# Democratic Group Signatures with Threshold Traceability

Dong Zheng[1], Xiangxue Li[2], Changshe Ma[3], Kefei Chen[1], and Jianhua Li[2]

[1] Department of Computer Science and Engineering, Shanghai Jiaotong University
dzheng@sjtu.edu.cn
[2] School of Information Security Engineering, Shanghai Jiaotong University
[3] School of Computer, South China Normal University

**Abstract.** Recently, democratic group signatures(DGSs) particularly catch our attention due to their great flexibilities, *i.e.*, *no group manager*, *anonymity*, and *individual traceability.* In existing DGS schemes, individual traceability says that any member in the group can reveal the actual signer's identity from a given signature. In this paper, we formally describe the definition of DGS, revisit its security notions by strengthening the requirement for the property of traceability, and present a concrete DGS construction with $(t,n)$-*threshold traceability* which combines the concepts of group signatures and of threshold cryptography. The idea behind the $(t,n)$-threshold traceability is to distribute between $n$ group members the capability of tracing the actual signer such that any subset of not less than $t$ members can jointly reconstruct a secret and reveal the identity of the signer while preserving security even in the presence of an active adversary which can corrupt up to $t-1$ group members.

Keywords: democratic group signature, anonymity, traceability, threshold traceability

## 1 Introduction

In 1991, Chaum and Heyst introduced the innovative concept of group signatures [9] that is further studied and improved successively in the literature [1,3,4,7]. A group signature scheme allows any member of a group to digitally sign a document in a manner such that a verifier can confirm that it came from the group, but does not know which individual in the group signed the document. In case of disputes the identity of the signer can be discovered by a *designated group authority*(group manager). Group signature is a very useful tool in real world, such as e-cash, e-voting or attestation in trusted computing group, etc.

We address that most group signature schemes rely on the existence of the group manager who requires the use of certificates to guarantee the authenticity of group signature so that he can trace the real signer from a given signature. However, such roles of the manager in traditional group signatures are not applicable to the following scenario[19]. In the presence of economic globalization joint venture is one of the most common and effective means of conducting business internationally. By building joint ventures companies form strategic alliances that help them to enter new economic markets and further their business goals in a cooperative effort without loosing own independence. Upon building a joint venture company, two or more "parent" companies agree to share capital, technology, human resources, risks and rewards in a formation of a new entity under shared control by a "board of directors", which consists of representatives of "parent" companies. The establishment of such shared control is tricky and relies generally on the "trust, but verify" relationship, *i.e.*, companies trust the information they receive from prospective partners, but it is a good business practice to verify the facts. Traditional group signatures contradicts to the requirements of individual tracing of group signatures and of joint control over the membership in the group. Additionally shared financial control is also an important issue in a joint venture. Shared control means that every member of the board is able to issue payment orders on behalf of the joint venture, but at the same time representatives of other companies, should be able to monitor the accounting to achieve fairness in the spending of shared funds.

To these goals, Manulis [19] proposed a group-oriented signature scheme, called democratic group signature(DGS), which modifies the traditional notion of group signatures by eliminating the role of the group manager. In a DGS scheme, group members can cooperatively initialize and maintain the group without relying on any centralized authority. DGSs provide three security properties: *anonymity, traceability, and unforgeability.* Anonymity requires that the identity of the actual signer is not known to non-members. Traceability requires that the signer's identity can be revealed by (and only by) the group members. Unforgeability requires that only group members are able to sign messages on behalf of the group. (The properties of traceability and unforgeability can be formalized in one definition as shown in [19]).

### 1.1 Related Work

Anonymity is becoming a major concern in many multi-user electronic commerce applications such as e-lotteries, e-cash and online games. Group-oriented signature schemes enable an entity of a group to produce a signature on behalf of the group. There are two major paradigms in anonymous group-oriented signature schemes: group signature as explained above and ring signature(also known as spontaneous anonymous group signature[25]).

Ring signature, introduced by Rivest et al.[21], is a paradigm for achieving 1-out-of-$n$ group signature. In a ring signature scheme, any single user/signer can conscript the public keys of $n$-1 other users to form a group of $n$ members. Then a signature can be generated by that single signer which can be publicly verified to be signed by one of the $n$ group members. But the group formation and the signature generation are both *spontaneous*, meaning that no participation or even knowledge of the other $n$-1 users are needed. The 1-out-of-$n$ signature generated this way is also anonymous(signer indistinguishable). Furthermore, the *anonymity* of ring signature is unconditional (information-theoretic) and exculpable (signer anonymous even after subpoenaing all $n$ secret keys and all communication transcripts), while in the traditional group signature scheme, the group manager has the capability of discovering the identity of the real signer who generated a given group signature.

In threshold ring signature schemes[6, 24], any group of $t$ entities spontaneously conscript arbitrarily $n - t$ entities to generate a publicly verifiable $t$-out-of-$n$ signature on behalf of the whole group, yet the actual signers remain anonymous. The spontaneity of these schemes is desirable for ad-hoc groups such as mobile ad-hoc networks. Like the ring signature, threshold ring signature does not support anonymity revocation mechanism.

Fujisaki and Suzuki described the term of traceable ring signature[13]. In traceable ring signature scheme, anyone who creates two signatures for different messages with respect to the same tag can be traced. As long as a signer does not sign on two different messages with respect to the same tag, the identity of the signer is indistinguishable from any of the possible ring members. Traceable ring signature is also known as one-time anonymous signature.

Group signature and ring signature schemes have been the subject of a great deal of recent work and found many practical applications[15, 26]. Detailed comparison of properties for group signature and ring signature can be found in Fig.1.

### 1.2 Motivation

DGS schemes eliminate the role of the centralized authority group manager. Given a group signature $\sigma$, the identity of the real signer who generated $\sigma$ can be traced by *any member* in the group. We believe that this requirement is somewhat *too loose*.

