A Revocable Group Signature Scheme with the Property of Hiding the Number of Revoked Users

Keita Emura[1], Atsuko Miyaji[2], and Kazumasa Omote[2]

[1] Center for Highly Dependable Embedded Systems Technology
[2] School of Information Science
Japan Advanced Institute of Science and Technology, 1-1, Asahidai, Nomi, Ishikawa, 923-1292, Japan
{k-emura,miyaji,omote}@jaist.ac.jp

Abstract. 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 to hide r. In this paper, we propose the first revocable group signature scheme, where r is kept hidden. To handle these properties, we newly define the security notion called anonymity w.r.t. the revocation which guarantees the unlinkability of revoked users.
Keywords: Group signature, Revocation, Hiding the Number 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, e.g., 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.

Many cryptographic attempts for the revocation of rights of users have been considered so far, especially, in group signature [12][3], anonymity revocation has

[3] The concept of group signature was investigated by Chaum and Heyst [12], and its typical usage is described as follows: The group manager (GM) issues a membership certificate to a signer. A signer makes a group signature by using its own membership certificate. A verifier anonymously verifies whether a signer is a member of a group or not. 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

been introduced [7, 8, 14, 25, 27, 28, 31][4]. 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 a negative information. As a concrete example, we introduce an application of revocable group signature for outsourcing businesses [20]. By applying group signature, the service authentication server (outsourcee) has only to verify whether a user is a legitimate member or not, and does not have to manage the list of identities of users. Therefore, the risk of leaking the list of identities of users can be minimized, and this is the merit of using group signature in identity management. After a certain interval, the service provider charges the users who have already used the service, by using the opening procedure of group signature. When a user would like to leave the group, or when a user have not paid, the service provider revokes this user. In this system, if r is revealed, then one may think that there might be many users who have stopped using the service, i.e., 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.

So, our main target is to propose a revocable group signature scheme with the property of hiding the number of revoked users r. 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 issue the polynomial times queries of the signing oracle). 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 [7, 11, 14, 31] (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. Nakanishi et al. [27] 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. In [8, 13, 25, 28] (which are verifier-local revocation (VLR) type group signature), revoked users are linkable.

verifiers do not have to identify individual signers, group signature is a useful and powerful tool for protecting signers' privacy.

[4] Since a long RSA modulus leads to certain inefficiency factors (e.g., long signatures, heavy complexity costs, and so on), we exclude RSA-based revocable group signatures (e.g., [29, 30]) in this paper.

In this case, anyone can guess r by executing the verification procedure. For the sake of clarity, we introduce the Nakanishi-Funabiki methodology [28] 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 $|RL|$. 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 σ (made by U_j) contains $f^{x_j+\beta}$ and h^{β} for some random β and some group elements f and h. If U_j has been revoked, then there exists h^{x_i} such that $e(f^{x_j+\beta}, h) = e(h^{x_i}h^{\beta}, f)$ 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., $(h^{x_i})^{r_i}$ for some random r_i), 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 [16], 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 group signature any longer. Next, as another methodology, consider the multi group signature [1] 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 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 group signature scheme with the property of hiding the number of revoked users r, by applying attribute-based group signature (ABGS) [15, 18, 21, 22]. 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 w.r.t. the revocation. As the main difference among our anonymity definition and previous ones, to guarantee the unlinkability of revoked users, $\mathcal{A}$ can issue the revocation queries against the challenge users.

2 Bilinear Groups and Complexity Assumptions

Definition 1 (Bilinear Groups). *Let $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ be cyclic groups with a prime order p, and $\mathbb{G}_1 = \langle g \rangle$ and $\mathbb{G}_2 = \langle h \rangle$. Let $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be an (efficient computable) bilinear map with the following properties: (1) bilinearity: for all $(g, g') \in \mathbb{G}_1^2$ and $(h, h') \in \mathbb{G}_2^2$, $e(gg', h) = e(g, h)e(g', h)$ and $e(g, hh') = e(g, h)e(g, h')$ hold, and (2) non-degeneracy : $e(g, h) \neq 1_T$, where 1_T is the unit element over $\mathbb{G}_T$.*

Definition 2 (The Computational Diffie-Hellman (CDH) assumption). *We say that the CDH assumption holds if for all probabilistic polynomial time (PPT) adversary $\mathcal{A}$, $\Pr[\mathcal{A}(g_1, g_1^a, g_1^b) = g_1^{ab}]$ is negligible, where $g_1 \in \mathbb{G}_1$ and $(a, b) \in \mathbb{Z}_p^2$.*

Definition 3 (The Decision Diffie-Hellman (DDH) assumption). *We say that the DDH assumption holds if for all PPT adversary $\mathcal{A}$, $|\Pr[\mathcal{A}(g_1, g_1', g_1^x, {g_1'}^x) = 0] - \Pr[\mathcal{A}(g_1, g_1', g_1^x, {g_1'}^r) = 0]|$ is negligible, where $(g_1, g_1') \in \mathbb{G}_1^2$ and $(x, r) \in \mathbb{Z}_p^2$ with $x \neq r$.*

Definition 4 (The Decision Linear (DLIN) assumption [7]). *We say that the DLIN assumption holds if for all PPT adversary $\mathcal{A}$, $|\Pr[\mathcal{A}(u, v, h, u^a, v^b, h^{a+b}) = 0] - \Pr[\mathcal{A}(u, v, h, u^a, v^b, \eta) = 0]|$ is negligible, where $(u, v, h, \eta) \in \mathbb{G}_2^4$ and $(a, b) \in \mathbb{Z}_p^2$.*

Definition 5 (The Hidden Strong Diffie-Hellman (HSDH) assumption [9]). *We say that ℓ-HSDH assumption holds if for all PPT adversary $\mathcal{A}$, $\Pr[\mathcal{A}(g_1, h, h^\omega, (g_1^{\frac{1}{\omega+c_i}}, h^{x_i})_{i=1,\ldots,\ell}) = (g_1^{\frac{1}{\omega+x}}, h^x) \wedge \forall x_i \neq x]$ is negligible, where $(g_1, h) \in \mathbb{G}_1 \times \mathbb{G}_2$ and $(\omega, x, x_1, \ldots, x_\ell) \in \mathbb{Z}_p^{\ell+2}$.*

Definition 6 (The Strong Diffie-Hellman (SDH) assumption [6]). *We say that q-SDH assumption holds if for all PPT adversary $\mathcal{A}$, $\Pr[\mathcal{A}(g_1, h, h^\omega, h^{\omega^2}, \ldots, h^{\omega^q}) = (g_1^{\frac{1}{\omega+x}}, x)]$ is negligible, where $(g_1, h) \in \mathbb{G}_1 \times \mathbb{G}_2$ and $(\omega, x) \in \mathbb{Z}_p^2$.*

Definition 7 (The external Diffie-Hellman (XDH) assumption [14]). *Let $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ be a bilinear group. We say that the XDH assumption holds if for all PPT adversary $\mathcal{A}$, the DDH assumption over $\mathbb{G}_1$ holds.*

3 Definitions of Group Signature

Here, we define the system operations of revocable group signature and security requirements (anonymity w.r.t. the revocation and traceability) by adapting [27]. Note that our definition follows the static group settings [4]. However, we can easily handle the dynamic group settings [3] (and non-frameability) by applying an interactive join algorithm.

Definition 8. *System Operations of Group Signature*

Setup *: This probabilistic setup algorithm takes as input the security parameter 1^κ, and returns public parameters params.*

KeyGen *: This probabilistic key generation algorithm (for GM) takes as input the maximum number of users N and params, and returns the group public key gpk, GM's secret key msk, all user's secret key $\{usk_i\}_{i\in[1,N]}$, and the initial revocation-dependent value $\mathcal{T}_0$.*

GSign *: This probabilistic signing algorithm (for a user U_i) takes as input gpk, usk_i, a signed message M, and a revocation-dependent value (in the period t) $\mathcal{T}_t$, and returns a group signature σ.*

Verify *: This deterministic verification algorithm takes as input gpk, M, σ, and $\mathcal{T}_t$, and returns 1 if σ is a valid group signature, and 0 otherwise.*

Revoke *: This (potentially) probabilistic revocation algorithm takes as input gpk, msk, a set of revoked users $RL_{t+1} = \{U_i\}$, and $\mathcal{T}_t$, and returns $\mathcal{T}_{t+1}$.*

Open *: This deterministic algorithm takes as input msk and a valid pair (M, σ), and returns the identity of the signer of σ 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 non-revoked user in t is $\{U_1, \ldots, U_N\} \setminus RL_t$. Under this setting, boomerang users (who re-join 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, σ) is input to the Open algorithm, then the Open algorithm easily detect this fact by using the Verify algorithm. So, we exclude this case from the definition of the Open algorithm.

