Practical Distributed Signatures in the Standard Model*

Yujue Wang[1,2], Duncan S. Wong[2], Qianhong Wu[3], Sherman S.M. Chow[4], Bo Qin[5], and Jianwei Liu[3]

[1] Key Laboratory of Aerospace Information Security and Trusted Computing Ministry of Education, School of Computer, Wuhan University, China
[2] Department of Computer Science City University of Hong Kong, Hong Kong, China
[3] School of Electronics and Information Engineering, Beihang University, China
[4] Department of Information Engineering Chinese University of Hong Kong, Hong Kong, China
[5] School of Information, Renmin University of China, Beijing, China
yjwang@whu.edu.cn, duncan@cityu.edu.hk, {qianhong.wu,liujianwei}@buaa.edu.cn, sherman@ie.cuhk.edu.hk, bo.qin@ruc.eud.cn

Abstract. A distributed signature scheme allows participants in a qualified set to jointly generate a signature which cannot be forged even when all the unqualified participants collude together. In this paper, we propose an efficient scheme for any monotone access structure and show its unforgeability and robustness under the computational Diffie-Hellman (CDH) assumption in the standard model. For 112-bit security, its secret key shares and signature fragments are as short as 255 bits and 510 bits, which are shorter than existing schemes assuming random oracle. We then propose two extensions. The first one allows new participants to dynamically join the system without any help from the dealer. The second one supports a type of multipartite access structures, where the participant set is divided into multiple disjoint groups, and each group is bounded so that a distributed signature cannot be generated unless a pre-defined number of participants from multiple groups work together.

Keywords: Distributed signature, threshold signature, secret sharing, monotone span program, multipartite access structure, standard model.

* This work is supported by the National Key Basic Research Program (973 program) through project 2012CB315905, by the National Nature Science Foundation of China through projects 61003214, 61173154, 61272501, 61202465 and 61370190, by the Beijing Natural Science Foundation through project 4132056, by the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China and by Open Research Fund of Beijing Key Laboratory of Trusted Computing. Sherman Chow is supported by the Early Career Scheme and the Early Career Award of the Research Grants Council, Hong Kong SAR (CUHK 439713), and grants (4055018, 4930034) from Chinese University of Hong Kong.

J. Benaloh (Ed.): CT-RSA 2014, LNCS 8366, pp. 307–326, 2014.

1 Introduction

A distributed signature scheme [12,24,25] enables a qualified set of participants to jointly generate a signature on a message. The participants have their shares of a (secret) signing key so that each of them can generate a signature fragment for a given message. A full signature can then be reconstructed by collecting the signature fragments from a qualified set of participants. This full signature would be computationally indistinguishable from the one generated directly using the signing key, and it should be unforgeable even if all the participants in the unqualified sets collude together. How the qualified set is represented may differ from construction to construction. The qualified set can be simply a threshold structure (which reduces distributed signature to its special case of threshold signature), or a more general notion of monotone access structure. For instance, a signing key may be shared among three participants $\{p_1, p_2, p_3\}$ such that the minimal qualified sets are $\{p_1, p_2\}$ and $\{p_2, p_3\}$. As $\{p_1, p_3\}$ is unqualified, existing threshold signature schemes does not apply to this distributed setting.

When multiple signers are involved, there are at least two properties we may expect from a distributed signature scheme. First, we want *robustness* such that the full signature can be reconstructed even if there were some invalid signature fragments. Second, it is also desirable if the scheme is *non-interactive* in both signature fragment generation and final signature reconstruction, i.e., every participant can locally compute a signature fragment for any given message, and after all these fragments are collected, reconstruction can take place without further help from any participants.

Non-interactive robust distributed cryptosystem is a useful cryptographic primitive for distributed systems [38]. A canonical example involves issuing signature from a number of parties for security and availability, such as issuing digital certificates and certifying transactions between companies. Daza, Herranz, and Sáez [13] discussed its application in metering, which provides a publicly-verifiable cryptographic proof counting the number of interactions between servers and clients, such as counting the number of visits to a web server (say, for advertisement accounting) by collecting signature fragments from the clients. On the other hand, one may use this scheme in another way. A company can launch a promotion campaign such that users can get reward when they see the ad-banner from this company from a sufficient number of different web sites.

1.1 Our Contributions

There are several known distributed signature schemes [12, 24, 25]. Our work improves the state-of-the-art in a few different dimensions. In detail, our scheme achieves these appealing properties:

1. *Provable Security under Standard Assumption.* Our scheme is proven secure under Computational Diffie-Hellman (CDH) assumption *without* relying on random oracles. Prior to our work, only the RSA-based scheme by Damgård-Dupont [12] is proved secure without random oracles.

2. *Expressive Access Structure.* Our construction is generic and applicable to any linear secret sharing schemes. As a result, our scheme supports expressive access structure since monotone span programs are equivalent to linear secret sharing schemes [4] and every monotone access structure can be realized by a linear secret sharing scheme [24, 43]. Sharing secret key can be tricky. In existing distributed RSA-based signature schemes [12,24], the Euler's totient function of RSA modulus should remain secret even for share-holder. Any non-trivial linear dependence of the rows allows reconstruction of the Euler's totient function. This requires the sub-matrices regarding all unqualified sets in the monotone span program to be full rank.
3. *Practical Efficiency.* Compared to the existing schemes (see Table 1), our scheme is more efficient as its secret key shares and signature fragments are 4 times and 2 times shorter than others, e.g., for 112-bit security, secret key shares and signature fragments of our scheme are as short as 255 bits and 510 bits, respectively. Moreover, our construction is *non-interactive*. All the existing distributed signature schemes [12, 24, 25] are interactive.

1.2 Extensions

We consider two extensions of our proposed schemes.

Dynamic Joining. In some scenarios, e.g., ad-hoc networks, new participants are expected to join the group dynamically. A trivial solution needs the help from a trusted dealer. Our first extension is a threshold signature scheme, which supports dynamic join without the presence of a dealer. A new participant just needs to talk to at least t existing users for a threshold t. To the best of our knowledge, the only known such scheme is proposed by Gennaro et al. under the RSA assumption [21] in random oracle model, yet our scheme is in standard model and more efficient (see Table 2).

Multipartite Access Structures. All participants have the same power in a regular threshold signature scheme. However, in real world applications, participants may be classified by their attributes such as titles, positions, etc., which in turn determine their power in signature generation. In a multipartite access structure [2,3,6,10,17–19,33,41,44], the participant set is divided into multiple disjoint groups and the participants in the same group have the equal power when reconstructing the signature. Obviously, this generalizes the threshold case. In recent years, multipartite access structures have been received considerable attentions, such as compartmented access structures [10, 17, 19, 44, 46, 47], weighted access structures [2,3,33,41], multi-level (hierarchical) access structures [6,10,17–19,44], partially hierarchical access structures [17], etc. For some of these, linear and efficient secret sharing schemes have been found [10, 41, 44, 46, 47].

Our second extension is designed specifically for *compartmented access structure with upper bounds* [17, 46]. There exists a threshold for all the participants, and an upper bound for each separate group, i.e., there is a quorum for signature issuing, but any group can not contribute more than the given upper bound even when all the participants in this group participate.

1.3 Related Work

There are a few different notions of signature related to distributed signatures.

Threshold Signatures. Threshold signatures have been received considerable attentions (e.g., see [11, 14, 15, 21–23, 31, 42, 45]). A signature can be created from the participation of any t or more signers among n potential signers. When $2t - 1 \leq n$, the scheme can be made robust. To realize robustness, Gennaro et al. [22] proposed two approaches to verify RSA signature fragments, which are based on the non-interactive information checking protocol, and undeniable signature requiring interactions between the verifier and the signers. The robust threshold RSA signature schemes also have been discussed in random oracle model [21, 42] and without random oracles [11, 31]. There is also a threshold version for digital signature standard (DSS) signatures [23].