Again take the example of joint ventures into account. Every member of the board can issue payment orders on behalf of the joint venture. And it is reasonable that the accounting should be monitored to achieve fairness in the spending of shared funds. However, it is unnecessary for them to reveal the actual signer for every accounting as these companies have the "trust" relationship and each one has the will to maximize the interest of the joint venture. Additionally, individual traceability would unduly provide every member in the group with the *unconstrained* use of the capability of tracing. Such action would in turn restrict the decision-making of the members as each individual in the board may trace the accounting and then according to his *short-term* profit judge/critique the member who made the decision.

Motivated by above points, we extend in this paper Manulis's DGS methodology [19, 20] to democratic group signatures with threshold traceability. In our contexts, group member can normally sign any message on behalf of the whole group. Given a signature $\sigma$ of a group of $n$ members, we mean by *threshold traceability* that *in case of dispute*, any subset of not less than $t$ parties can jointly reconstruct a secret and reveal the identity of the real signer who generated $\sigma$.

To the aforementioned goal, we exploit the method of publicly verifiable secret sharing(PVSS) introduced by Stadler[23]. A secret sharing scheme allows to share a secret among several participants such that only certain groups of them can recover it. A publicly verifiable secret sharing scheme is a verifiable secret sharing scheme with the property that the validity of the shares distributed by the dealer can be verified by any party; hence verification is not limited to the respective participants receiving the shares[22].

Combining the functionalities of DGS and of PVSS to realize "democratic group signature with threshold traceability" is our main focus in current work. We address that the so-called threshold traceability requires that for any resulting signature on any message, $t$ out of $n$ members can jointly trace the real signer by co-operatively performing some computation, while any subset of less than $t$ parties can not do this. This potentially means that when signing a message on behalf of the group, the signer has to embed *on the fly* some auxiliary information into the resulting signature. Conversely, if the signer distributes the information among the group *before* he generates the signature, any $t$ parties can then together reconstruct the data on the instant and each of them can *individually* trace the signatures originating from him.

One can see that our scheme complements existing group-oriented cryptographic primitives: group signature, democratic group signature, ring signature, threshold ring signature and one-time anonymous signature. It is an explicit goal that unlike the one-time anonymous signature, our scheme allows the signer to sign any message of his choice, and each time he may be traced by any qualified subset. Worth a special mention is that `threshold` of our terminology is executed in the process of tracing, whereas `threshold` of threshold ring signature is in the signing step[6].

### 1.3 Our Contributions

In this paper we first borrow the definition and security notions for democratic group signature schemes, slightly modified, from Manulis's paper, and then propose a DGS scheme with threshold traceability. Compared with related work, our scheme enjoys the following features:

- There is no need to have a group manager. This is a desirable property since the manager is a centralized authority.
- Our scheme allows anonymity: given a valid group signature, it is infeasible for anyone to identify the actual signer.
- Our scheme allows threshold traceability: given a valid group signature, any qualified subset can trace the real signer who generated it.
- Our scheme is unforgeable: only group members are able to sign messages on behalf of the group.

We illustrate the properties of our scheme in Fig.1 by comparing it with related cryptographic primitives.

### 1.4 Overview

The rest of this paper is organized as follows. Section 2 gives brief introductions to the complexity assumptions on which our scheme based and to a building

| | **group signatures** | **Manulis's democratic group signatures** | **ring signatures** | **our scheme** |
|---|---|---|---|---|
| unforgeability | ✓ | ✓ | ✓ | ✓ |
| anonymity | ✓ | ✓ | ✓ | ✓ |
| manager traceability | ✓ | × | × | ✓[1] |
| individual traceability | × | ✓ | × | ✓[2] |
| threshold traceability | × | × | × | ✓ |

[1,2] Our scheme provides $(t, n)$-traceability. If $t = 1$, our property of threshold traceability leads to individual traceability as emphasized in the work by Manulis. If $t = n$, threshold traceability means that the identity of the true signer can be revealed only if all $n$ members cooperatively perform some computation. We view the joint function of all $n$ members as that of a group manager.

**Fig. 1.** comparisons of properties of group-oriented signatures

block, namely, publicly verfiable secret sharing. We formalize the definition and security notions for DGS in Section 3. Formal notions of security properties include anonymity, unforgeability and threshold traceability. In Section 4, we present a concrete DGS scheme which satisfies our requirements. Section 5 concludes this paper.

**Notations.** Throughout this paper, let $\mathbb{Z}_q$ denote the set $\{0, 1, 2, \cdots, q-1\}$, and $\mathbb{Z}_q^*$ denote $\mathbb{Z}_q \backslash \{0\}$. By $\in_R S$, it means choosing a random element from the set $S$ with a uniform distribution. For an algorithm $\mathcal{A}$, we use $x \leftarrow \mathcal{A}$ to denote that $\mathcal{A}$ is executed on some specified input and its output is assigned to the variable $x$; if $\mathcal{A}$ is a probabilistic algorithm, we write $x \overset{R}{\leftarrow} \mathcal{A}$. Finally, throughout this paper, we often equate a user with his identity.

**Negligible Function.** We say a function $f : \mathbb{N} \rightarrow \mathbb{R}$ is *negligible* if for every constant $c \geq 0$, there exists an integer $k_c$ such that $f(k) < k^{-c}$ for all $k > k_c$.

## 2 Preliminaries

### 2.1 Complexity Assumptions

We first review a few concepts. Let $\mathbb{G}$ be a cyclic group of prime order $q$, $g$ the generator of $\mathbb{G}$. Consider the following problems:

**Computational Diffie-Hellman (CDH) Problem.** The CDH problem in $\mathbb{G}$ is defined as follows: given a 3-tuple $(g, g^a, g^b)$ as input, output $g^{ab} \in \mathbb{G}$. An algorithm $\mathcal{A}$ has advantage $\epsilon$ in solving CDH in $\mathbb{G}$ if

$$\Pr[\mathcal{A}(g, g^a, g^b) = g^{ab}] \geq \epsilon,$$

where the probability is over the random choice of $a, b \in \mathbb{Z}_q$ and of the random bits of $\mathcal{A}$.

