Strong Designated Verifier Signature Based on the Rank Metric

Hafsa Assidi, El Mamoun Souidi

Strong Designated Verifier Signature Based on the Rank Metric

Hafsa Assidi and El Mamoun Souidi

Mohammed V University in Rabat, Faculty of Sciences,
Laboratory of Mathematics, Computer Science, Applications and Information Security,
BP 1014 RP, Rabat 10000, Morocco
{assidihafsa,emsouidi}@gmail.com

Abstract. Strong designated verifier signatures (SDVS) allows users to produce signatures that are not publicly verifiable, such that no one other than the signer and the designated verifier can check the validity of a given signature, which preserves the privacy of the signer. This cryptographic primitive is very useful in different real life scenarios such as e-voting and e-bidding. In this paper, we propose a strong designated verifier signature scheme (SDVS) based on rank metric error correcting codes. Our construction makes a trade-off between efficiency and security requirements, for instance we achieve a signature of size 3510 bits and a public key of size equal to 23088 bits for the 80 security level. Furthermore, our proposal is quantum computer resistant since it is based on coding theory.

Keywords: Strong Designated Verifier Signature, Digital signature, Code-based Cryptography, LRPC Codes, Rank Metric, Post-quantum.

1 Introduction

A classical digital signature scheme is publicly verifiable where everyone can check the validity of a given signature. In some applications, such as e-voting and e-bidding, the signer wants to prove the validity of his signature to a specific user but not for others. As a consequence, the public verifiability of the signature is considered as an undesired feature.

To overcome this problem, Chaum and Antwerpen proposed the concept of undeniable signature [8] where the signer has a control over his signatures. In such a signature scheme, the verification is done in an interactive way between the signer and the verifier. However, the signer is able to decide when to prove but not whom verifying. Thereafter, Jakobsson *et al.* in [13] introduced the designated verifier signature (DVS). In a DVS scheme, the designated verifier is convinced by the validity of a signature but cannot transfer this conviction to others. Due to the non-transferability of DVS, the signature generated by the signer himself is indistinguishable from one simulated by the designated verifier. Jacobsson *et al.* [13] proposed a variant of DVS called strong designated verifier signature (SDVS). SDVS differs from DVS in the fact that the private key of a designated verifier is involved in the verification process and consequently there is no requirement for a third part to prove the validity of the designated verifier signature. The first formalisation of SDVS was presented by Saeednia *et al.* in [21] where an efficient construction of SDVS based on discrete logarithm problem is proposed. The authors in [21] introduced also the

notion of signer ambiguity where it is infeasible to guess if a signature is produced by the signer or simulated by the designated verifier. Thereafter, the work in [21] was extended by Laguillaumie *et al.* in [16] using a number theory construction of SDVS.
Later, many proposals of SDVS have been presented and are based on bilinear pairing like [11,15,14,17], these schemes are identity based strong designated verifier signature where the private keys of both the signer and the designated verifier are generated trough a key generator center. In 2013 Yang *et al.* [25] proposed a novel construction of SDVS with secure disavowability. In addition, Tian *et al.* [24] presented a systematic method to design strong designated verifier signature but without random oracles. The schemes in [25,24] are both based on computational Diffie-Hellman Problem. A recent work by Hu *et al.* in [12] has consisted of an SDVS scheme that supports the undeniability property besides the classical security requirements. In 2018, Lin *et al.* [18] proposed a new certificateless strong designated verifier signature scheme that is non-delegatable and verifies SSA-KCA security. Furthermore, Pereira de Almeida *et al.* presented in [1] a novel Dos defense mechanism based on strong designated verifier signatures.

The first provably secure code based SDVS was presented by Koochak Shooshtari *et al.* [22] presented . Then Rajabzadeh Asaar *et al.* proved in [2] that the scheme in [22] presents some weakness in the sense that it does not verify the signer ambiguity or non-transferability that is the main feature of strong designated verifier signatures. The authors of [2] showed also that the scheme in [22] is not strongly unforgeable if it does not preserve the non-transferability and they proposed in [2] a novel construction to overcome the aforementioned weakness. In the literature, we recognise also some constructions of SDVS that are derived from lattice assumptions such as [20].
Given that the number theoretic cryptography will not resist to the quantum computer as shown by Shor in his paper [23], the research for alternative solutions is very active. Code based cryptography, lattice based cryptography, multivariate cryptography and isogeny based cryptography are considered as an attractive and prominent alternative to classical cryptography in the era of quantum computers. In the literature, many cryptographic primitives are derived from coding theory assumptions such as group signature [3,5], ring and threshold ring signature [19,7] and also for authentication in RFID systems such as [4] where the authors proposed two mutual Zero-Knowledge authentication protocols based on error correcting code assumptions.

In the present paper, we propose a code based strong designated verifier signature using the LRPC codes. We use the signature scheme of Gaborit *et al.* [9] as the cryptographic primitive in our SDVS scheme. Our construction is resistant to quantum computer and fulfils the security requirements of an SDVS scheme. Namely, the correctness, the unforgeability, the non-transferability and the privacy of signer's identity. The practical results show that our proposal is practical, for instance, we achieve a signature of size equal to 3510 bits and a public key of size equal to 23088 bits for an 80 bits security level.
The organisation of the current paper is as follows: In Section 2, we recall some definitions from error correcting codes in rank metric, we recall also the algorithms that define a strong designated verifier signature and we give formal definitions of the security requirements. In Section 3, we present our proposed strong designated verifier signature. Section 4 is devoted to the security analysis of our SDVS proposal. in Section 5, we analyse the performance of the proposed SDVS scheme in terms of public key and signature sizes. We also analyse the

results and we make a comparison with some recent related works. We conclude in Section 6.

2 Backgrounds and Definitions

In this section, we define some general notations and we recall definitions related to error correcting codes with rank metric as well as the hard problems on codes that are used for cryptographic constructions.

- By $a \xleftarrow{\$} A$ we note an element a chosen uniformly at random from the set A.
- $Adv^{C}_{B,A}$: the advantage (the probability) that an adversary A breaks the property C of the scheme B.
- $Exp^{C}_{B,\mathcal{A}}$: is the experiment (the game) that describes how an adversary $\mathcal{A}$ can break the security property C of the scheme B.
- $a|b$: refers to the concatenation of two matrices or vectors a and b.
- ε: a value that is considered as negligible.
- x^T: refers to the transpose of the vector x.

2.1 Error Correcting Codes in Hamming Metric

Linear codes Let $GF(q)$ be the finite field of $q = p^s$ elements (p prime, and $s > 0$), n and k be non-negative integers with $k \leq n$. A linear code C of length n and dimension k over $GF(q)$ is a subspace of dimension k of the full space $GF(q)^n$.

