Research Article

Jean Belo Klamti* and M. Anwarul Hasan

A code-based hybrid signcryption scheme

https://doi.org/10.1515/jmc-2022-0002
received January 07, 2022; accepted March 21, 2023

Abstract: A key encapsulation mechanism (KEM) that takes as input an arbitrary string, i.e., a tag, is known as tag-KEM, while a scheme that combines signature and encryption is called signcryption. In this article, we present a code-based signcryption tag-KEM scheme. We utilize a code-based signature and an IND-CCA2 (adaptive chosen ciphertext attack) secure version of McEliece's encryption scheme. The proposed scheme uses an equivalent subcode as a public code for the receiver, making the NP-completeness of the subcode equivalence problem be one of our main security assumptions. We then base the signcryption tag-KEM to design a code-based hybrid signcryption scheme. A hybrid scheme deploys asymmetric- as well as symmetric-key encryption. We give security analyses of both our schemes in the standard model and prove that they are secure against IND-CCA2 (indistinguishability under adaptive chosen ciphertext attack) and SUF-CMA (strong existential unforgeability under chosen message attack).

Keywords: coding theory, signature scheme, public-key cryptography, code-based cryptography, signcryption

MSC 2020: 94A60

1 Introduction

In public-key cryptography, the authentication and confidentiality of communication between a sender and a receiver are ensured by a two-step approach called *signature-then-encryption*. In this approach, the sender uses a digital signature scheme to sign a message and then encrypt it using an encryption algorithm. The cost of delivering a message in a secure and authenticated way using the signature-then-encryption approach is essentially the sum of the cost of a digital signature and that of encryption.

In 1997, Zheng introduced a new cryptographic primitive called *signcryption* to provide both authentication and confidentiality in a single logical step [1]. In general, one can expect the cost of signcryption to be noticeably less than that of signature-then-encryption. Zheng's signcryption scheme is based on the hardness of the discrete logarithm problem. Since Zheng's work, a number of signcryption schemes based on different hard assumptions have been introduced, see, for example, [1–12]. Of these, the most efficient ones have followed Zheng's approach, i.e., used symmetric-key encryption as a black-box component [6–8]. It has been of interest to many researchers to study how a combination of asymmetric- and symmetric-key encryption schemes could be used to build efficient signcryption schemes in a more general setting.

To that end, Dent in 2004 proposed the first formal composition model for hybrid signcryption [13] and in 2005 developed an efficient model for signcryption KEMs in the *outsider*- and the *insider*-secure setting [14,15]. In the outsider-secure setting, the adversary is assumed to be distinct from the sender and receiver,

*** Corresponding author: Jean Belo Klamti,** Department of Electrical and Computer Engineering, University of Waterloo, 200 University Ave W, Waterloo, ON, N2L 3G1, Canada, e-mail: jbklamti@uwaterloo.ca
M. Anwarul Hasan: Department of Electrical and Computer Engineering, University of Waterloo, 200 University Ave W, Waterloo, ON, N2L 3G1, Canada, e-mail: ahasan@uwaterloo.ca

while in the insider-secure setting, the adversary is assumed to be a second party (i.e., either sender or receiver).

To improve the model for the insider-secure setting in hybrid signcryption, Bjørstad and Dent in 2006 proposed a model based on encryption tag-KEM rather than regular encryption KEM [16]. Their model provides a simpler description of signcryption with a better generic security reduction for the signcryption tag-KEM construction. A year after Bjørstad and Dent's work, Yoshida and Fujiwara reported the first study of multi-user setting security of signcryption tag-KEMs [17], which is a more suitable setting for the analysis of insider-secure schemes.

1.1 Motivation

Most of the aforementioned signcryption schemes are based on the hardness of either the discrete logarithm or the integer factorization problem and would be broken with the arrival of sufficiently large quantum computers. Therefore, it is of interest to design signcryption schemes for the postquantum era. Coding theory has some hard problems that are considered quantum-safe and in this article, we explore the design of code-based signcryption.

The first attempt for code-based signcryption was presented in 2012 by Mathew et al. [18]. After that work, an attribute-based signcryption scheme using linear codes was introduced in 2017 by Song et al. [19]. Code-based signcryption remains an active area of research, specifically to study the design of cryptographic primitives like signcryption schemes that are quantum-safe.

1.2 Contributions

In this article, we present a signcryption tag-KEM scheme using a probabilistic full domain hash (FDH) like code-based signature and a CCA2 secure version of McEliece's encryption scheme. The underlying code-based signature in our scheme is called *Wave* introduced by Banegas et al. [20], while the CCA2 secure version of the McEliece scheme is based on the Fujisaki–Okamoto transformation introduced by Cayrel et al. [21]. For the underlying McEliece scheme, we use a generator matrix of permuted Goppa subcodes as receivers' public keys. With this feature, we are able to reduce the public key size of our scheme and include the subcode equivalence problem as one of your security assumptions. Because of the latter, for the key recovery attack, even if an adversary is able to distinguish whether the underlying code is a Goppa code, it has to solve the subcode equivalence problem, which is NP-complete. Thus, with well-chosen parameters, the most efficient attack against our scheme will be a brute-force attack.

Using the signcryption tag-KEM, we design a code-based hybrid signcryption scheme. Then we give security analyses of these two schemes in the standard model assuming the insider-secure setting. Finally, we give a comparison of the hybrid signcryption with some relevant lattice-based signcryptions in terms of key and ciphertext sizes.

1.3 Organization

This article is organized as follows. In Section 2, we first recall some basic notions of coding theory and then briefly describe relevant encryption and signature schemes that are of interest to this work. Section 3 presents the definition and framework of signcryption and hybrid signcryption, and a brief review of the relevant security model. We present our signcryption and hybrid signcryption schemes in Section 4 and then provide security analyses of the proposed schemes in Section 5. We provide a set of parameters for the hybrid signcryption scheme in Section 6 and then conclude in Section 7.

1.4 Notations

In this article, we use the following notations:
- $\mathbb{F}_q$: finite field of size q where $q = p^m$ is a prime power.
- C: $\mathbb{F}_q$-linear code of length n.
- $\boldsymbol{x}$: a word or vector of $\mathbb{F}_q^n$.
- $\mathsf{wt}(\boldsymbol{x})$: weight of $\boldsymbol{x}$.
- $\mathbf{G}$ (resp. $\mathbf{H}$): generator (resp. parity-check) matrix of linear code C.
- $\mathcal{W}_{q,n,t}$ is the set of q-ary vectors of length n and weight t.
- sk_s (resp. sk_r): sender's (resp. receiver's) secrete key for signcryption.
- pk_s (resp. pk_r): sender's (resp. receiver's) public key for signcryption.

2 Preliminaries

In this section, we recall some notions pertaining to coding theory and code-based cryptography.

2.1 Coding theory and some relevant hard problems

Let us consider the finite field $\mathbb{F}_q$. A q-ary linear code C of length n and dimension k over $\mathbb{F}_q$ is a vector subspace of dimension k of $\mathbb{F}_q^n$. It can be specified by a full rank matrix $\mathbf{G} \in \mathbb{F}_q^{k\times n}$, called *generator matrix* of C, whose rows span the code. Namely, $C = \{\boldsymbol{x}\mathbf{G} \text{ s.t. } \boldsymbol{x} \in \mathbb{F}_q^k\}$. A linear code can also be defined by the right kernel of matrix $\mathbf{H} \in \mathbb{F}_q^{r\times n}$, called *parity-check matrix* of C, as follows:

$$C = \{\boldsymbol{x} \in \mathbb{F}_q^n \quad \text{s.t.} \quad \mathbf{H}\boldsymbol{x}^T = \boldsymbol{0}\}.$$

The *Hamming distance* between two codewords is the number of positions (coordinates) where they differ. The *minimal* distance of a code is the minimal distance of all codewords.

The *weight* of a word or vector $\boldsymbol{x} \in \mathbb{F}_q^n$, denoted by $wt(\boldsymbol{x})$, is the number of its nonzero positions. Then the *minimal* weight of a code C is the minimal weight of all nonzero codewords. In the case of linear code C, its minimal distance is equal to the minimal weight of the code.

Below we recall some hard problems that are relevant to our discussions and analyses presented in this article.

Problem 1. (Binary syndrome decoding problem) Given a matrix $\mathbf{H} \in \mathbb{F}_2^{r\times n}$, a vector $\boldsymbol{s} \in \mathbb{F}_2^r$, and an integer $\omega > 0$, find a vector $\boldsymbol{y} \in \mathbb{F}_2^n$ such that $\mathsf{wt}(\boldsymbol{y}) = \omega$ and $\boldsymbol{s} = \boldsymbol{y}\mathbf{H}^T$.

The syndrome decoding problem was proven to be NP-complete in 1978 by Berlekamp et al. [22]. It is equivalent to the following problem.

Problem 2. (General decoding problem) Given a matrix $\mathbf{G} \in \mathbb{F}_2^{k\times n}$, a vector $\boldsymbol{y} \in \mathbb{F}_2^n$, and an integer $\omega > 0$, find two vectors $\boldsymbol{m} \in \mathbb{F}_q^k$ and $\boldsymbol{e} \in \mathbb{F}_q^n$ such that $\mathsf{wt}(\boldsymbol{e}) = \omega$ and $\boldsymbol{y} = \boldsymbol{m}\mathbf{G} \oplus \boldsymbol{e}$.

The following problem is used in the security proof of the underlying signature that we use in this article. It was first considered by Johansson and Jonsson in [23]. It was analyzed later by Sendrier in [24].

Problem 3. (Decoding one out of many (DOOM) problem) Given a matrix $\mathbf{H} \in \mathbb{F}_q^{r\times n}$, a set of vector $\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_N \in \mathbb{F}_q^r$ and an integer ω, find a vector $\boldsymbol{e} \in \mathbb{F}_q^n$ and an integer i such that $1 \leq i \leq N$, $\mathsf{wt}(\boldsymbol{e}) = \omega$ and $\boldsymbol{s}_i = \boldsymbol{e}\mathbf{H}^T$.

Problem 4. (Goppa code distinguishing problem) Given a matrix $\mathbf{G} \in \mathbb{F}_2^{k \times n}$, decide whether $\mathbf{G}$ is a random binary or generator matrix of a Goppa code.

Faugère et al. [25] showed that Problem 4 can be solved in special cases of Goppa codes with high rate.

The following is one of the problems, which the security assumption of our scheme's underlying signature mechanism relies on.

Problem 5. (Generalized $(U, U+V)$ code distinguishing problem). Given a matrix $\mathbf{H} \in \mathbb{F}_q^{r \times n}$, decide whether $\mathbf{H}$ is a parity-check matrix of a generalized $(U, U+V)$ code.

Problem 5 was shown to be hard in the worst case by Debris-Alazard et al. [26] since it is NP-complete. Below, we recall the subcode equivalence problem, which is one of the problems on which the security assumption of our scheme is based. This problem was proven to be NP-complete in 2017 by Berger et al. [27].

Problem 6. (Subcode equivalence problem [27]) Given two linear codes $\mathcal{C}$ and $\mathcal{D}$ of length n and respective dimension k and k', $k' \le k$, over the same finite field $\mathbb{F}_q$, determine whether there exists a permutation σ of the support such that $\sigma(\mathcal{C})$ is a subcode of $\mathcal{D}$.

