AIMS Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 8(8): 18234–18250.
DOI:10.3934/math.2023927
Received: 03 March 2023
Revised: 18 April 2023
Accepted: 26 April 2023
Published: 29 May 2023

Research article

A quantum resistant universal designated verifier signature proof

P. Thanalakshmi[1]**, N. Anbazhagan**[2]**, Gyanendra Prasad Joshi**[3] **and Eunmok Yang**[4,*]

[1] Department of Applied Mathematics and Computational Sciences, PSG College of Technology, Coimbatore 641004, India

[2] Department of Mathematics, Alagappa University, Karaikudi 630003, India

[3] Department of Computer Science and Engineering, Sejong University, Seoul 05006, Korea

[4] Department of Information Security, Cryptology and Mathematics, Kookmin University, Seoul 02707, Korea

* **Correspondence:** Email: emyang@kongju.ac.kr; Tel: +82029105571; Fax: +82029105571.

Abstract: In order to ensure that only the designated person can verify the signer's signature on the message, Steinfeld et al. introduced the concept of Universal Designated Verifier Signature (UDVS), which enables a designator who has obtained a signature on a message from the signer to designate the signature to any desired designated verifier. This idea was developed to address the privacy concerns of the signature holder at the time of certificate distribution. They are appropriate for applications that demand the designer's secrecy. The fact that the designated verifier must generate a public key with regard to the signer's public parameter for signature verification is a significant drawback of UDVS methods. In cases where the verifier is unable to begin the key generation procedure, this constraint is inapplicable. Baek et al. developed the idea of "Universal Designated Verifier Signature Proof (UDVSP)", which does not require the verifier's public key for verification, to get around this restriction. All existing UDVSP constructions are based on a discrete logarithm problem, which is vulnerable to quantum computer attacks. As a result, an efficient quantum resistant UDVSP is built on a hard problem in coding theory, as suggested by NIST reports. The scheme's security against forgeability and impersonation attacks is examined using the random oracle model.

Keywords: coding theory; random oracle model; syndrome decoding problem; universal designated verifier signature proof
Mathematics Subject Classification: 11T71, 94A62

1. Introduction

In user certification systems, the Certification Authority (CA) issues a signed certificate to the user. The certificate consists of attributes of the user and the truth of certain statements about the user. When the user wants to convince any interested verifier about the truth of the statements, the user presents the certificate to the verifier. The verifier accepts the truth of the statements by ensuring CA's authentication in it. Systems that issue birth and death certificates, driving licenses, academic transcripts, etc. are examples of user certification systems. When CA employs an ordinary signature to authenticate a certificate, anyone who receives the certificate can easily verify the validity of the signature. Also, it is easy in the current electronic world to copy and distribute the certificate to an unlimited number of users and convince them that the information contained in it is true. Once the owner of the certificate sends out his/her certificate to convince a verifier, then he/she has no control any longer over the number of users who receive and learn the information in it. This poses a serious threat to the certificate owner's privacy.

Motivated by these privacy issues in certificate distribution systems, in order to convince any designated verifier that he/she is in possession of a valid signature of the signer without disclosing the actual signature, a designator who receives a valid signature on a message from a signer converts it into a Universal Designated Verifier Signature (UDVS) [1]. The primary distinction between UDVS and DVS is that, while in DVS the designation is only performed by the signer with the secret key, in UDVS the designation is carried out by the signature holder without knowledge of the secret key. Also, UDVS allows the designated verifier to construct a valid signature intended for him that is indistinguishable from the one produced by the designator. As a result, the verifier is unable to convince a third party that the assertion is true. When a signature holder owns a signature that authenticates his/her private information, such as health data, an income summary, a tenderer's offer in an e-auction system, etc., the non-transferability of UDVS enables it to protect the signature holder's privacy.

Although UDVS has many uses, one of its major drawbacks is that the selected verifier is required to create and certify public/private key pairs using the same public key parameter as the signer. The verifier might not always be motivated to complete the key configuration process. This is because it is only in the designator's interest to demonstrate the knowledge of a signature to a verifier and not the other way around. Let us consider the scenario stated in [2], where UDVS is less practical. Alice has a degree from University A, and she wants to submit an online employment application. She must persuade the employer that she is in fact the holder of the degree certificate that has been authorized by the registrar of University A. Since she believes that anyone who has gotten the certificate may use it for various purposes, she does not want to send her certificate. She, therefore, needs a particular kind of digital signature that can only persuade the employer that she has a legitimate university credential. The employer will be less likely to complete the key setup process in accordance with the public parameter established by the university just to verify Alice's certificate, even though it seems like UDVS would be an excellent fit to handle her diploma verification problem. This is due to the potential cost of key configuration using the public key infrastructure.

To solve this problem, Baek et al. proposed the idea of UDVSP [2], which is similar to UDVS and differs from UDVS in the verification process. In UDVS, the designator provides a proof that he/she genuinely has the signer's signature by using the verifier's public key. UDVSP, in contrast, does not need the verifier's public key and instead uses an interactive communication that the designator executes with the verifier to give a proof. The discrete logarithm problem on which all existing UDVSP constructs are built are vulnerable to attacks from quantum computers. Therefore, the major objective of this work is to provide a solution to privacy issues with certificate distribution in the quantum era because there are no UDVSP methods that are immune to quantum computer assaults. Since code-based cryptosystems are a leading contender for post-quantum cryptosystems, a code-based UDVSP scheme is put forth that consists of a code-based signature, a transform algorithm, and a verification protocol. The suggested approach would be a promising replacement for the current number-theoretic-based schemes.

The paper is organized as follows. Related work is discussed in the following section. The fundamental ideas of coding theory, the definition of the UDVSP system, and its security model are briefed in Section 3. Section 4 proposes an efficient code-based UDVS proof system, and Section 5 presents its security analysis. The performance analysis and comparison of the suggested scheme are presented in Section 6, and the paper is concluded in Section 7.

2. Related work

By adopting a bilinear group, Steinfeld et al. established the first UDVS scheme [1] and introduced the idea of a universal designated verifier signature. Two effective UDVS techniques based on conventional Schnorr/RSA signatures were proposed by Steinfeld et al. [3]. They expanded the currently used Schnorr/RSA signature techniques by adding extra features to UDVS that would allow it to make use of the key generation infrastructure and signing infrastructure that already exists. The drawback of UDVS, however, is that the designated verifier has to generate the key pairs with regard to the parameter selected by the signer. To overcome the above-mentioned limitation, Baek et al. presented two UDVSP methods, one based on the Boneh-Lynn-Shacham (BLS) signature and the other on the Boneh and Boyen (BB) signature. The difficulty of the discrete logarithm problem determines how secure both constructions are. All the above-mentioned schemes are constructed under certificate-based public key system. Zhang et al. introduced the first identity-based UDVS, which derives the user's public key from his or her identification [4]. It is an alternative to certificate-based schemes and simplifies the key management. Yang et al. proposed UDVS in a certificateless public key system [5] and offered the security proof in a random oracle model under the assumption of bilinear Diffie-Hellman. In [6], Chen et al. proposed the concept of Identity-based UDVSP and developed two Identity-based UDVSP systems employing bilinear pairings. The first system is based on the Hess signature scheme and the other is constructed on the Cha-Cheon signature, which eliminates the "Signature Transformation algorithm" suggested by Baek et al. in [2]. Existing UDVSP systems [2, 6] are based on the computational difficulties of solving discrete logarithm problems in a prime field's multiplicative group or an elliptic curve's points over a finite field.