A linear $[n, k]$ code can be defined either by its generator matrix or parity check matrix defined as follows:

Let C be an $[n, k]$ linear code over $GF(q)$. A matrix $G \in \mathcal{M}_{k,n}(GF(q))$ is a generator matrix of C if its rows form a basis of C. That is to say $C = \{mG, m \in GF(q)^k\}$.

A parity-check matrix $H \in \mathcal{M}_{n-k,n}(F_q)$ of a linear $[n, k]$-code C is defined as: $C = \{Hc^T = 0 | c \in GF(q)^n\}$.

Let H be a parity check matrix of an $[n, k]$ code C on $GF(q)$ and y belonging to $GF(q)^n$. The syndrome $s \in GF(q)^{n-k}$ of y associated to C is given by $s^T = Hy^T$ (Where w is an integer that represent a small Hamming weight).

Code-based cryptography relies on the assumption of the hardness of syndrome decoding problem, this problem is proved to be NP-complete by Berlekamp in [6]:

Problem 1 (Syndrome Decoding problem (SD)). The $SD(n, k, \omega)$ problem is formulated as follows: let n, k and ω be integers, given uniformly a random matrix $H \in \mathcal{M}_{k\times n}(GF(2))$ and an uniformly random syndrome $y \in GF(2)^k$, find a vector $s \in GF(2)^n$ such that $wt(s) \leq \omega$ and $H \cdot s^\top = y^\top$.

2.2 Error Correcting Codes in the Rank Metric

Let q be a power of a prime p and $GF(q)$ be the finite field with q elements. For an integer m, we define $GF(q^m)$ as a finite field of cardinality q^m. We consider $GF(q^m)$ as an m-dimensional vector space over $GF(q)$ and we denote by $\beta = (\beta_1, \cdots, \beta_m)$ an arbitrary basis of $GF(q^m)$ over $GF(q)$. Let $x = (x_1, \cdots, x_n) \in GF(q^m)^n$ where each x_j can be decomposed in the basis β as $x_j = \sum_{i=1}^{m} a_{ij}\beta_i$. We associate to the vector x the $m \times n$ matrix $A(x) = (a_{ij})_{1 \leq i \leq m, 1 \leq j \leq n}$. We denote by $rank(x)$ the rank weight of x which is the rank of the associated matrix $A(x)$.

$$rank(x) = RankA(x)$$

We define the distance between two vectors $(x, y) \in (GF(q^m))^2$ by

$$d_r(x, y) = rank(x - y)$$

A rank code C of length n and dimension k over $GF(q^m)$ is a subspace of dimension k of $GF(q^m)^n$ with the rank metric d_r.
By analogy to Hamming metric, the minimum rank distance of a code C can be defined as the minimum rank of non-zero vectors of the code C.

Definition 1. *Let $x = (x_1, \cdots, x_n) \in GF(q^m)^n$ be a vector of rank r. We denote by $E =< x_1, x_2, \cdots, x_n >$ the $GF(q)$-linear subspace of $GF(q^m)$ generated by $x_1, x_2, \cdots, x_n$. The vector space E is called the support of x.*

Definition 2. *Let e be an error vector of rank r and error support E. We denote by the errasure of dimension t of an error e a subspace T of dimension t of its error support E.*

Low Rank Parity Check (LRPC) codes

Definition 3 ([9]). *A Low Rank Parity Check (LRPC) code of rank d, length n and dimension k over $GF(q^m)$ is a code defined by an $(n-k) \times n$ parity check matrix $H = (h_{ij})$ (where $1 \leq i \leq n-k, 1 \leq j \leq n$), such that all its coordinates h_{ij} belong to the same $GF(q)$-subspace F of dimension d of $GF(q^m)$.*

The definition of syndrome decoding problem for the Hamming metric is extended to the rank metric which gives rise to the following hard problems [9].

Problem 2 (Approximate - Rank Syndrome Decoding problem). Let H be an $(n-k) \times n$ matrix over $GF(q^m)$ with $k \leq n$, $s \in GF(q^m)^{n-k}$ and $r \in \mathbb{N}^*$. The problem is to find $x \in GF(q^m)^n$ such that $Hx^t = s$ and $rank(x) = r$.

The Approximate RSD problem has been proven to be hard in [10] by Gaborit *et al.* where the proof is based on probabilistic reduction.

Problem 3 (Approximate - Rank Syndrome Decoding problem for augmented LRPC codes). Given a masked parity-check matrix $H' = A(R|H)P$ of an augmented LRPC codes and a random syndrome s, find a vector x of rank d such that $H'x^T = s$ (where P and A are invertible matrices in $GF(q)$ and $GF(q^m)$ respectively. The matrix R is chosen randomly in $GF(q^m)$ and H is a parity check matrix of an LRPC code).

The Approximate RSD for LRPC codes is a particular case of the Approximate RSD. The problem on itself is not proved to be hard, however it is difficult to resolve such problem under the assumption that is difficult to distinguish between the augmented LRPC codes and random codes [9]. On one hand, it is obvious that the family augmented LRPC codes is not a family of random codes, but they are weakly structured codes: the main point being that they have a parity-check matrix one part of which consists only in low rank coordinates the other part consisting in random entries. The attacker never has direct access to the LRPC matrix H, which is hidden by the augmented part. On the other hand, the minimum weight of augmented LRPC codes is smaller than the Gilbert Varshamov bound, hence natural attacks consist in trying to use their special structure to attack them. There exist general attacks for recovering the minimum weight of a code but these attacks have a fast increasing complexity especially when the size of the base field $GF(q)$ increases. More details on this attacks are discussed in [9].

2.3 Strong Designated Verifier Signature

In this subsection, we recall from [2] the building blocks that compose an SDVS scheme with their corresponding security properties.

Definition 4 (Strong Designated Verifier Signature). *A Strong Designated Verifier Signature is a sequence of five algorithms* $SDVS = (Setup, KeyGen, Sign, Verify, Sim)$ *such that:*

- $Setup(1^\lambda)$ *is a probabilistic algorithm that takes as input a security parameter* λ *and outputs* U *the set of users,* M *the message space and the public parameters of the scheme* pp.
- $KeyGen(pp)$ *is a deterministic algorithm that outputs* (sk_s, pk_s) *a secret and a public key of the signer* $\mathcal{S}$ *and* (sk_v, pk_v) *a private and a public key of the designated verifier* $\mathcal{V}$.
- $Sign(pp, sk_s, pk_s, pk_v, M)$ *is a probabilistic algorithm. Given the public parameters* pp*, the secret and public keys of the sign* (sk_s, pk_s)*, the public key of the designated verifier* pk_v *and the message* M*, this algorithm outputs a signature* $\sigma = Sign(pp, sk_s, pk_s, pk_v, M)$.
- $Verify(pp, pk_s, sk_v, pk_v, M, \sigma)$ *is a deterministic algorithm. It takes as input the public parameters* pp*, the secret key of the signer* sk_s *the secret and public key of the designated verifier* (sk_v, pk_v)*, the message* M *and a signature* σ *on* M*. It outputs* $b = 1$ *if* σ *is a valid signature on* M *and* $b = 0$ *otherwise.*
- $Sim(pp, sk_v, pk_v, pk_s, M)$*: is a probabilistic algorithm. Given the public parameters* pp*, the secret and public keys of the signer* (sk_v, pk_v)*, the public key of the designated verifier* pk_s *and the message* M*, this algorithm outputs a signature* $\sigma = Sign(pp, sk_v, pk_v, pk_s, M)$ *indistinguishable from one produced by the* $Sign$ *algorithm.*

2.4 Security model of SDVS

There are four security requirements that must be verified by a strong designated verifier signature namely: correctness, unforgeability, non-transferability and privacy of signer's identity [2].

