Group Signature with Deniability:
How to Disavow a Signature

Ai Ishida [*] Keita Emura [†] Goichiro Hanaoka [‡] Yusuke Sakai [§]

Keisuke Tanaka [¶]

January 19, 2015

Abstract

Group signature is a class of digital signatures with enhanced privacy. By using this type of signature, a user can prove membership of a specific group without revealing his identity, but in the case of a dispute, for a given signature, an authority can expose the identity of its signer. However, in some situations wherein it is necessary to only know whether a specified user is the signer of the given signature, the naive use of a group signature may be problematic since if the specified user is NOT the actual signer, then the identity of the actual signer will be exposed.

In this paper, inspired by this problem, we propose the notion of a *deniable group signature*, where with respect to a signature and a user, the opener can issue a proof that the opening result of the signature is NOT the specified user without revealing the actual signer. We also describe a fairly practical construction by extending the Groth group signature scheme (ASIACRYPT 2007). In particular, a denial proof in our scheme consists of 96 group elements, which is about twice the size of a signature in the Groth scheme. The proposed scheme is provably secure under the same assumptions as those of the Groth scheme.

Keywords: group signature, deniability, non-interactive zero-knowledge proof, bilinear map

1 Introduction

1.1 Background and Motivation

Anonymity is often required in various applications in which the users' personal information or privacy should be protected, and *group signature* is one of the most popular cryptographic tools for yielding anonymity. By using a group signature, a user can generate digital evidence (i.e., signature) that proves that he/she is a member of a specified group without revealing his/her identity. Furthermore, if needed, an authority can open the "embedded" identity from the above-mentioned signature and generate another digital evidence (i.e., opening proof) of this opening result. Therefore, in normal situations, users can anonymously prove their membership, and in the case of incidents (e.g., crimes), the identity of the actual signer can be revealed. Such a property seems quite useful for protecting the users' anonymity and tracing malicious users simultaneously. However, there are also some situations in which this property does not work effectively.

Roughly speaking, in a standard group signature, it is implicitly assumed that in the case of a crime, the signer of the signature related to this crime, is the suspect, and thus, the signer's identity of this signature

[*] Tokyo Institute of Technology, Japan. ishida0@is.titech.ac.jp

[†] National Institute of Information and Communications Technology (NICT), Japan. k-emura@nict.go.jp

[‡] National Institute of Advanced Industrial Science and Technology (AIST), Japan. hanaoka-goichiro@aist.go.jp

[§] National Institute of Advanced Industrial Science and Technology (AIST), Japan. The fourth author is supported by a JSPS Fellowship for Young Scientists. yusuke.sakai@aist.go.jp

[¶] Tokyo Institute of Technology, Japan. keisuke@is.titech.ac.jp

must be exposed. *However, this is not always the case.* In other words, in case the police has already detected a suspect of a crime and is going to check his alibi, a signature that the suspect claims to be his alibi might not actually be his, and if the identity of the actual signer is revealed, it will be *a serious violation of privacy.*

More concretely, assuming that the police needs to know whether the suspect was in a specific apartment at the time of the crime, and the entrance and exit control for the apartment is managed using a group signature, a naive way to confirm this is to ask the authority (i.e., the manager of the apartment) to just reveal the identities of *all* signatures that were used for the authentication of the entrance within that specified time period. Obviously, this results in the privacy invasion of innocent users.

For avoiding such a wag-the-dog situation, it is further required to yield a functionality for generating yet another kind of digital evidence, which *only* proves that for a given signature and an identity of a suspect, the actual signer of the signature is NOT the suspect.

1.2 Our Contribution

In this paper, we describe the construction of a group signature that yields the above-mentioned functionality. In particular, we propose a notion of a *deniable group signature*, the methodology for designing it, and a concrete instantiation. In addition to *all* the functionalities of a standard group signature, a deniable group signature yields another functionality that the authority (i.e., opener) can generate a *denial proof* that proves non-ownership of a signature. In other words, in a deniable group signature, for a given signature and an identity of a user, the authority can generate a proof for the fact that the actual signer is NOT that particular user (if this is the case).

We first discuss that it is possible to generically construct such a group signature by extending the Bellare-Shi-Zhang technique [5] and clarify the main difficulty for designing practical instantiations. In particular, we point out that it is not straightforward to apply the Groth-Sahai proofs [26] for generating denial proofs. The problem here is to find a way to prove that a user i is not the actual signer j, i.e., a non-equal statement, $i \neq j$, but such kind of language is not covered with the Groth-Sahai proof. Then, we show a concrete deniable group signature scheme based on the (modified) Groth group signature [25, 41] together with a dedicated technique for overcoming the above-mentioned difficulty.

The proposed scheme is provably secure in the standard model under the decisional linear (DLIN) assumption, q-strong Diffie-Hellman (q-SDH) assumption, q-U assumption, universal one-wayness of hash functions, and strong unforgeability of one-time signatures. The denial proof in this scheme consists of nine commitments, four pairing product equations, and five multi-scalar multiplication equations. The total size is 96 group elements and is about twice the size of a group signature in the Groth scheme.

1.3 Related Work

Group signatures [16, 43, 5, 4, 9, 35] and ring signatures [39, 18, 3, 17] are the most popular cryptographic primitives for anonymity. In the former case, a central authority called group manager is defined, which is divided into two roles, namely issuing certificates (issuer) and opening signatures (opener). Each user can generate a group signature, which is anonymously verifiable by using a membership certificate issued by the issuer, and the opener can identify the actual signer. In the latter case, each user has a pair of public key and secret key, and no central authority is required. A signer picks a set of users containing himself, say S, and generates a ring signature by using his own secret key and the pubic keys $\{pk_i\}_{i \in S}$. A verifier can verify whether a signer is contained in S, but it cannot be known who the signer is within S.

Since an opener is not defined in ring signatures, the false accusation problem becomes more serious. In fact, Komano, Ohta, Shimbo, and Kawamura [31] pointed out that "*the ring signature scheme allows the signer to shift the blame to entities (victims) because of its anonymity,*" and proposed a deniable ring signature, where a verifier and a user run interactive confirm/disavow protocols and the user can insist that "I am the actual signer" (confirm) or "I am NOT the actual signer" (disavow). Although Komano et al.'s scheme is secure in the random oracle model, later, a deniable ring signature scheme in the standard model was also proposed by Zeng et al. [44]. We note that their scheme is called conditionally anonymous ring signature instead of deniable ring signature.

Group signature was proposed by Chaum and Van Heyst [16]. In the research of security models, first, Bellare, Micciancio, and Warinschi (BMW) [4] defined the de facto standard security model of the group signature area. They showed that full anonymity and full traceability are sufficient for static group signatures. For extending from static group signatures to dynamic group signatures, Bellare, Shi, and Zhang (BSZ) [5] and Kiayias and Yung (KY) [29] independently developed security models. Later, Sakai et al. [41] showed that there is room for improving the BSZ model and considered a new security notion called opening soundness to prevent a signature hijacking attack.

Thus far, several efficient group signature schemes, such as Boneh-Boyen-Shacham [9] (and its CCA-anonymous version [20]), Camenisch-Lysyanskaya [12], Delerablée-Pointcheval [19], Furukawa-Imai [21], and Bichsel-Camenisch-Neven-Smart-Warinschi [6], have been proposed. Although these schemes are secure in the random oracle model, Boyen-Waters [10, 11] and Groth [25] proposed group signature schemes in the standard model. In particular, the Groth scheme applies efficient zero-knowledge proofs for bilinear groups, which are known as Groth-Sahai proofs [26]. In addition to these schemes, lattice-based constructions were proposed [24, 33, 32, 13].

As group signatures with additional functionality, group signatures with message-dependent opening [40, 38, 34] were considered in order to restrict the authority of the opener. In GS-MDO [40], the opener can open group signatures on specific signed messages, as decided by another authority called the admitter. In particular, an automated parking garage scenario was considered as an application of GS-MDO. In this case, a customer generates a group signature on the date he/she enters a garage, and if there is an accident (e.g., a person is murdered) in the garage, the opener opens all the signatures for the date of the accident to determine the customers present in the garage at the time of the accident. Again, if multiple customers enter the garage on the same date, then the false accusation problem occurs.

Abe et al. [1] considered non-snatching and undeniability in the traceable signature context, where no one (but the actual signer) can claim to be the signer of a signature, and no actual signer can deny being the signer of his signatures, respectively. Abe et al.'s traceable signature scheme, in addition to the opening and user tracing, allows the signer to claim non-ownership of a signature (as in the case of deniable ring signatures [31, 44]), while in the case of deniable group signatures, the opener can generate the proof for non-ownership of a signature.

Lyuu and Wu [37] considered group undeniable signatures where a verifier and a group manager run an interactive protocol that can prove the validity/invalidity of signatures without compromising anonymity.

Group encryption [2, 15, 28] is the encryption analog of the group signature, where an encryptor can make a ciphertext for a group member and can establish a proof that the receiver is a group member without any identification. Moreover, the opener can identify who the actual receiver is, as in the case of a group signature. Libert et al. proposed traceable group encryption [36], which captures the tracing capability as in the case of a traceable signature [27], where the opener can establish a user-specific trapdoor. Further, group members can prove that specific ciphertexts are intended for them or not by using the $\mathsf{CLAIM}/\mathsf{DISCLAIM}$ algorithms.

Thus far, several efficient group signature schemes have also been proposed in the random oracle model [9, 12, 19, 21, 6]. However, Canetti, Goldreich, and Halevi [14] pointed out that the random oracle methodology is problematic. Therefore, we consider only deniable group signature in the standard model.

1.4 Organization

The rest of this paper is organized as follows. We review some definitions of building blocks of our and the modified Groth schemes and the decisional linear assumption in Section 2. In Section 3, the concept of deniable group signatures and the security definitions are introduced. We propose a deniable group signature scheme in Section 4 and give security proofs.

2 Preliminaries

In this section, we provide definitions of some cryptographic assumptions and building blocks of our and the modified Groth schemes.