Distributed Signatures. An RSA-based distributed signature scheme for general access structures was proposed by Herranz, Padró and Sáez [24]. An RSA-based scheme in the standard model was given by Damgård and Thorbek [12], which introduced *linear integer secret sharing* to distribute RSA secret keys. Stinson and Strobl [45] generalized discrete-logarithm-based Schnorr's signature [37,40] into a threshold version. Distributed Schnorr's signature was studied by Herranz and Sáez [25], which also served as a building block for constructing distributed proxy signature [25,26]. However, these schemes are analyzed in the random oracle model.

Mesh Signatures. As a generalization of ring signatures, mesh signatures [9] can be generated by a qualified set of valid atomic signatures with anonymity. The only construction known [9] has complexities linear in the number of signers. In fact, the corresponding arborescent monotone access structure is a linear combination of *threshold gates*, as both *AND* and *OR gates* are special cases of threshold access structures. However, this scheme [9] cannot support monotone access structures without arborescent representations. Both distributed signature and mesh signature can be used to express that the signers are from a qualified group. A mesh signature can be generated by a single signer, while a distributed signature is usually generated by multiple signers.

*Attribute-Based Signatures.*In attribute-based signature [32], each signer is assigned with a set of attributes. A signer can generate a signature if the claim-predicate is satisfied by her attributes. Both monotone [29,32] and non-monotone access structures [34,35] can be realized by span programs. Like distributed signatures, the scheme has collusion resistance such that signers cannot create a signature that none of them are qualified to even if they pool their attributes together. Unlike distribute signatures, an attribute-based signature is generated by a single signer (with a qualified set of attributes). The claim-predicate can be different for each signature, which inherently makes the signature more complex, either in terms of signature size or underlying assumption. For example, the schemes of Okamoto-Takashima [34,35], which are based on decisional linear assumption, produce signatures of lengths increase linearly with the complexity of the access structure. The attribute-based signature scheme with threshold access structure due to Herranz et al. [29] is constant-size, yet based on a non-static assumption. Bellare and

Fuchsbauer [8] considered a more general primitive known as *policy-based signature*, which allows a signer to generate signature on some message that fits in some policy, while the privacy of the policy is preserved.

2 Definitions and Security Requirements

2.1 Secret Sharing and Monotone Span Program

In a secret sharing scheme [7,28,41], a *dealer* distributes the *shares* of some secret information to *participants* in such a way that the secret can be recovered when participants in a *qualified set* pool their shares together. An *access structure* is the collection of all qualified sets. An access structure is said to satisfy the *monotone increasing property* if any set that contains a qualified set is also qualified. In this paper, we will require all the secret sharing schemes are *perfect*, that is, unqualified sets cannot get any information about the secret.

Notations. Consider a group of participants $\mathcal{P} = \{p_1, \cdots, p_n\}$. We use Γ to denote the monotone access structure defined on $\mathcal{P}$. Due to its monotone increasing property of Γ, there exists a collection of minimal qualified sets which denoted by $min\Gamma$, such that their proper subsets are not qualified, that is, $\forall A \in min\Gamma$ and $\forall B \subsetneq A$, we have $B \notin \Gamma$. We also use $\overline{\Gamma} = 2^{\mathcal{P}} \setminus \Gamma$ to represent the collection of all unqualified sets of participants where $2^{\mathcal{P}}$ is the power set of $\mathcal{P}$. Clearly, $\overline{\Gamma}$ satisfies monotone decreasing property, i.e., $\forall A \in \overline{\Gamma}$ and $\forall B \subseteq A$, $B \in \overline{\Gamma}$. Similarly, let $max\overline{\Gamma}$ be the collection of all the maximal unqualified sets of participants which are not contained in any other unqualified ones.

Definition 1 (Perfect Secret Sharing [4]). *Let Γ be an access structure defined on a group of participants $\mathcal{P}$. For a secret sharing scheme $\mathcal{S}$ realizing Γ, $\mathcal{S}$ is said to be perfect if the following two properties are satisfied:*

- *for any qualified set $A \in \Gamma$, $\Pr[Re(s_A(k)) = k] = 1$ for every $k \in \mathbb{F}$;*
- *for any unqualified set $B \notin \Gamma$, $\Pr[s_B(k_1) = (s_i)_{p_i \in B}] = \Pr[s_B(k_2) = (s_i)_{p_i \in B}]$ for any two distinct secrets $k_1, k_2 \in \mathbb{F}$, and a list of any possible shares $(s_i)_{p_i \in B}$;*

where $Re(\cdot)$ is the reconstruction function of $\mathcal{S}$ and $s_A(k)$ denotes the shares of the secret k which are assigned to the participants in set A.

In this paper, we are interested in the *linear secret sharing schemes* (LSSS), that is, the shares are calculated by using a linear mapping, and also the secret information can be linearly represented by the shares in any qualified set. In the upcoming sections, we will use *monotone span programs* (MSP) to model linear secret sharing schemes, which was introduced in [30] by Karchmer and Wigderson. In fact, MSP was implicitly proposed before [10] by Brickell which was called *vector space secret sharing scheme*.

Definition 2 (Monotone Span Program [4,30]). *Let $\mathcal{P}$ be a group of participants, a and b be two positive integers. A monotone span program is a quadruple $\mathcal{M} = (\mathbb{F}, \boldsymbol{\tau}, M, \rho)$, where $\mathbb{F}$ is a field, $\boldsymbol{\tau}$ is a target vector in $\mathbb{F}^b$, M is an $a \times b$*

matrix over $\mathbb{F}$ *and* $\rho : \{1, 2, \cdots, a\} \to \mathcal{P}$ *labels each row of* M *by a participant in* $\mathcal{P}$*. The size of* $\mathcal{M}$ *is defined as the row number of* M*. For any set* $P \subseteq \mathcal{P}$*, there is a sub-matrix* M_P *of* M*, which consists of all the rows labeled by the participants in* P*. A set* $P \subseteq \mathcal{P}$ *is accepted by* $\mathcal{M}$ *if the target vector* $\boldsymbol{\tau}$ *can be spanned by the vectors in* M_P*. An access structure* Γ *defined on* $\mathcal{P}$ *is accepted by* $\mathcal{M}$ *if and only if* $\mathcal{M}$ *accepts all the sets* $P \in \Gamma$*.*

It is easy to see that, $\mathcal{M}$ not only defines a linear mapping from the matrix M to the participants in $\mathcal{P}$, but also defines a linear relationship between $\boldsymbol{\tau}$ and each sub-matrix M_P ($P \in \Gamma$) because $\boldsymbol{\tau}$ can be spanned by the rows of M_P. It is well known that, each monotone access structure can be realized by an LSSS [24,43] and MSP is equivalent to LSSS [4,30]. Thus, every monotone access structure can be realized by an MSP, and in such a way that, there may be several rows of M labeled to one participant $p_i \in \mathcal{P}$. However, for convenience to express our results in next sections, we assume there is a one to one correspondence between the rows of M and the participants in $\mathcal{P}$, and will use the vector $\boldsymbol{\omega_i}$ to denote the row of M which labeled by the participant $p_i \in \mathcal{P}$, i.e., $\boldsymbol{\omega_i} = \rho^{-1}(p_i)$.

For other details on secret sharing, the readers can refer to [4].

2.2 Distributed Signature Scheme

We proceed to review the definitions and security model of distributed signature schemes [12,24,25], which are in fact generalizations of threshold signature schemes [22,23]. Besides a group of participants $\mathcal{P} = \{p_1, p_2, \cdots, p_n\}$, we assume there exists a special trusted dealer D and a collector C. Anyone (including that in $\mathcal{P}$) can act as the collector C to run the public-known signature reconstruction algorithm without requiring or producing any secret. We also assume that there exists a secure channel between D and each participant p_i ($1 \le i \le n$), but we do not assume that between any pair of participants (including C). The access structure Γ will be represented by an MSP $\mathcal{M} = (\mathbb{F}, \boldsymbol{\tau}, M, \rho)$, and in which the target vector $\boldsymbol{\tau}$ is implicitly assigned to the dealer D.

