Tracing-by-Linking Group Signatures

Victor K. Wei

Department of Information Engineering
The Chinese University of Hong Kong
Shatin, Hong Kong
`kwwei@ie.cuhk.edu.hk`

Abstract. In a group signature [10], any one out of n group members can sign on behalf of the group without revealing the identity of the signer. However, the Open Authority (OA), in possession of a certain trapdoor, can revoke the anonymity and reveal the signer identity. The individual group member is futile to resist. In this paper, we initiate the study of tracing-by-linking (TbL) group signatures. An honest signer's identify cannot be revoked by any authority. However, if a group member signs twice, these two signatures can be linked and the identity of the *double signer* can be efficiently computed without trapdoor. We introduce security models for TbL group signatures, and construct the first examples whose security are reduced to well-known assumptions. Major security notions include anonymity, full linkability, and non-slanderability. The core of our technique is the successful transplant of TbL technique from single-term offline e-cash and blind signature settings [6, 13, 12] to group signatures. Our signatures have size $O(1)$.

1 Introduction

The balance between law enforcement needs and individual privacy has been one of the most vexing research topics in cryptography – public security versus individual privacy. In Chaum and van Heyst[10]'s group signature, the authority, more specifically the open authority (OA), holds the ultimate power to revoke the anonymity of a group signature. Even if the signer does not wish to have his anonymity revokes, it is powerless to resists. On the other end of the spectrum, the signer of a ring signature [11, 19] also enjoys anonymity. That anonymity cannot be taken away by any authority or group manager without the signer's voluntary cooperation.

The overriding power of the open manager to revoke anonymity has raised concerns of some prominent cryptographers. They raised the concern that future change of political leadership may cause the revocation power to fall into the wrong hands. On the other hand, the strong anonymity present in ring signatures, guaranties by statistical zero-knowledge is too strong for authorities and regulations, even in civilian and commercial systems. Different shades of balances between the public need to regulate/revoke and the individual need to protect privacy/anonymity are important issues in applications of group signatures including e-voting, e-cash, and direct anonymous attestation (DAA) in trusted computing.

In this paper, we seek a balance between public needs to regulate and individual needs to protect privacy/anonymity. In *tracing-by-linking (TbL)* group signature, the anonymity of the signer remains protected as long as the signer does not "double sign": such as double voting in e-voting schemes, double spend in e-cash schemes, or double-attest and become a rogue in DAA. However, if an entity double signs, there is a publically computable algorithm to *link* these two offeding signatures and the double signer's identity can be publically computed from these two linked signatures.

A rather desirable balance between regulation and privacy is achieved. The privacy of honest group members is protected. No authority can revoke it. On the other hand, the identity of an offender can be computed, by authorities or by any member of the public, and penalties sought.

Our Contributions.

- We initiate the study of tracing-by-linking (TbL) group signatures. We introduce its security model, and construct the first several TbL group signatures. In particular, we modify [3]'s generic group signature, [21, 4, 5]'s SDH-based group signature, and [1]'s strong-RSA-based group signature to constant-size TbL group signatures.
- Our TbL mechanism can be incorporated to many other applications. Using it, we construct efficient TbL ring signatures, enhance direct anonymous attestation by making the identity of the rogue signer publically computable. Other applications to one-show credentials with rogue identifying, e-voting, and e-cash are discussed.

The paper is **organized** as follows: In Section 2 contains preliminaries. Section 3 contains security models. Section 4 contains constructions and security theorems. Section 6 contains discussions and conclusions.

Related results. Results on linkability sporadically appear in the literature. Nakanishi, et al. studied linkable group signatures [17, 18]. Camenisch, et al. [8]'s one-show credentials used linkability. Brickell, et al. [7]'s Direct Anonymous Attestation (DAA) contained linkability. Liu, et al. [16] studied linkable ring signature. Brands [6]'s and Ferguson's [13, ?] offline anonymous e-cash scheme had a core technique for linking two double signers. Fischlin viewed the linkability in his anonymous group identification as *unfortunate* [14], P.126, L.3. Bellare, et al. remarked that linkability and anonymity cannot coexist in their model [2], P.623. Most of these linkability techniques do not result in the tracing of the double signer after two signatures are linked. Only e-cash schemes [6, 13, 12] efficiently computes the double signer's identity without trapdoor.

In this paper, we successfully transplanted TbL techniques from e-cash which is based on blind signatures, to a TbL mechanism in the group signature setting. The essence of the TbL technique is to fix (or commit) some randomness during ecash withdrawal and then require the user to use these randomness during the first commitment move of a 3-move proof. Double spending implies answering challenges twice with this fixed commitment. That results in the extraction of the rogue identity. We call this the *commit the commitment (CtC)* technique. We transplanted CtC to group signatures, and achieved TbL.

2 Preliminaries

Due to space limitations, we keep reviews to be bare essentials. We plan to add back more details in a full version.

We say N is a *safe product* if $N = pq$, $p = 2p' + 1$, $q = 2q' + 1$, p, q, p', and q' are sufficiently large primes. The set of *quadratic residues* in Z_N, denoted QR_N, consists of all squares in Z_n which are also squares in Z_p and in Z_q.
The **Strong RSA Assumption.** There exists no PPT algorithm which, on input a random λ-bit safe product N and a random $z \in QR(N)$, returns $u \in \mathbb{Z}_N^*$ and $e \in \mathbb{N}$ such that $e > 1$ and $u^e = z(\mathrm{mod} N)$, with non-negligible probability and in time polynomial in λ.

Let $e : G_1 \times G_2 \to G_T$ be a bilinear mapping, i.e. $e(u^a, v^b) = e(u, v)^{ab}$, for $u \in G_1$, $v \in G_2$, $a, b \in Z$. The **q-Strong Diffie-Hellman Problem (q-SDH)** is the problem of computing a pair $(g_1^{1/(\gamma+x)}, x)$ given a $(q+2)$-tuple $(g_1, g_2, g_2^{\gamma}, g_2^{\gamma^2}, \cdots, g_2^{\gamma^q}$. For details, see [5].

Camenisch-Stadler proof system with Commit the Commitment (CtC). Camenisch and Stadler presented a notation for proof systems[9]. Let I, J, K be index sets, and let $\mu : I \times J \to K \cup \{\perp\}$, where the parameter $x_\perp = 0$ by convention. In a nutshell: for the signature proof system denoted $SPK\{\{x_k : k \in K\} : \wedge_{i\in I}\ y_i = \prod_{j\in J} g_j^{x_{\mu(i,j)}}\}(M)$, a proof consists of $(\{y_i : i \in I\}||\{g_j : j \in J\}||\{z_k : k \in K\}||c||M||\mu)$ satisfying $c = \mathcal{H}(\{(y_i, t_i) : i \in I\}||\{g_j : j \in J\}||M||\mu\}$ where $t_i = y_i^c \prod_{j\in J} g_j^{z_{\mu(i,j)}}$ for each $i \in I$. For details, see [9].

We introduce a new notation to incorporate **CtC (commit the commitment)**. Let I, J, K, $\{x_k : k \in K\}$, $\{g_j : j \in J\}$, $y_i : i \in I\}$, $\mu : i \times J \to K$ be defined as above. Let k_1, k_2, k_3, k_4 be four distinct elements of K, and h_1 and h_2 be two fairly generated bases distinct from $\{g_j : j \in J\}$. For the signature proof system denoted

$$SPK\{\{x_k : k \in K\} : \\ (\bigwedge_{i\in I} y_i = \prod_{j\in J} g_j^{x_{\mu(i,j)}}) \wedge CtC(x_{k_1}, x_{k_2}, x_{k_3}, x_{k_4}, h_1, h_2)\}(M) \quad (1)$$

a proof consists of

$$\{y_i : i \in I\}||\{g_j : j \in J\} \\ ||\{z_k : k \in K\}||c||M||label||\mu||\tilde{y}||(k_1, k_2, k_3, k_4, h_1, h_2) \quad (2)$$

(where *label* is randomly generated by signer) satisfying

$$c = \mathcal{H}(\{(y_i, t_i) : i \in I\} \\ ||\{g_j : j \in J\}||M||label||\mu||\tilde{y}||\tilde{t}||\tilde{u}||(k_1, k_2, k_3, k_4, h_1, h_2)\} \quad (3)$$

where $t_i = y_i^c \prod_j g_j^{z_{\mu(i,j)}}$ for each $i \in I$, and

$$\tilde{t} = \tilde{y}^c h_1^{\tilde{z}_1} h_2^{\tilde{z}_2}, \quad (4)$$
$$\tilde{u} = \tilde{t}^c h_1^{\tilde{z}_3} h_2^{\tilde{z}_4}, \quad (5)$$

where $\tilde{z}_\ell = z_{k_\ell}$ is the response corresponding to the secret x_{k_ℓ}, for ℓ=1,2,3,4.

Abbreviate proof-of-knowledge as PoK. The value $\tilde{t}$ is the commitment in the 3-move (sub-)PoK $\{(x_{k_1}, x_{k_2}) : \tilde{y} = h_1^{x_{k_1}} h_2^{x_{k_2}}\}$. It is also "committed" itself as the *masked image* in the (sub-)PoK $\{(x_{k_3}, x_{k_4}) : \tilde{t} = h_1^{x_{k_3}} h_2^{x_{k_4}}\}$. Therefore the name *CtC (Commit the Commitment)*. Signing twice, i.e. completing SPK twice with different (M, *label*) pairs, results in the efficient extraction of the secrets x_{k_1} and x_{k_2}, without using any trapdoor.