Next, we define anonymity w.r.t. the revocation and traceability. As the main difference among our anonymity definition and previous ones, to guarantee the unlinkability of revoked users, $\mathcal{A}$ can issue the revocation queries against the challenge users. Note that we do not handle the CCA-anonymity, where an adversary $\mathcal{A}$ can issue the open queries. So, we just handle the CPA-anonymity [7] only in this paper. However, as mentioned by Boneh et al. [7], the CCA-anonymity can be handled by applying a CCA secure public key encryption for implementing the open algorithm.

Definition 9 (Anonymity w.r.t. the Revocation).

Setup *: The challenger $\mathcal{C}$ runs the* Setup *algorithm and the* KeyGen *algorithm, and obtains $params$, gpk, msk, and all $\{usk_i\}_{i=1}^N$. $\mathcal{C}$ gives $params$ and gpk to $\mathcal{A}$, and sets $t = 0$, $RU_0 = \emptyset$, and $CU = \emptyset$, where RU_0 denotes the (initial) set of ID's of revoked users, and CU denotes the set of ID's of corrupted users.*

Queries *: $\mathcal{A}$ can issue the following queries:*

- Revocation *: $\mathcal{A}$ can request the revocation of users $ID_{i_1}, \ldots, ID_{i_{k_{t+1}}}$ for some constant $k_{t+1} \in [1, N]$. $\mathcal{C}$ uns $\mathcal{T}_{t+1} \leftarrow$ Revoke$(msk, \{ID_{i_1}, \ldots, ID_{i_{k_{t+1}}}\}, \mathcal{T}_t)$ and adds $ID_{i_1}, \ldots, ID_{i_{k_{t+1}}}$ to RU_{t+1}.*
- Signing *: $\mathcal{A}$ can request a group signature on a message M for a user U_i where $ID_i \notin CU$. $\mathcal{C}$ runs $\sigma \leftarrow$ GSign$(gpk, usk_i, M, \mathcal{T}_t)$, where $\mathcal{T}_t$ is the current revocation-dependent value, and gives σ to $\mathcal{A}$.*
- Corruption *: $\mathcal{A}$ can request the secret key of a user U_i. $\mathcal{C}$ adds ID_i to CU, and gives usk_i to $\mathcal{A}$.*

Challenge *: $\mathcal{A}$ sends a message M^* and two users U_{i_0} and U_{i_1}, where $ID_{i_0}, ID_{i_1} \notin CU$. $\mathcal{C}$ chooses a bit $b \leftarrow \{0, 1\}$, and runs $\sigma^* \leftarrow$ GSign$(gpk, usk_{i_b}, M^*, \mathcal{T}_{t^*})$, where $\mathcal{T}_{t^*}$ is the current revocation-dependent value, and gives σ^* to $\mathcal{A}$.*

Queries *: The same as the previous one (Note that no corruption query for the challenge users is allowed).*

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

We say that anonymity holds if for all PPT adversaries $\mathcal{A}$, the advantage

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

is negligible.

There are two types of revocable group signature such that (1) any users can make a valid group signature, and anyone can distinguish whether a signer has been revoked or not [8, 13, 25, 28], or (2) no revoked user can make a valid group signature without breaking traceability [7, 11, 14, 27, 31]. We implicitly require the second type 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 $\mathcal{C}$ (that has msk) can break traceability to compute the challenge group signature σ^* 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 $\mathcal{A}$ can issue the corruption query against the challenge users, as in the Full-Anonymity [4]. 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 been expired). For example, if users are not intentionally revoked (e.g., a user has not paid in the outsourcing businesses example [20]), 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 $\mathcal{C}$ runs the* Setup *algorithm and the* KeyGen *algorithm, and obtains params, gpk, msk, and all $\{usk_i\}_{i=1}^N$. $\mathcal{C}$ gives params and gpk to $\mathcal{A}$, and sets $t = 0$, $RU_0 = \emptyset$, and $CU = \emptyset$, where RU_0 denotes the (initial) set of ID's of revoked users, and CU denotes the set of ID's of corrupted users.*

Queries *: $\mathcal{A}$ can issue the following queries:*

- Revocation *: $\mathcal{A}$ can request the revocation of users $ID_{i_1}, \ldots, ID_{i_{k_{t+1}}}$ for some constant $k_{t+1} \in [1, N]$. $\mathcal{C}$ runs $\mathcal{T}_{t+1} \leftarrow$ Revoke$(msk, \{ID_{i_1}, \ldots, ID_{i_{k_{t+1}}}\}, \mathcal{T}_t)$ and adds $ID_{i_1}, \ldots, ID_{i_{k_{t+1}}}$ to RU_{t+1}.*
- GSigning *: $\mathcal{A}$ can request a group signature on a message M for a user U_i where $ID_i \notin CU$. $\mathcal{C}$ runs $\sigma \leftarrow$ GSign$(gpk, usk_i, M, \mathcal{T}_t)$, where $\mathcal{T}_t$ is the current revocation-dependent value, and gives σ to $\mathcal{A}$.*
- Corruption *: $\mathcal{A}$ can request the secret key of a user U_i. $\mathcal{C}$ adds ID_i to CU, and gives usk_i to $\mathcal{A}$.*
- Opening *: $\mathcal{A}$ can request to a group signature σ on a message M. $\mathcal{C}$ returns the result of* Open(msk, M, σ) *to $\mathcal{A}$.*

Output *: $\mathcal{A}$ outputs a past interval $t^* \leq t$ for the current interval t, and (M^*, σ^*).*

We say that $\mathcal{A}$ wins if $(1) \wedge (2) \wedge ((3) \vee (4))$ holds, where

1. $\mathsf{Verify}(gpk, M^*, \sigma^*, \mathcal{T}_{t^*}) = 1$
2. *$\mathcal{A}$ did not obtain σ^* by making a signing query at M^*.*
3. *for $ID_{i^*} \leftarrow \mathsf{Open}(msk, M^*, \sigma^*)$, $ID_{i^*} \notin CU$*
4. *for $ID_{i^*} \leftarrow \mathsf{Open}(msk, M^*, \sigma^*)$, $ID_{i^*} \in RU_{t^*}$*

We say that traceability holds if for all PPT adversaries $\mathcal{A}$, the advantage

$$\mathsf{Adv}_{\mathcal{A}}^{trace}(1^\kappa) := \Pr[\mathcal{A} \textit{ wins}]$$

is negligible.

4 Other Cryptographic Tools

In this section, we introduce cryptographic tools applied for our construction.

BBS+ signature [2, 7, 19, 27]: Let L be the number of signed messages, and $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ be a bilinear group. Select $g, g_1, \ldots, g_L \xleftarrow{\$} \mathbb{G}_1$, $h \xleftarrow{\$} \mathbb{G}_2$, and $\omega \leftarrow \mathbb{Z}_p$, and compute $\Omega = g^\omega$. The signing key is ω and the verification key is $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, g_1, \ldots, g_{L+1}, h, \Omega)$. For a set of signed messages $(m_1, \ldots, m_L) \in \mathbb{Z}_p^L$, choose $r, y \xleftarrow{\$} \mathbb{Z}_p$, and compute $A = (g_1^{m_1} \cdots g_L^{m_L} g_{L+1}^r g)^{\frac{1}{\omega+y}}$. For a signature (A, r, y), the verification algorithm output 1 if $e(A, \Omega h^y) = e(g_1^{m_1} \cdots g_L^{m_L} g_{L+1}^r, h)$ holds. The BBS+ signature scheme satisfies existential unforgeability against chosen message attack (EUF-CMA)[5] under the q-SDH assumption.

