POLITECNICO DI TORINO
Repository ISTITUZIONALE

On Two Modifications of the McEliece PKE and the CFS Signature Scheme

Original
On Two Modifications of the McEliece PKE and the CFS Signature Scheme / D'Alconzo, G.. - In: INTERNATIONAL JOURNAL OF FOUNDATIONS OF COMPUTER SCIENCE. - ISSN 0129-0541. - 35:5(2024), pp. 501-512. [10.1142/S0129054123500132]

Availability:
This version is available at: 11583/2979617 since: 2023-07-05T13:11:39Z

Publisher:
World Scientific Publishing

Published
DOI:10.1142/S0129054123500132

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
World Scientific postprint/Author's Accepted Manuscript

Electronic version of an article published as INTERNATIONAL JOURNAL OF FOUNDATIONS OF COMPUTER SCIENCE, 35, 5, 2024, pp. 501-512 https://dx.doi.org/10.1142/S0129054123500132 © World Scientific Publishing Company https://www.worldscientific.com/doi/epdf/10.1142/S0129054123500132.

(Article begins on next page)

25 November 2026

On Two Modifications of the McEliece PKE and the CFS Signature Scheme

Giuseppe D'Alconzo
giuseppe.dalconzo@polito.it
Department of Mathematical Sciences, Politecnico di Torino

Abstract

This paper analyzes two schemes from a 2019 work by Liu, Yang, Han and Wang, namely, adapting the McEliece cryptosystem to obtain a new public-key encryption scheme, and designing an efficient CFS-like digital signature. It is shown that the new encryption scheme based on McEliece, even if it has longer public keys, is not more secure than the standard one. The security gap between the original scheme and its modification is presented. Moreover, while the proposed parameters for the digital signature scheme lead to a significant performance improvement, however, they introduce a vulnerability in the protocol. A key-forgery attack using the Support Splitting Algorithm as a subroutine is described, with a study of the computational cost of retrieving the secret key from the public one.

Keywords— Code-Based Signatures, CFS, McEliece

1 Introduction

Post-quantum cryptography. With the emergence of quantum computation, there is the urge of replacing cryptosystems used nowadays, based on the Factorization Problem and the Discrete Logarithm Problem, with quantum-resistant alternatives. Because of this, in 2016 the National Institute of Technology and Security (NIST) started a call to evaluate and standardize quantum-resistant cryptosystems[1]. There are two categories of primitives under evaluation: Key-Encapsulation Mechanisms (KEM) and Digital Signatures. In 2022, the first post-quantum algorithms have been chosen for standardization [AAC+22]: CRYSTALS-KYBER [BDK+18] in the KEM category and CRYSTALS-Dilithium [DKL+18], FALCON [FHK+18] and SPHINCS+ [BHK+19] in the Digital Signature one. However, the evaluation of three code-based KEMs (BIKE [ABB+17], Classic McEliece [BCL+17]

[1]NIST Post-Quantum Standardization process webpage: `https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization`, Accessed: 2023-02-14

and HQC [MAB+18]) is still ongoing. A new call for digital signature schemes started in 2023[2].

Code-based signatures. As we have seen above, three code-based Key-Encapsulation Mechanisms are still under evaluation for standardization. Unfortunately, the situation for signatures is very different: standardized algorithms are both lattice-based or hash-based and there were no code-based schemes among the last round proposals. Previous trapdoor schemes are either not secure, for example KKS [KKS97], or impractical, like CFS [CFS01]. In particular, a big effort has been done to improve the performance of CFS-like schemes, leading to dangerous vulnerabilities, like in [DMP21; BHLP17; ALP+18]. However, new constructions are emerging, for example Wave [DST19], which uses a new approach for the decoding problem, and LESS [BBPS21], based on the hardness of the Code Equivalence Problem. Other proposals are based on LDPC (Low-Density Parity-Check) codes [BBC+13] or codes in the rank metric [ABG+19; LT20]. The Zero-Knowledge proofs approach and the "MPC-in-the-head" paradigm are very promising [FJR22; FJR23; GPS22] and they seem to be the new frontier of code-based signatures.

Our contribution. In this work, we cryptanalyze the two schemes presented in [LYHW19]: the first one is a public-key cryptosystem, a modification of McEliece, that we call Modified McEliece (MME). The second scheme is a CFS-like digital signature using the MME scheme, denoted with LYHW19. Our main results are two. First, we show that MME has the same security, up to a polynomial factor, of the original McEliece, but the former has worse performance. In particular, we show that the adoption of two public matrices in MME does not add any additional security layer with respect to McEliece. Our second result concerns the security of the LYHW19 signature scheme, claimed, in [LYHW19], to be more efficient than the CFS one [CFS01]. We show that, for the proposed parameters, this scheme can be broken in practical times by the fact that the cardinality of a certain class of polynomials, and consequently the secret keys space, is too small. If larger parameters are used to establish security, there is a drop in performance.
This work is organized as follows: after recalling some preliminaries in Section 2, in Section 3 we present the public-key encryption scheme MME and we analyze its security with respect to textbook McEliece cryptosystem. In Section 4 we recall the digital signature LYHW19 and we show how to recover, using the proposed parameters, the secret key from the public key.

