Code-based Strong Designated Verifier Signatures: Security Analysis and a New Construction

Maryam Rajabzadeh Asaar

Department of Electrical and Computer Engineering, Science and Research Branch,
Islamic Azad University, Tehran, Iran.
asaar@srbiau.ac.ir

Abstract. Strong designated verifier signatures make the message authenticated only to a designated person called the designated verifier while privacy of the signer's identity is preserved. This primitive is useful in scenarios that authenticity, signer ambiguity and signer's privacy are required simultaneously such as electronic voting and tendering. To have quantum-attack-resistant strong designated verifier signatures as recommended in National Institute of Standards and Technology internal report (NISTIR 8105, dated April 2016), a provably secure code-based construction was proposed by Koochak Shooshtari et al. in 2016. In this paper, we show that this code-based candidate for strong designated verifier signatures does not have signer ambiguity or non-transferability, the main feature of strong designated verifier signatures. In addition, it is shown that it is not strongly unforgeable if a designated verifier transfers a signature to a third party. Then, a new proposal for strong designated verifier signatures based on coding theory is presented, and its security which includes strong unforgeability, signer ambiguity and privacy of the signer's identity properties is proved under Goppa Parameterized Bounded Decoding and the Goppa Code Distinguishing assumptions in the random oracle model.

Keywords: code-based signatures, strong designated verifier signatures, coding theory, provable security, random oracle model.

1 Introduction

Designated verifier signatures (DVS) are kinds of signatures that enable signers to prove the validity of a message to a specified entity called the designated verifier. Since the designated verifier by itself can generate these kinds of signatures on each message it wants, these types of signatures are non-transferable or signer ambiguous in the sense that the signature is created by the signer or the designated verifier. Hence, only the designated verifier which knows that it has not issued the signature is convinced of validity or invalidity of the message. In other words, the main feature of traditional signatures named as non-repudiation is not preserved while the message is authenticated. This concept was introduced by Jakobsson, Sako and Impagliazzoin in 1996 [14], and is used where a signer would like to determine who may be convinced by its signatures. For example, this primitive is a solution to the problem of electronic tendering of contracts [14]. Consider a scenario in which a company has a contract and would like put it to tender, and participants would like to propose their prices to win the contract. The participants signed their proposals to be authenticated. This procedure can be abused by tenderer in a way that it can show participant's proposal (the proposed price signed by a participant) to another participant to enable it to give a better proposal. With employing designated verifier signatures, each participant presents its proposal (the proposed price signed by designated verifier signatures by the participant and the company). In this case, there is no advantage for the company to show the participant's proposal to another participant since this kind of signature is ambiguous and nobody can be convinced who generates the proposal, the company or the participant. If participants' designated verifier signatures are captured before arriving to the company, signers can be identified since the company did not create the signature. To remove this problem in 1996, Jakobsson et al. [14] also briefly introduced another notion called strong designated verifier signatures (SDVS) which means that the designated verifier needs its secret key to verify the validity or invalidity of a signature. Therefore, nobody except for the designated verifier can specify who is the actual signer. In 2003, Saeednia et al. presented this notion formally [23], and in 2005 strengthen by Laguillaumie and Vergnaud [15]. Following Saeednia et al.'s work [23], some SDVS schemes [12, 13] have been proposed, where their security is proved based on hard problems in number theory.

In 1994, some research has been done by Shor [26] to show that quantum computers can break security of cryptographic algorithms based on number theory. With eminent emergence of quantum computers, number theory-based public key algorithms will be broken. In 2016, National Institute of Standards and Technology (NIST) in an internal report [6] emphasizes that it is vital to immigrate from currently used public key cryptosystems to quantum-attack-resistant counterparts. One of these recommended post-quantum cryptographic candidates is code-based cryptography.

In 1978, McEliece introduced the concept of code-based cryptography, and also presented the first code-based public key encryption scheme from the general decoding problem [18]. The proposed scheme [18] cannot be transformed to a signature scheme since it is not invertible. Niederreiter [20] modified McEliece code-based cryptosystem in 1986 such that it can be used to generate a signature scheme. Courtois, Finiasz and Sendrier [7] proposed the first practical code-based signature scheme called CFS scheme in 2001. They adapt the full domain hash approach of Bellare and Rogaway [3] to Niederreither encryption scheme [20] in a way that a message is concatenated with a counter before hashing to make hash values decodable. Although authors presented some security arguments, it does not support provable security. In 2008, Dallot [8] gave a slight modification to their signature scheme in a way that the counter is replaced with a random value, this new scheme is named modified CFS or Dallot scheme, and proved its security under Goppa Parameterized Bounded Decoding [4] and Goppa Code Distinguishing [24] assumptions in the random oracle model [3]. Following the work presented by Dallot [8], a few code-based signature schemes with additional properties such as identity-based signature [5], one-time signature [2], ring signature [29], threshold ring signature [28, 19, 9], blind signature [21], signcryption [17], undeniable signature [1] and strong designated verifier signature [25] have been proposed.

Contribution. In this paper, we show that the only provably secure candidate for code-based strong designated verifier signatures which was presented by Koochak Shooshtari et al. [25] does not have non-transferability or signer ambiguity. In addition, it is not strongly unforgeable if a designated verifier makes a designated verifier signature public verifiable which means that everyone can produce new strong designated verifier signatures for previously signed messages. Then, we propose a new strong designated verifier signature scheme from coding theory. It should be highlighted that our scheme compared to Koochak Shooshtari et al.'s scheme provides provable security for privacy of the signer's identity, while they claim that their scheme provides this property. In our construction, to have non-transferable signature scheme, we apply the paradigm "the signer or the designated verifier generates a signature" to Dallot signature scheme [8] to generate a publicly verifiable designated verifier signature. Hence, our proposal definitely will have signer ambiguity. Then, the encryption scheme presented by Niederreiter [20] is used to encrypt a piece of the designated verifier signature to create a strong form of that. Finally, its security is proved under hard problems in coding theory, Goppa Parametrized Bounded Decoding and the Goppa Code Distinguishing problems, in the random oracle model [3].

1.1 Organization of the paper

The rest of this paper is organized as follows. Section 2 presents background and complexity assumptions employed as the signature foundation, the outline of strong designated verifier signature algorithms and its security model. In Section 3, review of Koochak Shooshtari et al.'s scheme and its security weaknesses are presented. Our proposed scheme along with its formal security proof and efficiency analysis are given in Sections 4 and 5, respectively. Section 6 presents the conclusion.

2 Background

In this section, first the used notations in the paper are introduced, then, we review several fundamental backgrounds employed in this research, including coding theory, complexity assumptions, Dallot signature scheme, algorithms of a strong designated verifier signature scheme and its security model.

2.1 Notations

In this subsection, the notations used in the paper are defined.

- $\oplus$: X-OR operation.
- $w_H(y)$: the Hamming weight of a word y or the number of non-zero positions of y.
- y^T: transpose of a vector y.
- $\perp$: an empty string.
- $\top$: a special string.
- $\theta \leftarrow B(y_1, ...)$: the operation of assigning the output of algorithm B on inputs $y_1, ...$ to θ.
- $y \overset{\$}{\leftarrow} Y$: the operation of assigning a uniformly random element of Y to y.

2.2 Coding Theory

Let $\mathbb{F}_2$ be the field with two elements and a binary code $\mathcal{C}(n,k)$ be a linear subspace of dimension k of $\mathbb{F}_2^n$, where k and $n \in \mathbb{N}$. Elements of $\mathbb{F}_2^n$ and $\mathcal{C}$ are named words and codewords, respectively. Code $\mathcal{C}(n,k)$ is presented by a $(n-k) \times n$ binary parity check matrix H such that for a codeword $x \in \mathbb{F}_2^n$ belonged to $\mathcal{C}(n,k)$, we have $Hx^T = 0$ and the syndrome of a word $x \in \mathbb{F}_2^n$ is defined as $s = Hx^T$, where $s \in \mathbb{F}_2^{n-k}$. A syndrome s is said to be t-decodable if there exists a word $x \in \mathbb{F}_2^n$ such that $Hx^T = s$ and $w_H(x) \leq t$, where $t = \frac{n-k}{\log_2^n}$ is the error correcting capability of the code $\mathcal{C}(n,k)$.