<u>KeyGen</u>
1. Randomly generate a monic irreducible polynomial $g(x) \in \mathbb{F}_{2^m}[x]$ of degree t
2. Select a uniform random set of n different elements $\Gamma = (a_0, \ldots, a_{n-1}) \in \mathbb{F}_{2^m}^n$.
3. Compute a generator matrix $\mathbf{G}_{\mathsf{sk}} \in \mathbb{F}_2^{k \times n}$ of the binary Goppa code from g and Γ.
4. Randomly choose a full rank matrix $\mathbf{S} \in \mathbb{F}_2^{k \times k}$ and permutation matrix $\mathbf{P} \in \mathbb{F}_2^{n \times n}$ with $k = n - mt$.
5. Set $\mathsf{sk} = (g, \Gamma, \mathbf{S}^{-1}, \mathbf{P})$ and $\mathsf{pk} = \mathbf{G}_{\mathsf{pk}} = \mathbf{S}\mathbf{G}_{\mathsf{sk}}\mathbf{P}$.

<u>Encrypt</u>
Input: Public key $\mathsf{pk} = \mathbf{G}_{\mathsf{pk}}$ of the receiver and clear text $\boldsymbol{m}$.
Output: A ciphertext $\boldsymbol{c}$.
1. $\boldsymbol{y} \xleftarrow{\$} \mathcal{S}_{2,n,t}$
2. Compute $\boldsymbol{r} := \mathcal{H}_0(\boldsymbol{y} \| \boldsymbol{m})$
3. Compute $\boldsymbol{c}_0 := \boldsymbol{r}\mathbf{G}_{\mathsf{pk}} + \boldsymbol{y}$
4. Compute $\boldsymbol{c}_1 := \boldsymbol{m} \oplus \mathcal{H}_1(\boldsymbol{y})$.
5. Parse $\boldsymbol{c} := (\boldsymbol{c}_0 \| \boldsymbol{c}_1)$
6. Return $\boldsymbol{c}$

<u>Decrypt</u>
Input: Receiver's secret key $\mathsf{sk} = (g, \Gamma, \mathbf{S}^{-1}, \mathbf{P})$, a ciphertext $\boldsymbol{c}$ and two hash functions $\mathcal{H}_0$ and $\mathcal{H}_1$.
Output: A clear message $\boldsymbol{m}$.
1. Parse $\boldsymbol{c}$ into $(\boldsymbol{c}_0, \boldsymbol{c}_1)$
2. Compute $\tilde{\boldsymbol{c}}_0 = \boldsymbol{c}_0\mathbf{P}^{-1}$
3. Compute $(\boldsymbol{r}, \boldsymbol{e}) := \gamma_{\text{Goppa}}^{\text{McE}}(\mathsf{sk}, \boldsymbol{c}_0)$, where $\gamma_{\text{Goppa}}^{\text{McE}}$ is a decoding algorithm for Goppa code.
4. Compute $\boldsymbol{m} = \boldsymbol{c}_1 \oplus \mathcal{H}_1(\boldsymbol{e})$
5. Compute $\tilde{\boldsymbol{r}} = \mathcal{H}_0(\boldsymbol{e} \| \boldsymbol{m})$
6. If $\tilde{\boldsymbol{r}}\mathbf{G}_{\mathsf{pk}} \oplus \boldsymbol{y} \neq \boldsymbol{c}_1$:
7. Return $\perp$
8. Return $\boldsymbol{m}$

Figure 1: McEliece's scheme with Fujisaki–Okamoto transformation.

2.2 Code-based encryption

The first code-based encryption was introduced in 1978 by McEliece [28]. In Figure 1, we give the McEliece scheme Fujisaki–Okamoto transformation [21], which comprises three algorithms: key generation, encryption, and decryption.

The main drawback of the McEliece encryption scheme is its very large key size. To address this issue, many variants of McEliece's scheme have been proposed, see, for example, [29–34]. In order to reduce the size of both public and private keys in code-based cryptography, Niederreiter in 1986 introduced a new cryptosystem [35]. Niederreiter's cryptosystem is a dual version of McEliece's cryptosystem with some additional properties such that the ciphertext length is relatively smaller. Indeed, the public key in Niederreiter's cryptosystem is a parity-check matrix instead of a generator matrix. In addition, ciphertexts are syndrome vectors instead of erroneous codewords. However, the McEliece and the Niederreiter schemes are equivalent from the security point of view due to the fact that Problems 1 and 2 are equivalent.

Code-based hybrid encryption: A hybrid encryption scheme is a cryptographic protocol that features both an asymmetric- and a symmetric-key encryption scheme. The first component is known as key encapsulation mechanism (KEM), while the second is called data encapsulation mechanism (DEM). The framework was first introduced in 2003 by Cramer and Shoup [36], and later the first code-based hybrid encryption was introduced in 2013 by Persichetti [37] using Niederreiter's encryption scheme. Persichetti's scheme was implemented in 2017 by Cayrel et al. [38]. After Persichetti's work, some other code-based hybrid encryption schemes have been reported, e.g., [39].

2.3 Code-based signature

Designing a secure and practical code-based signature scheme is still an open problem. The first secure code-based signature scheme was introduced by Courtois et al. (CFS) [40]. It is a FDH like signature with two security assumptions: the indistinguishability of random binary linear codes and the hardness of syndrome decoding problem. To address some of the drawbacks of Courtois et al.'s scheme, Dallot proposed

Input: Public key $\mathsf{pk} = \mathbf{H}_{\mathsf{pk}}$, secret key $\mathsf{sk} = (\mathbf{H}_{\mathsf{sk}}, \mathbf{S}, \mathbf{P})$, dimension k of the generalized $(U, U+V)$-code, the dimension k_U of the code U, the dimension k_V of the code V and the weight ω of error vectors.
Output: $\mathsf{sign}(\boldsymbol{m})$

Sign
1. $\boldsymbol{r} \xleftarrow{\$} \mathbb{F}_2^{\lambda}$
2. $\boldsymbol{y_r} = \mathcal{H}(\boldsymbol{m} \| \boldsymbol{r})$
3. Compute $\boldsymbol{e} = \mathsf{Decode}_{\mathbf{H}_{\mathsf{sk}}}(\boldsymbol{y_r}(\mathbf{S}^{-1})^T)$
4. Return $\mathsf{sign}(\boldsymbol{m}) = (\boldsymbol{e}, \boldsymbol{r})$

Verif
1. Compute $\tilde{\boldsymbol{y}}_{\boldsymbol{r}} = \mathcal{H}(\boldsymbol{m} \| \boldsymbol{r})$
2. Compute $\tilde{\boldsymbol{y}} = \boldsymbol{u}\mathbf{H}_{\mathsf{pk}}^T$.
3. If $\tilde{\boldsymbol{y}}_{\boldsymbol{r}} \neq \tilde{\boldsymbol{y}}$:
   Return $\perp$
4. Else:
   Return valid

Figure 2: Wave signature scheme [41].

a modified version, called mCFS, which is provably secure. Unfortunately, this scheme is not practical due to the difficulties of finding a random decodable syndrome. In addition, the assumption of the indistinguishability of random binary Goppa codes has led to the emergence of attacks as described in [25]. One of the latest code-based signature schemes of this type is called Wave [41]. It is based on generalized $(U, U+V)$-codes. It is secure and more efficient than the CFS signature scheme. In addition, it has a smaller signature size than almost all finalist candidates in the NIST postquantum cryptography standardization process [42].

Apart from the FDH approach, it is possible to design signature schemes by applying the Fiat and Shamir transformation [43] to an identification protocol. To this end, one may use a code-based identification scheme like that of Stern [44], Jain et al. [45], or Cayrel et al. [46]. This approach, however, leads to a signature scheme with a very large signature size. To address this issue, Lyubashevsky's framework [47] can apparently be adapted. Unfortunately, almost all code-based signature schemes in Hamming metric designed by using this framework have been cryptanalyzed [48–53]. The only one that has remained secure so far is a rank metric-based signature scheme proposed by Aragon et al. [54].

In Figure 2, we recall Debris-Alazard et al.'s signature scheme (Wave), which is of our interest for this work. In Wave, the secret key is a tuple of three matrices $\mathsf{sk} = (\mathbf{S}, \mathbf{H}_{\mathsf{sk}}, \mathbf{P})$, where $\mathbf{S} \in \mathbb{F}_q^{r\times r}$ is an invertible matrix, $\mathbf{H}_{\mathsf{sk}} \in \mathbb{F}_q^{r\times n}$ is a parity-check matrix of a generalized $(U, U+V)$-code, and $\mathbf{P} \in \mathbb{F}_2^{n\times n}$ is a permutation matrix. The public key is a matrix $pk = \mathbf{H}_{\mathsf{pk}}$, where $\mathbf{H}_{\mathsf{pk}} = \mathbf{S}\mathbf{H}_{\mathsf{sk}}\mathbf{P}$. Steps for signature and verification processes are given in Figure 2. For additional details, the reader is referred to [41,55].

3 Signcryption and security model

In this section, we first recall the definition of signcryption followed by the signcryption tag-KEM framework and its security model under the insider setting.

3.1 Signcryption and its tag-KEM framework

Signcryption: A signcryption scheme is a tuple of algorithms $\mathsf{SC} = (\mathsf{Setup}, \mathsf{KeyGen}_s, \mathsf{KeyGen}_r, \mathsf{Signcrypt}, \mathsf{Unsigncrypt})$ [56], where:
* $\mathsf{Setup}(1^\lambda)$ is the common parameter generation algorithm with λ, the security parameter,
* KeyGen_s(resp. KeyGen_r) is a key-pair generation algorithm for the sender (resp. receiver),
* Signcrypt is the signcryption algorithm, and
* Unsigncrypt corresponds to the unsigncryption algorithm.

For more details on the design of signcryption, the reader is referred to [57] (Chap. 2, Sec. 3, p. 30).

Signcryption tag-KEM: A signcryption tag-KEM denoted by SCTKEM is a tuple of algorithms [16]:

$$\mathsf{SCTKEM} = (\mathsf{Setup}, \mathsf{KeyGen}_s, \mathsf{KeyGen}_r, \mathsf{Sym}, \mathsf{Encap}, \mathsf{Decap}),$$

where
- Setup is an algorithm for generating common parameters.
- KeyGen_s (resp. KeyGen_r) is the sender (resp. receiver) key generation algorithm. It takes as input the global information I and returns a private/public keypair $(\mathsf{sk}_s, \mathsf{pk}_s)$ (resp. $(\mathsf{sk}_r, \mathsf{pk}_r)$) that is used to send signcrypted messages.
- Sym is a symmetric key generation algorithm. It takes as input the private key of the sender sk_s and the public key of the receiver pk_r and outputs a symmetric key K together with internal state information ϖ.
- Encap takes as input the state information ϖ together with an arbitrary string τ, which is called a tag, and outputs an encapsulation E.

– Decap is the decapsulation/verification algorithm. It takes as input the sender's public key pk_s, the receiver's private key sk_r, an encapsulation E, and a tag τ. It returns either symmetric key K or the unique error symbol $\perp$.

Hybrid signcryption tag-KEM+DEM: It is simply a combination of a *sctkem* and a regular data encapsulation mechanism (DEM).