Bilinear Map. Bilinear groups are groups G and G_T with prime order p that have an efficiently computable bilinear map $e : G \times G \to G_T$. Let $\mathcal{G}(1^k)$ be a probabilistic polynomial time algorithm which outputs a group parameter $gk = (p, G, G_T, e, g)$ where k is a security parameter, p is the order of G and G_T, g is a generator of G, and e is a non-degenerate bilinear map $e : G \times G \to G_T$, i.e. $\forall a, b \in \mathbb{Z}, e(g^a, g^b) = e(g, g)^{ab}$ and $e(g, g) \neq 1$.

The q-strong Diffie-Hellman (q-SDH) assumption. The strong Deffie-Hellman assumption was introduced by Boneh and Boyen [8]. The q-SDH assumption holds for $\mathcal{G}$, when it is hard to find a pair $(m, g^{\frac{1}{1+x}}) \in \mathbb{Z}_p \times G$ when given $g, g^x, g^{x^2}, \ldots, g^{x^{q(k)}}$ as input.

The q-U Assumption. The q-U assumption is implied by a stronger assumption from Zhou and Lin [45] that is similar in nature. The q-U assumption holds for $\mathcal{G}$ if for any polynomial time algorithm $\mathcal{A}$ we have

$$\begin{aligned}\Pr[&(p, G, G_T, e, g) \leftarrow \mathcal{G}(1^k);\ x_1, r_1, \ldots, x_{q(k)}, r_{q(k)} \leftarrow \mathbb{Z}_q;\\ &f, h, z \leftarrow G;\ T = e(f, x);\ a_i = f^{r_i};\ b_i = h^{r_i} g^{x_i r_i} z;\\ &(V, A, B, m, S) \leftarrow \mathcal{A}(p, G, G_T, e, g, f, h, T, x_1, a_1, b_1, \ldots, x_{q(k)}, a_{q(k)}, b_{q(k)}) :\\ &V \notin \{g^{x_1}, \ldots, g^{x_{q(k)}}\} \ \wedge\ e(A, hV)e(f, B) = T \ \wedge\ e(S, Vg^m) = e(g, g)] \approx 0.\end{aligned}$$

The Decisional Linear Assumption (DLIN assumption). The decisional linear assumption was introduced [9]. The decisional linear assumption holds for $\mathcal{G}$, when it is hard to distinguish for randomly chosen group elements and exponents (f, g, h, f^r, g^s, h^t) whether $t = r + s$ or t is random.

Universal One-way Hash Function. A function generator HashGen(1^k) takes as input a security parameter k and outputs a function $\mathcal{H}$. The family of functions is said to be universal one-way when $\Pr[(x, s) \leftarrow \mathcal{A}(1^k); \mathcal{H} \leftarrow \text{HashGen}(1^k); x' \leftarrow \mathcal{A}(\mathcal{H}, s) : \mathcal{H}(x) = \mathcal{H}(x') \wedge x \neq x']$ is negligible for any polynomial time algorithm $\mathcal{A}$.

Strong One-time Signature. A signature scheme consists of three algorithms $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Ver})$, which satisfy the following correctness condition: For any security parameter $k \in \mathbb{N}$, any message $m \in \{0, 1\}^*$, the condition $\mathsf{Ver}_{vk}(m, \mathsf{Sign}_{sk}(m)) = 1$ holds, where vk and sk are output from KeyGen as $(vk, sk) \leftarrow \mathsf{KeyGen}$. In this paper, we use a one-time signature scheme which is secure against an adversary who has access to a single chosen message attack. The one-time signature is said to be strong, if the adversary cannot even create a different signature on the chosen message he already got signed. See [23] for a formal definition.

Non-interactive Proof. For a relation $R \in \{0, 1\}^* \times \{0, 1\}^*$ defining $L = \{x \mid (x, w) \in R \text{ for some } w\}$, a non-interactive proof system consists of three algorithms $(\mathsf{K}, \mathsf{P}, \mathsf{V})$ which satisfy the following correctness and soundness.

- Correctness: For any security parameter $k \in \mathbb{N}$, any common reference string $crs \leftarrow \mathsf{K}(1^k)$, and any pair $(x, w) \in R$, it holds $\mathsf{V}(crs, x, \mathsf{P}(crs, x, w)) = 1$.
- Soundness: For any security parameter $k \in \mathbb{N}$, any probabilistic polynomial time algorithm $\mathcal{A}$, the probability $\Pr[crs \leftarrow \mathsf{K}(1^k); (x, \pi) \leftarrow \mathcal{A}(crs) : \mathsf{V}(crs, x, \pi) = 1 \wedge x \notin L]$ is negligible.

Groth and Sahai introduced a framework for very efficient non-interactive proof for the satisfiability of relations in bilinear groups, including pairing product equations [26]. The proof system consists of algorithms $(\mathsf{K}_{\mathrm{NI}}, \mathsf{P}, \mathsf{V}, \mathsf{X})$. The algorithm $\mathsf{K}_{\mathrm{NI}}(gk)$ takes a group parameter gk as input and outputs (crs, xk) where crs is a common reference string and xk is an extraction key which can extract a witness from a proof. The algorithm $\mathsf{P}(crs, x, w)$ takes crs, an equation description x, and its witness w as input and outputs a proof π. This proof can be verified by running $\mathsf{V}(crs, x, \pi)$. The algorithm $\mathsf{X}_{xk}(crs, x, \pi)$ extracts a witness w from the proof π.

There are two types of the Groth-Sahai proof systems, $(\mathsf{K}_{\mathrm{NI}}, \mathsf{P}_{\mathrm{NIWI}}, \mathsf{V}_{\mathrm{NIWI}}, \mathsf{X}_{\mathrm{NIWI}})$ provides witness-indistinguishability and $(\mathsf{K}_{\mathrm{NI}}, \mathsf{P}_{\mathrm{NIZK}}, \mathsf{V}_{\mathrm{NIZK}}, \mathsf{X}_{\mathrm{NIZK}})$ provides zero-knowledge. The two types of proof can share a single common reference string. (Thus, multiple systems can use a common K_{NI}.) There exists a simulator that outputs a simulated common reference string crs and a trapdoor key tk. These simulated common reference strings are computationally indistinguishable from the common reference strings produced by K under the DLIN assumption. We say a proof system is perfect witness-indistinguishable, if, on a simulated common reference string, the proof π does not reveal anything about which witness was used by the prover when creating the proof. We say a proof system is perfect zero-knowledge, if there exists a simulator that produces a simulated proof and the simulated proof is perfectly indistinguishable from the proof which is produced by using a witness and a simulated common reference string.

In the Groth-Sahai proof system, to prove that committed variables satisfy a set of relations, the prover computes one commitment per variable and one proof element per relation. The non-interactive zero-knowledge (NIZK) proofs are available for pairing product equations, which are relations of the type $\prod_{i=1}^{n} e(\mathcal{A}_i, \mathcal{X}_i) \cdot \prod_{i=1}^{n} \prod_{j=1}^{n} e(\mathcal{X}_i, \mathcal{X}_j)^{a_{ij}} = t_T$ with $t_T = 1$ for variables $\mathcal{X}_1, \ldots, \mathcal{X}_n \in G$ and constants $\mathcal{A}_1, \ldots, \mathcal{A}_n \in G, a_{ij}$, for $i, j \in \{1, \ldots n\}$. Even if $t_T \neq 1$, still we can construct NIZK proofs if t_T can be decomposed to known base group elements $\tilde{g}, \hat{g} \in G$ such that $t_T = e(\tilde{g}, \hat{g})$. NIZK proofs also exist for multi-scalar multiplication equations, which are of the form $\prod_{i=1}^{m} \mathcal{A}_i^{y_i} \cdot \prod_{j=1}^{n} \mathcal{X}_j^{b_j} \cdot \prod_{i=1}^{m} \prod_{j=1}^{n} \mathcal{X}_j^{y_i \gamma_{ij}} = T$ for variables $\mathcal{X}_1, \ldots, \mathcal{X}_n \in G, y_1, \ldots, y_m \in \mathbb{Z}_p$ and constants $T, \mathcal{A}_1, \ldots, \mathcal{A}_m \in G, b_1, \ldots, b_m \in \mathbb{Z}_p$, and $\gamma_{ij} \in G$, for $i \in \{1, \ldots m\}, j \in \{1, \ldots n\}$. Here, we note that though the Groth-Sahai NIZK proof is efficient, it has a limitation of language. Especially, non-equal statement (e.g., $a \neq b$) is not covered the languages which can be proved by the Groth-Sahai NIZK proof. Also, as mentioned above, if a witness is a target group element, it needs to be decomposed into base group elements to apply the Groth-Sahai proof. However, it is hard because of the the pairing inversion problem [22].

Kiltz's Tag-based Encryption. We will use Kiltz's construction of tag-based encryption [30], which is explained below. Let $gk = (p, G, G_T, e, g)$ be a group description. The key generation algorithm $\mathsf{G}(1^k)$ chooses random integers $\zeta, \eta \leftarrow \mathbb{Z}_p$ and random elements $K, L \leftarrow G$, and sets the public key $pk = (F, H, K, L)$ where $F = g^{\zeta}$ and $H = g^{\eta}$ and the decryption key $dk = (\zeta, \eta)$. The encryption algorithm $\mathsf{E}_{pk}(t, m)$ outputs $y = (y_1, y_2, y_3, y_4, y_5) = (F^r, H^s, mg^{r+s}, (g^t K)^r, (g^t L)^s)$ where m is a plaintext, t is a tag, and r, s are randomness. The validity of the ciphertext is publicly verifiable by checking the two equations $e(F, y_4) = e(y_1, g^t K)$ and $e(H, y_5) = e(y_2, g^t L)$. Here, let $\mathsf{ValidCiphertext}_{pk}(t, y)$ be an algorithm verifying the validity of a ciphertext. The decryption algorithm $\mathsf{D}_{dk}(t, y)$ outputs $m = y_3/(y_1^{-\zeta} y_2^{-\eta})$ if the above two equations hold, otherwise outputs $\perp$. This tag-based encryption is secure against selective-tag weak chosen ciphertext attack under the DLIN assumption [30]. In the modified Groth scheme and our scheme, we use the same F, H as in the common reference string of non-interactive proofs.