2 Preliminaries

Notation. Let $\mathbb{N}$ and $\mathbb{R}_{\geq 0}$ be the sets of natural and non-negative real numbers, respectively. We denote with λ the security parameter. A function $\epsilon : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is *negligible* if there exists n_0 such that for every $n > n_0$ we have $\epsilon(n) \leq 1/p(n)$ for every polynomial p. A function not having this propriety is called *non-negligible*. A probability is *overwhelming* if it is equal to $1-\epsilon$, where ϵ is a negligible function. For a prime power q, $\mathbb{F}_q$ is the finite field with q elements, and $(\mathbb{F}_q)^n$ is the n-dimensional

[2]Post-Quantum Cryptography: Digital Signature Schemes `https://csrc.nist.gov/Projects/pqc-dig-sig`, Accessed: 2023-02-14

vector space over $\mathbb{F}_q$. The *Hamming weight* of a vector x is the number of its non-zero coordinates, and it is denoted by $\mathrm{w}(x)$. With $||$ we denote the concatenation of strings or vectors. The probability of an event A is denoted with $\mathbf{P}[A]$ and, in the rest of the article, the "big-O" notation $\mathcal{O}$ is used.

Linear codes and Goppa codes. An $[n, k, d]$ *linear code* $\mathcal{C}$ over $\mathbb{F}_q$ is a k-dimensional vector subspace of $(\mathbb{F}_q)^n$ such that $d = \min_{x \in \mathcal{C} \setminus \{0\}} \{\mathrm{w}(x)\}$. The parameters n, k and d are respectively called the *length*, the *dimension* and the *minimum distance* of the code $\mathcal{C}$. Given a basis $\mathcal{B}$ of $\mathcal{C}$, a *generator matrix* for $\mathcal{C}$ is a $k \times n$ matrix with coefficients in $\mathbb{F}_q$ having elements of $\mathcal{B}$ as rows. The *error correction capability* of a code is given by $t = \left\lfloor \frac{d-1}{2} \right\rfloor$. Given a vector c in $\mathcal{C}$ and a vector e in $(\mathbb{F}_q)^n$ of weight at most t, the *decoding* of a given $y = c + e$ is the procedure of recovering c and e. For the rest of this work, we use the convention that given a decoding algorithm $\mathbf{D}_{\mathcal{C}}$ and a vector $y = c + e$, the decoding of y is given by c in $\mathcal{C}$. For a random code, the decoding is a hard problem [BMV78], but there exist families of codes having efficient decoding algorithms.

For cryptographic constructions, a relevant family of efficiently decodable linear codes is given by *binary irreducible Goppa codes* [Ber73]. A binary irreducible Goppa code is a code over $\mathbb{F}_{2^m}$ defined by a monic irreducible polynomial $g(X)$ in $\mathbb{F}_{2^m}[X]$ of degree t and by an ordered set $L = \{\alpha_1, \ldots, \alpha_n\}$ in $\mathbb{F}_{2^m}$ called *support*. Usually, we set $n = 2^m$, that is, the support is the whole field $\mathbb{F}_{2^m}$. The Goppa code $\Gamma(g, L)$ is an $[n, n - mt, d \geq 2t + 1]$ linear code over $\mathbb{F}_{2^m}$. Given the polynomial $g(X)$ and the support L, there is a polynomial time decoding algorithm $D_{\Gamma(g,L)}$. For this work we do not need more details on the construction of $\Gamma(g, L)$ or on the decoding algorithm $D_{\Gamma(g,L)}$; the interested reader can see [Ber73; Pat75; MS77].

Code equivalence. Given two $[n, k, d]$ linear codes $\mathcal{C}$ and $\mathcal{C}'$ generated by G and G' respectively, they are *permutation equivalent* if there exist an invertible $k \times k$ matrix S and a $n \times n$ permutation matrix P such that $G' = SGP$. The problem of deciding if two codes are permutation equivalent is called *Permutation Code Equivalence Problem*, and it is shown to be harder than *Graph Isomorphism* [PR97]. However, the Support Splitting Algorithm [Sen00], on input two codes, returns, if any, the permutation between them. For a larger class of codes, and most notably on random codes, the Support Splitting Algorithm runs in polynomial time.

Public-key encryption schemes. A *public-key encryption scheme* is a tuple of polynomial-time algorithms $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$: KGen takes as input a security parameter λ in unary and returns a pair of public-secret keys, while Enc and Dec are the encryption and decryption algorithms. We want that for every λ, given $(\mathsf{pk}, \mathsf{sk}) = \mathsf{KGen}(1^\lambda)$, then $\mathsf{Dec}(\mathsf{pk}, \mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) = m$ holds with overwhelming probability for every suitable plaintext m.

