An r-Hiding Revocable Group Signature Scheme: Group Signatures with the Property of Hiding the Number of Revoked Users

Emura, Keita; Miyaji, Atsuko; Omote, Kazumasa

Journal of Applied Mathematics, 2014: Article ID 983040

2014-06-01

Journal Article

http://hdl.handle.net/10119/12841

Copyright © 2014 Keita Emura et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.
Research Article

An r-Hiding Revocable Group Signature Scheme: Group Signatures with the Property of Hiding the Number of Revoked Users

Keita Emura,1 Atsuko Miyaji,2 and Kazumasa Omote2

1 National Institute of Information and Communications Technology (NICT), 4-2-1 Nukui-Kitamachi, Koganei, Tokyo 184-8795, Japan
2 Japan Advanced Institute of Science and Technology (JAIST), 1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan

Correspondence should be addressed to Keita Emura; k-emura@nict.go.jp

Received 11 November 2013; Accepted 14 April 2014; Published 1 June 2014

Copyright © 2014 Keita Emura et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

If there are many displaced workers in a company, then a person who goes for job hunting might not select this company. That is, the number of members who quit is quite negative information. Similarly, in revocable group signature schemes, if one knows (or guesses) the number of revoked users (say r), then one may guess the reason behind such circumstances, and it may lead to harmful rumors. However, no previous revocation procedure can achieve hiding r. In this paper, we propose the first revocable group signature scheme, where r is kept hidden, which we call r-hiding revocable group signature. To handle this property, we newly define the security notion called anonymity with respect to the revocation which guarantees the unlinkability of revoked users.

1. Introduction

Imagine that there are many users who have stopped using a service. If this fact is published, then how would the newcomers feel about this? One may guess the reason behind such circumstances and may judge that those users did not find the service attractive or the service fee is expensive. The same thing may occur in other cases; for example, if there are many displaced workers in a company, then a person who goes for job hunting might not select this company. For example, the person might imagine there are many problematic employees in this company or might imagine the labor environment may not be good. That is, the number of members who quit is quite negative information.

Group Signature. Many cryptographic attempts for the revocation of rights of users have been considered so far. In this paper, we mainly focus on group signature. The concept of group signature was investigated by Chaum and van Heyst [1]. A typical usage of group signature is described as follows. The group manager (GM) issues a membership certificate to a signer. A signer makes a group signature by using their own membership certificate and sends it (with a signed message) to a verifier. The verifier anonymously verifies whether a signer is a member of a group or not. That is, the verifier checks the possession of a membership certificate without revealing themself. In order to handle some special cases (e.g., an anonymous signer behaves maliciously), GM can identify the actual signer through the open procedure. Since verifiers do not have to identify individual signers, group signature is a useful and powerful tool for protecting signers’ privacy.

As additional functionality of group signature, anonymity revocation has been introduced [2–11], where no revoked users can make a valid group signature or revoked users can be publicly detected even if they try to make a group signature. (Since a long RSA modulus might lead to certain inefficiency aspects (e.g., long signatures, heavy complexity costs, and so on), we exclude RSA-based revocable group signatures (e.g., [12, 13]) in this paper.) However, the number of revoked users (say r) is revealed in all previous revocable group signature schemes. As mentioned previously, the number of revoked users r is quite negative information. Next, we introduce applications of revocable group signature for outsourcing businesses [14] and biometric authentication [15].
as concrete examples, where revealing $r$ may lead to harmful rumors.

**Concrete Example 1 (identity management).** In this application, presented in [14], there are four entities: a user, outsourcee, opening manager (OM), and revocation manager (RM). Let outsourcee be in charge of providing the service to legitimate users. When a user requests the service, the user makes a group signature and sends it to outsourcee. Due to anonymity of the underlying group signature scheme, outsourcee does not have to identify individual users (protect users’ privacy). One important thing is that outsourcee does not have to manage a list of identities of users. That is, the risk of leaking user information (i.e., the user list) can be minimized, and this is the merit of using group signature in identity management. After a certain interval, for charging a service fee, OM detects a user by using the opening procedure of group signature. If a user does not pay a fee (or when a user wants to leave the service), then OM anounces the identity of this user to RM, and RM revokes this user from the system. In this system, if $r$ is revealed, then one may think that there might be many dropout users who have stopped using the service; that is, this service may not be interesting, or he/she have not paid the service fee; namely, the service fee may be expensive and so on. That is, “revealing $r$” may lead to harmful rumors.

**Concrete Example 2 (biometric authentication).** In this application presented in [15], there are four entities: a human user, a sensor client, a card issuer, and a service provider. A human user authenticates himself/herself to the service provider by using his/her biometric data preserved on a plastic card. A card issuer (with a group master secret key) issues a card to a human user which contains a signing key and his/her biometric data. Moreover, the card issuer can revoke users if malicious behavior occurs or a user loses his/her card. A sensor client extracts human user’s biometric trait (e.g., iris is used in [15]) and communicates with the service provider, so that the user will be authenticated by the service provider. The service provider verifies a group signature and provides a service (e.g., open a door) if the signature is valid. Due to anonymity, the service provider does not identify who the user is; even sensitive biometric information is treated.

In this system, if someone knows $r$ in this application, they may think that there might be many malicious behaviors, or there might be many lost cards; that is, good management may deteriorate, and so on. That is, “revealing $r$” may lead to harmful rumors.

**Our Target.** So, our main target is to propose a revocable group signature scheme with the property of hiding the number of revoked users $r$, which we call $r$-hiding revocable group signature. Then, we need to investigate the methodology for achieving the following.

1. The size of any value does not depend on $r$.
2. The costs of any algorithm do not depend on $r$, except the revocation algorithm executed by GM.
3. Revoked users are unlinkable.

In particular, if revoked users are linkable, then anyone can guess (i.e., not exactly obtain) $r$ by linking and counting revoked users. Although we assume that an adversary can obtain the polynomial (of the security parameter) number of group signatures, this assumption is not unreasonable (actually, the adversary can be allowed to access the signing oracle in polynomial times). In addition, $r$ is also a polynomial-size value. That is, this guessing attack works given that revoked users are linkable.