Groth's Certified Signature. Groth constructed a certified signature scheme where the authority issues a certificate of a verification key [25]. The certified signature scheme is existentially unforgeable against weak chosen message (EUF-wCMA) attack under the q-SDH assumption. For certification, there are two steps. In the first step, the protocol [25] generates a random $v = g^x$ such that the issuer learns v but only the user learns x. In the second step, a variant of the signature scheme of Zhou and Lin [45] is used to certify a verification key v. To set up the certified signature scheme, the authority picks random group elements $f, h, z \in G$, and sets the authority key (f, h, T) and the secret certification key z where $T = e(f, z)$. When the authority certifies a Boneh-Boyen key v, he picks random $r \in \mathbb{Z}_p$ and sets $(a, b) = (f^{-r}, (hv)^r z)$ as the certificate. The certificate is verified by checking $e(a, hv)e(f, b) = T$.

3 Deniable Group Signatures

In this section, we give the definition of deniable group signature which is a natural extension of the Bellare-Shi-Zhang (BSZ) model [5]. More precisely, we base on the Sakai et al. model [41], which modifies the BSZ model by introducing a new security notion called opening soundness.

3.1 Modification to the BSZ model

Our definition follows the BSZ [5] and Sakai et al. model [41]. In particular, our definition is obtained by a slight modification to the BSZ and Sakai et al. definitions. This modification is done for capturing the ability of the opener to establish denial proofs. For the ease of understanding (particularly for readers familiar with the standard group signature), we first highlight the differences between our definition and the BSZ model and then, provide the formal comprehensive definitions.

In deniable group signatures, we require that for (m, Σ) and a user j, the opener can establish a proof that the open result is not j. Hence, we add *new algorithms*, namely DOpen and DJudge, to the Sakai et al. model. The opener produces this denial proof by using the DOpen algorithm and validity of the proof can be judged by the DJudge algorithm.

According to the new algorithms, we also change the security definition. Since we allow the opener to produce the new type of opening, namely denial opening, we need to ensure that such openings do not compromise the anonymity of group signatures. In the deniable group signature, denial proofs strengthen an adversary's ability. Even in this situation, anonymity must be guaranteed. Thus, we allow the adversary to obtain denial proofs for any group signatures of his/her choice, as the BSZ definition allows him/her to obtain opening proofs of any signature. Furthermore, it is natural to expect that a denial proof for a signature Σ with respect to a user j does not leak any information beyond the fact that Σ is not generated by j. To capture this intuition, we allow the adversary in the anonymity game to obtain denial proofs *for the challenge* and it does not break the game trivial.

3.2 Formal Definition

We give the formal definition of deniable group signatures. First, we define the model of deniable group signature, but algorithms, except DOpen and DJudge (which are underlined in the following definition), are exactly the same as those of the Sakai et al. model.

Definition 3.1 (Deniable Group Signature). *A deniable group signature scheme $\mathcal{D}\text{-}\mathcal{GS}$ consists of the algorithms* ($\mathsf{GKg}, \mathsf{UKg}, \mathsf{Join}/\mathsf{Iss}, \mathsf{GSig}, \mathsf{GVf}, \mathsf{Open}, \mathsf{Judge}, \underline{\mathsf{DOpen}}, \underline{\mathsf{DJudge}}$)*:*

GKg**:** *The group key generation algorithm takes as input a security parameter 1^k ($k \in \mathbb{N}$), and returns a group public key gpk, an issuer key ik, and an opening key ok.*

UKg**:** *The user key generation algorithm, which is run by a user i, takes as input 1^k and gpk, and returns a public and private key pair (upk_i, usk_i). It is assumed that all parties can obtain an authentic copy of the public keys of all users.*

$\mathsf{Join}/\mathsf{Iss}$**:** *The pair of (interactive) algorithms are run by a user and the issuer, and takes as input gpk, upk_i, and usk_i from user i and gpk, upk_i, and ik from the issuer, respectively. If successful, the issuer stores the registration information of user i in* $\mathsf{reg}[i]$ *and the user obtains the corresponding secret signing key gsk_i. We denote* $\mathsf{reg} = \{\mathsf{reg}[i]\}_i$.

GSig**:** *The group signing algorithm takes as input gpk, gsk_i and a message $m \in \mathcal{M}_{GSig}$, and returns a group signature Σ.*

GVf: *The verification algorithm takes as inputs* gpk, Σ, *and* m, *and returns either* 1 *(indicating that* Σ *is a valid group signature), or* 0.

Open: *The opening algorithm takes as input* gpk, ok, m, Σ, *and* reg, *and returns* (i, τ_O), *where* i *is a user identity, and* τ_O *is a proof that user* i *computed* Σ.

Judge: *The judgement algorithm takes as inputs* gpk, i, upk_i, m, Σ, *and* τ_O, *and returns* 1 *if* Σ *is produced by user* i, *and* 0 *otherwise.*

DOpen: *The denial opening algorithm takes as input* gpk, j, ok, m, Σ, *and* reg, *and returns* $\tau_{D(j)}$, *where* j *is a user identity, and* $\tau_{D(j)}$ *is a proof that user* j *did not compute* Σ.

DJudge: *The denial judgement algorithm takes as inputs* gpk, j, upk_j, m, Σ, *and* $\tau_{D(j)}$, *and returns* 1 *if* Σ *is not produced by user* j, *and* 0 *otherwise.*

The model in [5] introduces four requirements for a group signature, namely, correctness, anonymity, non-frameability, and traceability. Furthermore, opening soundness is introduced by [41]. Here, we provide the definitions of correctness, anonymity, non-frameability, traceability, and opening soundness for a deniable group signature. The security model is extended from the dynamic group signature defined by Sakai et al. [41] and therefore, is almost the same except for the anonymity.

We first define several oracles used in security games. We newly introduce the DOpen oracle (which is underlined in the following definition) in addition to Sakai et al.'s definition.

AddU: This add user oracle runs UKg and Join/Iss protocol to add an honest user to the group. The oracle returns upk_i and adds i to HU.

CrptU: This corrupt user oracle allows $\mathcal{A}$ to add corrupt users. On input an identity i and upk, this oracle sets $upk_i \leftarrow upk$ and adds i to CU.

SndToU: This send to user oracle takes as input a user identity i, at first sets up a user public and private key pair $(upk_i, usk_i) \leftarrow \mathsf{UKg}(1^k, gpk)$ and adds i to HU. Then the oracle interacts with $\mathcal{A}$ who corrupts the issuer by running $\mathsf{Join}(gpk, upk_i, usk_i)$ and the respond of the user is returned to $\mathcal{A}$.

SndToI: This send to issuer oracle takes as input a user identity i, and interacts with $\mathcal{A}$ who corrupts the user i by running $\mathsf{Iss}(gpk, upk_i, ik)$. The user i needs to be in the set CU.

Ch: This challenge oracle takes as input a bit b, two identities i_0, i_1, and m, and returns $\Sigma^* \leftarrow \mathsf{GSig}(gpk, gsk_{i_b}, m)$ if both $i_0 \in \mathsf{HU}$ and $i_1 \in \mathsf{HU}$. If not, the oracle returns $\perp$. The oracle stores (m, Σ^*) in GSet, and stores i_0 and i_1 in ISet.

Open: This opening oracle takes as input m and Σ, and returns $(i, \tau_O) \leftarrow \mathsf{Open}(gpk, ok, m, \Sigma, \mathsf{reg})$ if $(m, \Sigma) \notin \mathsf{GSet}$ and $\perp$ otherwise.

DOpen: This deniable opening oracle takes as input a user identity j, m and Σ, and returns $\tau_{D(j)} \leftarrow \mathsf{DOpen}(gpk, j, ok, m, \Sigma, \mathsf{reg})$ if $(m, \Sigma) \notin \mathsf{GSet} \vee j \notin \mathsf{ISet}$ and $\perp$ otherwise.

USK: This user secret keys oracle takes as input $i \in \mathsf{HU}$, and returns the secret keys usk_i and gsk_i.

GSig: This signing oracle takes as input i and a message m, and returns $\Sigma \leftarrow \mathsf{GSig}(gpk, gsk_i, m)$ if $i \in \mathsf{HU}$. Otherwise, the oracle returns $\perp$.

RReg: This read registration table oracle takes as input i, and returns $\mathsf{reg}[i]$.

WReg: This write registration table oracle takes as input i and a value ρ, and modifies the contents of reg by setting $\mathsf{reg}[i] \leftarrow \rho$.

The correctness of group signatures [41] is required for ensuring that any honestly generated group signature is valid (i.e., it is accepted by the GVf algorithm), and the Open algorithm correctly identifies its actual signer and the proof generated by the Open algorithm is accepted by the Judge algorithm. For the correctness of deniable group signatures, we also require that a proof of an arbitrary user j, who is not the actual signer i, generated by the DOpen algorithm is accepted by the DJudge algorithm.

Definition 3.2 (Correctness)**.** *For any probabilistic polynomial time (PPT) adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$, and the security parameter $k \in \mathbb{N}$, we define the experiment $\mathrm{Exp}^{corr}_{\mathsf{D\text{-}GS},\mathcal{A}_1,\mathcal{A}_2}(k)$ as follows.*

```
$\mathrm{Exp}^{corr}_{\mathsf{D\text{-}GS},\mathcal{A}_1,\mathcal{A}_2}(k)$ :
    $(gpk, ik, ok) \leftarrow \mathsf{GKg}(1^k)$;  $\mathsf{HU} \leftarrow \emptyset$;  $(i, m, s) \leftarrow \mathcal{A}_1^{\mathsf{AddU}(\cdot),\mathsf{RReg}(\cdot)}(gpk)$
    $\Sigma \leftarrow \mathsf{GSig}(gpk, gsk_i, m)$;  $(i', \tau_O) \leftarrow \mathsf{Open}(gpk, ok, m, \Sigma, \mathsf{reg})$
    $j \leftarrow \mathcal{A}_2^{\mathsf{AddU}(\cdot),\mathsf{RReg}(\cdot)}(\Sigma, i', \tau_O, s)$ where $j \neq i \ \wedge \ j \in \mathsf{HU}$
    $\tau_{D(j)} \leftarrow \mathsf{DOpen}(gpk, j, upk_j, ok, m, \Sigma, \mathsf{reg})$
     Output 1 if the following holds :
        $\mathsf{GVf}(gpk, m, \Sigma) = 0 \ \vee \ i \neq i' \ \vee \ \mathsf{Judge}(gpk, i, upk_i, m, \Sigma, \tau_O) = 0$
                                          $\vee \ \mathsf{DJudge}(gpk, j, upk_j, m, \Sigma, \tau_{D(j)}) = 0$
     Otherwise return 0
```

A deniable group signature scheme is said to be satisfying correctness if the advantage $\mathrm{Adv}^{corr}_{\mathsf{D\text{-}GS},\mathcal{A}_1,\mathcal{A}_2}(k) := \Pr[\mathrm{Exp}^{corr}_{\mathsf{D\text{-}GS},\mathcal{A}_1,\mathcal{A}_2}(k) = 1]$ is negligible for any PPT adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$.

