Group Signatures with Time-bound Keys Revisited: A New Model and an Efficient Construction

Keita Emura
National Institute of Information and Communications Technology (NICT)
k-emura@nict.go.jp

Takuya Hayashi
National Institute of Information and Communications Technology (NICT)
takuya.hayashi@nict.go.jp

Ai Ishida
Tokyo Institute of Technology/National Institute of Advanced Industrial Science and Technology (AIST)
ishida0@is.titech.ac.jp

ABSTRACT

Chu et al. (ASIACCS 2012) proposed group signature with time-bound keys (GS-TBK) where each signing key is associated to an expiry time τ. In addition to prove the membership of the group, a signer needs to prove that the expiry time has not passed, i.e., $t < \tau$ where t is the current time. A signer whose expiry time has passed is automatically revoked, and this revocation is called natural revocation. Simultaneously, signers can be revoked before their expiry times have passed due to the compromise of the credential. This revocation is called premature revocation. A nice property of the Chu et al. proposal is that the size of revocation lists can be reduced compared to those of Verifier-Local Revocation (VLR) group signature schemes, by assuming that natural revocation accounts for most of signer revocations in practice, and prematurely revoked signers are only a small fraction. In this paper, we point out that the definition of traceability of Chu et al. did not capture unforgeability of expiry time of signing keys which guarantees that no adversary who has a signing key associated to an expiry time τ can compute a valid signature after τ has passed. We introduce a security model that captures unforgeability, and propose a GS-TBK scheme secure in the new model. Our scheme also provides the constant signing costs whereas those of the previous schemes depend on the bit-length of the time representation. Finally, we give implementation results, and show that our scheme is feasible in practical settings.

Keywords

Group Signatures; Time-bound Keys; Revocation

1. INTRODUCTION

1.1 Group Signatures and Revocation

Group signatures, proposed by Chaum and van Heyst [19], provide a functionality to anonymously prove the membership of a group. After a seminal work by Boneh, Boyen, and Shacham (BBS) [11], many pairing-based constructions have been proposed so far, e.g., [10, 14, 34, 35, 31, 28, 25, 21, 49]. Recently, lattice-based constructions also have been proposed, e.g., [26, 30]. Among them, providing the revocation functionality[1] is regarded as one of the major research topics of group signatures, where an authority can revoke the membership of users. One reason of the difficulties to provide revocation functionality in the group signature context is that a verifier needs to publicly confirm whether an anonymous signer has been revoked or not. To overcome this difficulty, several attempts have been made so far.

In revocable group signatures, there are two checks in the verification algorithm, the verification check and the revocation check. Then, almost all currently known revocable group signature schemes can be classified as follows.

1. Revoked signers cannot compute a signature that passes the verification check (and therefore no revocation check procedure is required in this type) [3, 4, 32, 33, 41, 44, 47, 46, 54].
2. Any signer can compute a signature that passes the verification check, but a verifier can check whether the signer has been revoked or not by the revocation check procedure [12, 15, 18, 35, 42, 43, 53].

An example of the first type scheme is the Libert-Peters-Yung revocable group signature scheme [33]: a ciphertext of broadcast encryption is published such that non-revoked signers are regarded as authorized users and they can decrypt the ciphertext. A non-revoked signer proves the decryption ability of the ciphertext for proving that the signer is not revoked. Since revoked signers cannot decrypt the ciphertext, revoked signers cannot compute a signature that passes the verification check in the first place. In this type, a signer not only needs to prove the membership of the group but also proves that the signer is not revoked, and the computational cost of the signing algorithm is relatively high compared to that of the second type scheme. On the other hand, the computational cost of the verification algorithm is relatively low compared to that of the first type scheme. More precisely, the verification cost is constant in terms of the number of revoked signers.

[1]We clearly distinguish revocation and anonymity revocation. The former means that signing keys are expired whereas the latter means that an authority called opener identifies the signer.

ASIA CCS '17, April 02-06, 2017, Abu Dhabi, United Arab Emirates

DOI: http://dx.doi.org/10.1145/3052973.3052979

Table 1: Efficiency Comparison of group signature schemes with time-bound keys

Scheme	Group public key size	Signature size	Signing key size	Revocation list size	Expiration info size	Signing cost	Verification cost	BU/ UET
Chu et al. [20]	$O(1)$	$O(\log T)$	$O(\log T)$	$O(R_{\mathsf{pre}} \log T)$	-	$O(\log T)$	$O(R_{\mathsf{pre}})^1$	NO
Liu et al. [37]	$O(\log T)$	$O(1)$	$O(\log T)$	$O(R_{\mathsf{pre}} \log T)$	-	$O(\log T)$	$O(R_{\mathsf{pre}})^1$	NO
Ours	$O(1)$	$O(1)$	$O(\log T)$	$O(R_{\mathsf{pre}})$	$O(\log T)$	$O(1)$	$O(R_{\mathsf{pre}})$	YES

T: The maximum size of expiry time.
R_{pre}: The number of "prematurely" revoked signers.
BU: Backward Unlinkability
UET: Unforgeability of Expiry Time of Signing Keys
[1] More precisely, this complexity is represented as $O(\log T + R_{\mathsf{pre}})$. Here, we assume that $\log T < R_{\mathsf{pre}}$.

An example of the second type scheme is the Boneh-Shacham Verifier-Local Revocation (VLR) group signature scheme [12]: a signer is not involved in the revocation procedure. Thus, any signer can compute a signature that passes the verification check. In order to check whether the signer of a signature is revoked or not, the verifier runs the revocation check procedure by using a revocation list RL_t that contains information of revoked signers at time period t. Since a signer just needs to prove the membership of the group, the computational cost of the signing algorithm is relatively low compared to that of the first type scheme. On the other hand, the computational cost of the verification algorithm is relatively high compared to that of the first type scheme, due to the additional revocation check procedure. Usually, the computational cost of the revocation check linearly depends on the size of revocation lists. Thus, reducing the size of revocation lists is highly desirable. However, information of revoked signers is added to the revocation list at each time, and the size of revocation list grows periodically.

1.2 Group Signatures with Time-bound Keys

Chu et al. [20] proposed group signature with time-bound keys (GS-TBK). We can regard that GS-TBK has the properties of both revocation types simultaneously. An expiry time τ is set to each signing key, and in addition to prove the membership of the group, a signer needs to prove that the expiry time has not passed, i.e., $t < \tau$ where t is the current time. Since signers whose expiry time has passed cannot compute a signature that passes the verification check in the first place, this can be classified as the first type. Chu et al. called this revocation "natural" revocation. Simultaneously, signers can be revoked before their expiry times have passed due to the compromise of the credential. Since a verifier runs the revocation check procedure, this can be classified as the second type. Chu et al. called this revocation "premature" revocation.

A nice property of the Chu et al. proposal is that the size of revocation lists can be reduced compared to those of VLR group signature schemes, by assuming that natural revocation accounts for most of signer revocations in practice, and prematurely revoked signers are only a small fraction. That is, a revocation list RL_t just needs to contain information of revoked signers whose expiry time τ has not passed (i.e., $t < \tau$). A small size revocation list leads to reduce the costs of revocation check.

1.3 Our Target and Contribution: A New Model and an Efficient Construction

We point out that the definition of traceability of Chu et al. [20] (and its journal version [37] also) did not capture the following case:

Forgery of expiry time : An adversary who has a signing key associated to an expiry time τ may compute a valid signature after τ has passed.

More precisely, the winning condition of the adversary of traceability in [20, 37] is defined as follows. Let (σ^*, m^*) be the output of the adversary and t^* be the time that the adversary outputs (σ^*, m^*). Then, it is required that (1) σ^* is a valid signature on the message m^* with the revocation list RL_{t^*}, (2) σ^* is not obtained from the signing queries with t^* on m^*, and (3) σ^* is NOT traced to a signer in $\mathsf{CU} \setminus \mathsf{RU}_{t^*}$ or the trace is failed, where CU is the set of corrupted signers, i.e., the adversary has their signing keys, and RU_{t^*} is a set of revoked signers at t^*. It seems natural to additionally consider the case that (4) σ^* is traced to a signer in $\mathsf{CU} \setminus \mathsf{RU}_{t^*}$ and $\tau^* < t^*$ holds where τ^* is the expiry time of the corresponding signing key of the traced signer. This unforgeability of expiry time should be considered due to the usage of time-bound signing keys. We remark that we do not find any particular attack against the schemes [20, 37] in the new model. Nevertheless, it seems meaningful to provide a provably secure scheme in the new model. In addition to unforgeability of expiry time, we also consider backward unlinkability [42] and non-frameability under the dishonest group manager setting that were not considered in [20, 37].

Next target is efficiency since the signing cost and the signature size of the Chu et al. scheme [20] linearly depend on $\log T$ where T is the maximum-length of time t. They apply the encoding technique proposed by Lin and Tzeng [36] for proving $t < \tau$. In the journal version [37], by using accumulators [46] together with the encoding, the signature size can be constant whereas the signing cost still linearly depends on $\log T$.

Our Contribution: In this paper, we define a new model of GS-TBK that captures unforgeability of expiry time of signing keys, and propose a scheme secure in this model. Moreover, in our scheme, the cost of the signing algorithm is constant whereas those of the previous schemes [20, 37] depend on the bit-length of the time representation. In addition to this, we further reduce the size of the revocation list compared to those of the previous ones. We give the efficiency comparison in Table 1.

For proving that the expiry time has not passed, we employ the Ohara et al. methodology [47] which efficiently implements the Libert-Peters-Yung revocable group signature scheme [33] in the random oracle model. Ohara et al. employed the Complete Subtree (CS) method [45] for revocation, as in the Libert-Peters-Yung (CS-based) construction, where each signer is assigned to a leaf node of a tree structure. The Libert-Peters-Yung scheme uses an identity-based encryption (IBE) scheme for instantiating the CS method in the public key setting [23]. Ohara et al. used a signature scheme instead of using an IBE scheme. A revocation list contains signatures of nodes which are determined by the CS method. In other words, signatures for revoked signers are not contained in the list. Non-revoked signers can prove that a signature related to the signers is contained in the list. We employ the Ohara et al. methodology such that a time t and an expiry time τ are assigned to leaves in a sequential order, and at the time t, leaves associated to $1, 2, \ldots, t-1$ are revoked. For the natural revocation, the group manager broadcasts expiry information ei_t. If $t < \tau$, then a signer whose signing key is not expired can prove that τ is not revoked by using ei_t. This ei_t helps signers to efficiently prove that an expiry time τ has not passed against the current time t where $t < \tau$ without showing τ. One drawback of our construction is that signers need to download expiration information ei_t at each time t though neither encryption for sending ei_t nor updating secret key is required. In the meantime, no additional expiry time-related value for signing is required in the previous schemes [20, 37]. At the expense of this drawback, we can achieve $O(1)$ signing cost and $O(R_{\mathsf{pre}})$-size revocation list whereas those of the previous schemes are $O(\log T)$ and $O(R_{\mathsf{pre}} \log T)$ respectively, where T is the maximum size of expiry time and R_{pre} is the number of "prematurely" revoked signers. Remark that in our scheme still signers are not required to obtain signer revocation-related information, i.e., revocation lists, for generating signatures. Moreover, ei_t is independent to the premature revocation, and its size does not depend on the number of revoked signers.