However, in the event of the advent of quantum computers, all of these strategies could be broken due to Shor's algorithm [7]. Indeed, Shor's technique can solve both the factorization issue and the discrete log problem in finite fields in polynomial time. As a result, utilising Shor's algorithm [7], the approaches in [8, 9] will no longer be secure in the quantum era. Quantum mechanisms, on the other hand, such as the non-cloning characteristic, measurement collapse, and the uncertainty principle, provide good foundations for unconditional security in the sense of information theory. Gottesman and Chuang [10] pioneered quantum digital signature (QDS) research in 2001, developing a quantum signature technique based on quantum one-way functions. Following that, several academics thoroughly investigated various sorts of quantum digital signature schemes. A practical quantum designated verifier signature strategy without entanglement based on quantum key distribution technology [11] and a quantum DVS scheme based on an asymmetric quantum public-key system [12] are primarily explored in the subject of designated verifier signatures. To address the security risk in the quantum era, researchers in cryptography developed new quantum-resistant primitives that appear to be resistant to an attacker with access to a quantum computer. It is referred to as "Post-Quantum Cryptography", which is an active area of fundamental research making way for the creation of post-quantum cryptosystems in the real-world. Security of post-quantum cryptosystems rely on different, hard mathematical problems that are resistant to being solved by a large-scale quantum computer. As a result, the cryptography community started to design post-quantum signatures that can withstand both quantum and classical computer attacks. In the world of quantum physics, hash functions, which are one-way functions, are sufficient for efficient and safe data transfer. Thus, it is thought that hash-based cryptosystems are promising quantum-immune cryptosystems. Thanalakshmi et al. proposed a few privacy providing signatures that include a quantum-resistant chameleon signature scheme [13] and a quantum-resistant designated verifier signature scheme [14] based on a homomorphic hash function and homomorphic pseudorandom generator. However, the creation of quantum-resistant homomorphic hash functions is crucial for the development of identification systems and unique signatures like the Chameleon signature and DVS signature in a quantum environment. As code-based signatures are very secure against classical and quantum attacks, they are considered to be another promising alternative to classical digital signatures. Also, code-based signatures come with many strong features apart from being secure from quantum attacks. They include efficient, executable and simple operations such as matrix-vector multiplications. Hence, in comparison to other number-theoretic-based signatures, its signature creation and verification methods are very quick and simple to deploy. Therefore, it is considered to be a practical option to create digital signature systems based on codes. The identity-based signature system [15], ring signature system [16], blind signature system [17], threshold ring signature system [18, 19], one-time signature system [20], signcryption system [21], undeniable signature system [22] and strong designated verifier signature system [23] are few kinds of code-based signature systems. Additionally, employing hard coding theory problems, the author of this research work presented the designated verifier signature [24] and chameleon signature [25]. They determined that the code-based scheme [25] provides a significantly smaller signature size than the hash-based scheme [13]. In this study, the author proposes the first quantum-resistant UDVSP system based on syndrome decoding a challenging coding theory problem. The author also examines the system's security in the context of the random oracle model and establishes that it is resistant to forgery and impersonation attacks.

3. Preliminaries

The UDVS proof system's formal definition and its security requirements are described in this section along with some preliminary discussions of coding theory.

3.1. Coding theory preliminaries

This section, recalls some of the basic definitions of coding theory [24] that are required for the construction of the proposed schemes.

Definition 1. *(Linear code) Let $\mathbb{F}_q$ be a finite field and a linear code C $[n, k]$ is a linear subspace of $\mathbb{F}_q^n$ which has the dimensions k and n. The words are the elements of $\mathbb{F}_q^n$, whereas codewords are the elements of C.*

Definition 2. *(Hamming distance) Let C be a linear code over $\mathbb{F}_q$. The number of locations in which any two codewords differ is the Hamming distance between them.*

Definition 3. *(Hamming weight) Let C be a linear code over $\mathbb{F}_q$. The number of nonzero coordinates of a codeword determines the codeword's Hamming weight.*

Definition 4. *(Minimum distance) Let C be a linear code over $\mathbb{F}_q$. The smallest Hamming distance among all codewords is the minimal distance d of C.*

A code's capacity for error correction depends on its minimum distance. Let d be the minimum distance in C. Then C can correct up to $t = \lfloor \frac{d-1}{2} \rfloor$.

Definition 5. *(Generator matrix) A generator matrix of a linear code C $[n, k]$ is a $k \times n$ matrix G whose row vectors form a basis for the vector subspace C. Thus, $C = \left\{ mG \in \mathbb{F}_q^n : m \in \mathbb{F}_q^k \right\}$.*

Definition 6. *(Parity-check matrix) A parity-check matrix of a $[n, k]$ linear code C is a $(n - k) \times n$ matrix H whose rows serve as the basis of the orthogonal complement of the vector subspace C. Thus, $C = \left\{ c \in \mathbb{F}_q^n : Hc^T = 0 \right\}$.*

Definition 7. *(Syndrome) Given a parity-check matrix H, the syndrome $s \in \mathbb{F}_q^{n-k}$ of any vector $x \in \mathbb{F}_q^n$ is defined as $s = Hx^T$.*

Definition 8. *(Binary syndrome decoding problem) Given an $(n - k) \times n$ parity-check matrix H for a $[n, k]$ linear code C over $\mathbb{F}_2$, a vector $s \in \mathbb{F}_2^{n-k}$, and an integer $t > 0$, find a vector $e \in \mathbb{F}_2^n$ of weight $wt(e) \leq t$ such that $He^T = s^T$. The syndrome decoding problem is denoted by $SD(n, n - k, t)$.*

Definition 9. *$(l - SD(n, n - k, t))$ The problem of solving l simultaneous instances of the $SD(n, n - k, t)$ is denoted by $l - SD(n, n - k, t)$.*

Definition 10. *(Syndrome decoding problem assumption) For a $[n, k]$-code, a probabilistic algorithm D is said to (τ, ϵ)-break the $SD(n, n - k, t)$ if it can decode the syndrome $s^T = He^T$ into an error vector e with weight $wt(e) \leq t$, with probability of at least ϵ.*

3.2. Universal designated verifier signature proof system

There are three participants in a UDVSP system: a signer, a designator (the person who holds the signature), and a designated verifier. A message's signature is created by the signer using the secret key. Following the creation of the key pair, the signer using the secret key signs a message and transmits it to the designator. It's crucial to send the signature through a secure connection to maintain the secrecy of the designator.

