On Provably Secure Code-based Signature and Signcryption Scheme

No Author Given

No Institute Given

Abstract. Signcryption is a cryptographic protocol that provides authentication and confidentiality as a single primitive at a cost lower than the combined cost of sign and encryption. Code-based cryptography, a likely candidate for post-quantum cryptography, provides an exciting alternative to number-theoretic cryptography. Courtois, Finiasz and Sendrier proposed the only practical code-based signature(CFS signature) at Asiacrypt 2001. But that signature scheme currently lacks a formal proof of security due to the existence of the high rate distinguisher proposed by Faugère et al. In this paper, we make use of an alternate key-construct for the CFS signature, and thus prove its existential unforgeability under chosen message attacks (EUF-CMA). Also, we propose a code-based signcryption scheme and prove its security. To the best of our knowledge, this is the first code-based, provably secure signature and signcryption scheme in literature.

Keywords. Signature, Signcryption, Code-based cryptography, CFS signature, Syndrome decoding.

1 Introduction

Authentication and confidentiality of a message are among important security goals achieved using cryptography. Confidentiality is achieved by encryption and signature achieves authentication. Signcryption as a primitive, aims at attaining the above goals at a lower cost than individually signing and encrypting or vice-versa. Zheng [32], in 1997 proposed the first digital signcryption scheme. Later, a formal model of security for signcryption schemes was provided by Baek et al. in [2], which includes signcrypted text indistinguishabile under chosen ciphertext attack (SC-IND-CCA2) for confidentiality and signature on the signcrypted text is existential unforgeable under chosen message attack (SC-EUF-CMA) for unforgeability. Also, a stronger notion of security called *insider security* was introduced by An et al. in [1], which proposed that a signcryption scheme needs to offer confidentiality even if all the private-keys except the receiver's private key are known (the private key of the sender in particular, is known to the adversary), and it must be unforgeable even if all the private keys except the private-key of the sender are known (in particular, the private key of the receiver is known to the adversary).

The notion of code-based cryptography was initiated by the encryption scheme proposed by McEliece [28] in 1978, which was based on the Bounded Decoding Problem. The aforementioned problem is NP-complete [5]. Niederreiter [18] proposed an encryption scheme which was effectively on the dual of the code used in the McEliece encryption scheme. In the Niederreiter system, the security was based on the hardness of the Syndrome Decoding Problem. The security of the schemes by McEliece [28] and Niederreiter [18] are shown to be equivalent in [22]. Stern [30] proposed an efficient code-based identification scheme, which did not require a trapdoor like Goppa codes and its corresponding decoding mechanism, unlike the above cryptosystems. But, to obtain a signature scheme based on [30] is practically infeasible as the signature size is very large. Courtois et al.[9] in 2001 proposed a signature scheme (CFS signature scheme) based on the hardness of Syndrome Decoding Problem. These signatures are practical only for high-rate linear codes (as the density of non-decodable syndromes is sparse). Although it has a relatively large signing time, the signature scheme was an exciting breakthrough, as it laid a foundation-stone for development of many code-based schemes in various cryptographic primitives. Barreto et al. [4] proposed one-time signature and Kabatianskii et al. [19] also proposed a signature which is secure only for few signatures. Otmani e al. [25] proved an attack on the above two schemes.

Motivation. The first practical code based signature scheme was reported in 2001 by Courtois et al. [10]. Later the authors added few more details in the scheme presented in [9]. The running time of the signing algorithm, is estimated to be $O(t^2 t! \log(n)^3)$, where t is the number of errors that can be corrected and n is the length of the code word. The formal proof for the signature was not presented in the paper, but to arrive at safe parameter values, the authors considered the attacks like Canteaut Chabaud (CC) attack [7] and Lee Brickell (LB) attack [20]. Assuming a threshold of 2^{80} binary work factor for security against specific attack, $(\log(n), t)$ values that would withstand the attacks and the corresponding signing times were estimated. Specifically in order to resist the attacks like the CC attack and LB attack, [9] suggests that the values such as (15,10), (16,9) are appropriate for $(\log(n), t)$. They have also shown that the running time of the signature algorithm is reasonable for these choice of parameters. In 2009, Finiasz et al. [16] considered an attack due to Daniel Bleichenbacher (communicated through the private communication to the authors of [16], but never published). The authors of [16] suggested the parameters must be set to (15,12) or (16,12) etc. for $(\log(n), t)$ values in view of this new attack. Subsequently, in 2011, Sendrier[29] studied another attack, namely decoding one out of many. The revised parameters that can be chosen according to [29] are (18,13), (19,12), (20,11) for $(\log(n), t)$. In 2007, a formal proof was given by Dallot [11] assuming the hardness of syndrome decoding and code indistinguishability. In 2010, Faugere et al. [14] cryptanalysed the McEliece variants with compact keys algebraically, which was subsequently extended in [12]. The analysis of [12] is an extended analysis of a high rate distinguisher for McEliece encryption in [13]. This result gives that the t values (the error-correcting capacities) must be set higher than previous estimates to achieve a provably secure system. In particular if $\log(n)$ (n is the length of the code) is chosen as 18, 19 or 20 then t values must be set to 85, 114, 157 respectively. But, these parameters lead to impractical signing times ($> 2^{220}$ operations).

Still the distinguisher does not imply a concrete attack on the scheme in [11] or in [9]. In fact, the randomised CFS signature has widely been conjectured to be unforgeable, even though a formal security argument does not exist currently. Hence, we need to explore alternatives to overcome the distinguisher to arrive at a proof of security.

Signcryption is an important primitive in applications such as e-commerce, secure and authentic e-mails, etc., because it offers both confidentiality and authentication simultaneously. However, designing a signcryption scheme using the paradigm 'sign-then-encrypt' will lead to an inefficient scheme in code based scenario.

Our Contribution. In this paper, we alter the key-construct of the CFS signature [9] (based on the key construct in [17]) and formally argue the security of the proposed signature in the EUF-CMA model. To do so, we introduce a new distinguishability assumption, which is weaker than the current Goppa-code distinguishability assumption. We also propose the first code-based signcryption scheme (also, the first signcryption not using the classical number-theoretic assumptions, to the best of our knowledge) using the Niederreiter's system [18] and the CFS signature, both using the modified key-construct. The signcryption scheme is loosely based on the construction in [21]. We formally prove the confidentiality (in the SC-IND-CCA2) and unforgeability (in the SC-EUF-CMA).

Organisation of paper. In the next section we provide some preliminaries. In section 3, we briefly introduce our weaker distinguishability assumptions. We also argue (informally) as to why it is weaker than the conventional distinguishability assumptions. In section 4, we give the proposed signature scheme along with the security proof. In section 5, we give the proposal for signcryption and provide a sketch of its security and identify a few secure parameters for the scheme. We conclude in section 5. The formal proof of security of the signcryption scheme is given in appendix A.

2 Preliminaries

Before proceeding to the preliminaries, one should note that $\mathsf{negl}(n)$ is a negligible value with respect to the parameter n. We now enlist some basics of coding theory and the definitions of the primitives involved.

2.1 Coding Theory

A binary linear-error correcting code of length n and dimension k or a $[n, k]$- code is a *k-dimensional* subspace of $\mathbb{F}_2^n$. If the minimum distance between code-words is d, then we denote the code as an $[n, k, d]$ code, where d is called the Hamming distance. The error correcting capability of the code is $t = \lfloor \frac{d-1}{2} \rfloor$. The generator matrix $G \in \mathbb{F}_2^{k \times n}$ of a $[n, k]$ linear code $\mathcal{C}$ is a matrix of rank k whose rows span the code $\mathcal{C}$. The parity-check matrix $H \in \mathbb{F}_2^{n-k \times n}$ of a $[n, k]$ code $\mathcal{C}$ is defined as a matrix satisfying $HG^T = 0$. Hence, we can define the code $\mathcal{C}$ as $\{mG : \forall m \in \mathbb{F}_2^k\}$ or $\{c : Hc^T = 0\}$. We now proceed to list the hard problems.