Example: $SPK\{(x_1, x_2, x_3, x_4)$: $y_1 = g_1^{x_1} g_2^{x_2} \quad \wedge \quad y_2 = g_2^{x_3} g_3^{x_4} \quad \wedge \quad y_3 = g_1^{x_2} g_3^{x_3} g_5^{x_4} \quad \wedge \quad CtC(x_1, x_2, x_3, x_4, h_1, h_2)\}(M)$. Note I={1,2,3}, J={1,2,3,4,5}, K={1, 2, 3, 4}, μ maps (1,1), (1,2), (2,2), (2,3), (3,1), (3,3), (3,5) to 1,2,3,4,2,3,4, respectively, and $\mu(i,j) = \perp$ elsewhere. A proof consists of

$$(y_1, y_2, y_3)||(g_1, g_2, g_3, g_4, g_5)||(z_1, z_2, z_3, z_4)||c||M||label||\tilde{y}||(1, 2, 3, 4, h_1, h_2) \quad (6)$$

(where *label* is randomly generated by the signer) satisfying

$$c = \mathcal{H}((y_1, t_1, y_2, t_2, y_3, t_3)||(g_1, g_2, g_3, g_4, g_5)||M||label||\mu||\tilde{y}||\tilde{t}||\tilde{u}||(1, 2, 3, 4, h_1, h_2))$$

where $t_1 = g_1^{z_1} g_2^{z_2} y_1^c$, $t_2 = g_2^{z_3} g_3^{z_4} y_2^c$, $t_3 = g_1^{z_2} g_3^{z_3} g_5^{z_4} y_3^c$, and $\tilde{t} = h_1^{z_1} h_2^{z_2} \tilde{y}^c$, $\tilde{u} = h_1^{z_3} h_2^{z_4} \tilde{t}^c$.

Let us demonstrate how to simulate a proof, Eq. (1): Assume μ, g_j's, and k_1, k_2, k_3, k_4, h_1, h_2 are given. The inputs to the SHVZK (special honest verifier zero-knowledge) simulator are c and $\{y_i : i \in I\}$.

1. Randomly generate $\tilde{y}$, z_{k_1}, and z_{k_2}. Compute $\tilde{t} = h_1^{z_{k_1}} h_2^{z_{k_2}} \tilde{y}^c$.
2. Randomly generate z_{k_3} and z_{k_4}. Compute $\tilde{u} = h_1^{z_{k_3}} h_2^{z_{k_4}} \tilde{t}^c$.
3. Randomly generate z_k, $k \in K \setminus \{k_1, k_2, k_3, k_4\}$. Compute $t_i = \prod_j g_j^{z_{\mu(i,j)}} y_i^c$, each $i \in I$.
4. Output the proof according to Equation (2).

The simulated proof is has a non-negligible statistically distance from the real-world proof generated with known secrets x_k, $k \in K$. However, it can be shown that the simulated proof is computationally indistinguishable from the real-world proof provided the DDH (Decisional Diffie-Hellman) Problem is hard.

3 Security Model

We present a security model for the tracing-by-linking (TbL) group signature. In a nutshell, we start with [3]'s security model for group signatures, and replace their pair of security notions, full traceability and non-frameability, by a new pair, full linkability and non-slanderability.

Motivated by DAA (Direct Anonymous Attestation) [7], our system consists of three types of entities in multitudes:

– Managers of Groups, or, equivalently, Certificate Authorities (CA's), with serial number cnt which stands for *counter value*.
– Users, or, equivalently, TPM (Trusted Platform Module), whose serial number is id.

– Verifiers with serial number bsn which stands for *basename*.

Remark: We have in mind the following: An individual user has one personal key pair, $(usk_{\mathsf{id}}, upk_{\mathsf{id}})$. It can get one certificate from each CA. Verifiers bsn will verify the proof-of-knowledge of possession of a certificate from cnt, in each execution run.

Syntax. A TbL group signature is a tuple (Init, GKg, UKg, Join, Iss, GSig, GVf, Link, Indict) where:

- Init: $1^\lambda \rightarrow \mathsf{param}$. On input the security parameter 1^λ, Protocol init generates public system parameters param. Included: an efficiently samplable one-way NP-relation whose specification is $\langle \mathcal{R}_{user} \rangle$, an efficiently samplable family of trapdoor one-way NP-relations whose specifications constitute $\mathcal{F}$=$\{\langle \mathcal{R}_{CA,i} \rangle : i\}$ and whose trapdoors are denoted gsk_i's.
- GKg:cnt$\xrightarrow{\$} \langle \mathcal{R}_{CA,\mathsf{cnt}} \rangle$. On input cnt, Protocol GKg samples $\mathcal{F}$ to get a relation whose specification is $\langle \mathcal{R}_{CA,\mathsf{cnt}} \rangle$ and whose trapdoor is gsk_{cnt}, and adds an entry (cnt, $\langle \mathcal{R}_{CA,cnt} \rangle$) to the public system parameters param. By convention, $\langle \mathcal{R}_{CA,\mathsf{cnt}} \rangle$ includes gpk_{cnt}.
- UKg: id $\xrightarrow{\$} (usk_{\mathsf{id}}, upk_{\mathsf{id}}) \in \mathcal{R}_{user}$. Protocol UKg accepts input id to sample a key pair from $\mathcal{R}_{user}$, adds an entry $(\mathsf{id}, upk_{\mathsf{id}})$ to the public system parameters param.
- Join,Iss is a pair of interactive protocols with common inputs cnt and id, and Iss's addition input gsk_{cnt}, and Join's additional inputs usk_{id}. At the conclusion, join obtains extended secret key $xsk_{\mathsf{id,cnt}}$, extended public key $xpk_{\mathsf{id,cnt}}$ which includes a certificate $cert_{\mathsf{id,cnt}}$ satisfying $(xpk_{\mathsf{id,cnt}}, cert_{\mathsf{id,cnt}}) \in \mathcal{R}_{CA,\mathsf{cnt}}$ and $(xsk_{\mathsf{id,cnt}}, xpk_{\mathsf{id,cnt}}) \in \mathcal{R}_{user}$, such that Iss does not know $xsk_{\mathsf{id,cnt}}$, and an entry $(\mathsf{id}, \mathsf{cnt}, xpk_{\mathsf{id,cnt}})$ is added to the public system parameters param. Below, we may sometimes use the notations xpk (resp. xsk) and upk (resp. usk) interchangeably without ambiguity from context.
- GSig: (id, cnt, $xsk_{\mathsf{id,cnt}}$, bsn, M) $\rightarrow \sigma$. It takes inputs id, cnt, $xsk_{\mathsf{id,cnt}}$, bsn, and a message M, returns a signature σ. By convention, the (extended) signature σ includes cnt, bsn, and M.
- GVf: $\sigma \rightarrow 0$ or 1. It takes input a signature σ, returns either 1 or 0 for valid or invalid.
- Link: $(\sigma, \sigma') \rightarrow 0$ or 1. It takes inputs two valid signatures, σ and σ', returns either 1 or 0 for linked or unlinked.
- Indict: $(\sigma, \sigma') \rightarrow \mathsf{id}$. It takes two valid and linked signatures σ and σ', returns id.

Notions of Security

The security notion **Correctness** is defined as follows:

Definition 1. *For i=1,2, let σ_i=GSig (id$_i$, cnt$_i$, $xsk_{\mathsf{id}_i,\mathsf{cnt}_i}$, bsn$_i$, M_i). A TbL group signature has* verification correctness *if GVf (σ_1)=1 with probability one. It has* linking correctness *if Link $(\sigma_1, \sigma_2) = 0$ with overwhelming probability when id$_1 \neq$ id$_2$ or cnt$_1 \neq$ cnt$_2$ or bsn$_1 \neq$ bsn$_2$. It has* indictment correctness *if Link $(\sigma_1, \sigma_2) = 1$ and Indict (σ_1, σ_2) = id$_1$ with overwhelming probability when*

$\mathsf{id}_1 = \mathsf{id}_2$ and $\mathsf{cnt}_1 = \mathsf{cnt}_2$ and $\mathsf{bsn}_1 = \mathsf{bsn}_2$. It is correct *if it has verification correctness, linking correctness, and indictment correctness.*

The following **oracles** define the attacker's capabilities.

– The *Random Oracle*: We use the Random Oracle normally.
– The *Corruption Oracle*: $\mathcal{CO} : (\mathsf{id}, \mathsf{cnt}) \rightarrow xsk_{\mathsf{id},\mathsf{cnt}}$. Upon input the id of a group member, it outputs xsk_{id}.
– The *Signing Oracle* $\mathcal{SO} : (\mathsf{id}, \mathsf{cnt}, \mathsf{bsn}, M) \rightarrow \sigma$. Upon inputs user id, CA cnt, Verifier bsn, and a message M, it outputs a valid signature, provided the public system parameters param contains an entry for $(\mathsf{id}, \mathsf{cnt})$.

Linkable Anonymity. Anonymity for TbL group signature is defined in the following experiment.

Experiment LA.
1. (*Initialization Phase*) Simulator $\mathcal{S}$ invokes Init, GKg, gets gsk, invokes UKg (resp. Join,Iss) $g_u \geq 2$ times to generate a set of joined users, denoted JU_S, with their extended user public keys xpk_{id}'s.
2. (*Probe-1 Phase*) $\mathcal{A}$ gets gsk and uses it to create and join new uers, and $\mathcal{A}$ queries the oracles.
3. (*Gauntlet Phase*) $\mathcal{A}$ selects two *gauntlet users* id_0, $\mathsf{id}_1 \in JU_S$, a message M, and give them to $\mathcal{S}$. Then $\mathcal{S}$ flips a fair coin $b \in \{0, 1\}$ and returns the *gauntlet signature* $\sigma_g = \mathcal{SO}(\mathsf{id}_b, M)$.
4. (*Probe-2 Phase*) $\mathcal{A}$ queries the oracles.
5. (*End Game*) $\mathcal{A}$ delivers an estimate $\hat{b} \in \{0, 1\}$ of b.