Finally, we implement our scheme and show that our scheme provides enough efficiency in practice. In our implementation we employ (type 3) Barreto-Naehrig (BN) curves [7] with 254-bit prime order, and use the RELIC library [2].

1.4 Related Work

Malina et al. [38, 39] also proposed group signatures with time-bound membership. However, different from ours and Chu et al. [20, 37], they do not consider premature revocation. Moreover, some information of expiry time (index k in [38, 39]) has to be contained in a signature. As mentioned in [20, 37], signers may not wish to leak such information (even partially) because it may be used to infer signers' identity. Hence, as in [20, 37], a signer is allowed to completely (i.e., in the sense of zero-knowledge proofs) hide his/her expiry time in our scheme.

The first VLR group signature scheme was proposed by Boneh and Shacham [12], and Nakanishi and Funabiki [42] considered the notion called backward unlinkability. A signature contains a target group (i.e., $\mathbb{G}_T$) element in their scheme. Later, they proposed a more efficient VLR group signature scheme [43] whose signature contains base group (or $\mathbb{Z}_p$) elements only. Hence, we employ the Nakanishi-Funabiki scheme proposed in [43] with a slight modification due to the curve selection since they employed (type 2) MNT curves [40] whereas we employ (type 3) BN curves.

2. PRELIMINARIES

In this section, we define the Complete Subtree algorithm for time-bound keys (CS-TBK), complexity assumptions, and the BBS+ signature scheme [6]. First, we give the definition of the CS-TBK algorithm which implements (a special case of) the CS method [45]. Let $\mathtt{BT}$ be a binary tree that has T leaf nodes where T is the maximum size of time. The algorithm finds subtrees that cover all non-revoked nodes. Note that, in the Ohara et al. revocable group signature scheme, each user is assigned to a leaf whereas each time is sequentially assigned to a leaf node in the algorithm.

Definition 2.1 (The CS-TBK Algorithm). *This algorithm takes as input a binary tree* $\mathtt{BT}$ *and the current time* t*, and outputs a set of nodes. If* η *is a non-leaf node, then* η_{left} *and* η_{right} *denote the left and right child of* η*, respectively. Each time is sequentially assigned to a leaf node.* $\mathsf{Path}(\eta)$ *denotes the set of nodes on the path from* η *to the root node. The description of the algorithm is given below.*

$\mathsf{CS\text{-}TBK}(\mathtt{BT}, t)$:
$\mathtt{X}, \mathtt{Y} \leftarrow \emptyset$;
$\forall 1 \le i < t$
Add $\mathsf{Path}(\eta)$ *to* $\mathtt{X}$ *where* i *is assigned to* η
$\forall x \in \mathtt{X}$
If $x_{left} \notin \mathtt{X}$ *then add* x_{left} *to* $\mathtt{Y}$
If $x_{right} \notin \mathtt{X}$ *then add* x_{right} *to* $\mathtt{Y}$
If $\mathtt{Y} = \emptyset$ *then add* $\mathtt{root}$ *to* $\mathtt{Y}$
Return $\mathtt{Y}$

In our GS-TBK scheme, each time t is assigned to a leaf node, and an expiry time τ is also assigned to a leaf node. That is, one leaf node is shared by multi signers if their expiry times are the same. If τ is assigned to a leaf node η, the group manager generates signatures of nodes contained in $\mathsf{Path}(\eta)$, and then these signatures are issued to signers whose expiry time is τ. Remark that randomnesses of these signatures are different for signers even they share the same leaf node. At the current time t, all leaf nodes of past time, i.e., all left-side leaves of the leaf node assigned to t are revoked.[2] Next, the group manager generates signatures of nodes generated by the CS-TBK algorithm, and publishes signatures as expiration information ei_t at time t. If $t < \tau$, then the corresponding signers have a signature of a node such that ei_t contains a signature of the same node.

We give an example in the case of $T = 8$ as follows. We show a case that τ has not passed in Fig 1, and also show a case that τ has passed in Fig 2. Let τ be assigned to the node 11. Then, signers whose expiry time is τ have signatures of nodes 1, 2, 5, and 11. In Fig 1, nodes 8 and 9 are revoked. Then, nodes 3 and 5 are selected as roots of subtrees. Thus,

[2]In the usual CS method, a user is associated to a leaf node, and who will be revoked is not predictable. So, the size of $\mathtt{Y}$ is $O(r \log(N/r))$ where N is the number of users (leaves), and r is the number of revoked signers. On the other hand, in our usage, though a time is associated to a leaf node as usual, leaves are "sequentially" revoked. So, the size of $\mathtt{Y}$ is at most $\log T$. This is essentially the same as the encoding for attribute-based encryption with range membership [5].

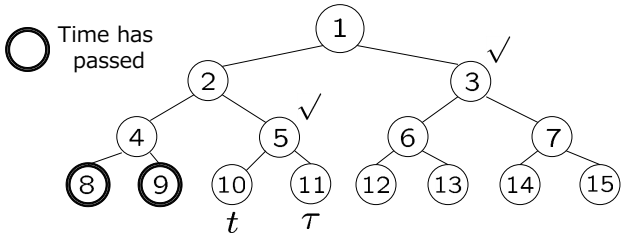

Figure 1: τ has not passed

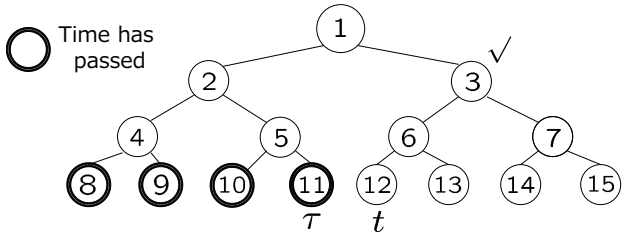

Figure 2: τ has passed

ei_t contains signatures of nodes 3 and 5. Then, the signers can prove that they have a signature of the node 5 (without revealing the node itself). In Fig 2, ei_t contains a signature of the node 3 only. Since τ has passed, signatures of 1, 2, 5, and 11 are not contained in ei_t.

Next, we define complexity assumptions. Let p be a λ-bit prime, $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ are groups of order p, $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a bilinear map, and $g, \widehat{g}$ are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. We use the asymmetric setting (type 3 curves), i.e., $\mathbb{G}_1 \neq \mathbb{G}_2$, and no efficient isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$ is known.

Next, we define decision Diffie-Hellman assumption on $\mathbb{G}_1$ (DDH1) as follows.

Definition 2.2 (DDH1 Assumption). *Let $D := (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, \widehat{g})$, $a, b \xleftarrow{\$} \mathbb{Z}_p^*$ and $Z \xleftarrow{\$} \mathbb{G}_1 \setminus \{g^{ab}\}$. We say that the DDH1 assumption holds if for any probabilistic polynomial time (PPT) adversary $\mathcal{A}$, the advantage $\mathrm{Adv}_{DDH1}(\lambda) := |\Pr[\mathcal{A}(D, g^a, g^b, g^{ab}) \to true] - \Pr[\mathcal{A}(D, g^a, g^b, Z) \to true]|$ is negligible.*

Next, we define the decision linear (DLIN) assumption. Here, we use an asymmetric variant [27].

Definition 2.3 (DLIN Assumption). *Let $D := (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \widetilde{g}, \widehat{g})$, $a, b, c, d \xleftarrow{\$} \mathbb{Z}_p^*$, $g' := \widetilde{g}^c$, $\widehat{g}' := \widehat{g}^c$, $h := \widetilde{g}^d$, $\widehat{h} := \widehat{g}^d$, and $Z \xleftarrow{\$} \mathbb{G}_1 \setminus \{h^{a+b}\}$. We say that the DLIN assumption holds if for any PPT adversary $\mathcal{A}$, the advantage $\mathrm{Adv}_{DLIN}(\lambda) := |\Pr[\mathcal{A}(D, g', \widehat{g}', h, \widehat{h}, \widetilde{g}^a, {g'}^b, h^{a+b}) \to true] - \Pr[\mathcal{A}(D, g', \widehat{g}', h, \widehat{h}, \widetilde{g}^a, {g'}^b, Z) \to true]|$ is negligible.*

Next, we define the q-Strong Diffie-Hellman (q-SDH) assumption as follows. Here, we use an asymmetric variant [16].

Definition 2.4 (q-SDH Assumption). *Let $D := (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, \widehat{g})$ and $x \xleftarrow{\$} \mathbb{Z}_p^*$. We say that the q-SDH assumption holds if for any PPT adversary $\mathcal{A}$, the advantage $\mathrm{Adv}_{q\text{-}SDH}(\lambda) := \Pr[\mathcal{A}(D, g^x, g^{x^2}, \ldots, g^{x^q}, \widehat{g}^x) \to (c, g^{1/(x+c)}) \in \mathbb{Z}_p \setminus \{-x\} \times \mathbb{G}_1]$ is negligible.*

Next, we define the discrete logarithm (DL) assumption (on $\mathbb{G}_1$) as follows.

Definition 2.5 (DL Assumption). *Let $D := (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, \widehat{g})$ and $x \xleftarrow{\$} \mathbb{Z}_p^*$. We say that the DL assumption holds if for any PPT adversary $\mathcal{A}$, the advantage $\mathrm{Adv}_{DL}(\lambda) := \Pr[\mathcal{A}(D, g^x) \to x]$ is negligible.*

Next, we introduce the BBS+ signature scheme [6], especially, the BBS+ signature scheme over a type 3 curve [16]. This scheme allows to sign L messages, and is existential unforgeable against chosen message attack under the q-SDH assumption. Let $g, h_0, h_1, \ldots, g_L$ be generators of $\mathbb{G}_1$, $\widehat{g}$ be a generator of $\mathbb{G}_2$, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be a bilinear map.

The BBS+ signature scheme [6]

Key Generation: Choose $\gamma \xleftarrow{\$} \mathbb{Z}_p^*$, and let $w = \widehat{g}^\gamma$. The verification key is $vk = w$, and the secret key is $sk = \gamma$.

Sign: For the messages $(m_1, \ldots, m_L) \in \mathbb{Z}_p^L$, choose $\xi, \zeta \xleftarrow{\$} \mathbb{Z}_p$ and compute $A = (g h_0^\zeta h_1^{m_1} \cdots h_L^{m_L})^{\frac{1}{\xi+\gamma}}$. Output the signature $\sigma = (A, \xi, \zeta)$.

Verify: For a signature $\sigma = (A, \xi, \zeta)$ and messages $(m_1, \ldots, m_L)$, if $e(A, \widehat{g}^\xi vk) = e(g h_0^\zeta h_1^{m_1} \cdots h_L^{m_L}, \widehat{g})$ holds, then output 1, and otherwise output 0.

3. DEFINITION OF GROUP SIGNATURES WITH TIME-BOUND KEYS

In this section, we give the definition of GS-TBK. We mainly follow the definition of Chu et al. [20, 37]. We additionally introduce unforgeability of expiry time of signing keys, backward unlinkability, and non-frameability against malicious group manager. Moreover, our model introduces expiration information ei_t.

In contrast to the model of group signatures [8, 9, 13, 52], a revocation token grt_i is generated when a signer i joins the group. Revocation tokens are modified according to the current time period t. We denote it $grt_{i,t}$, and $grt_{i,t}$ is contained in the revocation list RL_t if the signer i is revoked at t, and is used for the revocation check. We emphasize that if $\tau_i < t$, then $grt_{i,t}$ does not have to be contained in RL_t since expiry time has passed. On the other hand, in the case of VLR group signatures, all $grt_{i,t}, \ldots, grt_{i,T}$ need to be contained in RL_t. So, the size of revocation list can be reduced due to time-bound keys. Moreover, in the model, a signing key gsk_i is associated with an expiry time τ_i, and the signing algorithm takes as input the current time t.