Correctness: An SDVS is correct if for every valid secret key and public key of the signer and the designated verifier (sk_s, pk_s), (sk_v, pk_v) generated by $KeyGen$ algorithm and every message M we have:

$$Verify(pp, pk_s, sk_v, pk_v, M, \sigma) = 1$$

and

$$Verify(pp, pk_s, sk_v, pk_v, M, \sigma') = 1$$

Where $\sigma = Sign(pp, sk_s, pk_s, pk_v, M)$ and $\sigma' = Sim(pp, sk_v, pk_v, pk_s, M)$.

Unforgeability (UF): means that it is infeasible for an adversary $\mathcal{A}$ to produce a valid strong designated verifier signature without possessing the signer secret key sk_s or the designated verifier secret key sk_v. We consider the experiment between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ as described in Algorithm 1. An SDVS scheme is existentially unforgeable if the advantage $Adv^{UF}_{SDVS,\mathcal{A}}(\lambda)$ (which is the probability that the adversary $\mathcal{A}$ breaks the existential unforgeablility for the SDVS scheme) of the experiment in Algorithm 1 is negligible. A formal description of the unforgeability is given as it follows.

Algorithm 1 Unforgeability: Experiment $Exp^{UF}_{SDVS,\mathcal{A}}(\lambda)$

1. A challenger $\mathcal{C}$ runs the *Setup* algorithm to get the public parameters pp, runs the $KeyGen$ algorithm to get the signer's key pair (sk_s, pk_s) and the verifier's key pair (sk_v, pk_v). The triple (pp, pk_s, pk_v) are given to $\mathcal{A}$.
2. An adversary $\mathcal{A}$ is given access to the following oracles:
   - $\mathcal{O}_{sign}$: This oracle uses sk_s to produce a signature σ on a given message M, that is valid with regard to pk_s, pk_v and sends it to $\mathcal{A}$.
   - $\mathcal{O}_{sim}$: This oracle uses sk_v to produce a signature σ on a given message M, that is valid with regard to pk_s, pk_v and sends it to $\mathcal{A}$.
   - $\mathcal{O}_{ver}$: This oracle takes (m, σ) as a query and gives a bit that is 1 when σ is valid with regard to pk_s and pk_v, and 0 otherwise.
3. The adversary $\mathcal{A}$ returns a forged signature σ^* on a message M^* where the two conditions hold:
   - a) $Verify(pp, pk_s, sk_v, pk_v, M^*, \sigma^*) = 1$ and
   - b) The adversary $\mathcal{A}$ did not query the sign oracle $\mathcal{O}_{sign}$ and the sim oracle $\mathcal{O}_{sim}$ on the message M^*.

$$Adv^{UF}_{SDVS,\mathcal{A}}(\lambda) = Pr[a \text{ and } b \text{ occur}]$$

Definition 5. *An SDVS scheme is unforgeable if adversary $\mathcal{A}$ with at most q_v queries to $\mathcal{O}_{ver}$, q_s queries to $\mathcal{O}_{sign}$, q_{sim} to $\mathcal{O}_{sim}$ and q_{ro} random oracle queries has negligible success probability, that is, $Adv^{UF}_{SDVS,\mathcal{A}}(\lambda) \leq \varepsilon$ (Where ε is negligible).*

Non-Transferability: it means that the signature σ_0 generated by a signer $\mathcal{S}$ is indistinguishable from σ_1 the signature simulated by the designated verifier $\mathcal{V}$. We present forward a formal definition of non-transferability.

Definition 6. *An SDVS scheme is non-transferable if for all (sk_s, pk_s), (sk_v, pk_v), distinguisher $\mathcal{A}$ and message M we have:*

$$\left| pr\left[b' = b\right] - \frac{1}{2} \right| < \varepsilon$$

where $\sigma_0 = Sign(pp, sk_s, pk_s, pk_v, M)$, $\sigma_1 = Sim(pp, sk_v, pk_v, pk_s, M)$, $b \in \{0,1\}$, $b' = \mathcal{A}(pk_s, pk_v, sk_s, sk_v, \sigma_b)$ *and* ε *is negligible.*

Privacy of Signer's Identity (PSI): An SDVS scheme preserves the privacy of signer identity (PSI) if it is infeasible for an adversary $\mathcal{A}$ to guess the signer behind a given signature in the case when we have two or more potential signers. Formally, we define this property between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ as in the experiment bellow in Algorithm 2.

Algorithm 2 Privacy of Signer Identity $Exp^{PSI}_{SDVS,\mathcal{A}}(\lambda)$

1. A challenger $\mathcal{C}$ runs the *Setup* algorithm to get a public parameters pp, runs the $KeyGen$ algorithm to get tow signer's key pair (sk_{s0}, pk_{s0}), (sk_{s1}, pk_{s1}) and the verifier's key pair (sk_v, pk_v). Then $(pp, pk_{s0}, pk_{s1}, pk_v)$ are given to an adversary $\mathcal{A}$.
2. The adversary $\mathcal{A}$ is given access to the same oracles as in the unforgeability game (Algorithm 1).
3. The challenger $\mathcal{C}$ chooses randomly $b \in \{0,1\}$ and returns a signature $\sigma_b = Sign(pp, sk_{sb}, pk_{sb}, pk_v, M)$ to $\mathcal{A}$ where M is the signed message.
4. The adversary $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$ and wins the experiment if $b = b'$ and $\mathcal{A}$ has not made $\mathcal{O}_{ver}$ query on input (b, σ_b, pk_v, M).

Definition 7. *An SDVS scheme preserves the privacy of signer's identity if the advantage of an adversary* $\mathcal{A}$ *to win the experiment in Algorithm 2 is negligible* i.e.

$$Adv^{PSI}_{SDVS,\mathcal{A}}(\lambda) = \left| Pr[b = b'] - \frac{1}{2} \right| \leq \varepsilon$$

where ε *is negligible.*

3 The Proposed Strong Designated Verifier Signature

In this section, we present our proposal according to SDVS scheme based on error correcting codes. We explain in details the components used in our construction namely Algorithms 3, 4, 5, 6 and 7.