Oracle queries can be arbitrarily interleaved across Probe-1, Gauntlet, and Probe-2 Phases. $\mathcal{A}$ *wins* the Experiment LA if $\hat{b} = b$, id_0 (resp. id_1) has never been queried to $\mathcal{CO}$ or $\mathcal{SO}$. $\mathcal{A}$'s *advantage* is its winning probability minus half.

Definition 2 (Linkable Anonymity). *An TbL group signature is* linkably anonymous *if no PPT adversary has a non-negligible advantage in Experiment LA.*

Linkability, full linkability. Roughly speaking, full linkability means that any coalition without the group manager, in possession of q_ℓ user secret keys, can produce $q_\ell + 1$ valid signatures that are pairwise unlinked.

Formally, *full linkability* for TbL group signature is defined in terms of the following experiment.

Experiment FL.
1. $\mathcal{S}$ invokes Init, GKg, invokes UKg (resp. Join,Iss) q_u times, to generate a set JU of q_u users.
2. $\mathcal{A}$ makes q_k (resp. q_h, q_s) queries the Corruption (resp. Random, Signing) Oracle with arbitrary interleaf.
3. $\mathcal{A}$ delivers signatures σ_i, $1 \leq i \leq n'$, none of which is the output of a $\mathcal{SO}$ query.

Let $q_{s,trace}$ denote the number of users $\mathsf{id} \in JU$ for whom $\mathcal{A}$ queried $\mathcal{SO}$, i.e. $\mathcal{SO}(\mathsf{id}, \mathsf{cnt}, \mathsf{bsn}, M)$, more than once with the same triple $(\mathsf{id}, \mathsf{cnt}, \mathsf{bsn})$. $\mathcal{A}$ wins Experiment FL if it delivers $n' > q_k + q_{s,trace}$ valid signatures which are pairwise unlinked, i.e $\mathsf{GVf}(\sigma_i)$=1, for all i, $1 \le i \le q_k + q_{s,trace} + 1$; and $\mathsf{Link}(\sigma_i, \sigma_j)$=0, for all i, j, $1 \le i < j \le q_k + q_{s,trace} + 1$. The Adversary's *advantage* is its probability of winning.

Definition 3 (Full Linkability). *A TbL group signature is* fully linkable *if no PPT adversary has a non-negligible advantage in winning Experiment FL.*

Non-slanderability In a nutshell, non-slanderability means a coalition of users together with the group manager cannot produce signatures that are linked and indicted to a group member outside the coalition. Formally,

Experiment NS/SbV
1. $\mathcal{S}$ invokes Init, GKg, invokes UKg (resp. Join,Iss) q_u times, to generate a set JU_{static} of q_u users and give JU_{static} to Adversary $\mathcal{A}$. $\mathcal{S}$ also gives the group secret key gsk to $\mathcal{A}$.
2. $\mathcal{A}$ queries the oracles in arbitrary interleaf.
3. $\mathcal{A}$ delivers two signatures, σ and σ', that are valid, linked, and indicted to user $\mathsf{id}_g \in JU_{static}$.

$\mathcal{A}$ wins Experiment NS if it has never queried id_g to $\mathcal{CO}$, and it has never queried $\mathcal{SO}$ more than once with any triple $(\mathsf{id}_g, \mathsf{cnt}, \mathsf{bsn})$, for any cnt, bsn. $\mathcal{A}$'s *advantage* is his probability of winning. $\mathcal{A}$ wins the harder Experiment SbV if it wins Experiment NS and, additionally, one of the two delivered signatures is the output of a query to $\mathcal{SO}$.

Definition 4. *A TbL group signature is* non-slanderable *if no PPT adversary has a non-negligible advantage in Experiment NS. It is* slanderable-but-vindicatable (SbV)*, or simply* vindicatable*, if no PPT adversary has a non-negligible advantage in Experiment SbV.*

Remark: For an SbV group signature, the following scenario cannot be ruled out: $\mathcal{A}$ produces two signatures which are valid, linked, and indict to id who is never corrupted or *traced*. But id can vindicate itself by some efficient algorithm. However, slander-then-vindicate remains a hassle if not a vulnerability. It is best to achieve non-slanderability. [20] contained a TbL group signature which is SbV but not non-slanderable. The size of their signature is $O(n)$ where $n = |JU_{static}|$.

Linkable Security. Summarizing we have:

Definition 5 (Security). *A TbL group signature is* linkably secure *if it is correct, linkably anonymous, fully linkable, and non-slanderable. It is* linkably SbV-secure *if it is correct, linkably anonymous, fully linkable, and slanderable-but-vindicatable.*

Related security notions. Non-slanderability implies that *indictment* is accurate with overwhelming probability: If two valid and linked signatures indict to id_g, then non-slanderability implies no one except id_g could have generated them, with overwhelming probability. Full linkability specializes to *unforgeability* when $q_k + q_{s,trace} = 0$, and it specializes to *ordinary* linkability when $q_k + q_{s,trace} = 1$. *Exculpability* of *openable* group signature means that a coalition of users, together with the group manager (and the open authority) cannot produce a signature traced to an uncorrupted member. The TbL group signature does not have the "open signature" functionality, and the notion of exculpability is, in a sense, absorbed into into non-slanderability. [15]'s *misidentification attack* is related to a slander.

4 Constructions

We construct several TbL group signatures with provable security, including some with $O(1)$ size. For simplicity, fix cnt and bsn and omit them from notations.

4.1 A generic construction: the TbL-generic group signature

Given an efficiently samplable family of trapdoor one-way NP-relations whose specifications constitute $\mathcal{F} = \{\langle \mathcal{R}_{CA,i} \rangle\}$, and an efficiently samplable one-way NP-relation whose specification is $\langle \mathcal{R}_{user} \rangle$, construct a TbL group signature (Init, GKg, UKg, Join, Iss, GSig, GVf, Link, Indict) as follows:

Init, GKg, UKg are as specified in the Syntax. In particular, the public system parameters param include a full-domain secure hashing function $\mathcal{H}_b$.

Join,Iss are as follows:

1. Common inputs are cnt, id, param, $g_i = \mathcal{H}_b(g, \mathsf{cnt}, i)$. Additional input to Join (resp. Iss) is usk_{id} (resp. gsk_{cnt}).
2. Join randomly generates s_3, s_4, s_5, set $s_1 = usk_{\mathsf{id}}$, computes $s_2 = \mathsf{id} \cdot s_3$, $y_i = g_i^{s_i}$, $1 \le i \le 5$. Sends $xpk = (y_1, \cdots, y_5)$ and a proof-of-knowledge $\mathrm{PK}\{(s_2, s_3) : y_2 = g_2^{s_2} \ \wedge \ y_3 = g_3^{s_3} \ \wedge \ s_2 = \mathsf{id} \cdot s_3\}$ to Iss.
3. Iss verifies (id, cnt) has not been entered in param, verifies the proof-of-knowledge, then generates and sends cert satisfying $(xpk, cert) \in \mathcal{R}_{CA,\mathsf{cnt}}$. It adds the entry $(\mathsf{id}, xpk_{\mathsf{id,cnt}})$ to the public system parameters param. By convention, $xpk_{\mathsf{id,cnt}}$ includes $cert_{\mathsf{id,cnt}}$.

GSig: Upon inputs id, cnt, xsk_{id}, and a message M, it obtains $xpk_{\mathsf{id,cnt}}$, which includes $cert_{\mathsf{id,cnt}}$, from param and then returns a signature σ computed as the following signature proof-of-knowledge (SPK):

$$\begin{aligned} &SPK\{(xsk, xpk, cert) : xsk = (usk, s_2, s_3, s_4, s_5) \\ &\wedge xpk = (upk, g_2^{s_2}, g_3^{s_3}, g_4^{s_4}, g_5^{s_5}) \wedge (usk, upk) \in \mathcal{R}_{user} \wedge (xpk, cert) \in \mathcal{R}_{CA} \\ &\wedge CtC(s_2, s_3, s_4, s_5, h_1, h_2) \wedge g_i = \mathcal{H}_b(g, \mathsf{cnt}, i) \wedge h_i = \mathcal{H}_b(h, \mathsf{cnt}, \mathsf{bsn}, i)\}(M) \end{aligned} \quad (7)$$

By convention, σ includes cnt, bsn, and M.

GVf: Given a signature σ, check whether it is a signature proof-of-knowledge of Relation (7).

Link: Accepting two valid signatures $\sigma^{(b)}$ containing $(\tilde{y}^{(b)}, \tilde{t}^{(b)})$, b=1,2, it outputs linked if $\tilde{y}^{(1)} = \tilde{y}^{(2)}$ and $\tilde{t}^{(1)} = \tilde{t}^{(2)}$. It outputs $\mathsf{unlinked}$ otherwise. Note $\tilde{y}^{(b)}$ and $\tilde{t}^{(b)}$ are part of the CtC contained in the signatures according to Eq. (7).

Indict: Accepting two valid and linked signatures $\sigma^{(b)}$ containing $(\tilde{y}^{(b)}, \tilde{t}^{(b)})$, b=1,2, it uses the soundness of the CtC relation to extract the secrets s_2 and s_3. Then it indicts user $\mathsf{id} = s_2/s_3$.