Linear Encryption [7]: A public key is $pk = (u, v, h) \in \mathbb{G}_2$ such that $u^{X_1} = v^{X_2} = h$ for $X_1, X_2 \in \mathbb{Z}_p$. The corresponding secret key is (X_1, X_2). For a plaintext $M \in \mathbb{G}_2$, choose $\delta_1, \delta_2 \xleftarrow{\$} \mathbb{Z}_p$, compute a ciphertext $C = (F_1, F_2, F_3)$, where $F_1 = M \cdot h^{\delta_1, \delta_2}$, $F_2 = u^{\delta_1}$, and $F_3 = v^{\delta_2}$. C can be decrypted as $M = F_1/(F_2^{X_1} F_3^{X_2})$. The linear encryption is IND-CPA secure[6] under the DLIN assumption.

Signature based on proof of knowledge: In our group signature, we apply the conversion of the underlying interactive zero knowledge (ZK) proof into non-interactive ZK (NIZK) proof by applying the Fiat-Shamir heuristic [17]. We describe such converted signature based on proof of knowledge (SPK) as $SPK\{x : (y, x) \in R\}(M)$, where x is the knowledge to be proved, R is a relation

[5] First an adversary $\mathcal{A}$ is given vk from the challenger $\mathcal{C}$. Then $\mathcal{A}$ sends messages to $\mathcal{C}$ and obtains the corresponding signatures. Finally, $\mathcal{A}$ outputs a message/signature pair (M^*, σ^*). We say that $\mathcal{A}$ wins if (M^*, σ^*) is valid and $\mathcal{A}$ has not sent M^* as a signing query. The EUF-CMA security guarantees that the probability $\Pr[\mathcal{A}$ wins$]$ is negligible.

[6] First an adversary $\mathcal{A}$ is given pk from the challenger $\mathcal{C}$. Then $\mathcal{A}$ sends the challenge message (M_0^*, M_1^*) to $\mathcal{C}$, and $\mathcal{C}$ chooses $\mu \xleftarrow{\$} \{0, 1\}$, and computes the challenge ciphertext C^* which is a ciphertext of M_μ^*. $\mathcal{A}$ is given C^*, and outputs a bit μ'. The IND-CPA security guarantees that $|\Pr[\mu = \mu'] - \frac{1}{2}|$ is negligible.

(e.g., $y = g^x$ in the case of the knowledge of the discrete logarithm), and M is a 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.

5 Proposed Group Signature Scheme with Hiding of the Number of Revoked Users

In this section, we propose a group signature scheme hiding the number of revoked users by applying ABGS. Before explaining our scheme, we introduce ABGS as follows:

Attribute-based group signature (ABGS): ABGS [15, 18, 21, 22] 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 [23, 24, 26, 32], 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 [15, 21, 22]. By applying this explicitly attribute verification, anonymous survey for the collection of attribute statistics is proposed [15]. As one exception, the Fujii et al. ABGS scheme [18] achieves signer-attribute privacy, 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 for our construction), the dynamic property has been proposed in [15], where the attribute predicate can be updated without re-issuing the user's secret keys.

Our Methodology: We consider two attributes: (1) valid group user and (2) the user's identity (say U_i), and apply the dynamic property of ABGS [15] and the signer-attribute privacy of ABGS [18]. Here we explain our methodology. Let the initial access tree be represented as in Fig 1:

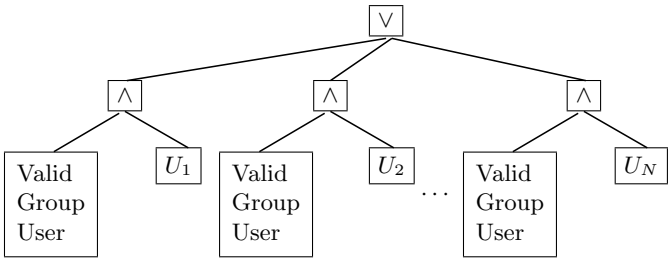

Fig. 1. Initial Access Tree

Due to the signer-attribute privacy, a user U_i can anonymously prove that he/she has attributes "valid group user" and "U_i". 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_1) is revoked, the tree structure is changed as in Fig 2.

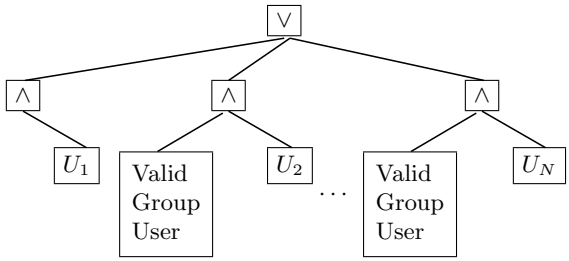

Fig. 2. Modified Access Tree

Due to the dynamic property of ABGS, this modification can be done without re-issuing the user's secret keys. By removing the attribute "valid group user" from the subtree of U_1, we can express the revocation of U_1, since U_1 cannot prove that his/her attributes satisfy the current access tree.

In addition, we propose a randomization and dummy attribute technique to implement the revocation procedure (Fig 3). We apply the Boldyreva multisignature [5], since it is applied for the computation of the membership certificate in the Fujii et al. ABGS. Let t be the time interval and v denote the attribute "valid group user".

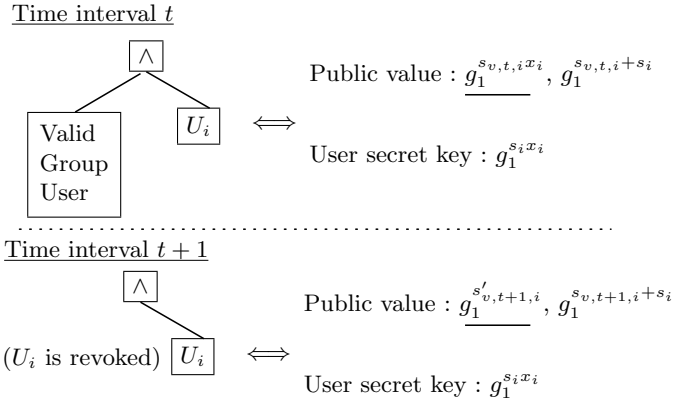

Fig. 3. Our Randomization and Dummy Attribute Technique

For a non-revoked user U_i, GM publishes the dummy value $g_1^{s_{v,t,i}x_i}$. Then U_i can compute $g_1^{(s_{v,t,i}+s_i)x_i} (= H_i)$ from $d_{T,t,i} = g_1^{s_{v,t,i}x_i}$ and U_i's secret key $B_i = g_1^{s_i x_i}$. Let U_i be revoked in the time interval $t+1$. Then, GM publishes a randomized dummy value $g_1^{s'_{v,t+1,i}}$ (instead of $g_1^{s_{v,t+1,i}x_i}$), and therefore U_i cannot compute $g^{(s_{v,t+1,i}+s_i)x_i}$ due to the CDH assumption. Note that $(g_1^{s_{v,t+1,i}+s_i}, g_1^{s_{v,t+1,i}x_i})$ and $(g_1^{s_{v,t+1,i}+s_i}, g_1^{s'_{v,t+1,i}})$ are indistinguishable, under the XDH assumption, where the DDH assumption holds in $\mathbb{G}_1$. Next, we give our group signature scheme.

Protocol 1 (Our revocable group signature).