The designator utilizes a random mask to construct a changed signature that covers the original signature after receiving a valid message-signature pair from the signer. Using an interactive protocol, the designator then persuades the chosen verifier that the changed signature was created using the signer's original signature. Formal definition of the UDVSP system in [2] is given below:

The following four polynomial-time algorithms, together with a protocol, constitute a UDVSP system [2].

Key Generation (1^{κ})**:** A probabilistic polynomial-time algorithm that generates a signer's public/secret key pair (pk_s, sk_s) using the security parameter κ as input. It is represented by $(pk_s, sk_s) \leftarrow KG(\kappa)$. Also, it produces the designator's key pair $(\overline{pk}, \overline{sk})$.

Sign: A probabilistic procedure that, given a message M and a secret signing key sk_s, outputs a signature σ on M. It is represented by $\sigma \leftarrow Sign(sk_s, M)$.

Verify: A deterministic procedure that returns $Accept$ when the signer's public key pk_s, the message M, and the signature σ are all valid and $Reject$ otherwise. It is represented by $d \leftarrow Verify(pk_s, \sigma, M)$, where $d \in \{Accept, Reject\}$.

Transform: A probabilistic process that creates and outputs a transformed signature employing the signer's public key pk_s, the input signature σ, and the designator's secret mask $\overline{sk}$. It is represented by $\overline{\sigma} \leftarrow Transform(pk_s, \overline{sk}, \sigma)$.

IVerify: This technique for interactive verification involves a designator $\mathcal{P}$ and designated verifier $\mathcal{V}$. The message M, the transformed signature $\overline{\sigma}$, and the signer's public key pk_s are the usual inputs for $\mathcal{P}$ and $\mathcal{V}$. The secret mask $\overline{sk}$ that is utilized to produce $\overline{\sigma}$ is the private input of $\mathcal{P}$. There is no private input available for $\mathcal{V}$. At the conclusion of the protocol, $\mathcal{P}$ attempts to persuade $\mathcal{V}$ that $\overline{\sigma}$ has been produced using the signer's valid signature σ and secret information $\overline{sk}$. This protocol's result is $Accept$ when $\overline{\sigma}$ is a valid signature else $Reject$. It is represented by $d \leftarrow IVerify[\mathcal{P}(\overline{sk}, pk_s, \overline{\sigma}, M) \leftrightarrow \mathcal{V}(pk_s, \overline{\sigma}, M)]$ where $d \in \{Accept, Reject\}$.

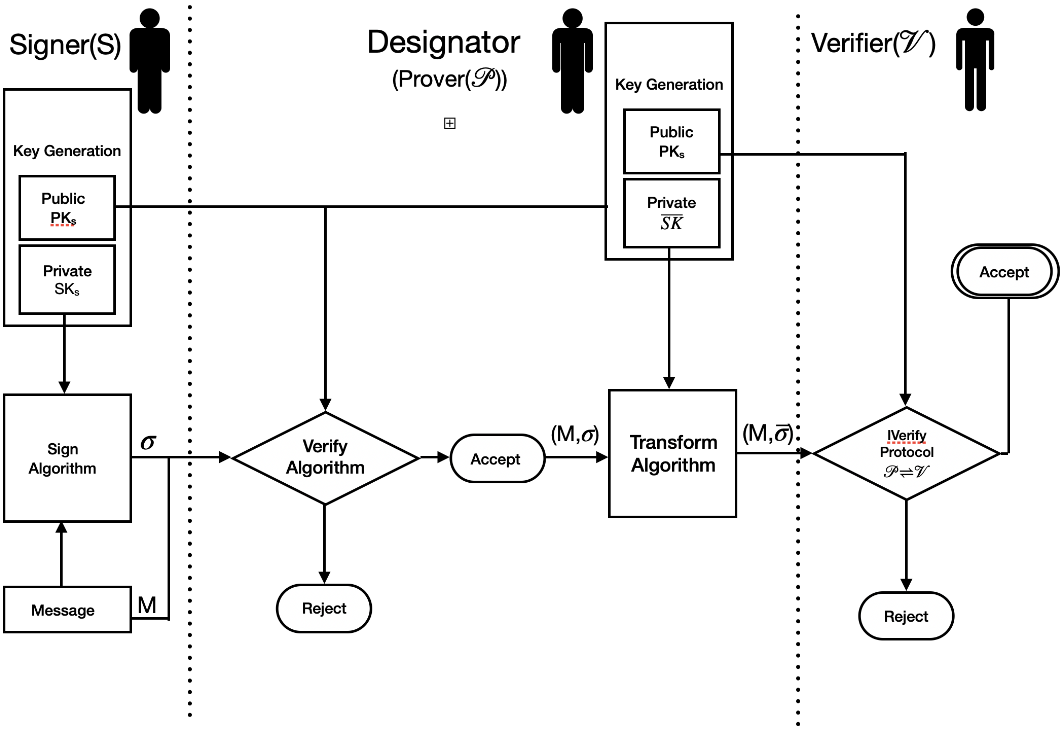

Figure 1. A UDVSP systems.

In a UDVSP system stated in Figure 1, the signer executes the Key Generation and Sign algorithms while the designator does the Verify and Transform algorithms. IVerify protocol is run between the designator and the designated verifier. It is to be noted that Verify is not publicly verifiable. Also, two consistency properties are defined in [2] for a system with UDVSP. The first property indicates that the Verify algorithm should accept the signer's signature σ on a message M as authentic. According to the second property, the IVerify protocol should accept the transformed signature $\overline{\sigma}$ created by the designator using the legal signature σ and the secret mask $\overline{sk}$ as valid.

3.3. Security notions of universal designated verifier signature proof system

The security concepts for the UDVSP system are summarized in this section and are found in [2]. The first security criterion of the UDVSP system requires that a signature provided by the signer be existentially unforgeable under an adaptive chosen message attack. As in [2], unforgeability against selected message attacks is defined in the following manner:

Definition 11. *(Unforgeability against chosen message attack)*

Let κ be the security parameter and the algorithms KG, Sign and Verify be used to generate UDVSP. Take the subsequent experiment $Exp^{f-cma}(\kappa)$ into consideration. A polynomial-time attacker A sends a new message $M*$ with a legitimate signature σ^* after making inquiries to the signing oracle $Sign(sk_s, .)$ in the experiment $Exp^{f-cma}(\kappa)$. If the oracle $Sign(sk_s, .)$ has not yet been queried for $M*$, then Verify $Accepts$ and outputs 1, else it $Rejects$ and outputs 0.

