Traceable Ring Signature

Eiichiro Fujisaki and Koutarou Suzuki

NTT Information Sharing Platform Laboratories, NTT Corporation,
1-1 Hikari-no-oka, Yokosuka, Kanagawa, 239-0847 Japan,
{fujisaki.eiichiro, suzuki.koutarou}@lab.ntt.co.jp

Abstract. The ring signature allows a signer to leak secrets anonymously, without the risk of identity escrow. At the same time, the ring signature provides great flexibility: No group manager, no special setup, and the dynamics of group choice. The ring signature is, however, vulnerable to malicious or irresponsible signers in some applications, because of its anonymity. In this paper, we propose a traceable ring signature scheme. A traceable ring scheme is a ring signature except that it can restrict "excessive" anonymity. The traceable ring signature has a *tag* that consists of a list of ring members and an *issue* that refers to, for instance, a social affair or an election. A ring member can make any signed but anonymous opinion regarding the issue, but only once (per tag). If the member submits another signed opinion, possibly pretending to be another person who supports the first opinion, the identity of the member is immediately revealed. If the member submits the same opinion, for instance, voting "yes" regarding the same issue twice, everyone can see that these two are linked. The traceable ring signature can suit to many applications, such as an anonymous voting on a BBS. We formalize the security definitions for this primitive and show an efficient and simple construction in the random oracle model.

1 Introduction

A ring signature scheme allows a signer to sign a message while preserving anonymity behind a group, called a "ring," which is selected by the signer. A verifier can check the validity of the signature, but cannot know who generated it among all possible ring members. In addition, two signatures generated by the same singer are unlinkable. Namely, it is infeasible for the verifier to determine whether the signatures are generated by the same signer. This notion was first formally introduced by Rivest, Shamir, and Tauman [24], and since then, this topic has been studied extensively in [19, 6, 1, 17, 16, 4], for instance. The ring signature is related to the notion of group signature, due to [10]. In the group signature, however, there is a group manager that has the power to revoke the anonymity of any signer if necessary. The group manager must also establish a special type of key assignment to create a group, and hence it is difficult to change the group dynamically. Some people say that the group manager is too strong because he can even revoke the anonymity of a honest signer. On the other hand, a ring signature scheme has no group manager, no special setup,

and allows ad-hoc group formation. In addition, a ring signature scheme is free from the risk of identity escrow.

Anonymity is not always good, however. While the group signature has too strong a traceability characteristic, an ordinary ring signature scheme has nothing at all to restrict anonymity. In this paper, we consider a ring signature scheme with a "gentle" anonymity restriction, which only prohibits "excessive" anonymity in some applications. Informally, we consider "one-more unforgeability" and "double-spending traceability" in the context of a ring signature.

Initially, these two notions appeared in the context of a blind signature scheme and a restricted blind signature scheme, as in [7] and [9], respectively. In the blind signature scheme, a user interacts with a signer a number of times and has the signer sign a blind message (In this stage, the signer may know the identity of the user, but not know the contents of the message). After the user transformed it to a "blind" signature, it cannot be traced to the user even by the signer. However, the user who obtained the blind signature from the signer cannot generate a "one-more" new signature. This property is called one-more unforgeability. The restricted blind signature has an additional property called double-spending, so that if a user "spends" a signature twice, he can be traced later [9, 22, 5]. Such a property can be used in the "off-line" anonymous e-cash systems. Note that the identity of a honest user is not threatened, even by the signer.

We incorporate these properties into the ring signature by introducing formal security requirements.

1.1 Our Contribution: Formalization and Construction

In this paper, we introduce the concept of a traceable ring signature. It preserves the flexibility of the ring signature: No group manager, no special setup for sharing secrets among members in a group, and the dynamics of group choice. It implies that the identity of a signer is never escrowed by a special person or group. A traceable ring signature has a *tag* $L = (issue, pk_N)$, where pk_N is the set of public keys of the ring members and *issue* refers to, for instance, an id of an election or some social issue. A ring member can sign a message using his own secret key and the verifier can verify the signature on the message with respect to tag L, but cannot know who generated the signature among all the possible ring members in L. If the signer signed the same message again with the same tag, everyone can see that the two signatures are linked, whereas if he signed a different message with the same tag, then not only is it evident that they are linked, but the anonymity of the signer is revoked. Informally, the security requirements we provide for this primitive are as follows:

- **Public Traceability -** Anyone who creates two signatures for different messages with respect to the same tag can be traced, where the trace can be done only with pairs of message/signature pairs and the tag.
- **Tag-Linkability (One-more unforgeability) -** Every two signatures generated by the same signer with respect to the same tag are linked, that is,

the total number of signatures with respect to the same tag cannot exceed the total number of ring members in the tag, if every any two signatures are not linked.

- **Anonymity -** 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. In addition, any two signatures generated with respect to two distinct tags are always unlinkable. Namely, it is infeasible for anyone to determine whether they are generated by the same signer.
- **Exculpability -** A honest ring member cannot be accused of signing twice with respect the same tag — Namely, an adversary cannot produce a traceable ring signature such that, along with one generated by the target, it can designate the target member in the presence of the publicly traceable mechanism. This should be infeasible even after the attacker has corrupted all ring members but the target.

The above security goals must be preserved under the so-called *adversarially-chosen key and sub-ring attack*, which Bender, Katz, and Morselli have formally addressed in [4]. In addition, our security model follows [4] in the sense that the role of PKI is minimal, namely it only maintains the global public-key list properly, which implies that malicious PKI can't harm a honest signer.

On one hand, our security goals are related to those of the group signature [3]. We stress that the standard unforgeability requirement (as in an ordinary ring signature) is unnecessary for the traceable ring signature because the combined requirements for tag-linkability and exculpability imply unforgeability. We discuss this issue later.

We show how to construct an efficient and conceptually-simple traceable ring signature scheme on an ordinary Abelian group, on which the DDH and discrete logarithm problems are hard, by using the Fiat-Shamir transformation.

1.2 Applications

There are several applications for the traceable ring signature.

An anonymous voting on a BBS - Suppose that some group of people is discussing some issue on a bulletin board via the Internet and wish to vote anonymously among themselves on that issue. They could write to the bulletin board anonymously; however, they do not want to engage a trusted party or establish a heavy setup protocol just for this vote. In addition, it is expected that some people in the group won't vote. An ordinary ring signature cannot be used here because it cannot restrict a member to only one vote. A traceable ring signature however can be applied to this case [1].

[1] We are aware of the fact that public traceability makes any anonymous signature primitive lose the deniability property as discussed in Sec. 2.3. However, it is sometimes more problematic to establish a trusted authority in some realistic situation. In case of pursuiting deniability, we can incorporate the technique of a receipt-free

An unclonable group identification "without the group manager" - Recently, Damgård, Dupont, and Pedersen proposed the notion of the unclonable group identification [12]. The traceable ring signature can be applied to this application. The original unclonable group identification requires a group manager, but the traceable ring signature does not.

A traceable ring signature scheme is "functionally" related to a restricted blind signature. Hence, it can be applied to a very primitive "off-line" anonymous e-cash system.

Another possible application is, for instance, k-times anonymous authentication [25]. Any traceable ring signature scheme can be efficiently transformed into a traceable ring signature scheme with k-times anonymity defined as in [25], but see also Sec. 6.2.

1.3 Related Works

Linkable ring signatures [17, 27, 18, 26, 2] are closely related to the traceable ring signature. A linkable ring signature scheme is a ring signature scheme with the property that two signatures generated by the same signer with respect to the same ring can be linked, although it doesn't need satisfy the anonymity revocation property. The earlier papers about linkable ring signatures [17, 18] didn't consider a realistic threat that a dishonest signer makes a honest signer accused of "double-spending" (The schemes in [17, 18] are vulnerable to the attack. See Sec. 3, where our first-step protocol is substantially the same as the schemes in [17, 18]). The recent papers [27, 2] take care of this problem, which makes the security conditions more complicated. Our security definitions of the traceable ring signature works also on the linkable ring signature, if the tracing algorithm is appropriately modified, which implies that the unforgeability requirement is unnecessary also for a linkable ring signature scheme [2]. Recently, Tsang and Wei proposed a short linkable ring signature [26], based on a short group identification from [13], which allows for a shorter length of communication than our proposed scheme as the number of the ring members grows huge. Their scheme is, however, not a ring signature in our sense, because a trusted party must set up the parameter of an accumulator and the scheme is vulnerable to a dishonest trusted party [3]. In addition, it doesn't seem to provide public traceability. To our knowledge, only the proposal in [27] seems to be able to incorporate into itself the anonymity revocation property, but our scheme is simpler and more efficient than that scheme.

