IET Information Security

Research Article

Journals
The Institution of Engineering and Technology

Multi-designated verifiers signature schemes with threshold verifiability: generic pattern and a concrete scheme in the standard model

ISSN 1751-8709
Received on 1st March 2018
Revised 3rd February 2019
Accepted on 25th February 2019
E-First on 3rd April 2019
doi: 10.1049/iet-ifs.2018.5063
www.ietdl.org

Parvin Rastegari[1] ✉, *Mohammad Dakhilalian*[1], *Mehdi Berenjkoub*[1], *Willy Susilo*[2]

[1]*Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan 84156-83111, Iran*

[2]*Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Wollongong NSW 2522, Australia*

✉ *E-mail: parvin.rastegari@ec.iut.ac.ir*

Abstract: In a designated verifier signature (DVS) scheme, the validity of the signature can only be checked by a designated entity chosen by the signer. Furthermore, the designated entity cannot convince a third party that the signature is generated by the signer. A multi-designated verifiers signature (MDVS) scheme is an extension of a DVS which includes multiple designated verifiers. To the best of the authors' knowledge, there are two existing patterns for an MDVS scheme. In the first pattern, every verifier of the set of designated verifiers can check the validity of the signature independently. In the second pattern, the cooperation of all designated verifiers is required for checking the validity of the signature. In this study, the authors propose a generic new pattern for an MDVS scheme in which a threshold number of the set of designated verifiers can check the validity of the signature. They also present a concrete MDVS scheme with threshold verifiability in the standard model. Moreover, they compare their scheme with other existing MDVS schemes. Finally, they briefly explain scenarios in which the proposed pattern can be applicable.

1 Introduction

Digital signature is an important primitive to provide integrity and authenticity of messages in security protocols [1]. A traditional digital signature scheme is publicly verifiable, that is every entity can check the validity of the signature by the signer's public key. The privacy of the signer is not preserved in a traditional digital signature, since a verifier can convince any third party that the signer has really signed a message by presenting the signer's signature on the message to the third party. As a result, although the public verifiability of digital signatures is a useful and necessary property in many applications, it is not a desirable property in applications such as e-votings, e-auctions and fair exchanges in which integrity and authenticity are required without disturbing the privacy of the signer.

Many researchers have proposed different solutions to overcome the conflicts between the authenticity and the privacy of the signer in digital signatures. In 1989, Chaum and Van Antwerpen [2] introduced the concept of undeniable signature in which some help of the signer is required in the verification phase. To avoid the interaction between the signer and the verifier, the concept of designated verifier signature (DVS)/proof was presented by Jakobsson *et al.* [3] and independently by Chaum [4] in 1996. In a DVS scheme, a signer (Alice) can convince a designated verifier (Bob) that she has really signed a message, while Bob cannot transfer this conviction to any third party since he can produce a signature indistinguishable from that generated by Alice. As a result, the authenticity of Alice is proved to Bob and also her privacy is preserved at the same time, without any interaction between Alice and Bob. In [3], Jakobsson *et al.* also introduced the concept of strong DVS, in which the private key of the designated verifier is required to verify the signature. In [5], Steinfeld *et al.* introduced the concept of universal DVS in which every party who holds the signer's traditional signature on a message is able to transform it to a designated signature for a specific verifier.

In [3], the idea of multiple designated verifiers was discussed. Later in 2003, Desmedt proposed the concept of multi-designated verifiers signature (MDVS) scheme as a generalisation of a DVS [6]. This notion was first formalised in 2004 by Laguillaumie and Vergnaud [7]. Since then, a number of MDVS schemes with various properties in different setting models have been proposed [8–23]. These MDVS schemes are categorised in the two following patterns:

- In the first pattern, every verifier of the set of designated verifiers is able to verify the validity of the signature by its own [3, 7, 9–14, 16–18, 21, 23]. This pattern is categorised in two cases:

  a The verification algorithm does not take any private keys of designated verifiers as input [3, 7, 12–14].
  b The verification algorithm takes the private key of a single verifier of the set of all designated verifiers as input [9–11, 14, 16–18, 21, 23].

In case 1, everyone who receives the signature can verify it, because any private keys of designated verifiers are not required in the verification phase. Hence, although the designated verifiers cannot transfer the signer's signature to any third party (since they can produce a signature indistinguishable from that generated by the signer), the privacy of the signer is not preserved, because everyone who receives the signature from the channel between the signer and the designated verifiers can verify it and convince about the signer's signature. This event contradicts the main goal of a DVS namely to preserve the privacy of the signer. As a result, we only consider case 2 as the first pattern in the rest of the paper.

- In the second pattern, all designated verifiers have to cooperate in order to verify the validity of the signature, since the verification algorithm takes the private keys of all designated verifiers as input [8, 15, 19, 20, 22].

In this paper, we propose a new pattern for an MDVS in which a threshold number of the set of designated verifiers are able to verify the validity of the signature, cooperatively. We also present a concrete MDVS scheme with threshold verifiability in the standard model, based on the proposed generic pattern. Moreover, after a comparison between the existing schemes with our proposed scheme, we explain scenarios in which our proposal can be applicable. Note that (M)DVS schemes are applicable in various applications such as e-votings [24], fair exchanges [25] and cloud computing [26, 27]. One can use each of the above patterns (first, second or our pattern) of an MDVS which is appropriate for the corresponding application, i.e. for each application the proper pattern must be selected.

The rest of this paper is organised as follows. In Section 2, some preliminaries are provided which will be used throughout this work. Section 3 covers the formal models and the basic security requirements of two existing patterns for an MDVS scheme as well as our proposed generic pattern for an MDVS with threshold verifiability. In Section 4, we present a concrete MDVS scheme with threshold verifiability in the standard model, based on our proposed generic pattern. In Section 5, a comparison between MDVS schemes is provided. We explain about the applications of our proposal in Section 6. Finally, Section 7 contains the concluding remarks.

2 Preliminaries

In this section, some required preliminaries are described which will be used throughout the paper.

2.1 Bilinear pairings

Let G_1 and G_2 be two multiplicative cyclic groups of prime order q and let g be a generator of G_1. There exists an admissible bilinear pairing $e: G_1 \times G_1 \to G_2$ if and only if the following properties are satisfied:

i. Bilinearity: $e(g^a, g^b) = e(g, g)^{ab}$, for all $a, b \in Z_q^*$.
ii. Non-degeneracy: i.e. $e(g, g) \neq 1_{G_2}$.
iii. Computability: There exists an efficient algorithm for computing $e(g, g)$.

It can be referred to [28] for more details about bilinear pairings.

2.2 Complexity assumptions

Some problems in bilinear pairings are considered as hard problems in complexity theory. Some of these hard problems are as follows:

- Computational bilinear Diffie-Hellman problem: On inputs $g, g^a, g^b, g^c \in G_1$, for unknown $a, b, c \in Z_q^*$, calculate $e(g, g)^{abc} \in G_2$.
- Decisional bilinear Diffie-Hellman (DBDH) problem: On inputs $g, g^a, g^b, g^c \in G_1$, for unknown $a, b, c \in Z_q^*$, and $X \in G_2$, decide whether $X = e(g, g)^{abc}$.
- Gap bilinear Diffie-Hellman (GBDH) problem: On inputs $g, g^a, g^b, g^c \in G_1$, for unknown $a, b, c \in Z_q^*$, calculate $e(g, g)^{abc} \in G_2$ with the help of the DBDH oracle $\mathcal{O}_{\text{DBDH}}$. The DBDH oracle $\mathcal{O}_{\text{DBDH}}$ is that given $g, g^a, g^b, g^c \in G_1$ and $X \in G_2$, outputs 1 if $X = e(g, g)^{abc}$ and 0 otherwise.

Definition 1: It is said that the $(t', \varepsilon')-$ GBDH assumption holds in (G_1, G_2), if no t'-time algorithm has advantage at least ε' in solving the GBDH problem in (G_1, G_2).

3 MDVS schemes

In this section, the formal models and the basic security requirements of two existing patterns and our proposed pattern for MDVS schemes are introduced.

3.1 Formal models of existing patterns

An MDVS scheme includes of a signer s and a set of n designated verifiers $\{v_1, v_2, \ldots, v_n\}$, and is defined by five main algorithms: Setup, signer key generation (SKG), verifiers key generation (VKG), designated signature generation (DSign) and designated signature verification (DVer). Two existing patterns are similar in all algorithms except in the DVer algorithm. These algorithms are defined as follows [7–23].

Setup: It is a probabilistic polynomial time (PPT) algorithm which takes as input a security parameter k and outputs system parameters *params*, i.e.

$$params \leftarrow Setup(k).$$

SKG: It is a PPT algorithm which takes as input *params* and outputs a private/public key pair (Sk_s, Pk_s) for the signer, i.e.

$$(Sk_s, Pk_s) \leftarrow SKG(params).$$

VKG: It is a PPT algorithm which takes as inputs *params* and the number of verifies n, and outputs private/public key pairs (Sk_{v_i}, Pk_{v_i}) for $i = 1, 2, \ldots, n$, i.e.

$$(Sk_{v_i}, Pk_{v_i}) \leftarrow VKG(params, n).$$

DSign: It is a PPT algorithm which takes as inputs a message m, the system parameters *params*, the signer's private key Sk_s and n designated verifiers' public keys $\mathscr{V} = \{Pk_{v_1}, Pk_{v_2}, \ldots, Pk_{v_n}\}$, and outputs a designated signature σ on message m, i.e.

$$\sigma \leftarrow DSign(m, params, Sk_s, \mathscr{V}).$$

DVer: This algorithm is defined differently in two existing patterns.