3.2 Insider security for signcryption tag-KEM

IND-CCA2 **game in signcryption tag**-KEM: It corresponds to a game between a challenger and a probabilistic polynomial-time (PPT) adversary $\mathcal{A}_{\mathsf{CCA2}}$ such that the latter tries to distinguish whether a given session key K is the one embedded in an encapsulation. During this game, $\mathcal{A}_{\mathsf{CCA2}}$ has adaptive access to three oracles for the attacked user corresponding to algorithms Sym, Encap, and Decap [16,17,57]. The game is described in Figure 3.

During Step 7, the adversary $\mathcal{A}_{\mathsf{CCA2}}$ is restricted not to make decapsulation queries on (E, τ) to the decapsulation oracle. The advantage of the adversary $\mathcal{A}$ is defined by

$$\mathsf{Adv}(\mathcal{A}_{\mathsf{CCA2}}) = |\Pr(b' = b) - 1/2|.$$

A signcryption tag-KEM is IND-CCA2 secure if, for any adversary $\mathcal{A}$, its advantage in the IND-CCA2 game is negligible with respect to the security parameter λ.

SUF-CMA **game for signcryption tag**-KEM: This game is a challenge between a challenger and a PPT adversary (i.e., a forger) $\mathcal{F}_{\mathsf{CMA}}$. In this game, the forger tries to generate a valid encapsulation E from the sender to any receiver, with adaptive access to the three oracles. The adversary is allowed to come up with the presumed secret key sk_r as part of his forgery [17] (Figure 4):

The adversary $\mathcal{F}_{\mathsf{CMA}}$ wins the SUF-CMA game if

$$\perp \neq \mathsf{Decap}(\mathsf{pk}_s, \mathsf{sk}_r, E, \tau)$$

Oracles

1. $\mathcal{O}_{\mathsf{Sym}}$ is the symmetric key generation oracle with input a public key pk, and computes $(K,\ \omega) = \mathsf{Sym}(\mathsf{sk}_s,\ \mathsf{pk})$. It then stores the value ω (hidden from the view of the adversary, and overwriting any previously stored values), and returns the symmetric key K.
2. $\mathcal{O}_{\mathsf{Encap}}$ is the key encapsulation oracle. It takes an arbitrary tag τ as input and checks whether there exists a stored value ω. If there is not, it returns $\perp$ and terminates. Otherwise, it erases the value from storage and returns $E = \mathsf{Encap}(\omega, \tau)$.
3. $\mathcal{O}_{\mathsf{Decap}}$ corresponds to the decapsulation/verification oracle. It takes an encapsulation E, a tag τ, any sender's public key pk as input and returns $\mathsf{Decap}(\mathsf{pk}, \mathsf{sk}_r, E, \tau)$.

IND-CCA2 Game for SCTKEM

1. $I := \mathsf{Setup}(1^\lambda)$
2. $(\mathsf{sk}_r, \mathsf{pk}_r) := \mathsf{KeyGen}_r(I)$
3. $(\mathsf{sk}_s, \mathsf{state}_1) := \mathcal{A}_{\mathsf{CCA2}}^{\mathcal{O}_{\mathsf{Sym}}, \mathcal{O}_{\mathsf{Encap}}, \mathcal{O}_{\mathsf{Decap}}}(\mathsf{pk}_r)$
4. $(K_1, \varpi) := \mathsf{Sym}(\mathsf{sk}_s, \mathsf{pk}_r)$, $K_0 \xleftarrow{\$} \mathcal{K}$ and $b \xleftarrow{\$} \{0,1\}$
5. $(\tau, \mathsf{state}_2) := \mathcal{A}_{\mathsf{CCA2}}^{\mathcal{O}_{\mathsf{Sym}}, \mathcal{O}_{\mathsf{Encap}}, \mathcal{O}_{\mathsf{Decap}}}(K_b, \mathsf{state}_1)$
6. $E := \mathsf{Encap}(\varpi, \tau)$
7. $b' := \mathcal{A}_{\mathsf{CCA2}}^{\mathcal{O}_{\mathsf{Sym}}, \mathcal{O}_{\mathsf{Encap}}, \mathcal{O}_{\mathsf{Decap}}}(E, \mathsf{state}_2)$

Figure 3: IND-CCA2 game [17].

SUF-CMA Game for SCTKEM
1. $I := \mathsf{Setup}(1^\lambda)$
2. $(\mathsf{sk}_s, \mathsf{pk}_s) := \mathsf{KeyGen}_r(I)$
3. $(E, \tau, \mathsf{sk}_r) := \mathcal{F}_{\mathsf{CMA}}^{\mathcal{O}_{\mathsf{Sym}}, \mathcal{O}_{\mathsf{Encap}}, \mathcal{O}_{\mathsf{Decap}}}(\mathsf{pk}_s)$

Figure 4: SUF-CMA game [17].

and the encapsulation oracle never returns E when he queries on the tag τ. The advantage of $\mathcal{F}_{\mathsf{CMA}}$ is the probability that $\mathcal{F}_{\mathsf{CMA}}$ wins the SUF-CMA game. A signcryption tag-KEM is SUF-CMA secure if the winning probability of the SUF-CMA game by $\mathcal{F}_{\mathsf{CMA}}$ is negligible.

Definition 1. A signcryption tag-KEM is said to be secure if it is IND-CCA2 and SUF-CMA secure.

3.3 Generic security criteria of hybrid signcryption tag-KEM+DEM

Security criteria for hybrid signcryption: The security of a hybrid signcryption tag-KEM+DEM depends on those of the underlying signcryption tag-KEM and DEM. However, it is important to note that in the standard model a signcryption tag-KEM is secure if it is both IND-CCA2 and SUF-CMA secure. Therefore, the generic security criteria for hybrid signcryption tag-KEM+DEM is given by the following theorem:

Theorem 1. [16,17] *Let* HSC *be a hybrid signcryption scheme constructed from a signcryption tag-*KEM *and a* DEM. *If the signcryption tag-*KEM *is* IND-CCA2 *secure and the* DEM *is one-time secure, then* HSC *is* IND-CCA2 *secure. Moreover, if the signcryption tag-*KEM *is* SUF-CMA *secure, then* HSC *is also* SUF-CMA *secure.*

4 Code-based hybrid signcryption

In this section, we first design a code-based signcryption tag-KEM scheme. Then we combine it with a one-time (OT) secure DEM for designing a hybrid signcryption tag-KEM+DEM scheme.

4.1 Code-based signcryption tag-KEM scheme

For designing our code-based signcryption tag-KEM scheme, we use the McEliece scheme as the underlying encryption scheme. More specifically, to achieve the IND-CCA2 security for our schemes, we use McEliece's scheme with the Fujisaki–Okamoto transformation [21,58]. The authors of ref. [21] gave an instantiation of this scheme using generalized Srivastava (GS) codes. Indeed, by using GS codes, it seems possible to choose secure parameters even for codes defined over relatively small extension fields. However, Barelli and Couvreur recently introduced an efficient structural attack [59] against some of the candidates in the NIST postquantum cryptography standardization process. Their attack is against code-based encryption schemes using some quasi-dyadic alternant codes with extension degree 2. It works specifically for schemes based on GS code called DAGS [20]. Therefore, in our work, we use the Goppa code with the Classic McEliece parameters. As for the underlying signature scheme, we use the code-based Wave [41] as described earlier.

The fact that we use Wave, the sender's secret key is a generalized $(U, U + V)$-code over a finite field $\mathbb{F}_q$ with $q > 2$. Its public key is a parity-check matrix of a code equivalent to the previous one. To reduce the public key size, we use a permuted Goppa subcode for the receiver's public key. Thus, we include the subcode equivalence problem as one of the security assumptions of our scheme. In Figure 5, we describe the algorithm Setup, which will provide common parameters for our scheme.

We give key generation algorithms in Figure 6, where we denote the sender key generation algorithm by KeyGen_s and that of the receiver by KeyGen_r. The receiver algorithm KeyGen_r returns as signcryption public key a generator matrix $\mathbf{G}_{\mathsf{pk},r} \in \mathbb{F}_2^{\tilde{k}\times n_r}$ of a Goppa subcode equivalent. It returns as signcryption secret key the tuple $(g_r, \Gamma_r, \mathbf{S}_r^{-1}, \mathbf{P}_r)$, where Γ_r and g_r are, respectively, the support and the polynomial of a Goppa code. $\mathbf{S}_r \in \mathbb{F}_2^{\tilde{k}\times k_r}$ is a full rank matrix and $\mathbf{P}_r$ a permutation matrix. The sender key generation algorithm KeyGen_s returns as private key three matrices $\mathbf{S}_s \in \mathbb{F}_3^{(n_s-k_s)\times(n_s-k_s)}$, $\mathbf{H}_{\mathsf{sk},s} \in \mathbb{F}_3^{(n_s-k_s)\times n_s}$, and $\mathbf{P}_s \in \mathbb{F}_2^{n_s\times n_s}$, where $\mathbf{S}_s \in \mathbb{F}_3^{(n_s-k_s)\times(n_s-k_s)}$ is an invertible matrix, $\mathbf{H}_{\mathsf{sk},s} \in \mathbb{F}_3^{(n_s-k_s)\times n_s}$ a parity-check matrix of a random generalized $(U, U+V)$-code and $\mathbf{P} \in \mathbb{F}_2^{n_s\times n_s}$ a permutation matrix. The sender public key is a parity-check matrix $\mathbf{H}_{\mathsf{pk},s} \in \mathbb{F}_3^{(n_s-k_s)\times n_s}$ of a generalized $(U, U+V)$ equivalent code given by $\mathbf{H}_{\mathsf{pk},s} = \mathbf{S}_s\mathbf{H}_{\mathsf{sk},s}\mathbf{P}_s$.

In Figure 7, we give the design of the symmetric key generation algorithm Sym of our scheme. The algorithm Sym takes as input the bit length ℓ of the symmetric encryption key. It outputs an internal state information ϖ and the session key K, where ϖ is randomly chosen from $\mathbb{F}_2^{\ell}$, and K is computed by using the hash function $\mathcal{H}_0$.

Figure 8 provides a description of the encapsulation and decapsulation algorithms of our signcryption tag-KEM scheme. We denote the encapsulation algorithm by Encap and the decapsulation by Decap. In the encapsulation algorithm, the sender first performs a particular Wave signature on the message $\boldsymbol{m} = \tau\|\varpi$, where ϖ corresponds to an internal state information and τ is the input tag. The signature in the Wave scheme comprises two parts: an error vector $\boldsymbol{e} \in \mathbb{F}_3^{n_s}$ and a random binary vector $\boldsymbol{y}$. In our scheme, $\boldsymbol{z}$ is the hash of a random coin $\boldsymbol{y} \in \mathbb{F}_2^{\kappa}$. The sender then performs an encryption of $\boldsymbol{m}' = \mathcal{H}_1(\tau)\|\varpi$. The encryption that we use in our scheme is the IND-CCA2 secure McEliece encryption scheme with the Fujisaki–Okamoto transformation introduced by Cayrel et al. [21]. During the encryption, the sender adaptively uses the random binary vector $\boldsymbol{y}$ as a random coin. The resulting ciphertext is denoted by $\boldsymbol{c}$. The output is given by $E = (\boldsymbol{e}, \boldsymbol{c})$.

In the decapsulation algorithm Decap, the receiver first performs recovery of the internal state information ϖ by using the algorithm Decrypt and the second part of the signature of $\boldsymbol{m}$. Then it verifies the signature and computes the session K by using ϖ.

