Noname manuscript No.
(will be inserted by the editor)

Multi-Designated Verifiers Signature Schemes with Threshold Verifiability

Generic Pattern and a Concrete Scheme in the Standard Model

Parvin Rastegari · Mehdi Berenjkoub

Received: date / Accepted: date

Abstract In a designated verifier signature (DVS) scheme, the validity of the signature can only be verified 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 is included of multiple designated verifiers. To the best of our knowledge, there are two existing patterns for an MDVS. In the first pattern, the cooperation of all designated verifiers is necessary for checking the validity of the signature. In the second pattern, every verifier of the set of designated verifiers can check the validity of the signature, independently. In this paper, we propose a generic new pattern for an MDVS in which a threshold number of the set of designated verifiers can check the validity of the signature. We present a concrete scheme and prove its security requierements in the standard model. Finally, we will propose some applications of this pattern.

Keywords Digital Signature · Designated Verifier Signature Scheme · Multi-Designated Verifiers Signature Scheme · Threshold Verifiability · Standard Model

1 Introduction

Digital signature is an important primitive to provide integrity and authenticity of messages in security protocols [1]. A traditional digital signature scheme

P. Rastegari
Department of Electrical and Computer Engineering, Isfahan University of Technology,Isfahan 84156-83111, Iran
Tel.: +98-913-2718203
E-mail: parvin.rastegari@ec.iut.ac.ir

M. Berenjkoub
Department of Electrical and Computer Engineering, Isfahan University of Technology,Isfahan 84156-83111, Iran

is publicly verifiable, i.e. 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 public verifiability of digital signatures is a useful and necessary property in some applications, it is not a desired property in applications such as e-votings, e-auctions, fair exchanges, etc., 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, authors in [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/proof (DVS/DVP) 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. 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 (SDVS), 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 (UDVS) 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 generalization of a DVS [6]. This notion was first formalized in 2004, by Laguillaumie et al. [7]. Since then, a number of MDVS schemes with different properties in different setting models have been proposed [8-12]. Readers can refer to [10] for a survey. These MDVS schemes are proposed based on two different patterns. In the first pattern, all designated verifiers have to cooperate in order to verify the validity of the signature, such as the proposed scheme in [7]. In the second pattern, every verifier of the set of designated verifiers is able to verify the validity of the signature by its own, such as the proposed scheme in [11]. 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 propose a concrete scheme in the standard model and present some applications of this pattern.

The contributions of this paper are summarized as follows:

- Proposing a new generic pattern for an MDVS with threshold verifiability.
- Introducing a concrete MDVS scheme with threshold verifiability in the standard model, based on the proposed generic pattern.
- Presenting some applications of the proposed pattern.

Generally, our proposed pattern for an MDVS with threshold verifiability is

useful in situations wherever an MDVS in the first pattern is applicable, but all designated verifiers may not be present for verifying the signature at the same time. Furthermore, the properties of our proposal allows us to overcome the conflicts between the undeniability and the privacy of the signer in some applications such as fair exchanges.
The rest of this paper is organized as follows. Section 2 covers the formal models and the basic security requirements of two existing patterns for MDVS schemes. In Section 3, we propose a new generic pattern for an MDVS with threshold verifiability and present the formal model and the basic security requirements for this pattern. 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, we present some applications of our proposed pattern. Section 6 contains the concluding remarks.

2 Existing Patterns for an MDVS Scheme

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

2.1 Formal Model

An MDVS scheme is included of a signer s and a set of n designated verifiers $\{v_1, v_2, \ldots, v_n\}$.
Definition 1. An MDVS scheme 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 [7-12]:
Setup: It is a probabilistic polynomial time (PPT) algorithm which takes as input a security parameter k and outputs system parameters $params$.

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

Signer Key Generation (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.

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

Verifiers Key Generation (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 $i = 1, 2, \ldots, n$.

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

Designated Signature generation (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 $\mathcal{V} = \{Pk_{v_1}, Pk_{v_2}, \ldots, Pk_{v_n}\}$, and outputs a designated signature σ on message m.

$$\sigma \longleftarrow DSign(m, params, Sk_s, \mathcal{V}).$$

Designated Signature Verification (DVer): This algorithm is defined differently in two existing patterns.

- DVer in the first pattern [7]: 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 $\mathcal{V} = \{Pk_{v_1}, Pk_{v_2}, \ldots, Pk_{v_n}\}$, and all verifiers' private keys $\mathcal{S} = \{Sk_{v_1}, Sk_{v_2}, \ldots, Sk_{v_n}\}$, and outputs 1 if the designated signature is valid and 0 otherwise.

$$0\,or1 \longleftarrow DVer((m, \sigma), params, Pk_s, \mathcal{V}, \mathcal{S}).$$

- DVer in the second pattern [11]: 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 $\mathcal{V} = \{Pk_{v_1}, Pk_{v_2}, \ldots, Pk_{v_n}\}$, and one verifier's private key $Sk_{v_i} \in \mathcal{S} = \{Sk_{v_1}, Sk_{v_2}, \ldots, Sk_{v_n}\}$, and outputs 1 if the designated signature is valid and 0 otherwise.

$$0\,or1 \longleftarrow DVer((m, \sigma), params, Pk_s, \mathcal{V}, Sk_{v_i}).$$

In the next subsection, the basic security requirements of an MDVS scheme will be described.

2.2 Security Requirements

Correctness, unforgeability and non-transferability (source hiding) are three basic requirements of an MDVS scheme [10]. In the following, the descriptions of these requirements are provided.

Correctness must be satisfied in an MDVS scheme. This property is considered as follows in two existing patterns.

- In the first pattern, if $\sigma \longleftarrow DSign(m, params, Sk_s, \mathcal{V})$, then the output of $DVer((m, \sigma), params, Pk_s, \mathcal{V}, \mathcal{S})$ must be 1 [7].
- In the second pattern, if $\sigma \longleftarrow DSign(m, params, Sk_s, \mathcal{V})$, then the output of $DVer((m, \sigma), params, Pk_s, \mathcal{V}, Sk_{v_i})$ (for all $i \in \{1, 2, \ldots, n\}$) must be 1. Furthermore, for any values (m, σ), $\mathcal{V}$ and Pk_s, if there exists an $Sk_{v_j} \in \mathcal{S}$ such that $1 \longleftarrow DVer((m, \sigma), params, Pk_s, \mathcal{V}, Sk_{v_j})$, then for any $Sk_{v_i} \in \mathcal{S}, (i \neq j)$, it must hold that $1 \longleftarrow DVer((m, \sigma), params, Pk_s, \mathcal{V}, Sk_{v_i})$[18].

Unforgeability is considered as existential unforgeability against chosen message attack (EUF-CMA) and is defined by the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ [12].