The anonymity of group signatures [41] is required so that an adversary, who can corrupt the issuer and malicious users cannot extract any user information from group signatures of honest users in the case when the adversary is able to access the Open oracle. In the anonymity of deniable group signatures, the adversary can also access the DOpen oracle. As mentioned above, the adversary can even query the challenge signature to the DOpen oracle except for querying the challenge users i_0 and i_1. Therefore, anonymity is guaranteed even if denial proofs for all users except i_0 and i_1 are provides for the challenge group signature.[1]

Definition 3.3 (Anonymity)**.** *For any PPT adversary $\mathcal{A}$ and security parameter $k \in \mathbb{N}$, we define the experiment $\mathrm{Exp}^{anon}_{\mathsf{D\text{-}GS},\mathcal{A}}(k)$ as follows.*

```
$\mathrm{Exp}^{anon}_{\mathsf{D\text{-}GS},\mathcal{A}}(k)$ :
    $b \leftarrow \{0, 1\}$
    $(gpk, ik, ok) \leftarrow \mathsf{GKg}(1^k)$;  $\mathsf{CU} \leftarrow \emptyset$;  $\mathsf{HU} \leftarrow \emptyset$;  $\mathsf{GSet} \leftarrow \emptyset$;  $\mathsf{ISet} \leftarrow \emptyset$
    $b' \leftarrow \mathcal{A}^{\mathsf{CrptU}(\cdot,\cdot),\mathsf{SndToU}(\cdot),\mathsf{WReg}(\cdot,\cdot),\mathsf{USK}(\cdot),\mathsf{Open}(\cdot,\cdot),\mathsf{DOpen}(\cdot,\cdot,\cdot),\mathsf{Ch}(b,\cdot,\cdot,\cdot)}(gpk, ik)$
    Return $b'$
```

A deniable group signature scheme is said to be satisfying anonymity if the advantage $\mathrm{Adv}^{anon}_{\mathsf{D\text{-}GS},\mathcal{A}} := |\Pr[\mathrm{Exp}^{anon}_{\mathsf{D\text{-}GS},\mathcal{A}}(k) = b] - \frac{1}{2}|$ is negligible for any PPT adversary $\mathcal{A}$.

The non-frameability of group signatures [41] is required so that an adversary who can corrupt the issuer, the opener, and malicious users except one honest user cannot produce a valid group signature of the honest user and its opening proof, which is accepted by the Judge algorithm. In the non-frameability of deniable group signatures, it is required that the adversary should not be able to forge a valid group signature whose denial opening proof for the honest user j is not accepted by the DJudge algorithm.

[1] As a remark, we exclude the case that an adversary requests a denial proof of either i_0 or i_1 for the challenge signature, since this trivially breaks the anonymity.

Definition 3.4 (Non-Frameability). *For any adversary $\mathcal{A}$ and security parameter $k \in \mathbb{N}$, we define the experiment* $\mathrm{Exp}^{nf}_{\mathsf{D\text{-}GS},\mathcal{A}}(k)$ *as follows.*

```
$\mathrm{Exp}^{nf}_{\mathsf{D\text{-}GS},\mathcal{A}}(k)$ :
    $(gpk, ik, ok) \leftarrow \mathsf{GKg}(1^k);\ \mathsf{CU} \leftarrow \emptyset;\ \mathsf{HU} \leftarrow \emptyset$
    $(m, \Sigma, j, \tau_O) \leftarrow \mathcal{A}^{\mathsf{SndToU}(\cdot),\mathsf{GSig}(\cdot,\cdot),\mathsf{WReg}(\cdot,\cdot),\mathsf{USK}(\cdot),\mathsf{CrptU}(\cdot,\cdot)}(gpk, ik, ok)$
    $\tau_{D(j)} \leftarrow \mathsf{DOpen}(gpk, j, upk_j, ok, m, \Sigma, \mathsf{reg})$
    $\mathit{Return}\ 1\ \mathit{if\ all\ of\ the\ following\ hold}:$
        $j \in \mathsf{HU}$
        $\mathsf{GVf}(gpk, m, \Sigma) = 1$
        $\mathsf{Judge}(gpk, j, upk_j, m, \Sigma, \tau_O) = 1\ \vee\ \mathsf{DJudge}(gpk, j, upk_j, m, \Sigma, \tau_{D(j)}) = 0$
        $\mathcal{A}\ \mathit{did\ not\ query}\ \mathsf{USK}(j)\ \mathit{and}\ \mathsf{GSig}(j, m)$
    $\mathit{Otherwise\ return}\ 0$
```

A deniable group signature scheme is said to be satisfying non-frameability if the advantage $\mathrm{Adv}^{nf}_{\mathsf{D\text{-}GS},\mathcal{A}}(k) := \Pr[\mathrm{Exp}^{nf}_{\mathsf{D\text{-}GS},\mathcal{A}}(k) = 1]$ *is negligible for any PPT adversary $\mathcal{A}$.*

The traceability of group signatures [41] is required for ensuring that an adversary, who can corrupt the opener, and malicious users cannot produce a valid group signature whose opening result is not valid (i.e., an invalid identity) or opening proof is not accepted by the Judge algorithm. In the traceability of deniable group signatures, it is also required that the adversary should not be able to produce a valid group signature whose opening proof is accepted by the Judge algorithm, but for the same user, the denial opening proof is accepted by the DJudge algorithm.

Definition 3.5 (Traceability). *For any PPT adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$, and security parameter $k \in \mathbb{N}$, we define the experiment* $\mathrm{Exp}^{trace}_{\mathsf{D\text{-}GS},\mathcal{A}_1,\mathcal{A}_2}(k)$ *as follows.*

```
$\mathrm{Exp}^{trace}_{\mathsf{D\text{-}GS},\mathcal{A}_1,\mathcal{A}_2}(k)$ :
    $(gpk, ik, ok) \leftarrow \mathsf{GKg}(1^k);\ \mathsf{CU} \leftarrow \emptyset;\ \mathsf{HU} \leftarrow \emptyset$
    $(m, \Sigma, s) \leftarrow \mathcal{A}_1^{\mathsf{CrptU}(\cdot,\cdot),\mathsf{SndToI}(\cdot),\mathsf{AddU}(\cdot),\mathsf{RReg}(\cdot),\mathsf{USK}(\cdot)}(gpk, ok)$
    $(i, \tau_O) \leftarrow \mathsf{Open}(gpk, ok, m, \Sigma, \mathsf{reg})$
    $\tau_{D(i)} \leftarrow \mathcal{A}_2^{\mathsf{CrptU}(\cdot,\cdot),\mathsf{SndToI}(\cdot),\mathsf{AddU}(\cdot),\mathsf{RReg}(\cdot),\mathsf{USK}(\cdot)}(i, \tau_O, s)$
    $\mathit{Return}\ 1\ \mathit{if\ the\ following\ two\ conditions\ hold}:$
        $\mathsf{GVf}(gpk, m, \Sigma) = 1$
        $\{i = 0\ \vee\ \mathsf{Judge}(gpk, i, upk_i, m, \Sigma, \tau_O) = 0\}$
                $\vee\ \{i \neq 0\ \wedge\ \mathsf{Judge}(gpk, i, upk_i, m, \Sigma, \tau_O) = 1$
                                $\wedge\ \mathsf{DJudge}(gpk, i, upk_i, m, \Sigma, \tau_{D(i)}) = 1\}$
    $\mathit{Otherwise\ return}\ 0$
```

A deniable group signature scheme is said to be satisfying traceability if the advantage $\mathrm{Adv}^{trace}_{\mathsf{D\text{-}GS},\mathcal{A}_1,\mathcal{A}_2}(k) := \Pr[\mathrm{Exp}^{trace}_{\mathsf{D\text{-}GS},\mathcal{A}_1,\mathcal{A}_2}(k) = 1]$ *is negligible for any PPT adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$.*

The opening soundness of group signatures [41] is required so that an adversary who can corrupt the issuer, the opener, and malicious users cannot produce a valid group signature and its opening proofs for i and j, which are both accepted by the Judge algorithm. In the opening soundness of deniable group signatures, it is also required that the adversary should not be able to produce a valid group signature and its denial opening proof for the actual signer which is accepted by the DJudge algorithm.

Definition 3.6 (Opening Soundness)**.** *For any PPT adversary $\mathcal{A}$ and security parameter $k \in \mathbb{N}$, we define the experiment* $\mathrm{Exp}^{snd}_{\mathsf{D\text{-}GS},\mathcal{A}}(k)$ *as follows.*

```
$\mathrm{Exp}^{snd}_{\mathsf{D\text{-}GS},\mathcal{A}}(k)$ :
    $(gpk, ik, ok) \leftarrow \mathsf{GKg}(1^k)$; $\mathsf{CU} \leftarrow \emptyset$; $\mathsf{HU} \leftarrow \emptyset$
    $(i, j, \tau_{O(i)}, \tau_{O(j)}, \tau_{D(i)}, m, \Sigma) \leftarrow \mathcal{A}^{\mathsf{CrptU}(\cdot),\mathsf{WReg}(\cdot,\cdot)}(gpk, ok, ik)$
    Return 1 if all of the following hold :
        $\mathsf{GVf}(gpk, m, \Sigma) = 1$
        $\mathsf{Judge}(gpk, i, upk_i, m, \Sigma, \tau_{O(i)}) = 1$
        $\{i \neq j \wedge \mathsf{Judge}(gpk, j, upk_j, m, \Sigma, \tau_{O(j)}) = 1\}$
                                    $\vee\ \mathsf{DJudge}(gpk, i, upk_i, m, \Sigma, \tau_{D(i)}) = 1$
    Otherwise return 0
```

A deniable group signature scheme is said to be satisfying opening soundness if the advantage $\mathrm{Adv}^{os}_{\mathsf{D\text{-}GS},\mathcal{A}}(k) := \Pr[\mathrm{Exp}^{os}_{\mathsf{D\text{-}GS},\mathcal{A}}(k) = 1]$ *is negligible for any PPT adversary $\mathcal{A}$.*

4 The Proposed Deniable Group Signature Scheme

In this section, we review the technique for proving a non-equal statement and explain that a deniable group signature can be constructed by applying this technique to the generic construction of a (standard) group signature presented by Bellare, Shi, and Zhang (BSZ) [5]. Then, we explain the difficulty of instantiating an efficient scheme even when a generic construction of a deniable group signature is given, and present our deniable group signature scheme, which is fairly efficient. Lastly, we discuss the size of the denial proofs of the proposed scheme.