Definition 1 Syndrome Decoding Problem. *For some parameters $[n, k, 2t + 1]$ given an $\mathbf{a} \in \mathbb{F}_2^{n-k}$ and a random matrix $H \in \mathbb{F}_2^{n-k \times n}$, find a vector $\mathbf{e} \in \mathbb{F}_2^n$ with weight $\mathsf{wt}(\mathbf{e}) \leq t$ such that $H\mathbf{e}^T = \mathbf{a}$.*

The advantage of a probabilistic polynomial time (PPT) algorithm $\mathcal{D}$ of solving the syndrome decoding problem for $[n, k, 2t + 1]$ code is denoted by $\mathsf{Adv}_{\mathcal{D}}^{\mathsf{SD}}(n, k)$. Syndrome Decoding Problem is hard (worst-case) for any random code [5]. Hence, $\mathsf{Adv}_{\mathcal{D}}^{\mathsf{SD}}(n, k) = \mathsf{negl}(n)$. But, for Goppa codes, there is a polynomial time algorithm for syndrome decoding.

Definition 2 Punctured Codes *[31] Let C be a code of length n and $S \subset N$ where N denotes the set $\{1, ..., n\}$. Let C_S denote the code which is obtained by deleting all coordinates of C in $N/\ S$. C_S is called punctured code of C in $N/\ S$.*

Definition 3 Equivalent Codes*[26] [31] Let C and D be two codes over the same field and of same length n. C and D are called equivalent, if there is a permutation $\pi : \{1, ..., n\} \to \{1, ..., n\}$ such that*
$(c_1, ..., c_n) \in C \Leftrightarrow \left(c_{\pi(1)}, ..., c_{\pi(n)}\right) \in D$.

The problem of finding code equivalence is proven to be harder than graph isomorphism problem[27].

Consider the problem with C be a code of length n and D a code of length m, such that $m \leq n$, Does there exist a subset $S \subset \{1, ..., n\}$, such that C_S and D are equivalent?

Definition 4 *Let M be a $k \times n$ matrix over a field F with columns $m_1, ..., m_n$ and $\tau : \{1, ..., q\} \to \{1, .., n\}$ an injection, such that $(q \leq n)$. The $k \times q$ matrix consists of the columns $m_{\tau(1)}, ..., m_{\tau(q)}$ (in this order) is denoted by M_τ.*

Definition 5 Equivalent Punctured Codes (EPC) *[31] Let M be a $k \times n$ matrix and H be a $k \times m$ matrix where $m \leq n$ over a field F, Does there exist a non-singular matrix $k \times k$ matrix T and an injective map τ such that $(TM)_\tau = TM_\tau = H$.*

The EPC problem was shown to be NP-completeby reducing three dimensional matching (3DM) problem reduced to Equivalent Punctured code problem [31], [24]. In fact, the hardness of EPC problem is the basus of the encryption scheme given in [31]. Thus we are justified in the following assumption.

Assumption 1 *There is no PPT algorithm $\mathcal{D}$ that can find a a non-singular matrix $k \times k$ matrix T and an injective map τ such that $(TM)_\tau = TM_\tau = H$, given M and H.*

In the construction of the signature presented in this paper, the security of signature is reduced to Equivalent Punctured Code problem.

2.2 Signature

Definition 6 *A signature scheme consists of a triple of algorithms (KeyGen,Sign,Verify) where,*

KeyGen is a PPT algorithm that takes as input the security parameter κ (or 1^κ) to return the key-pair (sk, pk), where sk is the signing-key which is kept as secret with the signer, and pk is the verification-key which is made public.

Sign is a PPT algorithm that takes as input the secret signing key sk and the document/message m from the message space and outputs a signature σ.

Verify is a polynomial time algorithm that takes as input the public verification key pk, the document/message m and the signature σ on the message and outputs $ACCEPT$ if σ is valid and $REJECT$ otherwise.

Definition 7 (Unforgeability.) *A signature scheme is said to be existentially unforgeable against chosen-message insider attack (EUF-CMA) if no PPT forger $\mathcal{F}$ has a non-negligible advantage in the following game:*

1. *The challenger runs KeyGen to generate a key pair (sk, pk). sk is kept secret while pk is given to the adversary $\mathcal{F}$.*
2. *The forger $\mathcal{F}$ adaptively makes a polynomial number of queries to the signature oracle and the hash oracles (if any).*
3. *The forger $\mathcal{F}$ produces a signature σ and wins the game if :*
   - *Verify(pk,m,σ) outputs $ACCEPT$ and*
   - *(m, σ) is not the output of any signature oracle, described in step 2.*

The probability that, for a parameter n a forger is able to forge a signature is denoted by $\mathsf{Succ}_{\mathcal{F}}^{euf-cma}(n)$

2.3 Signcryption

We first begin by formally defining a signcryption scheme. This formal definition is based on the definition given in [21].

Definition 8 *A signcryption scheme is a triple of algorithms ($\mathsf{UK_g}$, $\overline{\mathsf{S}}$, $\overline{\mathsf{U}}$) for a security parameter 1^k.*

*$(sk, pk) \leftarrow \mathsf{UK_g}(1^k)$ is the **Key-generation** algorithm which takes a security parameter k to generate the private/public key pair (sk,pk).*

$\sigma \leftarrow \overline{\mathsf{S}}(1^k,\ m,\ sk_S,\ pk_R)$ *is the* ***Signcryption*** *is a PPT algorithm which takes a security parameter k, the message m from a message space M, the sender's private key* sk_S *and receiver's public key* pk_R*, to output the signcrypted text* σ*.*

$((m,s),\mathsf{Accept})/\mathsf{Reject} \leftarrow \overline{\mathsf{U}}(1^k,\sigma,sk_R,pk_S)$ *is the* ***De-signcrypt*** *algorithm. The* ***De-Signcrypt*** *algorithm is a two-staged process.*

Stage 1 *It takes k,the signcrypted text* σ *and the receiver's private key* sk_R *as input to decrypt and get the signed message m, a verifiable signature s (extracted from* σ*) on m and the verification key* pk_S*, or* Reject *which indicates failure of de-signcrypt.*

Stage 2 *It takes the signature s and message m, both obtained from Stage 1, and verifies using the verification key* pk_S *to output* Accept *or* Reject*.*

The security notions of confidentiality and unforgeability (that also model the insider security notion) are described here. The notion is based on the notion mentioned in [21].

Definition 9 (Confidentiality.) *A signcryption scheme is semantically secure against chosen ciphertext attack (SC-IND-CCA2) if no PPT adversary* $\mathcal{A}$ *has a non-negligible advantage in the following game:*

1. *The challenger runs* $\mathsf{UK_g}$ *to generate a key pair* (sk_U,pk_U)*.* sk_U *is kept secret while* pk_U *is given to the adversary* $\mathcal{A}$*. For the others users* U'*(say), the challenger runs* $\mathsf{UK_g}$ *to generate* $(sk_{U'},pk_{U'})$*, and sends the tuple to the adversary. (Insider security).*
2. *In the first stage,* $\mathcal{A}$ *makes a polynomial number of queries to the following oracles:*

   Signcryption Oracle : $\mathcal{A}$ *prepares a message* $m \in M$ *and a public key* pk_R*, and queries the Signcryption Oracle (simulated by the challenger) for the result of* $\overline{\mathsf{S}}(m,sk_U,pk_R)$*. The result is returned if* $pk_R \neq pk_U$ *and* pk_R *is valid in the sense that* pk_R *is in the range of* $\mathsf{UK_g}$ *with respect to the security parameter. Otherwise, a symbol '*$\perp$*' is returned for rejection.*

   De-signcryption Oracle : $\mathcal{A}$ *produces a signcrypted text* σ *and queries for the result of* $\overline{\mathsf{U}}(\sigma,sk_U,pk_S)$*. The message is returned if the de-signcryption is successful and the signature is valid under the sender's public key. Otherwise, a symbol '*$\perp$*' is returned for rejection.*