The restricted blind signature [9, 22, 5, 20], including its variant [25], is functionally related to the traceable ring signature. In the restricted blind signature,

---

voting scheme [21] into a traceable ring signature scheme. In that case, a trusted party is necessary but only for the receipt-freeness. The other security properties of the traceable ring signature mentioned above hold true even against a dishonest trusted party.

[2] In [2], this implication has been suggested.

[3] The accumulater used in [26] is based on factoring where an RSA modulus n is a system parameter, while the factoring should be kept secret.

however, the user must interact with the signer (corresponding to the group manager) to obtain a blind signature, which corresponds to a special setup with the group manager. This setup may seem somehow similar to the registration to PKI. In particular, the k-times anonymous authentication [25] is closer, because it allows a user to use the "blind signature" permanently (similar to a public-key), once he obtained it from the signer. However, the (restricted) blind signature, including the k-times anonymous authentication, cannot allow ad-hoc group formation. After the signer issues the blind signatures to the user, an arbitrary subgroup including the user cannot be selected as a ring and the services cannot be exclusively restricted to the subgroup.

Recently, Damgård, Dupont, and Pedersen proposed unclonable group identification [12]. It is functionally very close to the k-times anonymous authentication in the sense that after a user obtains a "coin" from the group manager, he can utilize it permanently. However, it does not allow for ad-hoc group formation, either.

A traceable signature scheme [15] is a group signature scheme with traceability (in particular, from a signature to a user), but it requires a group manager.

2 Traceable Ring Signature: Definitions

2.1 Notations and Syntax

For probabilistic algorithm A, we write $y \leftarrow A(x_1, \ldots, x_n)$ to denote the experiment of running A for given $(x_1, \ldots, x_n)$, selecting r uniformly from an appropriate domain, and assigning the result of this experiment to the variable y, i.e., $y := A(x_1, \ldots, x_n; r)$. For probability spaces, $X_1, \ldots, X_k$, and k-ary predicate ϕ, we write $\Pr[x_1 \leftarrow X_1; x_2 \leftarrow X_2; \cdots : \phi(x_1, \ldots, x_k)]$ to denote the probability that the predicate $\phi(x_1, \ldots, x_k)$ is true after the experiments, "$x_1 \leftarrow X_1; x_2 \leftarrow X_2; \cdots$", are executed in that order. Let $\epsilon, \tau : \mathbb{N} \rightarrow [0,1] (\subset \mathbb{R})$ be positive $[0,1]$-valued functions. We say that $\epsilon(k)$ is negligible in k if, for any constant $c > 0$, there exists a constant, $k_0 \in \mathbb{N}$, such that $\epsilon(k) < (1/k)^c$ for any $k > k_0$. We say that $\tau(k)$ is overwhelming in k if $\epsilon(k) \triangleq 1 - \tau(k)$ is negligible in k. For ordered finite set S, we denote by a_S vector $(a_i)_{i \in S}$. For $n \in \mathbb{N}$, we often write N to denote an ordered set $(1, \ldots, n)$.

We refer to an ordered public key set $pk_N = (pk_1, \ldots, pk_n)$ as a ring. We define a traceable ring signature scheme as indicated below.

Syntax. A traceable ring signature scheme is a tuple of algorithms, $\Sigma = (\mathbf{Gen}, \mathbf{Sig}, \mathbf{Ver}, \mathbf{Trace})$, such that, for $k \in \mathbb{N}$, the following is true.

- **Gen**: A probabilistic polynomial-time (in k) algorithm that takes security parameter $k \in \mathbb{N}$ and outputs a public/secret-key pair (pk, sk).
- **Sig**: A probabilistic polynomial-time (in k) algorithm that takes a secret key, sk_i, where $i \in N$, tag $L = (issue, pk_N)$, and message $m \in \{0,1\}^*$, and that outputs signature σ.

- **Ver**: A deterministic polynomial-time (in k) algorithm that takes tag $L = (issue, pk_N)$, message $m \in \{0,1\}^*$, and signature σ, and outputs a bit.
- **Trace**: A deterministic polynomial-time (in k) algorithm that takes tag $L = (issue, pk_N)$, and two message/signature pairs, $\{(m,\sigma),(m',\sigma')\}$, and outputs one of the following strings: "indep," "linked," or pk, where $pk \in pk_N$.

For simplicity, we often write $(pk_N, sk_N) \leftarrow \mathbf{Gen}(1^k)$ to denote the experiment of $(pk_i, sk_i) \leftarrow \mathbf{Gen}(1^k)$ for $i \in N$ and assigning $(pk_N, sk_N) := (pk_i, sk_i)_{i \in N}$.

As an ordinary signature scheme, a traceable ring signature scheme must satisfy the following correctness conditions: For every $k \in \mathbb{N}$, every $n \in \mathbb{N}$, every $i \in N := \{1,\ldots,n\}$, every $issue \in \{0,1\}^*$, and every $m \in \{0,1\}^*$, if $(pk_N, sk_N) \leftarrow \mathbf{Gen}(1^k)$, and $\sigma \leftarrow \mathbf{Sig}_{sk_i}(L,m)$, where $L = (issue, pk_N)$, it holds with an overwhelming probability (in k) that $\mathbf{Ver}(L,m,\sigma) = 1$.

Public Traceability - A traceable ring signature scheme requires that the following condition must hold: For every $k \in \mathbb{N}$, every $n \in \mathbb{N}$, every $i, i' \in N := \{1,\ldots,n\}$, every $issue \in \{0,1\}^*$, and every $m, m' \in \{0,1\}^*$, if $(pk_N, sk_N) \leftarrow \mathbf{Gen}(1^k)$, $\sigma \leftarrow \mathbf{Sig}_{sk_i}(L,m)$, where $L = (issue, pk_N)$, and $\sigma' \leftarrow \mathbf{Sig}_{sk_{i'}}(L,m')$, it holds with an overwhelming probability (in k) that

$$\mathbf{Trace}(L,m,\sigma,m',\sigma') = \begin{cases} \text{"indep"} & \text{if } i \neq i', \\ \text{"linked"} & \text{else if } m = m', \\ pk_i & \text{otherwise .} \end{cases}$$

In addition, if $m \neq m'$, **Trace** never output "linked." Public traceability is a correctness condition, that is, it does not assure that the opposite holds. However, if a traceable signature scheme has tag-linkability (along with public traceability), $\mathbf{Trace}(L,m,\sigma,m',\sigma') =$ "indep" implies that these two signatures are generated by different signers. If it has exculpability, $\mathbf{Trace}(L,m,\sigma,m',\sigma') = pk_i$ implies that they are signed by the same signer i. Note that $\mathbf{Trace}(L,m,\sigma,m,\sigma') =$ "linked" doesn't mean that they are always generated by the same signer (because anyone can make a "dead" copy of any signature).

2.2 Security Definitions

In this section, we describe the formal security definitions for the traceable ring signature. We give three requirements: *tag-linkability*, *anonymity*, and *exculpability*. As mentioned earlier, the "standard unforgeability" requirement is unnecessary for the traceable ring signature. We discuss this issue later in Sec. 2.3.

The tag-linkability is significantly different from the other two requirements in the sense that it is to defend the system, not the users. Hence, we assume all users (signers) are potential cheaters, which leads to the model that a central adversary generates all the public/secret keys for the users. On the other hand, anonymity and exculpability are to protect user(s) from the rest of players, including the system provider and the adversarial users. In these settings, an adversary is given the target public key(s) and allowed to append a polynomial

number (in total) of new public keys to the global public-key list in any timing. Possibly, these public-keys can be related to the given target key(s). We assume that the global public-key list is maintained properly: A public-key should be referred to only one user and vice versa. The adversary is basically allowed to choose an arbitrary subring in the global public-key list, when it accesses the signing oracle(s) with respect to the target user(s). We call such an attack *the adversarially-chosen-key-and-sub-ring attack*, which Bender, Katz, and Morselli have formally addressed in [4]. In our security model, as in [4], the role of PKI is minimal, namely it only maintains the global public-key list properly, which implies that security requirements hold true against malicious PKI.

We give the formal definitions of the security requirements as follows.