A *key-recovery forger* $\mathcal{F}$ for a public-key encryption scheme is a probabilistic polynomial-time algorithm that, on input the public key pk, returns the secret key sk with non-negligible probability.

Many code-based public-key encryption schemes can be found in literature: the most notable are the McEliece (Section 3) [McE78] and the Niederreiter [Nie86] cryptosystems.

Digital signatures and CFS. A digital signature algorithm is a scheme composed by three algorithms $(\mathsf{KGen}, \mathsf{Sig}, \mathsf{Verify})$: KGen takes as input a security parameter λ in unary and returns a pair of public-secret keys, while Sig and Verify are the signature and verify algorithms. If $(\mathsf{pk}, \mathsf{sk}) = \mathsf{KGen}(1^\lambda)$ and $\sigma = \mathsf{Sig}(\mathsf{sk}, m)$, then $\mathsf{Verify}(\mathsf{pk}, \sigma, m)$ accepts the signature σ with overwhelming probability for every message m and security parameter λ.

We adapt the key-recovery forger in the case of digital signatures. A *key-recovery forger* $\mathcal{F}$ for a signature scheme is a probabilistic polynomial-time algorithm that, on input the public key pk, returns the secret key sk with non-negligible probability.

The CFS scheme [CFS01] is a code-based digital signature based on the *hash-and-sign* paradigm. The public key is a randomly scrambled binary irreducible Goppa code $\mathcal{C}$, while the secret key is the decoding algorithm $D_{\mathcal{C}}$. This scheme is highly impractical: in the signing algorithm, we compute the hash $h(m, i)$, where m is the message and i a nonce, until the digest is a decodable vector for $\mathcal{C}$. This operation requires a huge number of hashes and decoding, roughly $t!$, where t is the error correction capability of $\mathcal{C}$. The construction presented in [LYHW19] is based on CFS and a modified version of McEliece, to reduce the computational effort of the signing algorithm.

3 Modified McEliece Cryptosystem

In this section we recall the textbook definition of the McEliece public-key cryptosystem and the modified version from [LYHW19], proving that this modification does not improve the security of the scheme, even if it pays the cost of having longer keys.

3.1 McEliece and the modified version

The following is the textbook version of McEliece [McE78], the starting point of the modification given in [LYHW19]. Alternative versions of this scheme use different techniques to achieve higher levels of security and/or more compact keys. The public-key encryption cryptosystem McEliece (ME) is composed of the following three algorithms.

- $\mathsf{KGen}_{\mathsf{ME}}(1^\lambda)$: generate a Goppa code $\mathcal{C}$ over $\mathbb{F}_{2^m}$ with parameters $[n = 2^m, k, 2t + 1]$ according to λ. Let G be a generator matrix of $\mathcal{C}$ and let $\mathbf{D}_{\mathcal{C}}$ be an efficient decoding algorithm. Sample two random matrices with coefficients in $\mathbb{F}_2$: a $k \times k$ invertible matrix S and a $n \times n$ permutation matrix P. Set $G_{\text{pub}} = SGP$. Return (G_{pub}, t) as public key and $(S, P, \mathbf{D}_{\mathcal{C}})$ as secret key.
- $\mathsf{Enc}_{\mathsf{ME}}(m, (G_{\text{pub}}, t))$: the ciphertext of a message m in $\mathbb{F}_2^k$ is given by $mG_{\text{pub}} + e$, where e is a randomly chosen vector in $\mathbb{F}_{2^m}^n$ of weight t.
- $\mathsf{Dec}_{\mathsf{ME}}(c, (S, G, P, \mathbf{D}_{\mathcal{C}}))$: given the ciphertext c, compute cP^{-1} and apply the decoding algorithm $\mathbf{D}_{\mathcal{C}}$, obtaining the vector $x = mS$. The plaintext is given by $xS^{-1} = m$.

We recall that this scheme is not IND-CPA secure [NIKM08], for stronger variants achieving this security level and others see [DDMN12; Per18]. Moreover, the

state of the art uses matrices in systematic form as public key to reduce their size, an example can be Classic McEliece [BCL+17] (even if it is based on Niederreiter [Nie86], a cryptosystem equivalent to McEliece).

Now we recall the modified version of the cryptosystem above from [LYHW19]. We refer to this scheme as Modified McEliece (MME).

- $\mathsf{KGen}_{\mathsf{MME}}(1^\lambda)$: generate a Goppa code $\mathcal{C}$ over $\mathbb{F}_{2^m}$ with parameters $[n = 2^m, k, 2t+1]$ according to λ. Let G be a generator matrix of $\mathcal{C}$ and $\mathbf{D}_\mathcal{C}$ an efficient decoding algorithm. Sample two random $k \times k$ invertible matrices A and B and a $n \times n$ permutation matrix P, all with coefficients in $\mathbb{F}_2$. Set $G' = AGP$ and $G'' = BGP$. Moreover, set

$$\rho = (A+B)^{-1}B \text{ and } \gamma = \left(A + B\,(A+B)^{-1}\,B\right)^{-1};$$

  if such matrices are non-invertible, pick different A and B. Return (G', G'', t) as public key and $(P, \rho, \gamma, \mathbf{D}_\mathcal{C})$ as secret key.