Game I:

Setup: $\mathcal{C}$ runs $params \longleftarrow Setup(k)$, $(Sk_s, Pk_s) \longleftarrow SKG(params)$, and $(Sk_{v_i}, Pk_{v_i}) \longleftarrow VKG(params, n)$ for $i = 1, 2, \ldots, n$, to obtain $(params, (Sk_s, Pk_s), (\mathcal{S}, \mathcal{V}))$. Then $\mathcal{C}$ gives $(params, Pk_s, \mathcal{V})$ to $\mathcal{A}$.
Oracle Accesses: $\mathcal{A}$ has access to the following oracles:
$\mathcal{O}_{Sign}$. Refers to the designated signature oracle which takes as input a message m and outputs $\sigma = DSign(m, params, Sk_s, \mathcal{V})$.
$\mathcal{O}_{Ver}$. Refers to the verification oracle which takes as input a pair (m, σ) and outputs 1 if σ is a valid designated signature on m, and 0 otherwise.
Forgery: $\mathcal{A}$ outputs (m^*, σ^*). ($\mathcal{A}$ is not allowed to submit a query from $\mathcal{O}_{Sign}$ with input m^*.)

Note: Oracle accesses are defined a little different in different papers [7-12]. In the above game we have mentioned those which are more common and basic in the literature.

It is said that $\mathcal{A}$ wins Game I if σ^* is a valid designated signature on m^*, i. e:

- $1 \longleftarrow DVer((m^*, \sigma^*), params, Pk_s, \mathcal{V}, \mathcal{S})$, in the firt pattern, and
- There exists a public key $Pk_{v_j} \in \mathcal{V}$ such that $1 \longleftarrow DVer\left((m^*, \sigma^*), params, Pk_s, \mathcal{V}, Sk_{v_j}\right)$, in the second pattern.

Definition 2. An MDVS scheme is $(t'', \varepsilon'', q_S, q_V)-$unforgeable, if no PPT adversary with at most q_S queries from $\mathcal{O}_{Sign}$ and q_V queries from $\mathcal{O}_{Ver}$ can win Game I in time at most t'' with probability at least ε''.

Non-transferability (source hiding) 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 an signature indistinguishable from the one that should be generated by the signer. This property is defined the same in two existing patterns [7-12].
Definition 3. 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 $(\mathcal{S}, \mathcal{V})$, and a message m, outputs a designated signature σ, which is indistinguishable from $\sigma \longleftarrow DSign(m, params, Sk_s, \mathcal{V})$. In other words, for all PPT algorithms $\mathcal{D}$, for any security parameter k, $params \longleftarrow Setup(k)$, $(Sk_s, Pk_s) \longleftarrow SKG(params)$, $(\mathcal{S}, \mathcal{V}) \longleftarrow VKG(params, n)$ and any message m, the value of

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

must be negligible.
Note that an MDVS scheme must satisfy correctness, unforgeability and non-transferability as its basic security requirements. Some other advanced security requirements may be defined in some papers, such as non-delegability [13] or unforgeability against rogue key attack [11]. Here, we only focused on the basic security requirements to propose our new pattern.

3 Our Proposed Pattern for an MDVS Scheme with Threshold Verifiability

In the previous section, we described two existing patterns for an MDVS scheme. In Definition 1, it was mentioned that two existing patterns are the same in all algorithms except in the designated signature verification (DVer) phase. In the first pattern, the cooperation of all designated verifiers is necessary in order to verify the signature, while in the second pattern, every member of the set of verifiers is able to verify the signature by its own.
Here, we will 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 will use the notation $(t, n)-$MDVS for an MDVS which is verifiable by the cooperation of a threshold number t of n designated verifiers.

3.1 Formal Model

A multi designated verifiers signature scheme with threshold verifiability is included of a signer s and a set of n designated verifiers $\{v_1, v_2, \ldots, v_n\}$ and is defined as follows:
Definition 4. A $(t, n)-$MDVS scheme is defined by five main algorithms: Setup, Signer Key Generation (SKG), Verifiers Key Generation (VKG), Designated Signature generation (DSign) and Threshold Verification (Th.Ver). We will define these algorithms as follows:
Setup: It is similar to the setup phase in Definition 1, i.e.

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

Signer Key Generation (SKG): It is similar to the SKG phase in Definition 1, i.e.

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

Verifiers Key Generation (VKG): It is similar to the VKG in Definition 1, i.e. for $i = 1, 2, \ldots, n$.

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

Furthermore, in this phase, n designated verifiers may run a $(t, n)-$secret sharing [21] between themselves in order to obtain their shares of other verifiers secret keys.
Designated Signature generation (DSign): is similar to the Dsign phase in Definition 1, i. e.

$$\sigma \longleftarrow DSign(m, params, Sk_s, \mathcal{V}).$$

Threshold Verification (Th.Ver): 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 $\mathcal{V}$, and t verifiers' shares of all verifiers' secret keys $\mathcal{S}^t$, and outputs 1 if the designated signature is valid and 0 otherwise.

$$0\, or 1 \longleftarrow Th.Ver((m, \sigma), params, Pk_s, \mathcal{V}, \mathcal{S}^t).$$

3.2 Security Requirements

Correctness, unforgeability, non-transferability (source hiding) and threshold verifiability are four basic requirements of our proposed $(t, n)-$MDVS scheme.

Correctness must be satisfied in a $(t, n)-$MDVS scheme. i. e., if $\sigma \longleftarrow DSign(m, params, Sk_s, \mathcal{V})$, then the output of $Th.Ver((m, \sigma), params, Pk_s, \mathcal{V}, \mathcal{S}^t)$ must be 1 for all $\mathcal{S}^t$s. In other words, if σ is a valid designated signature on message m, it must pass the Th.Ver phase which is performed by the cooperation of any t verifiers of the set of all verifiers.

Unforgeability is considered as existential unforgeability against chosen message attack (EUF-CMA) and is again defined by Game I (in Section 2) between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$.
Definition 5. It is said that $\mathcal{A}$ wins Game I if σ^* is a valid signature on m^*, i.e. for all $\mathcal{S}^t$s, it holds that $1 \longleftarrow Th.Ver((m^*, \sigma^*), params, Pk_s, \mathcal{V}, \mathcal{S}^t)$. A $(t, n)-$MDVS scheme is $(t'', \varepsilon'', q_S, q_V)-$unforgeable, if no PPT adversary with at most q_S queries from $\mathcal{O}_{Sign}$ and q_V queries from $\mathcal{O}_{Ver}$ can win Game I in time at most t'' with probability at least ε''.

Non-transferability (source hiding) is considered to provide the privacy of the signer. As two existing patterns for an MDVS scheme, this property is ensured by a transcript simulation algorithm that can be performed by the cooperation of all designated verifiers to produce a signature indistinguishable from the one that should be produced by the signer.
Definition 6. Non-transferability is again defined by Definition 3 for a $(t, n)-$MDVS scheme, by an extra assumption that any set of less than t designated verifiers cannot create an indistinguishable signature from that generated by the signer.

Threshold Verifiability 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 7. A $(t, n)-$MDVS scheme is threshold verifiable if the signature can only be verified by the cooperation of at least t designated verifiers.