Experiment $(Exp^{f-cma}(\kappa))$:
$KG(\kappa)$ computes (pk_s, sk_s)
$A^{Sign(sk_s,.)}(pk_s)$ computes (M^*, σ^*)
$Verify(pk_s, M^*, \sigma^*)$
If $1 \leftarrow Verify(pk_s, M^*, \sigma^*)$ and M^* has not been questioned by $Sign(sk_s, .)$
output 1
Else
output 0

The adversary A's benefit in the above experiment is defined as $Adv_A^{f-cma}(\kappa) = Pr[Exp^{f-cma}(\kappa) = 1]$. Under a chosen message attack, the UDVSP system's underlying signature scheme is existentially unforgeable, if $Adv_A^{f-cma}(\kappa)$ is negligible.

The UDVSP system's second primary security need is that the IVerify protocol should resist against impersonation attack. In other words, a UDVSP system will stop an attacker from impersonating a trustworthy designator who receives a valid signature from a signer even if they do not obtain a valid signature from the signer. Type-1 and Type-2 attacks are the further classifications for impersonation attacks. In a Type-1 attack, an attacker who is aware of the transformed signature, takes part in the IVerify protocol as a dishonest designated verifier and repeatedly communicates with the truthful designator. The attacker then makes an attempt to fool another honest designated verifier by pretending to be the honest designator. The following is the description for security against the Type-1 attack in [2]:

Definition 12. *(Security against Type-1 impersonation attack)*

Consider the definitions in Definition 11. Consider an attacker named A that operates in polynomial time and consists of two sub-algorithms termed $\mathcal{V}'$ and $\mathcal{P}'$, which stand for a cheating designated verifier and a cheating designated designator, respectively. Assume that $\mathcal{P}$ stands for an honest designator and a function $Conv_{IVerify}$ that creates a transcript of the IVerify protocol dialogue between $\mathcal{P}$ and $\mathcal{V}'$. $\mathcal{P}'s$ and $\mathcal{V}'s$ random coins are represented by the random variable T, which is written as $T \leftarrow Conv_{IVerify}[\mathcal{P} \leftrightarrow \mathcal{V}']$. The following is a formal description of this experiment $Exp^{im-type1}(\kappa)$:

Experiment $Exp^{im-type1}(\kappa)$:
$KG(\kappa)$ computes (pk_s, sk_s)
$Sign(sk_s, M)$ computes σ
$Transform(\overline{sk}, pk_s, \sigma)$ computes $\overline{\sigma}$
$Conv_{IVerify}[\mathcal{P}(\overline{sk}, pk_s, \overline{\sigma}, M) \leftrightarrow \mathcal{V}'(pk_s, \overline{\sigma}, M)]$ for $i = 1, ..., p(\kappa)$ computes T_i
$IVerify[\mathcal{P}'((T_1, r_1^{\mathcal{V}'}), ..., (T_{p(\kappa)}, r_{p(\kappa)}^{\mathcal{V}'})) \leftrightarrow \mathcal{V}(pk_s, \overline{\sigma}, M)]$
output 1 if it $Accepts$
Else
output 0

In the experiment $Exp^{im-type1}(\kappa)$, first, a signer's key pair (pk_s, sk_s) is constructed for a a security parameter κ. $\mathcal{P}$ and $A = (\mathcal{V}', \mathcal{P}')$ are then each given access to the public key pks. Then a signature $sigma$ is generated on an arbitrarily chosen message M. Based on pks, a secret mask $\overline{sk}$ is selected, and $\overline{sk}$ is then used to obtain a transformed signature $\overline{\sigma}$ for σ. $A = (\mathcal{V}', \mathcal{P}')$ is then given $\overline{\sigma}$, $\mathcal{P}$ is given $\overline{sk}$ and $\mathcal{V}'$ now communicates with $\mathcal{P}$ $p(\kappa)$ times in the IVerify protocol, where $p(.)$ stands for a polynomial-time computable function. In the IVerify procedure, after gaining access to the transcripts of these interactions and the random coins that $\mathcal{V}'$ used in them, labelled T_i and $r_i^{\mathcal{V}'}$ respectively for $i = 1, ..., p(\kappa)$, $\mathcal{P}'$ tries to impersonate the honest designator $\mathcal{P}$ to an honest designated verifier $\mathcal{V}$.

Then, in the experiment, $A's$ advantage is $Adv_A^{im-type1}(\kappa) = Pr[Exp^{im-type1}(\kappa) = 1]$. If $Adv_A^{im-type1}(\kappa)$ is too small in κ, the UDVSP system is considered to be secure against impersonation under Type-1 attack. A Type-2 attack involves the attacker trying to create a transformed signature on their own, without using any previously acquired transformed signatures, and uses it to pretend to be an honest designated verifier in the IVerify procedure. According to the study of Baek et al. (2005) in [2], the definition for security against Type-2 attacks is as follows:

Definition 13. *(Security against impersonation under Type-2 attack)*

Consider the definitions as in Definition 11. Let A be a polynomial-time adversary. Take into account the experiment $Exp^{im-type2}(\kappa)$. In this example, a signer's key pair (pks, sks) is created using the security parameter κ. A random message M is then supplied to A, A then independently creates a designator's secret mask $\overline{sk'}$, a modified signature $\overline{\sigma'}$, and A participates in the IVerify protocol along with an honest designated verifier V. The following is a formal description of this experiment $Exp^{im-type2}(\kappa)$:

Experiment $Exp^{im-type2}(\kappa)$:
$KG(\kappa)$ computes (pk_s, sk_s)
$A(\overline{sk'}, pk_s, M)$ computes $\overline{\sigma'}$
$IVerify[A(\overline{sk'}, pk_s, \overline{\sigma'}, M) \leftrightarrow V(pk_s, \overline{\sigma'}, M)]$
output 1 if it $Accepts$
Else output 0

Therefore, in the experiment mentioned above, the benefit of A is $Adv_A^{im-type2}(\kappa) = Pr[Exp^{im-type2}(\kappa) = 1]$. The UDVSP system is regarded as secure against impersonation under Type-2 assault if $Adv_A^{im-type2}(\kappa)$ is too small in κ.

4. Proposed code-based universal designated verifier signature proof system

The suggested UDVSP system is made up of four sub-algorithms, namely, KG, Sign, Verify and Transform and a verification protocol IVerify as defined in [2]. The Sign algorithm uses mCFS algorithm [26]. By adding a random mask, the transform algorithm converts the publicly verifiable mCFS signature to an unverifiable signature. Through five passes of interaction, the IVerify protocol, the prover must persuade the authorized verifier that the transformed signature was created from a legitimate signature. The system's sub-algorithms and protocols are each explained below:

Key generation (1^{κ})**:** Select the parameters m and t such that the complexity of the t decoding in a Goppa code with length $n = 2^m$ and dimension $k = n - tm$ is at least 2^{κ}. Pick a Goppa code that corrects errors in t and a parity check matrix H of order $(n-k) \times n$. Allow the code to remain hidden. Select a permutation matrix R with order $n \times n$ and a non-singular matrix U with order $(n - k) \times (n - k)$ randomly. Output signer's public key $pk_s = H_s$ where $H_s = UHR$ and the secret key $sk_s = (U, H, R)$. Choose a matrix $P \in F_2^{k\times n}$ such that each row has a weight not exceeding t. Calculate V of order $(n-k)\times k$ using the formula $V = H_s P^T$. Output (H_s, V) as designator's public key and P as designator's private key. Solving an instance of $k - SD(n, n-k, t)$ is identical to recovering P from H_s and V. Let $g : \{0,1\}^* \times F_2^{n-k} \to F_2^{n-k} \backslash\{0\}$ define a hash function called g.