- DVer in the first pattern [18]: It is a deterministic polynomial time algorithm which takes as input *params*, the message/designated signature pair (m, σ), the signer's public key Pk_s, the verifiers' public keys $\mathscr{V} = \{Pk_{v_1}, Pk_{v_2}, \ldots, Pk_{v_n}\}$ and one verifier's private key $Sk_{v_i} \in \mathscr{S} = \{Sk_{v_1}, Sk_{v_2}, \ldots, Sk_{v_n}\}$, and outputs 1 if the designated signature is valid and 0 otherwise, i.e.

$$DVer((m, \sigma), params, Pk_s, \mathscr{V}, Sk_{v_i}) := 0/1.$$

- DVer in the second pattern [19]: It is a deterministic polynomial time algorithm which takes as input *params*, the message/designated signature pair (m, σ), the signer's public key Pk_s, the verifiers' public keys $\mathscr{V} = \{Pk_{v_1}, Pk_{v_2}, \ldots, Pk_{v_n}\}$ and all verifiers' private keys $\mathscr{S} = \{Sk_{v_1}, Sk_{v_2}, \ldots, Sk_{v_n}\}$, and outputs 1 if the designated signature is valid and 0 otherwise, i.e.

$$DVer((m, \sigma), params, Pk_s, \mathscr{V}, \mathscr{S}) := 0/1.$$

3.2 Formal model of our proposed pattern

Here, we propose a pattern for an MDVS in which the signature can be verified by the cooperation of a threshold number of designated verifiers. In the rest of the paper, we use the notation $(t, n)-$ MDVS for an MDVS which is verifiable by the cooperation of a threshold number t of n designated verifiers. Our pattern is also defined by five main algorithms: Setup, SKG, VKG, DSign and DVer. The Setup, SKG, VKG and DSign algorithms are defined similar to the existing patterns with an extra assumption that in the *VKG* phase, n designated verifiers may run a $(t, n)-$ secret sharing [29] between themselves in order to obtain their shares of other verifiers' secret keys. Moreover, the DVer algorithm is defined as follows.

DVer: It is a deterministic polynomial time algorithm which takes as inputs *params*, the message/designated signature pair (m, σ), the signer's public key Pk_s, the verifiers' public keys $\mathscr{V}$ and t verifiers' shares of all verifiers' secret keys $\mathscr{S}^t$, and outputs 1 if the designated signature is valid and 0 otherwise, i.e.

$$DVer((m, \sigma), params, Pk_s, \mathscr{V}, \mathscr{S}^t) := 0/1.$$

3.3 Security requirements

Correctness, unforgeability and non-transferability (source hiding) are three basic requirements of an MDVS scheme [7].

Correctness: Correctness must be satisfied in an MDVS scheme. This property is considered as follows:

- In the first pattern [18], if

$$\sigma \leftarrow DSign(m, params, Sk_s, \mathscr{V}),$$

then the output of $DVer((m,\sigma), params, Pk_s, \mathscr{V}, Sk_{v_i})$ (for all $i \in \{1,2,\ldots,n\}$) must be 1. Furthermore, for any values (m,σ), $\mathscr{V}$ and Pk_s, if there exists an $Sk_{v_j} \in \mathscr{S}$ such that

$$DVer((m,\sigma), params, Pk_s, \mathscr{V}, Sk_{v_j}) := 1,$$

then for any $Sk_{v_i} \in \mathscr{S}, (i \neq j)$, it must hold that

$$DVer((m,\sigma), params, Pk_s, \mathscr{V}, Sk_{v_i}) := 1.$$

- In the second pattern [19], if

$$\sigma \leftarrow DSign(m, params, Sk_s, \mathscr{V}),$$

then the output of $DVer((m,\sigma), params, Pk_s, \mathscr{V}, \mathscr{S})$ must be 1.

- In our pattern (i.e. $(t,n) -$ MDVS), if

$$\sigma \leftarrow DSign(m, params, Sk_s, \mathscr{V}),$$

then the output of $DVer((m,\sigma), params, Pk_s, \mathscr{V}, \mathscr{S}^t)$ must be 1 for all $\mathscr{S}^t$s. In other words, if σ is a valid designated signature on message m, it must pass the DVer phase which is performed by the cooperation of every t verifiers of the set of all verifiers.

Unforgeability: Unforgeability is considered as existential unforgeability against chosen message attack and is defined by the following game between an adversary $\mathscr{A}$ and a challenger $\mathscr{C}$ [7].

Setup: $\mathscr{C}$ runs Setup, SKG and VKG algorithms to obtain $(params, (Sk_s, Pk_s), (\mathscr{S}, \mathscr{V}))$ and gives $(params, Pk_s, \mathscr{V})$ to $\mathscr{A}$.

Oracle accesses: $\mathscr{A}$ has access to the following oracles:

$\mathscr{O}_{\text{Sign}}$: Refers to the designated signature oracle which takes as input a message m and outputs $\sigma \leftarrow DSign(m, params, Sk_s, \mathscr{V})$.

$\mathscr{O}_{\text{Ver}}$: Refers to the verification oracle which takes as input a pair (m,σ) and outputs 1 if σ is a valid MDVS on m, and 0 otherwise.

Forgery: $\mathscr{A}$ outputs (m^*, σ^*). ($\mathscr{A}$ is not allowed to submit a query from $\mathscr{O}_{\text{Sign}}$ with input m^*.)

It is said that $\mathscr{A}$ wins the unforgeability game if σ^* is a valid designated signature on m^*, i.e.

- In the first pattern, there exists a public key $Pk_{v_j} \in \mathscr{V}$ such that

$$DVer((m^*, \sigma^*), params, Pk_s, \mathscr{V}, Sk_{v_j}) := 1.$$

- In the second pattern

$$DVer((m^*, \sigma^*), params, Pk_s, \mathscr{V}, \mathscr{S}) := 1.$$

- In our proposed pattern, for all $\mathscr{S}^t$s, it holds that

$$DVer((m^*, \sigma^*), params, Pk_s, \mathscr{V}, \mathscr{S}^t) := 1.$$

Definition 2: It is said that an MDVS scheme is $(t'', \varepsilon'', q_S, q_V) -$ unforgeable, if no PPT adversary with at most q_S queries from $\mathscr{O}_{\text{Sign}}$ and q_V queries from $\mathscr{O}_{\text{Ver}}$ can win the unforgeability game in time at most t'' with probability at least ε''.

Non-transferability (source hiding): Non-transferability is considered to guarantee the privacy of the signer. Non-transferability is ensured by a transcript simulation algorithm that can be performed by the cooperation of all designated verifiers to generate a signature which is indistinguishable from the one that should be generated by the signer. This property is defined the same in two existing patterns [7–15, 17–23], as well as our proposed pattern (this property is defined a little different in [16]).

Definition 3: It is said that an MDVS scheme is non-transferable if there exists a PPT transcript simulation algorithm (*TS*) that on inputs public parameters *params*, the signer's public key Pk_s, the set of all verifiers' public/private keys $(\mathscr{S}, \mathscr{V})$ and a message m, outputs a designated signature σ, which is indistinguishable from which is produced by the signer. In other words, for all PPT algorithms $\mathscr{D}$, for any security parameter k, $params \leftarrow Setup(k)$, $(Sk_s, Pk_s) \leftarrow SKG(params)$, $(\mathscr{S}, \mathscr{V}) \leftarrow VKG(params, n)$ and any message m, the value of

$$\left| \Pr\left[\begin{pmatrix} \sigma_0 \leftarrow DSign(m, params, Sk_s, \mathscr{V}) \\ \sigma_1 \leftarrow TS(m, params, Pk_s, \mathscr{S}, \mathscr{V}) \\ b \in_R \{0,1\} \\ b' \leftarrow \mathscr{D}(\sigma_b, m, params, Pk_s, Sk_s, \mathscr{V}, \mathscr{S}) \end{pmatrix} : b = b'\right] - \frac{1}{2} \right| \tag{1}$$

must be negligible.

Remark 1: We must emphasise that unforgeability and non-transferability are considered as two basic security requirements for an MDVS scheme. However, there are some enhanced security requirements such as strong unforgeability, resistance against rogue-key attack, resistance against consistency attack and resistance against key exposure attack [17]. Unfortunately, there is not any security model for an MDVS which guarantees all these security requirements in the literature. In other words, in almost all schemes only unforgeability and non-transferability are modelled and proved. The scheme in [19] (which we will use it as the base of our concrete scheme) not only satisfies unforgeability and non-transferability in the standard model, but also it seems to be resistant against all other mentioned attacks, except the rogue key attack which can easily be solved by a proof of knowledge of a party's private key [17]. Note that this proof of knowledge is commonly applied in the traditional public key infrastructure by a certificate authority.

Remark 2: In 2005, a new security notion called as non-delegatability was proposed for DVS schemes [30]. According to this property, neither the signer nor the designated verifier is able to delegate the signing rights to a third party without revealing his/her private key. To the best of our knowledge, no non-delegatable MDVS scheme has been proposed in the literature till now (see [31]). Moreover, Tian *et al.* [32] pointed out that although the non-delegatability has been a focus of many recent researches, it may be undesirable in some applications. Therefore, the non-delegatable DVS schemes should be considered as a special category which are useful in specific applications where the responsibility of the signer is important and cannot be delegated to another entity. Hence, we do not consider this property in our proposal.

According to Remarks 1 and 2, we just consider the basic requirements of an MDVS namely correctness, unforgeability and non-transferability in our proposed new pattern. Moreover, the threshold verifiability is another security requirement of our proposal.

Threshold verifiability: This property is only considered in our proposed pattern for a $(t,n) -$ MDVS scheme and guarantees that at least the cooperation of a threshold number of designated verifiers is necessary in order to verify the validity of the signature. In other words, every subset of the set of n designated verifiers with t or more members should be able to check the validity of the signature, cooperatively and no subset of the set of n designated verifiers with less than t members can cooperate to check the validity of the signature.

Definition 4: A $(t, n)-$ MDVS scheme is threshold verifiable if the signature can only be verified by the cooperation of at least t designated verifiers.