4.2 Instantiation in SDH setting: The TbL-SDH group signature

We instantiate the generic constructing in Sec. 4.1 in the SDH group [21, 4, 5]. Let $\hat{e} : G_1 \times G_2 \rightarrow G_T$ be a bilinear mapping. The relations are instantiated as follows:

$$\begin{aligned}
\mathcal{R}_{user} &= \{(xsk, xpk) : xsk = (x_1, \cdots, x_5), xpk = (y_1, \cdots, y_5), \text{ each } y_i = g_{i,5}^{x_i}\} \\
\mathcal{R}_{CA,\mathsf{cnt}} &= \{(xpk, cert) : xpk = (y_1, \cdots, y_5), cert = ((A_1, e_1), \cdots, (A_5, e_5))), \\
&\qquad \text{each } A_i^{e_i+\gamma_i} y_i = g_{i,6}\}
\end{aligned}$$

gpk includes $u_1^{\gamma_1}, \cdots, u_5^{\gamma_5} \in G_2$, $gsk = (\gamma_1, \cdots, \gamma_5)$. Note $g_{i,j} = \mathcal{H}_b(g, \mathsf{cnt}, i, j) \in G_1$, $u_i = \mathcal{H}_b(u, \mathsf{cnt}, i) \in G_2$.

Init, GKg are the same as Sec. 4.1.

UKg: On input id, sample the relation $\mathcal{R}'_{user} = \{(x, y) : y = g_{i,5}^{x}\}$ to produce $(usk_{\mathsf{id}}, upk_{\mathsf{id}}) \in \mathcal{R}'_{user}$.

Join,Iss accepts common inputs cnt and id, Join's additional input usk_{id}, Iss's additional input gsk_{cnt}. Randomly generate x_2, x_3, x_4, compute $x_5 = \mathsf{id} \cdot x_4$, $y_i = g_{i,5}^{x_i}$ for all i. Let x_1=usk_{id}, y_1=upk_{id}, $xpk_{\mathsf{id},\mathsf{cnt}} = (x_1, \cdots, x_5)$, $ypk_{\mathsf{id},\mathsf{cnt}} = (y_1, \cdots, y_5)$. Note $(xsk_{\mathsf{id},\mathsf{cnt}}, xpk_{\mathsf{id},\mathsf{cnt}}) \in \mathcal{R}_{user}$, Join sends $xpk_{\mathsf{id},\mathsf{cnt}}$ and proof-of-knowledge $x_2 = x_3 \cdot \mathsf{id}$. Iss verifies, and returns $cert_{\mathsf{id},\mathsf{cnt}}$ satisfying $(xpk_{\mathsf{id},\mathsf{cnt}}, cert_{\mathsf{id},\mathsf{cnt}}) \in \mathcal{R}_{CA,\mathsf{cnt}}$. Note $cert_{\mathsf{id},\mathsf{cnt}} = ((A_1, e_2), \cdots, (A_5, e_5))$ and $A_i^{e_i+\gamma_i} y_i = u_i$, $u_i = \mathcal{H}_b(u, \mathsf{cnt}, i)$. Iss enters the entry $(\mathsf{id}, \mathsf{cnt}, xpk_{\mathsf{id},\mathsf{cnt}}, cert_{\mathsf{id},\mathsf{cnt}})$ into the public param.

GSig accepts input id, cnt, $xsk_{\mathsf{id},\mathsf{cnt}}$, bsn, message M. It outputs signature σ which is a signature proof of knowledge of the following system. (It essentially quintuplicates the proof system in [21, 5] and add CtC.)

$$\begin{aligned}
&SPK\{\{(x_i, A_i, e_i) : 1 \leq i \leq 5\}) : \\
&\quad (\wedge_{i:1 \leq i \leq 5} A_i^{\gamma_i + e_i} g_{i,5}^{x_i} = g_{i,6} \in G_1) \;\wedge\; CtC(x_5, x_4, x_3, x_2, h_1, h_2)\}(M)
\end{aligned}$$

where $gpk_{\mathsf{cnt}} = (u_1^{\gamma_1}, \cdots, u_5^{\gamma_5})$, $u_i = \mathcal{H}_b(u, \mathsf{cnt}, i)$, $h_i = \mathcal{H}_b(h, \mathsf{cnt}, \mathsf{bsn}, i)$.

Further details of GSig: Below, let $x_{i,5} = x_i$, $x_{i,3} = A_i$, $x_{i,4} = e_i$, $x_{i,1} = s_{i,1}$, $x_{i,2} = s_{i,2}$, each i. Denote (k_1,k_2,k_3,k_4)=((5,5), (4,5), (3,5), (2,5)). Note $(x_{k_1}, x_{k_2}, x_{k_3}, x_{k_4})$= (x_5, x_4, x_3, x_2). Compute the masked images $y_{i,j}$'s as follows:

$$\begin{gathered}
y_{i,1} = g_{i,1}^{s_{i,1}}, y_{i,2} = A_i g_{i,2}^{s_{i,1}}, y_{i,3} = y_{i,2}^{e_i} \\
y_{i,4} = [\hat{e}(g_{i,2}, u_i^{\gamma_i})\hat{e}(g_{i,4}, u_i)]^{s_{i,1}} \hat{e}(g_{i,2}, u_i)^{s_{i,2}}, \quad y_{i,5} = g_{i,1}^{e_i} g_{i,3}^{s_{i,1}}, \\
y_{i,6} = g_{i,5}^{x_i} g_{i,4}^{s_{i,1}}, \quad \text{all } i; \tilde{y} = h_1^{x_5} h_2^{x_4}, \quad \tilde{t} = h_1^{x_3} h_2^{x_2}
\end{gathered} \tag{8}$$

where $s_{i,2} = e_i s_{i,1}$, $g_{i,1} = g_i$, for i=1,2,3,4,5. Compute the commitments:

$$t_{i,1} = g_{i,1}^{r_{i,1}}, \quad t_{i,2} = R_i g_{i,2}^{r_{i,1}}, \quad t_{i,3} = y_{i,2}^{r_{e_i}},$$
$$t_{i,4} = [\hat{e}(g_{i,2}, u_i^{\gamma_i})\hat{e}(g_{i,4}, u_i)]^{r_{i,1}} \hat{e}(g_{i,2}, u_i)^{r_{i,2}}, \quad t_{i,5} = g_{i,1}^{r_{e_i}} g_{i,3}^{r_{i,1}}, \quad t_{i,6} = g_{i,5}^{r_{x_i}} g_{i,4}^{r_{i,1}},$$
$$\text{all } i; \tilde{u} = h_1^{r_{x_3}} h_2^{r_{x_2}}, \quad r_{x_5} = x_3, \quad r_{x_4} = x_2$$

Compute the challenge:

$$c = \mathcal{H}(\{(y_{i,j}, t_{i,j}) : 1 \le i \le 5, 1 \le j \le 6\}$$
$$||\{g_{i,j} : 1 \le i \le 5, 1 \le j \le 4\}||M||label||\mu||\tilde{y}||\tilde{t}||\tilde{u}||(k_1, k_2, k_3, k_4, h_1, h_2)\} \quad (9)$$

where *label* is randomly generated by the signer. Compute the responses:

$$z_{i,1} = r_{i,1} - cs_{i,1}, \quad z_{i,2} = r_{i,2} - cs_{i,2}, \quad z_{i,3} = R_i A_i^{-c}, \quad z_{i,4} = r_{e,i} - ce_i,$$
$$z_{i,5} = r_{x,i} - cx_i, \quad \text{all } i;$$

Output the signature σ:

$$\mathsf{cnt}||\mathsf{bsn}||\{y_{i,j} : 1 \le i \le 5, 1 \le j \le 6\}||\{g_{i,j} : 1 \le i \le 5, 1 \le j \le 4\}$$
$$||\{z_{i,j} : 1 \le i \le 5, 1 \le j \le 5\}||c||M||label||\tilde{y}||(k_1, k_2, k_3, k_4, h_1, h_2) \quad (10)$$

Protocol GVf accepts input σ, parses it, verifies $y_{i,4}$=$\hat{e}(y_{i,3}y_{i,6}g_{i,6}^{-1}, u_i)\ \hat{e}(y_{i,2}, u_i^{\gamma_i})$, and computes

$$t_{i,1} = g_{i,1}^{z_{i,1}} y_{i,1}^c, \quad t_{i,2} = z_{i,3} g_{i,2}^{z_{i,1}} y_{i,2}^c, \quad t_{i,3} = y_{i,2}^{z_{i,4}} y_{i,3}^c,$$
$$t_{i,4} = [\hat{e}(g_{i,2}, u_i^{\gamma_i})\hat{e}(g_{i,4}, u_i)]^{z_{i,1}} \hat{e}(g_{i,2}, u_i)^{z_{i,2}} y_{i,4}^c, \quad t_{i,5} = g_{i,1}^{z_{i,4}} g_{i,3}^{z_{i,1}} y_{i,5}^c, \quad (11)$$
$$t_{i,6} = g_{i,5}^{z_{i,5}} g_{i,4}^{z_{i,1}} y_{i,5}^c, \quad \tilde{t} = h_1^{z_{5,5}} h_2^{z_{4,5}} \tilde{y}^c, \quad \tilde{u} = h_1^{z_{3,5}} h_2^{z_{2,5}} \tilde{t}^c$$

Then it computs c according to Eq. (9), verifies the computed c equals to the value c contained in the signature σ. According to the outcome of this verification, outputs valid or $\mathsf{invalid}$.

Link accepts inputs $\sigma^{(b)}$, b=1,2, parses them according to Eq. (10) into $\sigma^{(b)}$= $\mathsf{cnt}^{(b)}$ || $\mathsf{bsn}^{(b)}$ || $\cdots$. Confirms $\mathsf{GVf}(\sigma^{(b)})$=1, b=1,2, and confirms $\mathsf{cnt}^{(1)} = \mathsf{cnt}^{(2)}$, $\mathsf{bsn}^{(1)} = \mathsf{bsn}^{(2)}$, $g_{i,j}^{(1)} = g_{i,j}^{(2)}$, $k_i^{(1)} = k_i^{(2)}$, $h_i^{(1)} = h_i^{(2)}$, all i, j. Then computes $\tilde{t}^{(b)} = h_1^{z_{5,5}^{(b)}} h_2^{z_{4,5}^{(b)}} (\tilde{y}^{(b)})^c$, b=1,2, Outputs linked if all confirmations above pass and $(\tilde{y}^{(1)}, \tilde{t}^{(1)}) = (\tilde{y}^{(2)}, \tilde{t}^{(2)})$; outputs $\mathsf{unlinked}$ otherwise. (By convention, identical signatures are linked.)