4.1 Generic Construction and Its Limitation

Here, we give a generic construction of deniable group signature which is an extension of the BSZ construction [5]. In the BSZ construction, each user i has a key pair (vk_i, sk_i) of a signature scheme. The issuer also has a key pair (vk_s, sk_s) of a signature scheme and the opener has a key pair (pk_e, sk_e) of a public key encryption scheme. For issuing a signing key to a user i, the issuer signs the message (i, vk_i) using his key sk_s and sends the signature $cert_i$ to the user i. A signer i can produce a signature s for a message m under vk_i. To make this verifiable without losing anonymity, the user makes an encryption C of $(i, vk_i, cert_i, s)$ using pk_e and also makes an NIZK proof π which proves that $cert_i$ is a valid certificate on (i, vk_i), i.e., $\mathsf{Vrfy}_{vk_s}((i, vk_i), cert_i) = 1$ and s is a valid signature on m. The opener can identity i by decrypting C using sk_e. Then, the opener produces an NIZK proof τ which proves that C is decrypted to $(i, vk_i, cert_i, s)$ under sk_e.

We can add deniability to the BSZ construction as follows. The opener produces an NIZK proof τ' where C is decrypted to $(i, vk_i, cert_i, s)$ under sk_e and $cert_i$ is NOT a valid certificate on (j, vk_j), i.e., $\mathsf{Vrfy}_{vk_s}((j, vk_j), cert_i) \neq 1$. Though this denial proof can be constructed by using general NIZK proofs [7], it is quite inefficient. The next attempt is to add deniability to an efficient group signature scheme (e.g., the modified Groth scheme [41]) by using an efficient NIZK proof (e.g., he Groth-Sahai proofs [26]). Unfortunately, this type of language (i.e., non-equal statement) is not vary compatible with the Groth-Sahai proofs, especially the Groth-Sahai NIZK proof.

4.2 The Proposed Scheme

In this subsection, we describe the proposed deniable group signature scheme based on the modified Groth scheme [41]. Before presenting the details of the proposed scheme, we note that the scheme diverges slightly from the model presented in Section 3.2, which follows the BSZ model [5]. The reason is that the proposed scheme is an extension of the modified Groth scheme. In particular, in [5], each user is assumed to independently generate a public and private key pair (upk_i, usk_i) and then, obtains a signing key gsk_i by running the interactive $\mathsf{Join}/\mathsf{Iss}$ algorithm with the issuer. In the modified Groth scheme and our scheme, on the other hand, a public and private key pair is jointly generated in the Join algorithm. This intuitively corresponds to a scheme in which the user key generation algorithm UKg is merged with the Join algorithm. The public key is stored in $\mathsf{reg}[i]$ by the issuer, and the corresponding private key is a part of the group signing key gsk_i. That is, reg is available for all parties in the modified Groth scheme and the proposed scheme.

To model the security of this type of scheme, a few minor changes are required to meet the security requirements defined in Section 3.2. Concretely, we no longer consider the WReg oracle in the all the security definitions and the CrptU oracle in the traceability definition. Moreover, the oracles AddU and SndToU no longer run the UKg algorithm. The details are discussed in [41] (ePrint version, page 10, paragraph "The Groth Scheme").

Our Basic Approach for an Efficient Instantiation. We will now present the details of the proposed scheme. In the case of a deniable group signature, the opener needs to issue a denial proof, which proves that the user j is not the actual signer without revealing user i itself. Here, we review the technique for proving a non-equal statement $i \neq j$ introduced by, e.g., [42] as follows: The technique is that to prove $a \neq b$, the prover takes $\ell \in \mathbb{Z}_p$ randomly and sets $c := (a/b)^\ell$ and the verifier checks $c \neq 1$ and the knowledge of ℓ. We note that even though the technique for proving the non-equality mentioned above has been proposed, this technique can not be straightforwardly applied to the modified Groth scheme (See Remark for details).

We give our proposed scheme in Figure 1. The proposed scheme is an extension of the modified Groth scheme [41], which has opening soundness added to the Groth scheme [25]. This is because the Groth scheme, which is the first efficient group signature scheme in the standard model, is vulnerable to signature hijacking attacks, whereas the modified Groth scheme is secure against such attacks. The modified Groth scheme uses a universal one-way hash function $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_p$, the Groth-Sahai proof systems $(\mathsf{K}_{\mathrm{NI}}, \mathsf{P}_{\mathrm{NIWI}}, \mathsf{V}_{\mathrm{NIWI}}, \mathsf{X}_{\mathrm{NIWI}})$ and $(\mathsf{K}_{\mathrm{NI}}, \mathsf{P}_{\mathrm{NIZK}}, \mathsf{V}_{\mathrm{NIZK}}, \mathsf{X}_{\mathrm{NIZK}})$, and a strong one-time signature $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Ver})$ as building blocks. Note that in the modified Groth scheme, we use a common K_{NI} for both systems and $\mathsf{X}_{\mathrm{NIZK}}$ is not used.

First, we will explain the GSig and Open algorithms of the modified Groth scheme as follows: In the GSig algorithm, a signer constructs two Groth-Sahai proofs. The first proof π, constructed via $\mathsf{P}_{\mathrm{NIWI}}$ shows the knowledge of a signature σ, a verification key v, and a part b of a certificate (a, b) that satisfies $e(a, hv)e(f, b) = T \wedge e(\sigma, vg^{\mathcal{H}(vk_{\mathrm{sots}})}) = e(g, g)$. The first part a can be revealed in the group signature. The second proof ψ, constructed via $\mathsf{P}_{\mathrm{NIZK}}$ demonstrates that the plaintext of y is the same as the witness σ used in π. That is, for a commitment $c = (c_1, c_2, c_3) = (F^{r_c}U^t, H^{s_c}V^t, g^{r_c+s_c}W^t\sigma)$ contained in π, there exists (r, s, t) such that $(c_1y_1^{-1}, c_2y_2^{-1}, c_3y_3^{-1}) = (F^rU^t, H^sV^t, g^{r+s}W^t)$. In the Open algorithm, the opener reveals $\tau_F = y_1^{1/d_F} = g^r$ and $\tau_H = y_2^{1/d_H} = g^s$ as a part of an opening proof. If a third party, given τ_F and τ_H wants to check the correspondence between the ciphertext (y_1, y_2, y_3) and the plaintext σ, he/she checks whether $e(F, \tau_F) = e(y_1, g)$, $e(H, \tau_H) = e(y_2, g)$, $\sigma\tau_F\tau_H = y_3$, and $e(\sigma, v_ig^{\mathcal{H}(vk_{\mathrm{sots}})}) = e(g, g)$ hold or not.

We note that a simple modification, where the opener makes an NIZK proof for $e(\sigma, v_jg^{\mathcal{H}(vk_{\mathrm{sots}})}) \neq e(g, g)$, does not work because of the limitation of languages of the "zero-knowledge version" of the Groth-Sahai proof (See Remark for details). To break the barrier, all witnesses need to be base group elements. Therefore, to prove $i \neq j$, we define the inequation to be proved on the base group G such that $v_i \neq v_j$ where $v_i, v_j \in G$. That is, the opener takes random $\ell \leftarrow \mathbb{Z}_p$ and set $c = \tau_\ell \cdot (\tau'_\ell)^{-1}$ where $\tau_\ell = v_i^\ell$ and $\tau'_\ell = v_j^\ell$. The proof ϕ, constructed via $\mathsf{P}_{\mathrm{NIZK}}$, shows the knowledge of the opening proof $(i, (\sigma, \tau_F, \tau_H))$, v_i, τ_ℓ, and τ'_ℓ which satisfy

$$e(F, \tau_F) = e(y_1, g) \ \wedge \ e(H, \tau_H) = e(y_2, g) \ \wedge \ \sigma\tau_F\tau_H = y_3$$

$$\wedge \ e(\sigma, v_ig^{\mathcal{H}(vk_{\mathrm{sots}})}) = e(g, g) \ \wedge \ e(v_i, \tau'_\ell) = e(v_j, \tau_\ell) \ \wedge \ c = \tau_\ell \cdot (\tau'_\ell)^{-1}.$$