   The queries may be asked adaptively,i.e., each query may depend on the answers to the previous queries.
3. $\mathcal{A}$ *produces two messages* $m_0,m_1 \in M$ *of equal length and a valid private key* sk_S*. The challenger takes a random* $b \overset{R}{\leftarrow} \{0,1\}$ *and computes a signcryption* $\sigma^* = \overline{\mathsf{S}}(m_b,sk_S,pk_U)$ *of* m_b *with the sender's private key* sk_S *under the receiver's public* pk_U*.* σ^* *is sent to* $\mathcal{A}$ *as a challenge.*
4. $\mathcal{A}$ *makes a number of queries as in the first stage with the restriction that it cannot query the de-signcryption oracle with* σ^**.*
5. *At the end of the game,* $\mathcal{A}$ *outputs a bit* b' *and wins if* $b' = b$*.*

$\mathcal{A}$*'s advantage is defined as* $Adv^{ind-cca}(\mathcal{A}) = Pr[b = b'] - \frac{1}{2}$ *and the probability that* $b = b'$ *is the probability that* $\mathcal{A}$ *wins the game.*

Definition 10 (Unforgeability.) *A signcryption scheme is said to be existentially unforgeable against chosen-message insider attack (SC-EUF-CMA) if no PPT forger* $\mathcal{F}$ *has a non-negligible advantage in the following game:*

1. *Follow the first two steps as that of confidentiality game.*
2. *The forger* $\mathcal{F}$ *produces a signcrypted text* σ *and a valid key pair* (sk_R,pk_R) *and wins the game if :*
   - $\overline{\mathsf{U}}(\sigma,sk_R,pk_U)$ *returns a tuple (m,s) such that the output of verification on* (m,s) *for the verification key* pk_U *is* Accept*.*
   - σ *is not the output of the signcryption oracle.*

3 Weak Distinguishability Assumptions

In this section, we introduce a weak Goppa distinguishability assumption. We assume that the public key construct using Goppa code is computationally indistinguishable from a random matrix. The public key and private key pair must be constructed in such a way that given public key, it should not be possible to reconstruct the private key[8]. Therefore signature scheme is a variant of the CFS, but uses an alternate public key construction such that the public key is no longer the permutation equivalent of the private code. The key construct is some what similar to the one used in [17], but there are subtle differences. The private keys used for signing are Q, H, P, where Q is an $n-k \times n-k$ invertible matrix, H a Parity Check Matrix of a binary code $\mathcal{C}$ of type $\mathit{Goppa}(n,k,t)$, which is always a nonzero matrix, and a random $n \times n$ permutation matrix P as per the CFS signature. But the public key $\widetilde{H}$ used in our scheme is $(n-k+1) \times n$ matrix, unlike $(n-k) \times n$ parity check matrix used as the public key in the CFS signature. Therefore the public key is no longer a permuted randomised parity check matrix of a Goppa code. Also the matrix Q is computed as a product of two randomly generated matrices H' of size $n-k \times n'$, where $n' = n-k+1$ and Q' of size $n' \times n-k$. The process is repeated until Q is invertible. We see that Q is invertible with probability at least 0.288 [6]. Hence, in roughly 4 trials we can expect to obtain an invertible Q. Also, $\mathbf{a}$ is selected such that $H'\mathbf{a}^T = 0$. Due to all these differences, we obtain a novel and provably secure signature scheme.

3.1 Key-Construct

The key generation involves the following steps

- Select H, a Parity Check Matrix of a binary code $\mathcal{C}$ of type $\mathit{Goppa}(n,k,t)$. and a random $n \times n$ permutation matrix P .
- Select randomly a $H' \in_R \mathbb{F}_2^{n-k\times n'}$ and a $Q' \in_R \mathbb{F}_2^{n'\times n-k}$, such that Q' is full-rank and compute matrix $Q = H'Q'$. Repeat the step until Q is invertible.
- Select an $\mathbf{a} \in \mathbb{F}_2^{n'}$, such that $H'\mathbf{a}^T = 0$ and Select $\mathbf{b} \in_R \mathbb{F}_2^{n}$..
- Compute a parity check matrix $\widetilde{H}$ as $\widetilde{H} = Q'HP \oplus \mathbf{a}^T\mathbf{b}$. $\widetilde{H}$ is a $n' \times n$ matrix. If $\widetilde{H}$ is not full-rank, repeat the process with a different random $\mathbf{b}$, until we obtain a full-rank $\widetilde{H}$.

3.2 Assumption

For the aforementioned key construct we make the following hardness assumptions which is harder than Goppa code Distinguishability assumption. The symbols in this section are as defined in the key construct (unless specified otherwise).

Assumption 2 *There is no PPT distinguisher $\mathcal{D}$ that can distinguish $\widetilde{H}$ from a parity check matrix R of a random $(n, k-1, t)$ linear code.*

Let the advantage of such a distinguisher be $\mathsf{Adv}_{\mathcal{D}}^{\mathsf{Dist}}(n,k) = |Pr[\mathcal{D}(\widetilde{H}) \to 1] - Pr[\mathcal{D}(R) \to 1]|$. We consider assumption 2 to be weaker than the Goppa-code distinguishability assumption. The reason for this can be explained based on another assumption.

Assumption 3 *Given $\widetilde{H}$, it is infeasible to retrieve H, a and b.*

Given $\widetilde{H}$ it is infeasible to find H'. The corresponding decisional version of assumption 3 is *”Given $\widetilde{H}$, does there exist a H' such that $H'\widetilde{H} = QHP$ for some Goppa code with parity matrix H, some $n-k \times n-k$ invertible matrix Q and $n \times n$ permutation matrix P”*. This is clearly Assumption 2

(if there doesn't exists such a H' then the input matrix is random). This is a generalisation of the Punctured Code problem. To elaborate, suppose this decision problem is solved, then we can solve the following problem,
For two matrices M and H does there exist a T and selections S_1, $S_2 \subseteq \{1, 2, \ldots n\}$ such that $(TM)_{S_1} = H_{S_2}$.
Another reason, why we consider this problem to be weaker than the Goppa distinguishability is based on the equations for the distinguisher. It is seen that the public key is not a parity check matrix of a permutation equivalent code of the secret code. We take the generator matrix corresponding to the public matrix. Hence, to solve the system to obtain the private keys, the following system of equations have to be solved,

$$\{g_{i1}(X_1^j Y_1 + a_j b_1) + \ldots + g_{in}(X_n^j Y_n + a_j b_n) = 0 \mid i \in \{1, \ldots, k\} \ \& \ j \in \{0, \ldots, n-k\}\}$$

where g_{ij} is the element of the generator matrix at the i^{th} row and j^{th} column. Unlike the system in [13], the system here is not trivially linearisable. Hence, the distinguishing based on the dimension may not hold good.

4 Proposed Signature Scheme

4.1 Scheme

System Parameters(κ). The following system parameters are used:

- Parameters of the code n, k, t for any $[n, k, 2t+1]$ linear code, with n, k determined by the security parameter κ, and $t = \frac{n-k}{\log_2 n}$.
- We define $n' = n - k + 1$.
- Hash function $\mathcal{G} : \mathbb{F}_2^n \times \{0,1\}^n \rightarrow \mathbb{F}_2^{n'}$

Key Generation(κ, *parameters*). The key generation is as mentioned in section 3 and we have ***private key:*** H, P, Q, H'; ***public key:*** $\widetilde{H}$; ***parameters:*** $[n, k, t], n', \mathcal{G}$
Sign(m, H, P, Q, H'). To sign a message $m \in \{0,1\}^l$

```
– repeat
      $r \xleftarrow{R} \mathbb{F}_2^n$
      $m'_1 \leftarrow \mathcal{G}(r, m)$
      $m_1 = H'{m'_1}^T$
      $s_1 = P^T Decode_H(Q^{-1} m_1)$ //If $m_1$ is not decodable for $H$, the decoding algorithm sets $s_1 = \perp$.
  until($s_1 \neq \perp$ && $m'_1 = \widetilde{H} s_1^T$)
– The signature is $\sigma = (s_1, r)$
```

Verify($m, \sigma, \widetilde{H}$). Verification of the signature $\sigma = (s_1, r)$ on m is done by checking $\widetilde{H}{s_1}^T \overset{?}{=} \mathcal{G}(r, m)$ and $wt(s_1) \leq t$. If TRUE then return ACCEPT, else return REJECT.
Note that m_1 is made a syndrome for H' for the word m'_1, and m_1 also made a syndrome for code word s_1 for QHP. When m'_1 is replaced with $\widetilde{H} s_1^T$ according to the signing procedure in $m_1 = H'{m'_1}^T$, observe that it becomes a syndrome for code word s_1 for QHP
Note 1. To elaborate on the scheme, we take two cases.