**Definition 1.** *We say that the $(t, \epsilon)$-CDH Assumption holds in $\mathbb{G}$ if no t-time algorithm has advantage at least $\epsilon$ in solving the CDH problem in $\mathbb{G}$.*

**Decisional Diffie-Hellman (DDH) Problem.** The DDH problem in $\mathbb{G}$ is defined as follows: given a 4-tuple $(g, g^a, g^b, g^c)$ as input, output `yes` if $ab = c$ and `no` otherwise.

One can easily show that an algorithm for solving CDH in $\mathbb{G}$ gives an algorithm for solving DDH in $\mathbb{G}$. The converse is generally believed to be false. More precisely, we define the advantage of an algorithm $\mathcal{A}$ in deciding the DDH problem in $\mathbb{G}$ as

$$\mathsf{AdvDDH}_{\mathcal{A}} \overset{\mathtt{def}}{=} \left| \begin{array}{l} \Pr[\mathcal{A}(g, g^a, g^b, g^{ab}) = \mathtt{yes} : a, b \overset{R}{\leftarrow} \mathbb{Z}_q] \\ -\Pr[\mathcal{A}(g, g^a, g^b, g^c) = \mathtt{yes} : a, b, c \overset{R}{\leftarrow} \mathbb{Z}_q] \end{array} \right|$$

where the probability is over the uniform random choice of the parameters to $\mathcal{A}$, and over the coin tosses of $\mathcal{A}$. We say that an algorithm $\mathcal{A}$ $(t, \epsilon)$-decides DDH in $\mathbb{G}$ if $\mathcal{A}$ runs in time at most $t$, and $\mathsf{AdvDDH}_{\mathcal{A}}$ is at least $\epsilon$.

**Definition 2.** *We say that the $(t, \epsilon)$-DDH Assumption holds in $\mathbb{G}$ if no t-time algorithm has advantage at least $\epsilon$ in solving the DDH problem in $\mathbb{G}$.*

### 2.2 Publicly Verifiable Secret Sharing Scheme

Secret sharing and its many variations form an important primitive in cryptography. A secret sharing scheme distributes a secret among several participants such that only certain groups of them can recover it. Publicly verifiable secret sharing allows that not just the participants can verify their own shares, but that anybody can verify[22]. Suppose that a dealer D wishes to share a secret value $s$ among $n$ participants $P_1, ...P_n$. Basically, PVSS schemes possess the following structure.

**Initialization** All system parameters are generated as part of the initialization. Each participant $P_i$ taking part in a run of the PVSS scheme registers a public key to be used with a public key encryption method. We assume w.l.o.g. that participants $P_1, ..., P_n$ are the actual participants in the run described below.

**Distribution** The protocol consists of two steps:

1. Distribution of the shares.
   To distribute a secret $s$, the dealer first generates the respective shares $s_i$ for participant $P_i$, for $i = 1, ..., n$, then publishes the encrypted share $E(s_i)$ and a proof to show that $E$ encrypts a share $s_i$.
2. Verification of the shares.
   Knowing the public keys for the encryption $E$, any party may verify the shares by running for each participant $P_i$ a noninteractive verification algorithm on the proof to verify that $E(s_i)$ is a correct encryption of a share for $P_i$. In case one or more verifications fail, the protocol is aborted.

**Reconstruction** The protocol consists of two steps:

1. Decryption of the shares.
   The participants of a qualified subset decrypt their shares $s_i$ from $E(s_i)$. They release $s_i$ plus a proof showing that the released share is correct.
2. Pooling the shares.
   The proofs are used to exclude the participants which are dishonest or fail to reproduce their share $s_i$ correctly. Reconstruction of the secret $s$ can be done from the shares of any qualified set of participants.

As the special protocol by Schoenmakers[22] will be used as a building block in our scheme, we review it below.

**Initialization** Let $\mathbb{G}_q$ denote a group of prime order $q$, $g, G$ independently selected generators of $\mathbb{G}_q$. Participant $P_i$ generates a private key $x_i \in_R \mathbb{Z}_q$ and registers $y_i = G^{x_i}$ as its public key.

**Distribution** The protocol consists of two steps:

1. Distribution of the shares.
   To distribute a secret $G^s$ with $s \in_R \mathbb{Z}_q$ among participants $P_1, ..., P_n$, the dealer picks a random polynomial $p(x)$ of degree $t-1$ with coefficients in $\mathbb{Z}_q$: $p(x) = \sum_{j=0}^{t-1} \alpha_j x^j$, where $\alpha_0 = s$. He keeps $p(x)$ secret but publishes the related commitments $C_j = g^{\alpha_j}$, $0 \le j < t$, and the encrypted shares $Y_i = y_i^{p(i)}$, $1 \le i \le n$. Finally, the dealer produces a proof of knowledge of the unique $p(i)$, satisfying $X_i = g^{p(i)}$, $Y_i = y_i^{p(i)}$, wherein $X_i = \prod_{j=0}^{t-1} C_j^{i^j}$, $1 \le i \le n$.
2. Verification of the shares.
   The verifier first computes $X_i = \prod_{j=0}^{t-1} C_j^{i^j}$, $a_{1i} = g^{r_i} X_i^c, a_{2i} = y_i^{r_i} Y_i^c$, $1 \le i \le n$, then checks that the hash of $X_i, Y_i, a_{1i}, a_{2i}, 1 \le i \le n$, matches $c$.

**Reconstruction** The protocol consists of two steps:

1. Decryption of the shares.
   Each participant first find the share $S_i = G^{p(i)}$ by computing $S_i = Y_i^{x_i^{-1}}$, then publishes $S_i$ plus a proof that the value $S_i$ is a correct decryption of $Y_i$.
2. Pooling the shares.
   Suppose w.l.o.g. that participants $P_i$ produce correct values for $S_i$, for $i = 1, ..., t$. The secret $G^s$ is re-derived as $G^s = \prod_{i=1}^{t} S_i^{\lambda_i}$ where $\lambda_i = \prod_{j \ne i} \frac{j}{j-i}$.