Goppa codes are a subclass of alternant codes [16], and widely used in code-based cryptography. Goppa codes $G(n,k)$ of t error correcting capability are of length $n = 2^m$ and dimension $k = n - mt$, where m and $t \in \mathbb{N}$. It is assumed that $\mathcal{DEC}_H$ be the decoding algorithm of Goppa code $G(n,k)$ with the parity check matrix H.

2.3 Complexity assumptions

Hard problems and security assumptions used in the paper are defined as follows [8, 10, 24].

Definition 1. Goppa Parameterized Bounded Decoding (GPBD) problem. Given a random $(n-k) \times n$ binary matrix H and a syndrome $s \in \mathbb{F}_2^{n-k}$, output a word $x \in \mathbb{F}_2^n$ such that $w_H(x) \leq \frac{n-k}{\log_2^n}$ and $Hx^T = s$.

Definition 2. Goppa Parametreized Bounded Decoding (GPBD) assumption. The GPBD problem is (τ, ϵ)-hard if there is no algorithm C which runs in time at most τ and with probability at least ϵ breaks the GPBD problem.

Definition 3. Goppa Code Distinguishing (GD) problem. Given a $(n-k) \times n$ binary parity check matrix H, output a bit $b \in \{0,1\}$ indicating if H is a random binary parity check matrix or a Goppa code random binary parity check matrix.

The advantage of the distinguisher C is defined as follows.

$$\begin{aligned} &Adv_C^{GD}(n,k) = \Pr[1 \leftarrow C(H) \mid H \overset{\$}{\leftarrow} G(n,k)] - \\ &\Pr[1 \leftarrow C(H) \mid H \overset{\$}{\leftarrow} B(n,k)] \end{aligned} \tag{1}$$

Definition 4. Goppa Code Distinguishing (GD) assumption. The GD problem is (τ, ϵ)-hard if there is no algorithm C which runs in time at most τ breaks the GD problem with probability $Adv_C^{GD}(n,k) \geq \epsilon$.

2.4 Strong designated verifier signature algorithms

A strong designated verifier signature scheme consists of Setup, Sign, Ver and Sim algorithms as follows.

- Setup: Given a system security parameter λ, it outputs the set of users $\mathcal{U}$, the message space $\mathcal{M}$ and other public parameters, π. It also outputs users' public keys pk and each user has its secret key sk; i.e. $(Para, (sk, pk)) \leftarrow Setup(\lambda)$, where $Para = \{\mathcal{U}, \mathcal{M}, \pi\}$.

- Sign: Given the system's parameter $Para$, signer's secret key sk_s and its corresponding public key pk_s, designated verifier's public key pk_v and the message $M \in \mathcal{M}$ or equivalently an input tuple $(Para, sk_s, pk_s, pk_v, M)$, it outputs a signature θ; i.e. $\theta \leftarrow Sign(Para, sk_s, pk_s, pk_v, M)$.

- Ver: Given the system's parameter $Para$, signer's public key pk_s and designated verifier's public key pk_v and its corresponding secret key sk_v, the signature θ and the message M, returns 1 if θ is valid; otherwise, it returns 0; i.e. $\{0, 1\} \leftarrow Ver(Para, pk_s, pk_v, sk_v, \theta, M)$.

- Sim: Given the system's parameter $Para$, designated verifier's secret key sk_v and its corresponding public key pk_v, signer's public key pk_s and the message $M \in \mathcal{M}$ or equivalently an input tuple $(Para, sk_v, pk_v, pk_s, M)$, it outputs the simulated signature θ; i.e. $\theta \leftarrow Sim(Para, sk_v, pk_v, pk_s, M)$.

Correctness. The correctness of a SDVS scheme requires that for any (pk_s, sk_s), (pk_v, sk_v) and message $M \in \mathcal{M}$,

$$\Pr[1 \leftarrow Ver(Para, pk_s, pk_v, sk_v, Sign(Para, sk_s, pk_s, pk_v, M), M)] = 1,$$

and

$$\Pr[1 \leftarrow Ver(Para, pk_s, pk_v, sk_v, Sim(Para, sk_v, pk_v, pk_s, M), M)] = 1.$$

2.5 Security model of strong designated verifier signature schemes

A strong designated verifier signature scheme (SDVS) [14] should be existentially unforgeable under an adaptive-chosen-message attack, non-transferable (signer ambiguous) and strong which means that it has privacy of signer's identity [15].

Unforgeability. Unforgeability means that nobody other than the signer or the designated verifier can generate a valid designated verifier signature scheme. To give a formal definition for unforgeability of strong designated verifier signature, the following game between an adversary A and a challenger C is considered to be played [14].

1. Setup: Algorithm C runs the Setup algorithm with a security parameter λ to obtain system's parameter $Para$ and user's key pair (pk, sk), then it sends $(pk, Para)$ to A.

2. The adversary A in addition to making queries to random oracles adaptively issues a polynomially bounded number of questions to the Sign, Sim and Ver oracles as follows.

   - Sign: Adversary A can ask for a strong designated verifier signature on the tuple (pk_s, pk_v, M), where $M \in \mathcal{M}$ is the message, pk_s is signer's public key and pk_v is designated verifier's public key. Then, C returns the signature $\theta \leftarrow Sign(Para, sk_s, pk_s, pk_v, M)$.

   - Sim: Adversary A can ask for a simulated strong designated verifier signature on the tuple (pk_v, pk_s, M), where $M \in \mathcal{M}$ is the message, pk_s is signer's public key and pk_v is designated verifier's public key. Then, C returns the simulated signature $\theta \leftarrow Sim(Para, sk_v, pk_v, pk_s, M)$.

   - Ver: Adversary A can ask for the validity or invalidity of a (simulated) strong designated verifier signature on the tuple (pk_s, pk_v, θ, M), where $M \in \mathcal{M}$ is the message, θ is a (simulated) signature, pk_s is signer's public key and pk_v is designated verifier's public key. Then, C returns $\{0, 1\} \leftarrow Ver(Para, pk_s, pk_v, sk_v, \theta, M)$.

3. Eventually, A returns a strong designated verifier signature θ^* on the message M^* with respect to public keys pk_s and pk_v, and wins the forgery game if the two following conditions hold:

   Condition 1. The relation $1 \leftarrow Ver(Para, pk_s, pk_v, sk_v, \theta^*, M^*)$ holds.

Condition 2. Adversary A has not made Sign query for input of (pk_s, pk_v, M^*) and Sim query on input of (pk_v, pk_s, M^*).

The formal definition of existential unforgeability of strong designated verifier signatures is given in Definition 5.

Definition 5. An SDVS scheme is $(\tau, q_{ro}, q_s, q_{sim}, q_v, \epsilon)$-existentially unforgeable against adaptive chosen message attack if there is no adversary which runs in time at most τ, makes at most q_{ro} random oracle queries, q_s Sign queries, q_{sim} Sim queries and q_v Ver queries, and can win the forgery game with probability at least ϵ.

Privacy of the Signer's Identity (PSI). A strong designated verifier signature scheme has privacy of the signer's identity (PSI) if nobody other than the designated verifier says who generates the signature in case of having two or more potential signers. To have a formal definition for PSI, the following game between an adversary A and a challenger C is considered to be played [15].

1. Setup: Algorithm C runs the Setup algorithm with a security parameter λ to obtain system's parameter $Para$ and user's key pair (pk, sk), then it sends $(pk, Para)$ to A.

2. The adversary A in addition to making queries to random oracles adaptively issues a polynomially bounded number of questions to the Sign, Sim and Ver oracles as explained in the unforgeability game.

3. Algorithm C outputs two signer's public keys pk_{s_0} and pk_{s_1}, and designated verifier's public key pk_v to A.

4. Adversary A asks for a designated verifier signature on the message M. In response, C chooses $b \in \{0,1\}$ at random, and returns $\theta_b \leftarrow Sign(Para, sk_{s_b}, pk_{s_b}, pk_v, M)$ to A.

5. Adversary A continues to issue queries as in Step 2.

6. Finally, A outputs a bit b' and wins the game if the two following conditions hold.

Condition 1. The relation $b' = b$ holds.

Condition 2. Adversary A has not made Ver query on input of (b, θ_b, pk_v, M).

The formal definition for this property is given in Definition 6.