Tag-Linkability - Let F be an adversary modeled as a probabilistic algorithm. It takes security parameter $k \in \mathbb{N}$ and outputs $L = (issue, pk_N)$ and $(n+1)$ message/signature pairs, $\{(m^{(1)},\sigma^{(1)}), \ldots, (m^{(n+1)},\sigma^{(n+1)})\}$, where $pk_N = (pk_1, \ldots, pk_n)$. We define the advantage of F against Σ to be

$$\mathbf{Adv}^{\text{forge}}_{\Sigma}(F)(k) \triangleq \Pr[\mathbf{Expt}^{F}(k) = 1]$$

where $\mathbf{Expt}^{F}(k)$ are:

1. $\Big(L, \{(m^{(1)},\sigma^{(1)}), \ldots, (m^{(n+1)},\sigma^{(n+1)})\}\Big) \leftarrow F(1^k)$;
2. Return 1 iff
   - $\mathbf{Ver}(L, m^{(i)}, \sigma^{(i)}) = 1$ for all $i \in \{1, \ldots, n+1\}$, and
   - $\mathbf{Trace}(L, m^{(i)}, \sigma^{(i)}, m^{(j)}, \sigma^{(j)}) =$ "indep" for all $i, j \in \{1, \ldots, n+1\}$, where $i \neq j$.

Definition 1. *We say that Σ is tag-linkable if for any probabilistic polynomial-time (in k) algorithm F, $\mathbf{Adv}^{\text{forge}}_{\Sigma}(F)(k)$ is negligible in k.*

Anonymity - Let D be an adversary modeled as a probabilistic algorithm. Let (pk_0, pk_1) be the two target public keys, where (pk_0, sk_0) and (pk_1, sk_1) are generated by $\mathbf{Gen}(1^k)$. Let $b \in \{0,1\}$ be a random hidden bit. D starts the game with target (pk_0, pk_1). D may do the following things polynomial number of times in an arbitrary order: D may append new public keys to the global public-key list and may access three signing oracles, $\mathbf{Sig}_{sk_b}$, $\mathbf{Sig}_{sk_0}$, and $\mathbf{Sig}_{sk_1}$, where

- $\mathbf{Sig}_{sk_b}$ is the challenge signing oracle with respect to sk_b for signing (L, m), and
- $\mathbf{Sig}_{sk_0}$ (resp. $\mathbf{Sig}_{sk_1}$) is the signing oracle with respect to sk_0 (resp. sk_1) for signing (L, m).

Here we assume that L should include both pk_0, pk_1; that is, pk_0, $pk_1 \in pk_N$ for $L = (issue, pk_N)$. In addition, the following condition must hold:

- If (L, m) and (L, m') are two queries of D to the challenge signing oracle $\mathbf{Sig}_{sk_b}$, then $m = m'$.

– If (L, m) is a query of D to $\mathbf{Sig}_{sk_b}$ and $(\tilde{L}, \tilde{m})$ is a query of D to $\mathbf{Sig}_{sk_0}$ or $\mathbf{Sig}_{sk_1}$, then $L \neq \tilde{L}$.

Finally, D outputs a bit b'. We define the advantage of D against Σ as

$$\mathbf{Adv}^{\mathrm{anon}}_{\Sigma}(D)(k) \triangleq \Pr\left[\begin{array}{l}(pk_0, sk_0),\ (pk_1, sk_1) \leftarrow \mathbf{Gen}(1^k);\\ b \leftarrow \{0,1\};\\ b' \leftarrow D^{\mathbf{Sig}_{sk_b}, \mathbf{Sig}_{sk_0}, \mathbf{Sig}_{sk_1}}(pk_0, pk_1)\end{array} : \quad b = b'\right] - \frac{1}{2}.$$

Definition 2. *We say that Σ is anonymous if, for every probabilistic polynomial-time (in k) adversary D, the advantage $\mathbf{Adv}^{\mathrm{anon}}_{\Sigma}(D)(k)$ is negligible in k.*

Remark 1. Our anonymity definition corresponds to Definition 3 in [4], which is not the strongest among their three definitions. It is, however, impossible for a traceable ring signature scheme to satisfy the strongest definition in [4], because the strongest definition requires that an adversary cannot distinguish which target generated the signature even when the adversary is given one of the target secrets; namely, all but one secret key in the ring is exposed. This condition and the public traceability cannot hold simultaneously.

Exculpability - Let A be a probabilistic algorithm as an adversary. Let pk be the target public key where (pk, sk) is generated by $\mathbf{Gen}(1^k)$. A starts the game with the target pk. A may do the following things a polynomial number of times in an arbitrary order. A may append new public keys to the global public-key list and may ask the signing oracle with respect to sk, $\mathbf{Sig}_{sk}$, to sign any $(\tilde{L}, \tilde{m})$, where $\tilde{L} = (\tilde{issue}, pk_{\tilde{N}})$, only with the restriction that $pk \in pk_{\tilde{N}}$. Finally, A outputs two pairs, (L, m, σ) and (L, m', σ'), where $L = (issue, pk_N)$. Here they should satisfy $pk \in pk_N$, $\mathbf{Ver}(L, m, \sigma) = 1$, and $\mathbf{Ver}(L, m', \sigma') = 1$. In addition, it must hold that at least one of (L, m, σ) or (L, m', σ') is not linked to any $(L, \hat{m}, \hat{\sigma})$ in the query/answer list between A and $\mathbf{Sig}_{sk}$ [4]. It is, however, allowed that one of them is linked to one in the query/answer list.

We say that A entraps a player with respect to pk if $\mathbf{Trace}(L, m, \sigma, m', \sigma') = pk$. We define the advantage of A against Σ, to be $\mathbf{Adv}^{\mathrm{entrap}}_{\Sigma}(A)(k) \triangleq$

$$\Pr\left[\begin{array}{l}(pk, sk) \leftarrow \mathbf{Gen}(1^k);\\ (L, m, \sigma), (L, m', \sigma') \leftarrow A^{\mathbf{Sig}_{sk}}(pk)\end{array} : \mathbf{Trace}(L, m, \sigma, m', \sigma') = pk\right].$$

Definition 3. *We say that Σ is exculpable if, for any probabilistic polynomial-time adversary A, $\mathbf{Adv}^{\mathrm{entrap}}_{\Sigma}(A)(k)$ is negligible in k.*

Remark 2. **In relation to the adaptively-chosen insider corruption attack:** One might think that the exculpability definition could be stronger when there are not only one but polynomially-many targets and the adversary can

[4] It implies two-fold. Our definition doesn't care for strong unforgeability. In addition, A is allowed to output a signature originally forged by himself with a copy (or linked one) from the query/answer list.

adaptively request the corruption of the target signers and finally attack one of the remaining uncorrupted targets. However, it is obvious that if an traceable ring signature satisfies this version of exculpability, then it also satisfies the improved definition, because the number of the ring members are at most polynomial (in security parameter k).

2.3 Discussion

As mentioned earlier, the standard unforgeability requirement (as defined in an ordinary ring signature) is inessential for a traceable ring signature scheme. We define unforgeability as the inability of an adversary that takes all public-key $pk_{\overline{N}}$ and, after having access to the signing oracle with (L, m, i), outputs (L', m', σ'), $L' = (issue', pk_{N'})$ and $N' \subset \overline{N}$, such that (L', m') never asked to the signing oracle. Here, for query (L, m, i), where $L = (issue, pk_N)$ and $i \in N \subset \overline{N}$, the signing oracle returns $\mathbf{Sig}_{sk_i}(L, m)$. We then have the following result.

Theorem 1. *If a traceable ring signature scheme is tag-linkable and exculpable, then it is unforgeable.*

Proof. Suppose for contradiction that there is an adversary A' against unforgeability. Let (L, m, σ) be the output of A', where $L = (issue, pk_N)$. Then, consider n independent pairs $\{(L, m^{(1)}, \sigma^{(1)}), \ldots, (L, m^{(n)}, \sigma^{(n)})\}$, such that $m^{(i)} \neq m$ and $\mathbf{Ver}(L, m^{(i)}, \sigma^{(i)}) = 1$ for all $i \in \{1, \ldots, n\}$. If every $n+1$ pairs are independent, then it contradicts tag-linkability. Therefore, there is an $i \in \{1, \ldots, n\}$ such that $\mathbf{Trace}(L, m, \sigma, m^{(i)}, \sigma^{(i)}) = pk \in pk_N$, because $m^{(i)} \neq m$ (Remember that **Trace** never outputs "linked" if $m^{(i)} \neq m$). This case, however, contradicts the exculpability requirement, because we can construct adversary A against exculpability, by using A' as a black box oracle as follows. For simplicity, we assume, without loss of generality, that A takes all public-keys as the targets, as discussed in Remark 2. A feeds all public-keys to A'. For any query of A', A asks the signing oracle the answer and returns it to A'. A' finally outputs (L, m, σ), where $L = (issue, pk_N)$. Then, A asks for n queries and obtains $(L, m^{(1)}, \sigma^{(1)})$, ..., $(L, m^{(n)}, \sigma^{(n)})$, where $m^{(i)} \neq m$ for all i. Since there is an i such that $\mathbf{Trace}(L, m, \sigma, m^{(i)}, \sigma^{(i)}) = pk \in pk_N$, A outputs (L, m, σ) and $(L, m^{(i)}, \sigma^{(i)})$, which contradicts exculpability. □

We note that a traceable ring signature always provides efficient confirmation and disavowal protocols (where we don't assume that these protocol are zero-knowledge). If a member of the ring wants to prove that a signature has been generated by himself, he can make another signature for a different message with the same tag, which would reveal his identity. Similarly, if a member of the ring wants to prove that a signature has not been generated by himself, he can submit another signature for an arbitrary message with the same tag, which shows that the second one is independent of the previous one. In some application it is undesirable, but *any* anonymous authentication primitive with public traceability (or linkability) cannot avoid this property.