However, no previous revocable signature scheme satisfying all requirements above has been proposed. For example, in revocable group signatures [2, 4, 11] (which are based on updating the group public values, e.g., using accumulators), either the size of public value or the costs of updating membership certificate depend on $r$. Nakano et al. [6] proposed a novel technique of group signature, where no costs of the GSign algorithm (or the Verify algorithm also) depend on $r$. However, their methodology requires that $r$ signatures are published to make a group signature, and therefore $r$ is revealed. Recently, Libert-Peters-Yung proposed two scalable group signature schemes [7, 8] by applying the Naor-Naor-Lotspeich (NNL) broadcast encryption framework [16]. However, at least one cost depends on $r$ (e.g., $O(r)$-size revocation list is required for signing in [7, 8] (of subset difference variant) and $O(r \log(N/r))$-size revocation list is required for signing in [8] (of complete subtree variant) and $N$ (the number of users) are publicly available). Therefore, $r$ is revealed. In [3, 5, 10, 17, 18] (which are verifier-local revocation (VLR) type group signature schemes), revoked users are linkable. In this case, anyone can guess $r$ by executing the verification procedure. For the sake of clarity, we introduce the Nakanishi-Funabiki methodology [10] as follows: let $RL = \{h^{x_1}, h^{x_2}, \ldots, h^{x_r}\}$ be the revocation list, where $x_i$ is the secret value of revoked user $U_i$. Note that, by adding dummy values, we can easily expand the size of revocation list [RL]. So, we can assume that $r$ is not revealed from the size of RL, but $r$ is revealed (or rather, guessed) as follows. Each group signature $\sigma$ (made by $U_j$) contains $f^x \cdot h^\beta$ and $h^\beta$ for some random $\beta$ and some group elements $f$ and $h$. If $U_j$ has been revoked, then there exists $h^{x_i}$ such that $\epsilon(f^x, h^\beta, h) = \epsilon(h^{x_i}h^\beta, f)$ holds. By counting such $i$, one can easily guess $r$ even if RL is expanded by dummy values. Since each value in RL is linked to a user (i.e., $h^{x_i}$ is linked to $U_i$), even if values in RL are randomized (e.g., $(h^{x_i})^r$ for some random $r$), this connection between a user and a value in RL is still effective. So, one can easily guess $r$ even if RL is randomized.

From the above considerations, no previous revocation procedure can be applied for hiding $r$. One solution has been proposed in [19], where only the designated verifier can verify the signature. By preventing the verification of signature from the third party, $r$ is not revealed from the viewpoint of the third party. However, this scheme (called anonymous designated verifier signature) is not publicly verifiable and is not group signature any longer. Next, as another methodology, we may consider multigroup signatures [20, 21] with two groups (valid user group and revoked user group). However, this attempt does not work, since each user is given his/her
membership certificate (corresponding to the group he/she belongs to) in the initial setup phase, and the revocation procedure is executed after the setup phase.

Our Contribution. In this paper, we propose the first r-hiding revocable group signature scheme in the random oracle model by applying attribute-based group signature (ABGS) [22–25]. By considering two attributes: (1) valid group user and (2) the user’s identity, we can realize the property of hiding r. To handle this property, we newly define the security notion called anonymity with respect to the revocation. As the main distinction among our anonymity definition and previous ones, to guarantee the unlinkability of revoked users, we can issue the revocation queries against the challenge users. Our scheme is secure under the computational Diffie-Hellman (CDH) assumption, the decision Diffie-Hellman (DDH) assumption and assuming a bilinear group (i.e., the external Diffie-Hellman (XDH) assumption), the decision linear (DLIN) assumption, the hidden strong Diffie-Hellman (HSDH) assumption, and the q-strong Diffie-Hellman (SDH) assumption, in the random oracle model. We apply the Boldyreva multisignature scheme [26] to revoke each user.

Related Work. There were several security definitions of group signatures until the Bellare-Micciancio-Warinschi work [27], which we call the BMW model. They showed that full-anonymity and full-traceability are enough to capture all security requirements that appeared before their work. Bellare et al. [28] extended the BMW model, which we call the BSZ model, to handle the dynamic group setting, where a user can join the system even after the system setup phase. Later, Sakai et al. [29] further extended the BSZ model for preventing signature hijacking. Independently, Kiayias and Yung also give a formal definition with dynamic join [30, 31], which are applied in our construction. Let PPT mean probabilistic polynomial time, and x ← X means that an element x is chosen at uniform random from a set X.

2. Preliminaries

In this section, we give definitions of bilinear groups and complexity assumptions and introduce cryptographic tools which are applied in our construction. Let PPT mean probabilistic polynomial time, and x ← X means that an element x is chosen at uniform random from a set X.

2.1. Bilinear Groups and Complexity Assumptions

Definition 1 (bilinear groups). Let G1, G2, and G T be cyclic groups of prime order p, and G1 = ⟨g⟩ and G2 = ⟨h⟩. Let e : G1 × G2 → G T be an efficient computable bilinear map with the following properties: (1) bilinearity: for all (g, g′) ∈ G1 2 and (h, h′) ∈ G2 2, e(g, h) e(g′, h′) = e(g, h) e(g′, h′); (2) nondegeneracy: e(g, h) 6= 1, where 1 is the unit element over G T.

Definition 3 (the decision Diffie-Hellman (DDH) assumption). We say that the DDH assumption holds if, for all PPT adversary A, Pr[A(g1, g′1, g2, g′2) ← G 2 1 ] = 0 − Pr[A(g1, g′1, g2, g′2) = 0] is negligible, where (g1, g′1) ← G 2 and (x, r) ← Z 2 p with x 6= r.

Definition 5 (the hidden strong Diffie-Hellman (HSDH) assumption [34]). We say that the HSDH assumption holds if, for all PPT adversary A, Pr[A(g1, h, hω, (g1/(ω+ξ)), H i=1,.. = 0] 6= Pr[A(g1, h, hω, (g1/(ω+ξ)), H i=1,.. = 0] is negligible, where (g1, h) ← G1 × G2, (ω, x1, . . . , xξ) ← Z p and x ∈ Z p.
Definition 6 (the strong Diffie-Hellman (SDH) assumption [56]). We say that the SDH assumption holds if, for all PPT adversary \( \mathcal{A} \), \( \Pr[\mathcal{A}(g_1, h, h^{\omega_1}, \ldots, h^{\omega_s}) = (g_1^{1/(\omega x)}, x)] \) is negligible, where \( (g_1, h) \leftarrow G_1 \times G_2 \), \( \omega \leftarrow \mathbb{Z}_p \), and \( x \in \mathbb{Z}_p \).

Definition 7 (the external Diffie-Hellman (XDH) assumption [4]). Let \((G_1, G_2, G_T)\) be a bilinear group. We say that the XDH assumption holds if for all PPT adversary \( \mathcal{A} \), the DDH assumption over \( G_1 \) holds.

2.2. Other Cryptographic Tools. In this section, we introduce cryptographic tools applied for our construction.

BBS+ Signature [2, 6, 32, 57]. Let \( L \) be the number of signed messages and let \((G_1, G_2, G_T)\) be a bilinear group. Select \( g, g_1, \ldots, g_L \leftarrow G_1 \), \( h \leftarrow G_2 \), and \( \omega \leftarrow \mathbb{Z}_p \), and compute \( \Omega = g^\omega \). The signing key is \( \omega \) and the verification key is \((p, G_1, G_2, G_T, g, g_1, \ldots, g_L, h, \Omega)\). \( G \) is a set of signed messages \((m_1, \ldots, m_L) \in \mathbb{Z}_p^L\), choose \( r, y \leftarrow \mathbb{Z}_p^L \), and compute \( A = (g_1^{m_1 \cdot r} \cdots g_L^{m_L \cdot r}, g_{r+1}^{1/(\omega + y)}) \). For a signature \((A, r, y)\), verification algorithm output 1 if \( e(A, \Omega h) = e(g_1^{m_1 \cdot r} \cdots g_L^{m_L \cdot r}, g_{r+1}^{1/(\omega + y)}) \) holds. The BBS+ signature scheme satisfies existential unforgeability against chosen message attack (EUF-CMA) under the \( \omega \)-SDH assumption. (First an adversary \( \mathcal{A} \) is given \( \omega \), choose messages to \( A \), and outputs a message/signature pair \((M^*, \sigma^*)\). We say that \( \mathcal{A} \) wins if \((M^*, \sigma^*)\) is valid and \( \mathcal{A} \) has not sent \( M^* \) as a signing query. The EUF-CMA security guarantees that the probability \( \Pr[\mathcal{A} \text{ wins}] \) is negligible.)