Definition 6. (Privacy of the Signer's Identity). An SDVS scheme is $(t, q_s, q_{sim}, q_v, \epsilon)$-PSI-secure if there is no adversary A which runs in time at most t; issues at most q_s Sign queries, q_{sim} Sim queries and q_v Ver queries, and can win the aforementioned game with probability that deviated from $\frac{1}{2}$ by more than ϵ.

Non-transferability (Signer ambiguity). A strong designated verifier signature scheme is said to be non-transferable or signer ambiguous if the signature generated by the signer is indistinguishable from the signature simulated by the designated verifier [14]. The formal definition of non-transferability is expressed in Definition 7.

Definition 7. An SDVS scheme is non-transferable (signer ambiguous) if Equation 2 holds for any probabilistic polynomial time distinguisher A, (pk_s, sk_s), (pk_v, sk_v) and any message $M \in \mathcal{M}$.

$$\left| \Pr \begin{bmatrix} \theta_0 \leftarrow Sign(Para, sk_s, pk_s, pk_v, M), \\ \theta_1 \leftarrow Sim(Para, sk_v, pk_v, pk_s, M), \\ b \leftarrow \{0,1\}, \\ b' \longleftarrow A(pk_s, pk_v, sk_s, sk_v, \theta_b) \\ : b' = b \end{bmatrix} - \frac{1}{2} \right| < \epsilon(\lambda), \qquad (2)$$

where $\epsilon(\lambda)$ is a negligible function in the security parameter λ, and the probability is taken over the randomness used in Sign and Sim, and the random coins used by A. If the probability is equal to $\frac{1}{2}$, the scheme is perfectly non-transferable [14].

3 Koochak Shooshtari et al.'s code-based strong designated verifier signature scheme and its security weaknesses

In this section, first we review the details of Koochak Shooshtari et al.'s scheme [25]; then, we show that it does not have signer ambiguity or non-transferability, and also it is not strongly unforgeable if a designated verifier transfers a signature to a party.

3.1 Details of Koochak Shooshtari et al.'s scheme

The scheme consists of the following algorithms:

1. Setup: The system parameters are as follows. Let n, k, m and $t \in \mathbb{N}$ be parameters for a Goppa code of length $n = 2^m$, dimension k and error correcting capability $t = \frac{n-k}{\log_2^n}$ such that t-decoding has complexity at least 2^λ for a security parameter λ. Let $g : \{0,1\}^* \to \{0,1\}^{n-k}$ be a random oracle and $f : \{0,1\}^n \to \{0,1\}^{n-k}$ be a deterministic function. It is assumed that $\tilde{H}$ be a $(n-k) \times n$ parity check matrix of a random binary Goppa code and $\mathcal{DEC}_{\tilde{H}}$ be its t-decoding algorithm. The public key is $pk = H = U\tilde{H}P$, and the secret key is $sk = (\mathcal{DEC}_{\tilde{H}}, U, P)$, where U is a random binary non-singular $(n-k)\times(n-k)$ matrix and P is a random $n \times n$ binary permutation matrix. Therefore, public parameters are $Para = \{n, k, m, t, g, f\}$.

2. Sign: To generate a strong designated verifier signature θ on the message $M \in \{0,1\}^*$, the signer chooses random numbers r_1 from $\{1, ..., 2^{n-k}\}$ and r_2 from $\mathbb{F}_2^n$ such that $w_H(r_2) = t$, computes $y = H_v r_2^T$ and $\alpha = g(M, f(r_2) \oplus r_1)$ and $x = \mathcal{DEC}_{\tilde{H}_s}(U_s^{-1}\alpha)P_s$. If $x = \perp$, it chooses another r_1, and repeats the signing procedure. The signature θ on the message M is (x, r_1, y).

3. Ver: Given $Para$, H_s, H_v and a signature $\theta = (x, r_1, y)$, the designated verifier first computes $r_2 = P_v^T \mathcal{DEC}_{\tilde{H}_v}(U_v^{-1}y)$, the signature θ on the message M is valid and outputs 1 if and only if $H_s x^T = g(M, f(r_2) \oplus r_1)$ and $w_H(x) \leq t$; otherwise, it outputs 0 and the signature is invalid.

4. Sim: To simulate a strong designated verifier signature θ on the message $M \in \{0,1\}^*$, the designated verifier chooses random numbers a from $\{1, ..., 2^{n-k}\}$ and b from $\mathbb{F}_2^n$ such that $w_H(b) = t$, computes $r_2' = P_v^T \mathcal{DEC}_{\tilde{H}_v}(U_v^{-1}g(a, b))$. If $r_2' = \perp$, it chooses another a, and repeats the simulation procedure. Then, sets $y' = g(a, b)$, $r_1' = f(r_2) \oplus r_1 \oplus f(r_2')$. The simulated signature θ on the message M is (x, r_1', y').

Correctness. The correctness of the signature $\theta = (x, r_1, y)$ is verified as follows, where $\alpha = g(M, f(r_2) \oplus r_1)$.

$$\begin{aligned}
& H_s x^T \\
& = (U_s \tilde{H}_s P_s)(\mathcal{DEC}_{\tilde{H}_s}(U_s^{-1}\alpha)P_s)^T \\
& = (U_s \tilde{H}_s P_s)(\mathcal{DEC}_{\tilde{H}_s}(U_s^{-1}g(M, f(r_2) \oplus r_1)P_s)^T \\
& = (U_s \tilde{H}_s P_s)P_s^T(\mathcal{DEC}_{\tilde{H}_s}U_s^{-1}g(M, f(r_2) \oplus r_1)^T) \\
& = U_s U_s^{-1} g(M, f(r_2) \oplus r_1) \\
& = g(M, f(r_2) \oplus r_1) \\
& = g(M, f(r_2') \oplus r_1').
\end{aligned} \tag{3}$$

3.2 Security analysis of Koochak Shooshtari et al.'s scheme

In what follows, we show that Koochak Shooshtari et al.'s scheme has some security weaknesses. Weakness 1 states that it is transferable of not signer ambiguous, and Weakness 2 indicates that it is not strongly unforgeable if a designated verifier signature is made public verifiable.

Weakness 1. Koochak Shooshtari et al.'s code-based strong designated verifier signature scheme is transferable. In other words, it is not signer ambiguous.

According to Definition 7, a SDVS scheme is non-transferable or signer ambiguous if the adversary A which has designated verifier's secret key and signer's secret key cannot tell who generates the signature (among two possible signers: the signer or the designated verifier). In our analysis, it is not necessary to consider all adversary's capabilities (having signer's secret key and designated verifier's secret key) as given in Definition 7. In fact, to show this weakness, just having designated verifier's secret key or receiving the decrypted some parts of the signature is sufficient. To do this, a designated verifier decrypts the value of y to obtain r_2, and gives $\theta = (x, r_1, y)$ along with r_2 to another party. In this case, the signature is made publicly verifiable by the designated verifier. Hence, everyone can check if $y = H_v r_2^T$, $w_H(r_2) = t$, $H_s x^T = g(M, f(r_2) \oplus r_1)$ and $w_H(x) \le t$ hold.

Since the designated verifier signature has been converted to a publicly verifiable signature under signer's public key, everyone can be sure that the signature θ is generated by the signer not the designated verifier. We should emphasize that the main reason for this weakness is that each designated verifier can simulate signatures only for messages that it receives their designated verifier signatures. In other words, it is impossible for a designated verifier to simulate valid signatures without having valid designated verifier signatures.