$\mathsf{Setup}(1^\kappa)$ *: Select a bilinear group* $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ *with prime order* p*, a bilinear map* $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$*,* $g, g_1, \ldots, g_4, \tilde{g} \xleftarrow{\$} \mathbb{G}_1$*,* $\tilde{h} \xleftarrow{\$} \mathbb{G}_2$*. Output params* $= (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, H, g, g_1, g_2, g_3, g_4, \tilde{g}, \tilde{h})$

$\mathsf{KeyGen}(params, N)$ *: Let* $(U_1, \ldots, U_N)$ *be all users. Set* $t = 0$*. Select* $\omega_1, \omega_2, X_1, X_2, x_1, \ldots, x_N, s_1, \ldots, s_N, s_{v,0,1}, \ldots, s_{v,0,N} \xleftarrow{\$} \mathbb{Z}_p^*$*. Compute*

- $u, v, h \in \mathbb{G}_2$ *with the condition* $u^{X_1} = v^{X_2} = h$ *(note that* (u, v, h) *is a public key of the linear encryption, and* (X_1, X_2) *is the corresponding secret key),*
- $K_{i,1} = g_1^{\frac{1}{\omega_1 + x_i}}$*,* $K_{i,2} = h^{x_i}$*, and* $B_i = g_1^{s_i x_i}$ *for all* $i \in [1, N]$*, and*
- $\Omega_1 = h^{\omega_1}$ *and* $\Omega_2 = h^{\omega_2}$*.*

For all $i \in [1, N]$*, choose* $s_{v,0,i}, y_{0,i}, r_{0,i} \xleftarrow{\$} \mathbb{Z}_p^*$*. If* $s_{v,0,i} + s_i = 0 \bmod p$*, then choose* $s_{v,0,i}$ *again until* $s_{v,0,i} + s_i \neq 0 \bmod p$ *holds. Set* $s_{T,0,i} := s_{v,0,i} + s_i$*, and compute*

- $h_{T,0,i} = g_1^{s_{T,0,i}}$
- $A_{0,i} = (g_1^{s_{T,0,i}} g_2^t g_3^{r_{0,i}} g_4)^{\frac{1}{\omega_2 + y_{0,i}}}$ *(which is a BBS+ signature for signed messages* $(s_{T,0,i}, t)$*), and*
- $d_{T,0,i} := g_1^{s_{v,0,i} x_i}$*.*

Set $Sign(s_{T,0,i}, i) = (A_{0,i}, y_{0,i}, r_{0,i})$*. Output*

- $gpk = (params, \Omega_1, \Omega_2, u, v, \mathcal{H})$*, where* $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_p^*$ *is a hash function which is modeled as a random oracle.*
- $msk = (X_1, X_2, s_1, \ldots, s_N, s_{v,0,1}, \ldots, s_{v,0,N}, x_1, \ldots, x_N, \mathsf{reg} := \{(K_{i,2}, i)\}_{i=1}^N)$*,*
- $usk_i = (K_{i,1}, K_{i,2}, B_i)$ *for all* $i \in [1, N]$*, and*
- $\mathcal{T}_0 = \{(Sign(s_{T,0,i}, i), h_{T,0,i}, d_{T,0,i})\}_{i=1}^N$*.*

$\mathsf{GSign}(gpk, usk_i, M, \mathcal{T}_t)$ *: Let* U_i *be a non-revoked user in the current time interval* t*. That is, for* $(Sign(s_{T,t,i}, i), h_{T,t,i}, d_{T,t,i}) \in \mathcal{T}_t$*,* $h_{T,t,i} = g_1^{s_{v,t,i} + s_i} := g_1^{s_{T,t,i}}$ *and* $d_{T,t,i} = g_1^{s_{v,t,i} x_i}$ *hold for some unknown exponent* $s_{v,t,i} \in \mathbb{Z}_p^*$*.* U_i *chooses* $r_1, r_2, \ldots, r_{10}, \delta_1, \delta_2 \xleftarrow{\$} \mathbb{Z}_p^*$*, sets* $\alpha = -r_1 r_2$*,* $\beta = -r_2 r_4$*,* $\beta' = r_5 y_{t,i} - r_4$*,* $\gamma = r_2 r_6 + r_7$*,* $\gamma' = r_4 r_8 + r_9$*, and* $\gamma'' = r_{10} y_{t,i}$*, and computes*

$$H_i = B_i \cdot d_{T,t,i} = g_1^{s_i x_i + s_{v,t,i} x_i} = h_{T,t,i}^{x_i},$$
$$T_1 = K_{i,1} \tilde{g}^{r_1}, T_2 = K_{i,2} \tilde{h}^{r_2}, T_3 = H_i \tilde{g}^{r_3}, T_4 = h_{T,t,i} \tilde{g}^{r_4}, T_5 = A_{t,i} \tilde{g}^{r_5},$$
$$C_1 = g^{r_1} \tilde{g}^{r_6}, C_2 = g^{\alpha} \tilde{g}^{r_7}, C_3 = g^{r_2} \tilde{g}^{r_8}, C_4 = g^{\beta} \tilde{g}^{r_9}, C_5 = g^{r_{10}} \tilde{g}^{-r_5}, C_6 = g^{\gamma''} \tilde{g}^{-r_4},$$
$$F_1 = K_{i,2} h^{\delta_1 + \delta_2}, F_2 = u^{\delta_1}, \text{ and } F_3 = v^{\delta_2}, \text{ and}$$

$$V = SPK\{(r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9, r_{10}, y_{t,i}, r_{t,i}, \alpha, \beta, \beta', \gamma, \gamma', \gamma'', \delta_1, \delta_2) :$$
$$\frac{e(T_1, \Omega_1 T_2)}{e(g_1, h)} = e(\tilde{g}, \Omega_1 T_2)^{r_1} e(T_1, \tilde{h})^{r_2} e(\tilde{g}, \tilde{h})^{\alpha}$$
$$\wedge \frac{e(T_4, T_2)}{e(T_3, h)} = \frac{e(\tilde{g}, T_2)^{r_4} e(T_4, \tilde{h})^{r_2} e(\tilde{g}, \tilde{h})^{\beta}}{e(\tilde{g}, h)^{r_3}}$$
$$\wedge \frac{e(T_5, \Omega_2)}{e(g_4, h) e(T_4, h) e(g_2, h)^t} = \frac{e(\tilde{g}, \Omega_2)^{r_5} e(g_3, h)^{r_{t,i}} e(\tilde{g}, h)^{\beta'}}{e(T_5, h)^{y_{t,i}}}$$
$$\wedge C_1 = g^{r_1} \tilde{g}^{r_6} \wedge C_2 = g^{\alpha} \tilde{g}^{r_7} \wedge C_2 = C_1^{-r_2} \tilde{g}^{\gamma}$$
$$\wedge C_3 = g^{r_2} \tilde{g}^{r_8} \wedge C_4 = g^{\beta} \tilde{g}^{r_9} \wedge C_4 = C_3^{-r_4} \tilde{g}^{\gamma'}$$
$$\wedge C_5 = g^{r_{10}} \tilde{g}^{-r_5} \wedge C_6 = g^{\gamma''} \tilde{g}^{-r_4} \wedge C_6 = C_5^{y_{t,i}} \tilde{g}^{\beta'}$$
$$\wedge \frac{T_2}{F_1} = \frac{\tilde{h}^{r_2}}{h^{\delta_1 + \delta_2}} \wedge F_2 = u^{\delta_1} \wedge F_3 = v^{\delta_2}\}(M)$$

Output $\sigma = (C_1, C_2, C_3, C_4, C_5, C_6, F_1, F_2, F_3, T_1, T_2, T_3, T_4, T_5, V)$[7].

$\mathsf{Verify}(gpk, M, \sigma, \mathcal{T}_t)$ *: Return 1 if* σ *is a valid group signature*[8]*, and 0 otherwise.*

$\mathsf{Revoke}(gpk, msk, \{U_i\}, \mathcal{T}_t)$ *: Let* $RL_{t+1} := \{U_i\}$ *be a set of revoked users. Set* $t \to t+1$*. For all* $i \in \{i | U_i \in RL_{t+1}\}$*, choose* $s'_{v,t+1,i} \xleftarrow{\$} \mathbb{Z}_p^*$*. For all* $i \in [1, N]$*, choose* $s_{v,t+1,i}, y_{t+1,i}, r_{t+1,i} \xleftarrow{\$} \mathbb{Z}_p^*$ *(until* $s_{v,t+1,i} + s_i \neq 0 \bmod p$ *holds), set* $s_{T,t+1,i} := s_{v,t+1,i} + s_i$*, and compute*

$$h_{T,t+1,i} = g^{s_{T,t+1,i}},$$
$$A_{t+1,i} = (g_1^{s_{T,t+1,i}} g_2^{t+1} g_3^{r_{t+1,i}} g_4)^{\frac{1}{\omega_2 + y_{t+1,i}}},$$

and compute $d_{T,t+1,i}$ *such that:*

$$d_{T,t+1,i} = \begin{cases} g_1^{s_{v,t+1,i} x_i} & (U_i \notin RL_{t+1}) \\ g_1^{s'_{v,t+1,i}} & (U_i \in RL_{t+1}) \end{cases}$$

and set $Sign(s_{T,t+1,i}, i) = (A_{t+1,i}, y_{t+1,i}, r_{t+1,i})$*. Output* $\mathcal{T}_{t+1} = \{(Sign(s_{T,t+1,i}, i), h_{T,t+1,i}, d_{T,t+1,i})\}_{i=1}^{N}$.