The algorithm Decrypt that we use in the decapsulation algorithm of our scheme is described in Figure 9. It is similar to that described in [21], but we introduce some modifications which are:

Setup

Input: (1^λ)

Output:

- Parameters of sender's generalized $(U, U+V)$-code: code length n_s, dimension k_U of U, dimension k_V of V, dimension $k_s = k_U + k_V$ of the generalized $(U, U+V)$-code, weight of error vector ω, cardinality q of the finite field $\mathbb{F}_q$.
- Parameters of receiver's Goppa code: degree m of extension $\mathbb{F}_{2^m}$ of $\mathbb{F}_2$, length n_r of the Goppa code, degree t of Goppa polynomial g_r, dimension $\tilde{k}$ of Goppa subcode.
- A cryptographic hash functions $\mathcal{H}_1 : \{0,1\}^* \longrightarrow \{0,1\}^{\tilde{k}}$
- A cryptographic hash functions $\mathcal{H}_0 : \{0,1\}^* \longrightarrow \{0,1\}^{\ell}$ where ℓ is the bit length of the symmetric encryption key.
- A hash function $\mathcal{H}_2 : \{0,1\}^* \longrightarrow \{0,1,2\}^{r_s}$ where $r_s = n_s - k_s$.
- A cryptographic hash function $\mathcal{H}_3 : \{0,1\}^* \longrightarrow \{0,1\}^{\tilde{k}+\ell}$
- An encoding function $\phi : \mathbb{F}_2^{\kappa} \longrightarrow \mathcal{W}_{2,n_r,t}$ where κ is a well chosen parameters such that $\kappa = \left\lfloor \binom{n_r}{t} \right\rfloor$ and $\mathcal{W}_{2,n_r,t}$ is the set of binary vectors of length n_r and Hamming weight t.

Figure 5: Description of the Setup algorithm for common parameters.

KeyGen_r
Input: Integers m, n_r, t and $\tilde{k}$.
Output: sk_r and pk_r.
1. Randomly generate a monic irreducible polynomial $g_r(x) \in \mathbb{F}_{2^m}[x]$ of degree t
2. Select a uniform random set of n_r different elements $\Gamma_r = (a_0, ..., a_{n_r-1}) \in \mathbb{F}_{2^m}^{n_r}$.
3. Compute a parity-check matrix $\mathbf{H}_{\mathsf{sk},r} \in \mathbb{F}_2^{mt \times n_r}$ of the binary Goppa code from g_r and Γ_r.
4. Randomly choose a full rank matrix $\mathbf{S}_r \in \mathbb{F}_2^{\tilde{k} \times k_r}$ and permutation matrix $\mathbf{P}_r \in \mathbb{F}_2^{n_r \times n_r}$ with $k_r = n_r - mt$.
5. Set $\mathsf{sk}_r = (g_r, \Gamma_r, \mathbf{S}_r, \mathbf{P}_r)$ and $\mathsf{pk}_r = \mathbf{G}_{\mathsf{pk},r} = \mathbf{S}_r \mathbf{G}_{\mathsf{sk},r} \mathbf{P}_r$.
6. Return sk_r and pk_r.

KeyGen_s
Input: Integers n_s, k_U and k_V.
Output: sk_s and pk_s
1. Choose randomly a parity-check matrix $\mathbf{H}_{sk}$ of a code $(U, U+V)$ over $\mathbb{F}_3$ such that $\dim(U) = k_U$ and $\dim(V) = k_V$.
2. Randomly choose a full rank matrix $\mathbf{S} \in \mathbb{F}_3^{(n_s-k_s)\times(n_s-k_s)}$ and a monomial matrix $\mathbf{P} \in \mathbb{F}_3^{n_s \times n_s}$.
3. Set $\mathsf{sk}_s = (\mathbf{S}_s, \mathbf{H}_{\mathsf{sk}}, \mathbf{P}_s)$ and $\mathbf{H}_{\mathsf{pk},s} = \mathbf{S}_s \mathbf{H}_{\mathsf{sk},s} \mathbf{P}_s$.
4. Return sk_s and $\mathsf{pk}_s = \mathbf{H}_{\mathsf{pk},s}$.

Figure 6: Description of the key generation algorithms.

Sym
Input: The bit length ℓ of the symmetric encryption key.
Output: An internal state information ϖ and a session key K.
1. $\boldsymbol{x} \xleftarrow{\$} \mathbb{F}_2^{\ell}$
2. Compute $K := \mathcal{H}_0(\boldsymbol{x})$
3. Set $\varpi := \boldsymbol{x}$
4. Return (K, ϖ)

Figure 7: Description of the Sym algorithm.

- we use an encoding function $\boldsymbol{\phi}$,
- the output is not only the clear message $\boldsymbol{m}$, but a pair $(\boldsymbol{m}, \boldsymbol{y})$, where $\boldsymbol{y}$ is the reciprocal image the error vector σ by the encoding function $\boldsymbol{\phi}$.

4.1.1 Completeness of our signcryption tag-KEM

Let τ be a tag, $(\mathsf{sk}_s, \mathsf{pk}_s)$ (resp. sk_r and pk_r) be sender's (resp. receiver's) key pair generated by the algorithm KeyGen with input 1^λ. Let $(K, \varpi) \coloneqq \mathsf{Sym}(\mathsf{sk}_s, \mathsf{pk}_r)$ be a pair of a session key and an internal state information. Let $E \coloneqq (\boldsymbol{e}, \boldsymbol{c})$ be an encapsulation of the internal state information ϖ. Assuming that the encapsulation and decapsulation are performed by an honest user, we have:
– The receiver can recover the pair $(\tau' \| \varpi, \boldsymbol{y})$ from $\boldsymbol{c}$ and verify successfully that

Encap
Input: (ϖ, τ) with $\tau \in \mathbb{F}_2^{\ell}$
Output: An encapsulation of the internal state information ϖ.
1. $\boldsymbol{y} \xleftarrow{\$} \mathbb{F}_2^{\kappa}$ with $\kappa = \left\lfloor \log_2 \binom{n}{t} \right\rfloor$
2. Compute $\boldsymbol{z} = \mathcal{H}_1(\boldsymbol{y})$
3. Compute $\boldsymbol{s} := \mathcal{H}_2(\tau \| \varpi \| \boldsymbol{z})$
4. Compute $\tilde{\boldsymbol{e}} := \mathsf{Decode}_{\mathbf{H}_{\mathsf{sk},s}}(\boldsymbol{s}(\mathbf{S}^{-1})^T)$
5. Compute $\boldsymbol{e} := \tilde{\boldsymbol{e}}\mathbf{P}$
6. Compute $\tau' = \mathcal{H}_1(\tau)$
7. Compute $\boldsymbol{r} := \mathcal{H}_1(\tau' \| \varpi \| \boldsymbol{y})$
8. Compute $\boldsymbol{c}_0 := \boldsymbol{r}\mathbf{G}_{\mathsf{pk},r} + \sigma$, where $\sigma = \phi(\boldsymbol{y})$ with ϕ an constant weight encoding function.
9. Compute $\boldsymbol{c}_1 := \mathcal{H}_3(\sigma) \oplus (\tau' \| \varpi)$.
10. Parse $\boldsymbol{c} := (\boldsymbol{c}_0 \| \boldsymbol{c}_1)$
11. Return $E := (\boldsymbol{e}, \boldsymbol{c})$

Decap
Input: $(\mathsf{sk}_r, \mathbf{H}_{\mathsf{pk},s}, E, \tau)$
Output: Session key K
1. Parse E as $(\boldsymbol{e}, \boldsymbol{c})$.
2. Compute $(\boldsymbol{x}, \boldsymbol{y}) := \mathsf{Decrypt}(\mathsf{sk}_r, \boldsymbol{c})$
3. Parse $\boldsymbol{x}$ as $(\tilde{\tau} \| \tilde{\varpi})$
4. If $\boldsymbol{e}\mathbf{H}_{\mathsf{pk},s}^T \neq \mathcal{H}_2(\tau \| \tilde{\varpi} \| \mathcal{H}_1(\boldsymbol{y}))$ or $\tilde{\tau} \neq \mathcal{H}_1(\tau)$:
5. Return $\perp$
6. Compute $K := \mathcal{H}_0(\varpi)$
7. Return K.

Figure 8: Description of the Encap and Decap algorithms.

Decrypt
Input: Secrete $\mathsf{sk} = (g_r, \Gamma_r, \mathbf{S}_r, \mathbf{P}_r)$ the receiver and a ciphertext $\boldsymbol{c}$.
Output: The pair $(\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{x} = \tau' \| \varpi$.
1. Parse $\boldsymbol{c}$ into $(\boldsymbol{c}_0, \boldsymbol{c}_1)$
2. Compute $\sigma := \gamma_{\text{Goppa}}^{\text{McE}}(\mathsf{sk}, \boldsymbol{c}_0)$, where $\gamma_{\text{Goppa}}^{\text{McE}}$ is a decoding algorithm for Goppa code.
3. $\boldsymbol{y} = \phi^{-1}(\sigma)$
4. Compute $\boldsymbol{x} = \boldsymbol{c}_1 \oplus \mathcal{H}_3(\sigma)$
5. Compute $\tilde{\boldsymbol{r}} = \mathcal{H}_1(\boldsymbol{x} \| \boldsymbol{y})$
6. If $\tilde{\boldsymbol{r}}\mathbf{G}_{\mathsf{pk},r} \oplus \sigma \neq \boldsymbol{c}_1$:
7. Return $\perp$
8. Return $(\boldsymbol{x}, \boldsymbol{y})$

Figure 9: Description of the Sym algorithm.

$$\boldsymbol{e}\mathbf{H}_{\mathsf{pk},s}^T = \mathcal{H}_2(\tau \| \varpi | \boldsymbol{y}) \quad \text{and} \quad \tau' = \mathcal{H}_1(\tau).$$

Otherwise, the receiver performs a successful signature verification of message $\boldsymbol{m} := \tau \| \varpi$ signed by an honest user using the dual version of mCFS signature.

– Therefore, it can compute the session key $K := \mathcal{H}_0(\varpi)$.

4.2 Code-based hybrid signcryption

Here, we use the signcryption tag-KEM described in Section 4.1 for designing a code-based hybrid signcryption. For the data encapsulation, we propose the use of a regular OT-secure symmetric encryption scheme. We denote the symmetric encryption algorithm being used by SymEncrypt and the symmetric decryption algorithm by SymDecrypt.

Figure 10 gives the design of our code-based hybrid signcryption tag-KEM+DEM. In this design, algorithms Setup, KeyGen_s, and KeyGen_r are the same as those of our signcrytion tag-KEM. Algorithms Sym and Encap are those of our signcryption tag-KEM in Section 4.1.

5 Security analysis

Before discussing the security of our hybrid scheme, let us consider the following assumptions for our security analysis:

Assumption 1: The advantage of PPT algorithm $\mathcal{A}$ to solve the decoding random linear codes problem is negligible with respect to the length n and dimension k of the code.

Assumption 2: The advantage of PPT algorithm $\mathcal{A}$ to solve the $(U, U + V)$ distinguishing problem is negligible with respect to the length n and dimension k of the code.

Assumption 3: The advantage of PPT algorithm $\mathcal{A}$ to solve the subcode equivalence problem is negligible with respect to the length n and dimension k of the code.