Indict accepts inputs $\sigma^{(b)}$, b=1,2, parses them like Link, confirms $\mathsf{Link}(\sigma^{(1)}, \sigma^{(2)}) = 1$ and confirms $c^{(1)} \neq c^{(2)}$. It outputs id=$(z_{5,5}^{(2)} - z_{5,5}^{(1)})\ (z_{4,5}^{(2)} - z_{4,5}^{(1)})^{-1} \in Z_{q_1}$, where $q_1 = \text{order}(G_1)$.

4.3 Instantiating in strong-RSA: The TbL-ACJT group signature

We also instantiate the generic TbL group signature of Section 4.1 to the setting similar to ACJT's group signature[1]. Let N be a safe product. The relations are instantiated as follows:

$$\mathcal{R}_{user} = \{(xsk, xpk) : xsk = (x_1, \cdots, x_9), xpk = (y_1, \cdots, y_9),\ \text{each } y_i = g_{i,5}^{x_i}\}$$
$$\mathcal{R}_{CA,\mathsf{cnt}} = \{(xpk, cert) : xpk = (y_1, \cdots, y_9), cert = ((A_1, e_1), \cdots, (A_9, e_9))),\ \text{each } A_i^{e_i} = y_i g_{i,6} \bmod N_{\mathsf{cnt}}\}$$

The group public key $gpk_{\mathsf{cnt}} = N_{\mathsf{cnt}}$. The group secret key, gsk, is the factoring of N_{cnt}. Note $g_{i,j} = \mathcal{H}_b(g, \mathsf{cnt}, i, j) \in QR_{N_{\mathsf{cnt}}}$.

Init, GKg are the same as Sec. 4.1.

UKg: On input id, sample the relation $\mathcal{R}'_{user} = \{(x, y) : y = g_{i,5}^{x}\}$ to produce $(usk_{\mathsf{id}}, upk_{\mathsf{id}}) \in \mathcal{R}'_{user}$.

Join,Iss accepts common inputs cnt and id, Join's additional input usk_{id}, Iss's additional input gsk_{cnt}. Randomly generate x_2, x_3, x_4, x_6, x_7, x_8, compute $x_5 = \mathsf{id} \cdot x_4$, $x_9 = \mathsf{id} \cdot x_8$, $y_i = g_{i,5}^{x_i}$ for all i. Let $x_1{=}usk_{\mathsf{id}}$, $y_1{=}upk_{\mathsf{id}}$, $xpk_{\mathsf{id,cnt}} = (x_1, \cdots, x_9)$, $ypk_{\mathsf{id,cnt}} = (y_1, \cdots, y_9)$. Note $(xsk_{\mathsf{id,cnt}}, xpk_{\mathsf{id,cnt}}) \in \mathcal{R}_{user}$, Join sends $xpk_{\mathsf{id,cnt}}$ and proof-of-knowledge $x_5 = x_4 {\cdot} \mathsf{id} \wedge x_9 = \mathsf{id} {\cdot} x_8$. Iss verifies, and returns $cert_{\mathsf{id,cnt}}$ satisfying $(xpk_{\mathsf{id,cnt}}, cert_{\mathsf{id,cnt}}) \in \mathcal{R}_{CA,\mathsf{cnt}}$. Note $cert_{\mathsf{id,cnt}} = ((A_1, e_2), \cdots, (A_9, e_9))$ and $A_i^{e_i} = y_i g_{i,6}$. Iss enters the entry $(\mathsf{id}, \mathsf{cnt}, xpk_{\mathsf{id,cnt}}, cert_{\mathsf{id,cnt}})$ into the public param.

GSig accepts input id, cnt, $xsk_{\mathsf{id,cnt}}$, bsn, message M. It outputs signature σ which is a signature proof of knowledge of the following system. (It essentially duplicates nine-fold the proof system in [1] and add CtC.)

$$SPK\{\{(x_i, A_i, e_i) : 1 \le i \le 9\}) : (\wedge_{i:1 \le i \le 9}\ A_i^{e_i} = g_{i,5}^{x_i} g_{i,6} \bmod N)$$
$$\wedge\ CtC(x_5, x_4, x_3, x_2, h_1, h_2)\ \wedge\ CtC(x_9, x_8, x_7, x_6, h_3, h_4)\}(M)$$

where $h_i = \mathcal{H}_b(h, \mathsf{cnt}, \mathsf{bsn}, i) \in QR_{N_{\mathsf{cnt}}}$. Masked images, commitments, challenge, responses, and the signature are

$$y_{i,1} = g_{i,1}^{s_{i,1}}, y_{i,2} = A_i g_{i,2}^{s_{i,1}}, y_{i,3} = y_{i,2}^{e_i}, y_{i,4} = g_{i,5}^{x_i} g_{i,2}^{s_{i,2}},$$
$$y_{i,5} = g_{i,1}^{e_i} g_{i,3}^{s_{i,1}}, y_{i,6} = g_{i,5}^{x_i} g_{i,4}^{s_{i,1}}, s_{i,2} = s_{i,1} e_i,\ \text{all } i;$$
$$\tilde{y} = h_1^{x_5} h_2^{x_4}, \tilde{t} = h_1^{x_3} h_2^{x_2}, \bar{y} = h_3^{x_9} h_4^{x_8}, \bar{t} = h_3^{x_7} h_4^{x_6},$$

$$t_{i,1} = g_{i,1}^{r_{i,1}}, t_{i,2} = R_i g_{i,2}^{r_{i,1}}, t_{i,3} = y_{i,2}^{r_{e_i}}, t_{i,4} = g_{i,1}^{r_{x_i}} g_{i,2}^{r_{i,2}}, t_{i,5} = g_{i,1}^{r_{e_i}} g_{i,3}^{r_{i,1}},$$
$$t_{i,6} = g_{i,5}^{r_{x_i}} g_{i,4}^{r_{i,1}},\ \text{all } i; \tilde{u} = h_1^{r_{x_3}} h_2^{r_{x_2}}, \bar{u} = h_1^{r_{x_7}} h_2^{r_{x_6}},$$
$$r_{x_5} = x_3, r_{x_4} = x_2, r_{x_9} = x_7, r_{x_8} = x_6$$

$$c = \mathcal{H}(\{(y_{i,j}, t_{i,j}) : 1 \le i \le 9, 1 \le j \le 6\} || \{g_{i,j} : 1 \le i \le 9, 1 \le j \le 4\}$$
$$|| M || label || \mu || \tilde{y} || \tilde{t} || \tilde{u} || (k_1, k_2, k_3, k_4, h_1, h_2) || \bar{y} || \bar{t} || \bar{u} || (k_5, k_6, k_7, k_8, h_3, h_4)\}$$

$$z_{i,1} = r_{i,1} - c s_{i,1}, \quad z_{i,2} = r_{i,2} - c s_{i,2}, \quad z_{i,3} = R_i A_i^{-c}, \quad z_{i,4} = r_{e,i} - c e_i,$$
$$z_{i,5} = r_{x,i} - c x_i,\ \text{all } i;$$

$$\sigma = (\{y_{i,j} : 1 \le i \le 9, 1 \le j \le 6\} || \{g_{i,j} : 1 \le i \le 9, 1 \le j \le 4\}$$
$$|| \{z_{i,j} : 1 \le i \le 9, 1 \le j \le 5\} || c || M || label || \tilde{y} || (k_1, k_2, k_3, k_4, h_1, h_2)$$
$$|| \bar{y} || (k_5, k_6, k_7, k_8, h_3, h_4))$$

GVf accepts input σ, parses it, verifies $y_{i,4} = y_{i,3}b^{-1}$, all i, and computes

$$t_{i,1} = g_{i,1}^{z_{i,1}} y_{i,1}^c,\ \ t_{i,2} = z_{i,3} g_{i,2}^{z_{i,2}} y_{i,2}^c,\ \ t_{i,3} = y_{i,2}^{z_{i,4}} y_{i,3}^c,\ \ t_{i,4} = g_{i,1}^{z_{i,5}} g_{i,2}^{z_{i,2}} y_{i,4}^c,$$
$$t_{i,5} = g_{i,1}^{z_{i,4}} g_{i,3}^{z_{i,1}} y_{i,5}^c, t_{i,6} = g_{i,5}^{z_{i,5}} g_{i,4}^{z_{i,1}} y_{i,5}^c,\ \text{all } i; \tilde{t} = h_1^{z_{5,5}} h_2^{z_{4,5}} \tilde{y}^c,$$
$$\tilde{u} = h_1^{z_{3,5}} h_2^{z_{2,5}} \tilde{t}^c, \bar{t} = h_3^{z_{9,5}} h_4^{z_{8,5}} \bar{y}^c, \bar{u} = h_3^{z_{7,5}} h_4^{z_{6,5}} \bar{t}^c$$

Then it computs c by mimicking GSig, verifies the computed c equals to the value c contained in the signature σ. It outputs 1 or valid if all verifications are OK. Otherwise, it outputs 0 or invalid.