$\mathsf{Open}(gpk, msk, M, \sigma)$ *: Compute* $\frac{F_1}{F_2^{X_1} F_3^{X_2}} = K$*, and search* i *such that* $(K_{i,2}, i) \in \mathsf{reg}$ *and* $K = K_{i,2}$*. 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 w.r.t. the revocation which guarantees the unlinkability of revoked users. So, in our scheme, no r is revealed.

[7] We give the detailed form of SPK V in the appendix.
[8] We give the procedure of the verification algorithm in the appendix.

6 Discussion

One drawback of our scheme is that the number of public values depends on N, since no common attribute can be applied for implementing the revocation procedure of "each" user. So, one may think that there might be a more trivial construction (without applying ABGS) if such big-size public value is allowed. For example, as one of the most simple group signature construction, let $g^{x_1}, \ldots, g^{x_N}$ be users' public keys, and GM randomizes these value such that $y_1 := (g^{x_1})^{r_{GM}}, \ldots, y_N := (g^{x_N})^{r_{GM}}$, and publishes $y := g^{r_{GM}}$. Each user (say U_i) proves the knowledge of x_i for the relation $(g^{r_{GM}})^{x_i}$ using the OR relation such that $SPK\{x : y^x = y_1 \vee \cdots \vee y^x = y_N\}(M)$ to hide the identity $i \in [1, N]$. If a user (say U_j) is revoked, then GM publishes a random value R_j (instead of $(g^{x_j})^{r_{GM}}$). In this case, the number of revoked users is not revealed under the DDH assumption, since $(g, g^{x_j}, g^{r_{GM}}, (g^{x_j})^{r_{GM}})$ 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. [33] proposed an attribute-based anonymous credential system by applying an efficient pairing-based accumulator proposed by Camenisch et al. [10]. 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 formulae $\vee_{i=1}^{N}(\text{valid group user} \wedge U_i)$ can be implemented or not in the Sudarsono et al. attribute-based proof system. In addition, their construction also requires 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 if such large-size public key is allowed.

7 Security Analysis

The security proofs of following theorems are given in the appendix.

Theorem 1. *The proposed group signature scheme satisfies anonymity w.r.t. the revocation under the DLIN assumption and the XDH assumption.*

Theorem 2. *The proposed group signature scheme satisfies traceability under the N-HSDH assumption, the CDH assumption, and Nt-SDH assumption, where t is the final time interval that $\mathcal{A}$ outputs (M^*, σ^*).*

8 Conclusion

In this paper, we propose a revocable group signature scheme with the property of hiding r, by applying ABGS. Under a XDH-hard elliptic curve with 170 bits p (as in [14, 28]), the size of signature is 7242 bits, where the size of an element

of $\mathbb{G}_1$ is 171 bits, the size of an element of $\mathbb{G}_2$ is 513 bits, and the size of the challenge c is 80 bits. Since the size of signature in [14] (resp. in [28]) is 1444 (resp. 1533) bits, there is space for improvement the signature size. In addition, proposing a r-hiding revocable group signature with small-size public key is also interesting future work.

References

1. Ateniese, G., Tsudik, G.: Some open issues and new directions in group signatures. In: Financial Cryptography. pp. 196–211 (1999)
2. Au, M.H., Susilo, W., Mu, Y.: Constant-size dynamic k-TAA. In: SCN. pp. 111–125 (2006)
3. Bellare, M., Shi, H., Zhang, C.: Foundations of group signatures: The case of dynamic groups. In: CT-RSA. pp. 136–153 (2005)
4. Bellare, M., Micciancio, D., Warinschi, B.: Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In: EUROCRYPT. pp. 614–629 (2003)
5. Boldyreva, A.: Threshold signatures, multisignatures and blind signatures based on the Gap-Diffie-Hellman-group signature scheme. In: Public Key Cryptography. pp. 31–46 (2003)
6. Boneh, D., Boyen, X.: Short signatures without random oracles and the SDH assumption in bilinear groups. J. Cryptology 21(2), 149–177 (2008)
7. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: CRYPTO. pp. 41–55 (2004)
8. Boneh, D., Shacham, H.: Group signatures with verifier-local revocation. In: ACM Conference on Computer and Communications Security. pp. 168–177 (2004)
9. Boyen, X., Waters, B.: Full-domain subgroup hiding and constant-size group signatures. In: Public Key Cryptography. pp. 1–15 (2007)
10. Camenisch, J., Kohlweiss, M., Soriente, C.: An accumulator based on bilinear maps and efficient revocation for anonymous credentials. In: Public Key Cryptography. pp. 481–500 (2009)
11. Camenisch, J., Lysyanskaya, A.: Dynamic accumulators and application to efficient revocation of anonymous credentials. In: CRYPTO. pp. 61–76 (2002)
12. Chaum, D., van Heyst, E.: Group signatures. In: EUROCRYPT. pp. 257–265 (1991)
13. Chen, L., Li, J.: VLR group signatures with indisputable exculpability and efficient revocation. In: SocialCom/PASSAT. pp. 727–734 (2010)
14. Delerablée, C., Pointcheval, D.: Dynamic fully anonymous short group signatures. In: VIETCRYPT. pp. 193–210 (2006)
15. Emura, K., Miyaji, A., Omote, K.: A dynamic attribute-based group signature scheme and its application in an anonymous survey for the collection of attribute statistics. Journal of Information Processing 17, 216–231 (2009)
16. Emura, K., Miyaji, A., Omote, K.: An anonymous designated verifier signature scheme with revocation: How to protect a company's reputation. In: ProvSec. pp. 184–198 (2010)
17. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: CRYPTO. pp. 186–194 (1986)
18. Fujii, H., Nakanishi, T., Funabiki, N.: A proposal of efficient attribute-based group signature schemes using pairings. IEICE technical report (in Japanese) 109(272), 15–22 (2009-11-05), http://ci.nii.ac.jp/naid/110007520932/en/

19. Furukawa, J., Imai, H.: An efficient group signature scheme from bilinear maps. IEICE Transactions 89-A(5), 1328–1338 (2006)
20. Isshiki, T., Mori, K., Sako, K., Teranishi, I., Yonezawa, S.: Using group signatures for identity management and its implementation. In: Digital Identity Management. pp. 73–78 (2006)
21. Khader, D.: Attribute based group signature with revocation. Cryptology ePrint Archive, Report 2007/241 (2007)
22. Khader, D.: Attribute based group signatures. Cryptology ePrint Archive, Report 2007/159 (2007)
23. Li, J., Au, M.H., Susilo, W., Xie, D., Ren, K.: Attribute-based signature and its applications. In: ASIACCS. pp. 13–16 (2010)
24. Li, J., Kim, K.: Hidden attribute-based signatures without anonymity revocation. International Journal of Information Sciences 180(9), 1681–1689 (2010)
25. Libert, B., Vergnaud, D.: Group signatures with verifier-local revocation and backward unlinkability in the standard model. In: CANS. pp. 498–517 (2009)
26. Maji, H.K., Prabhakaran, M., Rosulek, M.: Attribute-based signatures. In: CT-RSA. pp. 376–392 (2011)
27. Nakanishi, T., Fujii, H., Hira, Y., Funabiki, N.: Revocable group signature schemes with constant costs for signing and verifying. In: Public Key Cryptography. pp. 463–480 (2009)
28. Nakanishi, T., Funabiki, N.: A short verifier-local revocation group signature scheme with backward unlinkability. In: IWSEC. pp. 17–32 (2006)
29. Nakanishi, T., Funabiki, N.: Efficient revocable group signature schemes using primes. Journal of Information Processing 16, 110–121 (2008)
30. Nakanishi, T., Kubooka, F., Hamada, N., Funabiki, N.: Group signature schemes with membership revocation for large groups. In: ACISP. pp. 443–454 (2005)
31. Nguyen, L.: Accumulators from bilinear pairings and applications. In: CT-RSA. pp. 275–292 (2005)
32. Shahandashti, S.F., Safavi-Naini, R.: Threshold attribute-based signatures and their application to anonymous credential systems. In: AFRICACRYPT. pp. 198–216 (2009)
33. Sudarsono, A., Nakanishi, T., Funabiki, N.: Efficient proofs of attributes in pairing-based anonymous credential system. In: Privacy Enhancing Technologies. pp. 246–263 (2011)