Sign (sk_s, M)**:** The signer picks a value r in $\{1, ..., 2^{n-k}\}$ randomly and evaluates $u = R^T Decode_H\left(U^{-1}(g(M,r)^T)\right)$. The process is repeated with another r until $\left(u \neq \perp \& H_s u^T = g(M,r)^T\right)$ if $u = \perp$.

Verify (pk_s, M, σ)**:** The verifier checks whether $H_s u^T = g(M,r)^T$. If the equality is true, then $Accept$ is output. Otherwise, $Reject$ is output.

Transform $(\overline{sk}, pk_s, \sigma)$**:** The secret key $\overline{sk} = (e, P)$, where e is an element in F_2^k. Compute $y = u \oplus eP$ such that $wt(y) \leq t$. Output the transformed signature $\overline{\sigma} = (y, r)$.

IVerify: On receiving the transformed signature $\overline{\sigma} = (y, r)$, the designated verifier $\mathcal{V}$ checks whether $wt(y) \leq t$ and computes Y as $H_s y^T \oplus g(M,r)^T = Y^T$. The verifier accepts the transformed signature only when the designator $\mathcal{P}$ proves the knowledge of e that satisfies the relation $Ve^T = Y^T$ to the designated verifier $\mathcal{V}$ through IVerify protocol. $\mathcal{P}$ proves the knowledge of e to $\mathcal{V}$ by using three blending factors that have the advantage to hide the secret mask e: a random n-bit word x , a random permutation π over $\{1, ..., n\}$ and the private key P of order $k \times n$. Here, $\pi(P)$ denotes the permutation over n columns of matrix P. Hence, the row weights of P and $\pi(P)$ are the same. Also, $\forall \alpha \in F_2^k, \pi \xleftarrow{R} S_n$, $\pi(\alpha P) = \alpha\pi(P)$. The IVerify protocol that runs between $\mathcal{P}$ and $\mathcal{V}$ is denoted by $\mathcal{P}((e, P), (H_s, V), \overline{\sigma}, M) \leftrightarrow \mathcal{V}((H_s, V), \overline{\sigma}, M)$. The following is a description of it.

1) $\mathcal{P}$ computes the commitments c_1 and c_2 as $c_1 = h(\pi, H_s(x)^T)$
   $c_2 = h(\pi(eP \oplus x), \pi(P))$ and sends to $\mathcal{V}$.

2) $\mathcal{V}$ chooses $\alpha \in F_2^k$ at random and passes it to $\mathcal{P}$.

3) $\mathcal{P}$ computes $\beta = \pi((\alpha \oplus e)P \oplus x)$ and sends to $\mathcal{V}$.

4) Now, $\mathcal{V}$ sends an arbitrary challenge $b \in \{0, 1\}$ to $\mathcal{P}$.

5) $\mathcal{P}$ sends π, if $b = 0$
   $\mathcal{P}$ sends $\pi(P)$, if $b = 1$

6) If $\mathcal{V}$ receives π, then $\mathcal{V}$ checks whether the commitment $c_1 = h(\pi, H_s\pi^{-1}(\beta)^T \oplus V\alpha^T \oplus Y^T)$.
   If it receives $\pi(P)$, then $\mathcal{V}$ checks whether the commitment $c_2 = h(\beta \oplus \alpha\pi(P), \pi(P))$. Also, it checks whether each row of $\pi(P)$ has weight $\leq t$.
   If every check is successful, *Accept* is output; if not, *Reject*.

Prover $\mathcal{P}(\overline{sk}, pk_s\overline{\sigma}, M)$		**Verifier** $\mathcal{V}(pk_s, \overline{\sigma}, M)$
Let $h : F_2^n \times \{0,1\}^* \to F_2^n$ be a hash function.		
$x \overset{R}{\leftarrow} F_2^n, \pi \leftarrow S_n$		
$c_1 = h(\pi, H_s x^T)$		
$c_2 = h(\pi(eP \oplus x), \pi(P))$		
	$\xrightarrow{c_1,c_2}$	
	$\xleftarrow{\alpha}$	$\alpha \overset{R}{\leftarrow} F_2^k$
$\beta \leftarrow \pi((\alpha \oplus e)P \oplus x)$	$\xrightarrow{\beta}$	
	$\xleftarrow{challengeb}$	$b \overset{R}{\leftarrow} \{0,1\}$
$if\, b = 0$	$\xrightarrow{\pi}$	
		Check $c_1 = h(\pi, H_s\pi^{-1}(\beta)^T \oplus V\alpha^T \oplus Y^T)$
$if\, b = 1$	$\xrightarrow{\pi(P)}$	
		Check $c_2 = h(\beta \oplus \alpha\pi(P), \pi(P))$ check$wt(\pi(P)) \leq t$

Figure 2. IVerify protocol.

The designator $\mathcal{P}$ persuades the designated verifier $\mathcal{V}$ that the modified signature $\overline{\sigma}$ is derived from a legitimate signature σ without disclosing $\overline{sk}$ via the IVerify protocol shown in Figure 2. Additionally, it should be noted that the IVerify protocol does not require the selected verifier to do setup or compute the private or public key.

5. Security analysis

Completeness: The proposed UDVSP scheme clearly meets the completeness property, because any honest prover who is familiar with the blended secret mask e, the permutation π and a legitimate secret u can successfully respond to any question posed by the honest verifier at any time. Consequently, it is vital to assess the two key security needs of the scheme, including the unforgeability of the signature and the resistance against impersonation attacks. As the Sign algorithm in the proposed UDVSP system uses mCFS algorithm, it has been demonstrated that the signature is unforgeable to chosen message attacks in the random oracle paradigm [26]. Therefore, it is sufficient to evaluate the proposed scheme's security against impersonation attacks.

According to the following theorem, the suggested UDVSP system is secure against impersonation under Type-1 attack:

Theorem 1. *In the random oracle model under Type-1 attack, the code-based UDVSP system is secured against impersonation on the assumption that SD is hard.*

Proof: Let $A = (\mathcal{V}', \mathcal{P}')$ stand for an impersonator attempting to use a Type-1 attack to compromise the code-based UDVSP system. Let C be the challenger that runs the key generation algorithm as follows: The signer's public key is chosen at random by C and is a $(n-k) \times n$ binary matrix called H' Let C be the challenger that executes the following key generation algorithm: the signer's public key $(n-k) \times n$ binary matrix H' is chosen at random by C. Then C picks a matrix $V \in F_2^{(n-k)\times k}$ randomly and gives (H', V) to the adversary A as the prover's public key. Here, (H', V) is an instance of $K - SD(n, n-k, t)$. When A performs the impersonation attack, C tries to output P^* with row weight not exceeding t such that $H'P^{*T} = V$.