Definition 3 (Γ-Distributed Signature Scheme). *Let* $\mathcal{SS} = \langle \mathsf{KGen}, \mathsf{Sig}, \mathsf{Ver} \rangle$ *be a signature scheme and* Γ *be a general monotone access structure realized by MSP on the participant set* $\mathcal{P}$ *and a trusted dealer* D*. A* Γ*-distributed signature scheme* $\mathcal{DS}$ *for* $\mathcal{SS}$ *is a quadruple* $\mathcal{DS} = \langle \mathsf{DKGen}, \mathsf{SFGen}, \mathsf{SReCon}, \mathsf{Ver} \rangle$ *where all algorithms are polynomial-time computable.*

- DKGen: *On input* 1^κ *where* $\kappa \in \mathbb{N}$ *is a security parameter, and access structure* Γ*, the (randomized) distributed signature key generation algorithm, which is carried out by the dealer* D*, computes* $(PK, SK) \leftarrow \mathsf{KGen}(1^\kappa)$*, then generates* n *secret key shares* $(SK_1, \cdots, SK_n)$ *based on* Γ*. This algorithm also publishes some additional verification parameters* VP*. We denote* $(PK, SK_1, \cdots, SK_n, VP) \leftarrow \mathsf{DKGen}(1^\kappa, \Gamma)$*.*
- SFGen: *On input* 1^κ*, a message* m*, secret key share* SK_i*, public key* PK*, and public verification parameters* VP*, the signature fragment generation algorithm, which is carried by each participant* $p_i \in \mathcal{P}$*, generates a signature fragment* σ_i*. We denote* $\sigma_i \leftarrow \mathsf{SFGen}(1^\kappa, m, SK_i, PK, VP)$ *for any* $p_i \in \mathcal{P}$*.*

- SReCon: *On input* 1^κ*, a message* m*, signature fragments* $\{\sigma_i : p_i \in \mathbb{P}\}$ *where* $\mathbb{P} \subseteq \mathcal{P}$*, public key* PK*, public verification parameters* VP*, and access structure* Γ*, the signature reconstruction algorithm is carried out by the collector, reconstructs the signature* σ *from the signature fragments* σ_i*'s based on* Γ*. If the set of signature fragments is unqualified with respect to* Γ*, then outputs* $\perp$*. We denote* $\{\sigma, \perp\} \leftarrow \mathsf{SReCon}(1^\kappa, m, \{\sigma_i : p_i \in \mathbb{P}$ *such that* $\mathbb{P} \subseteq \mathcal{P}\}, PK, VP, \Gamma)$*.*
- Ver: *On input* 1^κ*, a message-signature pair* (m, σ) *and a public key* PK*, the deterministic verification algorithm, which can be carried out by anyone (including who are not in* $\mathcal{P}$*), outputs "1" if* σ *is a valid signature for* m *under the public key* PK*, or "0" otherwise. We denote* $\{1, 0\} \leftarrow \mathsf{Ver}(1^\kappa, m, \sigma, PK)$*.*

When the verification algorithm Ver in $\mathcal{DS}$ is just the same as that in $\mathcal{SS}$, no one can tell whether a signature is generated in a distributed or the typical centralized manner.

A distributed signature scheme is *correct*, if for all messages and all key tuples consisting of public key, secret key shares and public verification parameters, the signatures produced by signature reconstruction algorithm can be verified as valid under the corresponding public key. Formally, the correctness of a Γ-distributed signature scheme $\mathcal{DS} = \langle \mathsf{DKGen}, \mathsf{SFGen}, \mathsf{SReCon}, \mathsf{Ver} \rangle$ can be defined as follows.

Definition 4 (Correctness). *Γ-distributed signature scheme $\mathcal{DS}$ is* ***correct*** *if* $\mathsf{Ver}(1^\kappa, m, \sigma, PK) = 1$ *for any* $(PK, SK_1, \cdots, SK_n, VP) \leftarrow \mathsf{DKGen}(1^\kappa, \Gamma)$*, any* $P \in \Gamma$*, any* $m \in \{0,1\}^\star$*, and any* $\sigma \leftarrow \mathsf{SReCon}(1^\kappa, m, \{\sigma_i \leftarrow \mathsf{SFGen}(1^\kappa, m, SK_i, PK, VP) : p_i \in P\}, PK, VP, \Gamma)$*.*

Definition 5 (Unforgeability). *Given a Γ-distributed signature scheme $\mathcal{DS}$. Suppose $\mathcal{A}$ be a probabilistic polynomial time adversary who controls an unqualified set $P' \in \overline{\Gamma}$ of participants. Consider the following experiment for $\mathcal{A}$:*

- *On input* 1^κ *and* Γ*,* DKGen *is executed to get* (PK, SK) *and* $(SK_1, \cdots, SK_n)$*;*
- $\mathcal{A}$ *is given* 1^κ*,* PK*, and a list of secret key shares which belong to* P'*.*
- $\mathcal{A}$ *adaptively chooses* q_s $(q_s \in \mathbb{N})$ *messages* $m_1, \cdots, m_{q_s}$ *and interacts with* SFGen *and* SReCon *to obtain their signatures* $\sigma_1, \cdots, \sigma_{q_s}$*.*
- $\mathcal{A}$ *outputs a pair* (m, σ)*.* $\mathcal{A}$ *succeeds the game if* $\mathsf{Ver}(1^\kappa, m, \sigma, PK) = 1$ *and* $m \notin \{m_1, \cdots, m_{q_s}\}$*.*

If there is no such adversary $\mathcal{A}$ who could succeed with non-negligible probability in κ, then $\mathcal{DS}$ is said to be ***existentially unforgeable against adaptively chosen message attacks****.*

If an adversary $\mathcal{A}$ controls an unqualified set $P' \in \overline{\Gamma}$ of participants, then $\mathcal{A}$'s view contains not only all the signatures $\sigma_1, \cdots, \sigma_{q_s}$ for the adaptively chosen messages, but also all the intermediate states of the participants in P' and the public outputs on the execution of $\mathcal{DS}$. Furthermore, suppose $\mathcal{A}$ is a *malicious* adversary, then $\mathcal{A}$ can also make participants in P' deviate from the algorithms running, e.g., the corrupted participants can provide invalid signature fragments

for SReCon. If a distributed signature scheme resists such an adversary $\mathcal{A}$, then it is *robust*.

Definition 6 (Robustness). *Given a Γ-distributed signature scheme $\mathcal{DS}$. Suppose $\mathcal{A}$ be a malicious adversary who controls an unqualified set $P' \in \overline{\Gamma}$ of participants. $\mathcal{DS}$ is said to be $\overline{\Gamma}$-**robust** if for any $(PK, SK_1, \cdots, SK_n, VP) \leftarrow \mathsf{DKGen}(1^\kappa, \Gamma)$ and any message $m \in \{0,1\}^\star$, there exists $P \subseteq \mathcal{P} \setminus P'$ such that $\mathsf{Ver}(1^\kappa, m, \sigma, PK) = 1$ for any $\sigma \leftarrow \mathsf{SReCon}(1^\kappa, m, \{\sigma_i' : p_i \in P'\} \cup \{\sigma_i \leftarrow \mathsf{SFGen}(1^\kappa, m, SK_i, PK, VP) : p_i \in P\}, PK, VP, \Gamma)$.*

In fact, all the existing distributed signature schemes [12, 24, 25] are robust. Similar to the threshold cases, there is a requirement on the access structures to implement robust distributed signature schemes. In our case, the union of any two unqualified sets cannot cover the universal set of the participants. Otherwise, after discarding an unqualified set of signature fragments provided by malicious participants, the remaining ones will also be unqualified to recover the signature.

3 Our Basic Scheme

We first briefly review bilinear groups which will be used in our construction.