4 A Concrete $(t, n)-$MDVS scheme

In this section we will present a concrete $(t, n)-$MDVS scheme and prove its security requirements in the standard model (without random oracles). In [12] 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 first pattern in Section 2. The authors in [12] used the Waters' signature [15] as the base of their scheme. We will also use the Waters' signature to propose our concrete $(t, n)-$MDVS scheme in this section. Furthermore, we will use Shammir secret sharing [14] to provide threshold verifiability.

4.1 Preliminaries

Before proposing our concrete scheme, some required preliminaries will be described in this subsection.

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 $e : G_1 \times G_1 \longrightarrow G_2$ if and only if the following properties are satisfied.

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

It can be referred to [16] for more details about bilinear pairings.

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 (CBDH) 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 $\mathcal{O}_{DBDH}$. The DBDH oracle $\mathcal{O}_{DBDH}$ 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 8. 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).

4.2 Our Concrete Scheme

In this section, we use Waters' signature [15] and Shammir secret sharing [14] to propose our concrete $(t,n)-$MDVS scheme. Assume that messages are bit strings of length n_m. For generality, messages can be considered of arbitrary lengths and a hash function $H_m : \{0,1\}^* \longrightarrow \{0,1\}^{n_m}$ can be used to convert messages to the specific length. 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 $e : G_1 \times G_1 \longrightarrow G_2$ is an admissible bilinear pairing. Other parameters are random elements of G_1, i.e. $g_1, m', m_1, , m_{n_m} \in_R G_1$.

Signer Key Generation (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 $(Sk_s, Pk_s) = (x_s, g^{x_s})$.

Verifiers Key Generation (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 $i-$th 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 $(t,n)-$Shammir secret sharing [14] 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:

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

  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 that 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 $s_k = \sum_{i=1}^{n} Sk_{v_i} = \sum_{k=1}^{t} \lambda_k s_k$, where λ_k ,$(k = 1, 2, \ldots, t)$, are Lagrange coefficients, i.e. $\lambda_k = \prod_{i \in A-\{k\}} \frac{i}{i-k}$, where $A = \{1, 2, \ldots, t\}$).

Designated Signature generation (DSign): Let $m[\ell]$ denotes the $\ell-$th bit of the message m of length n_m. Define $\mathcal{M} \subseteq \{1, 2, \ldots, n_m\}$ 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 [15]:

$$\sigma' = (\sigma_1', \sigma_2') = (g_1^{Sk_s}(m' \prod_{\ell \in \mathcal{M}} m_\ell)^r, g^r). \tag{2}$$

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 $i-$th designated verifier. The signer outputs $\sigma = (\sigma_1, \sigma_2)$ as his designated signature for n designated verifiers.

Threshold Verification (Th.Ver): 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, suppose that $v_1, v_2, \ldots, v_t$ $(1 < t < n)$ will 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 $v_k (1 \leq k \leq t)$ computes:

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

  where Pk_s is the public key of the signer.
- Every $v_k (1 \leq k \leq t)$ 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, $(1 \leq j \leq t, j \neq k)$.
- v_j can verify the correctness of the received share of v_k, i.e. Ψ_k, by checking whether the equation $e(\Psi_k, g) = e(\Phi, g^{s_k})$ holds or not. If the equality 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.

– Each $v_k (1 \leq k \leq t)$ can calculate $\prod_1^t \Psi_k^{\lambda_k}$ and accepts the signature iff:

$$\prod_1^t \Psi_k^{\lambda_k} = \sigma_k. \tag{3}$$

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

4.3 Security Analysis

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

Correctness: Suppose that $\sigma = (\sigma_1, \sigma_2)$ is a valid $(t, n)-$MDVS on m, so it must pass the Th.Ver phase.

Lemma 1. Correctness holds in our $(t, n)-$MDVS scheme.

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

$$\begin{aligned}
\prod_1^t \Psi_k^{\lambda_k} &= \Phi^{\sum_{k=1}^t \lambda_k s_k} \\
&= \Phi^{\sum_{i=1}^n Sk_{v_i}} \\
&= e(g_1, Pk_s)^{\sum_{i=1}^n Sk_{v_i}} e((m' \prod_{\ell \in \mathcal{M}} m_\ell), \sigma_2)^{\sum_{i=1}^n Sk_{v_i}} \\
&= e(g_1, g^{Sk_s})^{\sum_{i=1}^n Sk_{v_i}} e((m' \prod_{\ell \in \mathcal{M}} m_\ell), g^r)^{\sum_{i=1}^n Sk_{v_i}} \\
&= e(g_1^{Sk_s}, g^{\sum_{i=1}^n Sk_{v_i}}) e((m' \prod_{\ell \in \mathcal{M}} m_\ell)^r, g^{\sum_{i=1}^n Sk_{v_i}}) \\
&= e(g_1^{Sk_s} (m' \prod_{\ell \in \mathcal{M}} m_\ell)^r, g^{\sum_{i=1}^n Sk_{v_i}}) \\
&= e(\sigma_1', \prod_{i=1}^n Pk_{v_i}) \\
&= \sigma_1.
\end{aligned} \tag{4}$$

According to (4), if $\sigma = (\sigma_1, \sigma_2)$ is a valid designated signature on message m then the output of $Th.Ver((m, \sigma), params, Pk_s, \mathcal{V}, \mathcal{S}^t)$ is 1 for all $\mathcal{S}^t \subset \mathcal{S}$. In other words, $\sigma = (\sigma_1, \sigma_2)$ passes the Th.Ver phase which is performed by the cooperation of any t verifiers of the set of all verifiers.

Unforgeability: In this part, the unforgeability of the proposed scheme is analyzed, according to Definition 5.

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

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

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

in which t'' is the required time for $\mathcal{A}$ 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).

Proof. Since Waters presented his scheme in the standard model in 2001 [15], many researchers have used his ideas in order to present and prove different encryption and signature schemes in the standard model [12]. We also used the Waters signature as a base for presenting our $(t, n)-$ MDVS scheme as in (2) and we will use his techniques to prove the unforgeability of our scheme. Our method of proof is similar to the method presented in [12] with some differences in details.

Suppose that there exists an adversary $\mathcal{A}$ who can $(t'', \varepsilon'', q_S, q_V)$ break the scheme by running Game I as Definition 5. By this assumption, we can construct an algorithm $\mathcal{B}$ which can solve a GBDH problem in time at most t' and with probability at least ε' by using $\mathcal{A}$ as a sub-routine.

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

- Setup: $\mathcal{B}$ 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 $0 \leq k_m l_m < q$ (Remember that q is the order of G_1 and G_2.). $\mathcal{B}$ 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 $\mathcal{B}$. In order to follow the proof more easily, define two following functions:

$$J(m) = x' + \sum_{\ell \in \mathcal{M}} x_\ell - k_m l_m,$$

$$K(m) = y' + \sum_{\ell \in \mathcal{M}} y_\ell,$$

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

- $\mathcal{B}$ 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. (Note that g^b is one of the inputs of the GBDH problem which $\mathcal{B}$ is trying to solve it).
- $\mathcal{B}$ sets $g_1 = g^c$. (Note that g^c is one of the inputs of the GBDH problem which $\mathcal{B}$ is trying to solve it).
- $\mathcal{B}$ 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: $m' \prod_{\ell \in \mathcal{M}} m_\ell = g_1^{J(m)} g^{K(m)}$.