Setup: Given the security parameter λ, this algorithm outputs the public parameters pp, the secret and public keys of signer and designated verifier respectively (sk_s, pk_s) and (sk_v, pk_v) as described in Algorithm 3.

Algorithm 3 $Setup(1^\lambda)$

Input: a security parameter λ
Output: a public parameters pp

– The public parameters are $pp = \{n, k, m, q, f, \Psi, h\}$ where $h : \{0,1\}^* \rightarrow \{0,1\}^{n-k}$, $f : GF(q^m)^* \rightarrow GF(q^m)^{n-k}$ are random oracles, $\Psi_{h(M),i} : GF(q^m)^{n-k} \rightarrow GF(q^m)^{n-k}$ is a random permutation with keys $h(M)$ for $i \in \{signer, verifier\}$ and $(n, k, m, q,)$ are the parameters of the code.

KeyGen: Given the security parameter λ and public parameters pp, it outputs the secret and public keys of signer and designated verifier respectively (sk_s, pk_s) and (sk_v, pk_v) as described in Algorithm 4.

Algorithm 4 $KeyGen(1^\lambda, pp)$

Input: λ a security parameter, pp the public parameters produced by Algorithm 3
Output: (sk_s, pk_s) and (sk_v, pk_v)

- The public key of the signer is $pk_s = H'_s = (A_s(R_s|H_s)P_s, l_s)$ where P_s is an $(n+t) \times (n+t)$ invertible matrix in $GF(q)$, A_s is $(n-k) \times (n-k)$ invertible matrix in $GF(q^m)$, R_s is an $(n-k) \times t$ random matrix in $GF(q^m)$, H_s is a parity check matrix of an LRPC code and l_s is an integer.
- The secret key of the user is $sk_s = ((R_s|H_s), P_s, A_s)$.
- The designated verifier public key is $pk_v = H'_v = (A_v(R_v|H_v)P_v, l_v)$ where P_v is an $(n+t) \times (n+t)$ invertible matrix in $GF(q)$, A_v is an $(n-k) \times (n-k)$ invertible matrix in $GF(q^m)$, R_v is an $(n-k) \times t$ random matrix in $GF(q^m)$, l_v is an integer and H_v is a parity check matrix of an LRPC code.
- The secret key of the user is $sk_v = ((R_v|H_v), P_v, A_v)$.

Sign: The signature algorithm takes as input the public parameters pp, the secret and public keys of the signer (sk_s, pk_s), the verifier's public key pk_v generated in the Setup and the KeyGen Algorithms and a message M. The signature process is explained in Algorithm 5. We denote by Ω the general errors/erasures decoding algorithm for LRPC codes used in [9].

Algorithm 5 $Sign(pp, sk_s, pk_s, pk_v, M)$

Input: pp, sk_s, pk_s, pk_v, M
Output: σ

1. The signer chooses $\alpha \xleftarrow{\$} GF(q^m)^{n+t}$ and pick t random elements $(e_1, \cdots, e_t)$ in $GF(q^m)$ and $x_v \xleftarrow{\$} GF(q^m)^{n+t}$ such that $rank(\alpha) \leq d$, $rank(x_v) \leq d$ and computes $y = H'_v \alpha^T$.
2. The signer computes $s = \Psi^{-1}_{h(M),s}(f(\alpha, y, H'_s, H'_v, M) - \Psi_{h(M),v}(H'_v x_v^T))$.
3. Decodes by the LRPC matrix H_s the syndrome $s' = A_s^{-1} s^T - R_s(e_1, \cdots, e_t)^T$ with erasure space $T =< e_1, \cdots, e_t >$ and r' errors by the decoding algorithm Ω.
4. If the decoding algorithm returns a word $(e_{t+1}, \cdots, e_{n+t})$ of weight $r = t + r'$, the signature is $\sigma = [x_s = (e_1, \cdots, e_{n+t})(P_s^T)^{-1}, x_v, y]$ else return to step 1.

Verify: The verification step consists of checking the validity of a given signature, it returns $True$ if the verification succeed and $False$ otherwise.

Algorithm 6 $Verify(pp, pk_s, sk_v, pk_v, M, \sigma)$

Input: $pp, pk_s, sk_v, pk_v, M, \sigma$
Output: $True$ or $False$

- The designated verifier receives a signature $\sigma = [x_s = (e_1, \cdots, e_{n+t})(P_s^T)^{-1}, x_v, y]$.
- He/she uses the decoding algorithm Ω to recover α from y using $(R_v|H_v), P_v, A_v$.
- He/she computes $a = f(\alpha, y, H'_s, H'_v, M)$,

if $\Psi_{h(M),s}(H'_s x_s^T) + \Psi_{h(M),v}(H'_v x_v^T) = a$, $rank(e) = r = t + r'$, $rank(\alpha) \leq d$ and $rank(x_v) \leq d$ **then**
  return $True$
else
  return $False$
end if

Sim: The simulation algorithm takes as input the public parameters pp, the secret and public keys of the designated verifier (sk_v, pk_v), the signer's public key pk_s generated in the Setup Algorithm 3 and a message M. The simulated signature is explained in Algorithm 7.

Algorithm 7 $Sim(pp, sk_v, pk_v, pk_s, M)$

Input: pp, sk_v, pk_v, pk_s, M
Output: σ'.
To simulate a signature on the message M, the designated verifier proceeds as follow:

1. Chooses randomly $\alpha' \xleftarrow{\$} GF(q^m)^{n+t}$ and pick t random elements $(e'_1, \cdots, e'_t)$ of $GF(q^m)$ and $x'_s \xleftarrow{\$} GF(q^m)^{n+t}$ such that $rank(\alpha') \leq d$, $rank(x'_s) \leq d$ and computes $y' = H'_v \alpha'^T$.
2. The designated verifier computes

$$s = \Psi^{-1}_{h(M),v}(f(\alpha', y', H'_s, H'_v, M) - \Psi_{h(M),s}(H'_s x_s'^T))$$

   .
3. The designated verifier, decodes by the LRPC matrix H_v the syndrome $s' = A_v^{-1} s^T - R_v(e'_1, \cdots, e'_t)^T$ with errasure space $T =< e'_1, \cdots, e'_t >$ and r' errors by the decoding algorithm Ω.
4. If the decoding algorithm returns a word $(e'_{t+1}, \cdots, e'_{n+t})$ of weight $r = t + r'$, the signature is $\sigma' = [x'_s, x'_v = (e'_1, \cdots, e'_{n+t})(P_v^T)^{-1}, y']$ else return to step 1.