Link accepts inputs $\sigma^{(b)}$, b=1,2, parses them according to Eq. (10) into $\sigma^{(b)}$= $\mathsf{cnt}^{(b)}$ || $\mathsf{bsn}^{(b)}$ || $\cdots$. Confirms GVf($\sigma^{(b)}$)=1, b=1,2, and confirms $\mathsf{cnt}^{(1)} = \mathsf{cnt}^{(2)}$, $\mathsf{bsn}^{(1)} = \mathsf{bsn}^{(2)}$, $g_{i,j}^{(1)} = g_{i,j}^{(2)}$, $k_i^{(1)} = k_i^{(2)}$, $h_i^{(1)} = h_i^{(2)}$, all i, j. Then computes $\tilde{t}^{(b)} = h_1^{z_{5,5}^{(b)}} h_2^{z_{4,5}^{(b)}} (\tilde{y}^{(b)})^c$, b=1,2, Outputs linked if all confirmations above pass and $(\tilde{y}^{(1)}, \tilde{t}^{(1)}) = (\tilde{y}^{(2)}, \tilde{t}^{(2)})$; outputs unlinked otherwise. (By convention, identical signatures are linked.)

Indict accepts inputs $\sigma^{(b)}$, b=1,2, parses them like Link, confirms Link($\sigma^{(1)}$, $\sigma^{(2)}$) = 1 and confirms $c^{(1)} \neq c^{(2)}$. It obtains $\mathsf{id} \cdot \Delta_4 = \Delta_5$ and $\mathsf{id} \cdot \Delta_8 = \Delta_9$, where $\Delta_i = z_{i,5}^{(2)} - z_{i,5}^{(1)}$, $1 \leq i \leq 9$. Using Euclidean algorithm it computes α and β satisfying $\alpha\Delta_4 + \beta\Delta_8 = \gcd(\Delta_4, \Delta_8) \in Z$. Hypothesizing Δ_i's are random, there is a $\epsilon_0 = \Theta(1)$ probability that $\gcd(\Delta_4, \Delta_8) = 1$. When this happens, Indict outputs $\mathsf{id}_g = \alpha\Delta_5 + \beta\Delta_9 \in Z$. Then $\mathsf{id} = \mathsf{id}_g$ if and only if $h_1^{\mathsf{id}} = h_1^{\mathsf{id}_g} \bmod N$, and id can be identified.

The above achieves indictment accuracy ϵ_0 using two units of CtC's. If an indictment accuracy $1 - \epsilon_I$ is desired, one can modify TbL-ACJT by using $O(\log(1/\epsilon_I))$ units of CtC's.

4.4 Security Theorems

Theorem 1. *Let e be a bilinear map, $e : G_1 \times G_2 \to G_T$. Assume the Random Oracle model. The TbL-SDH group signature has* correctness. *It is* linkably anonymous *if and only if the DDH Assumption holds in G_1. It is* fully linkable *provided the q-SDH Assumption holds in $G_1 \times G_2$. It is* non-slanderable *provided CDH is hard in G_1. In summary, the TbL-SDH group signature is* linkably secure *provided the DDH Assumption in G_1 and the q-SDH Assumption in $G_1 \times G_2$ both hold.*

Theorem 2. *Let N be a safe product. Assume the Random Oracle model. The TbL-ACJT group signature has* correctness. *It is* linkably anonymous *if and only if the DDH Assumption holds in QR_N. It is* fully linkable *under the strong RSA Assumption. It is* non-slanderable, *provided CDH is hard in QR_N. In summary, the TbL-ACJT group signature is* linkably secure *provided the DDH Assumption and the strong-RSA Assumption both hold in QR_N.*

Proofs are contained in the Appendix.

5 Concluding Discussions

We successfully transplanted "commit the commitment" (CtC) technique from a blind signature and e-cash setting [6, 13, 12] to the group signature setting. The result is a tracing-by-linking (TbL) group signature. So far, we have only presented signature schemes that output the double signer's identity when linked. But the CtC technique can be easily adapted to output the double signer's secret key for even harsher punishment, or to output just the double signer's public key. To do so, simply change the CtC commitments from $(x_{k_1}, x_{k_2}) = (\mathsf{id} \cdot s,\, s)$ to $(usk \cdot s,\, s)$, or $(upk \cdot s,\, s)$.

The TbL technique has many applications. The TbL group signature constitutes a natural **e-voting** system: simply collect and tally the signatures. The voting is anonymous, and double voting results in the identification of the culprit. TbL group signature is also a natural offline anonymous **e-cash** system. Withdrawal is essentially equivalent to Join,Iss. Spending is essentially equivalent to signing. Double spending results in revealed identities and penalties can be sought. The TbL group signature is also a DAA (Dirct Anonymous Attestation) system. Have multiple CA's in DAA is equivalent to having multiple groups in simultaneous existence. Attesting to a Basename bsn is equivalent to signing for Verifier bsn. Current DAA schemes [7] only tags the rogue who double-attests without revealing its identity. Our **Tracing-by-Linking DAA** systems can be constructed to efficiently compute the identity, or even the secret key, of the rogue without trapdoor. Such stronger deterrent makes superior DAA's. The commit-the-commitment technique can also be used to enhance one-show credential systems [8] by adding the feature to efficiently compute offender's identity without trapdoor.

References

1. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *CRYPTO 2000*, pages 255–270. Springer-Verlag, 2000.
2. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: formal definitions, simplified requirements and a construction based on general assumptions. In *EUROCRYPT'03*, volume 2656 of *LNCS*. Springer-Verlag, 2003.
3. M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: the case of dynamic groups. Cryptology ePrint Archive, Report 2004/077, 2004. http://eprint.iacr.org/.
4. D. Boneh and X. Boyen. Short signatures without random oracles. In *Eurocrypt 2004*, volume 3027 of *LNCS*, pages 56–73. Springer–Verglag, 2004.
5. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *CRYPYTO 2004*, volume ???? of *LNCS*. Springer–Verglag, 2004.
6. S. Brands. Untraceable off-line cash in wallet with observers. In *CRYPTO'93*, pages 302–318. Springer-Verlag, 1993.
7. E. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. Cryptology ePrint Archive, Report 2004/205, 2004. http://eprint.iacr.org/.

8. J. Camenisch and A. Lysyanskaya. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118. Springer-Verlag, 2001.
9. J. Camenisch and M. Stadler. Proof systems for general systems of discrete logarithms. ETH Technical Report No. 260, 1997. ftp://ftp.inf.ethz.ch/pub/publications/tech-reports/.
10. D. Chaum and E. van Heyst. Group signatures. In *EUROCRYPT'91*, volume 547 of *LNCS*, pages 257–265. Springer-Verlag, 1991.
11. R. Cramer, I. Damgard, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO'94*, pages 174–187. Springer-Verlag, 1994.
12. N. Ferguson. Extensions of single-term coins. In *CRYPTO'93*, pages 292–301. Springer-Verlag, 1993.
13. N. Ferguson. Single term off-line coins. In *Eurocrypt'93*, pages 318–328. Springer-Verlag, 1993.
14. M. Fischlin. The Cramer-Shoup strong-RSA signature scheme revisited. In *PKC*, pages 116–129, 2003.
15. A. Kiayias and M. Yung. The vector-ballot e-voting approach. In *FC 2004*, volume 3110 of *LNCS*, pages 72–89. Springer-Verlag, 2004.
16. J. K. Liu, V. K. Wei, and D. S. Wong. Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract). In *ACISP'04*, volume 3108 of *LNCS*, pages 325–335. Springer-Verlag, 2004.
17. T. Nakanishi, T. Fujiwara, and H. Watanabe. A linkable group signature and its application to secret voting. In *4th Int'l Symp. on Communicatin Theory and Appl.*, 1997.
18. T. Nakanishi, T. Fujiwara, and H. Watanabe. A linkable group signature and its application to secret voting. *Trans. of Information Processing Society of Japan*, 40(7):3085–3096, 1999.
19. R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *ASIACRYPT 2001*, pages 552–565. Springer-Verlag, 2001.
20. P. P. Tsang, V. K. Wei, M. H. Au, T. K. Chan, J. K. Liu, and D. S. Wong. Separable linkable threshold ring signatures. In *Indocrypt 2004*, 2004. To appear.
21. F. Zhang, R. Safavi-Naini, and W. Susilo. An efficient signature scheme from bilinear pairings and its applications. In *PKC 2004*, 2004.

A Proofs

A.1 Proof Sketch of Theorem 1

The intuition is that TbL-SDH's GSig does more things than [5]'s GSig. So [5]'s full traceability (resp. non-frameability) essentially implies TbL-SDH's full linkability (resp. non-slanderability). By the same token, TbL-SDH's anonymity is harder to accomplish than that of [5].

First, a technical lemma. Let $(g, g^\alpha, g^\beta, g^\gamma, g^\delta)$ be such that α, β, δ are random and $\gamma \in \{\alpha\beta,\ \delta\beta\}$ with equal probability. The **DDDH (Double Decisional Diffie-Hellman) Assumption** is that no PPT algorithm can determine which of the two equalities hold for γ with probability non-negligibly over half. Note α, β, γ, δ are not given, only their exponentiations are. Note the *DDH (Decisional*

Diffie-Hellman) Assumption is that no PPT algorithm can distinguish the following two cases with non-negligible probability over half: Let $(g, g^\alpha, g^\beta, g^\gamma)$ be such that α, β, are random and $\gamma{=}\alpha\beta$ or γ is random with half-half probability. Determine which is the case for γ.