- In the first case, assume m'_1 is a decodable syndrome for $\widetilde{H}$, i.e., there exists an s_1 such that $m'_1 = \widetilde{H} s_1^T$ and $wt(s_1) \leq t$. Then it is seen that $m_1 = H'{m'_1}^T = H'(Q'HP \oplus \mathbf{a}^T\mathbf{b})s_1^T = QHPs_1^T$ (since $Q = H'Q'$ and $H'\mathbf{a}^T = 0$). Hence, it is possible to decode m_1 using the decoding algorithm on H to obtain s_1 which is the solution of syndrome decoding of m'_1 for $\widetilde{H}$.

- In the second case, assume m'_1 is not decodable for $\widetilde{H}$. Hence, there does not exist any s_1 unlike the first case. But m_1 can either be decodable or not decodable for H. It is the property of any binary linear code of length n' and dimension k', to partition the space $\mathbb{F}_2^{n'}$ into $2^{n'-k'}$ partitions of size $2^{k'}$, using the syndromes. Hence, for H' (which has dimension 1), exactly two values m'_1 and m'_2(say) map to the same syndrome m_1. If m_1 is not decodable for H then both m'_1 and m'_2 are not decodable for $\widetilde{H}$. But, if m_1 is decodable (and can be decoded to s (say)) , then one of the two the values m'_1 and m'_2 is of the form $\widetilde{H}s^T$, whereas the other is not decodable. This can be proved by contradiction, as, assuming both m'_1 and m'_2 are decodable for $\widetilde{H}$, into s_1 and s_2 repsepctively. Then, $H'm'_1$ and $H'm'_2$ are also correspondingly decodable into s_1 and s_2 for H. But, we know that $H'm'_1 = H'm'_2 = m_1$. Hence, it is a contradiction that m_1 decodes into two vectors (both of weight $\leq t$). Therefore, only one vector is decodable, while the other is not decodable.

Hence, the expected time taken [9] for the above signature is $O(t!t^2m^3)$.

4.2 Security of the scheme

We now proceed to prove the unforgeability of the scheme under the EUF-CMA security notion in random oracle model.

Theorem 1 *The given scheme is EUF-CMA (under the random oracle model) if the syndrome decoding (SD) is hard to solve and the public key is computationally indistinguishable from the parity matrix of a random $(n, k-1, t)$ code R.*

Proof: We build the proof, by constructing a challenger $\mathcal{C}$ through a sequence of games **Game0, Game1,**$\cdots$. **Game0** is played by using the protocol as mentioned in EUF-CMA game. Successive games are obtained by small modifications of the preceding games, in such a way that the difference of the view in consecutive games is easily quantifiable. Let $q_{\mathcal{G}}, q_s$ be the maximum number of queries made by the forger $\mathcal{F}$ to the hash oracle of $\mathcal{G}$ and the signature oracle. We want to show that the advantage for the adversary $\mathcal{F}$ is equivalent to the advantage of solving the hard problem SD for a random code with parity check matrix R and some syndrome s.
To answer the hash queries and the signature queries, we maintain the lists, $\mathcal{G}^{list}, \sigma^{list}$ and Λ. If there is no value in a list we denote its output by $\perp$.

- The list $\mathcal{G}^{list}$ contains a tuple $((x, s), a)$ indexed by (r, m).
- The σ^{list} (the signature list) consists of entries of the form $(m, \sigma = (s, r))$.
- The list Λ consists of indices r of $\Lambda(m)$ for which the simulator is able to produce a signature on $\mathcal{G}(m, \Lambda(m))$,i.e., the list of r for which $\mathcal{G}(m, \Lambda(m))$ is a decodable syndrome.

Game 0. Here the challenger employs the actual scheme according to the EUF-CMA game. The private and public key pair are obtained by running the key generation algorithm given the scheme, to obtain secret key (Q, H, P, H'), where $H \leftarrow \mathsf{Goppa}(n, k)$ (a binary Goppa code), and the public key $\widetilde{H} = Q'HP \oplus \mathbf{a}^T\mathbf{b}$. $\widetilde{H}$ is given to $\mathcal{F}$. Also, $\mathcal{F}$ is given access to the hash oracle $\mathcal{G}$. The signature oracle functions as mentioned in the scheme. Let X_0 be the event that $\mathcal{F}$ wins Game 0. It is seen that Game 0 runs the EUF-CMA game on the proposed scheme perfectly. Hence,

$$Pr[X_0] = \mathsf{Succ}^{euf-cma}(\mathcal{F})$$

Game 1. (Simulation of hash oracle) In this game, the hash oracle for $\mathcal{G}$ is simulated, while the rest of the protocol is executed as in the previous game. The oracle is simulated as follows:

For the query on $\mathcal{G}$ of the form (m,r), we have two situations, depending on whether $r \in \Lambda(m)$. The simulation of the oracle is given below:

```
Input: A tuple $(m,r)$
Output: A syndrome $x$
if $r \neq \Lambda(m)$ then
   if $s = \perp$ then
     $s_1 \xleftarrow{R} \mathbb{F}_2^n$          // Since, the challenger may not be able to decode $\mathcal{G}(m,r)$,
     $x \leftarrow \widetilde{H}{s_1}^T$             // we take a $s$ randomly, and may not have weight  $< t$
     $s \leftarrow \perp$    $\mathcal{G}^{list}(m,r) \leftarrow ((x,s),s_1)$
  end
  return $\mathcal{G}(m,r) = x$
else
  if $x = \perp$ then
     $s_1 \xleftarrow{R} \mathbb{F}_2^n$ such that $wt(s_1) = t$
     $x \leftarrow \widetilde{H}s_1^T$                               //  $x$ is decodable, since $wt(s_1) \leq t$
     $s \leftarrow s_1$        $\mathcal{G}^{list}(m,r) \leftarrow ((x,s),s_1)$
   end
   return $\mathcal{G}(m,r) = x$
end
```

It is seen that, while the oracles are simulated in the Game 1, the distribution of these oracles remain unchanged from Game 0 (i.e., the randomness is maintained). Let the event that $\mathcal{F}$ wins Game 1 be denoted by X_1. Hence

$$Pr[X_1] = Pr[X_0]$$

Game 2. (Simulation of the signing oracle.) The signing oracle is simulated as follows:

```
Input: the message $m$ of length $l$ Output:  A signature $\sigma = (s_1, r)$
if $\Lambda(m) = \perp$ then
  $r \xleftarrow{R} \mathbb{F}_2^n$                         //Fix a $r$ such that $\mathcal{G}(m,r)$ is decodable, and
  $\Lambda(m) \leftarrow r$                        // $s$ such that $\widetilde{H}s_1^T = \mathcal{G}(m,r)$  and $wt(s) \leq t$
end
$((x,s_1),s_1) \leftarrow \mathcal{G}(m,\Lambda(m))$
if $(s_1 = \perp)$ then          //Incoherence, as $\mathcal{G}(m,r)$ was set earlier, when $r \neq \Lambda(m)$
  ABORT
else
  $r \leftarrow \Lambda(m)$
  $\Lambda(m) \leftarrow \perp$
Return $\sigma = (s_1, r)$.
```

The signature produced by the signing oracle, is valid according to the verification algorithm, since,$\widetilde{H}s_1^T = \mathcal{G}(m,r)$ and $wt(s_1) \leq t$.