Linear Encryption [2]. A public key is \( \rho = (u, v, h) \in G_1 \) such that \( u^{X_1} = v^{X_2} = h \) for \( X_1, X_2 \in \mathbb{Z}_p \). \( \rho \) is the corresponding secret key of \((X_1, X_2)\). For a plaintext \( M \in \mathbb{Z}_p \), choose \( \delta_1, \delta_2 \leftarrow \mathbb{Z}_p \) and compute a ciphertext \( C = (F_1, F_2, F_3) \), where \( F_1 = M \cdot h^{\delta_1} \cdot F_2 = u^{\delta_1} \), and \( F_3 = v^{\delta_2} \). \( C \) can be decrypted as \( M = F_1/F_2^{\delta_2} \cdot F_3 \). The linear encryption is IND-CPA secure under the DLIN assumption. (First an adversary \( \mathcal{A} \) is given \( \rho \) and chooses \( x' \leftarrow \mathbb{Z}_p \) and \( \delta \leftarrow \mathbb{Z}_p \), and computes \( C = (F_1, F_2, F_3) \), where \( F_1 = M' \cdot h^{\delta_1} \cdot F_2 = u^{\delta_1} \), and \( F_3 = v^{\delta_2} \). The IND-CPA security guarantees that \( \Pr[\mathcal{A} \text{ wins}] = 1/2 \) is negligible.)

Signature Based on Proof of Knowledge. In our group signature, we apply the conversion of the underlying interactive zero knowledge (ZK) proof into noninteractive ZK (NIZK) proof by applying the Fiat-Shamir heuristic [35]. We describe such converted proof based on proof of knowledge (SPK) as \( \text{SPK}(x : (y, x) \in R(M)) \), where \( x \) is the knowledge to be proved. \( R \) is the relation (e.g., \( y = g^x \) in the case of the knowledge of the discrete logarithm), and \( M \) is the signed message. The SPK has an extractor of the proved knowledge from two accepting protocol views whose commitments are the same but challenges are different.

3. Definitions of Group Signature with the Property of Hiding the Number of Revoked Users

Here, we define the syntax of revocable group signature and security requirements (anonymity with respect to the revocation and traceability) by adapting [6]. Note that our definition follows the static group settings as in the BMW model [27], but we can easily handle the dynamic group settings as in the BSZ model [28] (and non-frameability) by adding an interactive join algorithm.

Definition 8 (syntax of \( r \)-hiding revocable group signature).

Setup. This probabilistic setup algorithm takes as input the security parameter \( 1^k \) and returns public parameters \( \mathbb{P} \).

KeyGen. This probabilistic key generation algorithm (for GM) takes as input the maximum number of users \( N \) and returns the public group key \( \rho \), the secret key \( msk \), the initial secret key \( usk_0 \), a set of revoked users \( RL \), and the initialization revocation-dependent value \( \mathcal{T}_0 \).

GSign. This probabilistic signing algorithm (for a user \( U_i \)) takes as input \( \rho \), \( usk_i \), a signed message \( M \), and a revocation-dependent value \( \mathcal{T} \). It returns a group signature \( \sigma \).

Verify. This deterministic verification algorithm takes as input \( \rho \), \( M \), \( \sigma \), and \( \mathcal{T} \). It returns 1 if \( \sigma \) is a valid group signature and 0 otherwise.

Revoke. This (potentially) probabilistic revocation algorithm takes as input \( \rho \), \( msk \), a set of revoked users \( RL \), and returns \( \mathcal{T} \).

Open. This deterministic algorithm takes as input \( msk \) and a valid pair \((M, \sigma)\) and returns the identity of the signer of \( \sigma \) if \( ID \). If \( ID \) is not a group member, then the algorithm returns 0.

In the Revoke algorithm, we set \( RL_0 = \emptyset \) and assume that the nonrevoked users in \( t \) are \( \{U_1, \ldots, U_N\} \setminus RL_t \). Under this setting, boomerang users (who rejoin the group) are available (i.e., \( U_i \) such that \( U_i \in RL_{t-1} \) and \( U_i \notin RL_t \)). In addition, if an invalid pair \((M, \sigma)\) is input to the Open algorithm, then the Open algorithm easily detects this fact by using the Verify algorithm. So, we exclude this case from the definition of the Open algorithm.

Next, we define anonymity with respect to the revocation and traceability. As the main difference among our anonymity definition and previous ones, \( \mathcal{A} \) can issue the revocation queries against the challenge users in order to guarantee the unlinkability of revoked users. Note that we do not consider the CCA-anonymity, where an adversary \( \mathcal{A} \) can issue the open queries. So, we just handle the CPA-anonymity [2] only in this paper. However, as mentioned by Boneh et al. [2], the CCA-anonymity can be handled by applying a CCA secure public key encryption for implementing the open algorithm.
**Definition 9** (anonymity with respect to the revocation).

**Setup.** The challenger $C$ runs the Setup algorithm and the KeyGen algorithm and obtains params, gpk, msk, and all $\{usk_i\}_{i=1}^N$. $C$ gives params and gpk to $A$ and sets $t = 0$, $RU_0 = \emptyset$, and $CU = \emptyset$, where $RU_0$ denotes the (initial) set of IDs of revoked users and $CU$ denotes the set of IDs of corrupted users.

**Queries.** $A$ can issue the following queries.

- **Revocation.** $A$ can request the revocation of users $ID_{i_1}, \ldots, ID_{i_{k+1}}$ for some constant $k_{t+1} \in [1, N]$. $C$ runs $\mathcal{T}_{t+1} \leftarrow \text{Revoke}(msk, \{ID_{i_1}, \ldots, ID_{i_{k+1}}, \mathcal{T}_t\})$ and adds $ID_{i_1}, \ldots, ID_{i_{k+1}}$ to $RU_{t+1}$.

- **Signing.** $A$ can request a group signature on a message $M$ for a user $U_i$ where $ID_i \notin CU$. $C$ chooses a bit $b \leftarrow \{0, 1\}$ and runs $\sigma \leftarrow \text{GSign}(gpk, usk_i, M, \mathcal{T}_{t+1})$, where $\mathcal{T}_{t+1}$ is the current revocation-dependent value and gives $\sigma$ to $A$.

- **Corruption.** $A$ can request the secret key of a user $U_i$. $C$ adds $ID_i$ to $CU$ and gives $usk_i$ to $A$.

**Challenge.** $A$ sends a message $M^*$ and two users $U_{i_1}$ and $U_{i_2}$, where $ID_{i_1}, ID_{i_2} \notin CU$. $C$ chooses a bit $b' \leftarrow \{0, 1\}$ and runs $\sigma^* \leftarrow \text{GSign}(gpk, usk_i, M^*, \mathcal{T}_{t+1})$, where $\mathcal{T}_{t+1}$ is the current revocation-dependent value and gives $\sigma^*$ to $A$.