Assumption 4: The advantage of PPT algorithm $\mathcal{A}$ to solve the DOOM problem is negligible with respect to the length n and dimension k of the code.

Assumption 5: The advantage of PPT algorithm $\mathcal{A}$ to solve the Goppa code distinguishing problem is negligible with respect to the length n and dimension k of the code.

5.1 Information-set decoding algorithm

In code-based cryptography, the best-known nonstructural attacks rely on information-set decoding. The information-set decoding algorithm was introduced by Prange [60] for decoding cyclic codes. After the publication of Prange's work, there have been several works studying to invert code-based encryption schemes based on information-set decoding (see [61] Section 4.1).

<u>Signcrypt</u>
Input: A three tuple $(\mathsf{sk}_s,\ \mathsf{pk}_r,\ \boldsymbol{m})$
Output: The signcrypted message $\boldsymbol{c} = (E,\ C)$.
1. Compute $(K,\ \varpi)=\mathsf{Sym}(\mathsf{sk}_s,\ \mathsf{pk}_r)$
2. Compute $C = \mathsf{SymEncrypt}(K,\ \boldsymbol{m})$
3. Compute $E = \mathsf{Encap}(\varpi,\ C)$
4. Return (E, C)

<u>Unsigncrypt</u>
Input: A three tuple $(\mathsf{sk}_r,\ \mathsf{pk}_s,\ (\boldsymbol{c}, C))$
Output: The clear text $\boldsymbol{m}$
1. If $\mathsf{Decap}(\mathsf{sk}_r,\ \mathsf{pk}_s,\ \boldsymbol{c})=\perp$ return $\perp$
2. Compute $\boldsymbol{m} = \mathsf{SymDecrypt}(K, C)$
3. Return $\boldsymbol{m}$

Figure 10: Code-based hybrid signcryption from *sctkem* and DEM.

For a given linear code of length n and dimension k, the main idea behind the information-set decoding algorithm is to find a set of k coordinates of a garbled vector that are error free and such that the restriction of the code's generator matrix to these positions is invertible. Then, the original message can be computed by multiplying the encrypted vector by the inverse of the submatrix.

Thus, those k bits determine the codeword uniquely, and hence, the set is called an information set. It is sometimes difficult to draw the exact resistance to this type of attack. However, they are always lower-bounded by the ratio of information sets without errors to total possible information sets, i.e.,

$$R_{\mathrm{ISD}} = \frac{\binom{n-\omega}{k}}{\binom{n}{k}}, \tag{1}$$

where ω is the Hamming weight of the error vector. Therefore, well-chosen parameters can avoid these nonstructural attacks. In our scheme, we use the parameters of the Wave signature [41] for the sender and those of Classic McEliece [61] for the receiver in the underlying encryption scheme.

5.2 Key recovery attack

In code-based cryptography, usually, the first step in the key recovering attack is to perform a distinguishing attack on the public code in order to identify the family of the underlying code. Once successful, the attacker can then perform any well-known attack against this family of underlying codes to recover the secret key. When the underlying code is a Goppa code, the main distinguishing attack technique consists of evaluating the square code or the square of the trace code of the corresponding public code [25,62,63]. Note that this technique usually works for a Goppa code with a high rate. Compared to many other code-based encryption schemes, in which the public code is equivalent to an alternant or a Goppa code, in this work the public code is a permuted Goppa subcode. Thus, in addition to the indistinguishability of Goppa codes, the subcode equivalence problem becomes one of our security assumptions. Moreover, to the best of our knowledge, there is no attack reported in the literature on distinguishing a code equivalent to a Goppa subcode. Therefore, by using the subcode equivalence problem as a security assumption, we can keep our scheme out of the purview of the distinguishing attack even though the underlying code is a Goppa code.

Throughout the rest of our analysis, we assume that the attacker knows that the family of the underlying code is a Goppa code. In our case, the key recovery attack is at two different levels: the first one is on the sender side and the second one is on the receiver side.

On the receiver side, the key recovery attack consists of the recovery of the Goppa polynomial g_r and the support $\gamma_r = (\alpha_0, \ldots, \alpha_{n-1})$ from the public matrix. Therefore, the natural way for this is to perform a brute-force attack: one can determine the sequence $(\alpha_0, \ldots, \alpha_{n-1})$ from g_r and the set $\{\alpha_0, \ldots, \alpha_{n-1}\}$, or alternatively determine g_r from $(\alpha_0, \ldots, \alpha_{n-1})$. A good choice of parameters can avoid this attack for the irreducible Goppa code the number of choices of g_r is given by

$$\frac{1}{t}\sum_{d|t}\mu(d)q^{\frac{t}{d}}.$$

By using the parameters of Classic McEliece, we can see that the complexity for performing a brute-force attack to find Goppa polynomial is more than 2^{800} for the parameters proposed in [61].

It is also important to note that if the adversary has the knowledge of the underlying Goppa code C_{sk}, performing the key recovery attack implies solving a computational instance of a subcode equivalence problem. Indeed, this corresponds to finding the permutation σ such that $\sigma(C_{\mathsf{pk}})$ is a subcode of C_{sk}. We can see that finding the permutation σ is equivalent to solving the following system:

$$\mathbf{G}_{\mathsf{pk},r}\mathbf{X}_{\sigma}\mathbf{H}_{\mathsf{sk},r} = \mathbf{0} \tag{2}$$

where $\mathbf{H}_{\mathsf{sk},r}$ is a parity-check matrix of the underlying Goppa code $C_{\mathsf{sk},r}$, $\mathbf{G}_{\mathsf{sk},r}$ is the generator matrix of the public code C_{pk} and $\mathbf{X}_{\sigma} = (x_{i,j})$ is the matrix of the unknown permutation σ. Note that solving (2) is equivalent to solving a variant of permuted kernel problem [64]. A natural way to solve (2) is to use the brute force attack, and such an attack is of order $O(n!)$. However, the adversary could use Georgiades' technique [65], where its complexity is given in our case by

$$O\left(\frac{n!}{\tilde{k}!}\right). \tag{3}$$

Recently, Paiva and Terada introduced in [66] a new technique for solving (2). The workfactor of their attack applied to our scheme is given by:

$$\mathsf{WF}_{\mathsf{Attack}_{\mathsf{PaTe}}} = O\left(2^{(n-mt-\tilde{k}n^{-1/5})(\lceil \log(n)\rceil-1)-0.91n+\frac{\log n}{2}}\right). \tag{4}$$

From (3) and (4), we can see that a well-chosen set of parameters can avoid the attack of Georgiades as well as that of Paiva and Terada.

In the case of the sender, the key recovery attack consists of first solving the $(U, U+V)$ distinguishing problem for finite fields of cardinality $q = 3$. Therefore, under Assumption 3 and with a well-chosen set of parameters, this attack would fail.

5.3 IND-CCA and SUF-CMA security

In code-based cryptography, the main approach to a chosen-ciphertext attack against the McEliece encryption scheme consists of adding two errors to the received word. If the decryption succeeds, it means that the error vector in the resulting word has the same weight as the previous one. In our signcryption tag-KEM scheme, this implies either recovering the session key K or distinguishing encapsulation of two different session keys from $(\boldsymbol{e}, \boldsymbol{c}, \tau)$. We see that the recovery of the session key K corresponds to the recovery of plaintext in a IND - CCA2 secure version of McEliece's cryptosystem (see [21], Subsection 3.2). We now have the following theorem:

Theorem 2. *Under Assumptions 1, 3, and 5, the signcryption tag-*KEM *scheme described in Section* 4.1 *is* IND - CCA2 *secure.*

Proof. Let $\mathcal{A}_{\mathsf{CCA2}}$ be a PPT adversary against the signcryption tag-KEM scheme described in Section 4.1 in the signcryption tag-KEMIND - CCA2 game. Let us denote its advantage by $\varepsilon_{\mathsf{CCA2,SCTKEM}}$. For proving Theorem 2, we need to bound $\varepsilon_{\mathsf{CCA2,SCTKEM}}$.

Game 0: This game is the normal signcryption tag-KEM IND - CCA2 game. Let us denote by X_0 the event that the adversary wins Game 0 and $\mathsf{Pr}(X_0)$ the probability that it happens. Then we have

$$\mathsf{Pr}(X_0) = \varepsilon_{\mathsf{CCA2,SCTKEM}}.$$

Game 1: This game corresponds to the simulation of the hash function oracle. Indeed, it is the same as Game 0 except that adversary can have access to the hash function oracle: It looks for some pair $(\tau^*, \boldsymbol{y}^*) \in \mathbb{F}_2^{\lambda} \times \mathbb{F}_2^{\kappa}$ such that $\boldsymbol{e}\mathbf{H}_s^T = \mathcal{H}_2(\tau^*\|\varpi\|\mathcal{H}_1(\boldsymbol{y}^*))$. Then, it tries to continue by computing $\boldsymbol{c}'$. We can see that it could succeed at least when the following collisions happen:

$$\mathcal{H}_1(\tau^*) = \mathcal{H}_1(\tau) \quad \text{and} \quad \mathcal{H}_1(\tau^*\|\varpi\|\mathcal{H}_2(\boldsymbol{y}^*)) = \mathcal{H}_2(\tau\|\varpi\|\mathcal{H}_1(\boldsymbol{y})).$$

Therefore, if q_h is the number of queries allowed and X_1 the event that $\mathcal{A}_{\mathsf{CCA2}}$ wins game X_1, then we have:

$$|\mathsf{Pr}(X_1) - \mathsf{Pr}(X_0)| \le \frac{q_h}{\binom{n}{t}}.$$

Game 2: This game is the same as Game 1 except that the error vector $\boldsymbol{e}$ in the encapsulation output is generated randomly. We can see that the best to proceed is to split $\boldsymbol{c}$ as $(\boldsymbol{c}_0\|\boldsymbol{c}_1)$ and then try to invert either $\boldsymbol{c}_0$ for recovering the error σ or $\boldsymbol{c}_1$ for recovering directly the internal state ϖ_b. That means that the adversary is able either to solve the syndrome decoding problem or to invert a one-time pad function. Therefore, we have:

$$|\mathsf{Pr}(X_1) - \mathsf{Pr}(X_2)| \leq \varepsilon_{\mathsf{SD}} + \nu(\ell),$$

where $\varepsilon_{\mathsf{SD}}$ is the advantage of an adversary against the syndrome decoding problem, ν is a negligible function, and ℓ is the bit length of the symmetric encryption.

Game 3: This game is the same as Game 2. However, the change is in the key generation algorithm. Indeed, a random code is chosen as the underlying code instead of Goppa. We can see that this change is indistinguishable. In fact, distinguishing this change corresponds to solving in part the Goppa code distinguishing problem. Thus, we have

$$|\mathsf{Pr}(X_3) - \mathsf{Pr}(X_2)| \leq \varepsilon_{\mathrm{GCD}}(\lambda),$$

where $\varepsilon_{\mathrm{GCD}}(\lambda)$ is the advantage of a PPT adversary in the Goppa code distinguishing problem and λ the security parameter. If there is a PPT adversary $\mathcal{A}$ capable of distinguishing this change, we can use it to construct an adversary $\mathcal{A}_{\mathrm{GCD}}$ to solve the Goppa code distinguishing problem as follows:

1. Once receiving an instance $\mathbf{G} \in \mathbb{F}_2^{k\times n}$ of a generator matrix of a code C in Goppa code distinguishing problem, $\mathcal{A}_{\mathrm{GCD}}$ extracts a generator matrix $\mathbf{G}'$ of a subcode C' of C and forward it to $\mathcal{A}$.
2. $\mathcal{A}$ will reply by 1 if the change has happened, i.e., the underlying code is not a Goppa code. It will reply by 0 otherwise.
3. If $\mathcal{A}_{\mathrm{GCD}}$ receives 1 from $\mathcal{A}$, it means that C is not a Goppa code and $\mathcal{A}_{\mathrm{GCD}}$ outputs 0, otherwise it returns 1, i.e., C is a Goppa code.