3 Towards Our Scheme

Although our proposal is not very complicated, we construct our scheme step by step to understand more easily the concept behind our design.

Let us keep in mind the undeniable signature scheme proposed by Chaum [8]: Letting $y_i = g^{x_i} \in G$ be a public key of player i, the Chaum's undeniable signature on message M is $\sigma_i = H(M)^{x_i} \in G$, where H denotes a hash function. Now let $M = issue||pk_N$ where $pk_N = (pk_1, \ldots, pk_n)$ are a vector of n public-keys. Pick up at random $(n-1)$ elements, σ_j's, from G, where $j \neq i$. Then, set a NP-language

$$\mathcal{L} \triangleq \{(y_N, h, \sigma_N)) \mid \quad \exists\, i \in N \text{ such that } \log_g(y_i) = \log_h(\sigma_i).\},$$

where $h = H(issue||y_N)$ and $\sigma_N = (\sigma_1, \ldots, \sigma_n)$.

Then, consider a zero-knowledge based signature (using secret x_i) on this language. It is well-known that such a signature can be constructed by applying the technique of Cramer et al. [11] (one-out-of n honest-verifier zero-knowledge) to the Fiat-Shamir technique. The signature on m is then (σ_N, p), where $p = (c, z)$ is a (non-interactive) proof on $\mathcal{L}$ and $c = H(\sigma_N, a, m)$, where a is computed by p. We call this our first-step construction.

Suppose now that this scheme is applied to anonymous voting on BBS, where each user can write on BBS anonymously. Let $L = (issue, pk_N)$, where *issue* denotes the vote id number and pk_N corresponds to the authorized voters. Each voter simply sends message "yes" or "no" along with signature (σ_N, p) to a bulletin board via a sender-anonymous channel (such as the Internet in practice). If proof p is sound, a cheating player, say i, could not vote twice because it turns out $\sigma_i = \sigma_i' = h^{x_i}$, which takes the risk of revealing his identity.

However, this construction does not work well when an adversary is one of the voters. The problem is that an adversarial player, say j, can entrap an innocent player, say i, or at least void the first vote, with a significant probability. Player j waits for someone to send the first vote, say ("*yes*," (σ_N, p)), to the bulletin board. After seeing this signature, he generate a valid signature (σ_N', p') on message "*no*," using secret key x_j, following a valid signing procedure, except that he sets $\sigma_i' = \sigma_i$ and $\sigma_k' \neq \sigma_k$ for all $k \neq i$. He then sends ("*no*," σ_N', p') to the board. If the first vote is really generated by player i, player i cannot deny the second vote, because the second vote is a valid signature potentially generated by player i. At least, player i would lose his first vote, because he cannot prove which of two votes are valid.

Our solution is to make signer i fix every σ_j, $j \neq i$, depending on (L, m) and σ_i. More precisely, each point $(j, \log_h(\sigma_j))$ is forced to be on the line defined by $(i, \log_h(\sigma_i))$ and $(0, \log_h(H(L, m)))$. Intuitively, to generate a signature that will pass verification, player i must set $\sigma_i = h^{x_i}$, while to entrap player j, he must set at the same time that $(j, \log_h(\sigma_j))$ lies on the line defined by $(i, \log_h(\sigma_i))$ and $(0, \log_h(H(L, m)))$, which seems intractable. On the other hand, suppose that signer i generates two signatures, σ_N and σ_N', on m and m', $m \neq m'$, with respect to the same tag L. Every $(j, \log_h(\sigma_j))$ derived from the first σ_N

lies on the line defined by $(i, \log_h(\sigma_i))$ and $(0, \log_h(H(L,m)))$, whereas every $(j, \log_h(\sigma'_j))$ derived from the second σ'_N does on the line defined by $(i, \log_h(\sigma_i))$ and $(0, \log_h(H(L,m')))$. Since the first line intersects with the second line at $(i, \log_h(\sigma_i))$ and these are not the same line (because $H(L,m) \neq H(L,m')$), it holds that $\sigma_i = \sigma'_i$ and $\sigma_j \neq \sigma'_j$ for all $j \neq i$, which implies that the identity of the cheating player is traced. We formally prove in Sec. 5 that this approach successfully works. Interestingly, this scheme is more efficient than the first-step construction described above in terms of communication traffic.

4 An Efficient Traceable Ring Signature Scheme

In this section, we describe our proposal.

Let G be a multiplicative group of prime order q and let g be a generator of G. Let $H : \{0,1\}^* \to G$, $H' : \{0,1\}^* \to G$, and $H'' : \{0,1\}^* \to \mathbb{Z}_q$ be distinct hash functions (modeled as random oracles in the security statements below). These above are public parameters. The key generation for player i is as follows: Player i picks up random element x_i in $\mathbb{Z}_q$ and computes $y_i = g^{x_i}$. The public key of i is $pk_i = \{g, y_i, G\}$ and the corresponding secret key is $sk_i = \{pk_i, x_i\}$. The player i registers his public-key to PKI. We denote by $N = \{1, \ldots, n\}$ an ordered list of n players. We let $pk_N = (pk_1, \ldots, pk_n)$ be an ordered public-key list for set N. Let *issue* be an arbitrary string in $\{0,1\}^*$.

Signing protocol : To sign message $m \in \{0,1\}^*$ with respect to tag $L = (issue, pk_N)$, using the secret-key sk_i, proceed as follows:

1. Compute $h = H(L)$ and $\sigma_i = h^{x_i}$, using $x_i \in \mathbb{Z}_q$.
2. Set $A_0 = H'(L,m)$ and $A_1 = \left(\frac{\sigma_i}{A_0}\right)^{1/i}$.
3. For all $j \neq i$, compute $\sigma_j = A_0 A_1^j \in G$. Notice that every $(j, \log_h(\sigma_j))$ is on the line defined by $(0, \log_h(A_0))$ and (i, x_i), where $x_i = \log_h(\sigma_i)$.
4. Generate signature (c_N, z_N) on (L,m), based on a (non-interactive) zero-knowledge proof of knowledge for the relation derived from language

$$\mathcal{L} \triangleq \{(L, h, \sigma_N)) \mid \quad \exists i' \in N \text{ such that } \log_g(y_{i'}) = \log_h(\sigma_{i'}).\},$$

   where $\sigma_N = (\sigma_1, \ldots, \sigma_n)$, as follows:
   (a) Pick up random $w_i \leftarrow \mathbb{Z}_q$ and set $a_i = g^{w_i}, b_i = h^{w_i} \in G$.
   (b) Pick up at random $z_j, c_j \leftarrow \mathbb{Z}_q$, and set $a_j = g^{z_j} y_i^{c_j}, b_j = h^{z_j} \sigma_j^{c_j} \in G$ for every $j \neq i$.
   (c) Set $c = H''(L, A_0, A_1, a_N, b_N)$ where $a_N = (a_1, \ldots, a_n)$ and $b_N = (b_1, \ldots, b_n)$.
   (d) Set $c_i = c - \sum_{j \neq i} c_j \pmod q$ and $z_i = w_i - c_i x_i \pmod q$. Return (c_N, z_N), where $c_N = (c_1, \ldots, c_n)$ and $z_N = (z_1, \ldots, z_n)$, as a proof of $\mathcal{L}$.
5. Output $\sigma = (A_1, c_N, z_N)$ as the signature on (L,m).