**Queries.** It is the same as the previous one (note that no corruption query for the challenge users is allowed).

**Output.** $A$ outputs a guessing bit $b' \in \{0, 1\}$.

We say that anonymity holds if, for all PPT adversaries $A$, the advantage

$$\text{Adv}^{\text{anon}}_{A}(\kappa) := \Pr[b = b'] - \frac{1}{2}$$

is negligible.

There are two types of revocable group signature such that (1) any users can make a valid group signature, but anyone can check whether a signer has been revoked or not [3, 5, 10, 17], or (2) no revoked user can make a valid group signature without breaking traceability [2, 4, 6, 11]. We implicitly require the second type of revocable group signature, since clearly anonymity is broken if one of the challenge users is revoked in a first type scheme. We also require that the challenger $C$ (that has msk) can break traceability to compute the challenge group signature $\sigma^*$ for the case that a challenger user is revoked. Note that since msk is used for generating user’s secret keys, obviously any entity with msk makes an “untraceable” group signature, and this fact does not detract the security of our group signature.

One may think that the above anonymity definition can be extended that $A$ can issue the corruption query against the challenge users as in the full-anonymity [27]. It might be desired that $r$ is not revealed even if revoked users reveal their secret signing keys, since their signing keys are already meaningless (i.e., the rights of signing have expired). For example, if users are not intentionally revoked (e.g., a user has not paid in the outsourcing businesses example [14]), then users might reveal their secret signing keys to compromise the systems. Or, even if users are intentionally revoked (e.g., they feel that this service is not interesting in the outsourcing businesses example), they might reveal their secret signing keys as a crime for pleasure. However, even if $r$ is kept hidden when revoked users reveal their secret signing keys, one can easily guess $r$ by counting the number of revealed secret keys. So, in our opinion such secret key leakage resilient property is too strong, and therefore, our proposed group signature does not follow this leakage property.

Next, we define traceability.

**Definition 10** (traceability).

**Setup.** The challenger $C$ runs the Setup algorithm and the KeyGen algorithm and obtains params, gpk, msk, and all $\{usk_i\}_{i=1}^N$. $C$ gives params and gpk to $A$ and sets $t = 0$, $RU_0 = \emptyset$, and $CU = \emptyset$, where $RU_0$ denotes the (initial) set of IDs of revoked users and $CU$ denotes the set of IDs of corrupted users.

**Queries.** $A$ can issue the following queries.

- **Revocation.** $A$ can request the revocation of users $ID_{i_1}, \ldots, ID_{i_{k+1}}$ for some constant $k_{t+1} \in [1, N]$. $C$ runs $\mathcal{T}_{t+1} \leftarrow \text{Revoke}(msk, \{ID_{i_1}, \ldots, ID_{i_{k+1}}, \mathcal{T}_t\})$ and adds $ID_{i_1}, \ldots, ID_{i_{k+1}}$ to $RU_{t+1}$.

- **GSign.** $A$ can request a group signature on a message $M$ for a user $U_i$ where $ID_i \notin CU$. $C$ sends a message $M^*$ and two users $U_{i_1}$ and $U_{i_2}$, where $ID_{i_1}, ID_{i_2} \notin CU$. $C$ chooses a bit $b \leftarrow \{0, 1\}$ and runs $\sigma^* \leftarrow \text{GSign}(gpk, usk_i, M^*, \mathcal{T}_{t+1})$, where $\mathcal{T}_{t+1}$ is the current revocation-dependent value and gives $\sigma^*$ to $A$.

- **Corruption.** $A$ can request the secret key of a user $U_i$. $C$ adds $ID_i$ to $CU$ and gives $usk_i$ to $A$.

- **Opening.** $A$ can request to a group signature $\sigma$ on a message $M$. $C$ returns the result of $\text{Open}(msk, M, \sigma)$ to $A$.

**Output.** $A$ outputs a past interval $t^* \leq t$ for the current interval $t$, and $(M^*, \sigma^*)$.

We say that $A$ wins if (1) $\land (2) \land ((3) \lor (4))$ holds, where

- (1) $\text{Verify}(gpk, M^*, \sigma^*, \mathcal{T}_{t^*}) = 1$,
- (2) $\sigma^*$ did not obtain $\sigma^*$ by making a signing query at $M^*$, (3) for $ID_i \rightarrow \text{Open}(msk, M^*, \sigma^*), \text{ID}_i \notin CU$, (4) for $ID_i \rightarrow \text{Open}(msk, M^*, \sigma^*), \text{ID}_i \in RU_{t^*}$.

We say that traceability holds if, for all PPT adversaries $A$, the advantage

$$\text{Adv}^{\text{trace}}_{A}(\kappa) := \Pr[A \text{ wins}]$$

is negligible.
4. Proposed Group Signature Scheme with Hiding of the Number of Revoked Users

In this section, we propose an \( r \)-hiding revocable group signature scheme by applying ABGS. Before explaining our scheme, we introduce ABGS as follows.

**Attribute-Based Group Signature (ABGS).** ABGS [22–25, 58] is a kind of group signature, where a user with a set of attributes can prove anonymously whether he/she has these attributes or not. Anonymity means a verifier cannot identify who the actual signer is among group members. As a difference from attribute-based signature (ABS) [59–68], there is an opening manager (as in group signatures) who can identify the actual signer (anonymity revocation), and a verifier can “explicitly” verify whether a user has these attributes or not [22, 24, 25]. By applying this explicit attribute verification, anonymous survey for the collection of attribute statistics is proposed [22]. As one exception, the Fujii et al. ABGS [23] achieves signer-attribute privacy (as in ABS), where a group signature does not leak which attributes were used to generate it, except that assigned attributes satisfy a predicate. As another property (applied in our construction), the dynamic property has been proposed in [22], where the attribute predicate can be updated without reissuing the user’s secret keys.

**Our Methodology.** We consider two attributes: (1) valid group user and (2) the user’s identity (say \( U \)). We apply the dynamic property of ABGS [22] and the signer-attribute privacy of ABGS [23]. Here we explain our methodology. Let the initial access tree be represented as in Figure 1.

Due to the signer-attribute privacy, a user \( U \) can anonymously prove that he/she has attributes “valid group user” and “\( U \).” Namely, anyone can verify whether the signer’s attributes satisfy the access tree, without detecting the actual attribute (i.e., the user’s identity).

When a user (say \( U \)) is revoked, the tree structure is changed as in Figure 2.

Due to the dynamic property of ABGS, this modification can be done without reissuing the user’s secret keys. By removing the attribute “valid group user” from the subtree of \( U \), we can express the revocation of \( U \), since \( U \) cannot prove that his/her attributes satisfy the current access tree.

In addition, we propose randomization and dummy attribute techniques to implement the revocation procedure (Figure 3). We apply the Boldyreva multisignature scheme [26], since it is applied for the composition of the membership certificate in the Fujii et al. ABGS. Let \( r \) be the time interval and let \( v \) denote the attribute “valid group user.”

For a nonrevoked user \( U \), GM publishes the dummy value \( g_1^{x_1} \). Then \( U \) can compute \( g_i^{x_i} \). If \( U \) cannot compute \( g_i^{x_i} \) due to the DDH assumption, the DDH assumption holds in \( G_1 \). So, no one can decide whether \( U \) is a revoked user or not by observing either \( (g_i^{x_i}, g_i^{x_i'}) \) or \( (g_i^{x_i}, g_i^{x_i'}) \). This is our main idea for preventing revealing the number of revoked users \( r \).