As for its security, we have the following theorem[22].

**Theorem 1.** *Under the Diffie-Hellman assumption, the special PVSS scheme is secure in the random oracle model. That is, (i) the reconstruction protocol results in the secret distributed by the dealer for any qualified set of participants, (ii) any non-qualified set of participants is not able to recover the secret.*

In fact, for the special protocol a stronger result that the participants cannot get any partial information on the secret $G^s$ holds under the DDH assumption.

**Theorem 2.** *Under the DDH assumption and the random oracle assumption, the special PVSS scheme is secure. That is, (i) the reconstruction protocol results in the secret distributed by the dealer for any qualified set of participants, (ii) any non-qualified set of participants is not able to recover any (partial) information on the secret.*

## 3 Framework of Democratic Group Signatures

Current section defines a security model for the democratic group signatures with threshold traceability. We start by formally describing the definition of democratic group signatures.

### 3.1 Democratic Group Signatures

**Definition 3.** *A democratic group signature scheme is a tuple $\Pi$ =*(Setup, KeyGen, Sign, Verify, Trace, VTrace)*:*

**Setup:** *a probabilistic setup algorithm, taking as input the security parameters $k$, returns a public parameter param. We write param $\overset{R}{\leftarrow}$ Setup$(k)$;*

**KeyGen:** *a probabilistic key generation algorithm, taking as input param and a user's identity $ID_i \in \{0,1\}^*$, returns the secret/public key pair $(x_i, y_i)$ to the user. We write $(x_i, y_i) \overset{R}{\leftarrow}$ KeyGen$(ID_i, param)$;*

**Sign:** *a probabilistic signing algorithm, taking as input param, public keys of the group members, a message $m$ and the secret key $x_i$ of the real signer, returns the resulting signature $\sigma$ originating from the member $ID_i$ on behalf of the whole group. We write $\sigma \overset{R}{\leftarrow}$ Sign$(param, y_1, \cdots, y_n, m, x_i)$;*

**Verify:** *a deterministic signature verification algorithm, taking as input param, a message $m$, a candidate signature $\sigma$ on $m$ and the public keys of the participants in the group, returns 1 if $\sigma$ is a valid signature, 0 otherwise. We write (1 or 0) $\leftarrow$ Verify$(param, y_1, \cdots, y_n, m, \sigma)$;*

**Trace:** *a tracing protocol between $t$ out of $n$ group members, say, $ID_1, ..., ID_t$, taking as input param, a message $m$, a candidate signature $\sigma$ on $m$, secret keys of the $t$ participants, and the public keys of the participants in the group, returns the identity $ID_j (1 \leq j \leq n)$ from which $\sigma$ originated together with a proof $\pi$ of this fact. We write $(ID_j, \pi) \leftarrow$ Trace$(param, m, \sigma, x_1, ..., x_t, y_1, ..., y_n)$;*

**VTrace:** *a deterministic signer verification algorithm, taking as input param, an identity $ID_j$, a message $m$, a candidate signature $\sigma$ on $m$, the public keys of the participants in the group, and a candidate proof $\pi$, returns 1 if $(ID_j, \pi) =$ Trace$(param, m, \sigma, x_1, ..., x_t, y_1, ..., y_n)$ and the noninteractive verification algorithm run on $\pi$ is successful, 0 otherwise. We write (1 or 0) $\leftarrow$ VTrace$(param, m, \sigma, y_1, \cdots, y_n, ID_j, \pi)$.*

Consistency requires that $\forall m \in M, j \in \{1, ..., n\}$, Verify$(param, y_1, \cdots, y_n, m, \sigma)$=1 and VTrace$(param, m, \sigma, y_1, \cdots, y_n, ID_j, \pi) = 1$ hold, where $\sigma =$ Sign$(param, y_1, \cdots, y_n, m, x_j)$, $(ID_j, \pi) =$ Trace$(param, m, \sigma, x_1, ..., x_t, y_1, ..., y_n)$ and $M$ denotes the message space.

### 3.2 Security Notions for Democratic Group Signatures

We borrow the security notions for democratic signature schemes, slightly modified, from Manulis's paper. Before giving the security notions for DGS, we consider the following oracle which models the abilities of an adversary against DGS: $\mathcal{S}(\cdot,\cdot)$: a *signing oracle*, on a tuple $\langle ID_i, m\rangle$ comprising a user's identity $ID_i$ and a message $m$, returns a signature $\mathsf{Sign}(param, y_1, \cdots, y_n, m, x_i)$.

Let $\Pi$ = (Setup, KeyGen, Sign, Verify, Trace, VTrace) be a DGS scheme, $n$ a polynomial and $\mathcal{A}$ an adversary attacking the property of anonymity for DGS. Without loss of generality, we suppose hereafter the group $\mathbb{U} = \{ID_1, ID_2, ..., ID_n\}$. Our model takes insider attack into account by allowing the adversary to corrupt some fraction of the members and thereby come into possession of their secret keys. Given an initial information string, the adversary can adaptively ask for the secret key of the members up to $t-1$ out of $n$. $\mathcal{A}$ runs in three stages.

In the select stage the adversary is given an initial information string $I$ and outputs $t-1$ identities which indicate that it wants to corrupt, assumed without loss of generality to be users $ID_{n-t+1}, ..., ID_{n-1}$. In the find stage the adversary is given $I$ and the public keys of the honest members $ID_0, ..., ID_{n-t}$. It outputs two identities, say $ID_0$ and $ID_1$, for the honest users and a message $m \in \{0,1\}^*$. Based on a challenge bit $b$, one of the two identities is selected to yield a challenge signature on the message $m$, which is returned to the adversary, now in its guess stage. Finally $\mathcal{A}$ returns a bit $d$ as its guess of the challenge bit $b$. In each stage the adversary will output state information that is returned to it in the next stage. In the find and guess stages $\mathcal{A}$ is given signing oracles corresponding to the secret keys of the honest users. We now provide a formal definition.