- $\mathsf{Enc}_{\mathsf{MME}}(m, (G', G'', t))$: given a message m in $\mathbb{F}_2^k$, split it as $m = m_1 + m_2$, where m_1 is chosen at random. The ciphertext of m is given by (c_1, c_2) where

$$c_1 = m_1 G' + m_2 G'' + e_1 \text{ and } c_2 = mG' + m_1 G'' + e_2$$

  with e_1, e_2 random elements in $\mathbb{F}_2^n$ of weight t.

- $\mathsf{Dec}_{\mathsf{MME}}(c, (P, \rho, \gamma, \mathbf{D}_\mathcal{C}))$: given the ciphertext $c = (c_1, c_2)$, compute $c_1 P^{-1}$ and $c_2 P^{-1}$. Decode these vectors with $\mathbf{D}_\mathcal{C}$ and obtain x_1, x_2. The plaintext can be reconstructed by

$$m = (x_2 + x_1\rho)\gamma.$$

The major difference between this scheme and the original McEliece one is the randomization in the encryption procedure. This is achieved generating two random messages m_1 and m_2 from the plaintext m and using the two randomly generated public matrices G' and G'' to encode them. For an analysis of the correctness of this scheme, we refer to the original work [LYHW19].

3.2 Security analysis

The main point of the modification of MME is the use of two different public matrices G' and G'', as well as the randomization in the encryption procedure, splitting the message into two random additive shares $m = m_1 + m_2$.

In the following result we reduce the security of MME to the security of ME. The main observation is that, in the algorithm $\mathsf{KGen}_{\mathsf{MME}}(1^\lambda)$, the matrices G' and G'' generates the same linear code generated by the matrix GP. Moreover, the codes generated by G' and G'' are the same and hence, there exists a change of basis sending G' to G''. In this way, the link between matrices G' and G'' is used to reduce the key-recovery attack of MME to the one of ME.

Proposition 1. *Let $\mathcal{F}_{\mathsf{ME}}$ be a key-recovery forger able to retrieve the secret key for the scheme ME, then it is possible to design a key-recovery forger $\mathcal{F}_{\mathsf{MME}}$ for MME that uses $\mathcal{F}_{\mathsf{ME}}$ as a subroutine.*

Proof. Let $(P, \rho, \gamma, \mathbf{D}_{\mathcal{C}})$ be the secret key of the public key (G', G'', t) of the scheme MME. By hypothesis, let $\mathcal{F}_{\mathsf{ME}}$ be a forger for ME: given a public key for ME, it returns, with non-negligible probability p, the corresponding secret key. In formulas we have $\mathbf{P}\left[\mathcal{F}_{\mathsf{ME}}(\mathsf{pk}) = \mathsf{sk}\right] = p$.
Note that, by construction, the pair (G', t) can be viewed as a public key of ME, relative to the secret key $(A, P, \mathbf{D}_{\mathcal{C}})$. Moreover, observe that $G' = AGP$ and $G'' = BGP$ are generator matrices of the code generated by GP, a permutation of $\mathcal{C}$. In other words, the rows of G' generate the same linear space of the one generated by the rows of G''. This implies that there exists a $k \times k$ change of basis matrix Σ such that $G'' = \Sigma G'$. The computation of such Σ can be performed in polynomial time using matrix operations and we call this subroutine ChangeBasis. The forger $\mathcal{F}_{\mathsf{MME}}$ for MME is given in Figure 1.

$\underline{\mathcal{F}_{\mathsf{MME}}(G', G'', t)}$:
$(A, P, \mathbf{D}_{\mathcal{C}}) \leftarrow \mathcal{F}_{\mathsf{ME}}(G', t)$
$\Sigma \leftarrow \mathsf{ChangeBasis}(G', G'')$
$B \leftarrow \Sigma A$
$\rho \leftarrow (A + B)^{-1} B$
$\gamma \leftarrow \left(A + B\,(A + B)^{-1}\,B\right)^{-1}$
return $(P, \rho, \gamma, \mathbf{D}_{\mathcal{C}})$

Figure 1: Key-recovery forger for MME

The success probability of $\mathcal{F}_{\mathsf{MME}}$ is given by

$$\mathbf{P}\left[\mathcal{F}_{\mathsf{MME}}(G', G'', t) = (P, \rho, \gamma, \mathbf{D}_{\mathcal{C}})\right] = \mathbf{P}\left[\mathcal{F}_{\mathsf{ME}}((G', t)) = (A, P, \mathbf{D}_{\mathcal{C}})\right] = p$$

and hence, it is non-negligible. Now we analyze the time complexity $C_{\mathcal{F}_{\mathsf{MME}}}$ of $\mathcal{F}_{\mathsf{MME}}$. Suppose that $C_{\mathcal{F}_{\mathsf{ME}}}$ is the time complexity of $\mathcal{F}_{\mathsf{ME}}$. The subroutine ChangeBasis uses standard techniques from linear algebra and has complexity $\mathcal{O}(n^3)$; similarly, all the computations involving matrices can be performed in polynomial time, more precisely in $\mathcal{O}(\ell^3)$, where ℓ is the dimension of the matrices. Since in this scheme we have $\ell = k$ and $k \leq n$, this implies

$$C_{\mathcal{F}_{\mathsf{MME}}} = C_{\mathcal{F}_{\mathsf{ME}}} + \mathcal{O}(n^3).$$