Definition 7 (Bilinear Groups [20]). *Let q be a prime. Suppose $\mathbb{G}_1$ and $\mathbb{G}_2$ are cyclic groups of order q, and generated by g_1 and g_2, respectively. A group pair $(\mathbb{G}_1, \mathbb{G}_2)$ are said to be bilinear if there exists a cyclic group $\mathbb{G}_T$ and a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ such that:*

1. *Bilinearity: $\forall \mu \in \mathbb{G}_1$, $\forall \nu \in \mathbb{G}_2$, and $\forall a, b \in \mathbb{Z}$, $e(\mu^a, \nu^b) = e(\mu, \nu)^{ab}$;*
2. *Non-degeneracy: $e(g_1, g_2) \neq 1$ and thus is a generator of $\mathbb{G}_T$;*
3. *Efficiency: the map e and the group operations in $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ could be calculated efficiently.*

Our Construction. Our construction is based on the Waters signature scheme [27, 48].Let $\mathbb{G}$ be a group of order q, where q is a prime. For simplicity of presentation we set $\mathbb{G}_1 = \mathbb{G}_2 = \mathbb{G}$ such that $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is an efficient bilinear map. Suppose $H : \{0,1\}^\star \to \{0,1\}^\ell$ is a public collision-resistant hash function, where ℓ is derived from the system security parameter. The access structure Γ is represented by an MSP $\mathcal{M} = (\mathbb{Z}_q, \boldsymbol{\tau}, M, \rho)$.

– DKGen: The dealer randomly chooses a secret key k from $\mathbb{Z}_q$, and also a series of elements $g, g_0, \cdots, g_\ell \in \mathbb{G}$. The public key is a tuple

$$PK = (g, g_0, \cdots, g_\ell, e(g,g)^k).$$

To share the secret key k among the participants in $\mathcal{P}$, the dealer randomly chooses a vector $\boldsymbol{v} \in (\mathbb{Z}_q)^b$ that satisfies $\boldsymbol{v} \cdot \boldsymbol{\tau} = k \mod q$ and calculates the secret key shares as $k_i = \boldsymbol{v} \cdot \boldsymbol{\omega_i} \mod q$ for every $p_i \in \mathcal{P}$. The algorithm also publishes the verification parameters

$$VP = (e(g,g)^{k_1}, \cdots, e(g,g)^{k_n}).$$

– SFGen: All messages are taken as ℓ-bit strings. For any longer messages, hash function H should be applied first on them in order to shorten their length to ℓ. Given a message m denoted by $(\mathfrak{m}_1, \cdots, \mathfrak{m}_\ell)$, the algorithm randomly chooses a value $r_i \in \mathbb{Z}_q$ and generates signature fragment $\sigma_i = (\alpha_i, \beta_i)$ for the participant p_i ($p_i \in \mathcal{P}$) using the secret key share k_i as

$$\alpha_i = g^{k_i} \left(g_0 \prod_{j=1}^{\ell} g_j^{\mathfrak{m}_j} \right)^{r_i}, \quad \beta_i = g^{r_i}.$$

– SReCon: Given a message m, signature fragments $\{\sigma_i : p_i \in \mathbb{P} \text{ such that } \mathbb{P} \subseteq \mathcal{P}\}$, public key $PK = (g, g_0, \cdots, g_\ell, e(g,g)^k)$ and verification parameters $VP = (e(g,g)^{k_1}, \cdots, e(g,g)^{k_n})$, the collector discards all the invalid signature fragments by checking if

$$e(\alpha_i, g) \stackrel{?}{=} e(g,g)^{k_i} e(g_0 \prod_{j=1}^{\ell} g_j^{\mathfrak{m}_j}, \beta_i)$$

holds. If the remaining signature fragments constitute a qualified set P with regard to the access structure Γ, then there exist a series of values $\{d_i \in \mathbb{Z}_q : p_i \in P\}$ which can be efficiently found by solving the system of equations, such that

$$\boldsymbol{\tau} = \sum_{p_i \in P} d_i \boldsymbol{\omega_i} \mod q.$$

Thus, the signature $\sigma = (\alpha, \beta)$ can be reconstructed as follows

$$\alpha = \prod_{p_i \in P} {\alpha_i}^{d_i}, \ \beta = \prod_{p_i \in P} {\beta_i}^{d_i}.$$

Otherwise, outputs $\perp$.

– Ver: Given a message-signature pair $(m, \sigma = (\alpha, \beta))$ and public key $PK = (g, g_0, \cdots, g_\ell, e(g,g)^k)$, checks whether the following equality holds

$$e(\alpha, g) \stackrel{?}{=} e(g,g)^k e(g_0 \prod_{j=1}^{\ell} g_j^{\mathfrak{m}_j}, \beta).$$

If it is true, then the purported signature σ on message m is valid and accepted; otherwise it is invalid.

Theorem 1. *The proposed distributed signature scheme is correct.*

Proof. According to the definition of MSP, $\boldsymbol{\tau}$ can be linearly represented by using all $\boldsymbol{\omega_i}$'s of M_P where P is a qualified set $P \in \Gamma$. Thus, there exists a group of numbers $\{d_i \in \mathbb{Z}_q : p_i \in P\}$ such that $\sum_{p_i \in P} d_i \boldsymbol{\omega_i} = \boldsymbol{\tau} \mod q$ and they can be found by solving linear equations. Furthermore, we know

$$k = \boldsymbol{v} \cdot \boldsymbol{\tau} = \boldsymbol{v} \cdot \left(\sum_{p_i \in P} d_i \boldsymbol{\omega_i} \right) = \sum_{p_i \in P} d_i \left(\boldsymbol{v} \cdot \boldsymbol{\omega_i} \right) = \sum_{p_i \in P} d_i k_i \mod q.$$

Then, the signature $\sigma = (\alpha, \beta)$ can be computed as

$$\alpha = \prod_{p_i \in P} {\alpha_i}^{d_i} = g^{\sum_{p_i \in P} d_i k_i} \left(g_0 \prod_{j=1}^{\ell} g_j^{\mathtt{m}_j} \right)^{\sum_{p_i \in P} d_i r_i} = g^k \left(g_0 \prod_{j=1}^{\ell} g_j^{\mathtt{m}_j} \right)^r,$$

and

$$\beta = \prod_{p_i \in P} {\beta_i}^{d_i} = g^{\sum_{p_i \in P} d_i r_i} = g^r,$$

where $r = \sum_{p_i \in P} d_i r_i \mod q$ is also random because all the r_i's are randomly chosen.

Given a message-signature pair $(m, \sigma = (\alpha, \beta))$ and public key $PK = (g, g_0, \cdots, g_\ell, e(g,g)^k)$, the signature σ can be validated due to the following equalities

$$\begin{aligned} e(\alpha, g) &= e\left(g^k \left(g_0 \prod_{j=1}^{\ell} g_j^{\mathtt{m}_j} \right)^r, g \right) = e(g^k, g) e\left(\left(g_0 \prod_{j=1}^{\ell} g_j^{\mathtt{m}_j} \right)^r, g \right) \\ &= e(g,g)^k e(g_0 \prod_{j=1}^{\ell} g_j^{\mathtt{m}_j}, g^r) = e(g,g)^k e(g_0 \prod_{j=1}^{\ell} g_j^{\mathtt{m}_j}, \beta). \end{aligned}$$

□

3.1 Security Analysis

We first review a modular approach to prove unforgeability of distributed (threshold) signature schemes, which has been used in previous works (e.g., [22–24,42]). In detail, the unforgeability of a Γ-distributed signature scheme can be proved by first showing that the underlying standard signature scheme is unforgeable and then showing that the Γ-distributed signature scheme itself is *simulatable*. A simulatable Γ-distributed signature scheme $\mathcal{DS}$ requires that DKGen, SFGen and SReCon are simulatable for any probabilistic polynomial time adversary $\mathcal{A}$, that is, $\mathcal{A}$'s view on the execution of DKGen, SFGen and SReCon can be efficiently simulated only based on the public key PK and access structure Γ (represented by MSP) of $\mathcal{DS}$.