Game 4: This game is the same as Game 3 except that the public key is a random matrix instead of a generator matrix of a permuted subcode. We can see that this change is indistinguishable according to the subcode equivalence assumption. Thus, we have:

$$|\mathsf{Pr}(X_4) - \mathsf{Pr}(X_3)| \leq \varepsilon_{\mathrm{ES}}(\lambda),$$

where $\varepsilon_{\mathrm{ES}}(\lambda)$ is the advantage of a PPT adversary in the subcode equivalence problem and λ is the security parameter. Moreover, we can show that if an adversary $\mathcal{A}_{\mathsf{CCA2}}$ wins this game, we can use it to construct an adversary $\mathcal{A}_{\mathrm{McE}}$ for attacking the underlying McEliece scheme in the public key encryption IND-CCA2 game (called PKE.Game in Appendix A). For more details on the underlying McEliece encryption scheme and its IND-CCA2 security proof, the reader is referred to Appendix C. We now proceed as follows:

- Given the receiver public key pk, which corresponds to a receiver public key signcryption tag-KEM, $\mathcal{A}_{\mathrm{McE}}$ does the following:
  - ⋆ chooses randomly $(\varpi_0, \varpi_1) \xleftarrow{\$} \mathbb{F}_2^{\ell}$
  - ⋆ chooses randomly $\delta \xleftarrow{\$} \{0, 1\}$
  - ⋆ sends the public key pk and ϖ_δ to $\mathcal{A}_{\mathsf{CCA2}}$
- Given a tag τ from $\mathcal{A}_{\mathsf{CCA2}}$, $\mathcal{A}_{\mathrm{McE}}$:
  - ⋆ sends the pair $(\mathcal{H}_1(\tau)\|\varpi_0, \mathcal{H}_1(\tau)\|\varpi_1)$ to the encryption oracle of PKE.Game
  - ⋆ forwards $\boldsymbol{c}$ received from the encryption oracle to $\mathcal{A}_{\mathsf{CCA2}}$
- For every decryption query $(\boldsymbol{c}_i, \tau_i)$ from $\mathcal{A}_{\mathsf{CCA2}}$:
  - ⋆ if $\boldsymbol{c}_i = \boldsymbol{c}$, $\mathcal{A}_{\mathrm{McE}}$ return $\perp$ to $\mathcal{A}_{\mathsf{CCA2}}$, otherwise it sends $\boldsymbol{c}_i$ to the decryption oracle of PKE.Game.
  - ⋆ Receiving $\tau_i'\|\varpi_i$ from the decryption oracle:
  - ▷ if $\tau_i' \neq \mathcal{H}_1(\tau_i)$, $\mathcal{A}_{\mathrm{McE}}$ returns $\perp$ to $\mathcal{A}_{\mathsf{CCA2}}$, otherwise, it returns ϖ_i to $\mathcal{A}_{\mathsf{CCA2}}$.
- When $\mathcal{A}_{\mathsf{CCA2}}$ outputs $\tilde{\delta} = \delta$, $\mathcal{A}_{\mathrm{McE}}$ returns 1, otherwise, it returns 0.

Let $\varepsilon_{\mathrm{PKE}}$ be the advantage of $\mathcal{A}_{\mathrm{McE}}$ in the PKE.Game. Note that the target ciphertext $\boldsymbol{c}$ can be uniquely decrypted to $\mathcal{H}_1(\tau)\|\varpi_\delta$. Therefore, any $(\boldsymbol{c}, \tau')$ other than $(\boldsymbol{c}, \tau)$ cannot be a valid signcryption ciphertext unless collusion of $\mathcal{H}_1$ takes place, i.e., $\mathcal{H}_1(\tau_i) = \mathcal{H}_1(\tau)$. The correct answer to any decryption query with $\boldsymbol{c}_i = \boldsymbol{c}$ is $\perp$. Decryption queries from $\mathcal{A}_{\mathsf{CCA2}}$ are correctly answered since $\boldsymbol{c}_i$ is decrypted by the decryption oracle of PKE.Game.

When $\mathcal{A}_{\mathsf{CCA2}}$ outputs $\tilde{\delta}$, it means that ϖ_δ is embedded in $\boldsymbol{c}_i$ otherwise $\varpi_{1-\delta}$ is embedded. It means that the adversary $\mathcal{A}_{\mathrm{McE}}$ wins game PKE.Game with the same probability as $\mathcal{A}_{\mathsf{CCA2}}$ wins Game 4 when collision of $\mathcal{H}_1$ has happened. Let $\tilde{X}$ be the event collision of $\mathcal{H}_1$ has happened and $\tilde{X}_4$ the event $\mathcal{A}_{\mathrm{McE}}$ wins the PKE.Game. Let us denote by $\varepsilon_{\mathrm{pke}}$ the probability of the event $\tilde{X}_4$ and ε_{col} that of $\tilde{X}$. Therefore, we have:

$$\Pr(X_4|\tilde{X}) = \Pr(\tilde{X}_4) \Rightarrow \Pr(X_4) \leq \Pr(\tilde{X}_4) + \Pr(\tilde{X}).$$

By putting it all together, we conclude our proof. □

Theorem 3. *Under Assumptions 2 and 4, the signcryption tag-*KEM *scheme described in Section* 4.1 *is* SUF-CMA *secure.*

Proof. Let $\mathcal{F}_{\mathsf{CMA}}$ be an adversary against our signcryption tag-KEM in the SUF-CMA game and $\varepsilon_{\mathsf{CMA}}$ its advantage. For the forgery of our signcryption, adversary $\mathcal{F}_{\mathsf{CMA}}$ needs to first find a pair $(\boldsymbol{e}, \boldsymbol{y}) \in \mathcal{W}_{q,n,\omega} \times \mathbb{F}_2^{\tilde{k}}$ such that $\boldsymbol{e}\mathbf{H}_{\mathsf{pk},s}^T = \mathcal{H}_2(\tau\|\varpi\|\boldsymbol{y})$. Then, it will try to find $\boldsymbol{r} \in \mathbb{F}_2^{\kappa}$ such that $\mathcal{H}_1(\boldsymbol{r}) = \boldsymbol{y}$, i.e., it wins in the target pre-image free game (see Appendix B) against the cryptographic hash function $\mathcal{H}_1$. We can see that finding $(\boldsymbol{e}, \boldsymbol{y}) \in \mathcal{W}_{q,n,\omega} \times \mathbb{F}_2^{\tilde{k}}$ such that $\boldsymbol{e}\mathbf{H}_{\mathsf{pk},s}^T = \mathcal{H}_2(\tau\|\varpi\|\boldsymbol{y})$ corresponds to the forgery of the underlying Wave signature scheme. Let $\varepsilon_{\mathrm{PreIm}}$ be the advantage of an adversary in the pre-image free game against a cryptographic hash function. Let $\mathcal{A}_{\mathrm{Wave},\mathsf{CMA}}$ be an adversary against the Wave signature in the EUF-CMA game and $\varepsilon_{\mathrm{Wave,EUF}}$ its advantage. Let X be the event that $\mathcal{A}_{\mathrm{Wave},\mathsf{CMA}}$ wins. Let $\tilde{X}$ be the event that the adversary is able to find a pre-image $\boldsymbol{x}$ of $\boldsymbol{y}$ by $\mathcal{H}_1$ such that $\boldsymbol{x} \in \mathbb{F}_2^{\kappa}$. We have:

$$\Pr(\mathcal{F}_{\mathsf{CMA}}\ \text{wins}) = \Pr(X \quad \text{and} \quad \tilde{X}) \leq \Pr(X) + \Pr(\tilde{X}) \leq \varepsilon_{\mathrm{Wave},\mathsf{EUF}} + \frac{\varepsilon_{\mathrm{PreIm}}}{2^{\kappa}}.$$

Note that due to the fact that $\mathcal{H}_1$ is a cryptographic hash function, $\varepsilon_{\mathrm{PreIm}}$ is negligible and that concludes our proof. □

Corollary 1. *The signcryption tag-*KEM *described in* Section 4.1 *is secure.*

The aforementioned corollary is a consequence of Theorems 2 and 3. We then have the following.

Proposition 1. *Under Assumptions 1, 3, and 5, the hybrid signcryption tag-*KEM+DEM *scheme described in* Section 4.2 *is* IND-CCA2.

Proof. Proposition 1 is a consequence of Theorem 1. Indeed, under Assumptions 1, 3, and 5, the underlying signcryption tag-KEM is IND-CCA2 secure (Theorem 2). In addition, the symmetric encryption scheme used is OT-secure. Therefore, a direct application of Theorem 1 allows us to achieve the proof. □

Proposition 2. *Under Assumptions 2 and 4, the hybrid signcryption tag-*KEM+DEM *scheme described in* Section 4.2 *is* SUF-CMA *secure.*

Proof. Under Assumptions 2 and 4, the underlying signcryption tag-KEM is SUF-CMA secure and, therefore, according to the Theorem 1, the proposed hybrid signcryption tag-KEM+DEM is SUF-CMA secure. □

6 Parameter values

For our scheme, we choose parameters such that $\lambda_0 = \lambda + 2\log_2(q_{\mathsf{sign}})$ and λ_{McE} of the underlying Wave signature and McEliece's encryption, respectively, satisfy $\max(\lambda_0, \lambda_{\mathrm{McE}}) \leq \left\lfloor \binom{n_r}{t} \right\rfloor$. According to the sender and receiver keys, the size of our ciphertext is given by

Table 1: Parameter values of the proposed scheme

Parameter	n_s	k_U	k_V	ω	m	t	n_r	$\tilde{k}$	ℓ
Value	8,492	3,558	2,047	7,980	12	64	3,488	1,815	512

Table 2: Key sizes of the proposed scheme

User	Public key	Secret key
Receiver's key size	$\tilde{k}n_r$	$m(2n_r + t - \tilde{k}t) + \tilde{k}n_r$
Sender's key size	$r(n_s - r)\log_2(q)$	$(n_s(n_s + r) + r^2)\log_2(q)$

Table 3: Size comparison (in bits) of the proposed scheme with the lattice-based schemes of [9, 67,68]

Construction	Receiver's key size		Sender's key size		Ciph. size
	Pub. key	Sec. key	Pub. key	Sec. key	
SC_{TK} [9,67]	8.5×10^7	4.2×10^8	8.4×10^7	4.2×10^8	5.5×10^5
SC_{KEM} [9,67]	5.7×10^7	4.2×10^8	8.5×10^7	4.2×10^8	5.2×10^5
SC_{CHK} [9,68]	2.8×10^7	4.2×10^8	2.8×10^7	4.2×10^8	4.5×10^6
Sato and Shikata [9]	2.8×10^7	4.2×10^8	2.8×10^7	4.2×10^8	4.0×10^5
Our scheme	6.3×10^6	5.0×10^6	2.6×10^7	1.7×10^8	2.1×10^4

$$|E| = |\boldsymbol{e}| + |\boldsymbol{c}| + |C| = 2n_s + n_r + \tilde{k} + 2\ell.$$

Table 1 gives suggested values of the parameters of our scheme. These values have been derived using those of Wave [42] and Classic McEliece [61] for NIST PQC Level 1 security. According to the values given in Table 1, the ciphertext size in bits of our scheme is in the order of $|E| = 2.9 \times 10^4$.