We can conclude that the key-recovery forger $\mathcal{F}_{\mathsf{MME}}$ has the same complexity of $\mathcal{F}_{\mathsf{ME}}$ plus a negligible polynomial term $\mathcal{O}(n^3)$, and non-negligible success probability. □

	pk	|pk|
ME	(G_{pub}, t)	$nkm + \lceil \log_2(t) \rceil$
MME	(G', G'', t)	$2nkm + \lceil \log_2(t) \rceil$

Table 1: Comparison of the public key sizes (in bits).

The previous proposition shows how the security of the secret key of MME is at least as weak as the one of ME. We highlight that we do not propose an attack to

MME or ME but, forging the two schemes requires essentially the same effort and a set of parameters that is secure for MME, it is secure for ME and vice versa. If we take into account that the size of the public key of MME is roughly twice of the one of ME (Table 1), and the fact that the major drawback of ME is the size of the public key, the adoption of MME does not solve this problem.

4 The Signature Scheme

In this section, we report and analyze the CFS-like signature scheme presented in [LYHW19]. The protocol, that we call LYHW19, is the following:

- $\mathsf{KGen}_{\mathsf{LYHW19}}(1^\lambda)$: generate a Goppa code $\mathcal{C}$ over $\mathbb{F}_{2^m}$ with parameters $[n = 2^m, k, 2t+1]$ according to λ. Let G be a generator matrix of $\mathcal{C}$ and $\mathbf{D}_\mathcal{C}$ an efficient decoding algorithm. Sample two random $k \times k$ invertible matrices A and B and a $n \times n$ permutation matrix P, all with coefficients in $\mathbb{F}_2$. If the matrix $\left(A + B + AB^{-1}A\right)$ is singular, pick different A and B. Set $G' = AGP$ and $G'' = BGP$. Choose two hash functions $h_1, h_2 : \{0,1\}^* \to \{0,1\}^n$. Return (G', G'', t, h_1, h_2) as public key and $(A, B, P, \mathbf{D}_\mathcal{C})$ as secret key.
- $\mathsf{Sig}_{\mathsf{LYHW19}}(M, (A, B, P, \mathbf{D}_\mathcal{C}))$: given the message M, compute $d = h_1(M)$ and find two nonces i_1, i_2 in $\mathbb{N}$ such that $y_1 = h_1(d||i_1)P^{-1}$ and $y_2 = h_2(d||i_2)P^{-1}$ are two decodable vectors using $\mathbf{D}_\mathcal{C}$. From y_i we obtain messages $x_i = \mathbf{D}_\mathcal{C}(y_i)$, for $i = 1, 2$. Set $e_1 = y_1 + x_1$ and $e_2 = y_2 + x_2$. Now set

$$m_1 = \left(x_2 + x_1 B^{-1} A\right)\left(A + B + AB^{-1}A\right)^{-1},$$
$$m_2 = x_1 B^{-1} + \left(x_2 + x_1 B^{-1} A\right)\left(A + B + AB^{-1}A\right)^{-1} AB^{-1}.$$

  Return the tuple $(i_1, i_2, m_1, m_2, e_1, e_2)$ as a signature for M .
- $\mathsf{Verify}_{\mathsf{LYHW19}}(M, \sigma, (G', G'', t, h_1, h_2))$: given the signature σ for the message M, parse σ as $(i_1, i_2, m_1, m_2, e_1, e_2)$ and check that
  1. $m_1 G' + m_2 G'' + e_1 = h_1(h_1(M)||i_1)$, and
  2. $(m_1 + m_2)G' + m_1 G'' + e_2 = h_2(h_1(M)||i_2)$.

  The signature is valid if both checks pass, otherwise reject.

This scheme uses the CFS construction [CFS01], using brute-force hashing to find a decodable digest. While CFS does this process for a single decodable digest, LYHW19 searches two of them. The probability of finding a single such digest is $\sim \frac{1}{t!}$ and then, the number of hashes and decoding attempts in $\mathsf{Sig}_{\mathsf{LYHW19}}$ is roughly $2(t!)$.

The authors of LYHW19, in their work [LYHW19], propose some small parameters to improve the signing time of the scheme. Since they assume that the security of the signature scheme is based on the security of the MME encryption scheme, they propose to set t between 1 and 3. In this way the number of decoding attempts is very low, instead of 10! as proposed in the original CFS signature [CFS01].