Afterwards, $\mathcal{B}$ 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 $\mathcal{A}$. From the perspective of $\mathcal{A}$, all distributions are identical to those in the real world.

- Oracle Accesses: $\mathcal{A}$ has access to the $\mathcal{O}_{Sign}$ and $\mathcal{O}_{Ver}$ oracles as mentioned in Game I. $\mathcal{B}$ plays the role of theses oracles, i. e. when $\mathcal{A}$ inputs its queries to these oracles, $\mathcal{B}$ will generate the corresponding outputs for $\mathcal{A}$ as follows:
  - $\mathcal{O}_{Sign}$. On input a message m this oracle must output a valid designated signature σ on m. When $\mathcal{A}$ gives $\mathcal{O}_{Sign}$ the message m as input, $\mathcal{B}$ must generate a valid $\sigma = (\sigma_1, \sigma_2)$ without knowing the private key of the signer and designated verifiers (Note that $\mathcal{B}$ does not know a, b, c). To produce $\sigma = (\sigma_1, \sigma_2)$, $\mathcal{B}$ acts as follows:
    - If $J(m) = 0 \bmod q$, $\mathcal{B}$ aborts and reports a failure.
    - If $J(m) \neq 0 \bmod q$, $\mathcal{B}$ randomly selects $r \in_R Z_q^*$. Then $\mathcal{B}$ computes:

$$\sigma = (e(Pk_s^{\frac{-K(m)}{J(m)}} (m' \prod_{\ell \in \mathcal{M}} m_\ell)^r, \prod_{i=1}^{n} Pk_{v_i}), g^r Pk_s^{\frac{-1}{J(m)}}). \tag{5}$$

By Noting (5) and Defining $\tilde{r} = r - \frac{a}{J(m)}$, we have:

$$\begin{aligned}
\sigma_1 &= e(Pk_s^{\frac{-K(m)}{J(m)}} (m' \prod_{\ell \in \mathcal{M}} m_\ell)^r, \prod_{i=1}^{n} Pk_{v_i}) \\
&= e(g^{-a\frac{K(m)}{J(m)}} (g_1^{J(m)} g^{K(m)})^r, \prod_{i=1}^{n} Pk_{v_i}) \\
&= e((g_1^{J(m)} g^{K(m)})^{\frac{-a}{J(m)}} g_1^a (g_1^{J(m)} g^{K(m)})^r, \prod_{i=1}^{n} Pk_{v_i}) \\
&= e(g_1^a (g_1^{J(m)} g^{K(m)})^{r - \frac{a}{J(m)}}, \prod_{i=1}^{n} Pk_{v_i}) \\
&= e(g_1^a (m' \prod_{\ell \in \mathcal{M}} m_\ell)^{\tilde{r}}, \prod_{i=1}^{n} Pk_{v_i}),
\end{aligned} \tag{6}$$

and also:

$$\sigma_2 = g^r Pk_s^{\frac{-1}{J(m)}} = g^r g^{\frac{-a}{J(m)}} = g^{r - \frac{a}{J(m)}} = g^{\tilde{r}}. \tag{7}$$

According to (6) and (7), the signature $\sigma = (\sigma_1, \sigma_2)$ computed by (5), is a valid designated signature on m.

- $\mathcal{O}_{Ver}$. 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 $\mathcal{A}$ gives $\mathcal{O}_{Ver}$ the message/designated signature pair $(m, \sigma = (\sigma_1, \sigma_2))$ as input, $\mathcal{B}$ must verify the validity of σ without knowing the private key of the signer and designated verifiers (Note that $\mathcal{B}$ does not know a, b, c). To verify σ, $\mathcal{B}$ acts as follows:
  - If $J(m) = 0 \bmod q$, $\mathcal{B}$ submits

$$(g, g^a, \prod_{i=1}^{n} Pk_{v_i}, g^c, \frac{\sigma_1}{e(\sigma_2^{K(m)}, \prod_{i=1}^{n} Pk_{v_i})}), \tag{8}$$

to the DBDH oracle $\mathcal{O}_{DBDH}$ (Note that $\mathcal{B}$ is trying to solve a GBDH problem and has access to the $\mathcal{O}_{DBDH}$). Then $\mathcal{B}$ outputs 1 to $\mathcal{A}$ if the output of $\mathcal{O}_{DBDH}$ 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 (8) is a valid BDH tuple, as we have:

$$\begin{aligned} \frac{\sigma_1}{e(\sigma_2^{K(m)}, \prod_{i=1}^{n} Pk_{v_i})} &= \frac{e(g_1^a (m' \prod_{\ell \in \mathcal{M}} m_\ell)^r, \prod_{i=1}^{n} Pk_{v_i})}{e(\sigma_2^{K(m)}, \prod_{i=1}^{n} Pk_{v_i})} \\ &= \frac{e(g^{ca} (g^{K(m)})^r, \prod_{i=1}^{n} Pk_{v_i})}{e(g^{rK(m)}, \prod_{i=1}^{n} Pk_{v_i})} \\ &= e(g^{ca}, \prod_{i=1}^{n} Pk_{v_i}) \end{aligned} \tag{9}$$

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

$$(g, \prod_{i=1}^{n} Pk_{v_i}, m' \prod_{\ell \in \mathcal{M}} m_\ell, \frac{\sigma_2}{\hat{\sigma}_2}, \frac{\sigma_1}{\hat{\sigma}_1}), \tag{10}$$

to the DBDH oracle $\mathcal{O}_{DBDH}$. Then $\mathcal{B}$ outputs 1 to $\mathcal{A}$ if the output of $\mathcal{O}_{DBDH}$ 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 (10) 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):

$$\begin{aligned} \sigma_1 &= \prod_{k=1}^{t} \Psi_k^{\lambda_k} = \Phi^{\sum_{k=1}^{t} \lambda_k s_k} = \Phi^{\sum_{i=1}^{n} Sk_{v_i}} \\ &= e(g_1, Pk_s)^{\sum_{i=1}^{n} Sk_{v_i}} e(m' \prod_{\ell \in \mathcal{M}} m_\ell, \sigma_2)^{\sum_{i=1}^{n} Sk_{v_i}}. \end{aligned} \tag{11}$$