Verification protocol: To verify signature $\sigma = (A_1, c_N, z_N)$ on message m with respect to tag L, check the following:

1. Parse L as $(issue, pk_N)$. Check $g, A_1 \in G$, $c_i, z_i \in \mathbb{Z}_q$ and $y_i \in G$ for all $i \in N$. Set $h = H(L)$ and $A_0 = H'(L, m)$, and compute $\sigma_i = A_0 A_1^i \in G$ for all $i \in N$.
2. Compute $a_i = g^{z_i} y_i^{c_i}$ and $b_i = h^{z_i} \sigma_i^{c_i}$ for all $i \in N$.
3. Check that $H''(L, m, A_0, A_1, a_N, b_N) \equiv \sum_{i \in N} c_i \pmod q$, where $a_N = (a_1, \ldots, a_n)$ and $b_N = (b_1, \ldots, b_n)$.
4. If all the above checks are successfully completed, accept, otherwise reject.

Tracing protocol: To check the relation between (m, σ) and (m', σ'), with respect to the same tag L where $\sigma = (A_1, c_N, z_N)$ and $\sigma' = (A'_1, c'_N, z'_N)$, check the following:

1. Parse L as $(issue, pk_N)$. Set $h = H(L)$ and $A_0 = H'(L, m)$, and compute $\sigma_i = A_0 A_1^i \in G$ for all $i \in N$. Do the same thing for σ' and retrieve σ'_i, for all $i \in N$.
2. For all $i \in N$, if $\sigma_i = \sigma'_i$, store pk_i in **TList**, where **TList** is initially an empty list.
3. Output pk if pk is the only entry in **TList**; "linked" else if **TList** $= pk_N$; "indep" otherwise (i.e., **TList** $= \emptyset$ or $1 < \#$**TList** $< n$).

5 Security

In this section, we give security proofs for our traceable ring signature scheme.

Before proving tag-linkability for our scheme, we prove the following useful lemmas. We consider adversary A against our signature scheme above. A is given 1^k and allowed to access the random oracles, H' and H'', at most $q_{H'}$ and $q_{H''}$ times, respectively. Here it is not necessary that A is polynomial-time bounded. Then, we have the following lemmas.

Lemma 1. *Suppose that A outputs valid pair (L, m, σ).*

1. *The probability that $\#\{i \in N | \log_h(\sigma_i) = \log_g(y_i)\} < 1$ is at most $\frac{q_{H''}}{q}$, whereas*
2. *The probability that $\#\{i \in N | \log_h(\sigma_i) = \log_g(y_i)\} > 1$ is at most $\frac{q_{H'}}{q}$,*

where the probability is taken over the choices of H', H'' and the inner coin tosses of A.

Proof. Case 1 ($\#\{i \in N | \log_h(\sigma_i) = \log_g(y_i)\} < 1$): $\mathbf{Ver}(L, m, \sigma) = 1$ implies that $a_i = g^{z_i} y_i^{c_i} \in G$ and $b_i = h^{z_i} \sigma_i^{c_i} \in G$ for $i \in N$, which means that $\log_g(a_i) = z_i + c_i \cdot \log_g(y_i)$ and $\log_h(b_i) = z_i + c_i \cdot \log_h(\sigma_i)$ for $i \in N$. Note that if $\log_g(y_i) \neq \log_h(\sigma_i)$, c_i is determined. Hence, Case 1 implies that all c_i's, where $i \in N$, are uniquely determined. Since H'' is a random oracle, for any given $(L, m, A_0, A_1, a_N, b_N)$, the probability that $H''(L, m, A_0, A_1, a_N, b_N) = \sum_{i \in N} c_i$

(mod q), is at most q^{-1}. Therefore, for any A with at most $q_{H''}$ queries to random oracle H'', the probability of Case 1 is at most $\frac{q_{H''}}{q}$.

Case 2 ($\#\{i \in N | \log_h(\sigma_i) = \log_g(y_i)\} > 1$): Since $\sigma_i = A_0 A_1^i \in G$ for $i \in N$, every point $(i, \log_h(\sigma_i))$, $i \in N$, is on line $y = \log_h(A_1)x + \log_h(A_0)$. Case 2 implies that at least two points, $(i, \log_g(y_i))$'s, are on the line, which means, when pk_N are fixed, the line is determined, so $\log_h(A_0)$ and $\log_h(A_1)$ are determined. However, we also need $\log_h(A_0) = \log_h(H'(L(issue, pk_N), m))$, where $H'(L, m)$ is determined independently of the above line, because H' is a random oracle. Actually, the probability that $\log_h(H'(L, m)) = \log_h(A_0)$ is at most q^{-1} for given (L, m). Hence, for any adversary A with at most $q_{H'}$ number of queries to random oracle H', the probability of Case 2 is at most $\frac{q_{H'}}{q}$. □

Lemma 2. *Suppose A is defined above and it outputs $(L, m^{(1)}, \sigma^{(1)})$ and $(L, m^{(2)}, \sigma^{(2)})$, such that* **Trace**$(L, m^{(1)}, \sigma^{(1)}, m^{(2)}, \sigma^{(2)}) =$ *"indep". Let* **TList** *be the list defined above in our tracing protocol. Then, the probability that* $1 <$ #**TList** *is* $\frac{q_{H'}^2}{2q}$*, where the probability is taken over the choices of H' and the inner coin tosses of A.*

Proof. By $1 <$ #**TList**, the line defined by $\sigma^{(1)}$ intersects with the line defined by $\sigma^{(2)}$ at least at two points, which means that the two lines coincide. Hence, $A_0^{(1)} = H'(L, m^{(1)})$ and $A_0^{(2)} = H'(L, m^{(2)})$, because $\log_h A_0^{(1)} = \log_h A_0^{(2)}$ where $h = H(L)$. Therefore, the advantage of A is bounded by the probability that A can find a collision of outputs of H', which is $\frac{q_{H'}^2}{2q}$. □

Theorem 2 (Tag-Linkability). *Our proposed scheme is tag-linkable in the random oracle model.*

Proof. Suppose for contradiction that there is adversary F that takes 1^k and successfully outputs tag $L = (issue, pk_N)$ and $\{(m^{(1)}, \sigma^{(1)}), \ldots, (m^{(n+1)}, \sigma^{(n+1)})\}$.

Based on lemma 2, **Trace**$(L, m^{(i)}, \sigma^{(i)}, m^{(j)}, \sigma^{(j)})$ = "indep," for all i, j, means that, (with an overwhelming (i.e., $1 - \frac{q_{H'}^2}{2q}$) probability), $\sigma_k^{(i)} \neq \sigma_k^{(j)}$ holds, for all i, j, k, where $1 \leq i, j \leq n+1$, $i \neq j$, and $1 \leq k \leq n$. On the contrary, by Case 1 of Lemma 1, for every i, where $1 \leq i \leq n+1$, there exist $k \in N$ such that $\log_g(y_k) = \log_h(\sigma_k^{(i)})$ (with at least $(1 - \frac{(n+1)q_{H''}}{q})$ probability). Since $1 \leq k \leq n$, there exist i, j, k such that $\sigma_k^{(i)} = \sigma_k^{(j)}$, which contradicts the assumption (if the advantage of F exceeds $\max(\frac{q_{H'}^2}{2q}, \frac{(n+1)q_{H''}}{q})$).

Therefore, the probability that F can forge the proposed scheme above is at most $\max(\frac{q_{H'}^2}{2q}, \frac{(n+1)q_{H''}}{q})$, where $q_{H'}$ and $q_{H''}$ denotes the number of queries of F to random oracles, H' and H'', respectively. □

Before proceeding other theorems, we define a protocol, commonly used in some of the following proofs.

Procedure of SimNIZK.
On input: (L, m, h, A_0, A_1).
Output: (c_N, z_N).

1. For all $i \in N$, pick up at random $z_i, c_i \leftarrow_U \mathbb{Z}_q$, and set $a_i = g^{z_i} y_i^{c_i}, b_i = h^{z_i} \sigma_i^{c_i} \in G$, where $\sigma_i = A_0 A_1^i$.
2. Set $H''(L, m, A_0, A_1, a_N, b_N)$ as $c := \sum_{i \in N} c_i$, where $a_N = (a_1, \ldots, a_n)$ and $b_N = (b_1, \ldots, b_n)$. If $H''(L, m, A_0, A_1, a_N, b_N)$ has been already booked as a different value in query/answer list $Q_{H''}$, then output "failure," otherwise
3. Output (c_N, z_N), where $c_N = (c_1, \ldots, c_n)$ and $z_N = (z_1, \ldots, z_n)$.

We now show the following theorem.