Definition 8 (Simulatability [22–24,42]). *A Γ-distributed signature scheme $\mathcal{DS} = \langle$DKGen, SFGen,SReCon, Ver$\rangle$ is said to be **simulatable**, if for any probabilistic polynomial time adversary $\mathcal{A}$ who controls an unqualified set $P' \in \overline{\Gamma}$, there exist efficient (polynomial time) algorithms $\mathcal{S}_1$ to simulate $\mathcal{A}$'s view on the execution of* DKGen*, and $\mathcal{S}_2$ to simulate $\mathcal{A}$'s view on the execution of* SFGen *and* SReCon*:*

- *$\mathcal{S}_1$: on input public key PK, corrupted set P', and MSP which represents Γ in $\mathcal{DS}$, can simulate the adversary $\mathcal{A}$'s view on the execution of* DKGen*.*

– $\mathcal{S}_2$*: on input the outputs of* $\mathcal{S}_1$ *(including all the secret information with regard to the corrupted participants in* P'*, e.g., secret key shares), a message-signature pair* (m, σ)*, the public key* PK*, the corrupted set* P' *and MSP which represents* Γ *in* $\mathcal{DS}$*, can simulate the adversary* $\mathcal{A}$*'s view on the execution of* SFGen *and* SReCon *for generating* σ*.*

The next lemma states the requirements for the unforgeability of $\mathcal{DS}$, and will show that holding the view on the executions of DKGen and SFGen is useless for $\mathcal{A}$ to generate a signature forgery. The counterpart of the lemma for threshold signatures is given in [22,23,42]. It was used in distributed signature schemes [24].

Lemma 1. *The* $\mathcal{DS}$ *scheme is also unforgeable if the underlying signature scheme* $\mathcal{SS} = \langle \mathsf{KGen}, \mathsf{Sig}, \mathsf{Ver} \rangle$ *is unforgeable and the corresponding* Γ*-distribute signature scheme* $\mathcal{DS} = \langle \mathsf{DKGen}, \mathsf{SFGen}, \mathsf{SReCon}, \mathsf{Ver} \rangle$ *is simulatable.*

Regarding the security of our scheme, we have the following claim.

Theorem 2. *Let* $\Gamma \subset 2^{\mathcal{P}}$ *be an access structure and* $\mathcal{M} = (\mathbb{Z}_q, \boldsymbol{\tau}, M, \rho)$ *be a monotone span program realizing* Γ*. Then our* Γ*-distributed signature scheme is secure (robust and unforgeable under chosen message attacks) in the standard model, assuming that the underlying Waters signature scheme is unforgeable.*

Proof. It is easy to verify that the robustness can be achieved if $\mathcal{P} \setminus P' \in \Gamma$ for any $P' \in \overline{\Gamma}$. In detail, suppose the adversary $\mathcal{A}$ controls an unqualified set $P' \in \overline{\Gamma}$, then all the invalid signature fragments provided by P' can be detected during the execution of SReCon, and the signature can be reconstructed by $\mathcal{P} \setminus P'$.

For unforgeability, we will give two algorithms $\mathcal{S}_1$ and $\mathcal{S}_2$ to simulate the adversary $\mathcal{A}$'s view when $\mathcal{A}$ controls an unqualified set $P' \in \overline{\Gamma}$, then Lemma 1 can be used accordingly.

$\mathcal{S}_1$ takes the public key $PK = (g, g_0, \cdots, g_\ell, e(g,g)^k)$, the controlled set P' and access structure Γ with an MSP realization $\mathcal{M} = (\mathbb{Z}_q, \boldsymbol{\tau}, M, \rho)$ as input. In the proposed scheme, every participant $p_i \in P'$ holds a secret key share $k_i = \boldsymbol{v} \cdot \boldsymbol{\omega_i} \mod q$, where the vector $\boldsymbol{v}$ is randomly chosen from $(\mathbb{Z}_q)^b$ such that $k = \boldsymbol{v} \cdot \boldsymbol{\tau} \mod q$. As $\overline{\Gamma}$ is monotone decreasing, there exists a maximal unqualified set $\hat{P}' \in max\overline{\Gamma}$ such that $P' \subseteq \hat{P}'$. In order to simulate the adversary $\mathcal{A}$'s view, $\mathcal{S}_1$ randomly chooses a vector $\tilde{\boldsymbol{v}} \in (\mathbb{Z}_q)^b$ and gives every participant $p_i \in \hat{P}'$ a value $\tilde{k_i} = \tilde{\boldsymbol{v}} \cdot \boldsymbol{\omega_i} \mod q$. In fact, the outputs $\{\tilde{k_i} : p_i \in P'\}$ of $\mathcal{S}_1$ are computationally indistinguishable from the real secret key shares $\{k_i : p_i \in P'\}$, because both $\tilde{k_i}$'s and k_i's are uniformly distributed in $\mathbb{Z}_q$. Furthermore, it has been proved [4] that MSP is equivalent to the perfect linear secret sharing scheme, which means that the distribution of $\{k_i : p_i \in \hat{P}'\}$ are perfectly secure with regard to Γ. Thus, $\{\tilde{k_i} : p_i \in \hat{P}'\}$ are also perfectly secure towards the same Γ.

$\mathcal{S}_1$ also calculates the simulated verification parameters $\widetilde{VP}$ which are computationally indistinguishable from the real verification parameters VP. In detail, a part of $\widetilde{VP}$ which related to the participants in $\hat{P}'$ can be calculated as $\{e(g,g)^{\tilde{k_i}} : p_i \in \hat{P}'\}$, while the other ones can be computed as follows. Since $\hat{P}' \in max\overline{\Gamma}$, we know $\hat{P}' \cup \{p_s\} \in \Gamma$ for any participant $p_s \in \mathcal{P} \setminus \hat{P}'$. According

to the definition of monotone span program, $\boldsymbol{\omega_s}$ can be linearly represented by $\boldsymbol{\tau}$ and $\{\boldsymbol{\omega_i} : p_i \in \hat{P}'\}$:

$$\boldsymbol{\omega_s} = d_D\boldsymbol{\tau} + \sum_{p_i \in \hat{P}'} d_i\boldsymbol{\omega_i} \mod q,$$

where d_D and $\{d_i : p_i \in \hat{P}'\}$ are elements in $\mathbb{Z}_q$. Thus, for any participant $p_s \in \mathcal{P} \setminus \hat{P}'$, $\mathcal{S}_1$ computes

$$e(g,g)^{\tilde{k_s}} = e(g,g)^{d_D k + \sum_{p_i \in \hat{P}'} d_i \tilde{k_i}} = \left(e(g,g)^k\right)^{d_D} \cdot \prod_{p_i \in \hat{P}'} \left(e(g,g)^{\tilde{k_i}}\right)^{d_i}.$$

$\mathcal{S}_2$ takes the public key $PK = (g, g_0, \cdots, g_\ell, e(g,g)^k)$, the outputs $\{\tilde{k_i} : p_i \in \hat{P}'\}$ of $\mathcal{S}_1$, the controlled set P', a public known hash function H, and a message-signature pair (m, σ) as input. For each participant $p_i \in \hat{P}'$, $\mathcal{S}_2$ calculates the simulated signature fragment $\tilde{\sigma_i} = (\tilde{\alpha_i}, \tilde{\beta_i})$ as

$$\tilde{\alpha_i} = g^{\tilde{k_i}} \left(g_0 \prod_{j=1}^{\ell} g_j^{\mathtt{m}_j}\right)^{\tilde{r_i}}, \quad \tilde{\beta_i} = g^{\tilde{r_i}},$$

where $\tilde{r_i} \in \mathbb{Z}_q$ is randomly chosen by $\mathcal{S}_2$. Under the above $\widetilde{VP}$, for any participant $p_s \in \mathcal{P} \setminus \hat{P}'$, $\mathcal{S}_2$ can generate a simulated signature fragment $\tilde{\sigma_s} = (\tilde{\alpha_s}, \tilde{\beta_s})$ as

$$\tilde{\alpha_s} = \alpha^{d_D} g^{\sum_{p_i \in \hat{P}'} d_i \tilde{k_i}} \left(g_0 \prod_{j=1}^{\ell} g_j^{\mathtt{m}_j}\right)^{\tilde{r_s}}, \quad \tilde{\beta_s} = \beta^{d_D} g^{\tilde{r_s}},$$

where $\tilde{r_s} \in \mathbb{Z}_q$ is randomly chosen by $\mathcal{S}_2$. As it is easy to check the validity of these simulated signature fragments $\{\tilde{\sigma_i} : p_i \in \mathcal{P}\}$ under the public key PK and the simulated verification parameters $\widetilde{VP}$, $\{\tilde{\sigma_i} : p_i \in \mathcal{P}\}$ are computationally distinguishable from the real signature fragments $\{\sigma_i : p_i \in \mathcal{P}\}$ with regard to the given (m, σ).