**Definition 4.** *Let $\Pi$ = (Setup, KeyGen, Sign, Verify, Trace, VTrace) be a DGS scheme. For a PPT adversary $\mathcal{A}$, let $n$ be a polynomial, $b \in \{0,1\}$, consider the experiment:*

```
Experiment $\mathbf{Exp}^{AN\text{-}b}_{\mathcal{A},\Pi}(k)$
    $I \overset{R}{\leftarrow} \mathsf{Setup}(k)$;
    $(ID_{n-t+1}, ..., ID_{n-1}; st) \leftarrow \mathcal{A}(select,\ I)$;
    For $i = 0, ..., n-t$ do $(x_i, y_i) \overset{R}{\leftarrow} \mathsf{KeyGen}(I)$ EndFor;
    $(ID_0, ID_1; m; st) \leftarrow \mathcal{A}^{\mathcal{S}(\cdot,\cdot)}(find,\ I, y_0, ..., y_{n-t}; st)$;
    $\sigma \overset{R}{\leftarrow} \mathsf{Sign}(I, y_0, ..., y_{n-t}, m, x_b)$;
    $d \leftarrow \mathcal{A}^{\mathcal{S}(\cdot,\cdot)}(guess,\ \sigma; st)$;
    return $d$.
```

*The advantage of the adversary is defined as*

$$Adv^{AN}_{\mathcal{A},\Pi}(k) \overset{def}{=} |Pr[Exp^{AN\text{-}0}_{\mathcal{A},\Pi}(k) = 0] - Pr[Exp^{AN\text{-}1}_{\mathcal{A},\Pi}(k) = 0]|.$$

*Let $\Pi$=(Setup, KeyGen, Sign, Verify, Trace, VTrace) be a DGS scheme. We say that it is anonymous if the function $Adv^{AN}_{\mathcal{A},\Pi}(k)$ is negligible for any poly(k)-time adversary $\mathcal{A}$ and any polynomial $n$.*

Informally, unforgeability of a DGS scheme is equivalent to the nonexistence of an adversary capable, within the confines of a certain game, of forging a DGS signature that can pass the algorithm Verify after adaptively querying polynomially many democratic group signatures, and of forging a signature that can be traced to an honest member after corrupting some group members. We express unforgeability for DGSs as the following.

**Definition 5.** *We specify two experiments for capturing the following attacks:*

1. *After learning polynomially many signatures, an outsider generates a message-signature pair $(m, \sigma)$ such that $\mathsf{Verify}(param, y_1, \cdots, y_n, m, \sigma)$=1.*
2. *A group member impersonates someone in the group to generate a signature from which that one will be traced to be the real signer.*

*Let $\Pi$ = ($\mathsf{Setup}$, $\mathsf{KeyGen}$, $\mathsf{Sign}$, $\mathsf{Verify}$, $\mathsf{Trace}$, $\mathsf{VTrace}$) be a DGS scheme. For a PPT adversary $\mathcal{A}$ whose goal is to forge a group signature that can be checked as valid, let $n$ be a polynomial, $\mathbb{C}$ the group of corrupted members, $|\mathbb{C}| < n$, $\overline{\mathbb{C}} = \mathbb{U} - \mathbb{C}$, consider the experiments:*

```
Experiment $\boldsymbol{Exp}^{UF1}_{\mathcal{A},\Pi}(k)$
        $I \overset{R}{\leftarrow} \mathsf{Setup}(k)$;
        For $i = 0, ..., n-1$ do $(x_i, y_i) \overset{R}{\leftarrow} \mathsf{KeyGen}(I)$ EndFor;
        $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{S}(\cdot,\cdot)}(I, y_0, ..., y_{n-1})$;
        $\mathcal{A}$ wins if $m^*$ was not queried to the signing oracle,
                    and $\mathsf{Verify}(param, y_1, \cdots, y_n, m^*, \sigma^*)$=1;
        The advantage of the adversary is defined as
                    $Adv^{UF1}_{\mathcal{A},\Pi}(k) \overset{def}{=} Pr[\mathcal{A} \;\; wins]$.

Experiment $\boldsymbol{Exp}^{UF2}_{\mathcal{A},\Pi}(k)$
        $I \overset{R}{\leftarrow} \mathsf{Setup}(k)$;
        $(\mathbb{C}; st) \leftarrow \mathcal{A}(select, I)$;
        For $ID \in \overline{\mathbb{C}}$ do $(x_{ID}, y_{ID}) \overset{R}{\leftarrow} \mathsf{KeyGen}(I)$ EndFor;
        $y \leftarrow \{y_{ID} : ID \in \overline{\mathbb{C}}\}$;
        $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{S}(\cdot,\cdot)}(I, y; st)$;
        $\mathcal{A}$ wins if $m^*$ was not queried to the signing oracle,
                    $\mathsf{Verify}(param, y_1, \cdots, y_n, m^*, \sigma^*)$=1,
                    and algorithm $\mathsf{Trace}$ traces $\sigma^*$ to some member in $\overline{\mathbb{C}}$;
        The advantage of the adversary is defined as $Adv^{UF2}_{\mathcal{A},\Pi}(k) \overset{def}{=} Pr[\mathcal{A} \;\; wins]$.
```

*We denote by*

$$Adv^{UF}_{\mathcal{A},\Pi}(k) = Adv^{UF1}_{\mathcal{A},\Pi}(k) + Adv^{UF2}_{\mathcal{A},\Pi}(k)$$

*the advantage of an adversary in breaking the unforgeability of $\Pi$. We say that $\Pi$ is unforgeable if the function $Adv^{UF}_{\mathcal{A},\Pi}(k)$ is negligible for any poly(k)-time adversary $\mathcal{A}$ and any polynomial $n$.*

To describe proper security definitions for DGS, we should also consider its traceability. We revisit the requirement of individual traceability described in [19] by presenting the term threshold traceability. We express $(t, n)$ threshold traceability for DGSs as the following.