In the next section, we propose a concrete $(t, n)-$ MDVS scheme and prove its security requirements in the standard model (without random oracles).

4 Proposed concrete (t, n) – MDVS scheme

In this section, we present a concrete $(t, n)-$ MDVS scheme and prove its security requirements in the standard model (without random oracles). In [19], a universal designated multi-verifier signature scheme is presented in the standard model in which the cooperation of all designated verifiers is necessary to check the validity of the signature as the mentioned second pattern in Section 3. Ming and Wang [19] use the waters' signature [33] as the base of their scheme. We use the scheme in [19] as the base of our proposed concrete $(t, n)-$ MDVS scheme in this section. Furthermore, we use Shamir secret sharing [29] to provide threshold verifiability.

4.1 Our concrete scheme

In this section, we use Waters' signature [33] and Shamir secret sharing [29] to propose our concrete $(t, n)-$ MDVS scheme. For generality, messages can be considered of arbitrary lengths and a hash function $H_m: \{0, 1\}^* \to \{0, 1\}^{n_m}$ can be used to convert messages to the specific length n_m. The algorithms of our concrete scheme are as follows.

Setup: This PPT algorithm takes a security parameter k as input and outputs system parameters $params = \{G_1, G_2, q, g, e, g_1, m', m_1, \ldots, m_{n_m}\}$ in which G_1 and G_2 are two cyclic groups with prime order q of size k, g is a generator of G_1 and $e: G_1 \times G_1 \to G_2$ is an admissible bilinear pairing. Other parameters are random elements of G_1, i.e. $g_1, m', m_1, , m_{n_m} \in_R G_1$.

SKG: This PPT algorithm on input *params* picks a random $x_s \in_R Z_q^*$ as the private key of the signer Sk_s and computes the corresponding public key as $Pk_s = g^{x_s}$, then outputs $(Sk_s, Pk_s) = (x_s, g^{x_s})$.

VKG: This PPT algorithm on inputs *params* and the number of designated verifiers n picks a random element $x_i \in_R Z_q^*$ as the private key of the ith verifier, Sk_{v_i}, and computes the corresponding public key as $Pk_{v_i} = g^{x_i}$, then outputs $(Sk_{v_i}, Pk_{v_i}) = (x_i, g^{x_i})$ for $i = 1, 2, \ldots, n$.

Furthermore, in this phase, n designated verifiers run a $(t, n)-$ Shamir secret sharing [29] between themselves in order to obtain their shares of other verifiers' secret keys. This secret sharing is performed as follows:

- v_i $(i = 1, 2, \ldots, n)$ generates a polynomial $f_i(x)$ of degree $(t - 1)$ as follows:

$$f_i(x) = a_{i0} + a_{i1}x + a_{i2}x^2 + \cdots + a_{i(t-1)}x^{(t-1)},$$

  where $a_{i0} = Sk_{v_i} = x_i$, and $a_{ij} \in_R Z_q^*$ for $j = 1, 2, \ldots, (t - 1)$.
- v_i broadcasts $B_{ij} = g^{a_{ij}}$ for $j = 0, 1, \ldots (t - 1)$.
- v_i computes the v_k's share of his secret key as $s_{ik} = f_i(k)$ and sends it to v_k $(k = 1, 2, \ldots, n)$.
- Upon receiving s_{ik} from v_i, v_k verifies the correctness of his share by checking whether the equality $g^{s_{ik}} = \prod_{j=0}^{t-1} B_{ij}^{k^j}$ holds or not. If the equality does not hold, v_k requests from v_i to send him his share again.
- v_k computes his total share as $s_k = \sum_{i=1}^{n} s_{ik}$. (Note that by these assignments we have $\sum_{i=1}^{n} Sk_{v_i} = \sum_{k=1}^{t} \lambda_k s_k$, where λ_ks $(k = 1, 2, \ldots, t)$ are Lagrange coefficients, i.e. $\lambda_k = \prod_{i \in A \setminus \{k\}} (i/(i - k))$, where $A = \{1, 2, \ldots, t\}$).

DSign: Let $m[\ell]$ denotes the ℓth bit of the message m of length n_m. Define $\mathscr{M} \subseteq \{1, 2, \ldots, n_m\}$ to be the set of indices such that $m[\ell] = 1$. The signer, with the private key Sk_s, selects a random $r \in_R Z_q^*$ and computes Waters' signature as follows [33]:

$$\sigma' = (\sigma_1', \sigma_2') = \left(g_1^{Sk_s}\Big(m' \prod_{\ell \in \mathscr{M}} m_\ell\Big)^r, g^r\right).$$

Then, the signer sets $\sigma_2 = \sigma_2' = g^r$ and computes $\sigma_1 = e(\sigma_1', \prod_{i=1}^{n} Pk_{v_i})$, where Pk_{v_i}, $(i = 1, 2, \ldots, n)$, is the public key of the ith designated verifier. The signer outputs $\sigma = (\sigma_1, \sigma_2)$ as her designated signature for n designated verifiers.

DVer: In this phase, on inputs a message/designated signature pair $(m, \sigma = (\sigma_1, \sigma_2))$, every t members of the set of n designated verifiers are able to verify the validity of the signature, cooperatively. Without the loss of generality, let $v_1, v_2, \ldots, v_t$, $(1 < t < n)$, be the t verifiers who cooperate to check the signature. These t verifiers run the following stages in order to verify the validity of $\sigma = (\sigma_1, \sigma_2)$:

- Initially, each $v_k(1 \ \le k \ \le t)$ computes

$$\Phi = e(g_1, Pk_s)e\left(\left(m' \prod_{\ell \in \mathscr{M}} m_\ell\right), \sigma_2\right),$$

  where Pk_s is the public key of the signer.
- Every v_k, $(1 \le k \le t)$, computes $\Psi_k = \Phi^{s_k}$, where $s_k = \sum_{i=1}^{n} s_{ik}$ as described in the VKG phase. Then v_k broadcasts $\Psi_k = \Phi^{s_k}$ to other $t - 1$ verifiers who are cooperating to verify the signature, i.e. v_j, $(1 \le j \le t, j \ne k)$.
- v_j can verify the correctness of the received share of v_k, i.e. Ψ_k, by checking the following equality:

$$e(\Psi_k, g) = e(\Phi, g^{s_k}). \qquad (2)$$

If (3) does not hold, v_j requests from v_k to send him his share again. Note that Ψ_k, g and Φ are known to v_j and v_j can also compute g^{s_k} as follows:

$$g^{s_k} = g^{\sum_{i=1}^{n} s_{ik}} = \prod_{i=1}^{n} g^{s_{ik}} = \prod_{i=1}^{n} \prod_{j=1}^{t-1} B_{ij}^{k^j},$$

where B_{ij}s (for $i = 1, 2, \ldots, n$ and $j = 0, 1, \ldots (t - 1)$) were broadcasted in the second stage of the VKG phase.

- The signature is accepted if and only if the equation

$$\prod_{1}^{t} \Psi_k^{\lambda_k} = \sigma_1, \qquad (3)$$

  holds, where λ_ks $(k = 1, 2, \ldots, t)$ are Lagrange coefficients.

4.2 Security analysis

As mentioned in Section 3, correctness, unforgeability, non-transferability (source hiding) and threshold verifiability are four basic requirements for a $(t, n)-$ MDVS scheme. In this subsection, we analyse these properties of our proposed scheme.

Lemma 1: Correctness is satisfied in our $(t, n)-$ MDVS scheme.

Proof: In the DVer phase, t verifiers check whether the equality (3) (i.e. $\prod_1^t \Psi_k^{\lambda_k} = \sigma_1$) holds or not to verify the signature. We have

$$\begin{aligned}\prod_{1}^{t}\Psi_k^{\lambda_k} &= \Phi^{\Sigma_{k=1}^{t}\lambda_k s_k}\\ &= \Phi^{\Sigma_{i=1}^{n} Sk_{v_i}}\\ &= e(g_1, Pk_s)^{\Sigma_{i=1}^{n} Sk_{v_i}} e\left(\left(m' \prod_{\ell \in \mathscr{M}} m_\ell\right), \sigma_2\right)^{\Sigma_{i=1}^{n} Sk_{v_i}}\\ &= e(g_1, g^{Sk_s})^{\Sigma_{i=1}^{n} Sk_{v_i}} e\left(\left(m' \prod_{\ell \in \mathscr{M}} m_\ell\right), g^r\right)^{\Sigma_{i=1}^{n} Sk_{v_i}}\\ &= e(g_1^{Sk_s}, g^{\Sigma_{i=1}^{n} Sk_{v_i}}) e\left(\left(m' \prod_{\ell \in \mathscr{M}} m_\ell\right)^r, g^{\Sigma_{i=1}^{n} Sk_{v_i}}\right)\\ &= e\left(g_1^{Sk_s}\left(m' \prod_{\ell \in \mathscr{M}} m_\ell\right)^r, g^{\Sigma_{i=1}^{n} Sk_{v_i}}\right)\\ &= e\left(\sigma'_1, \prod_{i=1}^{n} Pk_{v_i}\right)\\ &= \sigma_1 .\end{aligned}$$

Hence, if $\sigma = (\sigma_1, \sigma_2)$ is a valid designated signature on m, then the output of $DVer((m, \sigma), params, Pk_s, \mathscr{V}, \mathscr{S}^t)$ is 1 for all $\mathscr{S}^t$s. In other words, $\sigma = (\sigma_1, \sigma_2)$ passes the DVer phase which is performed by the cooperation of every t verifiers of the set of all verifiers. □

Theorem 1: The proposed $(t, n)-$ MDVS scheme is $(t'', \varepsilon'', q_S, q_V)-$ unforgeable, assuming that $(t', \varepsilon')-$ GBDH assumption holds in (G_1, G_2), where

$$\varepsilon' \geq \frac{\varepsilon''}{4q_S(n_m + 1)},$$

$$t' \leq t'' + (4q_S + 5q_V + 1)T_{e1} + T_{e2} + (q_S + q_V + 1)T_p,$$

where t'' is the required time for $\mathscr{A}$ to forge a signature, T_{e1} and T_{e2} denote the time for computing an exponentiation in G_1 and G_2, respectively, and T_p is the time for a pairing computation in (G_1, G_2).