Another way to prove this weakness is that we show that probabilities of simulated signatures and real signatures are not the same. The probabilities of generating and simulating of a valid designated verifier signature are given as follows.

$$\theta = (x, r_1, y) : \begin{cases} r_1 \xleftarrow{\$} \mathbb{F}_2^{n-k} \\ r_2 \xleftarrow{\$} \mathbb{F}_2^{n} : w_H(r_2) \le t \\ y = H_v r_2^T \\ x = \mathcal{DEC}_{\tilde{H}_s}(U_s^{-1} g(M, f(r_2) \oplus r_1)) P_s \wedge x \neq \perp \end{cases} \tag{4}$$

$$\theta' = (x', r_1', y') : \begin{cases} a \xleftarrow{\$} \mathbb{F}_2^{n-k} \\ b \xleftarrow{\$} \mathbb{F}_2^{n} : w_H(b) \le t \\ r_2' = P_v^T \mathcal{DEC}_{\tilde{H}_v}(U_v^{-1} g(a, b)) \wedge r_2' \neq \perp \\ y' = g(a, b) \\ x' = x \\ r_1' = f(r_2') \oplus r_1 \oplus f(r_2) \end{cases} \tag{5}$$

Let $\overline{\theta}$ be a valid signature which is randomly chosen from the set of all valid signer's signatures intended to the verifier. Subsequently, we have distributions of probabilities as follows:

$$\Pr_{\theta} = \Pr_{r_1; r_2}\left[\overline{\theta} = \theta\right] = \frac{\sum_{i=0}^{t} \binom{n}{i}}{\binom{n}{t} 2^{(n-k)}}, \tag{6}$$

and

$$\Pr_{\theta'} = \Pr_{a; b}\left[\overline{\theta} = \theta'\right] = \frac{\sum_{i=0}^{t} \binom{n}{i}}{\binom{n}{t} 2^{(n-k)}} \gamma, \tag{7}$$

where the value of γ is the probability of receiving the designated verifier signature θ by a designated verifier. Therefore, without having the signature $\theta = (x, r_1, y)$, a designated verifier who does not have x, r_1 and y, which are required for signature simulation (refer to signature simulation algorithm), cannot simulate a valid designated verifier signature. As a consequence, the probability of simulating such a signature will be zero. This analysis shows that both distributions of probabilities are not the same. Hence, this scheme does not satisfy the non-transferability or signer ambiguity property.

Weakness 2. Koochak Shooshtari et al.'s code-based strong designated verifier signature scheme is not strongly unforgeable if a designated verifier signature is transferred, which means that everyone can simulates new signatures on previously signed messages.

If a designated verifier transfers a signature $\theta = (x, r_1, y)$ on the message M along with r_2 to another party, not only it can find out who is the real signer of a designated verifier signature, but also it can simulate new signatures on the same message. To generate a new signature on the same message M, that party chooses $\tilde{r}_2$ from $\mathbb{F}_2^n$ such that $w_H(\tilde{r}_2) = t$, computes $\tilde{y} = H_v \tilde{r}_2^T$, sets $\tilde{x} = x$ and sets $\tilde{r}_1 = f(r_2) \oplus r_1 \oplus f(\tilde{r}_2)$. Therefore, the simulated signature $\tilde{\theta}$ on the message M is $(\tilde{x}, \tilde{r}_1, \tilde{y})$ such that Ver algorithm on that signature returns 1 which means that the signature is valid.

4 Our code-based strong designated verifier signature scheme

In this section, first details of our proposed scheme is presented; then, its security is proved under GPBD and GD assumptions in the random oracle model [3].

4.1 Details of our proposed scheme

Our scheme consists of four algorithms as follows.

1. Setup: The system parameters are as follows. Let n, k, m and $t \in \mathbb{N}$ be parameters for a Goppa code of length $n = 2^m$, dimension k and error correcting capability $t = \frac{n-k}{\log_2^n}$ such that t-decoding has complexity at least 2^λ for a security parameter λ. Let $g : \{0,1\}^* \rightarrow \{0,1\}^{n-k}$, $h : \{0,1\}^* \rightarrow \{0,1\}^{n-k}$ be random oracles, and $f_{h(M),i}(.) : \{0,1\}^{n-k} \rightarrow \{0,1\}^{n-k}$ is a random permutation that encrypts messages of $n-k$ bits with keys $h(M)$ for $i \in \{s, v\}$, where the indices of s and v are used for a signer and a designated verifier, respectively. It is assumed that $\tilde{H}$ be a $(n-k) \times n$ parity check matrix of a random binary Goppa code and $\mathcal{DEC}_{\tilde{H}}$ be its t-decoding algorithm. The public key is $pk = H = U\tilde{H}P$, and the secret key is $sk = (\mathcal{DEC}_{\tilde{H}}, U, P)$, where U is a random binary non-singular $(n-k) \times (n-k)$ matrix and P is a random $n \times n$ binary permutation matrix. Therefore, public parameters are $Para = \{n, k, m, t, g(.), f_{h(M),i}(.), h\}$.

2. Sign: To generate a strong designated verifier signature θ on the message $M \in \{0,1\}^*$, the signer chooses random numbers r and x_v from $\mathbb{F}_2^n$ such that $w_H(r) \leq t$ and $w_H(x_v) \leq t$, computes $y = H_v r^T$ and $\alpha = f^{-1}_{h(M),s}(g(r, y, H_s, H_v, M) \oplus f_{h(M),v}(H_v x_v^T))$ and $x_s = \mathcal{DEC}_{\tilde{H}_s}(U_s^{-1}\alpha)P_s$. If $x_s = \perp$, it chooses another r, and repeats the signing procedure. The signature θ on the message M is (x_s, x_v, y).

3. Ver: Given $Para$, H_s, H_v and a signature $\theta = (x_s, x_v, y)$, the designated verifier first computes $r' = P_v^T \mathcal{DEC}_{\tilde{H}_v}(U_v^{-1} y)$, the signature θ on the message M is valid and outputs 1 if and only if $f_{h(M),s}(H_s x_s^T) \oplus f_{h(M),v}(H_v x_v^T) = g(r', y, H_s, H_v, M)$, and $w_H(r') \leq t$, $w_H(x_s) \leq t$ and $w_H(x_v) \leq t$; otherwise, it outputs 0 and the signature is invalid.

4. Sim: To simulate a strong designated verifier signature θ on the message $M \in \{0,1\}^*$, the designated verifier chooses random numbers r and x_s from $\mathbb{F}_2^n$ such that $w_H(r) \leq t$ and $w_H(x_s) \leq t$, computes $y = H_v r^T$ and $\alpha = f^{-1}_{h(M),v}(g(r, y, H_s, H_v, M) \oplus f_{h(M),s}(H_s x_s^T))$ and $x_v = \mathcal{DEC}_{\tilde{H}_v}(U_v^{-1}\alpha)P_v$. If $x_v = \perp$, it chooses another r, and repeats the simulation procedure. The simulated signature θ on the message M is (x_s, x_v, y).

Correctness. The correctness of the proposed scheme when the signature is generated by the Sign algotithem is as follows, and we use $y = H_v r^T$, $\alpha = f^{-1}_{h(M),s}(g(r, y, H_s, H_v, M) \oplus f_{h(M),v}(H_v x_v^T))$ and $x_s = \mathcal{DEC}_{\tilde{H}_s}(U_s^{-1}\alpha)P_s$ in what follows.

$$\begin{aligned} r' &= P_v^T \mathcal{DEC}_{\tilde{H}_v}(U_v^{-1} y) \\ &= P_v^T \mathcal{DEC}_{\tilde{H}_v}(U_v^{-1} H_v r^T) \\ &= P_v^T \mathcal{DEC}_{\tilde{H}_v}(U_v^{-1}(U_v \tilde{H}_v P_v) r^T) \\ &= P_v^T \mathcal{DEC}_{\tilde{H}_v}(\tilde{H}_v P_v r^T) \\ &= P_v^T P_v r^T \\ &= r. \end{aligned} \tag{8}$$

$$\begin{aligned}
&f_{h(M),s}(H_s x_s^T) \oplus f_{h(M),v}(H_v x_v^T) \\
&= f_{h(M),s}(H_s(\mathcal{DEC}_{\tilde{H}_s}(U_s^{-1}\alpha)P_s)^T) \oplus f_{h(M),v}(H_v x_v^T) \\
&= f_{h(M),s}(U_s \tilde{H}_s P_s(\mathcal{DEC}_{\tilde{H}_s}(U_s^{-1}\alpha)P_s)^T) \oplus f_{h(M),v}(H_v x_v^T) \\
&= f_{h(M),s}(U_s \tilde{H}_s P_s(P_s^T(\mathcal{DEC}_{\tilde{H}_s}(U_s^{-1}\alpha))^T)) \oplus f_{h(M),v}(H_v x_v^T) \\
&= f_{h(M),s}(U_s U_s^{-1}\alpha) \oplus f_{h(M),v}(H_v x_v^T) \\
&= f_{h(M),s}(\alpha) \oplus f_{h(M),v}(H_v x_v^T) \\
&= f_{h(M),s}(f_{h(M),s}^{-1}(g(r, y, H_s, H_v, M) \oplus f_{h(M),v}(H_v x_v^T))) \oplus f_{h(M),v}(H_v x_v^T) \\
&= g(r, y, H_s, H_v, M) \oplus f_{h(M),v}(H_v x_v^T) \oplus f_{h(M),v}(H_v x_v^T) \\
&= g(r, y, H_s, H_v, M) = g(r', y, H_s, H_v, M).
\end{aligned} \tag{9}$$

If $\theta = (x_s, x_v, y)$ is a valid strong designated verifier signature on the message M, $r' = r$, and the relation $f_{h(M),s}(H_s x_s^T) \oplus f_{h(M),v}(H_v x_v^T) = g(r', y, H_s, H_v, M)$ holds. Note that, if the signature is simulated by the Sim algorithm, its correctness can be shown in a similar way.