Similarly, since $\hat{\sigma} = (\hat{\sigma}_1, \hat{\sigma}_2)$ is a valid designated signature, we have $\hat{\sigma_2} = g^{\hat{r}}$ and

$$\begin{aligned}\hat{\sigma_1} &= \prod_{k=1}^{t} \hat{\Psi_k}^{\lambda_k} = \hat{\Phi}^{\sum_{k=1}^{t} \lambda_k s_k} = \hat{\Phi}^{\sum_{i=1}^{n} Sk_{v_i}} \\ &= e(g_1, Pk_s)^{\sum_{i=1}^{n} Sk_{v_i}} e(m' \prod_{\ell \in \mathcal{M}} m_\ell, \hat{\sigma_2})^{\sum_{i=1}^{n} Sk_{v_i}}. \end{aligned} \quad (12)$$

According to (11) and (12), we have:

$$\begin{aligned}\frac{\sigma_1}{\hat{\sigma_1}} &= e(m' \prod_{\ell \in \mathcal{M}} m_\ell, \frac{\sigma_2}{\hat{\sigma_2}})^{\sum_{i=1}^{n} Sk_{v_i}} \\ &= e(g^{cJ(m)} g^{K(m)}, g^{r-\hat{r}})^{\sum_{i=1}^{n} Sk_{v_i}} \end{aligned} \quad (13)$$

According to (13), and noting that $\frac{\sigma_2}{\hat{\sigma_2}} = g^{r-\hat{r}}$, $\prod_{i=1}^{n} Pk_{v_i} = g^{\sum_{i=1}^{n} Sk_{v_i}}$, and $m' \prod_{\ell \in \mathcal{M}} m_\ell = g^{cJ(m)} g^{K(m)}$, the tuple in (10) is a valid BDH tuple.

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

$$e(g,g)^{abc} = \left(\frac{{\sigma_1}^*}{e({\sigma_2}^{*K(m^*)}, \prod_{i=1}^{n} Pk_{v_i})}\right)^{(\sum_{i=1}^{n} d_i)^{-1}}. \quad (14)$$

It is easy to check that (14) holds, if $\sigma^* = (\sigma_1^*, \sigma_2^*)$ is a valid signature.

Time Analysis: Noting the above descriptions we can see that $\mathcal{B}$ 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 $\mathcal{A}$ 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 analyze the success probability of $\mathcal{B}$, we consider events in which $\mathcal{B}$ will not abort. $\mathcal{B}$ will not abort if both the two following events happen [19]:

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

Noting that $x', x_1, \ldots, x_{n_m} \in_R Z_{l_m}$, $l_m(n_m+1) < q$ and $0 \leq k_m l_m < q$, we have:

$$\begin{aligned} -q &< -k_m l_m \\ &\leq x' + \sum_{\ell \in \mathcal{M}} x_\ell - k_m l_m (= J(m)) \\ &\leq (l_m - 1)(n_m + 1) - k_m l_m \\ &= l_m(n_m + 1) - n_m - 1 - k_m l_m \\ &< q - n_m - 1 - k_m l_m < q, \end{aligned} \tag{15}$$

According to (15), since $-k_m l_m \leq J(m) \leq (l_m - 1)(n_m + 1) - k_m l_m$, $J(m)$ can take $(l_m - 1)(n_m + 1) + 1$ different values and since $-q < J(m) < q$, only for one of these values we have $J(m) = 0 \bmod q$. As a result:

$$Pr[J(m) = 0 \quad mod \quad q] = \frac{1}{(l_m - 1)(n_m + 1) + 1}. \tag{16}$$

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

$$Success\ Probability\ of\ \mathcal{B} = \varepsilon' \geq \varepsilon''.Pr[E_1 \bigcap E_2], \tag{17}$$

in which ε'' is is the least success probability of $\mathcal{A}$ to forge a signature. Noting (16) and that E_1 and E_2 are independent events, we have:

$$\begin{aligned} Pr[E_1 \bigcap E_2] &= Pr[E_1]Pr[E_2] \\ &= Pr[\bigcap_{i=1}^{q_S} E_{1i}]Pr[E_2] \\ &= (1 - Pr[\bigcup_{i=1}^{q_S} \bar{E}_{1i}])(\frac{1}{(l_m - 1)(n_m + 1) + 1}) \\ &\geq (1 - \frac{q_S}{(l_m - 1)(n_m + 1) + 1})(\frac{1}{(l_m - 1)(n_m + 1) + 1}) \\ &\geq (1 - \frac{q_S}{l_m})(\frac{1}{l_m(n_m + 1)}) = \frac{1}{4q_S(n_m + 1)}, \end{aligned} \tag{18}$$

where the rightmost equality is implied from $l_m = 2q_S$.
Noting (17) and (18), we have:

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

as the final result.

Non-transferability (source hiding) In this part, the non-transferability of the proposed scheme will be analyzed according to Definition 6.

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 6, 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:

$$\sigma_0 = (e(g_1^{Sk_s}(m' \prod_{\ell \in \mathcal{M}} m_\ell)^{r_0}, \prod_{i=1}^{n} Pk_{v_i}), g^{r_0}). \tag{19}$$

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:

$$\sigma_1 = (e(g_1, Pk_s)^{\sum_{i=1}^{n} Sk_{v_i}} e(m' \prod_{\ell \in \mathcal{M}} m_\ell, g^{r_1})^{\sum_{i=1}^{n} Sk_{v_i}}, g^{r_1}). \tag{20}$$

It is easy to see that σ_0 and σ_1 have the same distributions and hence they are indistinguishable.

Suppose that a challenger $\mathcal{C}$ selects a random element $r^* \in_R Z_q^*$ an sets $\sigma_2^* = g^{r^*}$, then picks a $b \in_R \{0,1\}$ by flipping a coin and sets σ_1^* as follows:

$$\sigma_1^* = \begin{cases} e(g_1^{Sk_s}(m' \prod_{\ell \in \mathcal{M}} m_\ell)^{r^*}, \prod_{i=1}^{n} Pk_{v_i}) & if\ b = 0 \\ e(g_1, Pk_s)^{\sum_{i=1}^{n} Sk_{v_i}} e(m' \prod_{\ell \in \mathcal{M}} m_\ell, g^{r^*})^{\sum_{i=1}^{n} Sk_{v_i}} & if\ b = 1 \end{cases}. \tag{21}$$

Noting (19), (20) and (21), we have:

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

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

Therefore, the distributions of σ_0 and σ_1 are identical and a distinguisher $\mathcal{D}$ cannot distinguish whether the signature is created by the signer or by TS. Also, any set of less than t designated verifiers cannot generate an indistinguishable signature from that created by the signer (Because they cannot calculate $\sum_{i=1}^{n} Sk_{v_i}$). Hence, the signature is unconditionally non-transferable.