A GS-TBK scheme $\mathcal{GS}\text{-}\mathcal{TBK}$ consists of six algorithms: (GKeyGen, Join/Issue, Revoke, Sign, Verify, Open) which are defined as follows.

Definition 3.1 (Syntax of GS-TBK).

GKeyGen: *The group key generation algorithm takes as input a security parameter $\lambda \in \mathbb{N}$, and outputs a group public*

key gpk and a master secret key msk. Set a registration table $\mathsf{reg} := \emptyset$. We assume that the maximum size of expiry time T also contained in gpk.

Join/Issue: *This is the pair of interactive algorithms which implement the joining protocol run by a user i and the group manager. The joining algorithm* Join *takes as input gpk, whereas the issuing algorithm* Issue *takes as input msk,* reg*, and an expiry time τ_i. Upon successful completion of the protocol, the* Join *algorithm outputs a signing key gsk_i, τ_i, and a user secret key usk_i, and the* Issue *algorithm outputs* reg *where* $\mathsf{reg}[i]$ *stores a revocation token grt_i and τ_i.*

Revoke: *The revocation algorithm takes as input gpk, msk, t,* reg*, and a set of signers to be revoked at t RU_t. Let a set of their revocation tokens grt_i and its expiry time τ_i be $\{\mathsf{reg}[i] := (grt_i, \tau_i)\}$ which are contained in* reg*. Set $\mathsf{RL}_t := \emptyset$. For each i, the algorithm computes $grt_{i,t}$ if τ_i has not passed, i.e., $t < \tau_i$, and stores $grt_{i,t}$ to RL_t. Moreover, the algorithm computes expiration information ei_t. Finally, the algorithm outputs (ei_t, RL_t).*

Sign: *The signing algorithm takes as input gpk, gsk_i, usk_i, a message to be signed m, the current time t, and ei_t, and outputs a signature σ.*

Verify: *The verification algorithm takes as input gpk, t, σ, m, and RL_t, and outputs either* valid *or* invalid*.*

Open: *The opening algorithm takes as input gpk, msk, t,* reg*, σ, m, and RL_t, and outputs the identity of the signer i or $\perp$.*

The correctness is defined as follows. This guarantees that a group signature generated by a signing key with τ_i at time t^*, where $t^* < \tau_i$ and the signer i is not revoked at time t^*, is valid, and the opening result correctly indicates i.

Definition 3.2 (CORRECTNESS). *For any PPT adversary $\mathcal{A}$ and the security parameter $\lambda \in \mathbb{N}$, we define the experiment $\mathrm{Exp}^{corr}_{\mathsf{GS\text{-}TBK},\mathcal{A}}(\lambda)$ as follows.*

$\mathrm{Exp}^{corr}_{\mathsf{GS\text{-}TBK},\mathcal{A}}(\lambda)$:
$(gpk, msk, \mathsf{reg}) \leftarrow \mathsf{GKeyGen}(\lambda)$; $\mathsf{HU} := \emptyset$
$(i, m, t^*) \leftarrow \mathcal{A}^{\mathsf{AddU},\mathsf{RReg},\mathsf{Revoke}}(gpk)$; $i \in \mathsf{HU} \setminus \mathsf{RU}_{t^*}$; $t^* < \tau_i$
$\sigma \leftarrow \mathsf{Sign}(gpk, gsk_i, usk_i, m, t^*, ei_{t^*})$
$j \leftarrow \mathsf{Open}(gpk, msk, t^*, \mathsf{reg}, \sigma, m, \mathsf{RL}_t)$
Return 1 *if the following holds* :
$\mathsf{Verify}(gpk, t^*, \sigma, m, \mathsf{RL}_{t^*}) = \mathsf{invalid} \vee i \neq j$
Otherwise return 0

- AddU*: The add user oracle allows an adversary $\mathcal{A}$ to add honest users to the group. On input an identity i and τ_i, this oracle computes (gsk_i, τ_i) by running* Join/Issue*. i is added to* HU*.*
- RReg*: On input i, the read-registration-table oracle reveals the content of the registration table* $\mathsf{reg}[i]$*.*
- Revoke*: Let $t-1$ be the time that the oracle is called. The revocation oracle allows $\mathcal{A}$ to revoke honest users. On input identities RU_t, this oracle runs $\mathsf{RL}_t \leftarrow \mathsf{Revoke}(gpk, msk, t, \mathsf{reg}, \mathsf{RU}_t)$, and outputs (ei_t, RL_t).*

We say that $\mathcal{GS}$-$\mathcal{TBK}$ is correct if the advantage

$$\mathrm{Adv}^{corr}_{\mathsf{GS},\mathcal{A}}(\lambda) := \Pr[\mathrm{Exp}^{corr}_{\mathsf{GS\text{-}TBK},\mathcal{A}}(\lambda) = 1]$$

is negligible for any PPT adversary $\mathcal{A}$.

The anonymity with backward unlinkability (BU-anonymity) is defined as follows. This guarantees that no signer identity is revealed from signatures even the corresponding signer has been revoked. We follow selfless CPA anonymity [20, 37] where an adversary is allowed to obtain signing keys except the challenge users' keys,[3] and is not allowed to access the open oracle.

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

$\mathrm{Exp}^{bu\text{-}anon}_{\mathsf{GS\text{-}TBK},\mathcal{A}}(\lambda)$:
$b \xleftarrow{\$} \{0,1\}$
$(gpk, msk, \mathsf{reg}) \leftarrow \mathsf{GKeyGen}(\lambda)$
$\mathsf{HU} := \emptyset$; $\mathsf{CU} := \emptyset$; $\mathsf{RU} := \emptyset$
$b' \leftarrow \mathcal{A}^{\mathsf{AddU},\mathsf{WReg},\mathsf{USK},\mathsf{Revoke},\mathsf{GSign},\mathsf{Ch}_b}(gpk)$
Return 1 *if* $b' = b$, *and* 0 *otherwise*

- AddU*: The add user oracle allows an adversary $\mathcal{A}$ to add honest users to the group. On input an identity i and τ_i, this oracle computes (gsk_i, τ_i) by running* Join/Issue*. i is added to* HU*.*
- WReg*: On input i and M, the write-registration-table oracle updates* $\mathsf{reg}[i]$ *to M.*
- USK*: On input i, the user-secret-keys oracle reveals (gsk_i, usk_i) and adds i to* CU*.*
- Revoke*: Let $t-1$ be the time that the oracle is called. The revocation oracle allows $\mathcal{A}$ to revoke honest users. On input identities RU_t, this oracle runs $\mathsf{RL}_t \leftarrow \mathsf{Revoke}(gpk, msk, t, \mathsf{reg}, \mathsf{RU}_t)$, outputs (ei_t, RL_t), adds RU_t to* RU*. Remark that i_0 and i_1 can be revoked if $t^* < t$.*
- GSign*: On input i and m where $i \in \mathsf{HU}$, the signing oracle computes $\sigma \leftarrow \mathsf{Sign}(gpk, gsk_i, usk_i, m, t, ei_t)$ and returns σ. Here, t is the current time that the oracle called.*
- Ch_b*: On input i_0, i_1, where $i_0, i_1 \in \mathsf{HU}$, and m^*, the challenge oracle computes $\sigma^* \leftarrow \mathsf{Sign}(gpk, gsk_{i_b}, usk_{i_b}, m^*, t^*, ei_{t^*})$ and returns σ^*. Here, $i_0, i_1 \notin \mathsf{CU}$, $i_0, i_1 \notin \mathsf{RU}$, $t^* < \tau_{i_0}$, and $t^* < \tau_{i_1}$ must hold.*

We say that $\mathcal{GS}$-$\mathcal{TBK}$ is BU-anonymous if the advantage

$$\mathrm{Adv}^{bu\text{-}anon}_{\mathsf{GS\text{-}TDL},\mathcal{A}}(\lambda) := |\Pr[\mathrm{Exp}^{bu\text{-}anon}_{\mathsf{GS\text{-}TBK},\mathcal{A}}(\lambda) = 1] - 1/2|$$

is negligible for any PPT adversary $\mathcal{A}$.

[3]We call anonymity full anonymity if the adversary is allowed to obtain all signing keys. As a theoretical result, selfless anonymity is weaker than full anonymity since the former can be constructed from one-way functions and NIZK arguments [17] whereas the latter implies pubic key encryption [1, 24, 48]. We employ the selfless anonymity in this paper, as in the previous works [20, 37] and VLR group signature schemes since a revocation token can be computed by a signing key.

The traceability is defined as follows. We mainly follow the definition of [20, 37] except that we additionally consider unforgeability of expiry time of signing keys (the winning condition (4) in the experiment). Traceability guarantees that no adversary who does not have a signing key can compute a valid signature. Moreover, it guarantees that a valid signature can be traced, and no adversary can produce a valid signature using a secret key after the expiry time of the signing key has passed.

Definition 3.4 (TRACEABILITY). *For any PPT adversary $\mathcal{A}$ and a security parameter $\lambda \in \mathbb{N}$, we define the experiment* $\mathrm{Exp}^{trace}_{\mathsf{GS\text{-}TBK},\mathcal{A}}(\lambda)$ *as follows.*

$\mathrm{Exp}^{trace}_{\mathsf{GS\text{-}TBK},\mathcal{A}}(\lambda)$:
$(gpk, msk, \mathsf{reg}) \leftarrow \mathsf{GKeyGen}(\lambda)$; $\mathsf{CU} := \emptyset$; $\mathsf{SSet} := \emptyset$
$(\sigma^*, m^*, t^*) \leftarrow \mathcal{A}^{\mathsf{SndToI},\mathsf{RReg},\mathsf{Revoke},\mathsf{GSign}}(gpk)$
Return 1 *if* $(1) \wedge (2) \wedge ((3) \vee (4))$ *holds* :
(1) $\mathsf{Verify}(gpk, t^*, \sigma^*, m^*, \mathsf{RL}_{t^*}) = \mathsf{valid}$
(2) $(t^*, \sigma^*, m^*) \notin \mathsf{SSet}$
(3) $i \leftarrow \mathsf{Open}(gpk, msk, t^*, \mathsf{reg}, \sigma^*, m^*, \mathsf{RL}_{t^*})$
$\wedge (i \notin \mathsf{CU} \setminus \mathsf{RU}_{t^*} \vee i = \perp)$
(4) $i \leftarrow \mathsf{Open}(gpk, msk, t^*, \mathsf{reg}, \sigma^*, m^*, \mathsf{RL}_{t^*})$
$\wedge i \in \mathsf{CU} \setminus \mathsf{RU}_{t^*} \wedge \tau_i < t^*$
Otherwise return 0

- SndToI*: The send-to-issuer oracle allows $\mathcal{A}$ to engage the joining protocol on behalf of the corrupted user i. Finally, gsk_i, τ_i, and usk_i are given to $\mathcal{A}$, and i is added to* CU.
- RReg*: On input i, the read-registration-table oracle reveals the content of the registration table* $\mathsf{reg}[i]$.
- Revoke*: Let $t-1$ be the time that the oracle is called. The revocation oracle allows $\mathcal{A}$ to revoke honest users. On input identities* RU_t*, this oracle runs* $\mathsf{RL}_t \leftarrow \mathsf{Revoke}(gpk, msk, t, \mathsf{reg}, \mathsf{RU}_t)$*, and outputs* (ei_t, RL_t).
- GSign*: On input i and m, the signing oracle computes* $\sigma \leftarrow \mathsf{Sign}(gpk, gsk_i, usk_i, m, t, ei_t)$*, returns σ, and adds* (t, σ, m) *to* SSet *. Here, t is the current time that the oracle called.*

We say that $\mathcal{GS}$-$\mathcal{TBK}$ is traceable if the advantage

$$\mathrm{Adv}^{trace}_{\mathsf{GS\text{-}TBK},\mathcal{A}}(\lambda) := \Pr[\mathrm{Exp}^{trace}_{\mathsf{GS\text{-}TBK},\mathcal{A}}(\lambda) = 1]$$

is negligible for any PPT adversary $\mathcal{A}$.