C simulates the hash oracle g as follows: for the query (M, r), C chooses an element $u \in F_2^n$ with $wt(u) \leq t$ randomly and computes $s^T = H'u^T$. It stores (u, s) and returns s as the response for $g(M, r)$. C then picks an element $e \in F_2^n$ randomly with $wt(e) \leq t$ and enumerates $y = u \oplus e$ such that $wt(y) \leq t$ and gives (y, r) to A as transformed signature on M. The transformed signature created in the previous simulation has the exact same distribution as the actual structure.
Since,

$$\begin{aligned} H'y^T \oplus g(M, r)^T &= H'(u \oplus e)^T \oplus g(M, r)^T \\ &= H'u^T \oplus H'e^T \oplus g(M, r)^T \\ &= H'e^T \\ &= Y^T \end{aligned}$$

The IVerify protocol is then simulated by C using an honest designator $\mathcal{P}$ and verifier $\mathcal{V}'$ and with the knowledge of e. For an arbitrary challenge $b \in \{0, 1\}$, C answers as follows:

For $b = 0$, C sets a random string to the commitment c_2. C randomly chooses x, π and computes $h(\pi, H'(x)^T)$ and sets as c_1. It gives the commitment c_1, c_2 to $\mathcal{V}'$. By simulating the verifier, C gets $\alpha \in F_2^k$, and computes $\beta = \pi(\alpha P \oplus e \oplus x)$ for some P satisfying the equation $H'P^T = V$ and sends to $\mathcal{V}'$. Here, P need not necessarily satisfy that each row weight $\leq t$. C computes β such that it satisfies $c_1 = h(\pi, H_s\pi^{-1}(\beta)^T \oplus V\alpha^T \oplus H'y^T \oplus g(M, r)^T)$. Thus, for $b = 0$, C makes a simulation which is identical to the results of a fair interaction.

For $b = 1$, C sets a random string to the commitment c_1. It chooses x, e at random and picks a random matrix P whose row weight does not exceed t and computes $h(\pi(e \oplus x), \pi(P))$ and sets as c_2. It gives the commitment c_1, c_2 to $\mathcal{V}'$. By simulating the verifier, C gets $\alpha \in F_2^k$, and computes $\beta = \pi(\alpha P \oplus e \oplus x)$ and sends to $\mathcal{V}'$. C computes β such that it satisfies $c_2 = h(\beta \oplus \alpha\pi(P), \pi(P))$. Thus, for $b = 1$, C makes a simulation that is identical to the results of a fair interaction.

It is clear that when C produces the commitments for some $b = d$ where $d \in \{0, 1\}$ and receives the same d as challenge then C succeeds in the simulation, else it fails in the simulation. Hence, C has a 50 percent chance of producing a successful simulation. After performing $p(\kappa)$ times the execution of the IVerify protocol, the dishonest verifier $\mathcal{V}'$ attempts to mimic the honest designator $\mathcal{P}$ to an honest designated verifier V in the protocol.

Let us consider the situation when impersonation succeeds. When a cheating prover $\mathcal{P}'$ sets the commitments c_1 and c_2 he/she should be prepared to answer any query (α, b) of the verifier. As $\alpha \in F_2^k$ and $b \in \{0, 1\}$, there are $(2^k)2$ possible queries. If $\mathcal{P}'$ is able to give correct answers for any two values of F_2^k for same commitments c_1 and c_2, then there exists at least two values say $\alpha \neq \alpha^{'}$ for which the queries $(\alpha, 0), (\alpha, 1), (\alpha^{'}, 0)$ and $(\alpha^{'}, 1)$ are answered correctly.

Now, let (β, π) be the response sent for the query $(\alpha, 0)$,
(β, z) be the response sent for the query $(\alpha, 1)$,
$(\beta^{'}, \pi^{'})$ be the response sent for the query $(\alpha^{'}, 0)$,
$(\beta^{'}, z^{'})$ be the response sent for the query $(\alpha^{'}, 1)$.

Here the value z with $wt(z) \leq t$ is sent in the place of $\pi(P)$ (similarly $z^{'}$ represents the expected value $\pi^{'}(P)$). Due to the fact that β is transmitted before the bit challenge b, the same β is used for $(\alpha, 0)$ and $(\alpha, 1)$ (similarly $\beta^{'}$ for $(\alpha^{'}, 0)$ and $(\alpha^{'}, 1)$). With both queries $(\alpha, 0)$ and $(\alpha^{'}, 0)$ ($(\alpha, 1)$ and $(\alpha^{'}, 1)$), the commitment c_1 and c_2 are the same.
Hence,
$h(\pi, H'\pi^{-1}(\beta)^T \oplus V\alpha^T \oplus Y^T) = c_1 = h(\pi^{'}, H'(\pi^{'})^{-1}(\beta^{'})^T \oplus V(\alpha^{'})^T \oplus Y^T)$
$h(\beta \oplus \alpha z\, z) = c_2 = h(\beta^{'} \oplus \alpha^{'} z^{'}\, z^{'})$.

This implies that either $\mathcal{P}'$ makes collisions on the hash function or the arguments of the hash function are equal. As h is collision-resistant, the following equalities exist.

$$\pi = \pi^{'} \tag{5.1}$$

$$H'\pi^{-1}(\beta)^T \oplus V\alpha^T \oplus Y^T = H'(\pi^{'})^{-1}(\beta^{'})^T \oplus V(\alpha^{'})^T \oplus Y^T \tag{5.2}$$

$$\beta \oplus \alpha z = \beta^{'} \oplus \alpha^{'} z^{'} \tag{5.3}$$

$$z = z^{'} \tag{5.4}$$

From Eqs (1) and (2),

$$H'\pi^{-1}(\beta \oplus \beta^{'})^T (\alpha \oplus \alpha^{'})^{T^{-1}} = V \tag{5.5}$$

From Eqs (3) and (4),

$$(\alpha \oplus \alpha^{'})^{-1}(\beta \oplus \beta^{'}) = z \tag{5.6}$$

Equations (5) and (6) imply

$$H'\pi^{-1}z^T = V \tag{5.7}$$

Hence, $P = \pi^{-1}z$ with $wt(\pi^{-1}z) = wt(z) \leq t$ constitutes a secret key for the public key (H', V). When an attacker impersonates the real prover and answers correctly the queries $(\alpha, 0), (\alpha, 1), (\alpha^{'}, 0)$

and $(\alpha^{'}, 1)$, the secret key is discovered. When at least 4 queries are correctly replied out of $(2^k)2$ possible queries, an instance of $k - SD(n, n - k, t)$ can be solved. Hence, when the cheating prover manages to answer correctly to the verifier with probability $\geq \frac{2}{2^k}$, the SD problem can be solved with a high degree of probability by a machine with polynomial time probabilistic behavior.