4 Security analysis

In this section, we analyse the security properties using the Random Oracle model of the proposed strong designated verifier signature scheme by proving, respectively, the correctness, the unforgeability, the non-transferability and the privacy of signer's identity.
Correctness: Let $\sigma = Sign(pp, sk_s, pk_s, pk_v, M)$ be a signature generated by the signer on a message M and $\sigma' = Sim(pp, sk_v, pk_v, pk_s, M)$ be a simulated signature produced by the designated verifier, the scheme is correct if

$$Verify(pp, pk_s, sk_v, pk_v, M, \sigma) = 1$$

and

$$Verify(pp, pk_s, sk_v, pk_v, M, \sigma') = 1$$

We prove only the correctness for signature generated by the signer because the proof for simulated signature is similar.
Let $\sigma = [x_s = (e_1, \cdots, e_{n+t})(P_s^T)^{-1}, x_v, y]$ be a signature generated as described in Algorithm 5, we have to prove the following: $\Psi_{h(M),s}(H'_s x_s^T) + \Psi_{h(M),v}(H'_v x_v^T) = a$, $rank(e) = r = t + r'$, $rank(\alpha) \leq d$ and $rank(x_v) \leq d$ where $a = f(\alpha, y, H'_s, H'_v, M)$ (Ψ, H'_s and H'_v are defined in Algorithm 3 and Algorithm 4).
We have on one hand:

$$\begin{aligned} H'_s x_s^T &= H'_s (P_s^T)^{-1T} (e_1, \cdots, e_{n+t})^T \\ &= A_s (R_s | H_s) P_s (P_s^T)^{-1T} (e_1, \cdots, e_{n+t})^T \\ &= A_s (R_s | H_s) (e_1, \cdots, e_{n+t})^T \\ &= s \end{aligned}$$

And on the other hand:

$$\begin{aligned} \Psi_{h(M),s}(H'_s x_s^T) + \Psi_{h(M),v}(H'_v x_v^T) &= \Psi_{h(M),s}(s) + \Psi_{h(M),v}(H'_v x_v^T) \\ &= f(\alpha, y, H'_s, H'_v, M) - \Psi_{h(M),v}(H'_v x_v^T) + \\ &\quad \Psi_{h(M),v}(H'_v x_v^T) \\ &= f(\alpha, y, H'_s, H'_v, M) \\ &= a \end{aligned}$$

The verifier can check easily that $rank(e) = r = t + r'$, $rank(\alpha) \leq d$ and $rank(x_v) \leq d$.

Unforgeability: We recall that this property means that it is infeasible for an adversary $\mathcal{A}$ to produce a valid SDVS without the knowledge of the signer secret key sk_s or the designated verifier secret key sk_v.

Theorem 1. *If there is an adversary $\mathcal{A}$ against the unforgeability of the scheme with non negligible probability then, there exists an adversary $\mathcal{C}$ that can solve the problem 3 with non negligible probability.*

Proof. We assume that there exists an adversary $\mathcal{A}$ who can produce a forged signature with success probability at most ε_1. Let $\mathcal{C}$ be an adversary who can solve an instance of the Problem 3 with probability equal to ε_2 *i.e.* the adversary $\mathcal{C}$ returns a vector x^* of rank less or equal to d such that $H'^* x^* = s^*$ where $H'^* = A^*(R^*|H^*)P^*$, $A^* \in GL_{n-k}(GF(q^m))$,

$P^* \in GL_{n+t}(GF(q))$, R^* is a random $(n-k) \times t$ matrix in $GF(q^m)$ and H^* is a parity check matrix of an LRPC code. The challenger $\mathcal{C}$ runs the $Setup$ algorithm to get the public parameters, runs the $KeyGen$ algorithm to get signer's and designated verifier's key pair (sk_s, pk_s) and (sk_v, pk_v) respectively. The adversary $\mathcal{C}$ provides $\mathcal{A}$ with public parameters pp, signer's public key $pk_s = H'_s$ and designated verifier public key $pk_v = H'_v$. The adversary $\mathcal{A}$ asks q_f, q_Ψ, q_{sign}, q_{sim} and q_v queries for the following oracles $f(.)$, $\Psi(.)$, $\mathcal{O}_{sign}$, $\mathcal{O}_{sim}$ and $\mathcal{O}_{ver}$ respectively.

- $f(.)$ queries: if $T_f[.]$ is defined for query $(\alpha, y, H'_s, H'_v, M)$, then $\mathcal{C}$ returns its value else, $\mathcal{C}$ returns a random value $T_f[\alpha, y, H'_s, H'_v, , M] \xleftarrow{\$} GF(q^m)^{n-k}$.
- $\Psi(.)$ or $\Psi^{-1}(.)$ queries: for a query under the form $\Psi^{-1}_{h(M),i} = H'_i x_i^T$ or under the form $\Psi_{h(M),i} = (f(\alpha, y, H'_s, H'_v, M) - \Psi_{h(M),\bar{i}}(H'_{\bar{i}} x_{\bar{i}}^T)$ (where $i \in \{signer, verifier\}$), the adversary $\mathcal{C}$ searches in $T_\Psi[.]$ and returns its value if it exists otherwise, it returns a random value from $GF(q^m)^{n-k}$ and send it to the adversary $\mathcal{A}$.
- $\mathcal{O}_{sign}$: for query (H'_s, H'_v, M), $\mathcal{C}$ chooses randomly $\alpha \xleftarrow{\$} GF(q^m)^{(n+t)}$, $x_v \xleftarrow{\$} GF(q^m)^{(n+t)}$ and $x_s \xleftarrow{\$} GF(q^m)^{n+t}$ such that $rank(\alpha) \leq d$ and $rank(x_v) \leq d$. The challenger $\mathcal{C}$ computes $y = H'_v \alpha^T$ and $a = \Psi^{-1}_{h(M),s}(H'_s x_s^T) + \Psi_{h(M),s}(H'_v x_v^T)$. If $T_f[\alpha, y, H'_s, H'_v, M]$ have been already defined, then $\mathcal{C}$ aborts, otherwise we make $T_f[\alpha, y, H'_s, H'_v, M] \leftarrow a$ and an SDVS signature on the message M under H'_s, H'_v is equal to $\sigma = (x_s, x_v, y)$. The adversary $\mathcal{C}$ sends σ to $\mathcal{A}$.
- $\mathcal{O}_{sim}$: for query (H'_v, H'_s, M), this oracle is programmed as the $\mathcal{O}_{sign}$ oracle.
- $\mathcal{O}_{ver}$ queries: for a query $(x_s, x_v, y, H'_v, H'_s, M)$, the challenger $\mathcal{C}$ searches in table $T_f[.]$ for the tuple $(\alpha, y, H'_v, H'_s, M)$ such that $y = H'_v \alpha^T$ and $rank(\alpha) \leq d$ and also in table $T_\Psi[.]$ for queries in form of $H'_s x_s^T$ and $H'_v x_v^T$ in order to have $\Psi_{h(M),s} = \Psi_{h(M),s}(H'_s x_s^T)$ and $\Psi_{h(M),v} = \Psi_{h(M),v}(H'_v x_v^T)$ and verifies if $\Psi_{h(M),s}(H'_s x_s^T) + \Psi_{h(M),v}(H'_v x_v^T) = f(\alpha, y, H'_v, H'_s, M)$ and $rank(x_x) \leq d$.
- Finally, the adversary $\mathcal{A}$ outputs a forged signature $\sigma^* = (x_s^*, x_v^*, y^*)$ on a message M^* under signer's and designated verifier's public keys pk_s, pk_v such that $Verify(pp, pk_s, pk_v, sk_v, \sigma^*, M^*) = 1$ and $\mathcal{A}$ has never questioned the Sign Algorithm for input (pk_v, pk_s, M^*).