The non-frameability is defined as follows. This guarantees that no adversary can produce a valid signature which is traced to an honest signer. Our definition allows that the group manager is corrupted, i.e., an adversary is given msk and is allowed to read reg. Here, honest means that the signer (i^* in the experiment) is added to the group via the SndToU oracle, and the USK oracle for i^* is not called. That is, the adversary does not know usk_{i^*}.

Definition 3.5 (NON-FRAMEABILITY). *For any PPT adversary $\mathcal{A}$ and a security parameter $\lambda \in \mathbb{N}$, we define the experiment* $\mathrm{Exp}^{nf}_{\mathsf{GS\text{-}TBK},\mathcal{A}}(\lambda)$ *as follows.*

$\mathrm{Exp}^{nf}_{\mathsf{GS\text{-}TBK},\mathcal{A}}(\lambda)$:
$(gpk, msk, \mathsf{reg}) \leftarrow \mathsf{GKeyGen}(\lambda)$
$\mathsf{HU} := \emptyset$; $\mathsf{CU} := \emptyset$; $\mathsf{SSet} := \emptyset$
$(\sigma^*, m^*, t^*, i^*) \leftarrow \mathcal{A}^{\mathsf{SndToU},\mathsf{RReg},\mathsf{USK},\mathsf{GSign}}(gpk, msk)$
Return 1 *if the following holds* :
(1) $\mathsf{Verify}(gpk, t^*, \sigma^*, m^*, \mathsf{RL}_{t^*}) = \mathsf{valid}$
(2) $i^* \leftarrow \mathsf{Open}(gpk, msk, t^*, \mathsf{reg}, \sigma^*, m^*, \mathsf{RL}_{t^*})$
(3) $i^* \in \mathsf{HU} \wedge i^* \notin \mathsf{CU} \wedge (t^*, \sigma^*, m^*) \notin \mathsf{SSet}$
Otherwise return 0

- SndToU*: The send-to-user oracle allows $\mathcal{A}$ to engage a joining protocol of the user i on the behalf of the corrupted group manager. i is added to* HU.
- RReg*: On input i, the read-registration-table oracle reveals the content of the registration table* $\mathsf{reg}[i]$.
- USK*: On input i, the user-secret-keys oracle reveals* (gsk_i, usk_i) *and adds i to* CU.
- GSign*: On input i and m, the signing oracle computes* $\sigma \leftarrow \mathsf{Sign}(gpk, gsk_i, usk_i, m, t, ei_t)$*, returns σ, and adds* (t, σ, m) *to* SSet *. Here, t is the current time that the oracle called.*

We say that $\mathcal{GS}$-$\mathcal{TBK}$ is non-frameable if the advantage

$$\mathrm{Adv}^{nf}_{\mathsf{GS\text{-}TBK},\mathcal{A}}(\lambda) := \Pr[\mathrm{Exp}^{nf}_{\mathsf{GS\text{-}TBK},\mathcal{A}}(\lambda) = 1]$$

is negligible for any PPT adversary $\mathcal{A}$.

4. THE PROPOSED GS-TBK SCHEME

In this section, we give the proposed GS-TBK scheme. For the natural revocation, we employ the Ohara et al. revocable group signature scheme [47], and for the premature revocation, we employ the Nakanishi-Funabiki VLR group signature scheme [43]. We slightly modify the Nakanishi-Funabiki scheme since our scheme is constructed over a type 3 curve whereas the Nakanishi-Funabiki scheme is constructed over a type 2 curve. As mentioned in [42, 43], revocation tokens can be implicitly used for tracing signers. That is, the group manager computes $grt_{i,t}$ for all grt_i, and checks the revocation check equation. If the equation holds with $grt_{i,t}$, then the signature is generated by the user i at time T. This methodology is essentially the same as Bichsel et al. [10] and its follow up works [22, 49]. Even though the opportunity of opening is not frequent, this methodology requires $O(N)$-times revocation check procedures for opening where N is the number of total users. Hence, we simply employ the ElGamal encryption that is employed to encrypt a user certificate (A in the scheme). Then, the opening cost is $O(1)$. For employing the ElGamal encryption scheme, we assume that the DDH problem is hard on $\mathbb{G}_1$ (which naturally holds since we employ type 3 curves).

High Level Description of Our Revocation Methods: Before giving our scheme, we give a high level description of the natural revocation and premature revocation respectively. First, we give a high level description of the natural revocation as follows. The group manager has two signing keys of the BBS+ signature scheme, γ_A

and γ_B. Time information is managed by a binary tree BT with T leaf nodes where T is the maximum size of time. Let an expiry time τ be associated to a leaf node η. Let $\mathsf{Path}(\eta) := (u_1, u_2, \ldots, u_\ell)$, where u_1 is the root node, $u_\ell = \eta$, and $\ell = \log T$. Assume that each u_i is encoded in a $\mathbb{Z}_p$ element. Then, a signer whose expiry time τ has a certificate $\{(A_j, \xi_j, \zeta_j)\}_{j\in[1,\ell]}$ as a signing key gsk where $A_j = (gh_0^{\zeta_j} h_1^{u_j} X)^{\frac{1}{\xi_j+\gamma_A}}$. Here, $X = h_2^x$, and $x = usk$ is known by the signer only (an output of the Join algorithm). ζ_j and ξ_j are random values, and g, h_0, and h_1 are public values. Each A_j is a BBS+ signature of two messages, the node $u_j \in \mathsf{Path}(\eta)$ and x. At the time t, the group manager runs the $\mathsf{CS\text{-}TBK}(\mathsf{BT}, t)$ algorithm, and let $Y := (v_1, v_2, \ldots, v_{\mathsf{num}})$ be the output of the algorithm. Expiration information ei_t contains $\{(B_{i,t}, \xi_i', \zeta_i')\}_{i\in[1,\mathsf{num}]}$ where $B_{i,t} = (gh_0^{\zeta_i'} h_1^{v_i} h_2^t)^{\frac{1}{\xi_i'+\gamma_B}}$. Each $B_{i,t}$ is a BBS+ signature of two messages $v_i \in Y$ and the current time t. Due to the CS method, if $\tau < t$, then there exists a node $u \in \mathsf{Path}(\eta) \cap Y$. So, a non-revoked signer can prove that there exist two signatures of the same node u contained in own gsk and ei_t respectively by using zero-knowledge proofs. Remark that if a signer whose expiry time $\tau > t$ tries to compute a valid group signature, then the signer needs to prepare the corresponding BBS+ signature B. This contradicts unforgeability of the BBS+ signature scheme, and thus unforgeability of expiry time of signing keys is guaranteed.

Second, we give a high level description of the premature revocation as follows. In the $\mathsf{Join}/\mathsf{Issue}$ phase, the group manager stores a revocation token $grt_i = \widetilde{X}_i$ where $\widetilde{X}_i = \widetilde{g}^{x_i} \in \mathbb{G}_1$ and $x_i = usk_i$. At the time t, the group manager chooses $y_t \xleftarrow{\$} \mathbb{Z}_p^*$, and sets $\widetilde{h}_t = \widetilde{g}^{y_t}$ and $\widehat{h}_t := \widehat{g}^{y_t} \in \mathbb{G}_2$. Then, $e(\widetilde{h}_t, \widehat{g}) = e(\widetilde{g}^{y_t}, \widehat{g}) = e(\widetilde{g}, \widehat{g}^{y_t}) = e(\widetilde{g}, \widehat{h}_t)$ hold. A group signature σ contains $\widetilde{h}_t^\beta$, $\widetilde{g}^{d(x_i+\beta)}$, $\widetilde{g}^d$ and $\widehat{g}^d$ where β and d are randomness chosen by the signer. If a signer i is prematurely revoked, then the group manager computes $grt_{i,t} := grt_i^{y_t} = \widetilde{h}_t^{x_i}$, and stores $grt_{i,t}$ to RL_t. Then, $grt_{i,t}$ satisfies $e(grt_{i,t}\widetilde{h}_t^\beta, \widehat{g}^d) = e(\widetilde{h}_t^{x_i+\beta}, \widehat{g}^d) = e(\widetilde{g}^{y_t(x_i+\beta)}, \widehat{g}^d) = e(\widetilde{g}^{d(x_i+\beta)}, \widehat{g}^{y_t}) = e(\widetilde{g}^{d(x_i+\beta)}, \widehat{h}_t)$. By checking whether the equation holds for each $grt_{i,t}$ one by one, the verifier can check whether the signer is prematurely revoked or not. The randomness d, chosen in each signing, prevents to link two signatures generated by the same signer at the same time period.

Next, we give our scheme as follows.

Proposed GS-TBK scheme

$\mathsf{GKeyGen}(\lambda)$**:** Choose $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, \widehat{g})$ where g and $\widehat{g}$ are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. Choose $f, \widetilde{g}, g_2, h_0, h_1, h_2 \xleftarrow{\$} \mathbb{G}_1$, $\gamma_A, \gamma_B, \gamma_O \xleftarrow{\$} \mathbb{Z}_p$, and a hash function $H : \{0,1\}^* \to \mathbb{Z}_p$ and $H' : \{0,1\}^* \to \mathbb{Z}_p$ (which are modeled as random oracles in the security proof). Set $vk_A = \widehat{g}^{\gamma_A}$, $vk_B = \widehat{g}^{\gamma_B}$, $g_1 = f^{\gamma_O}$, and $\mathsf{reg} = \emptyset$. Output $gpk = ((\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, \widehat{g}), f, \widetilde{g}, g_1, g_2, h_0, h_1, h_2, vk_A, vk_B, H, H')$ and $msk = (\gamma_A, \gamma_B, \gamma_O)$.