In Game 2, incoherence occurs if the oracle to $\mathcal{G}$ is queried initially for some (m,r) such that later r is set to $\Lambda(m)$. This happens with the probability $\frac{q_s}{2^n}$ (since the indices Λ are defined only when the signature oracle is queried). It can be noted that this incoherence is the only scenario in which $\mathcal{F}$ can distinguish Game 2 from Game 1. Therefore, for the event X_2 that $\mathcal{F}$ wins Game 2,

$$|Pr[X_1] - Pr[X_2]| \leq \frac{q_s}{2^n}$$

Game 3. (Changing the key generation algorithm) The parity check matrix R, for which the syndrome decoding problem needs to be solved, is taken as the private key, i.e., $H = R$. The public key is $\widetilde{H} = R'$, where $R'^T = [R^T|z^T]$ where $z \in_R \mathbb{F}_2^n$. The verification key $\widetilde{H}$ is given to the forger $\mathcal{F}$, while $\mathcal{C}$ keeps the remaining secret keys. By assumption 2

$$|Pr[X_2] - Pr[X_3]| \leq \mathsf{Adv}_{\mathcal{C}}^{\mathsf{Dist}}(n,k)$$

Game 4. In this game, the challenger modifies the winning condition. The challenger first gets a random $c \xleftarrow{R} \{1,\ldots,q_s+q_{\mathcal{G}}+1\}$. $\mathcal{F}$ wins the Game if, in addition to the above conditions (as given in the previous game), the forgery was made on the c-th query to the hash oracle $\mathcal{G}$. This occurs with the probability $\frac{1}{q_s+q_{\mathcal{G}}+1}$. For the event that $\mathcal{F}$ wins Game 4, X_4, we obtain

$$Pr[X_4] = \frac{Pr[X_3]}{q_s+q_{\mathcal{G}}+1}$$

Game 5. In this game the challenger modifies the hash oracle, incorporating the problem instance (syndrome s) in the $c-th$ query. Since, the key used is R' and not R, we require a syndrome of length n'. Hence, a bit generated uniformly at random, s_c, is appended to the end of s. Therefore, the output of the hash oracle for the $c-th$ query is s' such that $s'^T = [s^T|s_c]$. Since, in game 5, the forger can output a forgery only if the final bit s_c has been guessed correctly (then s' is consistent with s),

$$Pr[X_5] = Pr[X_4]/2$$

Let s^* be the signature output by the forger. It is seen that s^* is the solution to the bounded decoding problem on the syndrome s' for $\widetilde{H}$. Also, s^* is guaranteed to be the solution for the syndrome Rs^T on R. Hence, we have

$$Pr[X_5] \leq \mathsf{Adv}_{\mathcal{C}}^{SD}(n,k)$$

Now, combining all results and use of triangular inequality, we have:
$Succ^{euf-cma}(\mathcal{F}) \leq \frac{q_s}{2^n} + \mathsf{Adv}_{\mathcal{C}}^{\mathsf{Dist}}(n,k) + 2(q_s+q_{\mathcal{G}}+1)\mathsf{Adv}_{\mathcal{C}}^{SD}(n,k)$.
The detailed reduction of the equations to arrive at the final bound is available in the full version of the paper.
Hence, the success of probability of the forger is bound by the advantage the challenger has in solving the syndrome decoding problem. This implies the signature is unforgeable as long as the corresponding syndrome decoding instance is hard to solve. □

Since the scheme avoids the distinguisher attack, the parameters that can be used in this scheme can be as that of the parameters suggested by [29] and the signing time will be slightly greater than the signing time of the [9]

5 Proposed Signcryption Scheme

The proposed scheme is the first code-based signcryption scheme (to the best of our knowledge). This scheme, takes into consideration the idea of construction used in [23, 21].

5.1 Scheme

System Parameters(κ). The following system parameters are used:

- Parameters of the code n, k, t for any $[n, k, 2t+1]$ linear code, with n, k determined by the security parameter κ, and $t = \frac{n-k}{\log_2 n}$.

- We define $n' = n - k + 1$.
- Collision resistant Hash functions $\mathcal{H} : \mathbb{F}_2^{n' \times n} \times \mathbb{F}_2^{n'} \times \mathbb{F}_2^{n} \rightarrow \{0,1\}^l$ (assuming messages of length l) and $\mathcal{G} : \mathbb{F}_2^{n} \times \{0,1\}^n \times \mathbb{F}_2^{n' \times n} \times \mathbb{F}_2^{n' \times n} \rightarrow \mathbb{F}_2^{n'}$

Key Generation(κ,*parameters*). For a user U the key generation involves the following steps

- Select H_U, a Parity Check Matrix of a binary code $\mathcal{C}$ of type $Goppa(n, k, t)$.
- Select randomly a $n \times n$ permutation matrix P_U.
- Select $\mathbf{b}_U \in_R \mathbb{F}_2^n$.
- Select randomly H'_U of size $n - k \times n'$ and $Q'_U \in_R \mathbb{F}_2^{n' \times n-k}$, such that Q'_U is full-rank and compute the matrix $Q_U = H'_U Q'_U$. Repeat until Q_U is invertible.
- Select $\mathbf{a}_U$, such that, $H'_U \mathbf{a}_U^T = 0$.
- Compute a parity check matrix $\widetilde{H}_U$ as $\widetilde{H}_U = Q'_U H_U P_U \oplus \mathbf{a}_U^T \mathbf{b}_U$. $\widetilde{H}_U$ is a $n' \times n$ matrix. If $\widetilde{H}_U$ is not a full-rank matrix, we repeat the process with different random $\mathbf{b}_U$ until we obtain a full-rank $\widetilde{H}_U$.

Thus we have
private key: H_U, P_U, Q_U, H'_U; ***public key:*** $\widetilde{H}_U$; ***parameters:*** $\mathcal{H}, \mathcal{G}, n, k, t, n'$

Signcrypt($m, H_S, P_S, Q_S, H'_S, \widetilde{H}_R$). To signcrypt a message $m \in \{0,1\}^n$ from a sender S and a receiver R

- ***repeat***

```
    $r \xleftarrow{R} \mathbb{F}_2^n$, such that $wt(r) \leq t$;
    $m'_1 \leftarrow \mathcal{G}(r, m, \widetilde{H}_R, \widetilde{H}_S)$
    $m_1 = H'_S {m'_1}^T$
    $s_1 = P_S^T Decode_{H_S}(Q_S^{-1} m_1)$
***until***$(s_1 \neq \perp$ && $m'_1 = \widetilde{H}_S s_1^T)$
```

- Compute $U = \widetilde{H}_R r^T$
- Compute $V = m \oplus \mathcal{H}(\widetilde{H}_R, U, r)$
- The signcrypted text is $\sigma = (U, V, s_1)$

De-signcrypt($\sigma, H_R, P_R, Q_R, H'_R, \widetilde{H}_S$). When the signcrypted text $\sigma = (U, V, s_1)$ is received R does the following:

```
*Compute* $U' = H'_R U^T$
$r' = P_R^T Decode_{H_R}(Q_R^{-1} U')$.
*if*$(r' = \perp \,||\, U \neq \widetilde{H}_R {r'}^T \,||\, wt(s_1) > t)$
    *Return* Reject
*else*
    Compute $m' = V \oplus \mathcal{H}(\widetilde{H}_R, U, r')$
    *if*$(\widetilde{H}_S {s_1}^T \neq \mathcal{G}(r', m', \widetilde{H}_R, \widetilde{H}_S))$
        *Return* Reject
    *else*
        *Return* $((m', s_1)$,Accept)
*end*
```

Note 1: The signcryption scheme is more efficient than individually signing and encrypting, for the following reasons:

1. The scheme uses the same key-pair for both confidentiality and authentication.
2. Avoids the use of independently generated randomness and ephemeral keys while individually signing and authenticating.

3. The scheme avoids the use of additional authenticating mechanism which is required for non-malleability of the ciphertext.