The adversary $\mathcal{A}$ wins the unforgeability game with probability equal to

$$Pr[Event1] \times Pr[Event2|Event1]$$

where Event1 and Event2 are defined as follows:

- Event 1: The adversary $\mathcal{C}$ does not abort in Sign and Sim oracles.
- Event 2: The adversary $\mathcal{A}$ breaks the unforgeability of the scheme.

In order to compute the probability of the Event1, we distinguish two cases:

- Case 1: if $(\alpha y, H'_s, H'_v, M)$ produced in one Sign or Sim oracles has occurred by chance in a previous query to the oracle $f(.)$, then $bad \leftarrow True$ (The event bad refers to the adversary $\mathcal{C}$ aborts in $Sign$ and Sim algorithms). Given that there exist at most $(q_f + q_{sign} + q_{sim})$ entries in table $T_f[.]$ and the number of elements α chosen randomly in $GF(q^m)^{n+t}$ such that $rank(\alpha) \leq d$ is equal to ξ. As a consequence, the probability of this event for $(q_{sign} + q_{sim})$ queries is at most

$$\frac{(q_{sign} + q_{sim})(q_f + q_{sign} + q_{sim})}{\xi} \tag{1}$$

– Case 2: if the adversary $\mathcal{C}$ used the same random elements $\alpha \in GF(q^m)^n$ such that $rank(\alpha) \leq d$ in one $\mathcal{O}_{sign}$ or $\mathcal{O}_{sim}$ oracles, we have $bad = true$ and $\mathcal{C}$ makes at most $(q_{sign} + q_{sim})$ queries to Sign and Sim oracles. Therefore, the probability is at most $\frac{(q_{sign}+q_{sim})^2}{\xi}$ Consequently,

$$Pr[Event1] = 1 - Pr[bad] \geq 1 - \frac{(q_{sign} + q_{sim})(q_f + 2(q_{sign} + q_{sim}))}{\xi}$$

However, we have $Pr[Event2|Event1] \geq \varepsilon_1$.

As a consequence, the adversary $\mathcal{A}$ outputs a tuple $(x_s^*, , x_v^*, y^*, f, \Psi_{h(M),s}^{-1}, \Psi_{h(M),v})$ with probability at least

$$\varepsilon_1 - \frac{(q_{sign} + q_{sim})(q_f + 2(q_{sign} + q_{sim}))}{\xi}$$

The challenger $\mathcal{C}$ employs $\mathcal{A}$, guesses an index $1 \leq \gamma \leq q_\Psi$ and wishes that γ is the index of the query $\Psi_{h(M),s} = (f(\alpha^*, y^*, H_s', H_v', M) - \Psi_{h(M),v}(H_v' x_v^*)$ to the oracle $\Psi_{h(M),i}^{-1}$. The algorithm $\mathcal{C}$ outputs s^* as a response to this query with probability $\frac{1}{q_\Psi}$. The tuple $(x_s^*, , x_v^*, y^*, f, \Psi_{h(M),s}^{-1}, \Psi_{h(M),v})$ is a valid SDVS signature and as a consequence we have: $rank(x_s^*) \leq d$, $rank(x_v^*) \leq d$ and

$$H_s' x_s^{*T} = \Psi_{h(M^*),s}^*(\Psi_{h(M^*),v}(H_v x_v^{*T}) - f(\alpha^*, y^*, H_s^{'*}, H_v^{'*}))$$

We take $s^* = \Psi_{h(M^*),s}^*(\Psi_{h(M^*),v}(H_v x_v^{*T}) - f(\alpha^*, y^*, H_s^{'*}, H_v^{'*}))$ and then $\mathcal{C}$ solves the following instance of Problem 3 $H_s' x_s^{*T} = s^*$ with probability at least

$$\frac{\varepsilon_1}{q_\Psi} - \frac{(q_{sign} + q_{sim})(q_f + 2(q_{sign} + q_{sim}))}{\xi \cdot q_\Psi}$$

Thus, we conclude the proof. ■

Non-transferability Hereafter we discuss about the non transferability of the proposed SDVS scheme.

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

Proof. We keep the same notations as before, then, we have to prove that the signature produced by the signer and the one simulated by the designated verifier are indistinguishable. For this reason, we prove that the following distributions are the same

$$\sigma_{signer} = (x_s, x_v, y) : \begin{cases} \alpha \xleftarrow{\$} GF(q^m)^{n+t}, rank(\alpha) \leq d \\ x_v \xleftarrow{\$} GF(q^m)^{n+t}, rank(x_v) \leq d \\ y = H_v' \alpha^T \\ s_1 = \Psi_{h(M),s}^{-1}(f(\alpha, y, H_s', H_v', M) - \Psi_{h(M),v}(H_v' x_v^T)) \\ x_s = (e_1, \cdots, e_{n+t})(P_s^T)^{-1} \end{cases}$$

$$\sigma_{sim} = (x'_s, x'_v, y') : \begin{cases} \alpha' \xleftarrow{\$} GF(q^m)^{n+t}, rank(\alpha') \leq d \\ x'_s \xleftarrow{\$} GF(q^m)^{n+t}, rank(x'_v) \leq d \\ y' = H'_v \alpha'^T \\ s_2 = \Psi^{-1}_{h(M),s}(f(\alpha', y', H'_s, H'_v, M) - \Psi_{h(M),v}(H'_v x'^T_v)) \\ x'_v = (e'_1, \cdots, e'_{n+t})(P_v^T)^{-1} \end{cases}$$

We suppose that $\overline{\sigma}$ is a valid signature selected from the set of all the valid signature of the signer and we compute the following probabilities:

$$Pr_{\sigma_{signer}} = Pr_{\alpha, x_v}[\overline{\sigma} = \sigma_{signer}] = (Pr\{\alpha \xleftarrow{\$} GF(q^m)^{n+t}, rank(\alpha) \leq d\})^2$$

and

$$Pr_{\sigma_{sim}} = Pr_{\alpha', x'_v}[\overline{\sigma} = \sigma_{sim}] = (Pr\{\alpha' \xleftarrow{\$} GF(q^m)^{n+t}, rank(\alpha') \leq d\})^2$$