**Definition 6.** *Let $\Pi = ($Setup, KeyGen, Sign, Verify, Trace, VTrace$)$ be a DGS scheme. We say that $\Pi$ is $(t, n)$ traceable if given a valid signature: (1) any $t$ out of $n$ members can perform* Trace *algorithm and reveal the identity of the real signer; and (2) any $t-1$ out of $n$ members can not do this.*

What interest us is to combine group signature and PVSS so that we can embed into the resulting signature the on-the-fly tracing trapdoor from which a qualified subset of the members can successfully perform the tracing computation. To gain such a goal is far from trivial. The coming sections do this.

## 4 The Proposed Democratic Group Signature Scheme

We describe the construction for $(t, n)$-threshold tracing. Our solution follows the ideas as explained in the first section and is presented in detail below.

**Setup:** given a security parameter $\kappa$, the algorithm:
- generates a group $\mathbb{G}$ of prime order $q$;
- picks two group generators $g, h \in \mathbb{G}$ independently(hence no one knows the discrete log of $g$ with respect to $h$);
- chooses a cryptographically secure hash function $H : \{0,1\}^* \rightarrow \{0,1\}^\kappa$;
- defines and returns $param = (\mathbb{G}, q, g, h, H)$.

**KeyGen:** this randomized algorithm takes as input a parameter $n$, the number of members of the group, and proceeds as follows. With the group $\mathbb{G}$ and a generator $h \in \mathbb{G}$ as above, a user $ID_i$ generates a private key $x_i \in \mathbb{Z}_q^*$, and registers $y_i = h^{x_i}$ as its public key. Assume that $n$ participants in the group are $ID_i$ whose key pairs are $(x_i, y_i)(i = 1, ..., n)$ respectively.

**Sign:** given the group $\{ID_1, ID_2, ..., ID_n\}$, a user's key $x_k$, and a message $m \in \{0,1\}^*$, the process of signature generation consists of two steps:

(1) Distribution of the tracing trapdoor contained in signing.
Acting as a dealer, the member with identity $ID_k$ wishes to distribute a on-the-fly secret among the signing group. With $s \in_R \mathbb{Z}_q$, he first picks a random polynomial $p(x)$ of degree $t-1$ with coefficients in $\mathbb{Z}_q$: $p(x) = \sum_{j=0}^{t-1} \alpha_j x^j$ wherein $\alpha_0 = s$. $ID_k$ keeps this polynomial secret but publishes the related commitments $\tau = g^{\alpha_0}$, $\tau_j = g^{\alpha_j}$, for $1 \leq j < t$, and the encrypted shares $\eta_i = y_i^{p(i)}$, for $1 \leq i \leq n$. Let $\chi_i = \prod_{j=0}^{t-1} \tau_j^{i^j}$. To show that the values $\eta_i$ are consistent, $ID_k$ produces a proof[5] $ZKPK[p(i) : \chi_i = g^{p(i)} \bigwedge \eta_i = y_i^{p(i)}], 1 \leq i \leq n$. More concretely, the dealer outputs $a_{i1} = g^{w_i}, a_{i2} = y_i^{w_i}$, $e = H(\chi_1, \cdots, \chi_n, \eta_1, \cdots, \eta_n, a_{11}, \cdots, a_{n1}, a_{12}, \cdots, a_{n2})$ and $r_i = w_i - p(i)e$ by using $w_i \in_R \mathbb{Z}_q$, for $1 \leq i \leq n$. For clarity, we define the vector $\mathsf{share} = (\tau, \tau_1, \cdots, \tau_{t-1}, \eta_1, \cdots, \eta_n, e, r_1, \cdots, r_n)$.

(2) Signing integrated with the tracing trapdoor.

- Set $\gamma = g^{x_k}$.
- Specify $c = h^s y_k$. (This can be viewed as ElGamal encryption of $y_k$ under the pair $(g, h)$.)
- For $i \neq k$, select $z_{i1}, z_{i2}, \rho_i$ at random from $\mathbb{Z}_q$ and set $l_{i1} = H(m, \tau, \frac{c}{y_i})$, $l_{i2} = H(m, \gamma, y_i)$, $u_{i1} = (g^{l_{i1}} h)^{z_{i1}} (\tau^{l_{i1}} \frac{c}{y_i})^{\rho_i}$, $u_{i2} = (h^{l_{i2}} g)^{z_{i2}} (y_i^{l_{i2}} \gamma)^{\rho_i}$.
- Select $r_{k1}, r_{k2}$ at random from $\mathbb{Z}_q$ and set $l_{k1} = H(m, \tau, h^s)$, $l_{k2} = H(m, \gamma, y_k)$, $u_{k1} = (g^{l_{k1}} h)^{r_{k1}}$, $u_{k2} = (h^{l_{k2}} g)^{r_{k2}}$.
- Set $\rho_k = H(m, \tau, c, \gamma, u_{11}, \cdots, u_{n1}, u_{12}, \cdots, u_{n2}) - \sum_{j \neq k} \rho_j$, and $z_{k1} = r_{k1} - \rho_k s$, $z_{k2} = r_{k2} - \rho_k x_k$.
- Define the vector $\mathsf{sig} = (c, \gamma, \rho_1, \cdots, \rho_n, z_{11}, \cdots, z_{n1}, z_{12}, \cdots, z_{n2})$, and output the signature $\sigma$, computed as $\sigma = \langle \mathsf{share}, \mathsf{sig} \rangle$.

Remark 1. *We address that the signature obtained from the* Sign *algorithm is a signature of knowledge[8, 10, 16]:* $SK[(x_k, s) : \tau = g^s \bigwedge c = h^s y_k \bigwedge (y_1 = h^{x_1} \bigvee \cdots \bigvee y_n = h^{x_n})](m)$.