Next, we give our group signature scheme.

**Construction 1** (proposed \( r \)-hiding revocable group signature scheme).

**Setup** (1). Select a bilinear group \((G_1, G_2, G_T)\) with prime order \( p \) and a bilinear map \( e : G_1 \times G_2 \rightarrow G_T \). Let \( \gamma_1, \gamma_2, \ldots, \gamma_n \leftarrow G_1 \), \( \gamma \leftarrow G_2 \). Output params = \((p, G_1, G_2, G_T, e, \gamma_1, \gamma_2, \ldots, \gamma_n, \gamma, \mathcal{H})\), where \( \mathcal{H} : \{0, 1\}^* \rightarrow Z_p \) is a hash function modeled as a random oracle.

**KeyGen** (params, \( N \)). Let \((U_1, \ldots, U_N)\) be all users. Set \( t = 0 \).

Select \( \omega_1, \omega_2, X_1, \ldots, X_N, s_1, \ldots, s_N \leftarrow Z_p^* \).

(i) Compute \( u, v, h \in G_2 \) with the condition \( u^X = v^X = h \) (note that \( (u, v, h) \) is a public key of the linear encryption and \((X_1, X_2)\) is the corresponding secret key),

(ii) \( K_{i,1} = g_i^{h \omega_i} \), \( K_{i,2} = h^X_i \), and \( B_i = g_i^{x_i} \) for all \( i \in [1, N] \).

(iii) \( \Omega_1 = h^\omega_1 \) and \( \Omega_2 = h^\omega_2 \).

For all \( i \in [1, N] \), choose \( s_{\tau,0,j}, y_j, r_0,j \leftarrow Z_p^* \). If \( s_{\tau,0,j} + s_j \equiv 0 \mod p \), then choose \( s_{\tau,0,j} \) again until \( s_{\tau,0,j} + s_j \not\equiv 0 \mod p \) holds. Set \( s_{T,0,j} = s_{\tau,0,j} + s_j \) and compute

(i) \( h_{T,0,j} = g_i^{s_{T,0,j}} \).
Next, we explain the relations proved in SPK V which proves the following three things.

(1) A signer has a valid \((K_{i,1}, K_{i,2})\) generated by the KeyGen algorithm.

\[ e(K_{i,1}, \Omega_1 K_{i,2}) = e(g_1, h). \] (4)

(2) A signer has not been revoked.

(i) A nonrevoked signer can compute \(H_i = h^{r_2 r_6 + r_7}\) and prove that

\[ C_1 = g^{s_{T,0,i}} \land C_2 = g^{s_{T,0,i}} \land C_2 = C_1^{-T_2} g^{y}. \] (6)

Note that \(C_2 = C_1^{-T_2} g^{y} = (g^{s_{T,0,i}})^{-T_2} g^{y} = g^{s_{T,0,i} - T_2 r_7} g^{y} \) yields \(H_i = -r_1 r_2 \) and \(y = r_2 r_6 + r_7\).

(3) A nonrevoked user can compute \(H_i = h^{r_2 r_6 + r_7}\) from \(B_i\) and \(d_{T,J}\), where \(s_{T,J}\) is a signed message of \(A_{i,J}\). These satisfy the following relations:

\[ e(h_{T,J}, K_{i,2}) = e(H_i, h), \]
\[ e(A_{i,J}, \Omega_i h) = e(g_1^{s_{T,0,i}} g_2^{s_{T,0,i}} g_4, h). \] (7)
(ii) Since $H_t, h_{T, j}$, and $A_i$ are hidden such that $T_S = H_t g^S$, $T_4 = h_{T, j} g^S$, and $T_5 = A_i g^S$, these relations are represented as
\[
e(T_4, T_2) = \frac{e(T_4, T_2)^\delta e(T_4, h) e(g, h)^\delta}{e(T_3, h)}.
\]

\[
e(T_2, \Omega_2) = \frac{e(g, h) e(g_2, h) e(g_3, h)}{e(T_4, h)}.
\]

(iii) We need to guarantee the relations $\beta = -r_2 r_4$ and $\beta' = -s_5 y_{i,j} - r_4$ in the relations above. To prove these, introduce intermediate values $r' = r_2 r_8 + r_9$ and $\gamma' = r_10 y_{i,j}$ and prove that
\[
C_3 = g^2 g^3 \land C_4 = g^\beta g^S \land C_4 = C_3^\gamma g^\gamma,
\]
\[
C_5 = g^{\beta'} g^3 \land C_6 = g^\gamma g^{\gamma'} \land C_6 = C_5^\gamma g^\gamma.
\]

(iv) As in $\alpha$ and $\gamma$ explained before, relations $\beta = -r_2 r_4$, $\beta' = -s_5 y_{i,j} - r_4$, $\gamma' = r_2 r_8 + r_9$, and $\gamma'' = r_10 y_{i,j}$ are obtained from the relations above.

(v) Note that $(A_{i, j}, r_{i, j}, y_{i, j})$ is a BBS+ signature for signed messages $(s, T, i)$, and therefore $V$ depends on the current time interval $t$.

(3) A value for the Open algorithm is included in $\sigma$.

(i) $(F_1, F_2, F_3)$ is a ciphertext (of the linear encryption scheme) of the plaintext $K_{j, i}$, which can be computed by decrypting $(F_1, F_2, F_3)$ using msk.

According to the above explanations, compute the SPK $V$ proving the following relations (we give the detailed SPK in the Appendix):
\[
V = \text{SPK} \{ r_{i, j}, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9, r_{10}, y_{i, j}, \alpha, \beta, \beta', \gamma, \gamma', \delta_1, \delta_2 \}
\]
\[
e(T_4, \Omega_2) = \frac{e(g, T_4)^\delta e(T_4, h) e(g, h)^\delta}{e(T_3, h)}.
\]

\[
e(T_2, \Omega_2) = \frac{e(g, h) e(g_2, h) e(g_3, h)}{e(T_4, h)}.
\]

Output $\sigma = (C_1, C_2, C_3, C_4, C_5, C_6, F_1, F_2, F_3, T_1, T_2, T_3, T_4, T_5, V)$.

Verify($g_{pk}, M, \sigma, \mathcal{T}_j$). Return 1 if $\sigma$ is a valid group signature and 0 otherwise. We give the procedure of the verification algorithm in the Appendix.