4.2 Analysis of the proposed scheme

In this subsection, the properties of the proposal, unforgeability, non-transferability and privacy of the signer's identity are proved in the random oracle model (see [3] for the background).

Unforgeability, non-transferability and privacy of the signer's identity of our proposed scheme are proved in Theorems 1, 2 and 3, respectively.

Theorem 1. *If the GPBD problem is $(\tau_{GPBD}, \epsilon_{GPBD})$-hard and GD problem is $(\tau_{GD}, \epsilon_{GD})$-hard, then the proposed scheme is $(\tau, q_g, q_f, q_s, q_{sim}, q_v, \epsilon)$- strongly unforgeable against adversary A such that*

$$\begin{aligned}
\epsilon_{GPBD} &\geq \frac{\epsilon - \epsilon_{GD} - \frac{(q_s+q_{sim})(2(q_s+q_{sim})+q_g))}{\binom{n}{t}}}{q_f}, \\
\tau_{GPBD} &\leq \tau + 3mt^2(q_v + q_s + q_{sim}),
\end{aligned} \tag{10}$$

where n, k, t and m are system's constants. In addition, q_g, q_f, q_s, q_{sim} and q_v are the number of queries to oracles $g(.)$, $f_{h(M),i}(.)$ Sign, Sim and Ver, respectively.

Proof. It is assumed that there is an adversary A against unforgeability of the scheme with success probability ϵ. We construct another algorithm C to solve GPBD problem with success probability ϵ_{GPBD}. Given a random binary matrix H^* and a random vector s^*, algorithm C outputs x^* such that $H^*(x^*)^T = s^*$ and $w_H(x^*) \leq t$. Note that substituting the public key of the signer or the designated verifier with a random binary matrix H^* changes the success probability of the simulator C with advantage at most ϵ_{BD} to solve the permuted Goppa code distinguishing.

The algorithm C runs Setup on a security parameter λ to generate public parameters $Para = \{n, k, m, t\}$, and gets a random instance of the GPBD problem, (n, k, m, t, H^*, s^*), to set signer's public key, H_s and designated verifier's public key, H_v, to H^*. Then, it invokes the adversary A on $Para$, H_v and H_s. The adversary A runs in time at most τ, makes q_g queries to the random oracle $g(.)$ and q_f queries to the cipher oracle $f_{h(M),i}(.)$, and makes q_s queries to the Sign oracle, q_{sim} queries to the Sim oracle and q_v queries to the Ver oracle, and can win the unforgeability game with probability at least $\epsilon_1 = \epsilon - \epsilon_{BD}$. Algorithm C maintains initially empty associative tables $T_g[.]$ to simulate the random oracle $g(.)$ and $T_f[.]$ to simulate the cipher oracle $f_{h(M),i}(.)$, and answers A's oracle queries as described below.

- $g(.)$ queries: If $T_g[.]$ is defined for query (r, y, H_s, H_v, M), then, C returns its value; otherwise, C chooses $T_g[r, y, H_s, H_v, M] \overset{\$}{\leftarrow} \{0,1\}^{n-k}$, and returns $g(r, y, H_s, H_v, M)$ to A.

- $f_{h(M),i}(.)$ queries or $f_{h(M),i}^{-1}(.)$ queries : If $T_f[.]$ is defined for each query $f_{h(M),i}^{-1} = H_i x_i^T$, then, C returns its value; otherwise, C chooses $T_f[H_i x_i^T] \overset{\$}{\leftarrow} \{0,1\}^{n-k}$, and returns its value to A. Similarly, for queries in form of $f_{h(M),i} = (f_{h(M),\neg i}(H_{\neg i} x_{\neg i}^T) \oplus g(r, y, H_s, H_v, M))$, C searches $T_f[.]$ to return the value of $f_{h(M),i}^{-1} = H_i x_i^T$; if its value has not been defined, it returns a random value from $\{0,1\}^{n-k}$.

Note that, C keeps the table $T_f[.]$ which remembers whether $f_{h(M),i}$ is the answer to $f_{h(M),i}(H_i x_i^T)$ or

$f^{-1}_{h(M),i}$ is the answer to $f^{-1}_{h(M),i}(f_{h(M),\neg i}(H_{\neg i}x^T_{\neg i}) \oplus g(r, y, H_s, H_v, M))$.

- Sign queries: For a query (H_s, H_v, M), C chooses random numbers r, x_s and x_v from $\mathbb{F}_2^n$ such that $w_H(r) \leq t$, $w_H(x_s) \leq t$ and $w_H(x_v) \leq t$, and computes $y = H_v r^T$ and $\alpha = f_{h(M),s}(H_s x_s^T) \oplus f_{h(M),v}(H_v x_v^T)$. If $T_g[r, y, H_s, H_v, M]$ has already been defined, then, C halts, returns $\perp$, and sets *bad* $\leftarrow$ *true*; otherwise, it sets $T_g[r, y, H_s, H_v, M] \leftarrow \alpha$. Hence, the strong designated verifier signature $\theta = (x_s, x_v, y)$ on the message M with respect to public keys H_s and H_v is sent to A.

- Sim queries: For a query (H_v, H_s, M), C chooses random numbers r, x_v and x_s from $\mathbb{F}_2^n$ such that $w_H(r) \leq t$, $w_H(x_v) \leq t$ and $w_H(x_s) \leq t$, and computes $y = H_v r^T$ and $\alpha = f_{h(M),s}(H_s x_s^T) \oplus f_{h(M),v}(H_v x_v^T)$. If $T_g[r, y, H_s, H_v, M]$ has already been defined, then, C halts, returns $\perp$, and sets *bad* $\leftarrow$ *true*; otherwise, it sets $T_g[r, y, H_s, H_v, M] \leftarrow \alpha$. Hence, the strong designated verifier signature $\theta = (x_s, x_v, y)$ on the message M with respect to public keys H_s and H_v is sent to A.

- Ver queries: For a query $(x_s, x_v, y, H_v, H_s, M)$, C looks for the tuple (r, y, H_v, H_s, M) at table $T_g[.]$ such that $y = H_v r^T$ and $w_H(r) \leq t$ to obtain $g(r, y, H_v, H_s, M)$, then searches the table $T_f[.]$ for queries in form of $H_s x_s^T$ and $H_v x_v^T$ queries to obtain $f_{h(M),s} = f_{h(M),s}(H_s x_s^T)$ and $f_{h(M),v} = f_{h(M),v}(H_v x_v^T)$, and then checks if $f_{h(M),s}(H_s x_s^T) \oplus f_{h(M),v}(H_v x_v^T) = g(r, y, H_v, H_s, M)$ holds, and $w_H(x_v) \leq t$ and $w_H(x_s) \leq t$ or no. If all relations hold, the designated verifier signature is valid and returns 1; otherwise, it returns 0 to A.

- Finally, it is assumed that A outputs a signature $\theta^* = (x_s^*, x_v^*, y^*)$ on the message M^* with respect to public keys $pk_s = H_s$ and $pk_v = H_v$ with probability ϵ_1, and wins if $1 \leftarrow Ver(Para, pk_s, pk_v, sk_v, \theta^*, M^*)$, and A has not made Sign query for input of (pk_s, pk_v, M^*) and Sim query on input of (pk_v, pk_s, M^*).

The probability of A in returning a forged signature θ^* is $\epsilon_2 = \Pr[E_0]\Pr[E_1|E_0]$, where definitions of events E_0 and E_1 are given as follows.