Threshold verifiability In this part, the threshold verifiability of the proposed scheme will be analyzed according to Definition 7. In order to prove this property, first three following lemmas will be 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:

$$\sigma_1 = e(g_1, Pk_s)^{\sum_{i=1}^{n} Sk_{v_i}} e(m' \prod_{\ell \in \mathcal{M}} m_\ell), \sigma_2)^{\sum_{i=1}^{n} Sk_{v_i}},$$

  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 will show that if there exists an adversary $\mathcal{A}$ who can verify a signature without knowing $\sum_{i=1}^{n} Sk_{v_i}$, 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 $\mathcal{B}$ which can solve a GBDH problem in (G_1, G_2) in time at most t' and with probability at least ε' by using $\mathcal{A}$ as a sub-routine, where:

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

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

  Suppose that there exists an adversary $\mathcal{A}$ who can verify a signature without knowing $\sum_{i=1}^{n} Sk_{v_i}$, 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 $\mathcal{B}$ which can solve a GBDH problem in (G_1, G_2) in time at most t' and with probability at least ε' by using $\mathcal{A}$ as a sub-routine.
  A random GBDH challenge $g, g^a, g^b, g^c \in G_1$ is given to $\mathcal{B}$ and $\mathcal{B}$ tries to calculate $e(g, g)^{abc} \in G_2$ with the help of the DBDH oracle $\mathcal{O}_{DBDH}$. In order to solve this problem, $\mathcal{B}$ runs $\mathcal{A}$ as a sub-routine. $\mathcal{B}$ plays the following game with $\mathcal{A}$:
  - Setup: $\mathcal{B}$ 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:

$$J(m) = x' + \sum_{\ell \in \mathcal{M}} x_\ell - k_m l_m,$$

$$K(m) = y' + \sum_{\ell \in \mathcal{M}} y_\ell,$$

Then $\mathcal{B}$ 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 $j = 1, 2, \ldots, n_m$. $\mathcal{B}$ 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, $\mathcal{B}$ returnsPk_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 $\mathcal{A}$. From the perspective of $\mathcal{A}$, all distribution 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 $\mathcal{B}$ nor $\mathcal{A}$ can compute $\sum_{i=1}^{n} Sk_{v_i}$.

- Oracle Accesses: Suppose that $\mathcal{A}$ is trying to verify a designated signature $\sigma^* = (\sigma_1^*, \sigma_2^*)$ on a message m^*. $\mathcal{A}$ has access to the $\mathcal{O}_{Sign}$ and $\mathcal{O}_{Ver}$ oracles and $\mathcal{B}$ plays the role of theses oracles. $\mathcal{B}$ should answer $\mathcal{A}$'s queries without the knowledge of Sk_s and $\sum_{i=1}^{n} Sk_{v_i}$. When $\mathcal{A}$ inputs its queries to these oracles, $\mathcal{B}$ will generate the corresponding outputs for $\mathcal{A}$ as mentioned in the proof of Theorem 1.
  Note that $\mathcal{A}$ is not only allowed to send a request to the $\mathcal{O}_{Ver}$ for the verification of $\sigma^* = (\sigma_1^*, \sigma_2^*)$ on m^*, but also she is not allowed to send a request to the $\mathcal{O}_{Ver}$ 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 $\mathcal{A}$ is not allowed to send $(m^*, \sigma^* = (\sigma_1^*, \sigma_2^*))$ to the $\mathcal{O}_{Ver}$, she can pick a random $r' \in_R Z_q^*$ and calculate another signature $\sigma' = (\sigma_1', \sigma_2')$ on m^* as follows:

$$\sigma' = \begin{cases} \sigma_1' = \sigma_1^*.e((m' \prod_{\ell \in \mathcal{M}} m_\ell)^{r'}, \prod_{i=1}^{n} Pk_{v_i}) \\ \\ \sigma_2' = \sigma_2^*.g^{r'} \end{cases}. \tag{22}$$

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

$$\sigma^* = \begin{cases} \sigma_1^* = e(g_1^{Sk_s}(m' \prod_{\ell \in \mathcal{M}} m_\ell)^{r^*}, \prod_{i=1}^{n} Pk_{v_i}) \\ \\ \sigma_2^* = g^{r^*} \end{cases}, \tag{23}$$

  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 (22) and (23) we have:

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

  Therefore, if $\mathcal{A}$ is allowed to send $(m^*, \sigma' = (\sigma_1', \sigma_2'))$ to the $\mathcal{O}_{Ver}$, she can imply that $(m^*, \sigma^* = (\sigma_1^*, \sigma_2^*))$ is valid if the output of $\mathcal{O}_{Ver}$ is 1 and invalid otherwise. As a result, $\mathcal{A}$ is not allowed to send a request to the $\mathcal{O}_{Ver}$ for the verification of any signature on m^*, but she is allowed to send a request to the $\mathcal{O}_{Ver}$ for the verification of signatures on other messages and $\mathcal{B}$ 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 $\mathcal{A}$ (Remember that $\mathcal{B}$ is trying to solve a GBDH problem.). Since $\mathcal{A}$ accepts $\sigma^* = (\sigma_1^*, \sigma_2^*)$, $\mathcal{B}$ acts as follows:
  - If $J(m^*) \neq 0 \bmod q$, $\mathcal{B}$ aborts and reports a failure.
  - If $J(m^*) = 0 \bmod q$, $\mathcal{B}$ can solve the GBDH problem by obtaining $e(g,g)^{abc}$ as follows:

$$e(g,g)^{abc} = \left(\frac{{\sigma_1}^*}{e({\sigma_2^*}^{K(m^*)}, \prod_{i=1}^{n} Pk_{v_i})}\right)^{(\sum_{i=1}^{n} d_i)^{-1}}. \qquad (24)$$

    It is easy to check that (24) 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 $k-$th verifier v_k $(k = 1, 2, \ldots, n)$ knows $F(x) = \sum_{i=1}^{n} f_i(k) = \sum_{i=1}^{n} s_{ik} = s_k$. 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 Th.Ver 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 the equality in (3) without revealing $\sum_{i=1}^{n} Sk_{v_i}$.

Lemma 4. There isn' t 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. As a result, 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 Lemma 2, Lemma 3 and Lemma 4.

5 Some Applications of the Proposed Pattern

MDVS schemes have interesting properties which make them useful in many situations [3]. Our proposal for a $(t, n)-$MDVS scheme is more flexible and useful in such situations. In this section, some applications are proposed for a $(t, n)-$MDVS scheme.

5.1 General Applications

Consider a situation where a signer, Alice, wants to authenticate herself to n designated verifiers without loosing her privacy. Two existing patterns for MDVS schemes (introduced in Section 2) can be used in this scenario, but each of these patterns have some weaknesses as follows:

- In the first pattern, all of the n designated verifiers have to cooperate in order to verify the signature. But all of the n designated verifiers may not be present at the same time.
- In the second pattern, every verifier of the set of all designated verifiers is able to verify the signature by its own. But this may not be desirable for the signer and the other designated verifiers that one single verifier verifies the signature and accepts/rejects it by its own, since one verifier may be malicious and decide to accept/reject a signature without verifying it correctly for his/her own benefits.

In order to soften the mentioned challenges in two existing patterns of MDVS schemes, a $(t, n)-$MDVS scheme can be used by selecting t such that a set of t designated verifiers be always present in order to cooperate to verify the signature and also, it is acceptable for the signer and all designated verifiers that any set of t verifiers verify a signature and decide to accept/reject it. In order to use a $(t, n)-$MDVS scheme in these situations, two following assumptions are considered:

1. Ther is not any set of at least t designated verifiers thinking of cooperating to forge a signature (for other verifiers), maliciously.
2. Every set of designated verifiers may think of disturbing the privacy of the signer.