Appendix A: Detailed SPK

First, we explain the relations proved in SPK V. V proves that:

1. A signer has a valid $(K_{i,1}, K_{i,2})$ generated by the KeyGen algorithm.
   - $(K_{i,1}, K_{i,2})$ can be verified by using the public value Ω_1 such that:

$$e(K_{i,1}, \Omega_1 K_{i,2}) = e(g_1, h)$$

   - Since $K_{i,1}$ (resp. $K_{i,2}$) is hidden such that $T_1 = K_{i,1}\tilde{g}^{r_1}$, (resp. $T_2 = K_{i,2}\tilde{h}^{r_2}$), this relation is represented as:

$$\frac{e(T_1, \Omega_1 T_2)}{e(g_1, h)} = e(\tilde{g}, \Omega_1 T_2)^{r_1} e(T_1, \tilde{h})^{r_2} e(\tilde{g}, \tilde{h})^{\alpha}$$

– We need to guarantee the relation $\alpha = -r_1 r_2$ in the relation above. To prove this, introduce an intermediate value $\gamma = r_2 r_6 + r_7$, and prove that:

$$C_1 = g^{r_1}\tilde{g}^{r_6} \wedge C_2 = g^{\alpha}\tilde{g}^{r_7} \wedge C_2 = C_1^{-r_2}\tilde{g}^{\gamma}$$

Note that $C_2 = C_1^{-r_2}\tilde{g}^{\gamma} = (g^{r_1}\tilde{g}^{r_6})^{-r_2}\tilde{g}^{\gamma} = g^{-r_1 r_2}\tilde{g}^{-r_2 r_6+\gamma} = g^{\alpha}\tilde{g}^{r_7}$ yields $\alpha = -r_1 r_2$ and $\gamma = r_2 r_6 + r_7$.

2. A signer has not been revoked.
   – A non-revoked signer can compute $H_i = h_{T,t,i}^{\log_h K_{i,2}} = (g_1^{s_{T,t,i}})^{x_i}$ from B_i and $d_{T,t,i}$, where $s_{T,t,i}$ is a signed message of $A_{t,i}$. These satisfy the relations

$$e(h_{T,t,i}, K_{i,2}) = e(H_i, h)$$
$$e(A_{t,i}, \Omega_2 h^{y_{t,i}}) = e(g_1^{s_{T,t,i}} g_2^{t} g_3^{r_{t,i}} g_4, h)$$

   – Since H_i, $h_{T,t,i}$, and A_i are hidden such that $T_3 = H_i\tilde{g}^{r_3}$, $T_4 = h_{T,t,i}\tilde{g}^{r_4}$, and $T_5 = A_{t,i}\tilde{g}^{r_5}$, these relations are represented as:

$$\frac{e(T_4, T_2)}{e(T_3, h)} = \frac{e(\tilde{g}, T_2)^{r_4} e(T_4, \tilde{h})^{r_2} e(\tilde{g}, \tilde{h})^{\beta}}{e(\tilde{g}, h)^{r_3}}$$
$$\frac{e(T_5, \Omega_2)}{e(g_4, h) e(T_4, h) e(g_2, h)^{t}} = \frac{e(\tilde{g}, \Omega_2)^{r_5} e(g_3, h)^{r_{t,i}} e(\tilde{g}, h)^{\beta'}}{e(T_5, h)^{y_{t,i}}}$$

   – We need to guarantee the relations $\beta = -r_2 r_4$ and $\beta' = r_5 y_{t,i} - r_4$ in the relations above. To prove these, introduce intermediate values $\gamma' = r_4 r_8 + r_9$ and $\gamma'' = r_{10} y_{t,i}$, and prove that:

$$C_3 = g^{r_2}\tilde{g}^{r_8} \wedge C_4 = g^{\beta}\tilde{g}^{r_9} \wedge C_4 = C_3^{-r_4}\tilde{g}^{\gamma'}$$
$$C_5 = g^{r_{10}}\tilde{g}^{-r_5} \wedge C_6 = g^{\gamma''}\tilde{g}^{-r_4} \wedge C_6 = C_5^{y_{t,i}}\tilde{g}^{\beta'}$$

   As in α and γ explained before, relations $\beta = -r_2 r_4$, $\beta' = r_5 y_{t,i} - r_4$, $\gamma' = r_4 r_8 + r_9$, and $\gamma'' = r_{10} y_{t,i}$ are obtained from the relations above.
   – Note that $(A_{t,i}, r_{t,i}, y_{t,i})$ is a BBS+ signature for signed messages $(s_{T,t,i}, t)$, and therefore V depends on the current time interval t.
3. A value for the Open algorithm is included in σ.
   – (F_1, F_2, F_3) is a ciphertext (of the linear encryption scheme) of the plaintext $K_{i,2}$, which can be computed by decrypting (F_1, F_2, F_3) using msk.

Next, we describe the detailed SPK of our scheme as follows.

1. Choose $r_{r_1}, r_{r_2}, r_{r_3}, r_{r_4}, r_{r_5}, r_{r_6}, r_{r_7}, r_{r_8}, r_{r_9}, r_{r_{10}}, r_{y_{t,i}}, r_{r_{t,i}}, r_\alpha, r_\beta, r_{\beta'}, r_\gamma, r_{\gamma'}, r_{\gamma''}$, $r_{\delta_1}, r_{\delta_2} \xleftarrow{\$} \mathbb{Z}_p^*$.

2. Compute

$$R_1 = e(\tilde{g}, \Omega_1 T_2)^{r_{r_1}} e(T_1, \tilde{h})^{r_{r_2}} e(\tilde{g}, \tilde{h})^{r_\alpha},\ R_2 = \frac{e(\tilde{g}, T_2)^{r_{r_4}} e(T_4, \tilde{h})^{r_{r_2}} e(\tilde{g}, \tilde{h})^{r_\beta}}{e(\tilde{g}, h)^{r_{r_3}}},$$
$$R_3 = \frac{e(\tilde{g}, \Omega_2)^{r_{r_5}} e(g_3, h)^{r_{r_{t,i}}} e(\tilde{g}, h)^{r_{\beta'}}}{e(T_5, h)^{r_{y_{t,i}}}},\ R_4 = g^{r_{r_1}} \tilde{g}^{r_{r_6}},\ R_5 = g^{r_\alpha} \tilde{g}^{r_{r_7}}, R_6 = C_1^{-r_{r_2}} \tilde{g}^{r_\gamma},$$
$$R_7 = g^{r_{r_2}} \tilde{g}^{r_{r_8}},\ R_8 = g^{r_\beta} \tilde{g}^{r_{r_9}},\ R_9 = C_3^{-r_{r_4}} \tilde{g}^{r_{\gamma'}},\ R_{10} = g^{r_{r_{10}}} \tilde{g}^{-r_{r_5}},\ R_{11} = g^{r_{\gamma''}} \tilde{g}^{-r_{r_4}},$$
$$R_{12} = C_5^{r_{y_{t,i}}} \tilde{g}^{r_{\beta'}},\ R_{13} = \frac{\tilde{h}^{r_{r_2}}}{h^{r_{\delta_1} + r_{\delta_2}}},\ R_{14} = u^{r_{\delta_1}},\ R_{15} = v^{r_{\delta_2}},$$
$$c = \mathcal{H}(gpk, M, C_1, C_2, C_3, C_4, C_5, C_6, F_1, F_2, F_3, T_1, T_2, T_3, T_4, T_5, R_1, \ldots, R_{15}),$$
$$s_{r_i} = r_{r_i} + c r_i\ (i \in [1, 10]),\ s_{y_{t,i}} = r_{y_{t,i}} + c y_{t,i},\ s_{r_{t,i}} = r_{r_{t,i}} + c r_{t,i},$$
$$s_\alpha = r_\alpha + c\alpha,\ s_\beta = r_\beta + c\beta,\ s_{\beta'} = r_{\beta'} + c\beta',\ s_\gamma = r_\gamma + c\gamma,\ s_{\gamma'} = r_{\gamma'} + c\gamma',$$
$$s_{\gamma''} = r_{\gamma''} + c\gamma'',\ s_{\delta_1} = r_{\delta_1} + c\delta_1, \text{ and } s_{\delta_2} = r_{\delta_2} + c\delta_2,$$

3. Output $V = (c, \{s_{r_i}\}_{i=1}^{10}, s_{y_{t,i}}, s_{r_{t,i}}, s_\alpha, s_\beta, s_{\beta'}, s_\gamma, s_{\gamma'}, s_{\gamma''}, s_{\delta_1}, s_{\delta_2})$.