Thus, the proposed Γ-distributed signature scheme is simulatable. Since Waters signature scheme is unforgeable under chosen message attacks [27,48], then by Lemma 1, so does our scheme. □

3.2 Comparison

We compare the proposed scheme with existing distributed Schnorr signature scheme [25] and distributed RSA signature schemes [12, 24] in terms of key (share) sizes and signature (fragment) sizes at the same security level. It is well known that the bilinear map can be realized by utilizing pairings on some elliptic curves. We give a comparison according to the key sizes recommended by NIST [1, 20] in Table 1. Consider 112-bit security level, distributed Schnorr signature scheme requires the longest signature fragments (2272 bits), while distributed RSA signature scheme requires the longest secret key shares (2048 bits). Thus, our scheme is more practical with secret key shares and signature fragments as short as 255 bits and 510 bits, respectively.

Table 1. Comparison of distributed signature schemes for $\kappa = 112$ (bits)

Schemes	Key size	Key share size	Signature (fragment) size	Standard model
DL-based [25]	224	448	2272	×
RSA-based [24]	2048	2048	2052	×
RSA-based [12]	2048	2048	2048	✓
Our CDH-based	224-255	224-255	448-510	✓

4 Extensions

In this section, we will give two special extensions of our distributed signature scheme, which can capture some specific requirements in real-world applications.

4.1 Threshold Signatures with Dynamic Addition of Participants

We first give an extension of our distributed signature scheme by using a symmetric bivariate polynomial to share the signing key. Any two secret key shares generated by our scheme are correlated according to the symmetric property. We will give one more algorithm (i.e., PtAdd), which is executed by the new participants to generate his/her secret key share, on inputting the information that generated by other t or more participants. The same technique is used in the scheme of Gennaro et al. [21], which was originally used for admitting node in a short-lived mobile ad hoc network [39].

– DKGen: The dealer randomly chooses a secret key k from $\mathbb{Z}_q$, and also a series of elements $g, g_0, \cdots, g_\ell \in \mathbb{G}$. The public key is a tuple

$$PK = (g, g_0, \cdots, g_\ell, e(g,g)^k).$$

To share the secret key k among the participants in $\mathcal{P}$, the dealer constructs a symmetric bivariate polynomial

$$f(x,y) = \sum_{u=0}^{t-1} \sum_{v=0}^{t-1} c_{u,v} x^u y^v,$$

where the coefficients $c_{u,v}$'s are randomly chosen from $\mathbb{Z}_q$ such that $c_{u,v} = c_{v,u}$, $c_{0,0} = k$ and $c_{t-1,t-1}$ is nonzero. Thus, the secret key shares are computed as $k_i(x) = f(x,i) \mod q$ for all participants $p_i \in \mathcal{P}$. The algorithm also publishes the verification parameters $VP = \{e(g,g)^{k_i(0)} : p_i \in \mathcal{P}\}$.

– PtAdd: When a new participant p_s joining the group, he/she should be given a share of the secret key. In fact, his/her share can be computed with the help of other t participants and do not need the dealer. We assume the new participant p_s received t shares $k_i(s) = f(s,i) \mod q$ from the other parties. Without loss of generality, we assume these values are calculated by the participants in $P = \{p_1, \cdots, p_t\}$, that is, the new participant p_s holds

$$\{k_1(s) = f(s,1), \cdots, k_t(s) = f(s,t)\}.$$

Due to the symmetric property of the bivariate polynomial $f(x, y)$, i.e. $k_i(j) = f(j, i) = f(i, j) = k_j(i) \mod q$, the new participant p_s indeed holds $\{k_s(1) = f(1, s), \cdots, k_s(t) = f(t, s)\}$ and his/her share of the secret key can be calculated by using polynomial interpolation

$$k_s(x) = \sum_{i=1}^{t} \lambda_i(x) k_s(i) \mod q,$$

where $\lambda_i(x) = \prod_{j=1, j\neq i}^{t} \frac{x-j}{i-j} \mod q$. Then, he/she also publishes $e(g, g)^{k_s(0)}$. Thus, the participant set $\mathcal{P}$ and verification parameters VP are dynamically updated.

In fact, the new participant p_s can validate $k_s(0)$ by checking if the following equalities holds

$$e(g, g)^{k_s(0)} = e(g, g)^{\sum_{i=1}^{t} \lambda_i k_i(0)} = \prod_{i=1}^{t} \left(e(g, g)^{k_i(0)} \right)^{\lambda_i},$$

which is due to

$$k_s(0) = f(0, s) = \sum_{i=1}^{t} \lambda_i f(0, i) = \sum_{i=1}^{t} \lambda_i k_i(0) \mod q,$$

where λ_i's are the Lagrange coefficients $\lambda_i = \prod_{j=1, j\neq i}^{t} \frac{s-j}{i-j} \mod q$.

– SFGen: Given a message m denoted by $(\mathfrak{m}_1, \cdots, \mathfrak{m}_\ell)$, the algorithm randomly chooses a value $r_i \in \mathbb{Z}_q$ and generates signature fragment $\sigma_i = (\alpha_i, \beta_i)$ for the participant p_i ($p_i \in \mathcal{P}$) as

$$\alpha_i = g^{k_i(0)} \left(g_0 \prod_{j=1}^{\ell} g_j^{\mathfrak{m}_j} \right)^{r_i}, \quad \beta_i = g^{r_i}.$$

– SReCon: Given a message m, signature fragments $\{\sigma_i : p_i \in \mathbb{P}$ such that $\mathbb{P} \subseteq \mathcal{P}\}$, public key PK, and verification parameters VP, the collector discards all the invalid signature fragments by checking if

$$e(\alpha_i, g) \stackrel{?}{=} e(g, g)^{k_i(0)} e(g_0 \prod_{j=1}^{\ell} g_j^{\mathfrak{m}_j}, \beta_i)$$

holds. If there are remaining t or more valid signature fragments (e.g., $\{\sigma_1, \cdots, \sigma_t\}$), the signature $\sigma = (\alpha, \beta)$ of m can be reconstructed as follows

$$\alpha = \prod_{i=1}^{t} \alpha_i^{\lambda_i}, \ \beta = \prod_{i=1}^{t} \beta_i^{\lambda_i},$$

where λ_i's are the Lagrange coefficients $\lambda_i = \prod_{j=1, j\neq i}^{t} \frac{j}{j-i} \mod q$. Otherwise, outputs $\perp$.

– Ver: Given a message m, a signature $\sigma = (\alpha, \beta)$, and a public key $PK = (g, g_0, \cdots, g_\ell, e(g,g)^k)$, check if the following equality holds

$$e(\alpha, g) \stackrel{?}{=} e(g,g)^k e(g_0 \prod_{j=1}^{\ell} g_j^{\mathfrak{m}_j}, \beta).$$

If it is true, the signature σ is valid; otherwise it is invalid.

As the secret sharing scheme being used in the distributed secret key generation algorithm DKGen is a special linear threshold scheme, thus, according to Theorem 1, we have the following corollary.

Corollary 1. *The above threshold signature scheme is correct.*

Also, according to Theorem 2, we have the following claim.

Corollary 2. *The above threshold signature scheme is secure under CDH assumption in the standard model.*

Table 2 illustrates a performance comparison between our scheme with Gennaro et al.'s scheme [21]. Our scheme has shorter secret key shares and signature fragments, and introduces no additional parameters. As we have noted, due to the Euler's totient function of RSA modulus should keep unknown to all the participants, some additional parameters $(\Delta, \{\delta_i\}, \delta)$ should be introduced for realizing the same functionality in RSA setting.