Proof: Since Waters presented his ID-based encryption and signature scheme in the standard model in 2001 [33], many researchers have used his ideas to present and prove different encryption and signature schemes in the standard model [19]. We also use the Waters' techniques to prove the unforgeability of our scheme. Our method of proof is similar to the method presented in [19] with some differences in details.

Suppose that there exists an adversary $\mathscr{A}$ who can $(t'', \varepsilon'', q_S, q_V)$ break the scheme by running the unforgeability game according to Definition 2. By this assumption, we can construct an algorithm $\mathscr{B}$ which can solve a GBDH problem in time at most t' with probability at least ε' by using $\mathscr{A}$ as a subroutine.

A random GBDH challenge $g, g^a, g^b, g^c \in G_1$ is given to $\mathscr{B}$ and $\mathscr{B}$ tries to calculate $e(g, g)^{abc} \in G_2$ with the help of the DBDH oracle $\mathscr{O}_{\text{DBDH}}$. In order to solve this problem, $\mathscr{B}$ runs $\mathscr{A}$ as a subroutine. $\mathscr{B}$ plays the unforgeability with $\mathscr{A}$ and simulates $\mathscr{C}$ and all oracle accesses for $\mathscr{A}$ in this game, as follows:

- *Setup:* $\mathscr{B}$ sets an integer $l_m = 2q_S$ and chooses an integer $k_m \in \{0, 1, \ldots, n_m\}$ (n_m is the length of the message). Assume that $l_m(n_m + 1) < q$ and as a result $0 \leq k_m l_m < q$. $\mathscr{B}$ also randomly selects $x', x_1, \ldots, x_{n_m} \in_R Z_{l_m}$ and $y', y_1, \ldots, y_{n_m} \in_R Z_q$. These values are kept internal to $\mathscr{B}$. In order to follow the proof more easily, define two following functions:

$$J(m) = x' + \sum_{\ell \in \mathscr{M}} x_\ell - k_m l_m, \quad K(m) = y' + \sum_{\ell \in \mathscr{M}} y_\ell,$$

where for a message m, $\mathscr{M} \subseteq \{1, 2, \ldots, n_m\}$ is the set of indices such that $m[\ell] = 1$. Then $\mathscr{B}$ assigns the public key of the signer, the public keys of designated verifiers and other unknown system parameters as follows (Note that g^a, g^b, g^c are the inputs of the GBDH problem which $\mathscr{B}$ is trying to solve it.):

- $\mathscr{B}$ assigns the public key of the signer as $Pk_s = g^a$.
- $\mathscr{B}$ selects random numbers $d_i \in_R Z_q^*$ for $i = 1, 2, \ldots, n$ and sets $Pk_{v_i} = (g^b)^{d_i}$ as n designated verifiers' public keys.
- $\mathscr{B}$ sets $g_1 = g^c$.
- $\mathscr{B}$ assigns $m' = g_1^{x' - k_m l_m} g^{y'}$ and $m_j = g_1^{x_j} g^{y_j}$ for $j = 1, 2, \ldots, n_m$. By this assignment, for any message m we have

$$m' \prod_{\ell \in \mathscr{M}} m_\ell = g_1^{J(m)} g^{K(m)}.$$

Afterwards, $\mathscr{B}$ returns Pk_s, Pk_{v_i} (for $i = 1, 2, \ldots, n$) and $params = \{G_1, G_2, q, g, e, g_1, m', m_1, \ldots, m_{n_m}\}$ to $\mathscr{A}$. From the perspective of $\mathscr{A}$, all distributions are identical to those in the real world.

- *Oracle accesses:* $\mathscr{A}$ has access to the $\mathscr{O}_{\text{Sign}}$ and $\mathscr{O}_{\text{Ver}}$ oracles as mentioned in Section 3.3. $\mathscr{B}$ plays the role of theses oracles, i.e. when $\mathscr{A}$ inputs its queries to these oracles, $\mathscr{B}$ will generate the corresponding outputs for $\mathscr{A}$ as follows:

○ $\mathscr{O}_{\text{Sign}}$: On input a message m, this oracle must output a valid designated signature σ on m. When $\mathscr{A}$ gives $\mathscr{O}_{\text{Sign}}$ the message m as input, $\mathscr{B}$ must generate a valid $\sigma = (\sigma_1, \sigma_2)$ without the knowledge of the private key of the signer and designated verifiers (Note that $\mathscr{B}$ does not know a, b, c). To produce $\sigma = (\sigma_1, \sigma_2)$, $\mathscr{B}$ acts as follows:

- If $J(m) = 0 \bmod q$, $\mathscr{B}$ aborts and reports a failure.
- If $J(m) \neq 0 \bmod q$, $\mathscr{B}$ randomly selects $r \in_R Z_q^*$. Then $\mathscr{B}$ computes (see (4)) .

Noting (4) and defining $\tilde{r} = r - (a/J(m))$, we have

$$\begin{aligned}\sigma_1 &= e\left(Pk_s^{(-K(m)/J(m))}\left(m' \prod_{\ell \in \mathscr{M}} m_\ell\right)^r, \prod_{i=1}^{n} Pk_{v_i}\right)\\ &= e\left(g^{-a(K(m)/J(m))}(g_1^{J(m)} g^{K(m)})^r, \prod_{i=1}^{n} Pk_{v_i}\right)\\ &= e\left((g_1^{J(m)} g^{K(m)})^{(-a/J(m))} g_1^a (g_1^{J(m)} g^{K(m)})^r, \prod_{i=1}^{n} Pk_{v_i}\right)\\ &= e\left(g_1^a (g_1^{J(m)} g^{K(m)})^{r-(a/J(m))}, \prod_{i=1}^{n} Pk_{v_i}\right)\\ &= e\left(g_1^a \left(m' \prod_{\ell \in \mathscr{M}} m_\ell\right)^{\tilde{r}}, \prod_{i=1}^{n} Pk_{v_i}\right),\end{aligned}$$

and also

$$\sigma_2 = g^r Pk_s^{(-1/J(m))} = g^r g^{(-a/J(m))} = g^{r-(a/J(m))} = g^{\tilde{r}} .$$

Hence, the construction $\sigma = (\sigma_1, \sigma_2)$ computed by (4) is a valid designated signature on m.

$$\sigma = \left(e\left(Pk_s^{(-K(m)/J(m))}\left(m' \prod_{\ell \in \mathscr{M}} m_\ell\right)^r, \prod_{i=1}^{n} Pk_{v_i}\right), g^r Pk_s^{(-1/J(m))}\right). \tag{4}$$

○ $\mathscr{O}_{\text{Ver}}$: On input a message/designated signature pair $(m, \sigma = (\sigma_1, \sigma_2))$ this oracle must output 1 if σ is a valid designated signature on m and 0 otherwise. When $\mathscr{A}$ gives $\mathscr{O}_{\text{Ver}}$ the message/designated signature pair $(m, \sigma = (\sigma_1, \sigma_2))$ as input, $\mathscr{B}$ must verify the validity of σ without the knowledge of the private key of the signer and the designated verifiers (Note that $\mathscr{B}$ does not know $a, b, c.$). To verify σ, $\mathscr{B}$ acts as follows:

- If $J(m) = 0 \bmod q$, $\mathscr{B}$ submits

$$\left(g, g^a, \prod_{i=1}^{n} Pk_{v_i}, g^c, \frac{\sigma_1}{e\left(\sigma_2^{K(m)}, \prod_{i=1}^{n} Pk_{v_i}\right)}\right), \tag{5}$$

to the DBDH oracle $\mathscr{O}_{\text{DBDH}}$ (Note that $\mathscr{B}$ is trying to solve a GBDH problem and has access to the $\mathscr{O}_{\text{DBDH}}$.). Then $\mathscr{B}$ outputs 1 to $\mathscr{A}$ if the output of $\mathscr{O}_{\text{DBDH}}$ is 1 and 0 otherwise. It can be easily shown that if $(m, \sigma = (\sigma_1, \sigma_2))$ is a valid designated signature on m, then the tuple in (5) is a valid BDH tuple, as we have

$$\begin{aligned}\frac{\sigma_1}{e\left(\sigma_2^{K(m)}, \prod_{i=1}^{n} Pk_{v_i}\right)} &= \frac{e\left(g_1^a\left(m' \prod_{\ell \in \mathscr{M}} m_\ell\right)^r, \prod_{i=1}^{n} Pk_{v_i}\right)}{e\left(\sigma_2^{K(m)}, \prod_{i=1}^{n} Pk_{v_i}\right)} \\ &= \frac{e\left(g^{ca}\left(g^{K(m)}\right)^r, \prod_{i=1}^{n} Pk_{v_i}\right)}{e\left(g^{rK(m)}, \prod_{i=1}^{n} Pk_{v_i}\right)} \\ &= e\left(g^{ca}, \prod_{i=1}^{n} Pk_{v_i}\right).\end{aligned}$$