In other words, a set of at least t designated verifiers never think of forging a signature but may think of disturbing the privacy of the signer (Note that forging a signature is a worse malice in comparison with disturbing the privacy of the signer and these assumptions are reasonable.). By these assumptions, we are neither worry about forging a signature by a set of verifiers (by the first assumption and also noting that any set of less than t designated verifiers do not have enough information to create a signature), nor about disturbing the privacy of the signer because of the non-transferabiliy property of our proposed pattern (according to Definition 6).
As an example, consider an e-banking scenario in which the bank claims that

the customers will be authenticated without the loss of their privacy. In order to provide this claim, the bank can use a DVS scheme in which the customers' signatures can only be verified by an employee of the bank. In order to increase the security, an MDVS scheme can be used in which the cooperation of n employees of the bank is necessary in order to verify customers' signatures. In this case if one of the n employees is not present, signatures cannot be verified. Our proposal for a $(t, n)-$MDVS scheme can be useful in this scenario, since the cooperation of a threshold number of the employees is sufficient in order to verify the customers' signatures.
As an other example, consider an e-voting scenario in which the voters use a DVS scheme to sign their votes for a tallier (as a designated verifier) [13]. As a result, eligible voters are authenticated for the tallier and their privacy is preserved at the same time. In this case, the tallier (as the designated verifier) can generate an indistinguishable DVS from that created by an eligible voter (as the signer). Therefore, a malicious tallier can cast ballots instead of eligible voters. In order to prevent the tallier from this malice, a $(t, n)-$MDVS scheme can be used in which a threshold number t of n talliers can verify the voters' signatures cooperatively and at the same time any set of less than t talliers cannot generate an indistinguishable signature from that created by an eligible voter. Note that in this scenario it is supposed that:

1. The sets of more than t talliers never cooperate to create a signature instead of an eligible voter, but the sets of less than t talliers may think of this malice. Since in a $(t, n)-$MDVS scheme, any set of less than t talliers cannot generate an indistinguishable signature from that created by an eligible voter, this scheme is useful to prevent any set of less than t talliers from this malice.
2. Talliers may think of disturbing the privacy of the voters. Since in a $(t, n)-$MDVS scheme, talliers can cooperate to generate an indistinguishable signature from that created by an eligible voter, talliers cannot convince anyone of the voters' signatures and as a result talliers cannot disturb the privacy of the voters.

In this subsection, we described that how one can use a $(t, n)-$MDVS scheme to moderate the mentioned weaknesses of two existing patterns for an MDVS scheme. Furthermore, the interesting properties of a $(t, n)-$MDVS scheme allows us to soften the conflicts between the authentication and the privacy of the signer in some applications such as fair exchange protocols. In the next subsection, we will propose the application of a $(t, n)-$MDVS scheme in designing an Ambiguous Optimistic Fair Exchange (AOFE) protocol.

5.2 An AOFE Protocol Based on a $(t, n)-$MDVS Scheme

An Optimistic Fair Exchange (OFE) protocol is a fair way which allows two (sets of) parties (the initiator and the responder) to exchange information (or items) fairly. In an OFE protocol, there is a third party, called arbitrator, who

is only called in when a dispute happens between the two (sets of) parties. OFE is useful in many applications such as contract signing, fair negotiation, exchanging digital items on internet and so on.
Since the concept of OFE was introduced in 1997 [17], many researches have been done in order to model the security requirements and improve the security and efficiency of OFE protocols. In two recent works, the notion of ambiguity is considered to improve the security of an OFE and the concept of Ambiguous Optimistic Fair Exchange (AOFE) protocol is introduced [18, 19]. However, the proposed protocols in [18, 19] are so inefficient. In this subsection, we use our proposed $(t, n)-$MDVS scheme to apply ambiguity in an OFE protocol which is done between an initiator and a set of responders. Although some relaxed assumptions are considered in our proposal, but our idea is useful to design much more efficient AOFE protocols in comparison with the proposed protocols in [18, 19].
An OFE protocol consists of three types of parties: An initiator (Alice), a responder (Bob) and an arbitrator who is only called in when a dispute occurs. Traditionally, there are three message flows in an OFE protocol: Firstly, Alice sends a partial signature σ_p to Bob. This partial signature can be considered as the Alice's commitment to that she would send her full signature to Bob in the third message flow. Secondly, Bob sends his full signature to Alice. Finally, Alice sends her full signature to Bob. This is an execution of the protocol when Bob and Alice are faithful, but what if either Bob or Alice are not honest?

- If Alice refuses to send Bob her full signature in the third message flow, Bob can send σ_p and his full signature to the arbitrator and ask him to extract the Alice's full signature from σ_p. Note that the arbitrator must be able to extract Alice's full signature from σ_p for this situation. Then the arbitrator sends Alice's full signature (which is extracted by himself) to Bob and Bob's full signature (which Bob has sent to him) to Alice.
- If Bob refuses to send Alice his full signature in the second message flow, Alice should have no concern about giving away σ_p in the first message flow. For this purpose, anyone must be able to extract Alice's full signature from σ_p except the arbitrator. (Note that the arbitrator extracts Alice's full signature and sends it to Bob if and only if Bob has sent him his full signature for sending to Alice.)

It seems that by considering two mentioned items, there is not any concern about unfaithful behaviour of Alice or Bob. There are secure and efficient OFE protocols with these considerations in the literature. However, in two recent works [18, 19], the authors consider an enhanced security model for an OFE protocol in which the privacy of Alice is preserved. In this model, named an Ambiguous Optimistic Fair Exchange (AOFE) protocol, (Moreover the basic security requirements of an OFE,) σ_p must not be verifiable by anyone except Bob and the arbitrator and Bob must not be able to convince any third party (except the arbitrator) that Alice is the signer of σ_p.
The public verifiability of σ_p is undesirable in some applications. As an example, consider a situation in which a company A wants to buy an item from a