Revoke($g_{pk}, msk, [U_i], \mathcal{T}_j$). Let $RL_{t+1} := \{U_i\}$ be a set of revoked users. Let $t \rightarrow t + 1$. For all $i \in [1, N]$, choose $s_{t+1, i,j} \leftarrow Z_p$ (until $s_{t+1, i,j} \neq s_{t, i,j} \pmod{p}$), and set $s_{t+1, i,j} := s_{t+1, i,j} + s_{t, i,j}$ and compute
\[
h_{T, t+1, i} = g_{t+1}^i,
\]
\[
A_{t+1, i} = (g_{t+1}^i g_t^i g_{t+1}^i g_{t+1}^i)^{1/(\omega_2 y_{t+1, i})},
\]
and compute $d_{T, t+1, i}$ such that
\[
d_{T, t+1, i} = \begin{cases} \text{Sign}(s_{T, t+1, i}, i) = (A_{t+1, i}, h_{T, t+1, i}, d_{T, t+1, i}) & (U_i \in RL_{t+1}) \\ g_{t+1}^i & (U_i \notin RL_{t+1}) \end{cases},
\]
and set Sign$(s_{T, t+1, i}, i) = (A_{t+1, i}, h_{T, t+1, i}, d_{T, t+1, i})$. Output $\mathcal{T}_{t+1} = \{\text{Sign}(s_{T, t+1, i}, i) \mid h_{T, t+1, i}, d_{T, t+1, i} \}$.

Open($g_{pk}, msk, M, \sigma$). Compute $F_3 = F_2^\Delta K_{i, j}$ and search $i$ such that $(K_{j, i}, i) \in \text{reg}$ and $K = K_{j, i}$. If there is no such $i$, output 0. Otherwise, output $i$.

In our scheme, no public values have size dependent on $r$ and no costs of the GSign algorithm (or the Verify algorithm) depend on $r$ or $N$. In addition, our scheme satisfies anonymity with respect to the revocation which guarantees the unlinkability of revoked users. So, in our scheme, no $r$ is revealed.

5. Security Analysis

Theorem 11. The proposed group signature scheme satisfies anonymity with respect to the revocation under the DLIN assumption and the XDH assumption in the random oracle model.

Proof. This proof contains three games, Games 0, 1, and 2. Game 0 is the same as anonymity game. In Game 1, all $d_{T, t+1, i}$ where $i \in [1, N]$ is randomly chosen. Let $A'$ be the adversary who breaks anonymity with respect to the revocation of our
scheme and let \( B_1 \) be the simulator. First, \( B_1 \) chooses all values and sets up the scheme as in Game 0, except all \( d_i, r_i, s_i \), where \( i \in [1, N] \) are randomly chosen. Note that still \( B_1 \) can answer all queries issued from \( A_1 \) since \( B_1 \) knows all secret values and can compute \( h_i^{d_i, r_i} \) for all \( i \in [1, N] \). Under the XDH assumption, Game 0 and Game 1 are identical.

Game 2 is the same as Game 1, except that the challenge group signature contains the challenge ciphertext of the linear encryption. In Game 2, let \( C_2 \) be the challenger of the linear encryption and let \( A_2 \) be the adversary who breaks anonymity with respect to the revocation of our scheme. We construct algorithm \( B_2 \) that breaks the IND-CPA security of the linear encryption. First, \( C_2 \) gives the public key of the linear encryption \((u, v, h)\). \( B_2 \) chooses all values, except for \((u, v, h)\), and therefore \( B_2 \) can answer all queries issued from \( A_2 \). In the challenge phase, \( A_2 \) sends \((M^*, U_i, U_{i_1})\). Let \( h^{u_i} \) and \( h^{v_i} \) be (a part of) secret key of \( U_i \) and \( U_{i_1} \), respectively. \( B_2 \) sets \( M_i^* := h^{u_i} \) and \( M_{i_1}^* := h^{v_i} \) and sends \((M_i^*, M_{i_1}^*)\) to \( C_2 \) as the challenge messages of the linear encryption. \( C_2 \) sends the challenge ciphertext \( C^* \). \( B_2 \) sets \( C^* = (F_1, F_2, F_3) \) and computes the challenge group signature \( \sigma^* \). Note that \( B_2 \) does not know the random number \((r_1, r_2, \mu) \in \{0, 1\}^3 \) such that \( C^* = (h^{u_i}h^{v_i}z_i^{r_1}z_i^{r_2}, v_i^{r_1}, v_i^{r_2}) \), since \((r_1, r_2, \mu) \) are chosen by \( C_2 \). So, \( B_2 \) uses the bypass of the random oracle \( H \) for computing \( \sigma^* \) and includes \( C^* \) in \( \sigma^* \). Then, all values (except for \( C^* \)) are independent of \( \mu \). Note that even if \( U_i \) is revoked in the challenge interval, \( B_2 \) can compute \( \sigma^* \), since \( B_2 \) knows \( \text{msk} \). Finally, \( A_2 \) outputs the guessing bit \( \mu' \in \{0, 1\} \). \( B_2 \) outputs \( \mu' \) as the guessing bit of the IND-CPA game of the linear encryption.

**Theorem 12.** The proposed group signature scheme satisfies traceability under the N-HSDH assumption, the CDH assumption, and the NSDH-gamma assumption, where \( t \) is the final time interval that \( A_2 \) outputs \((M^*, \sigma^*)\).

**Proof.** From the winning conditions of traceability, that is, either \( ID_3 \notin CU \) or \( ID_3 \in RU_3 \), an adversary is divided into the following three types: \( A_1, A_2, \) and \( A_3 \), as follows: let \( A_1 \) be an adversary who outputs \((M^*, \sigma^*)\), where for \( ID_3 \) \( \leftarrow \text{Open}(\text{msk}, M^*, \sigma^*) \), \( ID_3 \notin CU \) and \( U_i \notin \{U_{i_1}, \ldots, U_N\} \) hold. Let \( A_2 \) be an adversary who outputs \((M^*, \sigma^*)\), where for \( ID_3 \) \( \leftarrow \text{Open}(\text{msk}, M^*, \sigma^*) \), \( ID_3 \notin CU \) and \( U_i \notin \{U_{i_1}, \ldots, U_N\} \) hold. In addition, let \( A_3 \) be an adversary who outputs \((M^*, \sigma^*)\), where for \( ID_3 \) \( \leftarrow \text{Open}(\text{msk}, M^*, \sigma^*) \), \( ID_3 \in RU_3 \) holds (note that since \( ID_3 \) \( \notin RU_3 \), \( U_i \notin \{U_{i_1}, \ldots, U_N\} \) holds).

We construct an algorithm \( B_3 \) (resp. \( B_1 \) and \( B_2 \) ) that breaks the N-HSDH assumption (resp. \( \gamma \)-SDH assumption, where \( \gamma \) is the number of signing queries, and the CDH assumption) by using \( A_1 \) (resp. \( A_2 \) and \( A_3 \) ).

First, we describe \( B_1 \). Let \( g_1, h, h^{x_1}, \{g_1^{(1/(\omega_i+r_i))}, h^{x_i}\} =_{\text{U}} \{1, 2, \ldots, N\} \) be an N-HSDH instance. \( B_1 \) selects \( U_i \in \{U_{i_1}, \ldots, U_N\} \), and choose all values, except for \( g_1, h_i \) and \( \Omega_i := h^{x_i} \). \( B_1 \) answers queries issued by \( A_1 \) as follows.

Revocation. \( \mathcal{A}_3 \) requests the revocation of users \( ID_{i_1}, \ldots, ID_{i_k} \) for some constant \( k \notin [1, N] \). Since \( B_1 \) knows \( \omega_2, B_3 \) adds \( ID_{i_1}, \ldots, ID_{i_k} \) to \( RU_1 \) and simply returns the result of the Revocation algorithm.