5.2 Security of the scheme

The security of the scheme is argued based on the security models given in definitions 9 and 10 in random oracle model.

Theorem 2 ***(Confidentiality.)*** *The given scheme is secure in the sense of SC-IND-CCA2 (under the random oracle model) if the syndrome decoding (SD) is hard to solve and the public key is computationally indistinguishable from the parity matrix of a random* $(n, k-1, t)$ *code R.*

We build the proof, by constructing a challenger $\mathcal{C}$ through a sequence of games **Game0, Game1,**$\cdots$. The complete proof is in Appendix A.

Theorem 3 ***(Unforgeability.)*** *The given scheme is unforgeable in the sense of SC-EUF-CMA (under the random oracle model) if the syndrome decoding (SD) is hard to solve and the public key is computationally indistinguishable from the parity matrix of a random* $(n, k-1, t)$ *code R.*

The proof of this theorem, follows the line of proof in theorem 1 and theorem 2. Hence, we do not elaborate on the same.

5.3 Parameters selection

We give some of the parameters for which our scheme will be practical and remain secure. The security proof explains the dependence of the scheme on the SD problem for security. The best-known attack for the signature is that by Bleichenbacher which is given in [16]. Also, the best known attack for syndrome decoding is Information-set decoding. A lower bound of the work factor for the attack is given in [16]. The parameters are selected according to [29]. In Table 1we present a few secure parameters. We also give the signing times of our signature scheme in table 1.

$(\log_2(n), t)$	**Security factor for Confidentiality in** $\log_2$	**Security factor for Authentication in** $\log_2$	**Time required for signing in the proposed scheme(approx. in** $\log_2$)
(18,13)	102.05	93.7	53.44
(19,12)	100.34	83.6	49.74
(20,11)	105.91	87.6	46.2

Table 1. Secure parameters for the scheme based on the bounds in [16]

6 Conclusion

In the paper, we introduced a weaker distinguishability assumption. This results in a modification of CFS signature, which allows a formal proof of security, reducing the unforgeability problem to syndrome decoding problem and the introduced assumption. Hence it overcomes the problems associated with the high rate distinguisher in [13]. This lays the foundation stone for the use of CFS schemes in various primitives. Also, existing primitives which have made use of the CFS signature can now be altered appropriately to achieve provable security. Also, in this paper we present a signcryption scheme. The scheme can be used in applications which require both confidentiality and authentication, instead of individually signing and encrypting, as the efficiency is improved. It can be noted that the key-construct in [3] can also be used for constructing the signature and the signcryption scheme. The parameters of the proposed signcryption could be improved by using the Parallel-CFS[15].

Also, it is interesting to investigate the possibility of using LDPC codes, and other codes with better decoding properties. The key construct may be sufficiently altered to enable the secure use of such codes. The subsequent improvement in efficiency has to be investigated further.

References

1. Jee Hea An, Yevgeniy Dodis, and Tal Rabin. On the security of joint signature and encryption. In Lars R. Knudsen, editor, *EUROCRYPT*, volume 2332 of *Lecture Notes in Computer Science*, pages 83–107. Springer, 2002.
2. Joonsang Baek, Ron Steinfeld, and Yuliang Zheng. Formal proofs for the security of signcryption. In *Public Key Cryptography - PKC 2002*, volume 2274 of *Lecture Notes in Computer Science*, pages 80–98. Springer, 2002.
3. Marco Baldi, Marco Bianchi, Franco Chiaraluce, Joachim Rosenthal, and Davide Schipani. Enhanced public key security for the mceliece cryptosystem. *CoRR*, abs/1108.2462, 2011.
4. Paulo S.L.M. Barreto and Rafael Misoczki. One-time signature scheme from syndromedecoding over generic error-correcting codes. Journal of Systems and Software 84(2), 2011. http://dx.doi.org/10.1016/j.jss.2010.09.016.
5. E. R. Berlekamp, R. J. McEliece, and H. C. Vantilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 1978.
6. Ian F. Blake and Chris Studholme. Properties of Random Matrices and Applications. http://www.cs.utoronto.ca/ cvs/coding/random.pdf, 2006. [Online; accessed 15-Dec-2006].
7. Anne Canteaut and Florent Chabaud. A new algorithm for finding minimum-weight words in a linear code: Application to mceliece's cryptosystem and to narrow-sense bch codes of length 511. *IEEE Transactions on Information Theory*, 44(1):367–378, 1998.
8. Irene Marquez Corbella and Ruud Pellikaan. Error-correcting pairs for a public-key cryptosystem. *CoRR*, abs/1205.3647, 2012.
9. Nicolas Courtois, Matthieu Finiasz, and Nicolas Sendrier. How to achieve a mceliece-based digital signature scheme. In Colin Boyd, editor, *ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 157–174. Springer, 2001.
10. Nicolas Courtois, Matthieu Finiasz, and Nicolas Sendrier. How to achieve a mceliece-based digital signature scheme. *INRIA Report*, 2001.
11. Léonard Dallot. Towards a concrete security proof of courtois, finiasz and sendrier signature scheme. In Stefan Lucks, Ahmad-Reza Sadeghi, and Christopher Wolf, editors, *WEWoRC*, volume 4945 of *Lecture Notes in Computer Science*, pages 65–77. Springer, 2007.
12. J.-C. Faugére, A Otmani, L. Perret, and J.-P. Tillich. Algebraic Cryptanalysis of McEliece variants with compact keys – toward a complexity analysis. In *SCC '10: Proceedings of the 2nd International Conference on Symbolic Computation and Cryptography*, pages 45–55, RHUL, June 2010.
13. Jean-Charles Faugère, Valérie Gauthier, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. A distinguisher for high rate mceliece cryptosystems. Information Theory Workshop (ITW), IEEE, 2011.
14. Jean-Charles Faugère, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. Algebraic cryptanalysis of mceliece variants with compact keys. In *EUROCRYPT*, pages 279–298, 2010.
15. Matthieu Finiasz. Parallel-cfs - strengthening the cfs mceliece-based signature scheme. In Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors, *Selected Areas in Cryptography*, volume 6544 of *Lecture Notes in Computer Science*, pages 159–170. Springer, 2010.
16. Matthieu Finiasz and Nicolas Sendrier. Security bounds for the design of code-based cryptosystems. In Mitsuru Matsui, editor, *ASIACRYPT*, volume 5912 of *Lecture Notes in Computer Science*, pages 88–105. Springer, 2009.
17. Ernst M. Gabidulin and Olaf Kjelsen. How to avoid the sidel'nikov-shestakov attack. In Andrew Chmora and Stephen B. Wicker, editors, *Error Control, Cryptology, and Speech Compression*, volume 829 of *Lecture Notes in Computer Science*, pages 25–32. Springer, 1993.
18. Niederreiter H. Knapsack-type cryptosystems and algebraic coding theory. *Prob Contr Inform Theor 15*, pages 159 – 166, 1986.