Table 2. Comparison of threshold signature schemes for $\kappa = 112$ (bits)

	Key share size	Signature fragment size	Additional parameters	Standard model
[21]	$2048t$	2048	Δ, $\{\delta_i\}$, δ	×
Ours	$224t$ to $255t$	448-510	×	✓

4.2 Distributed Signature Scheme for Multipartite Access Structures

In a multipartite access structure, the participant set $\mathcal{P}$ can be divided into u disjoint groups $\mathcal{G}_i$ $(i \in [1,u])$, i.e., $\mathcal{P} = \cup_{i=1}^{u} \mathcal{G}_i$ and $\mathcal{G}_i \cap \mathcal{G}_j = \varnothing$ if $i \neq j$. Furthermore, all participants in the same group $\mathcal{G}_i$ are equally powerful, that is, if a participant $p \in \mathcal{G}_i$ is in a qualified set $P \in \Gamma$, then p can be replaced by any participant $p' \in \mathcal{G}_i \setminus P$.

Our second extension is a distributed signature scheme for *compartmented access structures with upper bounds* [17, 46]. Our construction is based on the linear secret sharing scheme proposed by Tassa and Dyn [46]. Compartmented access structures with upper bounds can be defined as

$$\Gamma = \{\mathcal{I} \subseteq \mathcal{P} : \exists \mathcal{J} \subseteq \mathcal{I} \text{ such that } |\mathcal{J} \cap \mathcal{G}_i| \leq t_i, 1 \leq i \leq u, \text{ and } |\mathcal{J}| = t\},$$

where $1 \leq t \leq \min\{\sum_{i=1}^{u} t_i, n\}$. That is, t_i determines the power of group $\mathcal{G}_i$ ($i \in [1, u]$). As we will see in the following, for the scheme presented by Tassa and Dyn [46], more than t_i participants of $\mathcal{G}_i$ cannot contribute more to recovering the secret.

– DKGen: The dealer randomly chooses a secret key $k \in \mathbb{Z}_q$, and also a series of elements $g, g_0, \cdots, g_\ell \in \mathbb{G}$. The public key is a tuple

$$PK = (g, g_0, \cdots, g_\ell, e(g, g)^k).$$

To share the secret key k among the participants in $\mathcal{P}$, for each group $\mathcal{G}_i$ ($i \in [1, u]$), the dealer first constructs a random univariate polynomial

$$f_i(y) = \sum_{j=0}^{t_i - 1} c_{i,j} y^j$$

over $\mathbb{Z}_q$, where c_{i,t_i-1} is nonzero, and specifies a distinct identity $x_i \in \mathbb{Z}_q$. Furthermore, it is required that

$$\sum_{i=1}^{u} \sum_{j=0}^{t_i - 1} c_{i,j} = k \mod q.$$

Then using Lagrange interpolation to construct

$$f(x, y) = \sum_{i=1}^{u} \lambda_i(x) f_i(y) = \sum_{i=1}^{u} \sum_{j=0}^{t_i - 1} c_{i,j} \lambda_i(x) y^j \mod q,$$

where $\lambda_i(x) = \prod_{h=1, h \neq i}^{u} \frac{x - x_h}{x_i - x_h} \mod q$. The secret key share for participant $p_{i,j} \in \mathcal{G}_i$ ($i \in [1, u]$) can be calculated as $k_{i,j} = f(x_i, y_{i,j}) \mod q$, in which $y_{i,j}$ is the identity of $p_{i,j}$ such that $y_{i,j} \neq 1$. The verification parameters are published as

$$VP = \{e(g, g)^{k_{i,j}} : p_{i,j} \in \mathcal{P}\}.$$

In addition, as $t \leq \sum_{i=1}^{u} t_i$, the dealer should also publish $s = \sum_{i=1}^{u} t_i - t$ secret key shares, that is, the dealer random chooses s different points (x'_i, z'_i) where $x'_i \notin \{x_1, \cdots, x_u\}$ and calculates $k'_i = f(x'_i, z'_i) \mod q$.

– SFGen: Given a message m denoted by $(\mathfrak{m}_1, \cdots, \mathfrak{m}_\ell)$, the algorithm generates a signature fragment $\sigma_{i,j} = (\alpha_{i,j}, \beta_{i,j})$ for the participant $p_{i,j} \in \mathcal{P}$ as

$$\alpha_{i,j} = g^{k_{i,j}} \left(g_0 \prod_{h=1}^{\ell} g_h^{\mathfrak{m}_h} \right)^{r_{i,j}}, \quad \beta_{i,j} = g^{r_{i,j}},$$

where $r_{i,j}$ is randomly chosen from $\mathbb{Z}_q$.

– SReCon: Given a message m, signature fragments $\{\sigma_{i,j} : p_{i,j} \in \mathbb{P}$ such that $\mathbb{P} \subseteq \mathcal{P}\}$, public key $PK = (g, g_0, \cdots, g_\ell, e(g,g)^k)$, and verification parameters $VP = \{e(g,g)^{k_{i,j}} : p_{i,j} \in \mathcal{P}\}$, the collector discards all the invalid signature fragments by checking if

$$e(\alpha_{i,j}, g) \stackrel{?}{=} e(g,g)^{k_{i,j}} e(g_0 \prod_{h=1}^{\ell} g_h^{\mathfrak{m}_h}, \beta_{i,j})$$

holds. If there are remaining t or more valid signature fragments $\sigma_{i,j}$'s (suppose they belong to the participants in $P \subseteq \mathbb{P}$), the signature $\sigma = (\alpha, \beta)$ of m can be reconstructed as follows: because there exist $\sum_{i=1}^{u} t_i$ values of $d_{i,j}$'s and d_i's over $\mathbb{Z}_q$ (which can be efficiently found by solving the system of equations) such that

$$k = \sum_{p_{i,j} \in P} d_{i,j} k_{i,j} + \sum_{i=1}^{s} d_i k'_i \mod q,$$

the signature σ is calculated as

$$\alpha = \prod_{p_{i,j} \in P} {\alpha_{i,j}}^{d_{i,j}} \cdot \prod_{i=1}^{s} (g^{k'_i})^{d_i}, \ \beta = \prod_{p_{i,j} \in P} {\beta_{i,j}}^{d_{i,j}}.$$

Otherwise, outputs $\perp$.

– Ver: Given a message m, a signature $\sigma = (\alpha, \beta)$, and a public key $PK = (g, g_0, \cdots, g_\ell, e(g,g)^k)$, check whether the following equality holds

$$e(\alpha, g) \stackrel{?}{=} e(g,g)^k e(g_0 \prod_{h=1}^{\ell} g_h^{\mathfrak{m}_h}, \beta).$$

If it holds, then the signature σ is valid; otherwise it is invalid.

Tassa and Dyn [46] proved their linear secret sharing scheme for the compartmented access structures with upper bounds is perfect with probability $1 - O(1/q)$. It is easy to rewrite their scheme in a MSP representation, thus, according to Theorem 1 and Theorem 2, we have following corollaries.

Corollary 3. *The above distributed signature scheme for compartmented access structures with upper bounds is correct.*

Corollary 4. *The above distributed signature scheme for compartmented access structures with upper bounds is secure with probability* $1 - O(1/q)$ *under CDH assumption in the standard model.*

5 Conclusion

We proposed a distributed signatures scheme in the standard model based on the CDH assumption. Our scheme offers higher efficiency when compared with

existing schemes in the random oracle model. We also presented two special extensions of our construction. The first one can be used in the situation in which new participants can join the system without the help from a centralized dealer. The second one can be used for a type of multipartite access structures where all the disjoint groups are bounded to jointly generate a signature.