- If $J(m) \neq 0 \bmod q$, $\mathscr{B}$ can generate a valid designated signature $\hat{\sigma} = (\hat{\sigma}_1, \hat{\sigma}_2)$ on m as he generates the output of $\mathscr{O}_{\text{Sign}}$. Afterwards, $\mathscr{B}$ submits

$$\left(g, \prod_{i=1}^{n} Pk_{v_i}, m' \prod_{\ell \in \mathscr{M}} m_\ell, \frac{\sigma_2}{\hat{\sigma}_2}, \frac{\sigma_1}{\hat{\sigma}_1}\right), \tag{6}$$

to the DBDH oracle $\mathscr{O}_{\text{DBDH}}$. Then $\mathscr{B}$ outputs 1 to $\mathscr{A}$ if the output of $\mathscr{O}_{\text{DBDH}}$ is 1 and 0 otherwise. It can be easily shown that if $\sigma = (\sigma_1, \sigma_2)$ is a valid designated signature on m, then the tuple in (6) is a valid BDH tuple. Note that if $\sigma = (\sigma_1, \sigma_2)$ is a valid designated signature, we have $\sigma_2 = g^r$ and also according to (3)

$$\begin{aligned}\sigma_1 &= \prod_{k=1}^{t} \Psi_k^{\lambda_k} = \Phi^{\sum_{k=1}^{t} \lambda_k s_k} = \Phi^{\sum_{i=1}^{n} Sk_{v_i}} \\ &= e(g_1, Pk_s)^{\sum_{i=1}^{n} Sk_{v_i}} e\left(m' \prod_{\ell \in \mathscr{M}} m_\ell, \sigma_2\right)^{\sum_{i=1}^{n} Sk_{v_i}}.\end{aligned} \tag{7}$$

Similarly, since $\hat{\sigma} = (\hat{\sigma}_1, \hat{\sigma}_2)$ is a valid designated signature, we have $\hat{\sigma}_2 = g^{\hat{r}}$ and

$$\begin{aligned}\hat{\sigma}_1 &= \prod_{k=1}^{t} \hat{\Psi}_k^{\lambda_k} = \hat{\Phi}^{\sum_{k=1}^{t} \lambda_k s_k} = \hat{\Phi}^{\sum_{i=1}^{n} Sk_{v_i}} \\ &= e(g_1, Pk_s)^{\sum_{i=1}^{n} Sk_{v_i}} e\left(m' \prod_{\ell \in \mathscr{M}} m_\ell, \hat{\sigma}_2\right)^{\sum_{i=1}^{n} Sk_{v_i}}.\end{aligned} \tag{8}$$

According to (7) and (8), we have

$$\begin{aligned}\frac{\sigma_1}{\hat{\sigma}_1} &= e\left(m' \prod_{\ell \in \mathscr{M}} m_\ell, \frac{\sigma_2}{\hat{\sigma}_2}\right)^{\sum_{i=1}^{n} Sk_{v_i}} \\ &= e\left(g^{cJ(m)} g^{K(m)}, g^{r-\hat{r}}\right)^{\sum_{i=1}^{n} Sk_{v_i}}.\end{aligned} \tag{9}$$

According to (9) and noting that $(\sigma_2 / \hat{\sigma}_2) = g^{r-\hat{r}}$, $\prod_{i=1}^{n} Pk_{v_i} = g^{\sum_{i=1}^{n} Sk_{v_i}}$ and $m' \prod_{\ell \in \mathscr{M}} m_\ell = g^{cJ(m)} g^{K(m)}$, the tuple in (6) is a valid BDH tuple.

- *Forgery:* Suppose that $\mathscr{A}$ forges a signature $\sigma^* = (\sigma_1^*, \sigma_2^*)$ on message m^* (Remember that $\mathscr{B}$ is trying to solve a GBDH problem.). Since $\mathscr{A}$ creates $\sigma^* = (\sigma_1^*, \sigma_2^*)$, $\mathscr{B}$ acts as follows:
- If $J(m^*) \neq 0 \bmod q$, $\mathscr{B}$ aborts and reports a failure.
- If $J(m^*) = 0 \bmod q$, $\mathscr{B}$ can solve the GBDH problem by obtaining $e(g, g)^{abc}$ as follows:

$$e(g, g)^{abc} = \left(\frac{\sigma_1^*}{e\left(\sigma_2^{*\,K(m^*)}, \prod_{i=1}^{n} Pk_{v_i}\right)}\right)^{\left(\sum_{i=1}^{n} d_i\right)^{-1}}. \tag{10}$$

It is easy to check that (10) holds, if $\sigma^* = (\sigma_1^*, \sigma_2^*)$ is a valid signature.

Time analysis: Noting the above descriptions we can see that $\mathscr{B}$ needs a time $t' \leq t'' + (4q_S + 5q_V + 1)T_{e1} + T_{e2} + (q_S + q_V + 1)T_p$, for running the game, where t'' is the required time for $\mathscr{A}$ to forge a signature, T_{e1} and T_{e2} denote the time for an exponentiation in G_1 and G_2, respectively, and T_p is the time for a pairing in (G_1, G_2).

Probability analysis: In order to analyse the success probability of $\mathscr{B}$, we consider events in which $\mathscr{B}$ will not abort. $\mathscr{B}$ will not abort if both the two following events happen [19]:

- E_1: $J(m) \neq 0 \bmod q$ for all queries from $\mathscr{O}_{\text{Sign}}$. Let E_{1i} denotes the event that $J(m) \neq 0 \bmod q$ in the ith query from $\mathscr{O}_{\text{Sign}}$, hence $E_1 = \bigcap_{i=1}^{q_S} E_{1i}$.
- E_2: $J(m^*) = 0 \bmod q$.

It is easy to see that [19]

$$\Pr\,[J(m) = 0,\ \bmod q] = \frac{1}{(l_m - 1)(n_m + 1) + 1}. \tag{11}$$

By defining two events E_1 and E_2 as mentioned, we have

$$\text{Success Probability of } \mathscr{B} = \varepsilon' \geq \varepsilon'' \cdot \Pr[E_1 \cap E_2], \tag{12}$$

in which ε'' is the least success probability of $\mathscr{A}$ to forge a signature. Noting (11) and that E_1 and E_2 are independent events, we have

$$\begin{aligned}\Pr[E_1 \cap E_2] &= \Pr\,[E_1]\,\Pr\,[E_2] \\ &= \Pr[\bigcap_{i=1}^{q_S} E_{1i}]\,\Pr\,[E_2] \\ &= (1 - \Pr[\bigcup_{i=1}^{q_S} \bar{E}_{1i}])\left(\frac{1}{(l_m - 1)(n_m + 1) + 1}\right) \\ &\geq \left(1 - \frac{q_S}{(l_m - 1)(n_m + 1) + 1}\right)\left(\frac{1}{(l_m - 1)(n_m + 1) + 1}\right) \\ &\geq \left(1 - \frac{q_S}{l_m}\right)\left(\frac{1}{l_m(n_m + 1)}\right) = \frac{1}{4q_S(n_m + 1)},\end{aligned} \tag{13}$$

where the rightmost equality is implied from $l_m = 2q_S$.

Noting (12) and (13), we have

$$\text{Success Probability of } \mathscr{B} = \varepsilon' \geq \frac{\varepsilon''}{4q_S(n_m + 1)},$$

as the final result. □

Theorem 2: The proposed $(t, n)-$ MDVS scheme is unconditionally non-transferable.

Proof: Suppose that $\sigma_0 = (\sigma_{0_1}, \sigma_{0_2})$ is a designated signature on m which is produced by the signer and $\sigma_1 = (\sigma_{1_1}, \sigma_{1_2})$ is a designated signature on m which is produced by the transcript simulator (TS). According to Definition 3, we must prove that the value of (1) is negligible.

In order to generate σ_0, the signer, with the private key Sk_s, selects a random element $r_0 \in_R Z_q^*$ and computes $\sigma_0 = (\sigma_{0_1}, \sigma_{0_2})$ as follows:

$$\sigma_0 = \left(e\left(g_1^{Sk_s}\left(m' \prod_{\ell \in \mathscr{M}} m_\ell \right)^{r_0}, \prod_{i=1}^{n} Pk_{v_i} \right), g^{r_0} \right). \tag{14}$$

In order to generate σ_1, TS picks a random $r_1 \in_R Z_q^*$ and computes $\sigma_1 = (\sigma_{1_1}, \sigma_{1_2})$ as follows:

$$\sigma_1 = \left(e(g_1, Pk_s)^{\Sigma_{i=1}^{n} Sk_{v_i}} e\left(m' \prod_{\ell \in \mathscr{M}} m_\ell, g^{r_1} \right)^{\Sigma_{i=1}^{n} Sk_{v_i}}, g^{r_1} \right). \tag{15}$$

It is easy to see that σ_0 and σ_1 have the same distributions and hence they are indistinguishable. Suppose that a challenger $\mathscr{C}$ selects a random element $r^* \in_R Z_q^*$ and sets $\sigma_2^* = g^{r^*}$, then picks a $b \in_R \{0, 1\}$ by flipping a fair coin and sets σ_1^* as follows: (see (16)) .

$$\sigma_1^* = \begin{cases} e\left(g_1^{Sk_s}\left(m' \prod_{\ell \in \mathscr{M}} m_\ell \right)^{r^*}, \prod_{i=1}^{n} Pk_{v_i} \right) & \text{if } b = 0 \\ e(g_1, Pk_s)^{\Sigma_{i=1}^{n} Sk_{v_i}} e\left(m' \prod_{\ell \in \mathscr{M}} m_\ell, g^{r^*} \right)^{\Sigma_{i=1}^{n} Sk_{v_i}} & \text{if } b = 1 \end{cases}. \tag{16}$$

Noting (14), (15) and (16) we have

$$\Pr\left[\sigma^* = \sigma_0\right] = \Pr\begin{bmatrix} \sigma_1^* = \sigma_{0_1} \\ \sigma_2^* = \sigma_{0_2} \end{bmatrix} = \Pr\left[r^* = r_0\right] = \frac{1}{q-1},$$

$$\Pr\left[\sigma^* = \sigma_1\right] = \Pr\begin{bmatrix} \sigma_1^* = \sigma_{1_1} \\ \sigma_2^* = \sigma_{1_2} \end{bmatrix} = \Pr\left[r^* = r_1\right] = \frac{1}{q-1}$$

Therefore, the distributions of σ_0 and σ_1 are identical and a distinguisher $\mathscr{D}$ cannot distinguish whether the signature is created by the signer or by TS. As a result, the signature is unconditionally non-transferable. □

Threshold verifiability: In this part, the threshold verifiability of the proposed scheme is analysed according to Definition 4. In order to prove this property, first three following lemmas are considered.