Table 2 provides key sizes of our scheme in terms of relevant parameters. Then, in Table 3, we give a numerical comparison of key and ciphertext sizes of our scheme with some existing lattice-based hybrid signcryption schemes. The rationale behind comparing our scheme against lattice-based schemes is that no code-based hybrid signcryption scheme exists in the literature and the underlying hard problems in both codes- and lattice-based schemes are considered quantum safe. For the lattice-based schemes in our comparison, the parameters, including plaintext size of 512 bits, are from [9, Table 2]. We can see that for postquantum security level 1, the proposed scheme has the smallest key and ciphertext sizes.

7 Conclusion

In this article, we have proposed a new signcryption tag-KEM based on the coding theory. The security of our scheme relies on known hard problems in coding theory. We have used the proposed signcryption scheme to design a new code-based hybrid signcryption tag-KEM+DEM. We have proven that the proposed schemes are IND-CCA2 and SUF-CMA secure against any PPT adversary. The proposed scheme has a smaller ciphertext size compared to the pertinent lattice-based schemes.

Acknowledgement: The authors would like to thank the anonymous reviewers for their comments on an earlier version of this article.

Funding information: This work was supported by Ripple Impact Fund/Silicon Valley Community Foundation (Grant 2018-188473).

Conflict of interest: The authors state that there is no conflict of interest.

References

[1] Zheng Y. Digital signcryption or how to achieve cost (signature & encryption)≪ cost (signature) + cost (encryption). In: Advances in Cryptology–CRYPTO'97: 17th Annual International Cryptology Conference Santa Barbara, California, USA August 17–21, 1997 Proceedings. Springer; 1997. p. 165–79.
[2] Zheng Y, Imai H. How to construct efficient signcryption schemes on elliptic curves. Inform Process Lett. 1998;68(5):227–33.
[3] Steinfeld R, Zheng Y. A signcryption scheme based on integer factorization. In: Information Security: Third International Workshop, ISW 2000 Wollongong, Australia, December 20–21, 2000 Proceedings. Springer; 2000. p. 308–22.
[4] Yang X, Cao H, Li W, Xuan H. Improved lattice-based signcryption in the standard model. IEEE Access. 2019;7:155552–62.
[5] Li F, BinMuhaya FT, Khan MK, Takagi T. Lattice-based signcryption. Concurrency Computation Practice Experience. 2013;25(14):2112–22.
[6] Barreto PS, Libert B, McCullagh N, Quisquater JJ. Signcryption schemes based on the Diffie-Hellman problem. In: Practical Signcryption. Information Security and Cryptography. Berlin, Heidelberg: Springer; 2010. p. 57–69.
[7] Barreto PS, Libert B, McCullagh N, Quisquater JJ. Signcryption schemes based on bilinear maps. In: Practical Signcryption. Information Security and Cryptography. Berlin, Heidelberg: Springer; 2010. p. 71–97.
[8] Dent AW, Malone-Lee J. Signcryption schemes based on the RSA problem. In: Practical Signcryption. Berlin, Heidelberg: Springer; 2010. p. 99–117.
[9] Sato S, Shikata J. Lattice-based signcryption without random oracles. In: Post-Quantum Cryptography: 9th International Conference, PQCrypto 2018, Fort Lauderdale, FL, USA, April 9–11, 2018, Proceedings. Springer; 2018. p. 331–51.
[10] Yan J, Wang L, Wang L, Yang Y, Yao W. Efficient lattice-based signcryption in standard model. Math Problem Eng. 2013;2013:1–18.
[11] Le HQ, Duong DH, Roy PS, Susilo W, Fukushima K, Kiyomoto S. Lattice-based signcryption with equality test in standard model. Comput Standard Interfaces. 2021;76:103515.
[12] Zhao X, Wang X. An efficient identity-based signcryption from lattice. Int J Security Appl. 2014;8(2):363–74.
[13] Dent AW. Hybrid cryptography. Cryptology ePrint Archive. 2004.
[14] Dent AW. Hybrid signcryption schemes with insider security. In: Information Security and Privacy: 10th Australasian Conference, ACISP 2005, Brisbane, Australia, July 4–6, 2005. Proceedings 10. Springer; 2005. p. 253–66.
[15] Dent AW. Hybrid signcryption schemes with outsider security. In: Information Security: 8th International Conference, ISC 2005, Singapore, September 20–23, 2005. Proceedings 8. Springer; 2005. p. 203–17.
[16] Bjørstad TE, Dent AW. Building better signcryption schemes with tag-KEMs. In: Public Key Cryptography-PKC 2006: 9th International Conference on Theory and Practice in Public-Key Cryptography, New York, NY, USA, April 24–26, 2006. Proceedings 9. Springer; 2006. p. 491–507.
[17] Yoshida M, Fujiwara T. On the security of tag-KEM for signcryption. Electr Notes Theoret Comput Sci. 2007;171(1):83–91.
[18] Mathew KP, Vasant S, Rangan CP. On provably secure code-based signature and signcryption scheme. IACR Cryptol ePrint Archive. 2012;2012:585.
[19] Song Y, Li Z, Li Y, Li J. Attribute-based signcryption scheme based on linear codes. Inform Sci. 2017;417:301–9.
[20] Banegas G, Barreto PS, Boidje BO, Cayrel PL, Dione GN, Gaj K, et al. DAGS: Key encapsulation using dyadic GS codes. J Math Cryptol. 2018;12(4):221–39.
[21] Cayrel PL, Hoffmann G, Persichetti E. Efficient implementation of a CCA2-secure variant of McEliece using generalized Srivastava codes. In: Public Key Cryptography-PKC 2012: 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21–23, 2012. Proceedings 15. Springer; 2012. p. 138–55.
[22] Berlekamp E, McEliece R, Van Tilborg H. On the inherent intractability of certain coding problems (corresp.). IEEE Trans Inform Theory. 1978;24(3):384–6.
[23] Johansson T, Jonsson F. On the complexity of some cryptographic problems based on the general decoding problem. IEEE Trans Inform Theory. 2002;48(10):2669–78.
[24] Sendrier N. Decoding one out of many. In: Post-Quantum Cryptography: 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29–December 2, 2011. Proceedings 4. Springer; 2011. p. 51–67.
[25] Faugère JC, Gauthier-Umana V, Otmani A, Perret L, Tillich JP. A distinguisher for high-rate McEliece cryptosystems. IEEE Trans Inform Theory. 2013;59(10):6830–44.

[26] Debris-Alazard T, Sendrier N, Tillich JP. The problem with the SURF scheme. 2017. arXiv: http://arXiv.org/abs/arXiv:170608065.
[27] Berger TP, Gueye CT, Klamti JB. A NP-complete problem in coding theory with application to code-based cryptography. In: Codes, Cryptology and Information Security: Second International Conference, C2SI 2017, Rabat, Morocco, April 10–12, 2017, Proceedings-In Honor of Claude Carlet. Springer; 2017. p. 230–7.
[28] McEliece RJ. Jet Propulsion Laboratory. A public-key cryptosystem based on algebraic coding theory. DSN progress Report. 1978:42–4.
[29] Berger TP, Loidreau P. How to mask the structure of codes for a cryptographic use. Des Codes Crypt. 2005;35:63–79.
[30] Berger TP, Cayrel PL, Gaborit P, Otmani A. Reducing key length of the McEliece cryptosystem. In: Progress in Cryptology-AFRICACRYPT 2009: Second International Conference on Cryptology in Africa, Gammarth, Tunisia, June 21–25, 2009. Proceedings 2. Springer; 2009. p. 77–97.
[31] Misoczki R, Barreto PS. Compact McEliece keys from Goppa codes. In: Selected Areas in Cryptography: 16th Annual International Workshop, SAC 2009, Calgary, Alberta, Canada, August 13–14, 2009, Revised Selected Papers 16. Springer; 2009. p. 376–92.
[32] Misoczki R, Tillich JP, Sendrier N, Barreto PS. MDPC-McEliece: New McEliece variants from moderate density parity-check codes. In: 2013 IEEE International Symposium on Information Theory. IEEE; 2013. p. 2069–73.
[33] Barreto PS, Lindner R, Misoczki R. Monoidic codes in cryptography. PQCrypto. 2011;7071:179–99.
[34] Persichetti E. Compact McEliece keys based on quasi-dyadic Srivastava codes. J Math Cryptol. 2012;6(2):149–69.
[35] Niederreiter H. Knapsack-type cryptosystems and algebraic coding theory. Prob Control Inf Theory. 1986;15(2):159–66.
[36] Cramer R, Shoup V. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM J Comput. 2003;33(1):167–226.
[37] Persichetti E. Secure and anonymous hybrid encryption from coding theory. In: International Workshop on Post-Quantum Cryptography. Springer; 2013. p. 174–87.
[38] Cayrel PL, Gueye CT, Mboup EHM, Ndiaye O, Persichetti E. Efficient implementation of hybrid encryption from coding theory. In: Codes, Cryptology and Information Security: Second International Conference, C2SI 2017, Rabat, Morocco, April 10–12, 2017, Proceedings-In Honor of Claude Carlet 2. Springer; 2017. p. 254–64.
[39] Mathew KP, Vasant S, Rangan CP. Efficient Code-based hybrid and deterministic encryptions in the standard model. In: International Conference on Information Security and Cryptology. Springer; 2013. p. 517–35.
[40] Courtois NT, Finiasz M, Sendrier N. How to achieve a McEliece-based digital signature scheme. In: Advances in Cryptology-ASIACRYPT 2001: 7th International Conference on the Theory and Application of Cryptology and Information Security Gold Coast, Australia, December 9–13, 2001 Proceedings 7. Springer; 2001. p. 157–74.
[41] Debris-Alazard T, Sendrier N, Tillich JP. Wave: A new code-based signature scheme. Cryptology ePrint Archive: Report 2018/996; 2018. https://eprint.iacr.org/2018/996/20181022:154324.
[42] Banegas G, Debris-Alazard T, Nedeljković M, Smith B. Wavelet: Code-based postquantum signatures with fast verification on microcontrollers. 2021. arXiv: http://arXiv.org/abs/arXiv:211013488.
[43] Fiat A, Shamir A. How to prove yourself: Practical solutions to identification and signature problems. In: Advances in Cryptology-CRYPTO-86: Proceedings 6. Springer; 1987. p. 186–94.
[44] Stern J. A new identification scheme based on syndrome decoding. In: Advances in Cryptology–CRYPTO'93: 13th Annual International Cryptology Conference Santa Barbara, California, USA August 22–26, 1993 Proceedings. Springer; 1993. p. 13–21.
[45] Jain A, Krenn S, Pietrzak K, Tentes A. Commitments and efficient zero-knowledge proofs from learning parity with noise. In: Advances in Cryptology-ASIACRYPT 2012: 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2–6, 2012. Proceedings 18. Springer; 2012. p. 663–80.
[46] Cayrel PL, Véron P, Alaoui SMEY. A Zero-Knowledge Identification Scheme Based on the q-ary Syndrome Decoding Problem. In: Selected areas in cryptography. vol. 6544. Berlin, Heidelberg: Springer; 2010. p. 171–86.
[47] Lyubashevsky V. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In: Advances in Cryptology-ASIACRYPT 2009: 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6–10, 2009. Proceedings 15. Springer; 2009. p. 598–616.
[48] Biasse JF, Micheli G, Persichetti E, Santini P. LESS is more: code-based signatures without syndromes. In: Progress in Cryptology-AFRICACRYPT 2020: 12th International Conference on Cryptology in Africa, Cairo, Egypt, July 20–22, 2020, Proceedings 12. Springer; 2020. p. 45–65.
[49] Persichetti E. Efficient one-time signatures from quasi-cyclic codes: A full treatment. Cryptography. 2018;2(4):30.
[50] Persichetti E. Improving the efficiency of code-based cryptography. PhD thesis, University of Auckland; 2012.
[51] Fukushima K, Roy PS, Xu R, Kiyomoto S, Morozov K, Takagi T. Random code-based signature scheme (racoss). First round submission to the NIST post-quantum cryptography call. 2017.
[52] Li Z, Xing C, Yeo SL. A new code based signature scheme without trapdoors. Cryptology ePrint Archive. 2020.
[53] Song Y, Huang X, Mu Y, Wu W, Wang H. A code-based signature scheme from the Lyubashevsky framework. Theoret Comput Sci. 2020;835:15–30.
[54] Aragon N, Blazy O, Gaborit P, Hauteville A, Zémor G. 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. p. 728–58.