References

1. Barker, E., Barker, W., Burr, W., Polk, W., Smid, M.: Recommendation for Key Management-Part 1: General (Revision 3). NIST Special Publication 800-57, 1-147 (2012), http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general.pdf
2. Beimel, A., Weinreb, E.: Monotone Circuits for Monotone Weighted Threshold Functions. Information Processing Letters 97, 12–18 (2006)
3. Beimel, A., Tassa, T., Weinreb, E.: Characterizing Ideal Weighted Threshold Secret Sharing. SIAM J. Discrete Math. 22, 360–397 (2008)
4. Beimel, A.: Secret-Sharing Schemes: A Survey. In: Chee, Y.M., Guo, Z., Ling, S., Shao, F., Tang, Y., Wang, H., Xing, C. (eds.) IWCC 2011. LNCS, vol. 6639, pp. 11–46. Springer, Heidelberg (2011)
5. Benaloh, J., Leichter, J.: Generalized Secret Sharing and Monotone Functions. In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 27–35. Springer, Heidelberg (1990)
6. Beutelspacher, A., Wettl, F.: On 2-level Secret Sharing. Designs, Codes and Cryptography 3, 127–134 (1993)
7. Blakley, G.R.: Safeguarding Cryptographic Keys. In: National Computer Conference, vol. 48, pp. 313–317. AFIPS Press (1979)
8. Bellare, M., Fuchsbauer, G.: Policy-based Signatures. Cryptology ePrint Archive, Report 2013/413 (2013)
9. Boyen, X.: Mesh Signatures. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 210–227. Springer, Heidelberg (2007)
10. Brickell, E.F.: Some Ideal Secret Sharing Schemes. In: Quisquater, J.-J., Vandewalle, J. (eds.) EUROCRYPT 1989. LNCS, vol. 434, pp. 468–475. Springer, Heidelberg (1990)
11. Damgård, I., Dupont, K.: Efficient Threshold RSA Signatures with General Moduli and No Extra Assumptions. In: Vaudenay, S. (ed.) PKC 2005. LNCS, vol. 3386, pp. 346–361. Springer, Heidelberg (2005)
12. Damgård, I., Thorbek, R.: Linear Integer Secret Sharing and Distributed Exponentiation. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 75–90. Springer, Heidelberg (2006)
13. Daza, V., Herranz, J., Sáez, G.: Protocols Useful on the Internet from Distributed Signature Schemes. Int. J. Inf. Secur. 3, 61–69 (2004)
14. Desmedt, Y.: Society and Group Oriented Cryptography: A New Concept. In: Pomerance, C. (ed.) CRYPTO 1987. LNCS, vol. 293, pp. 120–127. Springer, Heidelberg (1988)
15. Desmedt, Y., Frankel, Y.: Threshold Cryptosystems. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 307–315. Springer, Heidelberg (1990)
16. El Gamal, T.: A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. IEEE Transactions on Information Theory IT-31(4), 469–472 (1985)

17. Farràs, O., Padró, C., Xing, C., Yang, A.: Natural Generalizations of Threshold Secret Sharing. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 610–627. Springer, Heidelberg (2011)
18. Farràs, O., Padró, C.: Ideal Hierarchical Secret Sharing Schemes. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 219–236. Springer, Heidelberg (2010)
19. Farràs, O., Martí-Farré, J., Padró, C.: Ideal Multipartite Secret Sharing Schemes. Journal of Cryptology 25(3), 434–463 (2012)
20. Galbraith, S.D., Paterson, K.G., Smart, N.P.: Pairings for Cryptographers. Discrete Applied Mathematics 156(16), 3113–3121 (2008)
21. Gennaro, R., Halevi, S., Krawczyk, H., Rabin, T.: Threshold RSA for Dynamic and Ad-Hoc Groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 88–107. Springer, Heidelberg (2008)
22. Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T.: Robust and Efficient Sharing of RSA Functions. J. Cryptol. 13, 273–300 (2000)
23. Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T.: Robust Threshold DSS Signatures. Information and Computation 164, 54–84 (2001)
24. Herranz, J., Padró, C., Sáez, G.: Distributed RSA Signature Schemes for General Access Structures. In: Boyd, C., Mao, W. (eds.) ISC 2003. LNCS, vol. 2851, pp. 122–136. Springer, Heidelberg (2003)
25. Herranz, J., Sáez, G.: Verifiable Secret Sharing for General Access Structures, with Application to Fully Distributed Proxy Signatures. In: Wright, R.N. (ed.) FC 2003. LNCS, vol. 2742, pp. 286–302. Springer, Heidelberg (2003)
26. Herranz, J., Sáez, G.: Revisiting Fully Distributed Proxy Signature Schemes. In: Canteaut, A., Viswanathan, K. (eds.) INDOCRYPT 2004. LNCS, vol. 3348, pp. 356–370. Springer, Heidelberg (2004)
27. Hohenberger, S., Waters, B.: Short and Stateless Signatures from the RSA Assumption. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 654–670. Springer, Heidelberg (2009)
28. Ito, M., Saito, A., Nishizeki, T.: Secret Sharing Scheme Realizing General Access Structure. In: IEEE Global Telecommunications Conference, pp. 99–102 (1987)
29. Herranz, J., Laguillaumie, F., Libert, B., Ràfols, C.: Short Attribute-Based Signatures for Threshold Predicates. In: Dunkelman, O. (ed.) CT-RSA 2012. LNCS, vol. 7178, pp. 51–67. Springer, Heidelberg (2012)
30. Karchmer, M., Wigderson, A.: On Span Programs. In: Proc. of the 8th IEEE Structure in Complexity Theory, pp. 102–111 (1993)
31. Li, J., Yuen, T.H., Kim, K.: Practical Threshold Signatures without Random Oracles. In: Susilo, W., Liu, J.K., Mu, Y. (eds.) ProvSec 2007. LNCS, vol. 4784, pp. 198–207. Springer, Heidelberg (2007)
32. Maji, H.K., Prabhakaran, M., Rosulek, M.: Attribute-Based Signatures. In: Kiayias, A. (ed.) CT-RSA 2011. LNCS, vol. 6558, pp. 376–392. Springer, Heidelberg (2011)
33. Morillo, P., Padró, C., Sáez, G., Villar, J.L.: Weighted Threshold Secret Sharing Schemes. Information Processing Letters 70, 211–216 (1999)
34. Okamoto, T., Takashima, K.: Efficient Attribute-Based Signatures for Non-monotone Predicates in the Standard Model. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 35–52. Springer, Heidelberg (2011)
35. Okamoto, T., Takashima, K.: Decentralized Attribute-Based Signatures. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 125–142. Springer, Heidelberg (2013)

36. Padró, C., Sáez, G., Villar, J.L.: Detection of Cheaters in Vector Space Secret Sharing Schemes. Designs, Codes and Cryptography 16(1), 75–85 (1999)
37. Pointcheval, D., Stern, J.: Security Arguments for Digital Signatures and Blind Signatures. Journal of Cryptology 13(3), 361–396 (2000)
38. Qin, B., Wu, Q., Zhang, L., Farràs, O., Domingo-Ferrer, J.: Provably Secure Threshold Public-Key Encryption with Adaptive Security and Short Ciphertexts. Information Sciences 210, 67–80 (2012)
39. Saxena, N., Tsudik, G., Yi, J.H.: Efficient Node Admission for Short-lived Mobile Ad Hoc Networks. In: 13th IEEE International Conference on Network Protocols, ICNP, pp. 269–278 (2005)
40. Schnorr, C.P.: Efficient Signature Generation by Smart Cards. J. Cryptol. 4, 161–174 (1991)
41. Shamir, A.: How to Share a Secret. Commun. of the ACM 22, 612–613 (1979)
42. Shoup, V.: Practical Threshold Signatures. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 207–220. Springer, Heidelberg (2000)
43. Simmons, G.J., Jackson, W.-A., Martin, K.M.: The Geometry of Shared Secret Schemes. Bulletin of the Institute of Combinatorics and Its Applications 1, 71–88 (1991)
44. Simmons, G.J.: How to (Really) Share a Secret. In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 390–448. Springer, Heidelberg (1990)
45. Stinson, D.R., Strobl, R.: Provably Secure Distributed Schnorr Signatures and a (t, n) Threshold Scheme for Implicit Certificates. In: Varadharajan, V., Mu, Y. (eds.) ACISP 2001. LNCS, vol. 2119, pp. 417–434. Springer, Heidelberg (2001)
46. Tassa, T., Dyn, N.: Multipartite Secret Sharing by Bivariate Interpolation. J. Cryptol. 22, 227–258 (2009)
47. Tassa T.: Hierarchical Threshold Secret Sharing. Journal of Cryptology 20, 237–264 (2007)
48. Waters, B.: Efficient Identity-based Encryption without Random Oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)