$\mathsf{Join}(gpk)/\mathsf{Issue}(msk, \mathsf{reg}, \tau_i)$**:**

- A signer i chooses $x_i \xleftarrow{\$} \mathbb{Z}_p$, sets $usk_i = x_i$, computes $X_i = h_2^{x_i}$ and $\widetilde{X}_i = \widetilde{g}^{x_i}$, and sends $(X_i, \widetilde{X}_i)$ to the group manager. Moreover, the signer i proves the knowledge of x_i to the group manager as follows.
  - The signer chooses $r_x \xleftarrow{\$} \mathbb{Z}_p$, computes $R = h_2^{r_x}$, $\widetilde{R} = \widetilde{g}^{r_x}$, $c_x \leftarrow H'(X_i, \widetilde{X}_i, R, \widetilde{R})$, and $s_x = r_x + c_x x_i$, and sends (s_x, c_x) to the group manager.
  - The group manager checks whether $c_x = H'(X_i, \widetilde{X}_i, h_2^{s_x}/X_i^{c_x}, \widetilde{g}^{s_x}/\widetilde{X}_i^{c_x})$.
- The group manager assigns a leaf node η to τ_i. Remark that if the same time has been assigned to another signer before, then the same leaf node is selected. For all $u_j \in \mathsf{Path}(\eta) := (u_1, u_2, \ldots, u_\ell)$, the group manager computes BBS+ signatures $\{(A_j, \xi_j, \zeta_j)\}_{j\in[1,\ell]}$ where $A_j = (gh_0^{\zeta_j} h_1^{u_j} X_i)^{\frac{1}{\xi_j+\gamma_A}}$, and sends $gsk_i = (\{(A_j, \xi_j, \zeta_j), u_i\}_{j\in[1,\ell]})$ and τ_i to the signer i.
- The group manager sets $grt_i = \widetilde{X}_i$ and stores $(\tau_i, grt_i, \{A_j\}_{j\in[1,\ell]})$ to $\mathsf{reg}[i]$.

$\mathsf{Revoke}(gpk, msk, t, \mathsf{reg}, \mathsf{RU}_t)$: Choose $y_t \xleftarrow{\$} \mathbb{Z}_p^*$, and compute $\widetilde{h}_t = \widetilde{g}^{y_t}$ and $\widehat{h}_t = \widehat{g}^{y_t}$.

Generating Expiration Information: For the current time t, obtain $Y := (v_1, v_2, \ldots, v_{\mathsf{num}}) \leftarrow \mathsf{CS\text{-}TBK}(\mathsf{BT}, t)$. Compute BBS+ signatures $\{(B_{i,t}, \xi_i', \zeta_i')\}_{i\in[1,\mathsf{num}]}$ where $B_{i,t} = (gh_0^{\zeta_i'} h_1^{v_i} h_2^t)^{\frac{1}{\xi_i'+\gamma_B}}$. Set $ei_t = (\widetilde{h}_t, \{(B_{i,t}, \xi_i', \zeta_i'), v_i\}_{i\in[1,\mathsf{num}]})$.

Generating Revocation List: For all $i \in \mathsf{RU}_t$, compute $grt_{i,t} = grt_i^{y_t}$ and set $\mathsf{RL}_t = (\widetilde{h}_t, \widehat{h}_t, \{grt_{i,t}\}_{i\in\mathsf{RU}_t})$.

Output (ei_t, RL_t).

$\mathsf{Sign}(gpk, gsk_i, usk_i, m, t, ei_t)$**:** Assume that $t < \tau_i$. Then, there exists a node u such that $((A, \xi, \zeta), u)$ is contained in gsk_i, where $A = (gh_0^\zeta h_1^u X_i)^{\frac{1}{\xi+\gamma_A}}$, and $((B_t, \xi', \zeta'), u)$ is contained in ei_t, where $B_t = (gh_0^{\zeta'} h_1^u h_2^t)^{\frac{1}{\xi'+\gamma_B}}$. Choose $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$ and compute

$$\psi_1 = f^\alpha, \psi_2 = Ag_1^\alpha, \text{ and } \psi_3 = B_t g_2^\alpha$$

Here, (ψ_1, ψ_2, ψ_3) is an ElGamal ciphertext with Kurosawa's randomness reuse technique [29]. Choose $\beta, d \xleftarrow{\$} \mathbb{Z}_p^*$ and compute

$$\psi_4 = \widetilde{h}_t^\beta, \psi_5 = \widetilde{g}^{d(x_i+\beta)}, \psi_6 = \widetilde{g}^d, \text{ and } \psi_7 = \widehat{g}^d$$

Here, $(\psi_4, \psi_5, \psi_6, \psi_7)$ is for the premature revocation check. Set $\delta = \alpha\xi$ and $\delta' = \alpha\xi'$, and compute a signature of proof of knowledge (SPK) V where

$$V = SPK\{(\alpha, \beta, \zeta, \xi, \zeta', \xi', u, x_i, \delta, \delta') :$$
$$\frac{e(\psi_2, vk_A)}{e(g, \widehat{g})} = \frac{e(h_0, \widehat{g})^\zeta e(h_1, \widehat{g})^u e(h_2, \widehat{g})^{x_i} e(g_1, vk_A)^\alpha e(g_1, \widehat{g})^\delta}{e(\psi_2, \widehat{g})^\xi}$$
$$\wedge \frac{e(\psi_3, vk_B)}{e(g, \widehat{g}) e(h_2, \widehat{g})^t} = \frac{e(h_0, \widehat{g})^{\zeta'} e(h_1, \widehat{g})^u e(g_2, vk_B)^\alpha e(g_2, \widehat{g})^{\delta'}}{e(\psi_3, \widehat{g})^{\xi'}}$$
$$\wedge \psi_1 = f^\alpha \wedge \psi_1^\xi f^{-\delta} = 1 \wedge \psi_1^{\xi'} f^{-\delta'} = 1 \wedge \psi_4 = \widetilde{h}_t^\beta$$
$$\wedge \psi_5 = \psi_6^{x_i+\beta}\}(m)$$

as follows. Remark that the current time t is not hidden and is not a witness. Moreover, ψ_7 is not explicitly included in the statement of V since the validity of ψ_7 can be verified by checking $e(\psi_6, \widehat{g}) = e(\widetilde{g}, \psi_7)$ holds.

- Choose $r_\alpha, r_\beta, r_\zeta, r_\xi, r_{\zeta'}, r_{\xi'}, r_u, r_x, r_\delta, r_{\delta'} \xleftarrow{\$} \mathbb{Z}_p^*$.
- Compute

$$R_1 = e(h_0, \widehat{g})^{r_\zeta} e(h_1, \widehat{g})^{r_u} e(h_2, \widehat{g})^{r_x} \times e(g_1, vk_A)^{r_\alpha} e(g_1, \widehat{g})^{r_\delta} e(\psi_2, \widehat{g})^{-r_\xi}$$
$$R_2 = e(h_0, \widehat{g})^{r_{\zeta'}} e(h_1, \widehat{g})^{r_u} e(g_2, vk_B)^{r_\alpha} \times e(g_2, \widehat{g})^{r_{\delta'}} e(\psi_3, \widehat{g})^{-r_{\xi'}}$$
$$R_3 = \psi_1^{r_\xi} f^{-r_\delta},\ R_4 = \psi_1^{r_{\xi'}} f^{-r_{\delta'}}$$
$$R_5 = \widetilde{h}_t^{r_\beta},\ R_6 = \psi_6^{r_x + r_\beta}$$

- Compute $c \leftarrow H(\psi_1, \ldots, \psi_7, R_1, \ldots, R_6, m)$.
- Compute $s_\alpha = r_\alpha + c\alpha$, $s_\beta = r_\beta + c\beta$, $s_\zeta = r_\zeta + c\zeta$, $s_\xi = r_\xi + c\xi$, $s_{\zeta'} = r_{\zeta'} + c\zeta'$, $s_{\xi'} = r_{\xi'} + c\xi'$, $s_u = r_u + cu$, $s_x = r_x + cx_i$, $s_\delta = r_\delta + c\delta$, and $s_{\delta'} = r_{\delta'} + c\delta'$.

Output $\sigma = (\psi_1, \ldots, \psi_7, c, s_\alpha, s_\beta, s_\zeta, s_\xi, s_{\zeta'}, s_{\xi'}, s_u, s_x, s_\delta, s_{\delta'})$.

$\mathsf{Verify}(gpk, t, \sigma, m, \mathsf{RL}_t)$**:** Parse $\mathsf{RL}_t = (\widetilde{h}_t, \widehat{h}_t, \{grt_{i,t}\}_{i \in \mathsf{RU}_t})$.

Verification Check: If $e(\psi_6, \widehat{g}) \neq e(\widetilde{g}, \psi_7)$, then output invalid. Otherwise, compute

$$R'_1 = e(h_0, \widehat{g})^{s_\zeta} e(h_1, \widehat{g})^{s_u} e(h_2, \widehat{g})^{s_x} e(g_1, vk_A)^{s_\alpha} \times e(g_1, \widehat{g})^{s_\delta} e(\psi_2, \widehat{g})^{-s_\xi} \left(\frac{e(\psi_2, vk_A)}{e(g, \widehat{g})}\right)^{-c}$$
$$R'_2 = e(h_0, \widehat{g})^{s_{\zeta'}} e(h_1, \widehat{g})^{s_u} e(g_2, vk_B)^{s_\alpha} e(g_2, \widehat{g})^{s_{\delta'}} \times e(\psi_3, \widehat{g})^{-s_{\xi'}} \left(\frac{e(\psi_3, vk_B)}{e(g, \widehat{g}) e(h_2, \widehat{g})^t}\right)^{-c}$$
$$R'_3 = \psi_1^{s_\xi} f^{-s_\delta},\ R'_4 = \psi_1^{s_{\xi'}} f^{-s_{\delta'}}$$
$$R'_5 = \widetilde{h}_t^{s_\beta} \psi_4^{-c},\ R'_6 = \psi_6^{s_x + s_\beta} \psi_5^{-c}$$

If $c \neq H(\psi_1, \ldots, \psi_7, R'_1, \ldots, R'_6, m)$, then output invalid.

Revocation Check: If there exists $grt_{i,t}$ such that $e(grt_{i,t}\psi_4, \psi_7) = e(\psi_5, \widehat{h}_t)$ holds, then output invalid.

Otherwise, output valid.

$\mathsf{Open}(gpk, msk, t, \mathsf{reg}, \sigma, m, \mathsf{RL}_t)$**:** If invalid $\leftarrow \mathsf{Verify}(gpk, t, \sigma, m, \mathsf{RL}_t)$, then output $\perp$. Otherwise, parse $\sigma = (\psi_1, \ldots, \psi_7, c, s_\alpha, s_\beta, s_\zeta, s_\xi, s_{\zeta'}, s_{\xi'}, s_u, s_x, s_\delta, s_{\delta'})$ and $msk = (\gamma_A, \gamma_B, \gamma_O)$. Compute $A = \psi_2 / \psi_1^{\gamma_O}$, search i such that $\mathsf{reg}[i]$ contains A, and output i. If no such an entry exists, then output $\perp$.

Security Analysis. Here, we show that the proposed scheme is BU-anonymous, traceable, and non-frameable.

Theorem 4.1. *The proposed GS-TBK scheme satisfies BU-anonymity if the DDH1 assumption and the DLIN assumption hold in the random oracle model.*

We define the following games.

Game 0: This is the same as the definition of BU-anonymity.

Game 1: This game is the same as Game 0 except for the challenge signature is computed by programming of the random oracle H.

Game 2: This game is the same as Game 0 except for the challenge signature $\sigma^* = (\psi_1^*, \psi_2^*, \psi_3^*, \psi_4^*, \psi_5^*, \psi_6^*, \psi_7^*, c^*, s_\alpha^*, s_\beta^*, s_\zeta^*, s_\xi^*, s_{\zeta'}^*, s_{\xi'}^*, s_u^*, s_x^*, s_\delta^*, s_{\delta'}^*)$, $\psi_2^*, \psi_3^* \xleftarrow{\$} \mathbb{G}_1$.