The first four equations demonstrate that i is the actual signer of Σ and the fifth equation demonstrates that the discrete logarithm of τ_ℓ and that of τ'_ℓ are the same. In the DJudge algorithm, one checks the NIZK proof and whether $c \neq 1$.

```
$\mathsf{GKg}(1^k)$:
$gk = (p, G, G_T, e, g) \leftarrow \mathcal{G}(1^k)$
$\mathcal{H} \leftarrow \mathrm{HashGen}(1^k)$
$(f, h, z) \leftarrow G$
$T = e(f, z)$
$(crs, xk) \leftarrow \mathsf{K}_{\mathrm{NI}}(gk)$
$(F, H, U, V, W, U', V', W') \leftarrow crs$
$K, L \leftarrow G$
$pk \leftarrow (F, H, K, L)$
Return $(gpk, ik, ok)$
    $\leftarrow ((gk, \mathcal{H}, f, h, T, crs, pk), z, xk)$
```

```
$\mathsf{Join/Iss(User}\ i{:}gpk;\ \mathsf{Issuer}{:}gpk, ik)$:
Run the coin-flipping protocol
  The user obtains $v_i = g^{x_i}$ and $x_i$
  and the issuer obtains $v_i$
  (Repeat until $v_i \neq \mathsf{reg}[j]$ for all $j$)
Issuer:
  $r \leftarrow \mathbb{Z}_p$
  $(a_i, b_i) \leftarrow (f^{-r}, (v_i h)^r z)$
  set $\mathsf{reg}[i] \leftarrow v_i$
  send $(a_i, b_i)$ to the user
User:
  If $e(a_i, hv_i)e(f, b_i) = T$
  set $gsk_i \leftarrow (x_i, a_i, b_i)$
```

```
$\mathsf{GSig}(gpk, gsk_i, m)$:
$(vk_{\mathrm{sots}}, sk_{\mathrm{sots}}) \leftarrow \mathsf{KeyGen}_{\mathrm{sots}}(1^k)$
    (Repeat until $\mathcal{H}(vk_{\mathrm{sots}}) \neq -x_i$)
$\rho \leftarrow \mathbb{Z}_p; a \leftarrow a_i f^{-\rho}; b \leftarrow b_i(hv_i)^{\rho}$
$\sigma \leftarrow g^{1/(x_i + \mathcal{H}(vk_{\mathrm{sots}}))}$
$\pi \leftarrow \mathsf{P}_{\mathrm{NIWI}}(crs, (gpk, a, \mathcal{H}(vk_{\mathrm{sots}})),$
                                        $(b, v_i, \sigma))$
$y \leftarrow \mathsf{E}_{pk}(\mathcal{H}(vk_{\mathrm{sots}}), \sigma)$
$\psi \leftarrow \mathsf{P}_{\mathrm{NIZK}}(crs, (gpk, y, \pi), (r, s, t))$
$\sigma_{\mathrm{sots}} \leftarrow \mathsf{Sign}_{sk_{\mathrm{sots}}}(vk_{\mathrm{sots}}, m, a, \pi, y, \psi)$
Return $\Sigma = (vk_{\mathrm{sots}}, a, \pi, y, \psi, \sigma_{\mathrm{sots}})$
```

```
$\mathsf{GVf}(gpk, \mathsf{reg}, m, \Sigma)$:
Return 1 if the following holds:
 $1 = \mathsf{Ver}_{vk_{\mathrm{sots}}}((vk_{\mathrm{sots}}, m, a, \pi, y, \psi), \sigma_{\mathrm{sots}})$
 $1 = \mathsf{V}_{\mathrm{NIWI}}(crs, (gpk, a, \mathcal{H}(vk_{\mathrm{sots}})), \pi)$
 $1 = \mathsf{V}_{\mathrm{NIZK}}(crs, (gpk, y, \pi), \psi)$
 $1 = \mathsf{ValidCiphertext}_{pk}(\mathcal{H}(vk_{\mathrm{sots}}), y)$
 $\mathsf{reg}[i] \neq \mathsf{reg}[j]$ for all $i \neq j$
else return 0
```

```
$\mathsf{Open}(gpk, ok, \mathsf{reg}, m, \Sigma)$:
If $\mathsf{GVf}(gpk, \mathsf{reg}, m, \Sigma) = 0$, return $(0, \perp)$
$(b, v, \sigma) \leftarrow \mathsf{X}_{xk}(crs, (gpk, a, \mathcal{H}(vk_{\mathrm{sots}})), \pi)$
$(d_F, d_H) \leftarrow xk$
$(y_1, y_2, \ldots, y_5) \leftarrow y$
$\tau_F = y_1^{1/d_F}, \tau_H = y_2^{1/d_H}$
Return $(i, (\sigma, \tau_F, \tau_H))$
   if there is $i$ so $v = \mathsf{reg}[i]$,
else $(0, \perp)$
```

```
$\mathsf{Judge}(gpk, i, \mathsf{reg}, m, \Sigma, (\sigma, \tau_F, \tau_H))$:
$v_i \leftarrow \mathsf{reg}[i]$
Return 1 if the following hold:
 $\mathsf{GVf}(gpk, \mathsf{reg}, m, \Sigma) = 1$
 $i \neq 0, e(\sigma, v_i g^{\mathcal{H}(vk_{\mathrm{sots}})}) = e(g, g)$
 $e(F, \tau_F) = e(y_1, g), e(H, \tau_H) = e(y_2, g)$
 $\sigma \tau_F \tau_H = y_3$
else return 0
```

```
$\mathsf{DOpen}(gpk, j, ok, \mathsf{reg}, m, \Sigma)$:
$(i, (\sigma, \tau_F, \tau_H)) \leftarrow \mathsf{Open}(gpk, ok, \mathsf{reg}, m, \Sigma)$
If $(i, (\sigma, \tau_F, \tau_H)) = (0, \perp)$, return $\perp$
$\ell \leftarrow \mathbb{Z}_p; \tau_\ell = v_i^\ell; \tau'_\ell = v_j^\ell$
$c = \tau_\ell \cdot (\tau'_\ell)^{-1}$
$\phi \leftarrow \mathsf{P}_{\mathrm{NIZK}}(crs, (gpk, y, v_j, c),$
                         $(\sigma, \tau_F, \tau_H, v_i, \tau_\ell, \tau'_\ell))$
Return $(\phi, c)$
```

```
$\mathsf{DJudge}(gpk, j, \mathsf{reg}, m, \Sigma, (\phi, c))$:
Return 1 if the following hold:
 $\mathsf{GVf}(gpk, \mathsf{reg}, m, \Sigma) = 1$
 $1 = \mathsf{V}_{\mathrm{NIZK}}(crs, (gpk, y, v_j, c), \phi)$
 $c \neq 1$
else return 0
```

Figure 1: The Proposed Deniable Group Signature Scheme

Remark. In the modified Groth scheme, we can confirm that the user i is an actual signer by checking the equation $e(\sigma, v_i g^{\mathcal{H}(vk_{\mathrm{sots}})}) = e(g, g)$. Now, the statement that we want to prove is $e(\sigma, v_j g^{\mathcal{H}(vk_{\mathrm{sots}})}) \neq e(g, g)$. From these, it is natural to think that when $\{e(\sigma, v_i g^{\mathcal{H}(vk_{\mathrm{sots}})})/e(\sigma, v_j g^{\mathcal{H}(vk_{\mathrm{sots}})})\}^\ell = c$, where i is the actual signer, a user j is not the signer and ℓ is a randomness, we check whether $c \neq 1$. However, the Groth-Sahai proof [26] has a limitation related to languages. If we provide an NIZK proof for the equation $\{e(\sigma, v_i g^{\mathcal{H}(vk_{\mathrm{sots}})})/e(\sigma, v_j g^{\mathcal{H}(vk_{\mathrm{sots}})})\}^\ell = c$ in the Groth-Sahai proof, we need to find $\tilde{g}, \hat{g} \in G$ such that $c = e(\tilde{g}, \hat{g})$. However, decomposing a pairing element is difficult because of the pairing inversion problem [22]. In contrast, in the proposed scheme, all witnesses are base group elements in our construction. Therefore, we can avoid to break the pairing inversion problem since all target group elements are decomposed into known base group elements.

4.3 Performance Evaluation

Since our proposed scheme is exactly same as the modified Groth scheme [25], except for the algorithms for generating and verifying denial proofs, the efficiency of the remaining algorithms is identical to that of the modified Groth scheme. However, it remains determine the size of the denial proofs, which we will discuss in this subsection.

The proof ϕ in our scheme shows the knowledge of the opening proof $(i, (\sigma, \tau_F, \tau_H))$, τ_ℓ, and τ'_ℓ which satisfy

$$e(F, \tau_F) = e(y_1, g) \ \wedge \ e(H, \tau_H) = e(y_2, g) \ \wedge \ \sigma\tau_F\tau_H = y_3$$
$$\wedge \ e(\sigma, v_i g^{\mathcal{H}(vk_{\mathrm{sots}})}) = e(g, g) \ \wedge \ e(v_i, \tau'_\ell) = e(v_j, \tau_\ell) \ \wedge \ c = \tau_\ell \cdot (\tau'_\ell)^{-1}.$$

Groth-Sahai proof [26] gives us the NIZK proof of knowledge for these 6 equations being simultaneously satisfiable that consists of 96 group elements. This is about twice of the size of a signature in the Groth scheme.

For the first equation, the second equation, and the fourth equation, we add the equation and rewrite the pairing product equation to get paring product equations of the type where we can produce zero-knowledge proof. For example, the first equation $e(F, \tau_F) = e(y_1, g)$ is changed to two equations $e(F, \tau_F) \cdot e(y'_1, g^{-1}) = 1 \ \wedge \ y'_1 \cdot y_1^{-1} = 1$ where y'_1 is a new witness. In the same way, $e(H, \tau_H) = e(y_2, g)$ is changed to $e(H, \tau_H) \cdot e(y'_2, g^{-1}) = 1 \ \wedge \ y'_2 \cdot y_2^{-1} = 1$ where y'_2 is a witness. Moreover, $e(\sigma, v_i g^{\mathcal{H}(vk_{\mathrm{sots}})}) = e(g, g)$ is changed to $e(\sigma, v_i g^{\mathcal{H}(vk_{\mathrm{sots}})}) \cdot e(g', g^{-1}) = 1 \ \wedge \ g' \cdot g^{-1} = 1$ where g' is a witness.

Therefore, this denial proof consists of 6 commitments and 3 new commitments, which consist of 3 group elements each, and four pairing product equations, which consist of 9 group elements but a linear equation consisting of 3 group elements, and five multi-scalar multiplication equations, which consist of 9 group elements each.

5 Security Analysis

In this section, we give proofs of security requirements, correcttness, anonymity, non-frameability, traceability, and opening soundness.

In the proof of anonymity, an adversary is allowed to issue DOpen queries even for the challenge group signature. Since the opening query for the challenge group signature is not allowed in the anonymity game of the modified Groth scheme, we cannot use the challenger of the modified Groth scheme. That is, the simulator needs to respond DOpen queries for the challenge group signature without knowing its opening result. The detail is given in Theorem 5.2. Except for from Game 7 to Game 8, translations between games are almost same as those of the modified Groth scheme [41].

In the proofs of other security requirements, intuitively, we can directly break the modified Groth scheme by using an adversary who breaks our scheme if the wining conditions are independent of deniability. In the deniability-related parts, we can also give a proof in a similar way by assuming the security of building blocks.

Theorem 5.1. *The proposed group signature scheme satisfies correctness.*

Proof. Correctness of the scheme follows from the correctness of both the modified Groth scheme and the NIZK proof of knowledge.