- E_0 : Algorithm C does not abort as a result of Sign and Sim simulations.
- E_1: Adversary A wins the forgery game.

To lower-bound the probability of $\Pr[E_0]$ and $\Pr[E_1|E_0]$, we need to compute the probability $\Pr[\neg bad]$, where the event *bad* indicate that C aborts in the Sign and Sim simulation. These probabilities are computed as follows.

Claim 1. $\Pr[E_0] = \Pr[\neg bad] \geq 1 - \frac{(q_s+q_{sim})(q_g+2q_s+2q_{sim})}{\binom{n}{t}}$.

Proof. The probability of the event E_0 is computed as follows.

- Case 1. If (r, y, H_v, H_s, M) generated in one Sign or Sim simulation has been occurred by chance in a previous query to the oracle $g(.)$, we have *bad* $\leftarrow$ *true*. Since there are at most $q_g + q_s + q_{sim}$ entries in the table $T_g[.]$ and the number of r randomly chosen from $\mathbb{F}_2^n$, $w_H(r) = t$, is $\binom{n}{t}$, the probability of this event for one Sign query or Sim query is at most $\frac{(q_g+q_s+q_{sim})}{\binom{n}{t}}$. Hence, the probability of this event for $q_s + q_{sim}$ queries is at most $\frac{(q_s+q_{sim})(q_g+q_s+q_{sim})}{\binom{n}{t}}$.

- Case 2. If C previously used the same randomness r from $\mathbb{F}_2^n$, $w_H(r) = t$, in one Sign or Sim simulation, we have *bad* $\leftarrow$ *true*. Since there are at most $q_s + q_{sim}$ Sign and Sim simulations, this probability is at most $\frac{(q_s+q_{sim})}{\binom{n}{t}}$. Therefore, for $q_s + q_{sim}$ Sign and Sim queries, the probability of this event is at most $\frac{(q_s+q_{sim})^2}{\binom{n}{t}}$.

Claim 2. $\Pr[E_1|E_0] \geq \epsilon_1$.

Proof. The value of $\Pr[E_1|E_0]$ is the probability that A wins the forgery game provided that C does not abort as a result of A's Sign (Sim) and Ver queries. If C did not abort as a result of A's queries, all its responses to those queries are valid. Therefore, by hypothesis A will win the forgery game with probability at least ϵ_1.

Therefore, the probability that A returns a tuple $(x^*_s, x^*_v, y^*, g, f^{-1}_{h(M^*),s}, f_{h(M^*),v})$ is at least

$$\epsilon_1 - \frac{(q_s + q_{sim})(2(q_s + q_{sim}) + q_g))}{\binom{n}{t}}.$$

Since $g(.)$ is a random oracle, the probability of the event that $g = g(r^*, y^*, H_s, H_v, M^*)$ is less than $\frac{1}{2^{(n-k)}}$, unless it is asked during the attack. Hence, in what follows it is likely that the query $(r^*, y^*, H_s, H_v, M^*)$ has been asked during a successful attack. Similarly, $f_{h(M^*),i}(.)$ or $f^{-1}_{h(M^*),i}(.)$, $i \in \{s, v\}$ is the cipher oracle and the the probability of the event that $f_{h(M^*),v} = f_{h(M^*),v}(H_v {x^*}^T_v)$ and $f^{-1}_{h(M^*),s} = f^{-1}_{h(M^*),s}(f_{h(M^*),v}(H_v {x^*}^T_v) \oplus g(r^*, y^*, H_s, H_v, M^*))$ is less than $\frac{2}{2^{(n-k)}}$, unless they are asked during the attack.

The lower bound of probability of wining the forgery game after making queries to $g(.)$ and $f_{h(M^*),i}(.)$ oracles is at least

$$\epsilon_1 - \frac{(q_s + q_{sim})(2(q_s + q_{sim}) + q_g))}{\binom{n}{t}} - \frac{3}{2^{(n-k)}}.$$

Algorithm C employs A, guesses an index $1 \leq \beta \leq q_f$, and hopes that β be the index of the query $f_{h(M^*),s} = (g(r^*, y^*, H_s, H_v, M^*) \oplus f_{h(M^*),v}(H_v {x^*}^T_v))$ to oracle $f^{-1}_{h(M^*),i}(.)$. Then, C responses with s^* for that query, and the probability of this event is $\frac{1}{q_f}$. Since the tuple $(x^*_s, x^*_v, y^*, g, f^{-1}_{h(M^*),s}, f_{h(M^*),v})$ is a valid signature, we have $w_H(x^*_s) \leq t$, $w_H(x^*_v) \leq t$ and

$$H_s {x^*}^T_s = f^{-1}_{h(M^*),s}(f_{h(M^*),v}(H_v {x^*}^T_v) \oplus g(r^*, y^*, H_s, H_v, M^*)).$$

With substituting the value of $f^{-1}_{h(M^*),s}(f_{h(M^*),v}(H_v {x^*}^T_v) \oplus g(r^*, y^*, H_s, H_v, M^*))$ with s^*, we have

$$H_s {x^*}^T_s = s^*$$

with probability at least

$$\frac{\epsilon - \epsilon_{GD} - \frac{(q_s+q_{sim})(2(q_s+q_{sim})+q_g))}{\binom{n}{t}}}{q_f}.$$

As a consequence, $x^*_s = x^*$ is a t-decodable of s^*.

Algorithm C's run-time τ_{GPBD} is A's run-time, τ, plus the time required to respond to q_s Sign queries, q_{sim} Sim queries and q_v Ver queries. Each Sign, Sim or Ver simulation takes three syndrome computations, where each one costs about mt^2. Therefore, C's run-time is $\tau_{GPBD} \leq \tau + 3mt^2(q_v + q_s + q_{sim})$. This completes the proof.

Theorem 2. *The proposed scheme is non-transferable.*

Proof. To prove non-transferability of the proposal, we show that the designated verifier signature simulated by the designated verifier is indistinguishable from the one created by the signer. Hence, we have to show that probabilities of the two following signatures are the same.

$$\theta = (x_s, x_v, y) : \begin{cases} r \xleftarrow{\$} \mathbb{F}_2^n : w_H(r) \leq t \\ x_v \xleftarrow{\$} \mathbb{F}_2^n : w_H(x_v) \leq t \\ y = H_v r^T \\ \alpha = f^{-1}_{M,s}(g(r, y, H_s, H_v, M) \oplus f_{M,v}(H_v x^T_v)) \\ x_s = \mathcal{DEC}_{\tilde{H}_s}(U_s^{-1}\alpha)P_s \wedge x_s \neq \perp \end{cases} \qquad (11)$$

$$\theta' = (x'_s, x'_v, y') : \begin{cases} r' \xleftarrow{\$} \mathbb{F}_2^n : w_H(r') \leq t \\ x'_s \xleftarrow{\$} \mathbb{F}_2^n : w_H(x'_s) \leq t \\ y' = H_v {r'}^T \\ \alpha = f^{-1}_{M,v}(g(r', y', H_s, H_v, M) \oplus f_{M,s}(H_s x'^T_s)) \\ x'_v = \mathcal{DEC}_{\tilde{H}_v}(U_v^{-1}\alpha)P_v \wedge x'_v \neq \perp \end{cases} \qquad (12)$$

Let $\overline{\theta}$ be a valid signature which is randomly chosen from the set of all valid signer's signatures intended to the verifier. Subsequently, we have distributions of probabilities as follows:

$$\Pr_{\theta} = \Pr_{r;x_v}\left[\overline{\theta} = \theta\right] = (\frac{\sum_{i=1}^{t}\binom{n}{i}}{\binom{n}{t}})^2, \tag{13}$$

and

$$\Pr_{\theta'} = \Pr_{r';x'_s}\left[\overline{\theta} = \theta'\right] = (\frac{\sum_{i=1}^{t}\binom{n}{i}}{\binom{n}{t}})^2, \tag{14}$$

This analysis means that both distributions of probabilities are the same. Hence, our proposal satisfies non-transferability or signer ambiguity property.