Next, we describe the verification of $\sigma = (C_1, C_2, C_3, C_4, C_5, C_6, F_1, F_2, F_3, T_1, T_2, T_3, T_4, T_5, c, \{s_{r_i}\}_{i=1}^{10}, s_{y_{t,i}}, s_{r_{t,i}}, s_\alpha, s_\beta, s_{\beta'}, s_\gamma, s_{\gamma'}, s_{\gamma''}, s_{\delta_1}, s_{\delta_2})$ as follows.

1. Compute

$$\tilde{R}_1 = e(\tilde{g}, \Omega_1 T_2)^{s_{r_1}} e(T_1, \tilde{h})^{s_{r_2}} e(\tilde{g}, \tilde{h})^{s_\alpha} (\frac{e(T_1, \Omega_1 T_2)}{e(g_1, h)})^{-c},$$
$$\tilde{R}_2 = \frac{e(\tilde{g}, T_2)^{s_{r_4}} e(T_4, \tilde{h})^{s_{r_2}} e(\tilde{g}, \tilde{h})^{s_\beta}}{e(\tilde{g}, h)^{s_{r_3}}} (\frac{e(T_4, T_2)}{e(T_3, h)})^{-c},$$
$$\tilde{R}_3 = \frac{e(\tilde{g}, \Omega_2)^{s_{r_5}} e(g_3, h)^{s_{r_{t,i}}} e(\tilde{g}, h)^{s_{\beta'}}}{e(T_5, h)^{s_{y_{t,i}}}} (\frac{e(T_5, \Omega_2)}{e(g_4, h) e(T_4, h) e(g_2, h)^t})^{-c},$$
$$\tilde{R}_4 = g^{s_{r_1}} \tilde{g}^{s_{r_6}} C_1^{-c},\ \tilde{R}_5 = g^{r_\alpha} \tilde{g}^{s_{r_7}} C_2^{-c},\ \tilde{R}_6 = C_1^{-s_{r_2}} \tilde{g}^{r_\gamma} C_2^{-c},$$
$$\tilde{R}_7 = g^{s_{r_2}} \tilde{g}^{s_{r_8}} C_3^{-c},\ \tilde{R}_8 = g^{s_\beta} \tilde{g}^{s_{r_9}} C_4^{-c},\ \tilde{R}_9 = C_3^{-s_{r_4}} \tilde{g}^{s_{\gamma'}} C_4^{-c},$$
$$\tilde{R}_{10} = g^{s_{r_{10}}} \tilde{g}^{-s_{r_5}} C_5^{-c},\ \tilde{R}_{11} = g^{s_{\gamma''}} \tilde{g}^{-s_{r_4}} C_6^{-c},\ \tilde{R}_{12} = C_5^{s_{y_{t,i}}} \tilde{g}^{s_{\beta'}} C_6^{-c},$$
$$\tilde{R}_{13} = \frac{\tilde{h}^{s_{r_2}}}{h^{s_{\delta_1} + s_{\delta_2}}} (\frac{T_2}{F_1})^{-c},\ \tilde{R}_{14} = u^{s_{\delta_1}} F_2^{-c}, \text{ and}$$
$$\tilde{R}_{15} = v^{r_{\delta_2}} F_3^{-c}.$$

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

2. Check $c = \mathcal{H}(gpk, M, C_1, C_2, C_3, C_4, C_5, C_6, F_1, F_2, F_3, T_1, T_2, T_3, T_4, T_5, \tilde{R}_1, \ldots, \tilde{R}_{15})$. If it holds, then output 1, and 0, otherwise.

Appendix B: Security Analysis

Proof of Theorem 1

Proof. Let $\mathcal{C}$ be the challenger of the linear encryption, and $\mathcal{A}$ be the adversary who breaks anonymity w.r.t. the revocation of our scheme. We construct the

algorithm $\mathcal{B}$ that breaks the IND-CPA security of the linear encryption. First, $\mathcal{C}$ gives the public key of the linear encryption (u, v, h). $\mathcal{B}$ chooses all values, except (u, v, h), and therefore $\mathcal{B}$ can answer all queries issued from $\mathcal{A}$.

In the challenge phase, $\mathcal{A}$ sends (M^*, U_{i_0}, U_{i_1}). Let $h^{x_{i_0}}$ and $h^{x_{i_1}}$ be (a part of) secret key of U_{i_0}, and U_{i_1}, respectively. $\mathcal{B}$ sets $M_0^* := h^{x_{i_0}}$ and $M_1^* := h^{x_{i_1}}$, and sends (M_0^*, M_1^*) to $\mathcal{C}$ as the challenge messages of the linear encryption. $\mathcal{C}$ sends the challenge ciphertext C^*. $\mathcal{B}$ sets $C^* = (F_1, F_2, F_3)$, and computes the challenge group signature σ^*. Note that $\mathcal{B}$ does not know the random number (δ_1^*, δ_2^*) and $\mu \in \{0, 1\}$ such that $C^* = (h^{x_{i_\mu}} h^{\delta_1^* + \delta_2^*}, u^{\delta_1^*}, v^{\delta_2^*})$, since $(\delta_1^*, \delta_2^*, \mu)$ are chosen by $\mathcal{C}$. So, $\mathcal{B}$ uses the backpatch of the random oracle $\mathcal{H}$ for computing σ^*, and includes C^* in σ^*. Then, all values (except C^*) is independent of μ. Note that even if U_{i_μ} is revoked in the challenge interval, $\mathcal{B}$ can compute σ^*, since $\mathcal{B}$ knows msk. If either U_{i_0} or U_{i_1} is revoked in the challenge interval, this fact is not used for guessing μ under the XDH assumption, since $(g_1^{s_{v,t+1,i_\mu} + s_{i_\mu}}, g_1^{s_{v,t+1,i_\mu} x_{i_\mu}})$ and $(g_1^{s_{v,t+1,i_\mu} + s_{i_\mu}}, g_1^{s'_{v,t+1,i_\mu}})$ are indistinguishable.

Finally, $\mathcal{A}$ outputs the guessing bit $\mu' \in \{0, 1\}$. $\mathcal{B}$ outputs μ' as the guessing bit of the IND-CPA game of the linear encryption. □

Proof of Theorem 2

Proof. Let $\mathcal{A}_1$ be an adversary who outputs (M^*, σ^*) where for $ID_{i^*} \leftarrow \mathsf{Open}(msk, M^*, \sigma^*)$, $ID_{i^*} \notin CU$ holds. As a case of the first one, let $\mathcal{A}_2$ be an adversary who outputs (M^*, σ^*) where for $ID_{i^*} \leftarrow \mathsf{Open}(msk, M^*, \sigma^*)$, $ID_{i^*} \notin CU$ and $U_{i^*} \notin \{U_1, \ldots, U_N\}$ holds. In addition, let $\mathcal{A}_3$ be an adversary who outputs (M^*, σ^*) where for $ID_{i^*} \leftarrow \mathsf{Open}(msk, M^*, \sigma^*)$, $ID_{i^*} \in RU$ holds. We construct an algorithm $\mathcal{B}_1$ (resp. $\mathcal{B}_2$ and $\mathcal{B}_3$) that breaks the N-HSDH assumption (resp. q-SDH assumption, where q is the number of signing queries, and the CDH assumption) by using $\mathcal{A}_1$ (resp. $\mathcal{A}_2$ and $\mathcal{A}_3$).