GSigning. \( \mathcal{A}_3 \) requests a group signature on a message \( M \) for a user \( U_i \) where \( \mathcal{A}_3 \notin CU \). \( B_3 \) computes \( \sigma \) by using the bypass of the random oracle \( H \) and gives \( \sigma \) to \( \mathcal{A}_3 \).

Corruption. \( \mathcal{A}_3 \) requests the secret key of a user \( U_i \). If \( U_i = U_{i_1} \), then \( B_3 \) aborts. Otherwise, \( B_3 \) adds \( ID_i \) to \( RU \) and gives \((K_{i_1}, K_{i_2}, B_i) \) to \( \mathcal{A}_3 \).
Opening. Since \( S_3 \) has \((X_1, X_2)\), \( S_3 \) simply returns the result of the Open algorithm.

Finally, \( S_4 \) outputs a past interval \( t^* \leq t \) for the current interval \( t \) and a pair \((M^*, \sigma^*)\). By using the extractor of SPK, \( S_4 \) gets \( H^* \), where \( e(K_{11}, \Omega, K_{12}^*) = e(g_1, h), e(h_{T_3}, K_{22}^*) = e(H^*, h) \), \( F_1 = K_{12}^* h^* s_1 \), \( F_2 = v \beta^1 \), and \( F_3 = v \beta^2 \). Hold.

From \( (F_1, F_2, F_3) \), \( S_3 \) obtains \( i \) by using the Open algorithm. If \( i \neq t^* \), then \( S_3 \) aborts. Otherwise, \( S_3 \) solves the CDH problem as follows. Since \( U_1 \in RL_4, S_3 \) has computed \( g_1^{s_2^i} \cdot g_1^{h_{T_3}^i} \cdot g_1^{s_{1,2}^i} \). That is, \( H^*_i = B_{r_i} \cdot g_1^{h_{T_3}^i x_i} = g_1^{s_{1,2}^i x_i} \) holds. So, \( S_3 \) outputs \( H^*_i / (g_1^{T_3 x_i}) = g_1^{s_2^i} \) as the solution of the CDH problem.

\[ \square \]

6. Discussion: Toward Efficient and Standard Model Construction

One drawback of our scheme is that the number of public values depends on \( N \), since no common attribute can be implemented for determining the revocation procedure of "each" user. So, one may think that there might be a more trivial construction (without applying ABGS) if such a big-size public value is allowed. For example, as one of the most simple group signature constructions, let \( g^1, \ldots, g^N \) be users’ public keys, and GM randomizes these values such that \( y_1 := (g^1)^{y_1} \), \( \ldots, y_N := (g^N)^{y_N} \), and publishes \( y := g^0 \). Each user (say \( U_i \)) proves the knowledge of \( x_i \); the relation \((g^i)^{y_i} = y_i \) holds. If a user (say \( U_j \)) is revoked, then GM publishes a random value \( R_j \) (instead of \((g^j)^{y_j} \)). If no revoked users are not revealed under the DDH assumption, \( (g, g^1, g^N, (g^i)^{y_i}) \) is a DDH tuple. However, this trivial approach requires \( N \)-dependent signing/verification cost, whereas our scheme achieves constant proving costs.

As another candidate, Sudarsono et al. [69] proposed an attribute-based anonymous credential system by applying an efficient pairing-based accumulator proposed by Camenisch et al. [70]. Since the Sudarsono et al. construction follows AND/OR relations of attributes, a revocable group signature scheme with the property of hiding \( r \) might be constructed. However, it is not obvious whether 2-DNF (disjunctive normal form) formulae are (valid group user) \( U_i \) can be implemented or not in the Sudarsono et al. attribute-based proof system. Later, Begum et al. [71] proposed an attribute-based anonymous credential system for CNF (conjunctive normal form) formulae which can be converted to DNF formulae. However, these constructions require the \( N \)-dependent-size \( (N \) is the number of attributes in this context) public values to update the witness of users as in our group signature scheme. So, we insist that proposing a revocable group signature scheme with both the property of hiding \( r \) and constant proving costs is not trivial even if such a large-size public key is allowed.

Libert et al. [72] proposed anonymous broadcast encryption, where information of a set of authorized users \( S \) (indicated by a user who encrypts a plaintext \( M \)) is not revealed from a ciphertext, except for the size of \( S \). More precisely, they first considered a scheme, where there are \( N \) public keys \( (N \) is the total number of users in this context) and the user encrypts \( M \) (resp., 0) by using the corresponding user’s public key if a user belongs to \( S \) (resp., does not belong to \( S \)), using key-private public key encryption [73]. Though the size of ciphertext is \( O(N) \), no information of \( S \), including its size, is revealed. In order to improve the size of ciphertext of their first scheme, Abdalla et al. applied robust broadcast encryption [74] and constructed an anonymous broadcast encryption scheme whose ciphertext size is \( O(|S|) \). In other words, they can reduce the ciphertext size at the expense of information of the size of \( S \). In r-hiding group signature, the size of the set of revoked user needs to be hidden. To do so, we essentially use the same methodology of the Libert et al.’s first scheme, that is, adding dummies. Though an anonymous broadcast encryption scheme with sublinear size ciphertext has been proposed [75], this scheme only achieves outsider anonymity, where the receiver’s identities are hidden from outsiders; this security notion seems not enough for our purpose. Therefore, if an efficient anonymous broadcast encryption scheme which hides the size of \( S \) can be constructed, then we might construct more efficient \( r \)-hiding group signature scheme by using the methodology of recent group signature schemes [7, 8], where nonrevocable users prove the decryption ability of a ciphertext of a broadcast encryption scheme.

Another (theoretical) drawback of our construction is that the size of public keys is \( \Omega(|S|) \) using random oracles. Actually, a typical group signature construction methodology for standard model construction has been appearing in several papers after the Groth construction [33], that is, using Groth-Sahai proofs [36] and structure-preserving signatures [37]. By applying this methodology, we can expect that an \( r \)-hiding revocable group signature scheme in the standard model can be constructed. However, big-size public values problem still remains. So, we need to propose a novel methodology for proposing an \( r \)-hiding revocable group signature scheme with small-size public key, even in the random oracle model.

Under a XDH-hard elliptic curve with 170-bit \( p \) (as in [4, 10]), the size of signature is 7242 bits, where the size of an element of \( G_1 \) is 171 bits, the size of an element of \( G_2 \) is 513 bits, and the size of the challenge \( c \) is 80 bits. Since the size of signature in [4] (resp., in [10]) is 1444 (resp., 1533) bits, there might be space for improvement of the signature size.

7. Conclusion

In this paper, for the first time we pointed out that the number of revoked users \( r \) is quite negative information in group signature, and we propose a revocable group signature scheme with the property of hiding \( r \), by applying ABGS. As a matter of first priority, proposing such a group signature scheme with small-size public parameter is an interesting future work. Then, we may be able to apply the standard model construction methodology for constructing an efficient scheme in the standard model.
Here, we describe the detailed SPK V as follows.