Remark 2. *Our main interest is to gain the threshold traceability for DGS schemes, so we do not here take into account the constant-size group-oriented signatures[14]. On second thoughts, as a valid signature $\sigma$ generated by our trick consists of a pair* $\langle$share, sig$\rangle$ *wherein the vector* share *is for the algorithms* Verify, Trace *and* VTrace*, it seems so far that the size of $\sigma$ is unlikely constant, even the component* sig *is short.*

Remark 3. *To get our goal, it is possible to use threshold broadcast encryption(TBE) schemes[11], besides PVSS. However, the performance of this potential trick needs careful discussion as some TBE schemes[11] further uses threshold secret sharing to be a building block.*

**Verify:** given a signing group $\{ID_1, ..., ID_n\}$, along with *param* and a pair $(m, \sigma)$, verification that $\sigma$ is a valid signature generated by the group consists of two steps as follows:

(1) Verification of the shares.
The verifier computes $\chi_i = \prod_{j=0}^{t-1} \tau_j^{i^j}$ from the $\tau_j$ values, $1 \leq i \leq n$. Using $y_i, \chi_i, \eta_i, r_i$, $1 \leq i \leq n$ and $e$ as input, the verifier computes $a_{i1}, a_{i2}$ as $a_{i1} = g^{r_i} \chi_i^e, a_{i2} = y_i^{r_i} \eta_i^e$, and checks whether the hash value of $\chi_i, \eta_i$, $a_{i1}, a_{i2}$, for $1 \leq i \leq n$, matches $e$. If yes, continue; stop otherwise.

(2) Verification of the signature.
First re-derive $u_{i1} = (g^{l_{i1}} h)^{z_{i1}} (\tau^{l_{i1}} \frac{c}{y_i})^{\rho_i}, u_{i2} = (h^{l_{i2}} g)^{z_{i2}} (y_i^{l_{i2}} \gamma)^{\rho_i}$, wherein $l_{i1} = H(m, \tau, \frac{c}{y_i})$ and $l_{i2} = H(m, \gamma, y_i)$, for $1 \leq i \leq n$, then check that the hash of $m, \tau, c, \gamma$, and $u_{i1}, u_{i2} (1 \leq i \leq n)$, matches the sum of $\rho_i, 1 \leq i \leq n$.

**Trace:** given a signing group $\{ID_1, ..., ID_n\}$, along with *param* and a pair $(m, \sigma)$, suppose w.l.o.g that $ID_i, 1 \leq i \leq t$, wish to trace $\sigma$ to an honest member of the group. Note that if the signature is not valid, we can surely turn to the direct discard. Detailed steps proceed as follows:

(1) Signature verification.
Call the Verify algorithm as a subroutine to make sure that the signature is valid, *i.e.*, $\mathsf{Verify}(param, y_1, \cdots, y_n, m, \sigma) = 1$.

(2) Reconstruction of the tracing trapdoor.
Using its private key $x_i$, each participant finds the share $\xi_i = h^{p(i)}$ from $\eta_i$ by computing $\xi_i = \eta_i^{x_i^{-1}}$. They publish $\xi_i$ plus a proof $\pi_i$ that the value $\xi_i$ is a correct decryption of $\eta_i$: $\pi_i = ZKPK[x_i : \eta_i = \xi_i^{x_i} \wedge y_i = h^{x_i}]$. Finally, the secret $\mu = h^s$ is obtained by Lagrange interpolation $h^s = \prod_{i=1}^{t} \xi_i^{\lambda_i}$ where $\lambda_i = \prod_{j \neq i} \frac{j}{j-i}$ is a Lagrange coefficient.

(3) Revealing identity.
With the secret $\mu$, locate in the signing group the identity, say, $ID_k$ whose public key matches $c\mu^{-1}$. Output $ID_k$ along with the vectors $\mathsf{proof} = (\pi_1, \cdots, \pi_t)$ and $\mathsf{dec} = (\xi_1, ..., \xi_t)$.

**VTrace:** given a signing group $\{ID_1, ..., ID_n\}$, $param$, $(m, \sigma)$, $ID_k$ along with the vectors proof, dec, the algorithm proceeds as follows. First, make sure that the signature is valid and that the identity $ID_k$ belongs to the signing group, and then use proof to verify the consistency of dec. If such an event occurs, then re-derive $\mu$ as in the Trace algorithm. Finally, output 1 if $c = \mu y_k$ holds, and 0 otherwise.

This ends the description of our detailed construction. It is easy to check that the requirements for consistency are satisfied, *i.e.*, if $\sigma$ is generated correctly and the algorithm Trace is run successfully, then the outputs of algorithms Verify and VTrace are 1.

Remark 4. *There are many situations in which PVSS schemes can be applied, such as electronic voting, threshold binding ElGamal, threshold revocable electronic cash, threshold software key escrow, etc. Here we offer a fresh application setting for PVSS and then obtain $(t, n)$-threshold tracing for DGS scheme.*

Remark 5. *For the property of traceability, our scheme is very general. More specially, if $t = 0$, this leads to the standard definition of untraceability for ring signatures; if $t = 1$, then we get a DGS with individual traceability proposed by Manulis; if $t = n$, then the resulting DGS is traceable as long as the whole group co-operatively perform the tracing computation. Practical systems can keep the proper capability of tracing by adaptively specifying the value of $t$.*

## 5 Conclusion

In this paper, we introduced a democratic group signature scheme with threshold traceability. By using this kind of signatures, any member of a group can sign any message on behalf of the whole group so that he remains anonymous to any verifier outside of the group and so that he will be inescapably traced by any qualified subset of the group. The scheme gives a more appropriate option to handle the shared financial control in joint ventures companies or other scenarios where “trust, but verify” relationship exists. Our design trick is to use a special PVSS as a building block. The proposed scheme provides a fresh application of

PVSS in the sense that PVSS is traditionally used as a secret sharing primitive or in Electronic Voting, threshold binding ElGamal situations[22].