Lemma 2: The knowledge of $\sum_{i=1}^{n} Sk_{v_i}$ is necessary (by the GBDH assumption in (G_1, G_2)) and sufficient (unconditionally) to verify the designated signature.

Proof: In order to prove Lemma 2, we will consider two following parts. In part 1 the sufficiency and in part 2 the necessity will be proved.

- *Part 1 (Sufficiency):* By receiving a message/designated signature pair $(m, \sigma = (\sigma_1, \sigma_2))$, everyone who knows $\sum_{i=1}^{n} Sk_{v_i}$, is able to verify the validity of the signature by checking whether the following equality holds:

$$\sigma_1 = e(g_1, Pk_s)^{\Sigma_{i=1}^{n} Sk_{v_i}} e\left(m' \prod_{\ell \in \mathscr{M}} m_\ell, \sigma_2 \right)^{\Sigma_{i=1}^{n} Sk_{v_i}}.$$

  Hence, the knowledge of $\sum_{i=1}^{n} Sk_{v_i}$ is sufficient to verify the designated signature.
- *Part 2 (Necessity):* In this part, we show that if there exists an adversary $\mathscr{A}$ who can verify a signature without the knowledge of $\sum_{i=1}^{n} Sk_{v_i}$, with at most q_S and q_V signature and verification queries, in time at most t'' and with probability at least ε'', then there exists an algorithm $\mathscr{B}$ which can solve a GBDH problem in (G_1, G_2) in time at most t' and with probability at least ε' by using $\mathscr{A}$ as a sub-routine, where

$$\varepsilon' \geq \frac{\varepsilon''}{4 q_S (n_m + 1)},$$

$$t' \leq t'' + (4 q_S + 5 q_V + 1) T_{e1} + T_{e2} + (q_S + q_V + 1) T_p .$$

Suppose that there exists an adversary $\mathscr{A}$ who can verify a signature without the knowledge of $\sum_{i=1}^{n} Sk_{v_i}$, with at most q_S and q_V signature and verification queries, in time at most t'' and with probability at least ε''. We can construct an algorithm $\mathscr{B}$ which can solve a GBDH problem in (G_1, G_2) in time at most t' and with probability at least ε' by using $\mathscr{A}$ as a sub-routine.

A random GBDH challenge $g, g^a, g^b, g^c \in G_1$ is given to $\mathscr{B}$, and $\mathscr{B}$ tries to calculate $e(g, g)^{abc} \in G_2$ with the help of the DBDH oracle $\mathscr{O}_{\text{DBDH}}$. In order to solve this problem, $\mathscr{B}$ runs $\mathscr{A}$ as a sub-routine. $\mathscr{B}$ plays the following game with $\mathscr{A}$:

- *Setup:* $\mathscr{B}$ selects $l_m, k_m, x', x_1, \ldots, x_{n_m}, y', y_1, \ldots, y_{n_m}$ as mentioned in the setup phase of the proof of Theorem 1. Similarly, define two following functions:

$$J(m) = x' + \sum_{\ell \in \mathscr{M}} x_\ell - k_m l_m, \quad K(m) = y' + \sum_{\ell \in \mathscr{M}} y_\ell,$$

  Then $\mathscr{B}$ assigns $Pk_s = g^a$, $g_1 = g^c$, $m' = g_1^{x' - k_m l_m} g^{y'}$ and $m_j = g_1^{x_j} g^{y_j}$ for $j = 1, 2, \ldots, n_m$. $\mathscr{B}$ also selects random numbers $d_i \in_R Z_q^*$ for $i = 1, 2, \ldots, n$ and sets $Pk_{v_i} = (g^b)^{d_i}$. Afterwards, $\mathscr{B}$ returns Pk_s, Pk_{v_i} (for $i = 1, 2, \ldots, n$) and

$$params = \{G_1, G_2, q, g, e, g_1, m', m_1, \ldots, m_{n_m}\} \tag{17}$$

  to $\mathscr{A}$. From the perspective of $\mathscr{A}$, all distributions are identical to those in the real world. Note that by the mentioned assignments, we have $\sum_{i=1}^{n} Sk_{v_i} = b \sum_{i=1}^{n} d_i$ and therefore, neither $\mathscr{B}$ nor $\mathscr{A}$ can compute $\sum_{i=1}^{n} Sk_{v_i}$.
- *Oracle accesses:* Suppose that $\mathscr{A}$ is trying to verify a designated signature $\sigma^* = (\sigma_1^*, \sigma_2^*)$ on a message m^*. $\mathscr{A}$ has access to the $\mathscr{O}_{\text{Sign}}$ and $\mathscr{O}_{\text{Ver}}$ oracles and $\mathscr{B}$ plays the role of these oracles. $\mathscr{B}$ should answer $\mathscr{A}$'s queries without the knowledge of Sk_s and $\sum_{i=1}^{n} Sk_{v_i}$. When $\mathscr{A}$ inputs its queries to these oracles, $\mathscr{B}$ will generate the corresponding outputs for $\mathscr{A}$ as mentioned in the proof of Theorem 1.

  Note that $\mathscr{A}$ is not only allowed to send a request to the $\mathscr{O}_{\text{Ver}}$ for the verification of $\sigma^* = (\sigma_1^*, \sigma_2^*)$ on m^*, but also she is not allowed to send a request to the $\mathscr{O}_{\text{Ver}}$ for the verification of any other signature $\sigma' = (\sigma_1', \sigma_2')$ on m^*. Since by receiving the message/designated signature pair $(m^*, \sigma^* = (\sigma_1^*, \sigma_2^*))$, even if $\mathscr{A}$ is not allowed to send $(m^*, \sigma^* = (\sigma_1^*, \sigma_2^*))$ to the $\mathscr{O}_{\text{Ver}}$, she can pick a random $r' \in_R Z_q^*$ and calculate another signature $\sigma' = (\sigma_1', \sigma_2')$ on m^* as follows:

$$\sigma' = \begin{cases} \sigma'_1 = \sigma_1^* \,.\, e\left(\left(m' \prod_{\ell \in \mathscr{M}} m_\ell \right)^{r'}, \prod_{i=1}^{n} Pk_{v_i} \right) \\ \sigma'_2 = \sigma_2^* \,.\, g^{r'} \end{cases}. \tag{18}$$

Note that if $\sigma^* = (\sigma_1^*, \sigma_2^*)$ is a valid signature on m^*, i.e.

$$\sigma^* = \begin{cases} \sigma_1^* = e\left(g_1^{Sk_s}\left(m' \prod_{\ell \in \mathscr{M}} m_\ell\right)^{r^*}, \prod_{i=1}^{n} Pk_{v_i}\right) \\ \sigma_2^* = g^{r^*} \end{cases}, \quad (19)$$

for a random $r^* \in_R Z_q^*$, then $\sigma' = (\sigma_1', \sigma_2')$ is also a valid signature on m^* for a random $r^* + r' \in_R Z_q^*$, since noting (18) and (19) we have

$$\sigma' = \begin{cases} \sigma'_1 = e\left(g_1^{Sk_s}\left(m' \prod_{\ell \in \mathscr{M}} m_\ell\right)^{r^* + r'}, \prod_{i=1}^{n} Pk_{v_i}\right) \\ \sigma'_2 = g^{r^* + r'} \end{cases}.$$

Therefore, if $\mathscr{A}$ is allowed to send $(m^*, \sigma' = (\sigma_1', \sigma_2'))$ to the $\mathscr{O}_{\text{Ver}}$, she can imply that $(m^*, \sigma^* = (\sigma_1^*, \sigma_2^*))$ is valid if the output of $\mathscr{O}_{\text{Ver}}$ is 1 and invalid otherwise. As a result, $\mathscr{A}$ is not allowed to send a request to the $\mathscr{O}_{\text{Ver}}$ for the verification of any signature on m^*, but she is allowed to send a request to the $\mathscr{O}_{\text{Ver}}$ for the verification of signatures on other messages and $\mathscr{B}$ will respond to her as mentioned in the proof of Theorem 1.

- *Verification:* Suppose that the signature $\sigma^* = (\sigma_1^*, \sigma_2^*)$ on message m^* is verified and accepted by $\mathscr{A}$ (Remember that $\mathscr{B}$ is trying to solve a GBDH problem.). As $\mathscr{A}$ accepts $\sigma^* = (\sigma_1^*, \sigma_2^*)$, $\mathscr{B}$ acts as follows:

  - If $J(m^*) \neq 0 \bmod q$, $\mathscr{B}$ aborts and reports a failure.
  - If $J(m^*) = 0 \bmod q$, $\mathscr{B}$ can solve the GBDH problem by obtaining $e(g, g)^{abc}$ as follows:

$$e(g, g)^{abc} = \left(\frac{\sigma_1^*}{e({\sigma_2^*}^{K(m^*)}, \prod_{i=1}^{n} Pk_{v_i})}\right)^{(\sum_{i=1}^{n} d_i)^{-1}}. \quad (20)$$

It is easy to check that (20) holds, if $\sigma^* = (\sigma_1^*, \sigma_2^*)$ is a valid signature.

Time and probability analysis are similar to those in the proof of Theorem 1. □

Lemma 3: Every set of at least t members of n designated verifiers can verify a designated signature, cooperatively without revealing $\sum_{i=1}^{n} Sk_{v_i}$.