Theorem 3. *If the GPBD problem is $(\tau_{GPBD}, \epsilon_{GPBD})$-hard and GD problem is $(\tau_{GD}, \epsilon_{GD})$-hard, then the proposed scheme is $(\tau, q_g, q_f, q_s, q_{sim}, q_v, \epsilon)$-PSI secure against adversary A such that*

$$\begin{array}{l} \epsilon_{GPBD} \geq \epsilon - \epsilon_{BD} - \frac{3}{2^{(n-k)}}, \\ \tau_{GPBD} \leq \tau + (q_s + q_{sim})(t!(m^3t^2) + 2mt^2), \end{array} \tag{15}$$

where n, k, t and m are system's constants. In addition, q_g, q_f, q_s, q_{sim} and q_v are the number of queries to oracles $g(.)$, $f_{h(M),i}(.)$ Sign, Sim and Ver, respectively.

Proof. It is assumed that there is an adversary A against privacy of the signer's identity of the scheme with success probability $\epsilon + \frac{1}{2}$. Then, we construct another algorithm C to solve GPBD problem with success probability ϵ_{GPBD}. Given a random binary matrix H^* and a random vector s^*, algorithm C outputs r^* such that $H^*(r^*)^T = s^*$ and $w_H(r^*) \leq t$. Note that substituting the public key of the designated verifier with a random binary matrix H^* changes the success probability of the simulator C with advantage at most ϵ_{BD} to solve the permuted Goppa code distinguishing.

The algorithm C runs Setup on a security parameter λ to obtain the public parameters $Para = \{n, k, m, t\}$, signers' public keys H_{s_0} and H_{s_1} and their corresponding secret keys. It gets a random instance of the GPBD problem, (n, k, m, t, H^*, s^*), and sets designated verifier's public key to H^*, and invokes the adversary A on $Para$ signers's public keys H_{s_0} and H_{s_1} and designated verifier public key $H_v = H^*$. The adversary A runs in time at most τ, makes q_g to the random oracle $g(.)$, q_f queries to the cipher oracle $f_{h(M),i}(.)$, q_s queries to the Sign oracle, q_{sim} queries to the Sim oracle and q_v queries to the Ver oracle, and can win the PSI game with probability at least $\epsilon_1 = \epsilon - \epsilon_{BD} + \frac{1}{2}$. Algorithm C maintains initially empty associative tables $T_g[.]$ and $T_f[.]$ to simulate oracles $g(.)$ and $f_{h(M),i}(.)$, respectively. Also, C keeps a list $T_s[.]$ to store issued signatures, and answers A's random and cipher oracle queries as explained in Theorem 1. Algorithm C responses Sign (Sim) and Ver queries as explained below.Note that since the scheme is perfectly non-transferable, it is enough to consider just Sign queries instead of considering Sign and Sim queries to simplify the proof.

– Sign queries: For a query (b, H_{s_b}, H_v, M), $b \in \{0, 1\}$, C generates the strong designated verifier signature $\theta_b = (x_{s_b}, x_v, y)$ on the message M following the real Sign algorithm since it knows signer's secret key. Then, θ_b is sent to A.

– Ver queries: For a query $(b, \theta_b, H_v, H_{s_b}, M)$, $b \in \{0, 1\}$, if the signature θ_b was ever returned by C, and it is in $T_s[.]$, it returns 1 meaning that the signature is valid; otherwise, it returns 0 meaning that the signature is invalid.

– Adversary A asks for the challenge strong designated verifier signature on the message M, and C in its response chooses $b \in \{0, 1\}$ at random, and computes θ_b in a way that it chooses x_v from $\mathbb{F}_2^n$ such that $w_H(x_v) \leq t$, sets $y = s^*$, makes $g(.)$ query on the tuple $(\top, s^*, H_{s_b}, H^*, M)$ to obtain α, where $\top$ is a special string. Then, it computes $x_s = \mathcal{DEC}_{\tilde{H}_s}(U_s^{-1}\alpha)P_s$. If $x_s = \perp$, it chooses another α for $T_g[\top, s^*, H_{s_b}, H^*, M]$, and repeats the signing procedure. Then, the signature θ_b and public keys H_{s_0}, H_{s_1} and H_v are sent to A.

- Adversary A continues to make a number of queries to the random oracle, cipher oracle, Sign (Sim) and Ver oracles with exception that it cannot query Ver oracle on the challenge signature θ_b. During simulation of the signature scheme, C updates its answers to the random oracle queries. For $g(.)$ queries in form of (r, y, H_s, H_v, M), where $y \neq s^*$, a random value from $\{0,1\}^{n-k}$ is returned. If $y = s^*$ and $w_H(r) \leq t$, a random value from $\{0,1\}^{n-k}$ is returned and $\top$ is replaced by r.

- Finally, it is assumed that A returns $b' = b$ with non-negligible probability $\epsilon_1 = \epsilon - \epsilon_{BD}$.

The probability of C in returning r^*, the solution to the random instance of the GPBD problem is computed as follows. Algorithm C simulates Sign (Sim) and Ver queries perfectly since it knows signers' secret keys. In a successful attack, A has to make query to the $g(.)$ oracle on the tuple $(r^*, s^*, H_{s_b}, H^*, M)$, where $s^* = H^* r^{*T}$. Otherwise, since $g(.)$ is a random oracle, its outputs are random, and A does not have information about $g = g(r^*, s^*, H_{s_b}, H^*, M)$, unless it guesses its value with probability at most $2^{-(n-k)}$. Similarly, $f_{h(M),i}(.)$ is a cipher oracle, and the probability that $f_{h(M),s_b} = f_{h(M),s_b}(H_{s_b} x_{s_b}^T)$ and $f_{h(M),v} = f_{h(M),v}(H_v x_v^T)$ is less than $\frac{2}{2^{(n-k)}}$. Hence, it is likely that all these queries are asked during a successful attack.

As a consequence, the solution r^*, $w_H(r^*) \leq t$ to the problem instance s^* is obtained with probability

$$\epsilon_{GPBD} \geq (\epsilon - \epsilon_{BD} + \frac{1}{2}) - (\frac{1}{2} + \frac{3}{2^{n-k}}) = \epsilon - \epsilon_{BD} - \frac{3}{2^{n-k}}.$$

Algorithm C's run-time τ_{GPBD} is A's run-time, τ, plus the time required to respond to q_s Sign queries, q_{sim} Sim queries and q_v Ver queries. Each Sign or Sim simulation takes two syndrome computations which each one costs mt^2 bit operations and $t!$ decodings which each one needs m^3t^2 bit operations. Therefore, C's run-time is $\tau_{GPBD} \leq \tau + (q_s + q_{sim})(t!(m^3t^2) + 2mt^2)$. This completes the proof.

5 Efficiency Analysis

Each signer's public key, H, is a $2^m \times mt$ matrix which takes $mt2^m$ bits to be stored, and also the signature θ in our scheme consists of three elements x_s, x_v and r, where all are $n = 2^m$-bit vectors of weight t which each one takes $\log_2^{\binom{2^m}{t}}$ bits to be stored. Hence, the size of the signature θ is $3 \log_2^{\binom{2^m}{t}}$. Computational cost of signature scheme is computed as follows. Signature generation or signature simulation, Sign or Sim algorithm, takes two syndrome computations which each one costs mt^2 bit operations and $t!$ decodings which each one needs m^3t^2 bit operations. As a consequence, the strong designated verifier signature generation or simulation costs $t!(m^3t^2) + 2mt^2$. Signature verification, Ver algorithm, needs two syndrome computations and one decoding, and so it costs $2mt^2 + m^3t^2$. Computational costs for Sign, Sim and Ver and signature size are summarized in Table 1.

Table 1. Computation costs of our scheme

Computational Costs	Sign Cost	Sim Cost	Ver Cost	Signature Size
The proposal	$t!(m^3t^2)$ $+2mt^2$	$t!(m^3t^2)$ $+2mt^2$	(m^3t^2) $+2mt^2$	$3 \log_2^{\binom{2^m}{t}}$

To have an efficient signature scheme, it is recommended that the number of decoding computations for signing messages are reduced, so the parameter t should be small as possible. In 2001, Courtois et al. [7] proposed to use $m = 16$ and $t = 9$, but these parameters are not resistant against the generalized birthday attack [11]. In 2009, Finiasz and Sendrier [11] recommended $m = 22$ and $t = 9$, and with these parameters, the security level is $2^{81.7}$ and the generalized birthday attack can be prevented. For parameters $m = 22$ and $t = 9$, each signer's public key is about 99MBytes, signature size will be 530 bits, Sign or Sim takes $2^{39.8}$ bit operations, and Ver costs $2^{19.8}$ bit operations. However, the size of public keys in code-based cryptography such as Dallot scheme [8] and our scheme is large, some efforts have been done to reduce it [27, 22].