Observe that, as shown in Subsection 3.2, the two matrices G' and G'' in the public key generate the same code, that we call $\mathcal{C}'$. This code is the same of the one generated by GP. Therefore, we have that $\mathcal{C}'$ and $\mathcal{C}$, the secret code generated by G, are permutation equivalent. Recall that the code $\mathcal{C}$ is a Goppa code and it is defined

by a monic irreducible polynomial $g_{\mathcal{C}}(X)$ in $\mathbb{F}_{2^m}[X]$. A possible attack on the secret key of LYHW19 is to guess the polynomial $g(X)$ and check the permutation equivalence between the Goppa code generated by $g(X)$ and the code $\mathcal{C}'$.

A small t, as used in [LYHW19], improves the performances and the number of hashes/decodings in the signing process, but leads to some security issues, as shown in [LS01]. We show how to design a practical key-recovery attack.

Proposition 2. *The signature scheme* LYHW19*, with parameters proposed in [LYHW19], that is t equal to* 1*,* 2 *or* 3*, admits a key-recovery forger.*

Proof. A key-recovery forger $\mathcal{F}_{\mathsf{LYHW19}}$ can be defined as follows. Given the public key (G', G'', t), we can enumerate all the monic irreducible polynomials in $\mathbb{F}_{2^m}[X]$ and store them in a list L. Then, for each $g(X)$ in L, check if the irreducible Goppa codes $\mathcal{C}$ defined by $g(X)$ is permutation equivalent to the code generated by G' (or, equivalently, by G''). When we find an equivalence via the permutation matrix P returned by the Support Splitting Algorithm, we use P to compute the secret matrices A and B with standard techniques from linear algebra, as done in the proof of Prop. 1. Then the forger returns the secret key $(A, B, P, \mathbf{D}_{\mathcal{C}})$.

Let us analyze the overall complexity of this attack. The cardinality of the list L is $\mathcal{O}(2^{mt}/t)$ [LN97, Th. 3.25]. Then, the number of irreducible Goppa codes to check is $\mathcal{O}(2^{mt}/t)$. This number can be decreased up to $\mathcal{O}(2^{m(t-3)}/(mt))$ using techniques from [LS01], even if this step is not crucial in our proof.
Let C_{SSA} be the complexity of the Support Splitting Algorithm. Then, the overall complexity of $\mathcal{F}_{\mathsf{LYHW19}}$ is given by

$$\mathcal{O}\left(\frac{2^{m(t-3)}}{mt} C_{\mathrm{SSA}}\right) + \mathcal{O}(n^3),$$

where the second factor refers to the matrix operations to retrieve A and B from P and the generator matrix of the Goppa code $\mathcal{C}$. The Support Splitting Algorithm has average complexity $\mathcal{O}(n^3)$ [LS01] and then the forger will achieve average complexity

$$\mathcal{O}\left(\frac{2^{m(t-3)}}{mt} n^3\right) + \mathcal{O}(n^3) = \mathcal{O}\left(\frac{2^{m(t-3)}}{mt} n^3\right).$$

Since by definition $n = 2^m$, we can conclude that the forger has average complexity $\mathcal{O}(2^{mt}/(mt))$.

When we take into account the parameters suggested for LYHW19, that are $t = 1, 2, 3$, regardless of the choice of m, the forger becomes polynomial in $n = 2^m$, having complexity $\mathcal{O}(2^{3m}/(3m)) = \mathcal{O}(n^3)$. □

The previous result shows how the balance between security and efficiency is very subtle. For any $n = 2^m$, in Table 2 we show how the number of hashes and decoding attempts in the signing process of LYHW19 and the complexity of the attack showed in Prop. 2 are affected when t is increased.

We want to highlight that the LYHW19 signature scheme is not itself insecure with the appropriate choices of parameters, however, it inherits the drawbacks of CFS. To achieve 80 bit of security, a t equal to 33 should be used, while for 128 bits of security, it is needed $t = 56$ [BLP08]. These two parameters imply a highly impractical number of hash and decoding computations (2^{124} and 2^{250}, respectively).

t	number of hashes/dec.	$C_{\mathcal{F}_{\mathsf{LYHW19}}}$
3	12	$\mathcal{O}(n^3/(3\log_2 n))$
5	240	$\mathcal{O}(n^5/(5\log_2 n))$
8	80640	$\mathcal{O}(n^8/(8\log_2 n))$
10	$7257600 \approx 2^{23}$	$\mathcal{O}(n^{10}/(10\log_2 n))$

Table 2: Comparison between the expected number of hashes and decoding attempts in the signing procedure of LYHW19 and the average complexity $C_{\mathcal{F}_{\mathsf{LYHW19}}}$ of the forger.