Proof: Consider the polynomial $F(x) = \sum_{i=1}^{n} f_i(x)$ of degree $t - 1$. Note that after the secret sharing mentioned in the VKG phase of the scheme, the kth verifier v_k $(k = 1, 2, \ldots, n)$ knows $F(k) = \sum_{i=1}^{n} f_i(k) = \sum_{i=1}^{n} s_{ik} = s_k$. As a result, every set of at least t members of n designated verifiers can compute the intercept of $F(x)$ (i.e. $F(0) = \sum_{i=1}^{n} Sk_{v_i}$) by Lagrange interpolation as $F(0) = \sum_{k=1}^{t} \lambda_k F(k)$. Hence, every set of at least t members of n designated verifiers have the necessary and sufficient condition mentioned in Lemma 2 (i.e. the knowledge of $\sum_{i=1}^{n} Sk_{v_i}$) to verify the signature, but as mentioned in the DVer phase of the scheme, they do not require to compute and reveal $\sum_{i=1}^{n} Sk_{v_i}$ in order to verify a signature and they are able to verify the signature by checking equality (3) without revealing $\sum_{i=1}^{n} Sk_{v_i}$. □

Lemma 4: There is no any set of less than t members of n designated verifiers who can verify a designated signature.

Proof: Note that since $F(x) = \sum_{i=1}^{n} f_i(x)$ is a polynomial of degree $t - 1$, the knowledge of the coordinates of at least t points of $F(x)$ is necessary to determine $F(x)$ and as a result its intercept $F(0) = \sum_{i=1}^{n} Sk_{v_i}$. No set of less than t members of n designated verifiers can obtain this necessary condition and as a result they are not able to compute $\sum_{i=1}^{n} Sk_{v_i}$, cooperatively. Therefore, no set of less than t members of n designated verifiers have the necessary condition mentioned in Lemma 2 (i.e. the knowledge of $\sum_{i=1}^{n} Sk_{v_i}$) in order to verify the signature. □

Theorem 3: The proposed $(t, n) -$ MDVS scheme is threshold verifiable, i.e. the signature can be verified by the cooperation of at least t designated verifiers.

Proof: The proof is implied directly from Lemmas 2–4. □

Remark 3: Note that in our proposed concrete $(t, n) -$ MDVS scheme, t verifiers can cooperate to generate an indistinguishable signature from that produced by the signer, as they can select a random $r \in_R Z_q^*$ and compute $\sigma_2 = g^r$ and

$$\sigma_1 = (e(g_1, Pk_s)e((m' \prod_{\ell \in \mathscr{M}} m_\ell), g^r))^{\sum_{k=1}^{t} \lambda_k s_k}. \quad (21)$$

As a result, our proposed scheme is applicable in scenarios in which the verifiers do not obtain any benefits from this malice (see Section 6 for an example scenario). Otherwise, an extra assumption must be considered that any set of at least t verifiers never cooperate for producing a signature for the others which should be reasonable according to the corresponding scenario. Totally, the applicability of a scheme depends on the corresponding scenario in which the scheme is applied (see Section 6 for more explanations).

5 Comparison

In Table 1, a comparison between MDVS schemes is provided. A few number of MDVS schemes have been proposed in the literature, till now [3, 7–23]. Since in each of [8, 14] two schemes have been proposed, we denote theme as [8]-I, [8]-II, [14]-I and [14]-II, respectively. As mentioned in Section 1, among these proposed MDVS schemes, some of the themes do not require any private keys of designated verifiers in the verification phase (case 1 of the first pattern). The proposed schemes in [3, 7, 12, 13] and the first proposed scheme in [14] (i.e. [14]-I) are in this category. As these schemes do not provide one of the main purposes of DVS schemes, namely to preserve the privacy of the signer, we do not consider them in Table 1. In this table:

- n denotes the number of designated verifiers.
- $|aG|$ denotes the binary length of a elements of the group G.
- T_{E1}, T_{E2}, T_P and T_M denote the required time for the computation of an exponentiation in G_1, the computation of an exponentiation in G_2, a pairing computation from G_1 to G_2 and a cryptographic multi-linear map from G_1 to G_2, respectively.
- T_{ENC}, T_{DEC}, T_{GMAC} and T_{VMAC} denote the required time for a symmetric encryption, a symmetric decryption, a MAC generation and a MAC verification, respectively.

The second column of Table 1 shows the signature size of the corresponding scheme. As shown, the signature size of our proposed scheme is fixed and this low communication cost is an important option in many scenarios.

The third and the fourth columns of Table 1 show the cost for the signer to produce a signature and the cost for each verifier to verify the signature. As shown, these costs are also fixed in our proposal.

The fifth column of Table 1 shows whether the security requirements (unforgeability and non-transferability) of the signature scheme are provable in the standard or random oracle model. As shown in Table 1, only the scheme in [19] is proposed in the standard model (without random oracles). As discussed in [34], the schemes in which their security requirements are proved in the ROM are not secure when the random oracles are replaced with the real-world primitives. Since we use the scheme in [19] as the base

Table 1 Comparison between MDVS schemes

Scheme	Signature size	Signing cost	Each Verifire's verification cost	Security model	Pattern	Threshold verifiability?
[8]-I	$\|nG_2\|$	$1T_{E1} + nT_P$	$1T_{E1} + 1T_P$	ROM	first	×
[9]	$\|(n+1)G_1\|$	$(n+1)T_{E1}$	$3T_{E1} + 1T_P$	—[a]	first	×
[10]	$\|(n+4)G_1\|$	$(n+5)T_{E1} + 1T_P$	$2T_{E1} + 4T_P$	—[a]	first	×
[23]	$\|nG_2\|$	$(n+3)T_{E1} + nT_P$	$1T_{E2} + 1T_P$	ROM	first	×
[11]	$\|(n+2)G_1\|$	$(n+4)T_{E1}$	$1T_{E1} + (n+2)T_P$	ROM	first	×
[21]	$\|2G_1\| + \|1\mathbb{Z}_q^*\|$	$5T_{E1} + 1T_{E2} + 1T_M$	$6T_{E1} + 1T_{E2} + 1T_M$	—[a]	first	×
[14]-II	$\|1\mathbb{Z}_q^*\| + \|(n+1)G_1\|$	$(n+2)T_{E1}$	$1T_{E1} + (2n)T_P$	ROM	first	×
[16]	$\|3G_1\| + \|2\mathbb{Z}_q^*\|$	$3T_{E1}$	$3T_{E1} + 2T_P$	ROM	first	×
[17][b]	$\|(n+3)G_1\| + \|2\mathbb{Z}_q^*\|$ $+F(n)$	$(n+6)T_{E1}$ $+1T_{ENC} + 1T_{GMAC}$	$2T_P + 3T_{E1}$ $+1T_{DEC} + 1T_{VMAC}$	ROM	first	×
[18]	$\|4\mathbb{Z}_q^*\| + \|nG\|$	$(n+3)T_E$	$(n+4)T_E$	ROM	first	×
[8]-II	$\|1G_2\|$	$1T_{E1} + 1T_P$	$1T_{E1} + (3n-2)T_P$	ROM	second	×
[19]	$\|1G_1\| + \|1G_2\|$	$3T_{E1} + 1T_P$	$1T_{E_2} + 2T_P$	standard	second	×
[20]	$\|1G_1\| + \|1G_2\|$	$3T_{E1} + 1T_P$	$(n+3)T_{E1} + (2n-1)T_P$	ROM	second	×
[15]	$\|1G_1\| + \|1G_2\|$	$2T_{E1} + 1T_P$	$1T_{E1} + 1T_P$	ROM	second	×
[22]	$\|2G_1\|$	$3T_{E1}$	$3T_{E1}$	ROM	second	×
ours	$\|1G_1\| + \|1G_2\|$	$3T_{E1} + 1T_P$	$1T_{E2} + 2T_P$	Standard	NEW	✓

[a] Security proofs are not provided for these schemes.

[b] In this scheme, a symmetric encryption scheme ENC and a MAC function are used in the signing algorithm and the output of the MAC is a function of n which is denoted as $F(n)$.

of our proposed (t, n)-MDVS scheme, our scheme is provable secure in the standard model, too.

The sixth column of Table 1 shows the pattern of the scheme. As mentioned before, each pattern can be useful according to the scenario in which the scheme is applied. In Section 6, we explain about scenarios in which our proposed pattern can be applicable.

6 Application

MDVS schemes (as an extension of DVS schemes) are useful in many applications such as e-voting, e-auction, fair exchange, cloud computing and so on. As mentioned before, the choice of the pattern depends on the corresponding scenario. Although both existing patterns are useful in many applications, there are situations in which neither of them are suitable. For example, consider a scenario in which a signer must be authenticated to a set of designated verifiers without disturbing her privacy with the following situations:

i. It is not desirable (for the signer or other designated verifiers) that an individual verifier verifies a signature.
ii. The time is important and the signature must be verified as soon as possible, but all designated verifiers may not be presented at the verification process time.

It is clear that neither of the existing patterns is appropriate for this scenario, since the first pattern does not satisfy case 1 and the second pattern does not satisfy case 2. In these situations, our proposed (t, n)-MDVS scheme can be useful.

Example: In [27], an m-healthcare cloud computing scenario under the malicious model is proposed in which the privacy of the patients is satisfied by a traditional DVS scheme. In this scenario, a patient (Alice) is authenticated by an individual authorised physician (Dr. Brown) in a cloud system by her DVS on her health information (HI). Dr. Brown can authenticate Alice by her DVS, but cannot transfer this conviction to any party (such as other physicians). After Alice is authenticated to Dr. Brown, he sends her some medical treatment (MT) according to her HI. Now, consider a situation in which Alice wants to obtain MT from multi (not an individual) authorised physicians, but she does not desire that anyone (except the set of all authorised physicians) be informed about her identity. In this scenario, an MDVS scheme can be used to authenticate Alice (as the signer) to a set of authorised physicians (as the designated verifiers). But which patterns of MDVS schemes are appropriate in this scenario? In order to select the most suitable pattern, note the following descriptions.