Lemma 3 *The DDDH Assumption holds if and only if the DDH Assumption holds.*

Proof of Lemma 3: That the DDH Assumption implied the DDDH Assumption is straightforward. We proceed to the opposite direction. Let $\mathcal{B}$ be a PPT solver of the DDDH Problem. Consider its performance when given the following double DDH Problem: [DDH1: $(g,\ g^\alpha,\ g^\beta,\ g^{\gamma_1})$] and [DDH2: $(g,\ g^\delta,\ g^\beta,\ g^{\gamma_2})$], where α, β, δ are random, and $\gamma_1{=}\alpha\beta$ or is random with half-half probability, and $\gamma_2{=}\alpha\delta$ or is random with half-half probability. Let $\gamma \in \{\gamma_1,\ \gamma_2\}$ with half-half probability. Give the "generalized" DDDH Problem [GDDDH: $(g,\ g^\alpha,\ g^\beta,\ g^\gamma,\ \delta)$] to $\mathcal{B}$ to solve. With probability half, GDDDH is a DDDH Problem, and in that case $\mathcal{B}$ solves it with probability $1/2 + \epsilon_1$. Otherwise, GDDDH is not a DDDH Problem, and let us consider $\mathcal{B}$'s performance in this scenario. Let ϵ_2 denote the probability $\mathcal{B}$ outputs $\perp$ meaning the problem is not DDDH. $\epsilon_2 = 0$ if he is not allowed to do so. Then $\mathcal{B}$ outputs either DDDH decision with equal probability $(1 - \epsilon_2)/2$ because there is a symmetry w.r.t. the two cases.

Let us build an algorithm $\mathcal{B}'$ to solve DDH1 and DDH2: $\mathcal{B}'$ outputs yes to DDH1 (resp. DDH2) if $\mathcal{B}$ outputs g^{γ_1} (resp. g^{γ_2}) on input GDDDH, and $\mathcal{B}'$ outputs no otherwise. Then

$$\frac{1}{2}\Pr\{\mathcal{B}' \text{ solves DDH1}\} + \frac{1}{2}\Pr\{\mathcal{B}' \text{ solves DDH2}\}$$
$$= \frac{1}{2}\Pr\{\mathcal{B} \text{ solves DDDH}\} + \epsilon_2 + \frac{1}{2}(\frac{1}{2} - \epsilon_2)$$

Therefore, $\mathcal{B}'$ has a probability non-negligibly over half of solving either DDH Problem. □

Now we proceed to prove Theorem 1. First, **correctness** is straightforward.

Anonymity. Let the DDDH Problem be to determine whether $\gamma = \alpha\beta$ or $\delta\beta$, given $(\hat{g},\ \hat{g}^\alpha,\ \hat{g}^\beta,\ \hat{g}^\gamma,\ \hat{g}^\delta)$ and $\gamma \in \{\alpha\beta, \delta\beta\}$ with 50-50 probability. All elements are in G_1. Note α, β, γ, δ are not given.

Simulator $\mathcal{S}$: Denote the DDDH Problem as $(\hat{g},\ \hat{g}^\alpha,\ \hat{g}^\beta,\ \hat{g}^\gamma,\ \hat{g}^\delta)$ where $\gamma \in \{\alpha\beta, \delta\beta\}$ with equal probability. $\mathcal{S}$ randomly generates ρ_g, ρ_h, ρ_3, and backpatches the Random Oracle to $g_{4,5} = \mathcal{H}_c(4,5) = \hat{g}$, $g_{5,5} = \mathcal{H}_c(5,5) = g_{4,5}^{\rho_g}$, $h_2 = \mathcal{H}_v(2) = \hat{g}^\beta$, $h_1 = \mathcal{H}_v(1) = h_2^{\rho_h^{-1}}$. During the Initialization Phase of Experiment LA, $\mathcal{S}$ selects two users id_0 and id_1 and sets the public key of user id_b, b=0,1, to satisfy $g_{4,5}^{x_4^{(0)}} = \hat{g}^\alpha$, $g_{4,5}^{x_4^{(1)}} = \hat{g}^\delta$, $g_{5,5}^{x_5^{(b)}} = g_{4,5}^{x_4^{(b)} \cdot \mathsf{id}_b \rho_g} = (\hat{g}^\alpha)^{\mathsf{id}_b \rho_g}$. $\mathcal{S}$ randomly generates other secrets x_i's, A_i's, e_i's, and computes their corresponding certificates using gsk in its possession.

There is a non-negligible probability that the gauntlet users selected by $\mathcal{A}$ in the Gauntlet Phase are id_0 and id_1. Without loss of generality, we consider only this scenario, for the sake of simplicity. The Corruption Oracle is simulated by simply giving Adversary the secret user keys it wants.

Simulating a sequence of $\mathcal{SO}$ queries w.r.t. the same user: We consider the computations of all $\mathcal{SO}$ queries w.r.t. the same id together. In this consideration, $\mathcal{SO}$ accepts inputs id, messages $M^{(\tau)}$, to output signatures $\sigma^{(\tau)}$, where $1 \leq \tau \leq q_{s,\mathsf{id}}$, and $q_{s,\mathsf{id}}$ is the number of times id is queried to $\mathcal{SO}$, as follows:

1. For each new id queried to $\mathcal{SO}$, randomly generate $\tilde{x}_{\mathsf{id}}$ and compute $\tilde{y}_{\mathsf{id}} = h_1^{\tilde{x}_{\mathsf{id}}}$. Below, we abbreviate the notations to $\tilde{y} = \tilde{y}_{\mathsf{id}}$ and $\tilde{x} = \tilde{x}_{\mathsf{id}}$ without contextual ambiguity.
2. Obtain user id's public keys $(g_i^{x_i},\ A_i,\ e_i)$'s from the public database of all joined users, JU. Randomly generate masking values $s_{i,1}$'s and $s_{i,2}$'s. Compute all masked images $y_{i,j}$'s by EQ. (8).
3. Randomly generate challenge $c^{(\tau)}$, $1 \leq \tau \leq q_{s,\mathsf{id}}$.
4. Generate the first signature $\sigma^{(1)}$ as follows:
   (a) Randomly generate $z_{i,j}^{(1)}$'s.
   (b) Compute all commitments $t_{i,j}^{(1)}$'s according to Eq. (11), including
   $$\tilde{t}^{(1)} = h_1^{z_{5,5}^{(1)}} h_2^{z_{4,5}^{(1)}} \tilde{y}^{c^{(1)}}, \tag{12}$$
   (c) Backpatch the hash output in Eq. (9) to the generated value $c^{(1)}$ above in Step (3). Output signature $\sigma^{(1)}$ according to Eq. (10).
5. Generate the second signature $\sigma^{(2)}$ as follows:
   (a) Randomly generate $z_{5,5}^{(2)}$ and $z_{4,5}^{(2)}$ satisfying
   $$0 = \Delta_{z_{5,5}} + \rho_h \Delta_{z_{4,5}} + \Delta_c \tilde{x} \tag{13}$$
   where $\Delta_{z_{i,5}} = z_{i,5}^{(2)} - z_{i,5}^{(1)}$, all i, and $\Delta_c = c^{(2)} - c^{(1)}$.
   (b) Randomly generate other $z_{i,j}^{(2)}$'s. Compute all commitments $t_{i,j}^{(2)}$'s according to Eq. (11),
   (c) Backpatch the hash output in Eq. (9) to the generated value $c^{(2)}$ above in Step (3). Output signature $\sigma^{(2)}$ according to Eq. (10).
6. The signature $\sigma^{(\tau)}$, $\tau \geq 3$, is generated similarly: Randomly generate $z_{5,5}^{(\tau)}$ and $z_{4,5}^{(\tau)}$ satisfying
   $$0 = \Delta_{z_{5,5}}^{(\tau)} + \rho_h \Delta_{z_{4,5}}^{(\tau)} + \Delta_c^{(\tau)} \tilde{x} \tag{14}$$
   where $\Delta_{z_{i,5}}^{(\tau)} = z_{i,5}^{(\tau)} - z_{i,5}^{(1)}$, all i, and $\Delta_c^{(\tau)} = c^{(\tau)} - c^{(1)}$. Generate other $z_{i,j}^{(\tau)}$'s, compute $t_{i,j}^{(\tau)}$'s, backpatch to $c^{(\tau)}$, output $\sigma^{(\tau)}$.

Note Eqs. (14) implies $\tilde{t}^{(\tau)} = \tilde{t}^{(1)}$. The CtC structure in the signature implies the following: When $\mathcal{A}$ queries $\mathcal{SO}$ with the same id twice or more, it is equivalent to rewinding $\mathcal{SO}$ with the same commitment $\tilde{t}$ to extract $\tilde{y}$'s secrets (x_5, x_4), where $\tilde{y} = h_1^{x_5} h_2^{x_4}$. According to well-known theory, such rewinding extracts either secrets (x_5, x_4) or the discrete log $\rho = \log_{h_1} h_2$. In this case, the latter, not the former, is extracted. Therefore, $\mathcal{SO}$ can be simulated without knowing (x_5, x_4). That other secrets are not needed in simulating $\mathcal{SO}$ follows from [5].

Setting up the Gauntlet to solve DDH: The gauntlet users id_0 and id_1 are never queried to $\mathcal{CO}$ or $\mathcal{SO}$. During the Gauntlet Phase of Experiment LA, $\mathcal{S}$ computes the gauntlet signature σ_g by the following special procedure:

1. Set $\tilde{y} = \hat{g}^{\gamma}$.
2. Obtain user id_0's (resp, id_1's) extended public keys , ($g_i^{x_i}$, A_i, e_i)'s, from the public database of all joined users, JU. Randomly generate masking values $s_{i,1}$'s and $s_{i,2}$'s. Compute all masked images $y_{i,j}$'s by EQ. (8).
3. Randomly generate challenge c.
4. Generate the gauntlet signature σ_g as follows:
   (a) Randomly generate $z_{i,j}$'s.
   (b) Compute all commitments $t_{i,j}$'s according to Eq. (11), and compute $\tilde{t} = h_1^{z_{5,5}} h_2^{z_{4,5}} \tilde{y}^c$,
   (c) Backpatch the hash output in Eq. (9) to the generated value c above. Output signature σ_g according to Eq. (10).