5 Conclusions

We analyzed the two cryptosystem proposed in [LYHW19], the MME public-key encryption scheme and the LYHW19 signature scheme. These two schemes are modification of the well-knows construction McEliece (ME) and CFS, respectively. We showed that the adoption of these new alternatives does not bring any advantage, even if their security level is roughly equivalent to the one of the scheme they try to improve. In particular, MME is not weaker than the standard scheme ME, but the adoption of a longer public key does not bring any additional security, while LYHW19 has the same issue as CFS: for a reasonable level of security, it becomes impractical. The field of code-based signature schemes is very prosperous, and the research is still looking for other solutions which maintain the original security of CFS but also decrease the computational complexity of the signing algorithm.

Acknowledgments

The author is a member of the INdAM Research group GNSAGA and acknowledges support from TIM S.p.A. through the Ph.D. scholarship. The author would like to thank the anonymous reviewers for the valuable comments, which helped to improve the overall quality of this work.

References

[AAC+22] G. Alagic, D. Apon, D. Cooper, Q. Dang, T. Dang, J. Kelsey, J. Lichtinger, C. Miller, D. Moody, R. Peralta, *et al.*, "Status report on the third round of the NIST post-quantum cryptography standardization process," *US Department of Commerce, NIST*, 2022 (cit. on p. 1).

[ABB+17] N. Aragon, P. S. Barreto, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, S. Gueron, T. Guneysu, C. A. Melchor, *et al.*, "BIKE: bit flipping key encapsulation," 2017 (cit. on p. 1).

[ABG+19] N. Aragon, O. Blazy, P. Gaborit, A. Hauteville, and G. Zémor, "Durandal: a rank metric based signature scheme," in *Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part III 38*, Springer, 2019, pp. 728–758 (cit. on p. 2).

[ALP+18] J. Alperin-Sheriff, Y. Lee, R. Perlner, W. Lee, and D. Moody, *Official comments on pqsigRM*, `https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/zeHJEzWdv2Y/m/1YAVqa3mBQAJ`, Accessed 2023-02-14, 2018 (cit. on p. 2).

[BBC+13] M. Baldi, M. Bianchi, F. Chiaraluce, J. Rosenthal, and D. Schipani, "Using LDGM codes and sparse syndromes to achieve digital signatures," in *Post-Quantum Cryptography: 5th International Workshop, PQCrypto 2013, Limoges, France, June 4-7, 2013. Proceedings 5*, Springer, 2013, pp. 1–15 (cit. on p. 2).

[BBPS21] A. Barenghi, J.-F. Biasse, E. Persichetti, and P. Santini, "LESS-FM: fine-tuning signatures from the code equivalence problem," in *International Conference on Post-Quantum Cryptography*, Springer, 2021, pp. 23–43 (cit. on p. 2).

[BCL+17] D. J. Bernstein, T. Chou, T. Lange, I. von Maurich, R. Misoczki, R. Niederhagen, E. Persichetti, C. Peters, P. Schwabe, N. Sendrier, *et al.*, "Classic McEliece: conservative code-based cryptography," *NIST submissions*, 2017 (cit. on pp. 1, 5).

[BDK+18] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, "CRYSTALS-Kyber: a CCA-secure module-lattice-based KEM," in *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, IEEE, 2018, pp. 353–367 (cit. on p. 1).

[Ber73] E. Berlekamp, "Goppa codes," *IEEE Transactions on Information Theory*, vol. 19, no. 5, pp. 590–592, 1973 (cit. on p. 3).

[BHK+19] D. J. Bernstein, A. Hülsing, S. Kölbl, R. Niederhagen, J. Rijneveld, and P. Schwabe, "The SPHINCS+ signature framework," in *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, 2019, pp. 2129–2146 (cit. on p. 1).

[BHLP17] D. Bernstein, A. Hülsing, T. Lange, and L. Panny, *Comments on racoss*, `https://helaas.org/racoss/`, Accessed 2023-02-14, 2017 (cit. on p. 2).

[BLP08] D. J. Bernstein, T. Lange, and C. Peters, "Attacking and defending the McEliece cryptosystem," in *Post-Quantum Cryptography: Second International Workshop, PQCrypto 2008 Cincinnati, OH, USA, October 17-19, 2008 Proceedings 2*, Springer, 2008, pp. 31–46 (cit. on p. 8).

[BMV78] E. Berlekamp, R. McEliece, and H. Van Tilborg, "On the inherent intractability of certain coding problems (corresp.)," *IEEE Transactions on Information Theory*, vol. 24, no. 3, pp. 384–386, 1978 (cit. on p. 3).

[CFS01] N. T. Courtois, M. Finiasz, and N. Sendrier, "How to achieve a McEliece-based digital signature scheme," in *International Conference on the Theory and Application of Cryptology and Information Security*, Springer, 2001, pp. 157–174 (cit. on pp. 2, 4, 7).

[DDMN12] N. Dottling, R. Dowsley, J. Muller-Quade, and A. C. Nascimento, "A CCA2 secure variant of the McEliece cryptosystem," *IEEE Transactions on Information Theory*, vol. 58, no. 10, pp. 6672–6680, 2012 (cit. on p. 4).