19. Gregory Kabatianskii, E. Krouk, and Ben J. M. Smeets. A digital signature scheme based on random error-correcting codes. In Michael Darnell, editor, *IMA Int. Conf.*, volume 1355 of *Lecture Notes in Computer Science*, pages 161–167. Springer, 1997.
20. P. J. Lee and E. F. Brickell. An observation on the security of mcelieces public- key cryptosystem. In *EUROCRYPT88*, page 275280, 1988.
21. Chung Ki Li, Guomin Yang, Duncan S. Wong, Xiaotie Deng, and Sherman S. M. Chow. An efficient signcryption scheme with key privacy. In Javier Lopez, Pierangela Samarati, and Josep L. Ferrer, editors, *EuroPKI*, volume 4582 of *Lecture Notes in Computer Science*, pages 78–93. Springer, 2007.
22. Yuan Xing Li, Robert H. Deng, and Xin mei Wang. On the equivalence of McEliece's and Niederreiter's public-key cryptosystems. *IEEE Transactions on Information Theory*, 40(1):271–, 1994.
23. John Malone-Lee and Wenbo Mao. Two birds one stone: Signcryption using rsa. In Marc Joye, editor, *CT-RSA*, volume 2612 of *Lecture Notes in Computer Science*, pages 211–225. Springer, 2003.
24. M.Garey and D.Johnson. Computers and intractability. a guide to the theory of incompleteness. 1979.
25. Ayoub Otmani and Jean-Pierre Tillich. An efficient attack on all concrete kks proposals. In *PQCrypto*, pages 98–116, 2011.
26. Raphael Overbeck and Nicolas Sendrier. Code-based cryptography. pages 95–137, 2008.
27. Erez Petrank and Ron M Roth. Is code equivalence easy to decide? *Information Theory, IEEE Transactions on*, 43, 1997.
28. McEliece R.J. A public-key cryptosystem based on algebraic coding theory. *JPL DSN Progress Report*, pages 114–116, 1978.
29. Nicolas Sendrier. Decoding one out of many. In *PQCrypto*, pages 51–67, 2011.
30. Jacques Stern. A new identification scheme based on syndrome decoding. In Douglas R. Stinson, editor, *CRYPTO*, volume 773 of *Lecture Notes in Computer Science*, pages 13–21. Springer, 1993.
31. Christian Wieschebrink. Two NP-complete problems in coding theory with an application in code based cryptography. In *IEEE International Symposium on Information Theory*, pages 1733–1737, 2006.
32. Yuliang Zheng. Digital signcryption or how to achieve cost(signature & encryption) << cost(signature) + cost(encryption). In *Advances in Cryptology, CRYPTO - 1997*, volume 1294 of *Lecture Notes in Computer Science*, pages 165–179. Springer, 1997.

A Proof of confidentiality for the signcryption scheme

Theorem 4 ***(Confidentiality.)*** *The given scheme is secure in the sense of SC-IND-CCA2 (under the random oracle model) if the syndrome decoding (*SD*) is hard to solve and the public key is computationally indistinguishable from the parity matrix of a random $(n, k-1, t)$ code R.*

Proof: We build the proof, by constructing a challenger $\mathcal{C}$ through a sequence of games **Game0, Game1,**$\cdots$. **Game0** is the adaptation of the protocol to the SC-IND-CCA2 game. Successive games are obtained by small modifications of the preceding games, in such a way that the difference of the adversarial advantage in consecutive games is easily quantifiable.
Let $q_{\mathcal{H}}, q_{\mathcal{G}}, q_s, q_u$ be the maximum number of queries made by the adversary $\mathcal{A}$ to the oracles for the hash queries $\mathcal{H}, \mathcal{G}$, the signcryption oracle and the de-signcryption oracle. We want to show that the advantage for the adversary $\mathcal{A}$ is bounded by the advantage of solving the hard problem SD for a random code with parity check matrix R.
To answer the hash queries and the signcryption and the de-signcryption queries, we maintain the lists, $\mathcal{G}^{list}, \mathcal{H}^{list}, \sigma^{list}$ and Λ. If there is no value in a list we denote its output by $\perp$.

- The list $\mathcal{G}^{list}$ contains a tuple $((x, s), a)$ indexed by $(r, m, \widetilde{H}_R, \widetilde{H}_S)$.
- The list $\mathcal{H}^{list}$ consists of strings $\rho \in \{0,1\}^l$ indexed by $(\widetilde{H}_R, U, r)$ where $\widetilde{H}_R$ and $\widetilde{H}_S$ are $(n-k) \times n$ sized parity check matrices, and $U \in \mathbb{F}_2^{n-k}$ and $r \in \mathbb{F}_2^n$ such that $wt(r) \leq t$.
- The σ^{list} (the signature list) consists of entries of the form $(m, \sigma = (U, V, s))$.
- The list Λ consists of indices r of $\Lambda(m)$ for which the simulator is able to produce a signature on $\mathcal{G}(m, \Lambda(m, \widetilde{H}_R, \widetilde{H}_S)), \widetilde{H}_R, \widetilde{H}_S)$, i.e., the list of r for which $\mathcal{G}(m, \Lambda(m, \widetilde{H}_R, \widetilde{H}_S)), \widetilde{H}_R, \widetilde{H}_S)$ is a decodable syndrome.

Game 0. This is the standard SC-IND-CCA2 game. The private and public key pair are obtained by running the key generation algorithm given the scheme, to obtain secret key (Q_U, H_U, P_U, H'_U), where $H_U \leftarrow \mathsf{Goppa}(n,k)$ (a binary Goppa code), and the public key $\widetilde{H}_U = Q'_U H_U P_U \oplus a_U b_U^T$. $\widetilde{H}_U$ is given to $\mathcal{A}$. Also, $\mathcal{A}$ is given access to the hash oracles $\mathcal{H}$ and $\mathcal{G}$. The signcryption oracle and designcryption oracle function as mentioned in the scheme. Let X_0 be the event that $\mathcal{A}$ wins Game 0. It is seen that Game 0 runs the SC-IND-CCA2 game on the proposed scheme perfectly. Therefore $Pr[X_0] - \frac{1}{2} = Adv_{\mathcal{A}}^{ind-cca}(n,k)$.

Game 1. (Simulation of hash oracles) In this game, the hash oracles for $\mathcal{G}$ and $\mathcal{H}$ are simulated, while the rest of the protocol is executed as in the previous game. The two oracles are simulated as follows:

For the query on $\mathcal{G}$ of the form $(r, m, \widetilde{H}_R, \widetilde{H}_S)$, we have two situations, depending on whether $r = \Lambda(m, \widetilde{H}_R, \widetilde{H}_S)$. The simulation of the oracle is given below:

```
Input: A tuple $(m, r, \widetilde{H}_R, \widetilde{H}_S)$
Output: A syndrome $x$
if $r \neq \Lambda(m, \widetilde{H}_R, \widetilde{H}_S)$ then
    if $s_1 = \perp$ then
        $s_1 \xleftarrow{R} \mathbb{F}_2^n$
        $x \leftarrow \widetilde{H}_S s_1^T$
        $\mathcal{G}^{list}(r, m, \widetilde{H}_R, \widetilde{H}_S) \leftarrow ((x, \perp), s_1)$
    end
    return $\mathcal{G}(r, m, \widetilde{H}_R, \widetilde{H}_S) = x$
else
    if $x = \perp$ then
        $s_1 \xleftarrow{R} \mathbb{F}_2^n$ such that $wt(s_1) = t$
        $x \leftarrow \widetilde{H}_S s_1^T$
        $\mathcal{G}^{list}(r, m, \widetilde{H}_R, \widetilde{H}_S) \leftarrow ((x, s_1), s_1)$
    end
    return $\mathcal{G}(r, m, \widetilde{H}_R, \widetilde{H}_S) = x$
end
```

For the query to $\mathcal{H}$ of the form $(\widetilde{H}_R, U, r)$, the challenger searches $\mathcal{H}^{list}$ for the tuple $(\widetilde{H}_R, U, r)$. If found, the corresponding value from the list is returned, else return a random string $\rho \xleftarrow{R} \{0,1\}^l$ and store the tuple $((\widetilde{H}_R, U, r),\rho)$ in $\mathcal{H}^{list}$. Let X_1 be the event that $\mathcal{A}$ wins Game 1.It is seen that, while the oracles are simulated in Game 1, the distribution of the output of these oracles remain unchanged (i.e., the randomness is maintained) from Game 0. Hence $Pr[X_1] = Pr[X_0]$

Game 2. (Simulation of the signcryption oracle.) The signcryption oracle is simulated as follows:

```
Input: the tuple $(m, \widetilde{H}_R, \widetilde{H}_U)$ Output: A signcrypted text $\sigma = (U, V, s_1)$
if $\Lambda(m, \widetilde{H}_R, \widetilde{H}_U) = \perp$ then
    $r \xleftarrow{R} \mathbb{F}_2^n$, such that $wt(r) \leq t$     $\Lambda(m, \widetilde{H}_R, \widetilde{H}_U) \leftarrow r$
end
$((x, s_1), s_1) \leftarrow \mathcal{G}(\Lambda(m, \widetilde{H}_R, \widetilde{H}_U), m, \widetilde{H}_R, \widetilde{H}_U)$
if$(s_1 = \perp)$ then     ABORT
else
    $r \leftarrow \Lambda(m, \widetilde{H}_R, \widetilde{H}_U)$     $\Lambda(m, \widetilde{H}_R, \widetilde{H}_U) \leftarrow \perp$     $U = \widetilde{H}_R r^T$
    $V = m \oplus \mathcal{H}(\widetilde{H}_R, U, r)$
end
```

Return $\sigma = (U, V, s_1)$. The simulation of the signcryption is an extension of the signing oracle simulation presented in the previous proof. It is thus, seen that the s_1 is a valid signature on m

for verification key $\widetilde{H}_U$. Also, the remaining signcrypted text is also valid, and follows from the signcrypt algorithm given in the scheme.