Additionally, one can observe that our scheme allows anyone to determine whether two signatures of the same group have been issued by the same group member, which can be achieved via the elements $\gamma$ of the given signatures. This results in the property of linkability that is very useful in many settings[17]. We omit the formalization of linkability as it is not our main focus. By applying our trick, it is not hard to design a DGS scheme with threshold traceability which has no linkability property.

As can be seen, the property of threshold traceability is at the cost of signature size. It seems so far unlikely to make the resulting signature constant[12] since we need to share a secret with which any qualified subset can trace the actual signer. Another important issue worth to further discuss is the security reduction which is asymptotic and loose in our theorems. It is interesting to design an efficient DGS with threshold traceability and with more satisfactory (ideally tight) security reduction.

## References

1. Bellare M., Micciancio D., Warinschi B.: Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. Eurocrypt 2003, Lecture Notes in Computer Science, vol 3027, 614-629, Springer-Verlag(2003).
2. Bellare M., Palacio A.: GQ and Schnorr Identification Schemes: Proofs of Security against Impersonation under Active and Concurrent Attacks. Crypto 2002, Lecture Notes in Computer Science, vol 2442, 162-177, Springer-Verlag(2002).
3. Bellare M., Shi H., Zhang C.: Foundations of group signatures: the case of dynamic groups, CT-RSA 2005, Lecture Notes in Computer Science, vol 3376, 136-153, Springer-Verlag(2005).
4. Boneh D., Boyen X., Shacham H.: Short group signatures. Crypto 2004, Lecture Notes in Computer Science, vol 3152, 41-55, Springer-Verlag(2004).
5. Bresson E., Stern J.: Proofs of knowledge for non-monotone discrete-log formulae and applications. ISC 2002, Lecture Notes in Computer Science, vol 2433, 272-288, Springer-Verlag(2002).
6. Bresson E., Stern J., Szydlo M.: Threshold Ring Signatures and Applications to Ad-hoc Groups. Crypto 2002, Lecture Notes in Computer Science, vol 2442, 465-480, Springer-Verlag(2002).
7. Camenisch J., Stadler M.: Efficient group signature schemes for large groups. Crypto 1997, Lecture Notes in Computer Science, vol 1294, 410-424, Springer-Verlag(1997).
8. Camenisch J., Stadler M.: Proof systems for general systems of discrete logarithms. ETH Technical Report, available at: ftp://ftp.inf.ethz.ch/pub/publications /techreports/ (1997).
9. Chaum D., Heyst E.: Group signatures. Eurocrypt 1991, Lecture Notes in Computer Science, vol 547, 257-265, Springer-Verlag(1991).
10. Cramer R., Damgard I. and Schoenmakers B. Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. Crypto 1994, 174-187, Springer-Verlag(1994).

11. Daza V., Herranz J., Morillo P. and Rafols C. CCA2-secure threshold broadcast encryption with shorter ciphertexts. Provable Security 2007, LNCS4784, 35-50, Springer-Verlag(2007).
12. Dodis Y., Kiayias A., Nicolosi A., Shoup V.: Anonymous Identification in Ad Hoc Groups. Eurocrypt 2004, Lecture Notes in Computer Science, vol 3027, 609-626, Springer-Verlag(2004).
13. Fujisaki E., Suzuki K.: Traceable Ring Signature. Public Key Cryptography 2007, Lecture Notes in Computer Science, vol 4450, 181-200, Springer-Verlag(2007).
14. Groth, J.: Fully Anonymous Group Signatures without Random Oracles. Asiacrypt 2007, Lecture Notes in Computer Science, vol 4833, 164-180, Springer-Verlag(2007).
15. Herranz J., Saez G.: A provably secure ID-based ring signature scheme. Cryptology ePrint Archive, Report 2003/261, available at:http://eprint.iacr.org.
16. Lipmaa H.: Proofs of knowledge of certain problems, available at: http://www.cs.ut.ee/helger/crypto/link/zeroknowledge/pok.php.
17. Liu J., Wong D.: Linkable Ring Signatures: Security Models and New Schemes. ICCSA 2005, Lecture Notes in Computer Science, vol 3481, 614-623, Springer-Verlag(2005).
18. Liu J., Wei V., Wong D.: Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract) ACISP 2004, Lecture Notes in Computer Science, vol 3108, 325-335, Springer-Verlag(2004).
19. Manulis M.: Democratic group signature on example of joint ventures, ACM Symposium on Information, Computer and Communications Security(ASIACCS'06). ACM Press, 2006. Full version at: http://eprint.iacr.org/2005/446.
20. Manulis M., Sadeghi A., Schwenk J.: Linkable democratic group signatures. ISPEC 2006, Lecture Notes in Computer Science, vol 3903, 187-201, Springer-Verlag(2006).
21. Rivest R., Shamir A., Tauman Y.: How to leak a secret. Asiacrypt 2001, Lecture Notes in Computer Science, vol 2248, 552-565, Springer-Verlag(2001).
22. Schoenmakers B.: A simple publicly verifiable secret sharing scheme and its application to electronic voting. Crypto 1999, Lecture Notes in Computer Science, vol 1666, 148-164, Springer-Verlag(1999).
23. Stadler M.: Publicly verifiable secret sharing. Eurocrypt 1996, Lecture Notes in Computer Science, vol 1070, 190-199, Springer-Verlag(1996).
24. Tsang P., Wei V., Au M.: Separable linkable threshold ring signatures. Indocrypt 2004, Lecture Notes in Computer Science, vol 3348, 384-298, Springer-Verlag(2004).
25. Wei V.: A bilinear spontaneous anonymous threshold signature for ad hoc groups, Cryptology ePrint Archive, Report 2004/039, available at: http://eprint.iacr.org/.
26. Zhang F., Kim K.: ID-Based blind signature and ring signature from pairings. Asiacrypt 2002, Lecture Notes in Computer Science, vol 2501, 533-547, Springer-Verlag(2002).