[DKL+18] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé, "Crystals-dilithium: A lattice-based digital signature scheme," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 238–268, 2018 (cit. on p. 1).

[DMP21] G. D'Alconzo, A. Meneghetti, and P. Piasenti, "Security issues of CFS-like digital signature algorithms," *arXiv preprint arXiv:2112.00429*, 2021 (cit. on p. 2).

[DST19] T. Debris-Alazard, N. Sendrier, and J.-P. Tillich, "Wave: A new family of trapdoor one-way preimage sampleable functions based on codes," in *International Conference on the Theory and Application of Cryptology and Information Security*, Springer, 2019, pp. 21–51 (cit. on p. 2).

[FHK+18] P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, Z. Zhang, *et al.*, "Falcon: Fast-Fourier lattice-based compact signatures over NTRU," *Submission to the NIST's post-quantum cryptography standardization process*, vol. 36, no. 5, 2018 (cit. on p. 1).

[FJR22] T. Feneuil, A. Joux, and M. Rivain, "Syndrome decoding in the head: shorter signatures from zero-knowledge proofs," in *Advances in Cryptology–CRYPTO 2022: 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara,*

CA, USA, August 15–18, 2022, Proceedings, Part II, Springer, 2022, pp. 541–572 (cit. on p. 2).

[FJR23] ——, "Shared permutation for syndrome decoding: New zero-knowledge protocol and code-based signature," *Designs, Codes and Cryptography*, vol. 91, no. 2, pp. 563–608, 2023 (cit. on p. 2).

[GPS22] S. Gueron, E. Persichetti, and P. Santini, "Designing a Practical Code-Based Signature Scheme from Zero-Knowledge Proofs with Trusted Setup," *Cryptography*, vol. 6, no. 1, p. 5, 2022 (cit. on p. 2).

[KKS97] G. Kabatianskii, E. Krouk, and B. Smeets, "A digital signature scheme based on random error-correcting codes," in *IMA International Conference on Cryptography and Coding*, Springer, 1997, pp. 161–167 (cit. on p. 2).

[LN97] R. Lidl and H. Niederreiter, *Finite fields*, 20. Cambridge university press, 1997 (cit. on p. 8).

[LS01] P. Loidreau and N. Sendrier, "Weak keys in the McEliece public-key cryptosystem," *IEEE Transactions on Information Theory*, vol. 47, no. 3, pp. 1207–1211, 2001 (cit. on p. 8).

[LT20] T. S. C. Lau and C. H. Tan, "MURAVE: A New Rank Code-Based Signature with MUltiple RAnk VErification," in *Code-Based Cryptography: 8th International Workshop, CBCrypto 2020, Zagreb, Croatia, May 9–10, 2020, Revised Selected Papers*, Springer, 2020, pp. 94–116 (cit. on p. 2).

[LYHW19] X. Liu, X. Yang, Y. Han, and X. A. Wang, "A secure and efficient code-based signature scheme," *International Journal of Foundations of Computer Science*, vol. 30, no. 04, pp. 635–645, 2019 (cit. on pp. 2, 4, 5, 7–9).

[MAB+18] C. A. Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, E. Persichetti, G. Zémor, and I. Bourges, "Hamming quasi-cyclic (HQC)," *NIST PQC Round*, vol. 2, no. 4, p. 13, 2018 (cit. on p. 2).

[McE78] R. J. McEliece, "A public-key cryptosystem based on algebraic," *Coding Thv*, vol. 4244, pp. 114–116, 1978 (cit. on pp. 3, 4).

[MS77] F. J. MacWilliams and N. J. A. Sloane, *The theory of error correcting codes*. Elsevier, 1977, vol. 16 (cit. on p. 3).

[Nie86] H. Niederreiter, "Knapsack-type cryptosystems and algebraic coding theory," *Prob. Contr. Inform. Theory*, vol. 15, no. 2, pp. 157–166, 1986 (cit. on pp. 3, 5).

[NIKM08] R. Nojima, H. Imai, K. Kobara, and K. Morozov, "Semantic security for the McEliece cryptosystem without random oracles," *Designs, Codes and Cryptography*, vol. 49, no. 1, pp. 289–305, 2008 (cit. on p. 4).

[Pat75] N. Patterson, "The algebraic decoding of Goppa codes," *IEEE Transactions on Information Theory*, vol. 21, no. 2, pp. 203–207, 1975 (cit. on p. 3).

[Per18] E. Persichetti, "On the CCA2 security of McEliece in the standard model," in *International Conference on Provable Security*, Springer, 2018, pp. 165–181 (cit. on p. 4).

[PR97] E. Petrank and R. M. Roth, "Is code equivalence easy to decide?" *IEEE Transactions on Information Theory*, vol. 43, no. 5, pp. 1602–1604, 1997 (cit. on p. 3).

[Sen00] N. Sendrier, "Finding the permutation between equivalent linear codes: The support splitting algorithm," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1193–1203, 2000 (cit. on p. 3).