Game 3: This game is the same as Game 1 except for the challenge signature $\sigma^* = (\psi_1^*, \psi_2^*, \psi_3^*, \psi_4^*, \psi_5^*, \psi_6^*, \psi_7^*, c^*, s_\alpha^*, s_\beta^*, s_\zeta^*, s_\xi^*, s_{\zeta'}^*, s_{\xi'}^*, s_u^*, s_x^*, s_\delta^*, s_{\delta'}^*)$, $\psi_5^* \xleftarrow{\$} \mathbb{G}_1$.

Let S_i be the event that $\mathcal{A}$ successfully guesses b in Game i.

Lemma 4.1. $|\Pr[S_0] - \Pr[S_1]| \leq (1 - q_h/p)$ *where* q_h *the number of hash queries.*

Proof. In Game 1, for computing the challenge signature, first compute $\psi_1^*, \ldots, \psi_7^*$ as in the scheme. Next, randomly choose $c, s_\alpha, s_\beta, s_\zeta, s_\xi, s_{\zeta'}, s_{\xi'}, s_u, s_x, s_\delta, s_{\delta'} \xleftarrow{\$} \mathbb{Z}_p$, and compute $R'_1 = e(h_0, \widehat{g})^{s_\zeta} e(h_1, \widehat{g})^{s_u} e(h_2, \widehat{g})^{s_x} e(g_1, vk_A)^{s_\alpha} e(g_1, \widehat{g})^{s_\delta} e(\psi_2, \widehat{g})^{-s_\xi} \left(\frac{e(\psi_2, vk_A)}{e(g, \widehat{g})}\right)^{-c}$, $R'_2 = e(h_0, \widehat{g})^{s_{\zeta'}} e(h_1, \widehat{g})^{s_u} e(g_2, vk_B)^{s_\alpha} e(g_2, \widehat{g})^{s_{\delta'}} e(\psi_3, \widehat{g})^{-s_{\xi'}} \left(\frac{e(\psi_3, vk_B)}{e(g, \widehat{g}) e(h_2, \widehat{g})^t}\right)^{-c}$, $R'_3 = \psi_1^{s_\xi} f^{-s_\delta}$, $R'_4 = \psi_1^{s_{\xi'}} f^{-s_{\delta'}}$, $R'_5 = \widetilde{h}_t^{s_\beta} \psi_4^{-c}$, and $R'_6 = \psi_6^{s_x + s_\beta} \psi_5^{-c}$. Next, programming the random oracle H such that $c := H(\psi_1, \ldots, \psi_7, R'_1, \ldots, R'_6, m)$, and send $\sigma = (\psi_1, \ldots, \psi_7, c, s_\alpha, s_\beta, s_\zeta, s_\xi, s_{\zeta'}, s_{\xi'}, s_u, s_x, s_\delta, s_{\delta'})$ to $\mathcal{A}$. If programming is failure (i.e., c colludes with a value returned by H), output a random bit and aborts. If programming is not failure, $\Pr[S_0] = \Pr[S_1]$ holds. Since c is randomly chosen from $\mathbb{Z}_p$, the failed probability is at most q_h/p. □

Lemma 4.2. $|\Pr[S_1] - \Pr[S_2]| \leq \mathrm{Adv}_{DDH1}(\lambda)$ *where* q_h *the number of hash queries.*

Proof. Let $((\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, f, \widehat{f}), f^a, f^b, Z)$ be a DDH1 instance. We construct an algorithm $\mathcal{B}$ that distinguishes $Z = f^{ab}$ or not. $\mathcal{B}$ implicitly sets $\alpha := a$ and $\gamma_O := b$ (thus, $g_1 = f^{\gamma_O} = f^b$). $\mathcal{B}$ chooses $r \xleftarrow{\$} \mathbb{Z}_p$, and sets $g_2 := f^r$. $\mathcal{B}$ chooses all values, except f, g_1, and g_2. Since $\mathcal{B}$ has all secret values, $\mathcal{B}$ can respond all queries issued by $\mathcal{A}$. In the challenge phase, $\mathcal{B}$ selects (A, B_{t^*}) according to the scheme. $\mathcal{B}$ sets $\psi_1^* := f^a$, $\psi_2^* := AZ$, and $\psi_3^* := B_{t^*} Z^r$. $\mathcal{B}$ computes other components, except s_α^* is computed by programming of the random oracle H. If $Z = f^{ab}$, then $\mathcal{B}$ correctly simulates Game 1, and if Z is a random value, then $\mathcal{B}$ correctly simulates Game 2. □

Lemma 4.3. $|\Pr[S_1] - \Pr[S_2]| \leq \mathrm{Adv}_{DLIN}(\lambda)(1/q_A q_R - q_s q_h/p)$ *where* q_A, q_R, q_s, *and* q_h *are the number of* AddU, Revoke, GSign, *and hash queries respectively.*

Proof. Let $((\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \widetilde{g}, \widehat{g}), g', \widehat{g}', h, \widehat{h}, \widetilde{g}^a, g'^b, Z)$ be a DLIN instance. We construct an algorithm $\mathcal{B}$ that distinguishes $Z = h^{a+b}$ or not. $\mathcal{B}$ guesses when the challenge user, say i^*, is added in the group with the probability $1/q_A$, and guesses the challenge time t^* with the probability $1/q_R$. We assume that the guesses are correct. $\mathcal{B}$ chooses $y'_{t^*} \xleftarrow{\$} \mathbb{Z}_p$, and implicitly sets $x_{i^*} := a$, $\beta^* := b$, and $y_{t^*} := y'_{t^*} c$ where

$g' := \widetilde{g}^c$ and $\widehat{g}' := \widehat{g}^c$ for some $c \in \mathbb{Z}_p$. $\mathcal{B}$ chooses $y_t \xleftarrow{\$} \mathbb{Z}_p$ for $t \neq t^*$. Then, $\mathcal{B}$ can compute $(\widetilde{h}_t, \widehat{h}_t)$ as follows.

$$(\widetilde{h}_t, \widehat{h}_t) = \begin{cases} (\widetilde{g}^{y_t}, \widehat{g}^{y_t}) & (t \neq t^*) \\ ((g')^{y'_{t^*}}, (\widehat{g}')^{y'_{t^*}}) & (t = t^*) \end{cases}$$

Since $\mathcal{B}$ has all secret keys of signers, except i^*'s one, $\mathcal{B}$ can respond all queries issued by $\mathcal{A}$ if these are not related to i^*. Hence, we show the simulation of revocation queries $\mathsf{Revoke}(\mathsf{RU}_t)$ where $i^* \in \mathsf{RU}_t$, and signing queries $\mathsf{GSign}(i^*, m)$.

For revocation queries, $\mathcal{B}$ can revoke i^* by computing $grt_{i^*,t} = (\widetilde{g}^a)^{y_t}$ when $t \neq t^*$. Remark that $\mathcal{B}$ is not required to compute grt_{i^*,t^*} since i^* is not revoked at the challenge time t^*. This leads to backward unlinkability.

For signing queries at $t = t^*$, $\mathcal{B}$ randomly chooses $\psi_1, \ldots, \psi_3 \xleftarrow{\$} \mathbb{G}_1$ and $\beta, d \xleftarrow{\$} \mathbb{Z}_p$, and computes $\psi_4 = \widetilde{h}_{t^*}^{\beta}$. Then $\psi_4^{1/y_{t^*}} = ((\widetilde{g}^{y'_{t^*}c})^{\beta})^{1/y'_{t^*}c} = \widetilde{g}^{\beta}$ hold. $\mathcal{B}$ computes $\psi_5 = (\widetilde{g}^a \widetilde{g}^{\beta})^d$, $\psi_6 = \widetilde{g}^d$, and $\psi_7 = \widehat{g}^d$. Now, the revocation check relation $e(\psi_5, \widehat{h}_{t^*}) = e((\widetilde{g}^a \widetilde{g}^{\beta})^d, \widehat{g}^{y'_{t^*}c}) = e((\widetilde{g}^a \psi_4^{1/y_{t^*}})^d, \widehat{g}^{y'_{t^*}c})$ $= e((\widetilde{g}^{y'_{t^*}c})^{x_{i^*}} \psi_4, \widehat{g}^d) = e(grt_{i^*,t^*}\psi_4, \psi_7)$ holds. Other components are computed by programming of the random oracle H.

For signing queries at $t \neq t^*$, $\mathcal{B}$ randomly chooses $\psi_1, \ldots, \psi_4 \xleftarrow{\$} \mathbb{G}_1$ and $d \xleftarrow{\$} \mathbb{Z}_p$, and computes $\psi_5 := (\widetilde{g}^a \cdot \psi_4^{1/y_t})^d$, $\psi_6 = \widetilde{g}^d$, and $\psi_7 = \widehat{g}^d$. Then, the revocation check relation $e(\psi_5, \widehat{h}_t) = e((\widetilde{g}^a \cdot \psi_4^{1/y_t})^d, \widehat{g}^{y_t}) = e((\widetilde{g}^a)^{y_t}\psi_4, \widehat{g}^d) = e(grt_{i^*,t}\psi_4, \psi_7)$ holds. Other components are computed by programming of the random oracle H.

For computing the challenge signature, $\mathcal{B}$ chooses $\psi_1^*, \psi_2^*, \psi_3^* \xleftarrow{\$} \mathbb{G}_1$, computes $\psi_4^* = (g'^b)^{y'_{t^*}} = (\widetilde{g}^{cb})^{y'_{t^*}} = (\widetilde{g}^{y'_{t^*}c})^b = \widetilde{h}_{t^*}^{\beta^*}$, and sets $\psi_5^* = Z$, $\psi_6^* = h$, and $\psi_7^* = \widehat{h}$. Other components are computed by programming of the random oracle H. If $Z = h^{a+b}$, then $\mathcal{B}$ correctly simulates Game 2, and if Z is a random value, then $\mathcal{B}$ correctly simulates Game 3. □

Since now the challenge signature does not depend on the challenge bit, $\Pr[S_3] = 1/2$. This concludes the proof. □

Theorem 4.2. *The proposed GS-TBK scheme satisfies traceability in the random oracle model under the q-SDH assumption and the knowledge of secret key (KOSK) assumption.*

As in the Ohara et al. scheme, we introduce the KOSK assumption [51] where the adversary is required to reveal the secret key of the honest users. The reason why we need to introduce the assumption is explained as follows. In the Join algorithm, a user sends $X_i = h_2^{x_i}$ (and $\widetilde{X}_i = \widetilde{g}^{x_i}$ also). The group manager signs x_i by using the signing key of the BBS+ signature scheme such that $A_j = (gh_0^{\zeta_j} h_1^{u_j} X_i)^{\frac{1}{\xi_j + \gamma_A}}$. Due to the form of the BBS+ signature scheme, the group manager can sign x_i without knowing x_i. On the other hand, in the security proof, the simulator needs to send a signed message x_i in order to send a signing query to the signing oracle of the underlying BBS+ signature scheme. So, we use the KOSK assumption.