Corollary: The code-based UDVSP system in the random oracle model is protected against Type-2 attack on the assumption that SD is hard.

Proof: In a Type-1 attack, an attacker engages in repeated interactions with an honest designator while acting as a dishonest designated verifier in the IVerify protocol. The attacker tries to appear to other trustworthy designated verifiers as an honest designator using the knowledge of the information they have obtained. Contrarily, a Type-2 assault just ignores the previously acquired altered signature and produces a fresh proof to pose as an honest designator to a trustworthy designated verifier in the IVerify protocol. Theorem 1 ensures that even after repeatedly dealing with an honest designator, a Type-1 attacker cannot obtain any meaningful information to pass for them, which infers that the advantage to impersonate an honest designator by Type-2 attacker and by Type-1 attacker is the same. Hence, the security of the scheme against the impersonation under Type-1 attacker is considered.

Soundness: It is ensured by Theorem 1.

The anonymous credential system is another part of UDVS that is connected [28–30]. As stated in [1], this area of research, however, is more concerned with user privacy issues such "selective disclosure" of attribute information and "unlinkability" of user transaction data. Our work and work on UDVS [1–4] (in general) focuses more on offering an effective method of persuading designated verifiers that a signature holder actually has a legitimate signature from the original signer. As a result, we have avoided using complicated zero knowledge proof procedures, such those seen in many credential systems [28–30].

6. Performance analysis and comparison

Stern's identification scheme [27] can be used to construct the IVerify protocol. Then, the soundness of error will be 2/3, whereas in the proposed five-pass IVerify protocol the dishonest prover has only a 50% chance of fooling the verifier and passing the verification test. Here, the soundness of error is 1/2. Thus, by using the IVerify protocol, the soundness of error can be reduced from 2/3 to 1/2. When the protocol runs several times, the probability that the dishonest prover succeeds is very low without knowing the secret key. As a result, any desired level of security can be attained in fewer rounds and thereby reducing the communication complexity.

The proposed UDVSP's performance is contrasted with that of the current UDVSP systems. Consider a Goppa code $[n, k]$, where $n = 2^m$, k and $t = \frac{n-k}{m}$ are the length, dimension and error-correcting capability of the code. For a security parameter κ, the complexity of t-decoding is least 2^{κ}. These parameters are used to calculate all aspects of the scheme, including the length of the signature and the algorithmic complexity required to generate it. The signature in the proposed UDVSP system is a pair (u, r), where u is a bit vector of length n and weight t , which can be stored in only $\log_2 \binom{n}{t}$ bits, and r requires mt bits. Consequently, the signature's size is $\log_2 \binom{2^m}{t} + mt$. The production of signatures takes $t!$ decodings, each requiring m^3t^2 bit operations. The cost of generating a signature is therefore $t!(m^3t^2)$. One syndrome computation, requiring m^3t^2 bit operations, is needed for verification. As a result, the cost of creating a signature is $t!(m^3t^2)$.

Comparison of computational costs of the proposed UDVSP transformed signature generation, IVerify protocol and transformed signature length with that of some existing UDVSP are given in Table 1. The transformed signature consists of two elements (y, r) where y is an n bit vector of weight t and r is an mt bit vector. Consequently, $\log_2 \binom{2^m}{t} + mt$ is the size of transformed signature. The creation of a transformed signature, which involves one matrix-vector multiplication (MVM), takes kn bit operations. IVerify protocol requires 6 matrix-vector multiplications $H_s y^T$, $H_s x^T$, eP, $(\alpha + e)P$, $H_s \pi^{-1}(\beta)^T$, and $V\alpha^T$ where $H_s \in \mathbb{F}_2^{(n-k)\times n}$, $V \in \mathbb{F}_2^{(n-k)\times k}$, $P \in \mathbb{F}_2^{k\times n}$, $y, x, \beta \in \mathbb{F}_2^n$, $\alpha \in \mathbb{F}_2^k$ with $wt(y) \leq t$. Hence, its computational cost consists of $(n-k)t+(n-k)n+kn+kn+(n-k)n+kn < 3(n-k)n+3kn < 3n^2$ bit operations. The advantage of the proposed UDVSP is that it is quantum-safe since the security of it depends on the hardness of SD. Also, compared to the UDVSP-BB and UDVSP-Hess schemes, the transformed signature length is shorter. Although the proposed UDVSP system's transformed signature length is marginally longer than the UDVSP-BLS scheme, the transform signature generation and IVerify protocol of the proposed scheme are faster compared to the UDVSP-BLS scheme as it requires only matrix multiplications without any exponentiation and complex pairing operations. Comparison of proposed UDVSP with existing UDVSP is given in Tables 1 and 2.

Table 1. Comparison of signatures in UDVSPs.

Scheme	Hard Problem	Key gen. Time	Sign Time	Verify Time	Sig. Length (bits)
BLS [2]	CDH	1exp	1exp	2Pairing	160
BB [2]	SDH	2exp.	1exp.	2 Pairing + 2exp.	320
Hess [6]	CDH	1 Multip.	1 Pairing	2 Pairing + 1exp.	1120
mCFS [26]	SD	2MVM	Decoding	1 MVM.	270

CDH: Computational Diffie-Hellman, SDH: Strong Diffie-Hellman.

Table 2. Comparison of transformed signatures in UDVSPs.

Scheme	Trans. Time	IVerify Time	TSig. Length (bits)	QS
UDVSP-BLS [2]	1exp.	2 pairing +3exp.	160	No
UDVSP-BB [2]	1exp.	2pairing +3exp.	320	No
ID-UDVSP-Hess [6]	1 Multip.	2pairing +3exp.	1120	No
Prop.UDVSP-mCFS	1MVM.	< 3exp.	270	Yes

QS: Quantum Safe, TSig: Transform signature.

7. Conclusions

In order to address the privacy concerns raised by the distribution of signed digital certificates in the quantum era, this work proposes a post quantum UDVSP scheme based on coding theory. The proposed system eliminates the disadvantage of UDVS, which requires a designated verifier to generate a key pair with the signer's public key as a parameter in order to verify the designator's claim. The comparison of the proposed system to the current UDVSP reveals that the proposed system is very fast in the signing and verification process, as well as resistant to quantum computer attacks. Furthermore, when using the random oracle architecture, the system is protected against forgeability and impersonation threats, according to the security study.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. 2020-0-00085, Research on Security Technology for 6G Telecommunication with 5G+).

Anbazhagan would like to thank RUSA Phase 2.0 (F 24-51/2014-U), DST-FIST (SR/FIST/MS-I/2018/17), DST-PURSE 2nd Phase programme (SR/PURSE Phase 2/38), Govt. of Indiaase 2/38, Govt. of India.

Conflict of interest

The authors declare no conflict of interest.