[55] Debris-Alazard T, Sendrier N, Tillich JP. Wave: A new family of trapdoor one-way preimage sampleable functions based on codes. In: Advances in Cryptology-ASIACRYPT 2019: 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8–12, 2019, Proceedings, Part I. Springer; 2019. p. 21–51.
[56] Baek J, Steinfeld R, Zheng Y. Formal proofs for the security of signcryption. In: Public Key Cryptography: 5th International Workshop on Practice and Theory in Public Key Cryptosystems, PKC 2002 Paris, France, February 12–14, 2002 Proceedings 5. Springer; 2002. p. 80–98.
[57] Yung M. Practical signcryption. Germany: Springer Science & Business Media; 2010.
[58] Fujisaki E, Okamoto T. Secure integration of asymmetric and symmetric encryption schemes. In: Advances in Cryptology-CRYPTO-99: 19th Annual International Cryptology Conference Santa Barbara, California, USA, August 15-19, 1999 Proceedings. Springer; 1999. p. 537–54.
[59] Barelli E, Couvreur A. An efficient structural attack on NIST submission DAGS. In: Advances in Cryptology-ASIACRYPT 2018: 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2–6, 2018, Proceedings, Part I 24. Springer; 2018. p. 93–118.
[60] Prange E. The use of information sets in decoding cyclic codes. IEEE Trans Inform Theory. 1962;8(5):5–9.
[61] Bernstein DJ, Chou T, Lange T, von Maurich I, Misoczki R, Niederhagen R, et al. Classic McEliece: conservative code-based cryptography. NIST submissions. 2017.
[62] Pellikaan R, Márquez-Corbella I. Error-correcting pairs for a public-key cryptosystem. In: Journal of Physics: Conference Series. vol. 855. IOP Publishing; 2017. p. 012032.
[63] Mora R, Tillich JP. On the dimension and structure of the square of the dual of a Goppa code. Des Codes Cryptogr. 2023;91(4):1351–72.
[64] Lampe R, Patarin J. Analysis of some natural variants of the PKP algorithm. Cryptology ePrint Archive. 2011.
[65] Georgiades J. Some remarks on the security of the identification scheme based on permuted kernels. J Cryptol. 1992;5:133–7.
[66] Paiva TB, Terada R. Cryptanalysis of the binary permuted kernel problem. In: Applied Cryptography and Network Security: 19th International Conference, ACNS 2021, Kamakura, Japan, June 21–24, 2021, Proceedings, Part II. Springer; 2021. p. 396–423.
[67] Chiba D, Matsuda T, Schuldt JC, Matsuura K. Efficient generic constructions of signcryption with insider security in the multi-user setting. In: Applied Cryptography and Network Security: 9th International Conference, ACNS 2011, Nerja, Spain, June 7–10, 2011. Proceedings. vol. 11. Springer; 2011. p. 220–37.
[68] Nakano R, Shikata J. Constructions of signcryption in the multi-user setting from identity-based encryption. In: Cryptography and Coding: 14th IMA International Conference, IMACC 2013, Oxford, UK, December 17–19, 2013. Proceedings. Springer; 2013. p. 324–43.

Appendix

A PKE.Game

Here, we recall the IND-CCA2 game for PKE called PKE.Game in our scheme. The decryption oracle is denoted by $\mathcal{O}$ (Figure A1).

In Step 4, the adversary $\mathcal{A}_{\mathrm{McE}}$ is restricted not to make request to $\mathcal{O}$ on the ciphertext $\boldsymbol{c}$. Clear texts $\boldsymbol{m}_0$ and $\boldsymbol{m}_1$ must have the same length. $\mathcal{A}_{\mathrm{McE}}$ wins when $\tilde{b} = b$, and its advantage corresponds to the probability that it wins this game, which is denoted by $\varepsilon_{\mathrm{pke}}$.

B Target preimage-free

Target preimage-free function is a special case of universal one-way function. An adversary is given $(\mathcal{H}, \boldsymbol{y})$ (chosen at random in their domain) and then attempts to find $\boldsymbol{x}$ such that $\mathcal{H}(\boldsymbol{x}) = \boldsymbol{y}$. Let $\chi_\lambda = \{X\}$ be a collection of domains and $\chi = \{\chi_\lambda\}_{\lambda\in\mathbb{N}}$. Let $\tilde{\mathcal{H}}_\lambda = \{\mathcal{H} : X \longrightarrow \{0,1\}^\lambda : X \in \chi_\lambda\}$ and $\tilde{\mathcal{H}} = \{\tilde{\mathcal{H}}_\lambda\}_{\lambda\in\mathbb{N}}$. Note that X is identified by the description of $\mathcal{H}$. Let $\mathcal{A}_{\mathrm{PreIm}}$ be an adversary playing the following game (Figure A2).

$\mathcal{A}_{\mathrm{PreIm}}$ wins the game when $\mathcal{H}(\boldsymbol{x}) = \boldsymbol{y}$ and the advantage of $\mathcal{A}_{\mathrm{PreIm}}$ is the probability that it wins Preimage.Game for a given $\mathcal{H} \longrightarrow \tilde{\mathcal{H}}_\lambda$ and $\boldsymbol{y} \in \{0,1\}^\lambda$. We say that $\tilde{\mathcal{H}}$ is Target Preimage free with regard to χ when the advantage $\varepsilon_{\mathrm{PreIm}}$ of $\mathcal{A}_{\mathrm{PreIm}}$ is negligible.

C Security of the McEliece encryption with Fujisaki–Okamoto transformation

For the IND-CCA security of McEliece's scheme described in Figure 1, we need the following definition:

Definition 2. (γ-uniformity [21]) A public key encryption scheme Π is called γ-uniform and $\mathcal{R}$ be the set where the randomness to be used in the (probabilistic) encryption is chosen. For a given key-pair $(\mathsf{pk}, \mathsf{sk})$, $\boldsymbol{x}$ be a plaintext and a string $\boldsymbol{y}$, we define

$$\gamma(\boldsymbol{y}) = \Pr[\boldsymbol{r} \xleftarrow{\$} \mathcal{R} : \boldsymbol{y} = \mathcal{E}_{\mathsf{pk}}(\boldsymbol{x}, \boldsymbol{r})],$$

where the notation $\mathcal{E}_{\mathsf{pk}}(\boldsymbol{x}, \boldsymbol{r})$ makes the role of the randomness $\boldsymbol{r}$ explicit. We say that Π is γ-uniform if, for any key-pair $(\mathsf{pk}, \mathsf{sk})$, any plaintext $\boldsymbol{x}$ and any ciphertext $\boldsymbol{y}$, $\gamma(\boldsymbol{x}, \boldsymbol{y}) \leq \gamma$ for a certain $\gamma \in \mathbb{R}$.

We now can state the following lemma.

Lemma 1. *The McEliece scheme with the Fujisaki–Okamoto transformation described in* Figure 1 *is γ uniform with*

PKE. Game
1. Step 1: $(\mathsf{pk}, \mathsf{sk}) \longleftarrow \mathsf{KeyGen}(1^\lambda)$
2. Step 2: $(\boldsymbol{m}_0, \boldsymbol{m}_1, \rho) \longleftarrow \mathcal{A}^{\mathcal{O}}_{\mathrm{McE}}(\mathsf{pk})$
3. Step 3: $b \xleftarrow{\$} \{0,1\}$ and $\boldsymbol{c} \longleftarrow \mathsf{PKE.Encrypt}_{\mathsf{pk}}(\boldsymbol{m}_b)$ where $\mathsf{PKE.Encrypt}_{\mathsf{pk}}$ (resp. $\mathsf{PKE.Decrypt}_{\mathsf{sk}}$) is the encryption (resp. decryption) algorithm in the PKE scheme.
4. Step 4: $\tilde{b} \longleftarrow \mathcal{A}^{\mathcal{O}}_{\mathrm{McE}}$

Figure A1: PKE.Game.

Preimage. Game
1. Step 1: $\mathcal{H} \longrightarrow \tilde{\mathcal{H}}_\lambda$
2. Step 2: $\boldsymbol{y} \longrightarrow \{0,1\}^\lambda$
3. Step 3: $\boldsymbol{x} \longrightarrow \mathcal{A}_{\mathrm{PreIm}}(\mathcal{H}, \boldsymbol{y})$ such that $\boldsymbol{x} \in X$.

Figure A2: Preimage game.

$$\gamma = \frac{1}{2^{\tilde{k}}\binom{n}{t}}$$

Proof. For any vector $\boldsymbol{y} \in \mathbb{F}_2^{n_r}$, either $\boldsymbol{y}$ is a word at distance t from the code C of generator matrix $\mathbf{G}_{\mathsf{pk},r}$, or it isn't. When $\boldsymbol{y}$ is not a distance t of C, the probability for it to be a valid ciphertext is equal to 0. Else there is only one choice for $\boldsymbol{r}$ and $\boldsymbol{e}$ such that $\boldsymbol{y} = \boldsymbol{r}\mathbf{G}_{\mathsf{pk},r} \oplus \boldsymbol{e}$, i.e.,

$$\mathsf{Pr}(d(\boldsymbol{y}, C)) = t = \frac{1}{2^{\tilde{k}}\binom{n_r}{t}}$$

Theorem 4. *Under Assumptions 1, 3, and 5, the McEliece scheme based on a subcode of Goppa code with the Fujisaki–Okamoto transformation described in* Figure 1 *is* IND - CCA2 *secure.*

Proof. In Figure 1, the symmetric encryption used is the XOR function which is a one-time pad. Under Assumptions 1 and 3, the old McEliece encryption scheme is one-way secure. Therefore, according to Theorem 12 of [58], the McEliece scheme with the Fujisaki–Okamoto transformation is IND - CCA2 secure. □