As a consequence, we have $Pr_{\sigma_{signer}} = Pr_{\sigma_{sim}}$ ■

Privacy of signer identity (PSI)

Theorem 3. *If there exist an adversary $\mathcal{A}$ who can break the PSI property of the scheme with non-negligible probability, then there exists an adversary $\mathcal{C}$ that can solve Problem 3 with non negligible probability.*

Proof. We suppose that there exists an adversary $\mathcal{A}$ who can break the PSI of the scheme with success probability at most ε_1. We consider $\mathcal{C}$ as an adversary who can solve an instance of Problem 3 with probability equal to ε_2 *i.e.* the adversary $\mathcal{C}$ returns a vector x^* of rank less or equal to d such that $H'^* x^* = s^*$ where $H'^* = A^*(R^*|H^*)P^*$, $A^* \in GL_{n-k}(GF(q^m))$, $P^* \in GL_{n+t}(GF(q))$ and R^* is a random $(n-k) \times t$ matrix in $GF(q^m)$. The adversary $\mathcal{C}$ runs the Setup algorithm in order to get the public parameters pp, runs the $KeyGen$ algorithm (Algorithm 4) to get the public keys of the two signers H'_{s_0} and H'_{s_1} with their corresponding secret keys. The adversary $\mathcal{C}$ sets the designated verifier's public key $H'_v = H'^*$ and the adversary $\mathcal{A}$ makes q_f query to the $f(.)$ oracle, q_Ψ query to $\Psi_{h(M),i}$ oracle, q_{sign} query to the sign oracle $\mathcal{O}_{sign}$, q_{sim} query to the sim oracle $\mathcal{O}_{sim}$ and q_v query to the verification oracle $\mathcal{O}_{ver}$. The oracles $f(.)$ and $\Psi_{h(M),i}(.)$ are programmed as in the proof of Theorem 1 [unforgeability], at the beginning we take empty tables $T_f[.]$, $T_\Psi[.]$ and $T_s[.]$ (in which we store the issued signatures). The oracle queries are as follows:

- Sign query $\mathcal{O}_{sign}$: for a query (b, H'_{s_b}, H_v, M) where $b \in \{0,1\}$, the adversary $\mathcal{C}$ returns a signature $\sigma_b = (x_{s_b}, x_v, y)$ on a message M by running the sign algorithm since $\mathcal{C}$ has the signer's secret key and then transfers σ_b to $\mathcal{A}$.
- Verify queries $\mathcal{O}_{ver}$: for a query $(b, \sigma_b, H'_v, H'_{s_b}, M)$ where $b \in \{0,1\}$. This oracle returns 1 if σ_b is in $T_s[.]$ and σ_b was never returned by $\mathcal{C}$ and 0 otherwise.
- The adversary $\mathcal{C}$ chooses randomly $b \in \{0,1\}$, $x_v \in GF(q^m)^{(n+t)}$ with $rank(x_v) \leq d$, puts $y = s^*$, makes query to the oracle $f(.)$ on the tuple $(T, s^*, H'_{s_b}, H'^*, M)$, computes x_s as in Algorithm 5 (Step 2, 3 and 4) and returns to the adversary $\mathcal{A}$ the signature σ_b, the public keys H'_{s_0}, H'_{s_1} and the designated verifier public key H'_v.
- After making a number of queries to the aforementioned oracles, the adversary changes the answers of the oracles adequately. In the case of queries of the form (x^*, y, H'_s, H'_v, M) where $y \neq s^*$, it returns a random value from $GF(q^m)^{n-k}$. If $y = s^*$ and $rank(x^*) \leq d$ it returns a random value from $GF(q^m)^{n-k}$ and changes T by x.

– The adversary $\mathcal{A}$ returns $b' = b$.

To succeed in the PSI attack, the adversary $\mathcal{A}$ has to make query to the $f(.)$ oracle on the tuple $(x^*, s^*, H'_{s_b}, H'^*, M)$ where $s^* = H'^* x^{*T}$. Since $f(.)$ is a random oracle, $\mathcal{A}$ can guess its value with probability $\frac{1}{(q^m)^{n-k}}$. In addition, the probability that $\Psi_{h(M),s_b} = \Psi_{h(M),s_b}(H'_{s_b} x^T_{s_b})$ and $\Psi_{h(M),v} = \Psi_{h(M),v}(H'_v x^T_v)$ is less than $\frac{2}{(q^m)^{n-k}}$. Consequently, x^* is a solution to the following instance $H'^* x^{*T} = s^*$ where $rank(x^*) \leq d$ of Problem 3 with probability ε_2 such that:

$$\varepsilon_2 \geq \varepsilon_1 - \frac{3}{(q^m)^{n-k}}$$

■

5 Parameters and results

In this section, we give parameters for the proposed strong designated verifier signature scheme in Table 1 for different security levels. We also compare our results with some related works in Table 2 and 3, in particular with the post-quantum constructions of SDVS namely [22] and [2]. The comparison is done in terms of security properties, public key and signature sizes.

– The signer's and designated verifier's public key size:

$$\begin{aligned} size_{(pk_s)} &= size_{(pk_v)} \\ &= (n-k)(n+t)mlog_2(q) - bits \end{aligned}$$

– The signature size is computed as follows:

$$\begin{aligned} size_{(sig)} &= size(x_s) + size(x_v) + size(y) \\ &= [2(n+t) + (n-k)]mlog_2(q) - bits \end{aligned}$$

Security level	n	k	t	m	q	Public key size (bits)	Signature size (bits)
80 bits	32	16	5	39	2	23088	3510
110 bit	40	20	5	45	2	40000	4950
120 bit	16	8	2	18	2^8	16000	6336
130 bit	16	8	2	18	2^{40}	96000	31680

Table 1. Parameters for different security levels.

6 Conclusion

In this paper, we have proposed an efficient strong designated verifier signature scheme from coding theory assumptions that is supposed to be resistant to quantum computers. Our approach relies on using rank metric codes rather than classical Hamming codes;

Scheme	Correctness	Non-transferability	Unforgeability	Privacy of signer identity
Scheme of [22]	Yes	No	No	Yes
Scheme of [2]	Yes	Yes	Yes	Yes
Our scheme	Yes	Yes	Yes	Yes

Table 2. Comparison in terms of security properties with some related works.

Scheme	Hard problem	Security model	Signature size(bit)	Public key size (bit)
Scheme of [22]	Syndrome Decoding	Random oracle	624	57.75
Scheme of [2]	Syndrome Decoding	Random oracle	530	99
Our scheme	Rank Syndrome Decoding	Random oracle	3510	0.003

Table 3. Comparison in terms of public key and signature sizes.

indeed we proposed to use LRPC codes. Our construction combines efficiency and security requirements, as we have achieved a reasonable size for the public key length and for the signature size. In addition, the security properties required for an SDVS scheme are fulfilled. As a perspective to the present paper, we consider to propose in a future work an SDVS scheme that combines between efficiency and security especially in the era of post quantum cryptography.