Note $\mathcal{S}$ does not flip a fair coin b. In a sense, it outsources the flipping to the poser of the DDDH Problem.

Solving the DDDH Problem: $\mathcal{S}$ computes as above. When $\mathcal{A}$ returns $\hat{b}$, $\mathcal{S}$ answers $\gamma = \alpha\beta$ (resp. $\gamma = \delta\beta$) if $\hat{b} = 0$ (resp. $\hat{b} = 1$).

The Simulator $\mathcal{S}$'s *advantage* in solving DDDH is his probability of answering correctly, minus half. That $\mathcal{S}$'s advantage in solving DDDH equals $\mathcal{A}$'s advantage in Experiment LA is implied by the following Lemma. (The proof of the Lemma is mechanical and tedious. We omit it for this version of the paper.)

Lemma 4 *If the DDDH Problem instance is $\gamma = \alpha\beta$ (resp. $\gamma = \delta\beta$), then the gauntlet signature is computationally indistinguishable from $\mathcal{SO}(\mathsf{id}_0,M)$ (resp. $\mathcal{SO}(\mathsf{id}_1,M)$) in the Random Oracle model.*

The above Lemma implies that the TbL-SDH group signature is linkably anonymous provided the DDDH Problem is hard in the Random Oracle model. Now we prove the opposite direction. Assume $\mathcal{A}$ can solve DDH with non-negligible advantage. Then, given the gauntlet signature σ_g in Experiment LA, $\mathcal{A}$ proceeds as follows: Parse σ_g to obtain $\tilde{y}$. Solve the DDH Problem ($g_{5,5}^{\mathsf{id}_0} g_{4,5}$, $g_{5,5}^{x_5} g_{4,5}^{x_4} = (g_{5,5}^{\mathsf{id}_0} g_{4,5}^{x_4}$, $h_1^{\mathsf{id}_0} h_2$, $\tilde{y}$). If the DDH answer is yes (resp. no), answer $\hat{b} = 0$ (resp. 1) in Experiment LA. By Lemma 4, $\mathcal{A}$'s advantage in Experiment LA equals its advantage in solving DDH. Therefore, the TbL-SDH group signature is linkably anonymous if and only if the DDH Assumption holds in G_1.

Non-slanderability. Let the CDH Problem be to compute $\hat{g}^{\alpha\beta}$ given $\hat{g}$, $\hat{g}^{\alpha}$, $\hat{g}^{\beta}$. Note α and β are not given. Let $\mathcal{A}$ be a PPT adversary who has non-negligible advantage in Experiment NS. Backpatch the Random Oracle to $h_2 = \mathcal{H}_v(2) = \hat{g}^{\beta}$ and $g_{4,5} = \mathcal{H}_c(4,5) = \hat{g}$. Set the user public key so that $g_{4,5}^{x_4} = \hat{g}^{\alpha}$. To assist $\mathcal{A}$, publish $s_{4,1}$ and x_5. When $\mathcal{A}$ returns a valid signature σ, parse it to obtain $\tilde{y} = h_1^{x_5} h_2^{x_4}$. Then $\hat{g}^{\alpha\beta} = \tilde{y} h_1^{-x_5} = h_2^{x_4}$ solves the CDH Problem.

Full linkability. Assume PPT Adversary $\mathcal{A}$ has a non-negligible advantage in Experiment FL, with $q_{FL} = q_k + q_{s,trace}$. I.e. $\mathcal{A}$ makes q_k queries to the Corruption Oracle and makes double, and therefore tracing, queries $\mathcal{SO}(b, \mathsf{id})$, $b = 1$, for a total number of $q_{s,trace}$ different users' id's, to produce $q_{FL} + 1$ valid and pairwise unlinked signatures. We show how to build a simulator $\mathcal{S}$ to solve $(q_{FL} - 1)$-SDH.

Let $\hat{e} : G_1 \times G_2 \to G_T$, and $\psi : G_2 \to G_1$ be a homomorphism, with $\psi(w_i) = v_i$. Let $i \in \{1, \cdots, 5\}$. Randomly generate π_i, ρ_i, and backpatch the Random Oracle to $g_{i,5} = \mathcal{H}_c(i,5) \leftrightarrow g_{i,6}^{\pi_i}$ and $g_{i,6} = \mathcal{H}_c(i,6) \leftrightarrow v_i^{\rho_i}$. Let the SDH Problem inputs be v_i, $w_i^{\gamma_i^j}$, $0 \le j \le q$, $q = q_{FL} - 1$. Randomly generate $e_{i,u}$, $x_{i,u}$, $1 \le u \le q+1$. Compute $f_i(\sigma) = \prod_{u=1}^{q+1} (\sigma + e_{i,u}) = \sum_{j=0}^{q+1} a_{i,j} \sigma^j \in Z_{q_1}[\sigma]$, where $q_1 = \text{order}(G_1)$. Let

$$\hat{w}_i = w_i^{f_i(\sigma)}|_{\sigma=\gamma_i} \tag{15}$$

$$\begin{aligned} A_{i,u} &= \psi(\hat{w}_i)^{(1-x_{i,u}\pi_i)/(\gamma_i+e_{i,u})} \\ &= \psi(w_i)^{f_i(\sigma)(1-x_{i,u}\pi_i)/(\sigma+e_{i,u})}|_{\sigma=\gamma_i}. \end{aligned} \tag{16}$$

Then we have $q + 1$ user secret keys $(x_{i,u}, A_{i,u}, e_{i,u})$, for each i, satisfying $A_{i,u}^{\gamma_i + e_{i,u}} g_{i,5}^{x_{i,u}} = g_{i,6}$, $1 \le u \le q+1$. Note $A_{i,u}$'s are computable by Eq. (16) and $\hat{g}_1$ is computable by Eq. (15).

$\mathcal{A}$ has a non-negligible advantage in Experiment FL. Therefore, it can be rewound to extract a new set of user secrets $(\bar{x}, \bar{A}, \bar{e}, \bar{s}_1, \bar{s}_2)$ satisfying $\bar{A}^{\gamma_i + \bar{e}}\ g_{i,5}^{\bar{x}}\ g_{i,2}^{\bar{s}_2 - \bar{s}_1 \bar{e}} = g_{i,6}$ for some i. Let $\Delta = \bar{s}_2 - \bar{s}_1 \bar{e}$. If $\Delta = 0$, then $\bar{A}^{\gamma_i + \bar{e}} g_{i,5}^{\bar{x}} = g_{i,6}$ and the pair $(\hat{A}, \bar{e})$ solves the SDH Problem where $\hat{A} = \bar{A}^{(\rho_i - \rho_i \pi_i \bar{x})^{-1}}$.

If $\Delta \ne 0$, then the tuple $(\tilde{x}, \tilde{A}, \bar{e})$ satisfies $\tilde{A}^{\bar{e} + \gamma_i} g_{i,5}^{\tilde{x}} = g_{i,6}$, where $\tilde{x} = \hat{x}/(1 - \rho\Delta)$, $\tilde{A} = \bar{A}^{1/(1-\rho\Delta)}$. The pair $(\hat{A}, \bar{e})$ solves the SDH Problem where $\hat{A} = \tilde{A}^{(\rho_i - \rho_i \pi_i \tilde{x})^{-1}}$. □

Boneh, et al. [5] presented a short group signature, and proved its full traceability among other things. In their Section 7, they extended their scheme to achieve full exculpability, but without proof of that extension. Our proof of full linkability above is also a complete proof of the full traceability of the exculpable extension, or Section-7 extension, of [5]'s short group signature. Therefore we have

Corollary 5 *The strongly exculpable extension of Boneh, et al.'s short group signature [5], Section 7, has full traceability, provided the q-SDH Assumption holds, in the random oracle model.*

See [5] for formal definitions of the terms.

A.2 Proof sketch of Theorem 2

Proofs of **correctness**, **anonymity**, **non-slanderability** are similar to those of Theorem 1, and are omitted.

Full linkability. Assume Adversary $\mathcal{A}$ has a non-negligible advantage in Experiment FL. Randomly generate π_i, ρ_i, and backpatch the Random Oracle to $g_{i,5} = g_{i,6}^{\pi_i}$ and $g_{i,2} = g_{i,6}^{\rho_i}$. Rewind $\mathcal{A}$ to extract a new set of committed user secrets $\{(\bar{A}_i,\ \bar{e}_i,\ \bar{x}_i,\ \bar{s}_{i,1},\ \bar{s}_{i,2})$: all $i\}$, one it did not corrupt or trace. We have $y_{i,3}{=}(\bar{A} g_{i,2}^{\bar{s}_{i,1}})^{\bar{e}_i} {=} g_{i,6} y_{i,4} {=} g_{i,6} g_{i,5}^{\bar{x}_i} g_{i,2}^{\bar{s}_{i,2}}$. Then $\bar{A}_i^{\bar{e}_i} = g_{i,6}^{\Delta}$ where $\Delta = 1 + \bar{x}\pi_i + \Delta' \rho_i$, $\Delta' = \bar{s}_{i,2} - \bar{e}\bar{s}_{i,1}$.

Rewind $\mathcal{A}$ to obtain $\bar{A}^{(2)}$, $\bar{e}^{(2)}$, $\Delta^{(2)}$ satisfying $(\bar{A}^{(2)})^{\bar{e}^{(2)}} = g_{i,6}^{\Delta^{(2)}}$. Hypothesizing Δ and $\Delta^{(2)}$ are random, there is a $\Theta(1)$ probability that they are co-prime. In this case, use the Euclidean algorithm to compute α, β such that $\alpha\Delta + \beta\Delta^{(2)}{=}1$ and $\square$