(1) Choose
\[ r_{r_1}, r_{r_2}, r_{r_3}, r_{r_4}, r_{r_5}, r_{r_6}, r_{r_7}, r_{r_8}, r_{r_9}, r_{r_{10}}, r_{r_{11}}, r_{r_{12}}, r_{r_{13}}, r_{r_{14}}, r_{r_{15}} = \frac{s}{Z_p}. \]

(2) Compute
\[ R_1 = e(\tilde{g}, \tilde{\Omega}_1 T_2) e(T_1, \tilde{h})^t e(\tilde{g}, \tilde{h})^s, \]
\[ R_2 = \frac{e(\tilde{g}, T_2)^t e(T_4, \tilde{h}) e(\tilde{g}, \tilde{h})^s}{e(\tilde{g}, h)^s}, \]
\[ R_3 = \frac{e(\tilde{g}, \tilde{\Omega}_2) e(g_3, h)^{t_1} e(\tilde{g}, \tilde{h})^{s_2}}{e(T_5, h)^{s_3}}, \]
\[ R_4 = e(\tilde{g}, \tilde{\Omega}_1) e(g_3, h)^{t_1} e(\tilde{g}, \tilde{h})^{s_2}, \]
\[ R_5 = \frac{e(\tilde{g}, T_2)^t e(T_4, \tilde{h}) e(\tilde{g}, \tilde{h})^s}{e(\tilde{g}, h)^s}, \]
\[ R_6 = C_1 t_1 e(\tilde{g}, \tilde{h})^s, \]
\[ R_7 = g^r_2 \tilde{g}^{r_2}, \]
\[ R_8 = g^r_2 \tilde{g}^{r_2}, \]
\[ R_9 = C_3 t_3 e(\tilde{g}, \tilde{h})^s, \]
\[ R_{10} = g^r_2 \tilde{g}^{r_2} \tilde{g}^{r_3}, \]
\[ R_{11} = e(\tilde{g}, \tilde{h})^{s_2}, \]
\[ R_{12} = e(\tilde{g}, \tilde{h})^{s_3}, \]
\[ R_{13} = \frac{\tilde{h}^r}{e(\tilde{g}, h)^r}, \]
\[ R_{14} = u^s, \]
\[ R_{15} = y^s. \]
\[ c = \mathcal{H}(gpk, M, \{C_i\}_{i=1}^{10}, \{T_i\}_{i=1}^{5}, \{R_i\}_{i=1}^{15}), \]
\[ s_{r_1} = r_{r_1} + c r_{r_1} \]
\[ s_{r_{12}} = r_{r_{12}} + c r_{r_{12}}, \]
\[ s_\alpha = r_\alpha + c \alpha, \]
\[ s_\beta = r_\beta + c \beta, \]
\[ s_\gamma = r_\gamma + c \gamma, \]
\[ s_\delta = r_\delta + c \delta, \]
\[ s_{\tilde{h}} = e(T_1, \tilde{h})^{s_3}, \]
\[ s_{\tilde{h}} = e(T_5, h)^{s_3}, \]
\[ s_{\tilde{h}} = e(T_1, \tilde{h})^{s_3}, \]
\[ s_{\tilde{h}} = e(T_5, h)^{s_3}. \]

Note that a verifier computes \( e(g_2, h)^s \) to check whether \( \sigma \) is made in the time interval \( t \) or not.

(2) Check
\[ c = \mathcal{H}(gpk, M, \{C_i\}_{i=1}^{10}, \{T_i\}_{i=1}^{5}, \{R_i\}_{i=1}^{15}). \]
\[ S = V = \left( C_i, \{s_{r_1}\}_{i=1}^{10}, \{s_{\tilde{h}}\}, \{s_\alpha\}, \{s_\beta\}, \{s_\gamma\}, \{s_\delta\} \right). \]

So, the final form of a group signature is described as
\[ \sigma = (C_1, C_2, C_3, C_4, C_5, C_6, F_1, F_2, F_3, T_1, T_2, T_3, T_4, T_5, \mathcal{H}(gpk, M, \{C_i\}_{i=1}^{10}, \{T_i\}_{i=1}^{5}, \{R_i\}_{i=1}^{15}), \{s_{r_1}\}_{i=1}^{10}, \{s_{\tilde{h}}\}, \{s_\alpha\}, \{s_\beta\}, \{s_\gamma\}, \{s_\delta\}). \]

Next, we give the detailed verification procedure. Let
\[ \sigma = (C_1, C_2, C_3, C_4, C_5, C_6, F_1, F_2, F_3, T_1, T_2, T_3, T_4, T_5, \mathcal{H}(gpk, M, \{C_i\}_{i=1}^{10}, \{T_i\}_{i=1}^{5}, \{s_{r_1}\}_{i=1}^{10}, \{s_{\tilde{h}}\}, \{s_\alpha\}, \{s_\beta\}, \{s_\gamma\}, \{s_\delta\}). \]

(1) Compute
\[ R_1 = e(\tilde{g}, \tilde{\Omega}_1 T_2)^{t_1} e(T_1, \tilde{h})^{t_2} e(\tilde{g}, \tilde{h})^{s_2} \left( \frac{e(T_1, \tilde{\Omega}_1 T_2)}{e(\tilde{g}, \tilde{h})} \right)^{c} \]
\[ R_2 = e(\tilde{g}, T_2)^{t_2} e(T_4, \tilde{h})^{t_3} e(\tilde{g}, \tilde{h})^{s_3} \left( \frac{e(T_4, T_2)}{e(T_3, h)} \right)^{c}, \]
\[ R_3 = e(\tilde{g}, \tilde{\Omega}_2) e(g_3, h)^{t_1} e(\tilde{g}, \tilde{h})^{s_2} \left( \frac{e(T_5, \tilde{\Omega}_2)}{e(g_3, h)} \left( \frac{e(T_5, h)}{e(T_4, h) e(g_2, h)} \right)^{c} \right), \]
\[ R_4 = e(g_2, g_1)^{c}, \]
\[ R_5 = e(\tilde{g}, \tilde{\Omega}_1) e(g_3, h)^{t_1} e(\tilde{g}, \tilde{h})^{s_2}, \]
\[ R_6 = e(g_2, g_1)^{c}, \]
\[ R_7 = e(g_2, g_1)^{c}, \]
\[ R_8 = e(g_2, g_1)^{c}, \]
\[ R_9 = e(g_2, g_1)^{c}, \]
\[ R_{10} = e(g_2, g_1)^{c}, \]
\[ R_{11} = e(g_2, g_1)^{c}, \]
\[ R_{12} = e(g_2, g_1)^{c}, \]
\[ R_{13} = e(g_2, g_1)^{c}, \]
\[ R_{14} = e(g_2, g_1)^{c}, \]
\[ R_{15} = e(g_2, g_1)^{c}. \]

(2) Check \( c = \mathcal{H}(gpk, M, C_1, C_2, C_3, C_4, C_5, C_6, F_1, F_2, F_3, T_1, T_2, T_3, T_4, T_5, \mathcal{R}_{11}, \ldots, \mathcal{R}_{15}). \) If it holds, then output 1, and 0 otherwise.

**Disclosure**

A preliminary version of this paper appears in the 14th International Conference on Information Security and Cryptology, ICISC 2011 [76]. This is the full version. We add estimations of recent revocable group signature schemes [7, 8] that appeared after publishing our conference version and have added content presented in the 4th International Conference on Provable Security, ProvSec 2010 [19].

**Conflict of Interests**

The authors declare that there is no conflict of interests regarding the publication of this paper.

**References**