We can construct an algorithm that extracts a BBS+ signature by applying the Forking lemma [50]. More precisely, from the winning condition $i \notin \mathsf{CU} \setminus \mathsf{RU}_{t^*}$, an adversary needs to produce a forged group certificate A that is not issued via the SndToI oracle, or needs to produce a forged certificate of non-revoked signers B that is not generated when the Revoke oracle is called. Since the signature output by the adversary is valid, forged BBS+ signatures are extracted from the signature. Unforgeability of expiry time of signing keys is also reduced to unforgeability of the BBS+ signature scheme. That is, if an adversary can produce a valid signature though an expiry time τ_i has passed, i.e., $\tau_i < t^*$, then there exists a BBS+ signature B which is valid and is not contained in RL_{t^*}. So, we can construct an algorithm that extracts such B by applying the Forking lemma. Thus, if the extraction works well, then Theorem 4.2 holds. We prove that the following lemma for these extractions.

Lemma 4.4. *The SPK V proves the knowledge $\alpha, \beta, \zeta, \xi, \zeta', \xi', u, x_i, \delta, \delta'$ such that $\psi_1 = f^{\alpha}$, $\psi_2 = (gh_0^{\zeta} h_1^u h_2^{x_i} g_1^{\gamma_A \alpha + \delta})^{\frac{1}{\xi + \gamma_A}}$, $\psi_3 = (gh_0^{\zeta'} h_1^u h_2^t g_2^{\gamma_B \alpha + \delta'})^{\frac{1}{\xi' + \gamma_B}}$, $\psi_4 = \widetilde{h}_t^{\beta}$, and $\psi_5 = \psi_6^{x_i + \beta}$.*

Proof. By the knowledge extractor for V, we can obtain $\alpha, \beta, \zeta, \xi, \zeta', \xi', u, x_i, \delta, \delta'$ such that

$$\frac{e(\psi_2, vk_A)}{e(g, \widehat{g})} = \frac{e(h_0, \widehat{g})^{\zeta} e(h_1, \widehat{g})^u e(h_2, \widehat{g})^{x_i} e(g_1, vk_A)^{\alpha} e(g_1, \widehat{g})^{\delta}}{e(\psi_2, \widehat{g})^{\xi}} \quad (1)$$

$$\frac{e(\psi_3, vk_B)}{e(g, \widehat{g}) e(h_2, \widehat{g})^t} = \frac{e(h_0, \widehat{g})^{\zeta'} e(h_1, \widehat{g})^u e(g_2, vk_B)^{\alpha} e(g_2, \widehat{g})^{\delta'}}{e(\psi_3, \widehat{g})^{\xi'}} \quad (2)$$

$$\psi_1 = f^{\alpha} \quad (3)$$

$$\psi_1^{\xi} f^{-\delta} = 1 \quad (4)$$

$$\psi_1^{\xi'} f^{-\delta'} = 1 \quad (5)$$

$$\psi_4 = \widetilde{h}_t^{\beta} \quad (6)$$

$$\psi_5 = \psi_6^{x_i + \beta} \quad (7)$$

From (1), the equation

$$e(\psi_2, vk_A \widehat{g}^{\xi}) = e(h_0^{\zeta} h_1^u h_2^{x_i}, \widehat{g}) e(g_1, vk_A^{\alpha} \widehat{g}^{\delta}) e(g, \widehat{g})$$

holds. Set $\psi_2 = g^{\theta}$, $h_0 = g_1^{\theta_0}$, $h_1 = g_1^{\theta_1}$, $h_2 = g_1^{\theta_2}$, and $g_1 = g^{\mu}$ for some $\theta, \theta_1, \theta_2, \mu \in \mathbb{Z}_p$. Since $vk_A = \widehat{g}^{\gamma_A}$, $e(g, \widehat{g})^{\theta(\xi + \gamma_A)} = e(g, \widehat{g})^{\mu(\theta_0 \zeta + \theta_1 u + \theta_2 x_i + \gamma_A \alpha + \delta) + 1}$, and thus $\theta(\xi + \gamma_A) = \mu(\theta_0 \zeta + \theta_1 u + \theta_2 x_i + \gamma_A \alpha + \delta) + 1 \bmod p$ holds. This means $\psi_2 = g^{\theta} = (g^{\mu(\theta_0 \zeta + \theta_1 u + \theta_2 x_i + \gamma_A \alpha + \delta) + 1})^{\frac{1}{\xi + \gamma_A}} = (gh_0^{\zeta} h_1^u h_2^{x_i} g_1^{\gamma_A \alpha + \delta})^{\frac{1}{\xi + \gamma_A}}$ holds. Similarly, from (2), $\psi_3 = (gh_0^{\zeta'} h_1^u h_2^t g_2^{\gamma_B \alpha + \delta'})^{\frac{1}{\xi' + \gamma_B}}$ holds. From (3), the extracted α satisfies $\psi_1 = f^{\alpha}$. Then, from (4) and (5), $\delta = \alpha\xi$ and $\delta' = \alpha\xi'$ holds. Finally, from (6) and (7), the extracted x_i and β satisfy $\psi_4 = \widetilde{h}_t^{\beta}$ and $\psi_5 = \psi_6^{x_i + \beta}$. □

Theorem 4.3. *The proposed GS-TBK scheme satisfies non-frameability in the random oracle model under the DL assumption.*

Proof. Let $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \widetilde{g}, \widehat{g}, \widetilde{g}^x)$ be a DL instance. We construct an algorithm $\mathcal{B}$ that breaks the DL problem as follows. Let q_A be the number of SndToU queries. $\mathcal{B}$ guesses the user $i^* \in [1, q_A]$ that $\mathcal{A}$ outputs in the final phase. We assume the guess is correct with the probability $1/q_A$. $\mathcal{B}$ chooses $\theta_2 \xleftarrow{\$} \mathbb{Z}_p$ and sets $h_2 = \widetilde{g}^{\theta_2}$, $usk_{i^*} := x$, and $X_{i^*} := (\widetilde{g}^x)^{\theta_2} = h_2^x$, and $\widetilde{X}_{i^*} := \widetilde{g}^x$. $\mathcal{B}$ chooses other all values as in the scheme. When i^* is added to the group via the SndToU query, $\mathcal{B}$ chooses $s_x, c_x \xleftarrow{\$} \mathbb{Z}_p$, sets $c_x := H'(X_i, \widetilde{X}_{i^*}, h_2^{s_x}/X_{i^*}^{c_x}, \widetilde{g}^{s_x}/\widetilde{X}_{i^*}^{c_x})$, and sends (s_x, c_x) to $\mathcal{A}$. For a signing query $(\cdot, i^*)$, $\mathcal{B}$ programs the random oracle H and

computes a signature. Finally, $\mathcal{A}$ outputs a signature. $\mathcal{B}$ rewinds $\mathcal{A}$ and extracts x^* from the signatures output by $\mathcal{A}$ by applying the Forking lemma [50]. Since the signatures are traced to i^*, the extracted x^* satisfies $X_{i^*} = h_2^{x^*}$. So, $\mathcal{B}$ outputs x^* if the extraction works well. □

5. IMPLEMENTATION

In this section, we give our implementation results. Our implementation environment is as follows: CPU: Xeon E5-2660 v3 @ 2.60GHz, and gcc 4.9.2. We set the maximum size of time T is 2,048, and each day is assigned to a leaf node (thus $\log_2 T = 11$). This setting is the same as that of Liu et al. [37]. Our implementations use the RELIC library (ver.0.4.1) [2] for elliptic curve operations and the pairing operation. We note that we employ asymmetric pairing settings ((type 3) Barreto-Naehrig (BN) curves [7]) with 254-bit order. In this setting, the sizes of a scalar value in $\mathbb{Z}_p$, an element in $\mathbb{G}_1$, an element in $\mathbb{G}_2$, and an element in $\mathbb{G}_T$ are 32 bytes, 33 bytes, 65 bytes, and 256 bytes, respectively. Then, the signature size is 615 bytes, and the size of expiration information is $98 + 105 \log_2 T$ bytes (1,253 bytes when $\log_2 T = 11$).

Next we show benchmarks of algorithms, except Verify, in Table 2. Here, we assume that the pre-computable values, e.g., $e(h_0, \widehat{g})$, are computed in the GKeyGen algorithm. Moreover, we prepare tables for fixed point scalar multiplications in the GKeyGen algorithm. Note that the Revoke algorithm consists of two sub procedures, generating expiration information ei_t and generating revocation list RL_t. The former cost depends on $\log_2 T$, and the latter cost linearly depends on the number of prematurely revoked signers R_{pre}. In the worst case (only the most left leaf is revoked), ei_t consists of $\log_2 T$ BBS+ signatures.

Table 2: Benchmarks (milliseconds)

Algorithms	Benchmarks
GKeyGen	11.395 (incl. 7.704 as pre-computations)
Join	0.287
Issue	3.863
Sign	3.695
Revoke	3.590 $(ei_t)^\dagger$/0.150 $(\mathsf{RL}_t)^\ddagger$

†: The worst case when we set $T = 2,048$.
‡: For prematurely revoking one signer.

It is particularly worth noting that the computational cost of our Sign algorithm is constant in terms of both the time representation and the number of revoked signers. Moreover, the running time of our Sign algorithm is less than 4 msec.

Next, we show the Verify algorithm. Remark that in the usual VLR group signature schemes, the cost of the verification algorithm (more precisely the revocation check) linearly depends on the number of total revoked signers whereas in GS-TBK it just linearly depends on the number of prematurely revoked signers due to time-bound keys. Thus, we show the running time of the Verify algorithm for several numbers of prematurely revoked signers R_{pre}. Recall that the Verify algorithm consists of two sub procedures, the verification check and the revocation check. The former is independent of R_{pre} whereas the latter depends on R_{pre}. Let

$$R_{\mathsf{all}} := R_{\mathsf{pre}} + R_{\mathsf{natural}}$$

be the total number of revoked signers, where R_{natural} be the number of naturally revoked signers, and we set

$$\mathsf{Rate} := R_{\mathsf{pre}}/R_{\mathsf{all}}$$

For example, when $R_{\mathsf{all}} = 1,000,000$ and $\mathsf{Rate} = 0.2$, then $R_{\mathsf{pre}} = 200,000$ and $R_{\mathsf{natural}} = 800,000$.

First, we show the running time of the verification check i.e., the running time of the Verify algorithm with $\mathsf{Rate} = 0$ in Fig 3. The running time is approximately 11.5 msec regardless of R_{all}.

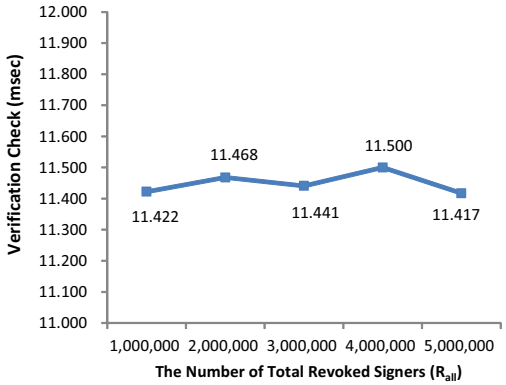

Figure 3: The Running Time of the Verification Check

Next, we show the Verify algorithm for each Rate. We set $R_{\mathsf{all}} = 5,000,000$ and show the running time of the Verify algorithm in Fig 4.