(i) Suppose that a forger forges a signature in this scenario. It is clear that the patient, Alice, does not bother in this case, since suppose that this malice is occurred and some MTs are sent to Alice, accordingly. As Alice knows that she has not created such signature, she can be aware that these MTs are illegal. However, forging a signature can bother the set of designated verifiers, since their time is lost for verifying a forged signature, preparing the corresponding MTs and sending them to Alice. All three patterns are proper in this sense, since all of them are unforgeable against an outsider forger.

(ii) In this scenario, any set of verifiers do not obtain any benefits from forging a signature for other verifiers; even if they can do this malice (as in our proposed concrete scheme in which a threshold number t of n designated verifiers can produce an indistinguishable signature from that created by Alice which is explained in Remark 3).

As a result, our proposed pattern is suitable in this sense as well as two other existing patterns.

(iii) The privacy of the signer must be preserved in this scenario, i.e. anyone except the set of the authorised physicians should not be able to convince about Alice's signature. This requirement is also satisfied in our proposed pattern as well as two other existing patterns by the non-transferability property.

(iv) Suppose that a malicious physician wants to reject a valid signature of Alice, since he/she does not have enough time to cooperate in processing HI of Alice and sending the corresponding MT. Neither of the two existing patterns is suitable for preventing a physician from this malice. In the first pattern, an individual physician can verify the signature and thus refuses from accepting a valid signature according to his/her benefits. In the second pattern, a malicious verifier can send an incorrect share for the verification to other verifiers. However, our pattern can prevent the verifiers from this malice, since every verifier can verify the correctness of the shares received from the others by checking equality (2). As a result, only our pattern is suitable in this sense.

(v) It is clear that in the mentioned scenario, the signature must be verified and the corresponding MTs must be sent to the patient as soon as possible. The second pattern is not suitable in this case, since the signature cannot be verified even if one of the n physicians is not present at the verification process time and one cannot leave the verification process to the future, while in our pattern, the presence of t verifier of all n verifiers is sufficient for running the verification process. Hence, the first pattern and our new pattern are suitable in this sense.

Totally, only our new pattern satisfies all five mentioned conditions, and is the most suitable pattern in the mentioned m-healthcare scenario.

7 Conclusion

A new generic pattern for a MDVS scheme was proposed in which a threshold number t of n designated verifiers are able to verify the signature, cooperatively. This pattern was called as $(t, n)-$ MDVS scheme. Unforgeability, non-transferability and threshold verifiability were introduced as three basic security requirements of a $(t, n)-$ MDVS scheme. Afterwards, a concrete $(t, n)-$ MDVS scheme was proposed based on pairings and its basic security requirements were proved in the standard model (without random oracles). Moreover, the proposal was compared with the existing MDVS schemes. Finally, some explanations about the application of the proposed pattern were provided. Introducing enhanced security requirements for a $(t, n)-$ MDVS scheme, proposing other concrete schemes and presenting other applications of the proposed pattern can be considered as the future works in this field.

8 References

[1] Rivest, R.L., Shamir, A., Adleman, L.: 'A method for obtaining digital signatures and public-key cryptosystems', *Commun. ACM*, 1978, **21**, (2), pp. 120–126

[2] Chaum, D., Van Antwerpen, H.: 'Undeniable signatures'. Conf. on the Theory and Application of Cryptology, Santa Barbara, California, USA, August 1989, pp. 212–216

[3] Jakobsson, M., Sako, K., Impagliazzo, R.: 'Designated verifier proofs and their applications'. Int. Conf. on the Theory and Applications of Cryptographic Techniques, Saragossa, Spain, 1996, pp. 143–154

[4] Chaum, D.: 'Private signature and proof systems'. U.S. Patent 5,493,614, 1996

[5] Steinfeld, R., Bull, L., Wang, H., *et al.*: 'Universal designated-verifier signatures'. Int. Conf. on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, 2003, pp. 523–542

[6] Desmedt, Y.: 'Verifier-designated signatures'. Rump Session, Crypto'3, Santa Barbara, California, USA, 2003

[7] Laguillaumie, F., Vergnaud, D.: 'Multi-designated verifiers signatures'. Int. Conf. on Information and Communications Security, Berlin, Germany, 2004, pp. 495–507

[8] Ng, C. Y., Susilo, W., Mu, Y.: 'Universal designated multi verifier signature schemes'. Proc. 11th Int. Conf. on Parallel and Distributed Systems, Fukuoka, Japan, July 2005, vol. 2, pp. 305–309

[9] Shailaja, G., Kumar, K.P., Saxena, A.: 'Universal designated multi verifier signature without random oracles'. 9th Int. Conf. on Information Technology, ICIT'06, Bhubaneswar, India, 2006, pp. 168–171

[10] Chow, S.S.: 'Identity-based strong multi-designated verifiers signatures'. European Public Key Infrastructure Workshop, Turin, Italy, 2006, pp. 257–259

[11] Laguillaumie, F., Vergnaud, D.: 'Multi-designated verifiers signatures: anonymity without encryption', *Inf. Process. Lett.*, 2007, **102**, (2–3), pp. 127–132

[12] Li, Y., Susilo, W., Mu, Y., *et al.*: 'Designated verifier signature: definition, framework and new constructions'. Int. Conf. on Ubiquitous Intelligence and Computing, Hong Kong, China, 2007, pp. 1191–1200

[13] Chow, S.S.: 'Multi-designated verifiers signatures revisited', *IJ Netw. Secur.*, 2008, **7**, (3), pp. 348–357

[14] Vergnaud, D.: 'New extensions of pairing-based signatures into universal (multi) designated verifier signatures', *Int. J. Found. Comput. Sci.*, 2009, **20**, (1), pp. 109–133

[15] Chang, T.Y.: 'An ID-based multi-signer universal designated multi-verifier signature scheme', *Inf. Comput.*, 2011, **209**, (7), pp. 1007–1015

[16] Tian, H.: 'A new strong multiple designated verifiers signature for broadcast propagation'. Third Int. Conf. on Intelligent Networking and Collaborative Systems (INCoS), Fukuoka, Japan, 2011, pp. 268–274

[17] Tian, H.: 'A new strong multiple designated verifiers signature', *Int. J. Grid Util. Comput.*, 2012, **3**, (1), pp. 1–11

[18] Au, M.H., Yang, G., Susilo, W., *et al.*: '(Strong) multidesignated verifiers signatures secure against rogue key attack', *Concurrency Comput. Pract. Exp.*, 2014, **26**, (8), pp. 1574–1592

[19] Ming, Y., Wang, Y.: 'Universal designated multi verifier signature scheme without random oracles', *Wuhan Univ. J. Nat. Sci.*, 2008, **13**, (6), pp. 685–691

[20] Seo, S.H., Hwang, J.Y., Choi, K.Y., *et al.*: 'Identity-based universal designated multi-verifiers signature schemes', *Comput. Stand. Interfaces*, 2008, **30**, (5), pp. 288–295

[21] Yang, B., Xiao, Z., Yang, Y., *et al.*: 'A strong multi-designated verifiers signature scheme', *Front. Electr. Electron. Eng. Chin.*, 2008, **3**, (2), pp. 167–170

[22] Deng, L., Zeng, J., Huang, H.: 'ID-based multi-signer universal designated multi-verifier signature based on discrete logarithm', *Chiang MAI J. Sci.*, 2018, **45**, (1), pp. 617–624

[23] Ng, C.Y., Susilo, W., Mu, Y.: 'Designated group credentials'. Proc. of the 2006 ACM Symp. on Information, Computer and Communications Security, Taipei, Taiwan, 2006, pp. 59–65

[24] Zuo, L., Kumar, N., Tu, H., *et al.*: 'Detection and analysis of secure intelligent universal designated verifier signature scheme for electronic voting system', *J. Supercomput.*, 2014, **70**, (1), pp. 177–199

[25] Huang, Q., Yang, G., Wong, D.S., *et al.*: 'Ambiguous optimistic fair exchange'. Int. Conf. on the Theory and Application of Cryptology and Information Security, Melbourne, Australia, 2008, pp. 74–89

[26] Shin, S., Kwon, T.: 'A survey of public provable data possession schemes with batch verification in cloud storage', *J. Internet Serv. Inf. Secur.*, 2015, **5**, (3), pp. 37–47

[27] Zhou, J., Lin, X., Dong, X., *et al.*: 'PSMPA: patient self-controllable and multi-level privacy-preserving cooperative authentication in distributedm-healthcare cloud computing system', *IEEE Trans. Parallel Distrib. Syst.*, 2015, **26**, (6), pp. 1693–1703

[28] Boneh, D., Franklin, M.: 'Identity-based encryption from the Weil pairing'. Annual Int. Cryptology Conf., Santa Barbara, California, USA, 2001, pp. 213–229

[29] Shamir, A.: 'How to share a secret', *Commun. ACM*, 1979, **22**, (11), pp. 612–613

[30] Li, Y., Lipmaa, H., Pei, D.: 'On delegatability of four designated verifier signatures'. Int. Conf. on Information and Communications Security, Beijing, China, 2005, pp. 61–71

[31] Shim, K.A.: 'On delegatability of designated verifier signature schemes', *Inf. Sci.*, 2014, **281**, pp. 365–372

[32] Tian, H., Jiang, Z., Liu, Y., *et al.*: 'A non-delegatable strong designated verifier signature without random oracles'. 4th Int. Conf. on Intelligent Networking and Collaborative Systems (INCoS), Bucharest, Romania, 2012, pp. 237–244

[33] Waters, B.: 'Efficient identity-based encryption without random oracles'. Annual Int. Conf. on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, 2005, pp. 114–127

[34] Bellare, M., Rogaway, P.: 'Random oracles are practical: a paradigm for designing efficient protocols'. Proc. of the 1st ACM Conf. on Computer and Communications Security, Fairfax, Virginia, USA, 1993, pp. 62–73