)
25. Das, D., Hoffstein, J., Pipher, J., Whyte, W., Zhang, Z.: Modular lattice signatures, revisited. Des. Codes Cryptogr. 88(3), 505–532 (2019)
26. Diemert, D., Gellert, K., Jager, T., Lyu, L.: More efficient digital signatures with tight multi-user security. In: Proceedings Public-Key Cryptography. Lecture Notes in Computer Science, vol. 12711, pp. 1–31, Springer, Berlin (2021)
27. Bellare, M., Namprempre, C., Neven, G.: Security proofs for identity-based identification and signature schemes. J. Cryptol. 22(1), 1–61 (2009)
28. Goldwasser, S., Micali, S., Rivest, R.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM J. Comput. 17(2), 281–308 (1988)

How to cite this article: Haidary Makoui, F., Gulliver, T.A., Dakhilalian, M.: A new code-based digital signature based on the McEliece cryptosystem. IET Commun. 17, 1199–1207 (2023). https://doi.org/10.1049/cmu2.12607