References

1. R. Steinfeld, L. Bull, H. Wang, J. Pieprzyk, International conference on the theory and application of cryptology and information security, In: *Advances in cryptology-asiacrypt 2003*, Heidelberg: Springer, 2003, 523–542. https://doi.org/10.1007/978-3-540-40061-5_33
2. J. Baek, R. Safavi-Naini, W. Susilo, International conference on the theory and application of cryptology and information security, In: *Advances in cryptology-asiacrypt 2003*, Heidelberg: Springer, 2005, 644–661. https://doi.org/10.1007/11593447_35
3. R. Steinfeld, H. Wang, J. Pieprzyk, Efficient extension of standard Schnorr/RSA signatures into universal designated-verifier signatures, In: *Public key cryptography-PKC 2004*, Heidelberg: Springer, 2004, 86–100. https://doi.org/10.1007/b95631
4. R. Zhang, J. Furukawa, H. Imai, Short signature and universal designated verifier signature without random oracles, In: *Applied cryptography and network security*, Heidelberg: Springer, 2005, 483–498. https://doi.org/10.1007/b137093
5. M. Yang, X. Q. Shen, Y. M. Wang, Certificateless universal designated verifier signature schemes, *The Journal of China Universities of Posts and Telecommunications*, **14** (2007), 85–90. https://doi.org/10.1016/S1005-8885(07)60154-X
6. X. Chen, G. Chen, F. Zhang, B. Wei, Y. Mu, Identity-based universal designated verifier signature proof system, *International Journal of Network Security*, **8** (2009), 52–58. https://doi.org/10.1007/11596042_85

7. P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on quantum computer, *SIAM Rev.*, **41** (1999), 303–332. https://doi.org/10.1137/S0036144598347011

8. J. Li, N. Qian, Y. Zhang, X. Huang, An efficient certificate-based designated verifier signature scheme, *Comput. Informatics*, **35** (2016), 1210–1230.

9. P. Rastegari, M. Berenjkoub, M. Dakhilalian, W. Susilo, Universal designated verifier signature scheme with non-delegatability in the standard model, *Inform. Sciences*, **419** (2019), 321–334. https://doi.org/10.1016/j.ins.2018.12.020

10. D. Gottesman, I. Chuang, Quantum digital signatures, arXiv: quant-ph/0105032.

11. M. Zheng, K. Xue, S. Li, N. Yu, A practical quantum designated verifier signature scheme for E-voting applications, *Quantum Inf. Process.*, **20** (2021), 230. https://doi.org/10.1007/s11128-021-03162-5

12. X. Xin, L. Ding, C. Li, Y. Sang, Q. Yang, F. Li, Quantum public-key designated verifier signature, *Quantum Inf. Process.*, **21** (2022), 33. https://doi.org/10.1007/s11128-021-03387-4

13. P. Thanalakshmi, R. Anitha, N. Anbazhagan, W. Cho, G. P. Joshi, E. Yang, A hash-based quantum-resistant chameleon signature scheme, *Sensors*, **21** (2021), 8417. https://doi.org/10.3390/s21248417

14. P. Thanalakshmi, R. Anitha, N. Anbazhagan, C. Park, G. P. Joshi, C. Seo, A hash-based quantum-resistant designated verifier signature scheme, *Mathematics*, **10** (2022), 1642. https://doi.org/10.3390/math10101642

15. P. L. Cayrel, P. Gaborit, M. Girault, Identity-based identification and signature schemes using correcting codes, *International Workshop on Coding and Cryptography (IWCC)*, Fujian, China, 2007, 69–78.

16. D. Zheng, X. Li, K. Chen, Code-based ring signature scheme, *Int. J. Netw. Secur.*, **5** (2004), 154–157.

17. R. Overbeck, A step towards QC blind signatures, *IACR Cryptol. ePrint Arch.*, **2009** (2009), 102.

18. D. S. Wong, K. Fung, J. K. Liu, V. K. Wei, On the RS-code construction of ring signature schemes and a threshold setting of RST, In: *International conference on information and communications security*, Heidelberg: Springer, 2003, 34–46. https://doi.org/10.1007/978-3-540-39927-8_4

19. L. Dallot, D. Vergnaud, Provably secure code-based threshold ring signatures, In: *IMA international conference on cryptography and coding*, Heidelberg: Springer, 2009, 222–235. https://doi.org/10.1007/978-3-642-10868-6_13

20. P. S. Barreto, R. Misoczki, J. M. A. Simplicio, One-time signature scheme from syndrome decoding over generic error-correcting codes, *J. Syst. Software*, **84** (2011), 198–204. https://doi.org/10.1016/j.jss.2010.09.016

21. K. P. Mathew, S. Vasant, C. P. Rangan, A provably secure signature and signcryption scheme using the hardness assumptions in coding theory, In: *International conference on information security and cryptology*, Cham: Springer, 2013, 342–362. https://doi.org/10.1007/978-3-319-12160-4_21

22. C. Aguilar-Melchor, S. Bettaieb, P. Gaborit, J. A. Schrek, A code-based undeniable signature scheme, In: *IMA international conference on cryptography and coding*, Heidelberg: Springer, 2013, 99–119. https://doi.org/10.1007/978-3-642-45239-0_7

23. M. R. Asaar, M. Salmasizadeh, M. R. Aref, Code-based strong designated verifier signatures security analysis and a new construction, *IACR Cryptol. ePrint Arch.*, **2016** (2016), 779.

24. P. Thanalakshmi, R. Anitha, A new code-based designated verifier signature scheme, *Int. J. Commun. Syst.*, **31** (2018), e3803. https://doi.org/10.1002/dac.3803

25. P. Thanalakshmi, R. Anitha, A quantum resistant chameleon hashing and signature scheme, *IETE J. Res.*, **68** (2022), 2271–2282. https://doi.org/10.1080/03772063.2019.1698323

26. L. Dallot, Towards a concrete security proof of Courtois, Finiasz and Sendrier signature scheme, In: *Western european workshop on research in cryptology*, Heidelberg: Springer, 2007, 65–77. https://doi.org/10.1007/978-3-540-88353-1_6

27. J. Stern, A new paradigm for public key identification, *IEEE T. Inform. Theory*, **42** (1996), 1757–1768. https://doi.org/10.1109/18.556672

28. A. Lysyanskaya, R. Rivest, A. Sahai, S. Wolf, Pseudonym systems, In: *International workshop on selected areas in cryptography*, Heidelberg: Springer, 1999, 184–199. https://doi.org/10.1007/3-540-46513-8_14

29. D. Chaum, H. Antwerpen, Undeniable signatures, In: *Conference on the theory and application of cryptology*, New York: Springer, 1990, 212–216. https://doi.org/10.1007/0-387-34805-0_20

30. J. Camenisch, A. Lysyanskaya, An efficient system for non-transferable anonymous credentials with anonymity revocation, In: *Advances in cryptology-eurocrypt 2001*, Heidelberg: Springer, 2001, 93–118. https://doi.org/10.1007/3-540-44987-6_7

AIMS Press