□

Theorem 5.2. *The proposed group signature scheme satisfies anonymity if the DLIN assumption holds in G, the one-time signature scheme is strong existential unforgeable, and the hash function is universal one-way.*

Proof. Let $\mathcal{A}_{anon}$ be an adversary that has the advantage ϵ in the anonymity game. Now, we gradually modify the game played by $\mathcal{A}_{anon}$. In the following S_i denotes the event that $\mathcal{A}_{anon}$ successfully guesses the bit $b = b'$ interacting with the environment of Game i.

Game 0. Game 0 is identical to the game in the definition of anonymity. In this game, we have $\Pr[S_0] = \frac{1}{2}+\epsilon$.

Game 1. We modify the behavior of the Open oracle and the DOpen oracle as follows. If they receive a valid group signature which reuses the verification key vk^*_{sots} of the challenge signature Σ^*, the game aborts. By the strong existential unforgeability of one-time signature scheme, this modification does not change the success probability of $\mathcal{A}_{anon}$ with more than negligible amount, that is, we have that $|\Pr[S_0] - \Pr[S_1]|$ is negligible.

Game 2. We further modify the Open oracle and the DOpen oracle to abort when a queried group signature contains vk_{sots} where $\mathcal{H}(vk_{\mathrm{sots}}) = \mathcal{H}(vk^*_{\mathrm{sots}})$. By the universal one-wayness of the hash function, this modification does not change the success probability of $\mathcal{A}_{anon}$ with more than negligible amount, that is, we have that $|\Pr[S_1] - \Pr[S_2]|$ is also negligible.

Game 3. Now, we modify the way to generate the public key for the tag-based encryption. We set $K = g^{\kappa}, L = g^{\lambda}$ and store κ, λ. This modification does not vary the behavior of the adversary $\mathcal{A}_{anon}$, that is, $\Pr[S_2] = \Pr[S_3]$.

Game 4. We then modify how the Open oracle and the DOpen oracle obtain a signer identity i. Until Game 3, when the Open oracle and the DOpen oracle receive a query, they first extract a witness (b, v, σ) from the proof π by using the extraction key xk and search for i such that $\mathsf{reg}[i] = v$. However, in Game 4, the Open oracle and the DOpen oracle search for i such that $e(\sigma, v_i g^{\mathcal{H}(vk_{\mathrm{sots}})}) = e(g, g)$ going through reg. This verification equation uniquely defines v_i given σ and $\mathcal{H}(vk_{\mathrm{sots}})$. Furthermore, since the soundness of π guarantees that σ is a valid signature on $\mathcal{H}(vk_{\mathrm{sots}})$ under the extracted v, v_i identified in above equation must be identical to v. Hence, $\Pr[S_3] = \Pr[S_4]$.

Game 5. In Game 5, we modify how the Open oracle and the DOpen oracle obtain the signature σ. When the oracles receive a valid group signature, they use κ and λ to decrypt the tag-based ciphertext and extract σ instead of extracting from the proof of knowledge π. By the validity check of the tag-based ciphertext and the soundness of the NIZK proof ψ, this gives the same signature σ which we obtain when running the extractor on the NIWI proof of knowledge. Hence, $\Pr[S_4] = \Pr[S_5]$.

Game 6. Now, we change how to produce (τ_F, τ_H), which is a part of opening proof, is generated. Instead of using xk, the Open oracle and the DOpen oracle use κ and λ to compute (τ_F, τ_H) as

$$\tau_F = (y_4/y_1^{\kappa})^{1/\mathcal{H}(vk_{\mathrm{sots}})}, \tau_H = (y_5/y_2^{\lambda})^{1/\mathcal{H}(vk_{\mathrm{sots}})}, \text{ and } \sigma = y_3/\tau_F\tau_H.$$

The response of the Open oracle and the DOpen oracle in Game 6 are exactly same with one in Game 5. Hence, $\Pr[S_5] = \Pr[S_6]$.

Game 7. In Game 6, the Open oracle and the DOpen oracle no longer need the extraction key xk. We therefore now switch to using a simulated common reference string crs that provides perfect witness-indistinguishability and perfect zero-knowledge. Since a simulated common reference string and a real common reference string are computationally indistinguishable under the DLIN assumption, the success probability of the adversary $\mathcal{A}_{anon}$ will not change by more than a negligible amount, hence we have that $|\Pr[S_6] - \Pr[S_7]|$ is negligible. Furthermore, proofs ψ and ϕ are simulated with zero-knowledge trapdoor.

Game 8. Finally, we change the component y_3 in the challenge to a random element. As shown in [41], this will not introduce more than a negligible change in the success probability of the adversary $\mathcal{A}_{anon}$ assuming the DLIN assumption holds. However, one point is different from the proof of the modified Groth scheme. In anonymity game of deniable group signature, the adversary can query even the challenge signature to the DOpen oracle exempt for the challenge users. Therefore, a denial proof $\phi \leftarrow \mathsf{P}_{\mathrm{NIZK}}(crs, (gpk, y, v_j, c), (\sigma, \tau_F, \tau_H, v_i, \tau_\ell, \tau'_\ell))$ needs to be generated even though the witnesses $(\sigma, \tau_F, \tau_H, v_i, \tau_\ell, \tau'_\ell)$ are not known. Since a denial proof in the proposed scheme is NIZK, the simulator can produce a simulated proof. More precisely, when the simulator receives a denial open query (m, Σ, j), the simulator verifies the signature first and, if it is not valid, he returns $\perp$. In the case that the signature is valid, he generates a simulated proof ϕ from the zero-knowledge trapdoor and random c from G, and outputs (ϕ, c). The randomness c has the same distribution as $(v_i/v_j)^\ell$ where ℓ is random in $\mathbb{Z}_p$, hence $|\Pr[S_7] - \Pr[S_8]|$ is negligible.

In Game 8, we can conclude that $\Pr[S_8] = \frac{1}{2}$, because the view of the adversary is independent from the challenge bit b. In particular, the challenge $(vk^*_{\mathrm{sots}}, a, \pi, y, \psi, \sigma^*_{\mathrm{sots}})$ contains no information of bit b. Indeed, vk^*_{sots} is independently generated, a is re-randomized and uniformly random, the perfectly witness-indistinguishable proof π distributes independently from the witness, and y is a random encryption. Also, the proof ψ does not contain the information of b since the proof is computed from y and π by using the zero-knowledge trapdoor. Moreover, $\psi, \sigma^*_{\mathrm{sots}}$ is a signature of $(vk^*_{\mathrm{sots}}, m, a, \pi, y, \psi)$ and the oracles behave independently of bit b.

□

Theorem 5.3. *The proposed group signature scheme satisfies non-frameability if the modified Groth scheme has non-frameability.*

Proof. We assume the adversary $\mathcal{A}_{nf}$ who breaks the non-frameability of the proposed scheme. Let (m, Σ, j, τ_O) be the output of $\mathcal{A}_{nf}$. The adversary $\mathcal{A}_{nf}$ has two types of forgery, one is producing the valid group signature of honest user j and its acceptable opening proof and the other is producing the valid group signature of honest user j where his denial opening of the signature by DOpen is not accepted by DJudge.

The first forgery, where the Judge algorithm outputs 1, is trivially captured by the forgery of the non-frameability of the modified Groth scheme. Since the modified Groth scheme has non-frameability under the q-SDH assumption, the strong existential unforgeability of one-time signature scheme, and universal one-wayness of hash function, this type of forgery will not happen.

In the second forgery, where the DJudge algorithm outputs 0, because the group signature is valid, opening result $(i, (\sigma, \tau_F, \tau_H))$ of the signature has valid structure, that is, $\sigma = g^{1/(x_i + \mathcal{H}(vk_{\mathrm{sots}}))}$. If $i \neq j$, j's denial opening of the signature, that the adversary outputs, by DOpen is accepted by DJudge, because the proposed scheme satisfies correctness. This contradicts the wining condition of the adversary. So, from now on, we assume that $i = j$. Then, σ is a forgery of the certified signature. More precisely, if $i = j$, σ is a forgery of the certified signature. More precisely, let $\mathcal{A}^{type2}_{nf}$ be an adversary who breaks the non-frameability of the proposed scheme using second type of forgery and we can construct the adversary $\mathcal{B}$ who breaks the existential unforgeability against weak chosen message attack of the certified signature. Here, by assuming that $\mathcal{A}^{type2}_{nf}$ will query to the GSig oracle in $q(k)$ times, $\mathcal{B}$ is constructed as follows. Before running the game, $\mathcal{B}$ tries to guess the user j that $\mathcal{A}^{type2}_{nf}$ will frame in the non-frameability game of the proposed scheme. The probability is at least $\frac{1}{N(k)}$ where $N(k)$ is an upper bound of the number of honest users. In the existential unforgeability game, $\mathcal{B}$ gets the challenge verification key and certificate $(v^*, cert^*)$ after he gets a description gk and sends a public authority key ak to the challenger. $\mathcal{B}$ queries $\mathcal{H}(vk_{\mathrm{sots},i})$ and gets $\sigma_i = g^{1/(x^* + \mathcal{H}(vk_{\mathrm{sots},i}))}$ in advance, where $1 \leq i \leq q$ and x^* is the signing key of v^* which is unknown to $\mathcal{B}$. After that, $\mathcal{B}$ generates keys of the proposed scheme by following GKg and using gk which is given by the challenger and sends them to $\mathcal{A}^{type2}_{nf}$. When $\mathcal{A}^{type2}_{nf}$ accesses to oracles $\mathsf{SndToU}(\cdot)$, $\mathsf{WReg}(\cdot)$, $\mathsf{GSig}(\cdot, \cdot)$, $\mathsf{USK}(\cdot)$, and $\mathsf{CrptU}(\cdot, \cdot)$, $\mathcal{B}$ can easily respond to the queries because he has all keys of the proposed scheme. However, only when $\mathcal{A}^{type2}_{nf}$ accesses to oracles $\mathsf{SndToU}(j)$ and $\mathsf{GSig}(j, \cdot)$, for each queries, $\mathcal{B}$ lets $(v^*, cert^*)$ as j's verification key and certificate, and uses σ_i which he got from the challenger of the existential unforgeability game to simulate the signing. We do not need to care about $\mathsf{USK}(j)$ since $\mathcal{A}^{type2}_{nf}$ never queries $\mathsf{USK}(j)$ on the wining condition of the non-frameability game. Finally, $\mathcal{A}^{type2}_{nf}$ outputs the forgery $(m, \Sigma = (vk^*_{\mathrm{sots}}, \cdot, \cdot, \cdot, \cdot, \cdot), j, \tau_O)$ which satisfies $\sigma = g^{1/(x^*+\mathcal{H}(vk^*_{\mathrm{sots}}))}$ where $(i,(\sigma,\tau_F,\tau_H)) \leftarrow \mathsf{Open}(gpk, ok, \mathsf{reg}, m, \Sigma)$ and $i \neq j$. By the strong existential unforgeability of one-time signature scheme, vk^*_{sots} is not one of $vk_{\mathrm{sots},i}$ which is used in $\mathsf{GSig}(j,\cdot)$ with overwhelming probability. Moreover, by universal one-wayness of hash function, $\mathcal{H}(vk^*_{\mathrm{sots}})$ does not collide with one of $vk_{\mathrm{sots},i}$, that is, $\mathcal{H}(vk^*_{\mathrm{sots}}) \neq \mathcal{H}(vk_{\mathrm{sots},i})$ holds, with overwhelming probability. Therefore, $\mathcal{B}$ can extract σ from (m,Σ) and output $(cert^*, \mathcal{H}(vk^*_{\mathrm{sots}}), \sigma)$ as a forgery of the certified signature. This contradicts that the certified signature is existential unforgeable against weak chosen message attack.