company B. Firstly, A sends a partial signature σ_p to B. σ_p is considered as a commitment that A wants to buy the item from B with a proposed price. Then B stops the protocol and presents σ_p to a third company C and encourages C to propose a higher price. B can repeat this procedure with A and C and finally sells the item with the highest price to A or C. We call this unfaithful behaviour of B as horse trade. Note that in an OFE protocol, σ_p can only be created by A and this prevents B from forcing A to buy an item (compulsory sale) by providing a forged σ_p to the arbitrator. As a result, an OFE protocol prevents B from compulsory sale but not from horse trade. In order to prevent B from horse trade, σ_p must be ambiguous (not public verifiable).
To the best of our knowledge, there are only two works in the literature which consider the ambiguity of σ_p and present AOFE protocols in order to prevent the problems of public verifiability of σ_p [18, 19]. But the AOFE protocols presented in [18, 19] are not so efficient. The protocol in [18] requires a selective-tag weakly CCA secure tag-based encryption, a weakly unforgeable signature, a strong one-time signature and a general NIZK (Non-Intractive Zero Knowledge) proof system in order to compute σ_p. Also, in [19] a general construction (not a concrete protocol) is presented for an AOFE protocol and in this construction, a Traceable Ring Signature (TRS) is requiered in order to compute σ_p. Since there is not efficient TRS schemes (either in the sense of the computation cost or the length of the signature) in the standard model in the literature, the protocol in [19] is not efficient, too.
With some relaxed assumptions in an AOFE protocol, our proposed pattern for a $(t, n)-$MDVS scheme is useful to produce σ_p in a much more efficient way (in comparison with two existing AOFE protocols in [18, 19]). Again, consider the situation in which company A wants to buy an item from company B. We can consider σ_p as a $(t, n)-$MDVS scheme from the signer s (an employee of company A) to the set of n designated verifiers $\{v_1, v_2, \ldots, v_n\}$ (in which $v_1, v_2, \ldots, v_{n-1}$ are $n-1$ employees of company B and v_n is the arbitrator). Here, it is assumed that the sets of at least t employees of company B may not think of compulsory sale or changing the proposed price, but they may think of horse trade (This is a reasonable assumption, since compulsory sale and changing the proposed price are worse malice in comparison with horse trade). Firstly, A sends a $(t, n)-$MDVS (from s to $\{v_1, v_2, \ldots, v_n\}$) as σ_p to B which is considered as a commitment that A wants to buy the item from B with a proposed price. Secondly, t employees of B verify σ_p cooperatively, and send the item to A, if σ_p is valid. Finally, A sends its full signature to B which can be used by B to withdraw the proposed price from A's bank account. This is a faithful execution of the protocol, but what if either some employees of B or A are not honest?

- If A refuses to send B its full signature in the third message flow, $t-1$ employees of B can send σ_p (and their shares for verifying it) and the item to the arbitrator and ask him to extract the A's full signature from σ_p. The arbitrator computes his share for verifying σ_p and verifies σ_p by using his share and the other $t-1$ shares received from B. Note that no set of less

than t employees of company B can cooperate to create an indistinguishable σ_p from that produced by A (the signer). By considering this fact and the assumption that no set of at least t employees of company B may think of compulsory sale or changing the proposed price, the arbitrator can trust that σ_p is really created by A. Then the arbitrator sends A's full signature (which is extracted by himself) to B and the item (which the employees of B have sent to him) to A.

– If B refuses to send A the item in the second message flow, Alice should have no concern about giving away σ_p in the first message flow, since anyone (except the arbitrator) is able to extract A's full signature from σ_p. Furthermore, the employees of B cannot convince any third party C (for any malice such as horse trade) that A is really creates σ_p, since σ_p is a $(t, n)-$MDVS scheme and every set of at least t employees of B can cooperate to produce an indistinguishable σ_p from that created by A (non-transferability). As a result, A does not have any concern about disturbing its privacy by giving away σ_p.

6 Conclusion

A new generic pattern for a multi-designated verifiers signature (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. 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). Finally, some applications of this scheme in e-banking, e-voting and ambiguous optimistic fair exchange were introduced, briefly. Introducing extra security requirements (such as non-delegatability) for a $(t, n)-$MDVS scheme, proposing other concrete schemes and presenting advanced security protocols (such as e-voting, e-auction, fair exchange, e-cash, etc.) based on this scheme, can be considered as future works in this field.

References

1. Rivest, R.L., Shamir, A., Adleman, L., A method for obtaining digital signatures and public-key cryptosystems, Communications of the ACM, vol. 2, no. 21, pp. 120−126, (1978)
2. Chaum, D., van Antwerpen, H., Undeniable signatures, In Advances in Cryptology−CRYPTO89 Proceedings, Springer New York, pp. 212−216, (1989)
3. Jakobsson, M., Sako, K., Impagliazzo, R., Designated verifier proofs and their applications, In Advances in Cryptology−EUROCRYPT96, Springer Berlin Heidelberg, pp. 143154, (1996)
4. Chaum, D., Private signature and proof systems, U.S. Patent 5,493,614,(1996)
5. Steinfeld, R., Bull, L.,Wang, H., Pieprzyk, J., Universal designated−verifier signatures, In Advances in Cryptology−Asiacrypt 2003, Springer Berlin Heidelberg, pp. 523−542, (2003)

6. Desmedt, Y., Verifier-designated signatures, In Rump Session, Crypto 2003, (2003)
7. Laguillaumie, F., Vergnaud, D., Multi-designated verifiers signatures, In Information and Communications Security, Springer Berlin Heidelberg, pp. 495−507 (2004)
8. Chow, S.S., Identity-based strong multi-designated verifiers signatures, In Public Key Infrastructure, Springer Berlin Heidelberg, pp. 257−259, (2006)
9. Chow, S.S., Multi-designated verifiers signatures revisited, International Journal of Network Security, vol. 7, no. 3, pp. 348−357, (2008)
10. Tian, H., A new strong multiple designated verifiers signature, International Journal of Grid and Utility Computing, vol.3, no. 1, pp. 1−11, (2012)
11. Au, M.H., Yang, G., Susilo, W., Zhang, Y., (Strong) multidesignated verifiers signatures secure against rogue key attack, Concurrency and Computation: Practice and Experience, vol. 26, no. 8, pp. 1574−1592, (2014)
12. Ming, Y., Wang, Y., Universal designated multi verifier signature scheme without random oracles, Wuhan University Journal Of Natural Sciences, vol. 13, no. 6, pp.685−691, (2008)
13. Shim, K.A.,On delegatability of designated verifier signature schemes', Information Sciences, vol. 281, no. 10, pp. 365−372, (2014)
14. Shamir, A., How to share a secret, Communications of the ACM, vol. 22, no. 11, pp. 612−613, (1979)
15. Waters, B., Efficient identity-based encryption without random oracles, In Advances in Cryptology, EUROCRYPT 2005, Springer Berlin Heidelberg, pp. 114−127, (2005)
16. Boneh, D., Franklin, M., Identity-based encryption from the Weil pairing, In Advances in Cryptology, CRYPTO 2001, Springer Berlin Heidelberg, pp. 213−229, (2001)
17. Asokan, N., Schunter, M., Waidner, M., Optimistic protocols for fair exchange, in ACM CCS 97, ACM, pp. 717, (1997)
18. Huang, Q., Yang, G., Wong, D.S. and Susilo, W., Ambiguous optimistic fair exchange: Definition and constructions, Theoretical Computer Science, 562, pp. 177−193, (2015)
19. Ganjavi, R., Asaar, M.R. and Salmasizadeh, M., An ambiguous optimistic fair exchange protocol with traceability, In Telecommunications (IST), 2014 7th International Symposium on. IEEE. pp. 919−924, (2014)