References

1. de Almeida, M.P., de Sousa Júnior, R.T., García-Villalba, L.J., Kim, T.: New dos defense method based on strong designated verifier signatures. Sensors **18**(9), 2813 (2018). https://doi.org/10.3390/s18092813
2. Asaar, M.R., Salmasizadeh, M., Aref, M.R.: Code-based strong designated verifier signatures: Security analysis and a new construction. IACR Cryptology ePrint Archive **2016**, 779 (2016)
3. Assidi, H., Ayebie, E.B., Souidi, E.M.: A code-based group signature scheme with shorter public key length. In: Proceedings of the 13th International Joint Conference on e-Business and Telecommunications (ICETE 2016) - Volume 4: SECRYPT, Lisbon, Portugal, July 26-28, 2016. pp. 432–439 (2016). https://doi.org/10.5220/0005969204320439
4. Assidi, H., Ayebie, E.B., Souidi, E.M.: Two mutual authentication protocols based on zero-knowledge proofs for RFID systems. In: Information Security and Cryptology - ICISC 2017 - 20th International Conference, Seoul, South Korea, November 29 - December 1, 2017, Revised Selected Papers. pp. 267–283 (2017). https://doi.org/10.1007/978-3-319-78556-1_15
5. Ayebie, B.E., Assidi, H., Souidi, E.M.: A new dynamic code-based group signature scheme. In: Codes, Cryptology and Information Security - Second International Conference, C2SI 2017, Rabat, Morocco, April 10-12, 2017, Proceedings - In Honor of Claude Carlet. pp. 346–364 (2017). https://doi.org/10.1007/978-3-319-55589-8_23
6. Berlekamp, E.R., McEliece, R.J., van Tilborg, H.C.A.: On the inherent intractability of certain coding problems (corresp.). IEEE Trans. Information Theory **24**(3), 384–386 (1978). https://doi.org/10.1109/TIT.1978.1055873
7. Cayrel, P., Alaoui, S.M.E.Y., Hoffmann, G., Véron, P.: An improved threshold ring signature scheme based on error correcting codes. In: Arithmetic of Finite Fields - 4th International Workshop, WAIFI 2012, Bochum, Germany, July 16-19, 2012. Proceedings. pp. 45–63 (2012)
8. Chaum, D., Antwerpen, H.V.: Undeniable signatures. In: Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings. pp. 212–216 (1989). https://doi.org/10.1007/0-387-34805-0_20

9. Gaborit, P., Ruatta, O., Schrek, J., Zémor, G.: Ranksign: An efficient signature algorithm based on the rank metric. In: Post-Quantum Cryptography - 6th International Workshop, PQCrypto 2014, Waterloo, ON, Canada, October 1-3, 2014. Proceedings. pp. 88–107 (2014). https://doi.org/10.1007/978-3-319-11659-4_6
10. Gaborit, P., Zémor, G.: On the hardness of the decoding and the minimum distance problems for rank codes. IEEE Trans. Information Theory **62**(12), 7245–7252 (2016). https://doi.org/10.1109/TIT.2016.2616127
11. Gorantla, M.C., Boyd, C., Nieto, J.M.G.: Strong designated verifier signature in a multi-user setting. In: Seventh Australasian Information Security Conference, AISC 2009, Wellington, New Zealand, January 2009. pp. 21–31 (2009)
12. Hu, X., Tan, W., Xu, H., Wang, J., Ma, C.: Strong designated verifier signature schemes with undeniable property and their applications. Security and Communication Networks **2017**, 7921782:1–7921782:9 (2017). https://doi.org/10.1155/2017/7921782
13. Jakobsson, M., Sako, K., Impagliazzo, R.: Designated verifier proofs and their applications. In: Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding. pp. 143–154 (1996). https://doi.org/10.1007/3-540-68339-9_13
14. Kancharla, P.K., Gummadidala, S., Saxena, A.: Identity based strong designated verifier signature scheme. Informatica, Lith. Acad. Sci. **18**(2), 239–252 (2007)
15. Kang, B., Boyd, C., Dawson, E.: A novel identity-based strong designated verifier signature scheme. Journal of Systems and Software **82**(2), 270–273 (2009). https://doi.org/10.1016/j.jss.2008.06.014
16. Laguillaumie, F., Vergnaud, D.: Designated verifier signatures: Anonymity and efficient construction from any bilinear map. In: Security in Communication Networks, 4th International Conference, SCN 2004, Amalfi, Italy, September 8-10, 2004, Revised Selected Papers. pp. 105–119 (2004). https://doi.org/10.1007/978-3-540-30598-9_8
17. Lal, S., Verma, V.: Identity based strong designated verifier proxy signature schemes. IACR Cryptology ePrint Archive **2006**, 394 (2006)
18. Lin, H.: A new certificateless strong designated verifier signature scheme: Non-delegatable and SSA-KCA secure. IEEE Access **6**, 50765–50775 (2018). https://doi.org/10.1109/ACCESS.2018.2809437
19. Melchor, C.A., Cayrel, P., Gaborit, P.: A new efficient threshold ring signature scheme based on coding theory. In: Post-Quantum Cryptography, Second International Workshop, PQCrypto 2008, Cincinnati, OH, USA, October 17-19, 2008, Proceedings. pp. 1–16 (2008). https://doi.org/10.1007/978-3-540-88403-3_1
20. Noh, G., Jeong, I.R.: Strong designated verifier signature scheme from lattices in the standard model. Security and Communication Networks **9**(18), 6202–6214 (2016). https://doi.org/10.1002/sec.1766
21. Saeednia, S., Kremer, S., Markowitch, O.: An efficient strong designated verifier signature scheme. In: Information Security and Cryptology - ICISC 2003, 6th International Conference, Seoul, Korea, November 27-28, 2003, Revised Papers. pp. 40–54 (2003). https://doi.org/10.1007/978-3-540-24691-6_4
22. Shooshtari, M.K., Ahmadian-Attari, M., Aref, M.R.: Provably secure strong designated verifier signature scheme based on coding theory. Int. J. Communication Systems **30**(7) (2017). https://doi.org/10.1002/dac.3162
23. Shor, P.W.: Algorithms for quantum computation: Discrete logarithms and factoring. In: 35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994. pp. 124–134 (1994). https://doi.org/10.1109/SFCS.1994.365700
24. Tian, H., Jiang, Z., Liu, Y., Wei, B.: A systematic method to design strong designated verifier signature without random oracles. Cluster Computing **16**(4), 817–827 (2013). https://doi.org/10.1007/s10586-013-0255-x
25. Yang, B., Yu, Y., Sun, Y.: A novel construction of SDVS with secure disavowability. Cluster Computing **16**(4), 807–815 (2013). https://doi.org/10.1007/s10586-013-0254-y