In Game 2, incoherence occurs if the oracle to $\mathcal{G}$ is queried initially for some $(r, m, \widetilde{H}_R, \widetilde{H}_S)$ such that later r is set to $\Lambda(m, \widetilde{H}_R, \widetilde{H}_S)$. This happens with the probability $\frac{q_s}{\binom{n}{t}}$ (since the indices Λ are defined only when the signcryption oracle is queried). It can be noted that this incoherence is the only scenario in which $\mathcal{F}$ can distinguish Game 2 from Game 1. Therefore, for the event X_2,that $\mathcal{A}$ wins the Game 2, we obtain, $|Pr[X_1] - Pr[X_2]| \leq \frac{q_s}{\binom{n}{t}}$

Game 3. (Simulation of the designcryption oracle.) For the designcryption oracle queried on $(s_1, U, V, \widetilde{H}_U, \widetilde{H}_S)$ the following is done:

- The challenger searches the $\mathcal{H}^{list}$ for the tuple $(\widetilde{H}_U, U, \lambda)$ such that $\widetilde{H}_U \lambda^T = U$. If it exists in the list, then the corresponding vector X is given as output. If no such tuple is found (i.e., the hash for such a tuple has not been queried) then it fixes $\lambda = \perp$ and gives the corresponding output from the hash oracle as X.
- It obtains $m' = V \oplus X$.
- The challenger then searches $\mathcal{G}^{list}$ for the tuple of the form $(\lambda, m', \widetilde{H}_U, \widetilde{H}_S)$ where $\widetilde{H}_U \lambda^T = U$ or $\lambda = \perp$. If the tuple is not in $\mathcal{G}^{list}$, the challenger adds it to the list.
- Now the challenger verifies if $\widetilde{H}_U {s_1}^T \overset{?}{=} \mathcal{G}(\lambda, m', \widetilde{H}_U, \widetilde{H}_S)$.If the condition holds and $\widetilde{H}_U \lambda^T = U$, then the challenger returns m' as the message. If condition holds but $\lambda = \perp$ then challenger ABORTS citing failure. If the condition does not hold at all, then the challenger returns $\perp$, as symbol of rejection of invalid signcrypted text.

In the above game, if the challenger aborts, it implies that the adversary created the signcrypted text without querying the hash oracles. Hence, the probability of aborting is $\frac{q_d}{2^l}$.This scenario (of ABORT) would not occur in Game 2. Hence, for the event X_3 that $\mathcal{A}$ wins Game 3, we have $|Pr[X_3] - Pr[X_2]| \leq \frac{q_d}{2^l}$

Game 4. (Changing the key generation algorithm) The adversary has access to the private keys of all users except the user U. Hence, for the other users, the keys are generated as in the scheme, and given to the adversary. For the user U, $\mathcal{C}$ selects the private key $H_U = R$. The public key is $\widetilde{H}_U = R'$, where $R'^T = [R^T | z^T]$ where $z \in_R \mathbb{F}_2^n$. The verification key $\widetilde{H}_U$ is given to the adversary $\mathcal{A}$. It follows from assumption 2 that $|Pr[X_3] - Pr[X_4]| \leq \mathsf{Adv}_{\mathcal{C}}^{\mathsf{Dist}}(n, k)$ where X_4 is the event that $\mathcal{A}$ wins Game 4.

Game 5. (Challenge ciphertext) The challenger takes the message m_b, and does the following to create the challenge, which would aid the challenger in solving the problem instance, syndrome s (where $wt(s) > 2t+1$). The challenger wants to find $r \in \mathbb{F}_2^n$ with $wt(r) \leq t$ such that $s = Rr^T$. The challenger generates the challenge cipher-text as follows:

- C sets $U^* = s'$, where $s'^T = [s^T | s_c]$ where s_c is a randomly generated bit.
- For the query on $\mathcal{H}$, $\mathcal{C}$ sets a special symbol $\top$, randomly generates a vector y and stores it in $\mathcal{H}^{list}$ as $(\widetilde{H}_U, U^*, \top, y)$. And for the query on $\mathcal{G}$, again uses the special symbol $\top$, also a random decodable syndrome (say x, with the decoded vector s_1) is given (just as in the simulation in $\mathcal{G}$ oracle and the Signcrypt oracle), and stores in $\mathcal{G}^{list}$ the tuple $(m_b, \top, \widetilde{H}_U, \widetilde{H}_S, (x, s_1), s_1)$. The signing is simulated just as in signcryption oracle.
- The challenger set $V = m_b \oplus y$

Also, the challenger has to now alter the answer to the hash queries in the following way:

- For the $\mathcal{H}$ oracle, for any query $(\widetilde{H}_U, \widetilde{H}_U s^T, r)$ where $\widetilde{H}_U r^T \neq \widetilde{H}_U s^T$, some random value is returned. If $\widetilde{H}_U r^T = U^*$ and $weight(r) \leq t$, then the value y is returned, and $\top$ is replaced by r in the tuple. The valid r thus obtained is the solution to the problem instance.

- For the $\mathcal{G}$ oracle, for any query $(m, r, \widetilde{H}_U, \widetilde{H}_S)$ if $\widetilde{H}_U r^T = U^*$ and $m = m_b$ output x,else output any random syndrome. The valid r thus obtained is the solution to the problem instance.

Just as in the proof of unforgeability, the decodability of U^* depends on correctly predicting s_c, which occurs with probability $\frac{1}{2}$. If X_5 be the event that $\mathcal{A}$ wins Game 5, we can clearly claim that, $|Pr[X_4] - Pr[X_5]| \leq \mathsf{Adv}_{\mathcal{C}}^{SD}(n,k)/2$ where $\mathsf{Adv}_{\mathcal{C}}^{SD}(n,k)$ is the advantage that some PPT algorithm $\mathcal{C}$ has at solving the syndrome decoding problem (SD) on R.

Game 6. (Challenge ciphertext) In this game, the challenger C again alters the procedure of producing the challenge ciphertext. For the ciphertext, the process of creating U and s is the same, but changes for V. The challenger C sets $V = z$,where $z \xleftarrow{R} \{0,1\}^l$. Clearly, now the challenge ciphertext generated is completely random. But, even in game 5, the ciphertext generated was random as we blinded the message with a completely random component. Hence, there is no change in the distribution of the ciphertext space, i.e., $Pr[X_5] = Pr[X_6]$, where X_6 is the event that $\mathcal{A}$ wins Game 6. Also, it can noted that the probability of correctly guessing the choice b by the adversary $\mathcal{A}$ is exactly half,i.e., $Pr[X_6] = \frac{1}{2}$

Accumulating all the above results and using triangular inequality we have the following result: $Adv^{ind-cca2}(\mathcal{A}) \leq \frac{q_s}{\binom{n}{t}} + \frac{q_d}{2^l} + \mathsf{Adv}_{\mathcal{C}}^{\mathsf{Dist}}(n,k) + \mathsf{Adv}_{\mathcal{C}}^{SD}(n,k)/2$. Hence, the advantage of the adversary is bound by the advantage of the challenger in solving the syndrome decoding problem and the weak distinguishability. □