First, we describe $\mathcal{B}_1$. Let $g_1, h, h^{\omega_1}, \{(g_1^{\frac{1}{\omega_1 + x_i}}, h^{x_i})\}_{i=1,\ldots,N}$ be an N-HSDH instance. $\mathcal{B}_1$ selects $U_{i^*} \in \{U_1, \ldots, U_N\}$, and choose all values, except g_1, h, and $\Omega_1 := h^{\omega_1}$. $\mathcal{B}_1$ answers queries issued by $\mathcal{A}_1$ as follows:

Revocation : $\mathcal{A}_1$ requests the revocation of users $ID_{i_1}, \ldots, ID_{i_{k_t}}$ for some constant $k_t \in [1, N]$. Since $\mathcal{B}_1$ knows ω_2, $\mathcal{B}_1$ adds $ID_{i_1}, \ldots, ID_{i_{k_t}}$ to RU_t, and simply returns the result of the Revoke algorithm.

GSigning : $\mathcal{A}_1$ requests a group signature on a message M for a user U_i where $ID_i \notin CU$. Since $\mathcal{B}_1$ does not know $g_1^{x_i}$, $\mathcal{B}_1$ computes σ by using the backpatch of the random oracle $\mathcal{H}$, and gives σ to $\mathcal{A}$.

Corruption : $\mathcal{A}_1$ requests the secret key of a user U_i. If $U_i = U_{i^*}$, then $\mathcal{B}_1$ aborts. Otherwise, $\mathcal{B}_1$ sets $(g_1^{\frac{1}{\omega_1 + x_i}}, h^{x_i}) = (K_{i,1}, K_{i,2})$, chooses $s'_i \xleftarrow{\$} \mathbb{Z}_p^*$, sets $s'_i = s_i x_i$, and computes $B_i = g^{s'_i}$. $\mathcal{B}_1$ adds ID_i to CU, and gives $(K_{i,1}, K_{i,2}, B_i)$ to $\mathcal{A}_1$.

Opening : Since $\mathcal{B}_1$ has (X_1, X_2), $\mathcal{B}_1$ simply returns the result of the Open algorithm.

Finally, $\mathcal{A}_1$ outputs a past interval $t^* \leq t$ for the current interval t, and a pair (M^*, σ^*). By using the extractor of SPK, $\mathcal{B}_1$ gets: $(K^*_{i,1}, K^*_{i,2}, H^*_i)$, where $e(K^*_{i,1}, \Omega_1 K^*_{i,2}) = e(g_1, h)$, $e(h_{T,t,i}, K^*_{i,2}) = e(H^*_i, h)$, $F_1 = K^*_{i,2} h^{\delta_1+\delta_2}$, $F_2 = u^{\delta_1}$, and $F_3 = v^{\delta_2}$ hold. From (F_1, F_2, F_3), $\mathcal{B}_1$ obtains i by using the Open algorithm. If $i \neq i^*$, then $\mathcal{B}_1$ aborts. Otherwise, $\mathcal{B}_1$ outputs $(K^*_{i,1}, K^*_{i,2})$ as a solution of the N-HSDH problem.

Next, we describe $\mathcal{B}_2$ that outputs a forged BBS+ signature. Let $\mathcal{C}$ be the challenger of the BBS+ signature. $\mathcal{B}_2$ is given $(g, g_1, g_2, g_3, g_4, h, \Omega_2)$ from $\mathcal{C}$. $\mathcal{B}_2$ chooses all values, except $(g, g_1, g_2, g_3, g_4, h, \Omega_2)$. For each revocation query, $\mathcal{B}_2$ issues N signing queries to $\mathcal{C}$ for obtaining $A_{\cdot,i}$. So, $\mathcal{B}_2$ needs to issue the signing query in Nt times. For other queries, $\mathcal{B}_2$ can answer since $\mathcal{B}_2$ knows all other secret values. Finally, $\mathcal{A}_3$ outputs a past interval $t^* \leq t$ for the current interval t, and a pair (M^*, σ^*). By using the extractor of SPK, $\mathcal{B}_2$ gets: $(A_{t^*,i^*}, y_{t^*,i^*}, r_{t^*,i^*})$, where $e(A_{t^*,i^*}, \Omega_2 h^{y_{t^*,i^*}}) = e(g_1^{s_{T,t^*,i^*}} g_2^{t^*} g_3^{r_{t^*,i^*}} g_4, h)$. Note that, since $U_{i^*} \notin \{U_1, \ldots, U_N\}$, $\mathcal{B}_2$ does not obtain $(A_{t^*,i^*}, y_{t^*,i^*}, r_{t^*,i^*})$ from $\mathcal{C}$. So, $\mathcal{B}_2$ outputs a forged BBS+ signature $(A_{t^*,i^*}, y_{t^*,i^*}, r_{t^*,i^*})$.

Finally, we describe $\mathcal{B}_3$ that breaks the CDH assumption. Let (g_1, g_1^a, g_1^b) be an CDH instance. $\mathcal{B}_3$ selects $U_{i^*} \in \{U_1, \ldots, U_N\}$, sets $x_{i^*} := a$ and $s_{i^*} := b$, and choose all values, except g_1 and usk_{i^*}. $\mathcal{B}_3$ answers queries issued by $\mathcal{A}_3$ as follows:

Revocation : $\mathcal{A}_3$ requests the revocation of users $ID_{i_1}, \ldots, ID_{i_{k_t}}$ for some constant k_t. Since $\mathcal{B}_3$ knows ω_2, $\mathcal{B}_3$ adds $ID_{i_1}, \ldots, ID_{i_{k_t}}$ to RU_t, and simply returns the result of the Revoke algorithm.

GSigning : $\mathcal{A}_3$ requests a group signature on a message M for a user U_i where $ID_i \notin CU$. $\mathcal{B}_3$ computes σ by using the backpatch of the random oracle $\mathcal{H}$, and gives σ to $\mathcal{A}$.

Corruption : $\mathcal{A}_3$ requests the secret key of a user U_i. If $U_i = U_{i^*}$, then $\mathcal{B}_3$ aborts. Otherwise, $\mathcal{B}_3$ adds ID_i to CU, and gives $(K_{i,1}, K_{i,2}, B_i)$ to $\mathcal{A}_3$.

Opening : Since $\mathcal{B}_3$ has (X_1, X_2), $\mathcal{B}_3$ simply returns the result of the Open algorithm.

Finally, $\mathcal{A}_3$ outputs a past interval $t^* \leq t$ for the current interval t, and a pair (M^*, σ^*). By using the extractor of SPK, $\mathcal{B}_3$ gets: H^*_i, where $e(K^*_{i,1}, \Omega_1 K^*_{i,2}) = e(g_1, h)$, $e(h_{T,t,i}, K^*_{i,2}) = e(H^*_i, h)$, $F_1 = K^*_{i,2} h^{\delta_1+\delta_2}$, $F_2 = u^{\delta_1}$, and $F_3 = v^{\delta_2}$ hold. From (F_1, F_2, F_3), $\mathcal{B}_3$ obtains i by using the Open algorithm. If $i \neq i^*$, then $\mathcal{B}_3$ aborts. Otherwise, $\mathcal{B}_3$ solves the CDH problem as follows. Since $U_i \in RL_t$, $\mathcal{B}_3$ has computed $g_1^{s_{v,t,i^*}} \cdot g_1^b = g_1^{s_{v,t,i^*}+s_{i^*}}$ and $g_1^{s'_{v,t,i^*}}$. That is, $H^*_i = B_{i^*} \cdot g_1^{s_{v,t,i^*} x_i} = g_1^{ab+a s_{v,t,i^*} x_i}$ holds. So, $\mathcal{B}_3$ outputs $H^*_i / (g_1^a)^{s_{v,t,i^*}} = g_1^{ab}$ as the solution of the CDH problem. □