□

Theorem 5.4. *The proposed group signature scheme satisfies traceability if the modified Groth scheme has traceability.*

Proof. We assume the adversary $\mathcal{A}_{trace}$ who breaks the traceability of the proposed scheme. The adversary $\mathcal{A}_{trace}$ has two types of forgery, one is producing the valid group signature whose opening result is not valid or opening proof is not accepted by Jugde and the other is producing the valid group signature, whose opening proof is accepted by Jugde, but for the same user the denial opening proof is accepted by DJugde. The first forgery is just also the forgery of the traceability of the modified Groth scheme. Since the modified Groth scheme has traceability, this type of forgery will not happen. In the second forgery, from the fact that denial opening proof is accepted by DJudge, we require $c \neq 1$ where denial opening proof is (ϕ, c). However, since the Judge algorithm outputs 1 for i, i is the signer of Σ. Then $c = 1$ holds. Therefore, the DJudge algorithm, that checks $c \neq 1$, never output 1, and this case never happen.

□

Theorem 5.5. *The proposed group signature scheme satisfies opening soundness if the modified Groth scheme has opening soundness.*

Proof. We assume the adversary $\mathcal{A}_{os}$ who breaks the opening soundness of the proposed scheme. The adversary $\mathcal{A}_{os}$ has two types of forgery, one is producing the valid group signature and its opening proofs of i and j which are both accepted by Judge and the other is producing the valid group signature , its opening proof which is accepted by Judge, and its denial opening proof which is also accepted by DJudge. The first forgery is just also the forgery of the opening soundness of the modified Groth scheme. Since the modified Groth scheme has opening soundness, this type of forgery will not happen. In the second forgery, from the fact that denial opening proof is accepted by DJudge, we require $c \neq 1$ where denial opening proof is (ϕ, c). However, since the Judge algorithm outputs 1 for i, i is the signer of Σ. Then $c = 1$ holds. Therefore, the DJudge algorithm, that checks $c \neq 1$, never output 1, and this case never happen.

□

References

[1] M. Abe, S. S. M. Chow, K. Haralambiev, and M. Ohkubo. Double-trapdoor anonymous tags for traceable signatures. In *ACNS*, pages 183–200, 2011.

[2] L. E. Aimani and M. Joye. Toward practical group encryption. In *ACNS*, pages 237–252, 2013.

[3] M. H. Au, J. K. Liu, W. Susilo, and T. H. Yuen. Secure ID-based linkable and revocable-iff-linked ring signature with constant-size construction. *Theor. Comput. Sci.*, 469:1–14, 2013.

[4] M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *EUROCRYPT*, pages 614–629, 2003.

[5] M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. In *CT-RSA*, pages 136–153, 2005.

[6] P. Bichsel, J. Camenisch, G. Neven, N. P. Smart, and B. Warinschi. Get shorty via group signatures without encryption. In *SCN*, pages 381–398, 2010.

[7] M. Blum, P. Feldman, and S. Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *STOC*, pages 103–112, 1988.

[8] D. Boneh and X. Boyen. Short signatures without random oracles. In *EUROCRYPT*, pages 56–73, 2004.

[9] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *CRYPTO*, pages 41–55, 2004.

[10] X. Boyen and B. Waters. Compact group signatures without random oracles. In *EUROCRYPT*, pages 427–444, 2006.

[11] X. Boyen and B. Waters. Full-domain subgroup hiding and constant-size group signatures. In *Public Key Cryptography*, pages 1–15, 2007.

[12] J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *CRYPTO*, pages 56–72, 2004.

[13] J. Camenisch, G. Neven, and M. Rückert. Fully anonymous attribute tokens from lattices. In *SCN*, pages 57–75, 2012.

[14] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited (preliminary version). In *STOC*, pages 209–218, 1998.

[15] J. Cathalo, B. Libert, and M. Yung. Group encryption: Non-interactive realization in the standard model. In *ASIACRYPT*, pages 179–196, 2009.

[16] D. Chaum and E. van Heyst. Group signatures. In *EUROCRYPT*, pages 257–265, 1991.

[17] S. S. M. Chow, V. K.-W. Wei, J. K. Liu, and T. H. Yuen. Ring signatures without random oracles. In *ASIACCS*, pages 297–302, 2006.

[18] C.-K. Chu and W.-G. Tzeng. Identity-committable signatures and their extension to group-oriented ring signatures. In *ACISP*, pages 323–337, 2007.

[19] C. Delerablée and D. Pointcheval. Dynamic fully anonymous short group signatures. In *VIETCRYPT*, pages 193–210, 2006.

[20] M. Fischlin. Communication-efficient non-interactive proofs of knowledge with online extractors. In *CRYPTO*, pages 152–168, 2005.

[21] J. Furukawa and H. Imai. An efficient group signature scheme from bilinear maps. *IEICE Transactions*, 89-A(5):1328–1338, 2006.

[22] S. D. Galbraith, F. Hess, and F. Vercauteren. Aspects of pairing inversion. *IEEE Transactions on Information Theory*, 54(12):5719–5728, 2008.

[23] S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2):281–308, 1988.

[24] S. D. Gordon, J. Katz, and V. Vaikuntanathan. A group signature scheme from lattice assumptions. In *ASIACRYPT*, pages 395–412, 2010.

[25] J. Groth. Fully anonymous group signatures without random oracles. In *ASIACRYPT*, pages 164–180, 2007.

[26] J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *EUROCRYPT*, pages 415–432, 2008.

[27] A. Kiayias, Y. Tsiounis, and M. Yung. Traceable signatures. In *EUROCRYPT*, pages 571–589, 2004.

[28] A. Kiayias, Y. Tsiounis, and M. Yung. Group encryption. *IACR Cryptology ePrint Archive*, 2007:15, 2007.

[29] A. Kiayias and M. Yung. Group signatures: Provable security, efficient constructions and anonymity from trapdoor-holders. *IACR Cryptology ePrint Archive*, 2004:76, 2004.

[30] E. Kiltz. Chosen-ciphertext security from tag-based encryption. In *TCC*, pages 581–600, 2006.

[31] Y. Komano, K. Ohta, A. Shimbo, and S. Kawamura. Toward the fair anonymous signatures: Deniable ring signatures. In *CT-RSA*, pages 174–191, 2006.

[32] F. Laguillaumie, A. Langlois, B. Libert, and D. Stehlé. Lattice-based group signatures with logarithmic signature size. In *ASIACRYPT*, pages 41–61, 2013.

[33] A. Langlois, S. Ling, K. Nguyen, and H. Wang. Lattice-based group signature scheme with verifier-local revocation. In *Public Key Cryptography*, pages 345–361, 2014.

[34] B. Libert and M. Joye. Group signatures with message-dependent opening in the standard model. In *Topics in Cryptology - CT-RSA 2014 - The Cryptographer's Track at the RSA Conference 2014, San Francisco, CA, USA, February 25-28, 2014. Proceedings*, pages 286–306, 2014.

[35] B. Libert, T. Peters, and M. Yung. Group signatures with almost-for-free revocation. In *CRYPTO*, pages 571–589, 2012.

[36] B. Libert, M. Yung, M. Joye, and T. Peters. Traceable group encryption. In *Public Key Cryptography*, pages 592–610, 2014.

[37] Y.-D. Lyuu and M.-L. Wu. Convertible group undeniable signatures. In *ICISC*, pages 48–61, 2002.

[38] K. Ohara, Y. Sakai, K. Emura, and G. Hanaoka. A group signature scheme with unbounded message-dependent opening. In *8th ACM Symposium on Information, Computer and Communications Security, ASIA CCS '13, Hangzhou, China - May 08 - 10, 2013*, pages 517–522, 2013.

[39] R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *ASIACRYPT*, pages 552–565, 2001.

[40] Y. Sakai, K. Emura, G. Hanaoka, Y. Kawai, T. Matsuda, and K. Omote. Group signatures with message-dependent opening. In *Pairing*, pages 270–294, 2012.

[41] Y. Sakai, J. C. N. Schuldt, K. Emura, G. Hanaoka, and K. Ohta. On the security of dynamic group signatures: Preventing signature hijacking. In *Public Key Cryptography*, pages 715–732, 2012.

[42] J. C. N. Schuldt and K. Matsuura. Efficient convertible undeniable signatures with delegatable verification. *IEICE Transactions*, 94-A(1):71–83, 2011.

[43] V. K. Wei, T. H. Yuen, and F. Zhang. Group signature where group manager, members and open authority are identity-based. In *ACISP*, pages 468–480, 2005.

[44] S. Zeng and S. Jiang. A new framework for conditionally anonymous ring signature. *Comput. J.*, 57(4):567–578, 2014.

[45] S. Zhou and D. Lin. Shorter verifier-local revocation group signatures from bilinear maps. In *CANS*, pages 126–143, 2006.