Theorem 3 (Anonymity). *Our proposed scheme is anonymous under the decisional Diffie-Hellman assumption in the random oracle model.*

Proof. Suppose that there is an adversary D with advantage ϵ, which means that, by definition, D can correctly guess b with probability $\epsilon + \frac{1}{2}$. We now construct an algorithm A to solve the decisional Diffie-Hellman problem. Let (g_1, g_2, u, v) be a given instance, where $g_1, g_2, u, v \in G$. When (g_1, g_2, u, v) is a DDH tuple, $\log_{g_1}(u) = \log_{g_2}(v)$ holds. We construct A as follows:

1. A is given instance (g_1, g_2, u, v).
2. A picks up at random $b \leftarrow \{0, 1\}$.
3. A sets $g := g_1$, $y_b := u$ and, picking up at random $t \in \mathbb{Z}_q$, $y_{1-b} := y_b g^t$.
4. A feeds y_0, y_1 to D.
5. In case D submits a fresh query to random oracles, H' and H'', A picks up random elements in G and $\mathbb{Z}_q$ respectively, to reply with. Then, A stores the query/answer pairs in the lists, $Q_{H'}$ and $Q_{H''}$, respectively.
6. In case D submits a fresh query to random oracle H, A picks up at random $r_1, r_2 \leftarrow \mathbb{Z}_q$ and returns ${g_1}^{r_1} {g_2}^{r_2}$. Then, A stores the value as well as (r_1, r_2) in query/answer list Q_H.
   In this simulation, if A picks up the same $g_1^{r_1} g_2^{r_2}$ again, namely, $H(L) = H(L')$ happens for $L \neq L'$, A aborts. However, such an event happens at most $\frac{q_H}{q}$, which is negligible in k, where q_H denotes the total number of queries of D to H.
7. In case D submits a query (L, m) to $\mathbf{Sig}_{sk_b}$, A sets $g_1^{r_1} {g_2}^{r_2}$ as $h := H(L)$ and $\sigma_b := u^{r_1} v^{r_2}$, picking up at random $r_1, r_2 \in \mathbb{Z}_q$. Then, A picks up a random element A_0 as $H'(L, m)$. If $H(L)$ and $H'(L, m)$ have been already stored in Q_H and $Q_{H'}$, respectively, A uses these stored values. A sets A_1 and σ_N, by using A_0 and σ_b. Then, A simulates a NIZK proof on language

$$\mathcal{L} \triangleq \{(L, h, \sigma_N)) \mid \quad \exists i' \in N \text{ such that } \log_g(y_{i'}) = \log_h(\sigma_{i'}).\},$$

   following procedure **SimNIZK** described above to get (c_N, z_N), where $c_N = (c_1, \ldots, c_n)$ and $z_N = (z_1, \ldots, z_n)$. If **SimNIZK** succeeds, A returns $\sigma = (A_1, c_N, z_N)$ to D, otherwise A halts.
8. In case D submits a query (L, m) to $\mathbf{Sig}_{sk_0}$, if $b = 0$ do the same thing as in Step 7. Otherwise, A sets $g_1^{r_1} {g_2}^{r_2}$ as $h := H(L)$ and $\sigma_0 := u^{r_1} v^{r_2} (g_1^{r_1} g_2^{r_2})^t$, picking up at random $r_1, r_2 \in \mathbb{Z}_q$. Then, A picks up a random element A_0 as $H'(L, m)$. If $H(L)$ and $H'(L, m)$ have been already stored in Q_H and $Q_{H'}$,

respectively, A uses these stored values. A sets A_1 and σ_N, by using A_0 and σ_0. Then, A simulates a NIZK proof on language

$$\mathcal{L} \triangleq \{(L, h, \sigma_N)) \mid \quad \exists i' \in N \text{ such that } \log_g(y_{i'}) = \log_h(\sigma_{i'}).\},$$

following procedure **SimNIZK** described below to get (c_N, z_N), where $c_N = (c_1, \ldots, c_n)$ and $z_N = (z_1, \ldots, z_n)$. If **SimNIZK** succeeds, A returns $\sigma = (A_1, c_N, z_N)$ to D, otherwise A halts.

9. In case D submits a query (L, m) to $\mathbf{Sig}_{sk_1}$, do the same thing as in Step 8.
10. Finally, D outputs b'. If $b = b'$, A output 1, otherwise A flips a coin $b'' \in \{0, 1\}$ to output.

The advantage of A against the DDH problem is defined as

$$\Pr[A(g_1, g_2, u, v) = 1 \mid (g_1, g_2, u, v) \in \text{DDH}] - \Pr[A(g_1, g_2, u, v) = 1 \mid (g_1, g_2, u, v) \notin \text{DDH}].$$

We say that A succeeds in simulation if no collision happens in simulating random oracle H and **SimNIZK** succeeds in simulating proofs for all queries of D to the signing oracles. **SimNIZK** fails in generating a proof with at most probability $\frac{q_{H''}}{q}$, where $q_{H''}$ denotes the total number of queries of D to H''. Hence, the probability that **SimNIZK** fails at least once in this game is bounded by $\frac{q_{\mathbf{Sig}} \cdot q_{H''}}{q}$, where $q_{\mathbf{Sig}}$ denotes the total number of queries of D to the signing oracles.

We evaluate the following probabilities on the condition that A succeeds in simulation.

Notice that if (g_1, g_2, u, v) is a DDH tuple and a reply of the signing oracles, $\mathbf{Sig}_{sk_b}$, $\mathbf{Sig}_{sk_0}$, and $\mathbf{Sig}_{sk_1}$, is identical to the real signature using sk_b, sk_0, and sk_1, respectively (on the condition that **SimNIZK** succeeds in simulating a proof).

On the other hand, if it is a random tuple, hidden bit b is perfectly independent of the adversary's view.

Hence, we have $\Pr[b = b' | (g_1, g_2, u, v) \in \text{DDH}] = \epsilon + \frac{1}{2}$ by assumption and $\Pr[b = b' | (g_1, g_2, u, v) \notin \text{DDH}] = \frac{1}{2}$.

Therefore, $\Pr[A(g_1, g_2.u, v) = 1 | (g_1, g_2, u, v) \in \text{DDH}] = \Pr[b = b' | (g_1, g_2, u, v) \in \text{DDH}] + \Pr[b \neq b' | (g_1, g_2, u, v) \in \text{DDH}] \cdot \Pr[b'' = 1 | (g_1, g_2, u, v) \in \text{DDH} \wedge b \neq b']$ $= \left(\epsilon + \frac{1}{2}\right) + \left(1 - \left(\epsilon + \frac{1}{2}\right)\right) \cdot \frac{1}{2} = \frac{\epsilon}{2} + \frac{3}{4}$.

On the other hand, $\Pr[A(g_1, g_2, u, v) = 1 | (g_1, g_2, u, v) \notin \text{DDH}] = \Pr[b = b' | (g_1, g_2, u, v) \notin \text{DDH}] + \Pr[b \neq b' | (g_1, g_2, u, v) \notin \text{DDH}] \cdot \Pr[b'' = 1 | (g_1, g_2, u, v) \notin \text{DDH} \wedge b \neq b'] = \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} = \frac{3}{4}$.

Based on this estimation, the advantage of A is $\frac{1}{2} \cdot \epsilon$, if A succeeds in simulation. Therefore, the advantage of A is bounded by

$$\frac{1}{2} \cdot \epsilon - \frac{q_H}{q} - \frac{q_{\mathbf{Sig}} \cdot q_{H''}}{q}.$$

To suppress the advantage of A to be negligible in k, ϵ must be negligible in k. □

Before proceeding to the exculpability statement, we prove the following lemma. Let A be an adversary against exculpability for our scheme. Let $q_{H'}, q_{H''}$ denote the total number of queries to the random oracles, H', H'', respectively. Here it is not necessary that A is polynomial-time bounded. Then, we have the following.

Lemma 3. *When A entraps player i, the probability that $\log_h(\sigma_i) \neq \log_g(y_i)$ is at most $\frac{(n-1)(n-2)q_{H'}^2}{2q} + \frac{q_{H''}}{q}$. The probability is taken over the choices of H', H'' and the inner coin tosses of A.*

Proof. Assume that $\log_h(\sigma_i) \neq \log_g(y_i)$. Based on lemma 1, if $\mathbf{Ver}(L, m, \sigma) = 1$, the probability that $\#\{i \in N | \log_h(\sigma_i) = \log_g(y_i)\} < 1$ is at most $\frac{q_{H''}}{q}$. Hence, for σ and σ' that A outputs, there are $j, k \in N$, with an overwhelming probability, such that $\log_h(\sigma_j) = \log_g(y_j)$ and $\log_h(\sigma'_k) = \log_g(y_k)$, which implies that

$$\log_h(y_j) = \log_h(A_1) \cdot j + \log_h(A_0) \tag{1}$$
$$\log_h(y_k) = \log_h(A'_1) \cdot k + \log_h(A'_0). \tag{2}$$

Since $\log_h(\sigma_i) \neq \log_g(y_i)$, it holds that $j, k \neq i$.