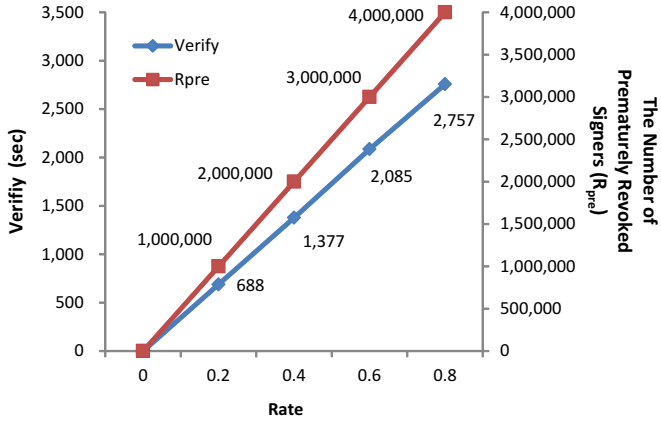

Figure 4: The Running Time of the Verify Algorithm ($R_{\mathsf{all}} = 5,000,000$)

Since the revocation check requires $O(R_{\mathsf{pre}})$-times pairing computations[4], the running time of the Verify algorithm linearly depends on R_{pre}. Nevertheless, as a reasonable assumption, the natural revocation accounts for most of signer revocations in practice and prematurely revoked signers are only a small fraction. Thus, for a relatively small Rate, our scheme is still efficient in practice.

[4]In the previous schemes [20, 37], no pairing computation is required for the revocation check (just $O(R_{\mathsf{pre}})$-times exponentiations are required). Thus, our revocation check is inefficient than those of the previous schemes due to the pairing computations. However, at the expense of this inefficiency, our scheme provides backward unlinkability.

As a reference, we show the running time of the Verify algorithm for each Rate and R_{all} in Fig 5.

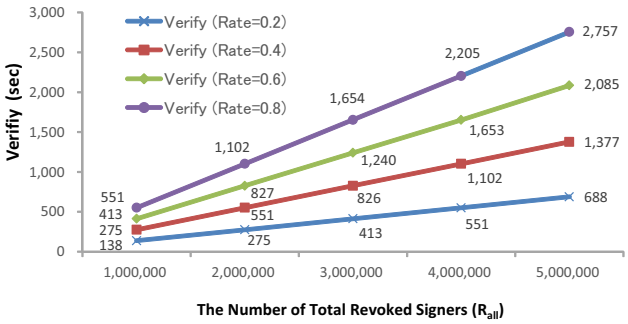

Figure 5: The Running Time of the Verify Algorithm (Rate = 0.2, 0.4, 0.6, 0.8)

6. CONCLUSION

In this paper, we revisit the definition of GS-TBK given in [20, 37], and give a new security model that considers unforgeability of expiry time of signing keys. Moreover, the computational cost of our signing algorithm is constant whereas those of the previous schemes depend on the bit-length of the time representation. We also give implementations.

Our GS-TBK scheme and previous schemes are secure in the random oracle model. Since we employ the Ohara et al. revocable group signature scheme which implements the Libert-Peters-Yung revocable group signature scheme [33] in the random oracle model, we might be able to employ the Libert-Peters-Yung revocable group signature scheme for implementing time-bound keys. Moreover, as mentioned by Libert and Vergnaud [35], the Nakanishi-Funabiki revocation technique [42] itself does not depend on random oracles. Thus, a GS-TBK scheme in the standard model might be constructed by employing the Libert-Peters-Yung and the Nakanishi-Funabiki schemes. Though it is theoretically interest, on the other hand, there is room for argument on the efficiency of the scheme. Since a signature of the Libert-Peters-Yung scheme contains about 100 group elements, it seems difficult to achieve a practical efficiency. Hence, we pursue a practical efficiency in this paper, and leave the standard model construction as a future work.

In addition to the standard model construction, removing pairing computations from the revocation check, e.g., employing [15], is an interesting future work of this paper.

Acknowledgment: This work was partially supported by JSPS KAKENHI Grant Number JP16K00198.

7. REFERENCES

[1] M. Abdalla and B. Warinschi. On the minimal assumptions of group signature schemes. In *Information and Communications Security*, pages 1–13, 2004.

[2] D. F. Aranha and C. P. L. Gouvêa. RELIC is an Efficient LIbrary for Cryptography. https://github.com/relic-toolkit/relic.

[3] N. Attrapadung, K. Emura, G. Hanaoka, and Y. Sakai. A revocable group signature scheme from identity-based revocation techniques: Achieving constant-size revocation list. In *Applied Cryptography and Network Security*, pages 419–437, 2014.

[4] N. Attrapadung, K. Emura, G. Hanaoka, and Y. Sakai. Revocable group signature with constant-size revocation list. *Comput. J.*, 58(10):2698–2715, 2015.

[5] N. Attrapadung, G. Hanaoka, K. Ogawa, G. Ohtake, H. Watanabe, and S. Yamada. Attribute-based encryption for range attributes. In *Security and Cryptography for Networks*, pages 42–61, 2016.

[6] M. H. Au, W. Susilo, and Y. Mu. Constant-size dynamic k-TAA. In *Security and Cryptography for Networks*, pages 111–125, 2006.

[7] P. S. L. M. Barreto and M. Naehrig. Pairing-friendly elliptic curves of prime order. In *Selected Areas in Cryptography*, pages 319–331, 2005.

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

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

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

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

[12] D. Boneh and H. Shacham. Group signatures with verifier-local revocation. In *ACM CCS*, pages 168–177, 2004.

[13] J. Bootle, A. Cerulli, P. Chaidos, E. Ghadafi, and J. Groth. Foundations of fully dynamic group signatures. In *Applied Cryptography and Network Security*, pages 117–136, 2016.

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

[15] J. Bringer and A. Patey. VLR group signatures - how to achieve both backward unlinkability and efficient revocation checks. In *SECRYPT*, pages 215–220, 2012.

[16] J. Camenisch, M. Drijvers, and A. Lehmann. Anonymous attestation using the strong Diffie Hellman assumption revisited. In *Trust and Trustworthy Computing*, pages 1–20, 2016.

[17] J. Camenisch and J. Groth. Group signatures: Better efficiency and new theoretical aspects. In *Security in Communication Networks*, pages 120–133, 2004.

[18] S. Canard, G. Fuchsbauer, A. Gouget, and F. Laguillaumie. Plaintext-checkable encryption. In *CT-RSA*, pages 332–348, 2012.

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

[20] C. Chu, J. K. Liu, X. Huang, and J. Zhou. Verifier-local revocation group signatures with time-bound keys. In *ASIACCS*, pages 26–27, 2012.

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

[22] D. Derler and D. Slamanig. Fully-anonymous short dynamic group signatures without encryption. Cryptology ePrint Archive, Report 2016/154, 2016. http://eprint.iacr.org/2016/154.

[23] Y. Dodis and N. Fazio. Public key broadcast encryption for stateless receivers. In *ACM DRM*, pages 61–80, 2002.

[24] K. Emura, G. Hanaoka, Y. Sakai, and J. C. N. Schuldt. Group signature implies public-key encryption with non-interactive opening. *Int. J. Inf. Sec.*, 13(1):51–62, 2014.

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

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

[27] M. Green and S. Hohenberger. Universally composable adaptive oblivious transfer. In *ASIACRYPT*, pages 179–197, 2008.

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

[29] K. Kurosawa. Multi-recipient public-key encryption with shortened ciphertext. In *Public Key Cryptography*, pages 48–63, 2002.

[30] B. Libert, S. Ling, K. Nguyen, and H. Wang. Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. In *EUROCRYPT*, pages 1–31, 2016.

[31] B. Libert, F. Mouhartem, T. Peters, and M. Yung. Practical "Signatures with efficient protocols" from simple assumptions. In *ASIACCS*, pages 511–522, 2016.

[32] B. Libert, T. Peters, and M. Yung. Group Signatures with Almost-for-Free Revocation. In *CRYPTO 2012*, pages 571–589, 2012.

[33] B. Libert, T. Peters, and M. Yung. Scalable Group Signatures with Revocation. In *EUROCRYPT*, pages 609–627, 2012.

[34] B. Libert, T. Peters, and M. Yung. Short group signatures via structure-preserving signatures: Standard model security from simple assumptions. In *CRYPTO*, pages 296–316, 2015.

[35] B. Libert and D. Vergnaud. Group signatures with verifier-local revocation and backward unlinkability in the standard model. In *Cryptology and Network Security*, pages 498–517, 2009.

[36] H. Lin and W. Tzeng. An efficient solution to the millionaires' problem based on homomorphic encryption. In *Applied Cryptography and Network Security*, pages 456–466, 2005.

[37] J. K. Liu, C. Chu, S. S. M. Chow, X. Huang, M. H. Au, and J. Zhou. Time-bound anonymous authentication for roaming networks. *IEEE Trans. Information Forensics and Security*, 10(1):178–189, 2015.

[38] L. Malina, J. Hajny, and Z. Martinasek. Efficient group signatures with verifier-local revocation employing a natural expiration. In *SECRYPT*, pages 555–560, 2013.

[39] L. Malina, J. Hajny, and V. Zeman. Light-weight group signatures with time-bound membership. *Security and Communication Networks*, 9(7):599–612, 2016.

[40] A. Miyaji, M. Nakabayashi, and S. Takano. New explicit conditions of elliptic curve traces for FR-reduction. *IEICE Transactions*, 84-A(5):1234–1243, 2001.

[41] T. Nakanishi, H. Fujii, Y. Hira, and N. Funabiki. Revocable Group Signature Schemes with Constant Costs for Signing and Verifying. In *Public Key Cryptography*, pages 463–480, 2009.

[42] T. Nakanishi and N. Funabiki. Verifier-local revocation group signature schemes with backward unlinkability from bilinear maps. In *ASIACRYPT*, pages 533–548, 2005.

[43] T. Nakanishi and N. Funabiki. A short verifier-local revocation group signature scheme with backward unlinkability. In *IWSEC*, pages 17–32, 2006.

[44] T. Nakanishi and N. Funabiki. Revocable group signatures with compact revocation list using accumulators. *IEICE Transactions*, 98-A(1):117–131, 2015.

[45] D. Naor, M. Naor, and J. Lotspiech. Revocation and tracing schemes for stateless receivers. *Electronic Colloquium on Computational Complexity (ECCC)*, (043), 2002.

[46] L. Nguyen. Accumulators from bilinear pairings and applications. In *CT-RSA*, pages 275–292, 2005.

[47] K. Ohara, K. Emura, G. Hanaoka, A. Ishida, K. Ohta, and Y. Sakai. Shortening the Libert-Peters-Yung revocable group signature scheme by using the random oracle methodology. Cryptology ePrint Archive, Report 2016/477, 2016. http://eprint.iacr.org/2016/477.

[48] G. Ohtake, A. Fujii, G. Hanaoka, and K. Ogawa. On the theoretical gap between group signatures with and without unlinkability. In *AFRICACRYPT*, pages 149–166, 2009.

[49] D. Pointcheval and O. Sanders. Short randomizable signatures. In *CT-RSA*, pages 111–126, 2016.

[50] D. Pointcheval and J. Stern. Security proofs for signature schemes. In *EUROCRYPT*, pages 387–398. Springer-Verlag, Berlin, 1996.

[51] T. Ristenpart and S. Yilek. The power of proofs-of-possession: Securing multiparty signatures against rogue-key attacks. In *EUROCRYPT*, pages 228–245, 2007.

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

[53] L. Wei and J. Liu. Shorter verifier-local revocation group signature with backward unlinkability. In *Pairing-Based Cryptography*, pages 136–146, 2010.

[54] S. Zhou and D. Lin. Shorter Verifier-Local Revocation Group Signatures from Bilinear Maps. In *CANS*, pages 126–143, 2006.