6 Conclusion

In this paper, first we showed that the only candidate for strong designated verifier signature scheme based on coding theory does not have signer ambiguity or non-transferability and it is not strongly unforgeable if a designated verifier makes a signature public verifiable. Then, in order to have a code-based strong designated verifier signature scheme as recommended by NISTIR 8105, a new construction was proposed, and its security was proved under Goppa Parameterized Bounded Decoding and the Goppa Code Distinguishing assumptions in the random oracle model. It should be emphasized that this post-quantum primitive can be widely employed in electronic e-commerce services and auction protocols. As a future work, we focus on presenting its extension to other forms of strong designated verifier signatures based on coding theory such as code-based (strong) designated verifier proxy signature scheme.

7 Acknowledgements

The authors would like to appreciate anonymous reviewers for their valuable comments on this work.

References

1. C. Aguilar-Melchor, S. Bettaieb, P. Gaborit, and J. Schrek. A code-based undeniable signature scheme. In *Proc. of the 14th IMA Int. Conf. on Cryptography and Coding-IMACC 2013*, pages 99–119, Oxford, UK, 17-19 December 2013. Springer-Verlag, Berlin.
2. P.S.L.M. Barreto, R. Misoczki, and M. A. Simplicio Jr. One-time signature scheme from syndrome decoding over generic error-correcting codes. *Journal of Systems and Software*, 84(2):198–204, 2011.
3. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proc. of the 1st ACM Conf. on Computer and Communications Security (CCS 1993)*, pages 62–73, Fairfax, VA, USA, 3-5 November 1993. ACM, New York, NY.
4. E.R. Berlekamp, R.J. McEliece, and H.C.A. Van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.
5. P. L. Cayrel, P. Gaborit, and M. Girault. Identity-based identification and signature schemes using correcting codes. In *Proc. of the Int. Workshop on Coding and Cryptology (WCC 2007)*, pages 69–78, Versailles, France, 16-20 April 2007. Springer-Verlag, Berlin.
6. L. Chen, S. Jordan, Y.K. Liu, D. Moody, R. Peralta, R.Perlner, and D. Smith-Tone. Report on post-quantum cryptography. Internal Report 8105, National Institute of Standards and Technology, http://dx.doi.org/10.6028/NIST.IR.8105, April 2016.
7. N. T. Courtois, M. Finiasz, and N. Sendrier. How to achieve a McEliece-based digital signature scheme. In *Proc. of the 7th Int. Conf. on the Theory and Application of Cryptology and Information Security-Advances in Cryptology-ASIACRYPT 2001*, pages 157–174, Gold Coast, Australia, 9-13 December 2001. Springer-Verlag, Berlin.
8. L. Dallot. Towards a concrete security proof of Courtois, Finiasz and Sendrier signature scheme. In *Proc. of the 2nd Western European Workshop on Research in Cryptology-WEWoRC 2007*, pages 65–77, Bochum, Germany, 4-6 July 2008. Springer-Verlag, Berlin.
9. L. Dallot and D. Vergnaud. Provably secure code-based threshold ring signatures. In *Proc. of the 12th Int. Conf. on the Cryptography and Coding*, pages 222–235, Cirencester, UK, 15-17 December 2009. Springer-Verlag, Berlin.
10. M. Finiasz. Nouvelles constructions utilisant des codes correcteurs derreurs en cryptographie clef publique. In *These de doctorat, cole Polytechnique*, Paris, France (in French), October 2004.
11. M. Finiasz and N. Sendrier. Security bounds for the design of code-based cryptosystems. In *Proc. of the 15th Int. Conf. on the Theory and Application of Cryptology and Information Security-Advances in Cryptology-ASIACRYPT 2009*, pages 88–105, Tokyo, Japan, 6-10 December 2009. Springer-Verlag, Berlin.
12. Q. Huang, D. S. Wong G. Yang, and W. Susilo. Efficient strong designated verifier signature schemes without random oracle or with non-delegatability. *International Journal of Information Security*, 10(6):373–385, 2011.
13. X. Huang, W. Susilo, Y. Mu, and F. Zhang. Short (identity-based) strong designated verifier signature schemes. In *Proc. of the 2nd Int. Conf. on Information Security Practice and Experience, ISPEC 2006*, pages 214–225, Hangzhou, China, 11-14 April 2006. Springer-Verlag, Berlin.
14. M. Jakobsson, K. Sako, and R. Impagliazzo. Designated verifier proofs and their applications. In *Proc. of the Int. Conf. on Theory and Application of Cryptographic Techniques, Advances in Cryptology EUROCRYPT 1996*, pages 143–154, Saragossa, Spain, 12-16 May 1996. Springer-Verlag, Berlin.

15. F. Laguillaumie and D. Vergnaud. Designated verifier signatures: anonymity and efficient construction from any bilinear map. In *Proc. of the 4th Int. Conf. on Security in Communication Networks, SCN 2004*, pages 105–119, Amalfi, Italy, 8-10 September 2004. Springer-Verlag, Berlin.
16. F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes.* North-Holland, 1977.
17. K. P. Mathew, S. Vasant, and C. P. Rangan. A provably secure signature and signcryption scheme using the hardness assumption in coding theory. In *Proc. of the 16th Int. Conf. on Information Security and Cryptology-ICISC 2013*, pages 99–119, Seoul, Korea, 27-99 November 2013. Springer-Verlag, Berlin.
18. R.J. McEliece. A public-key cryptosystem based on algebraic coding theory. *DSN Progress Report 42-44*, (2):114–116, 1978.
19. C.A. Melchor, P.L. Cayrel, P. Gaborit, and F. Laguillaumie. A new efficient threshold ring signature scheme based on coding theory. *IEEE Transactions on Information Theory*, 57(7):4833–4842, 2011.
20. H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory*, 15(2):159–166, 1986.
21. R. Overbeck. A step towards QC blind signatures. IACR Cryptology ePrint Archive, 2009.
22. P. S. L. M. Barreto R. Misoczki. Compact McEliece keys from Goppa codes. In *Proc. of the 16th Int. Workshop on Selected Areas in Cryptography, SAC 2009*, pages 376–392, Calgary, Canada, 13-14 August 2009. Springer-Verlag, Berlin.
23. S. Saeednia, S. Kremer, and O. Markowitch. An efficient strong designated verifier signature scheme. In *Proc. of the 6th Int. Conf. on Information Security and Cryptology, ICISC 2003*, pages 40–54, Seoul, Korea, 27-28 November 2003. Springer-Verlag, Berlin.
24. N. Sendrier. Cryptosystmes cl publique bass sur les codes correcteurs derreurs. In *Habilitation diriger les recherches, Universit Pierre et Marie Curie*, Paris, France (in French), March 2002.
25. M. Koochak Shooshtari, M. Ahmadian-Attari, and M. R. Aref. Provably secure strong designated verifier signature scheme based on coding theory. *International Journal of Communication Systems*, doi: 10.1002/dac.3162.(?):??–??, 2016.
26. P.W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proc. of the 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, Santa Fe, New Mexico, USA, 20-22 November 1994. IEEE.
27. P. Gaborit A. Otmani T. P. Berger, P. L. Cayrel. Reducing key length of the McEliece cryptosystem. In *Proc. of the 2nd Int. Conf. on Cryptology in Africa, Progress in Cryptology AFRICACRYPT 2009*, pages 77–97, Gammarth, Tunisia, 21-25 June 2009. Springer-Verlag, Berlin.
28. D.S. Wong, K. Fung, J. K. Liu, and V.K. Wei. On the RS-code construction of ring signature schemes and a threshold setting of RST. In *Proc. of the 5th Int. Conf. on Information and Communications Security- ICICS 2003*, pages 34–36, Huhehaote, China, 10-13 October 2003. Springer-Verlag, Berlin.
29. D. Zheng, X. Li, and K. Chen. Code-based ring signature scheme. *International Journal of Network Security*, 5(2):154–157, 2007.