By assumption, line $y = \log_h(A_1) \cdot x + \log_h(A_0)$ intersects with line $y = \log_h(A'_1) \cdot x + \log_h(A'_0)$ at $x = i$. Hence, we have

$$\log_h(A_1) \cdot i + \log_h(A_0) = \log_h(A'_1) \cdot i + \log_h(A'_0). \tag{3}$$

By (1), (2), and (3), we have

$$A \cdot \log_h(A_0) + B \cdot \log_h(A'_0) = C, \tag{4}$$

where A, B, C are fixed when i, j, k, $\log_g(y_j)$ and $\log_g(y_k)$ are fixed. Remember that $A_0 = H'(L, m)$ and $A'_0 = H'(L, m')$ must hold, where $L = (issue, pk_N)$. Note that $H'(L, m), H'(L, m')$ are fixed *after* i, j, k, $\log_g(y_j)$ and $\log_g(y_k)$ are fixed. Hence, the probability that A_0 and A'_0 satisfy (4) is at most $\frac{q_{H'}^2}{2q}$, because H' is a random oracle.

The probability that A_0, A'_0 satisfy (4) is the same in every $j, k \in N - \{i\}$, $j \neq k$; Hence, the probability that $\log_h(\sigma_i) \neq \log_g(y_i)$ is at most $\frac{(n-1)(n-2)q_{H'}^2}{2q} + \frac{q_{H''}}{q}$. □

When adversary A entraps player i, there are two possibilities: One is the case that A really forges the signature of player i (possibly, after seeing her/his real signature). Namely, it is the case that $\log_h(\sigma_i) = \log_h(\sigma'_i) = \log_g(y_i)$. The other case $\log_h(\sigma_i) = \log_h(\sigma'_i) \neq \log_g(y_i)$, means that A does not forge the signatures of player i but, letting σ, σ' be generated by A, the i-th entries of them, σ_i and σ'_i, are the same. This lemma implies that if A entraps player i, it is the case, with an overwhelming probability, that A has really forged a signature of player i.

Theorem 4 (Exculpability). *Our proposed scheme is exculpable under the discrete logarithm assumption in the random oracle model.*

A very rough strategy for proving the theorem is as follows: Based on lemma 3, we know that if an adversary A against exculpability for our scheme can entraps the target player i, then it is the case with an overwhelming probability that A has actually forged a signature of player i, i.e., $\log_h \sigma_i = \log_g y_i$. In addition, by lemma 1, we realize that that it is "never" a potential signature of any other player at the same time, i.e., $\log_h \sigma_j \neq \log_g y_j$, for $j \neq i$ (with an overwhelming probability). This implies that by the standard rewinding, we have $c_i \neq c_i''$ for the target i, which breaks the discrete log of the target y_i and leads to the contradiction.

Proof. Suppose that there is adversary A that takes pk and entraps the player with respect to pk. Then, we can construct algorithm A' that solves the discrete logarithm problem. Let $g, Y \in G$ be a given instance of discrete logarithm problem. The goal of A' is to output $\log_g Y$. We construct A' as follows.

Without loss of generality, we assume that the id number of the target player is i. Hence, A' sets $y_i := Y$ and feeds $pk_i = \{y_i, g\}$ to adversary A.

A may access the random oracles, H, H', H'', and the signing oracle, at most $q_H, q_{H'}, q_{H''}$ and $q_{\mathbf{Sig}}$ times, respectively. In case A submits a fresh query to random oracles, H' and H'', A' picks up random elements in G and $\mathbb{Z}_q$ respectively, to use as a reply, maintaining the query/answer lists, $Q_{H'}$ and $Q_{H''}$, respectively. In case A submits a fresh query to random oracle H, A' picks up random $v \in \mathbb{Z}_q$ and return g^v to A, maintaining query/answer list Q_H. In case A submits query $(\tilde{L}, \tilde{m})$, to the signing oracle, A' returns σ as follows.

1. Pick up random $v \in \mathbb{Z}_q$, to set value $\tilde{h} := H(\tilde{L})$ as g^v. Pick up random $\tilde{A}_0$ as $H'(\tilde{L}, \tilde{m})$. If $H(\tilde{L})$ and $H'(\tilde{L}, \tilde{m})$ have been already booked in Q_H and $Q_{H'}$, respectively, use these stored values. Set $\tilde{\sigma_i} := y_i^v$.
2. Compute $\tilde{A}_1$ and $\tilde{\sigma}_N$. Then use **SimNIZK** on input $(\tilde{L}, \tilde{m}, \tilde{h}, \tilde{A}_0, \tilde{A}_1)$. **SimNIZK** returns $(\tilde{c}_N, \tilde{z}_N)$ except for a negligible probability $\frac{q_{H''}}{q}$. If **SimNIZK** fails in simulating a proof, then A' aborts.
   The probability that **SimNIZK** fails at least once in this game is bounded by $\frac{q_{\mathbf{Sig}} \cdot q_{H''}}{q}$.
3. Return $\tilde{\sigma} = (\tilde{A}_1, \tilde{c}_N, \tilde{z}_N)$ and store the query/answer pair in the list $Q_{\mathbf{Sig}}$.

Finally, A outputs (L, m, σ) and (L, m', σ'). A entraps player i with probability ϵ, which is the advantage of A. Then, A' works as follows. Since at least one of (L, m, σ) and (L, m', σ') is not an entry in $Q_{\mathbf{Sig}}$, A' renames the value (L, m, σ) and rename the other (L, m', σ') (If both are not an entry in $Q_{\mathbf{Sig}}$, A' swaps the names at random). Then, A' picks up a new random element $c'' \in Z_q$, where if c'' is identical to the first $H''(L, m, A_0, A_1, a_N, b_N)$, A' halts. However, this occurs only with probability q^{-1}. Then, A' runs A again on the same random coins except that $c'' := H''(L, m, A_0, A_1, a_N, b_N)$. There is some probability that A finally outputs (L, m, σ'') (and another pair $(L, ., .)$) such that $\sigma'' = (A_1, c_N'', z_N'')$. As studied in [23], such an event happens with probability $\frac{1}{q_{H''}}\epsilon$, on the condition that A succeeds in the first run. Then, A' checks that $c_i \neq c_i''$. If $c_i = c_i''$, A'

halts, otherwise output $\frac{z_i''-z_i}{c_i-c_i''}$, which implies that A' outputs $\log_g(Y)$ on input (g, Y, G), because $a_i = g^{z_i} y_i^{c_i} = g^{z_i''} y_i^{c_i''}$ and $y_i = Y$.

We now claim that the probability that $c_i \neq c_i''$ is overwhelming in k: By lemma 3, if adversary A entraps player i, it is the case with an overwhelming probability that A has really forged the signature of player i; namely, $\log_h(\sigma_i) = \log_g(y_i)$. On one hand, since $c \neq c''$, there is at least a $t \in N$, such that $c_t \neq c_t''$. By lemma 1, however, the possibility that $\#\{i \in N | \log_h(\sigma_i) = \log_g(y_i)\} > 1$ is at most $\frac{q_{H'}}{q}$. Therefore, we conclude $t = i$ because at least, $\log_h(\sigma_i) = \log_g(y_i)$.

To sum up, the success probability of A' is bounded by

$$\frac{\epsilon^2}{q_{H''}} - \frac{1}{q} - \frac{q_{\mathbf{Sig}} q_{H''}}{q} - \frac{(n-1)(n-2)q_{H'}^2}{2q} - \frac{q_{H''}}{q} - \frac{q_{H'}}{q}.$$

To suppress the advantage of A' to be negligible in k, ϵ, the advantage of A, must be negligible in k. □

Remark 3 (On-Line Extractor). The standard rewinding strategy works well on our scheme in the game of exculpability but it only provides a loose security reduction. Actually, for adversary A that runs in time T with advantage ϵ, we construct algorithm A' breaking the discrete-log problem in time $T' \approx 2T$ with probability $\epsilon' \approx \frac{\epsilon^2}{q_H''}$ in the proof of Theorem 4. Based on Fischlin's technique [14], we can replace, at a small efficiency cost, our non-interactive zero-knowledge part in the signing protocol with one for which there is an on-line extractor; that is, one can extract the secret witness from the adversary without rewinding. Here, if A attacks the new scheme in time T with advantage ϵ, then there is algorithm A' breaking the discrete-log problem in time $T' = O(T)$ with probability $\epsilon' \approx \epsilon$.

6 Some Other Remarks

6.1 Threshold version of Traceable Ring Signature.

The extension of our proposal to a t-out-of-n traceable ring signature is straightforward. Let S be the set of t signers. First of all, each signer in S makes signature his own $\sigma_i = h^{x_i}$, where $h = H(L)$, and distributes σ_i to the other signers. Then, each signer in S computes every other signature σ_i, $i \notin S$, as point $(i, \log_h \sigma_i)$ lies on a polynomial curve of degree t, $y = \alpha(x)$, uniquely defined from $(t+1)$ points, $(0, \log_h A_0)$, $(k_1, x_{k_1}), \ldots, (k_t, x_{k_t})$, where $A_0 = H'(L, m)$ and $S = \{k_1, ..., k_t\}$. Actually, each signer in S can locally compute σ_i, $i \notin S$, as $\sigma_i = \prod_{j=0}^{t} (A_j)^{i^j} \in G$ for all $i \notin S$, where $A_0 = H(L, m) \in G$, and $A_j = \prod_{k \in S} (\sigma_k / A_0)^{m_{j,k}} \in G$ for $j = 1, ..., t$, where

$$\begin{pmatrix} m_{1,k_1} & \cdots & m_{1,k_t} \\ \vdots & \ddots & \vdots \\ m_{t,k_1} & \cdots & m_{t,k_t} \end{pmatrix} = \begin{pmatrix} {k_1}^1 & \cdots & {k_1}^t \\ \vdots & \ddots & \vdots \\ {k_t}^1 & \cdots & {k_t}^t \end{pmatrix}^{-1}$$

is the inverse matrix of van der Monde matrix. Notice that there exists a polynomial of degree t, $\alpha(x) \in \mathbb{Z}_q[x]$, such that $A_0 = h^{\alpha(0)} \in G$ and $\sigma_i = h^{\alpha(i)} \in G$ for every i. Then they collaborate and generate a NIZK based signature on (L, m), p, by applying the technique of [11], with respect to the language

$$\mathcal{L} \triangleq \{(L, h, \sigma_N)) \mid \quad \exists\, S \subset N \text{ such that } \#S \geq t \text{ and } \log_g(y_i) = \log_h(\sigma_i) \text{ for } i \in S\}.$$

Finally, the signers output signature $\sigma = (A_1, \ldots, A_t, p)$, where $p = (\beta(x), z_N)$ and $\beta(x)$ is a polynomial of degree $(n - t)$ in $\mathbb{Z}_q[x]$.

6.2 k-Times Anonymity on the Same Tag

Any traceable ring signature scheme can be efficiently transformed into a traceable ring signature scheme with k-times anonymity in the sense of [25], where the k-times anonymity means that a signer is allowed to sign messages with respect to the same tag at most k times without being traced. It is simply obtained by regarding $(i, \mathbf{Sig}_{sk}((L, i), m))$ as a signature on m, with respect to tag L, where the verifier checks if $\mathbf{Ver}((L, i), m) = 1$ and $1 \leq i \leq k$ (Here the signer need not publish i in order). It is obvious that the identity of a signer is not revealed if the signer is enough careful not to issue the same index twice on the same tag. We, however, remark that this implementation has a weakness in the unlinkability property, while it satisfies the condition of the k-time anonymity defined in [25], because whether or not the two signatures have been generated by the different signers can be easily determined, if the two signatures have the same tag and index. The scheme appeared in [25], too, substantially has the same problem.

References

1. M. Abe, M. Ohkubo, and K. Suzuki. Efficient threshold signer-ambiguous signatures from variety of keys. *IEICE Trans. Fund.*, vol.E87-A, no.2:471–479, 2004.
2. M. H. Au, S. S. M. Chow, W. Susilo, and P. P. Tsang. Short linkable ring signatures revisited. In *EUROPKI 2006*, LNCS 4043, pages 101–115, 2006.
3. M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. In *CT-RSA '05*, 2005.
4. A. Bender, J. Katz, and R. Morselli. Ring signatures:stronger definitions, and constructions without random oracles. In S. Halevi and T. Rabin, editors, *TCC 2006*, LNCS 3876, pages 60–79. Springer-Verlag, 2006.
5. S. Brands. Untraceable off-line cash in wallet with observers. In D. Stinson, editor, *CRYPTO '93*, LNCS 773, pages 302–318. Springer-Verlag, 1993.
6. E. Bresson, J. Stern, and M. Szydlo. Threshold ring signatures and applications to ad-hoc groups. In Moti Yung, editor, *CRYPTO 2002*, LNCS 2442, pages 465–480. Springer-Verlag, 2002.
7. D. Chaum. Blind signatures for untraceable payments. In D. Chaum, R. Rivest, and A. Sherman, editors, *CRYPTO '82*, pages 199–204. Prenum Publishing Corporation, 1982.
8. D. Chaum. Zero-knowledge undeniable signatures. In *EUROCRYPT 1990*, pages 458–464, 1990.

9. D. Chaum, A. Fiat, and M. Naor. Untraceable electronic cash. In S. Goldwasser, editor, *CRYPTO '88*, LNCS 403, pages 319–327. Springer-Verlag, 1990.
10. D. Chaum and E. Van Heyst. Group signatures. In D. W. Davies, editor, *EUROCRYPT '91*, LNCS 547, pages 257–265. Springer-Verlag, 1991.
11. R. Cramer, I. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO 1994*, pages 174–187, 1994.
12. I. Damgard, K. Dupont, and M. Pedersen. Unclonable group identification. In S. Vaudenay, editor, *EUROCRYPT 2006*, LNCS 4004, pages 555–572. Springer-Verlag, 2006.
13. Y. Dodis, A. Kiayias, A. Nicolosi, and V. Shoup. Anonymous identification in ad hoc groups. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, LNCS 3027, pages 609–626. Springer-Verlag, 2004.
14. M. Fischlin. Communication-efficient non-interactive proofs of knowledge with online extractor. In *CRYPTO 2005*, LNCS 3621. Springer-Verlag, 2005.
15. A. Kiayias, Y. Tsiounis, and M. Yung. Traceable signatures. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, LNCS 3027. Springer-Verlag, 2004.
16. Y. Komano, K. Ohta, A. Shimbo, and S. Kawamura. Toward the fair anonymous signatures: Deniable ring signatures. In D. Pointcheval, editor, *CT-RSA '06*, LNCS 3860, pages 174–191. Springer-Verlag, 2006.
17. J. K. Liu, V. K. Wei, and D. S. Wong. Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract). In *ACISP 2004*, LNCS 3108, pages 325–335, 2004.
18. J. K. Liu and D. S. Wong. Linkable ring signatures: Security models and new schemes. In *ICCSA 2005*, LNCS 3481, pages 614–623, 2005.
19. M. Naor. Deniable ring authentication. In *CRYPTO 2002*, pages 481–498, 2002.
20. T. Okamoto. An efficient divisible electronic cash scheme. In D. Coppersmith, editor, *CRYPTO'95*, LNCS 963, pages 438–451. Springer-Verlag, 1995.
21. T. Okamoto. Receipt-free electronic voting schemes for large scale elections. In *Security Protocols Workshop*, pages 25–35, Paris, 1997.
22. T. Okamoto and K. Ohta. Universal electronic cash. In *CRYPTO '91*, LNCS 576, pages 324–337. Springer-Verlag, 1992.
23. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 2000.
24. R. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In C. Boyd, editor, *Asiacrypt 2001*, LNCS 2248, pages 552–565. Springer-Verlag, 2001.
25. I. Teranishi, J. Furukawa, and K. Sako. k-times anonymous authentication. In P.J. Lee, editor, *Asiacrypt 2004*, LNCS 3329, pages 308–322. Springer-Verlag, 2004.
26. P. P. Tsang and V. K. Wei. Short linkable ring signatures for e-voting, e-cash and attestation. In *IPSEC 2005*, 2005.
27. P. P. Tsang, V. K. Wei, T. K. Chan, M. H. Au, J. K. Liu, and D. S. Wong. Separable linkable threshold ring signatures. In *INDCRYPT 2004*, LNCS 3348, pages 389–398, 2004.