Fang and Feng *J Wireless Com Network* (2021) 2021:113
https://doi.org/10.1186/s13638-021-01948-w

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Group signature with time-bound keys and unforgeability of expiry time for smart cities

Junli Fang and Tao Feng*

*Correspondence:
fengt@lut.cn
School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China

Abstract

Internet of Things (IoT) lays the foundation for the various applications in smart cities, yet resource-constrained IoT devices are prone to suffer from devastating cyberattacks and privacy leak threats, thus are inevitability supposed as the weakest link of the systems in smart cities. Mitigating the security risks of data and the computing limitation of edge devices, especially identity authentication and key validity management of group devices are essential for IoT system security. In order to tackle the issues of anonymity, traceability, unforgeability of expiry time as well as efficient membership revocation for life-cycle management of devices in IoT setting, we presented a dynamic time-bound group signature with unforgeability of expiry time. Unforgeability of expiry time disables a revoked signer to create a valid signature by means of associating the signing key with an expiry time. The anonymity and traceability of the proposed scheme contribute to the identity privacy of the entities and supervision for authority agency. Moreover, our proposal is feasible in the resource-constrained setting for efficient computational cost of signing and verification algorithms.

Keywords: Group signatures, Unforgeability of expiry time, Time-bound keys, Internet of things, Smart cities

1 Introduction

Internet of Things (IoT), which is capable of sensing the physical world by ubiquitous smart devices and building a transparent information world, is considered to be one of the most fundamental and indispensable technologies for smart cities [1]. To provide citizens convenient public resource utilization and public administrations services, advanced application-oriented intelligent IoT ecosystems have mushroomed in various industries, such as vehicles, logistics, meteorology, architecture, geology, hydrology, and sharing economy. As a result, the growth spurts on city populations, smart terminals, and data abound to follow in the foreseeable future and will conversely impose great challenges on the city infrastructures and IoT devices. Hence, various solutions including data reduction dimensionality [2], transmission power optimization [3], multi-hop path optimization [4], secure key management [5, 6], and edge computing [7] are proposed to optimize the energy, the real-time, and distributed performance of applications in smart cities.

As described in Fig. 1, IoT devices are usually grouped according to their functionality or locations to perform the tasks on data collection, transmission, and commands execution. Traditional resource-limited IoT devices are typically not secured-by-design and located at the edge of the smart cities ecosystems, thus they are vulnerable to security and privacy threats that are prone to trigger devastating losses [8]. Generally, data confidentiality, integrity, and device authentication in groups must be guaranteed [5]. In addition, anonymity is indispensable because of identity privacy leakage, and traceability is expected when supervision or audit organizations need to find out malicious devices. The optimum solution to the questions mentioned above is the scheme employing group signature [9]. In a group signature, any group member is allowed to generate signatures anonymously representing the entire group, and a signature can be opened to reveal the misbehaved members in case of a dispute. Consequently, group signature has been proved to be an appropriate way to ensure authentication, anonymity, and traceability in numerous privacy-preserving intricate schemes [10–12].

In general, the following security and privacy problems are essential for practical and need to be considered. Firstly, efficient flexible registration and revocation functionality is indispensable for practical purposes due to constantly mobile devices and incompletely reliable signals. Dynamic group signature is more complex but is more efficient and available than static group signature in mobile IoT settings because without frequent initializations. Revocations usually cause the degradation of efficiency. Spontaneously, it is important to speed up revocation checks especially in resource-constrained settings. Moreover, it is necessary to maintain security after group members are revoked. Besides, it is worth especially noting that valid time of data and devices is crucial for lifecycle management and even counting economic value. However, the forgery attack to the expiration time of occupancy will cause exploiting inappropriately data, device or service, and consequently injurious to the interest of stakeholders. In addition, the encryption procedure employed in group signature is heavy for IoT devices and thus is necessary to be removed or reduced using a novel signature scheme. Naturally, how to realize efficient revocable dynamical group signature with unforgeability of expiry time in the IoT setting is a tough but critical question.

1.1 Related work

Group signatures introduced by Chaum and van Heys [9] were strictly formalized as static BMW mode [13] and further extended to the circumstance of the dynamic BSZ model [14]. In the static BMW model setting, the group manager is responsible for opening signatures and generating keys honestly for predefined group members at the setup stage. Yet in the dynamically BSZ model, any new member is allowed to join the group at all moments after finishing the initial setup, while the monolithic group manager in the static model is separated into issuer and opener. These ingenious works have introduced general constructions, which have become the implicit framework for most of the following group signatures. Subsequently, Sakai et al. [15] explored a slightly modified scheme by defining the notion of weak opening soundness, which requires no malicious user can fabricate an opening proof and allege ownership of a signature issued by an honest one. Weak opening soundness is reasonable in the practical setting because

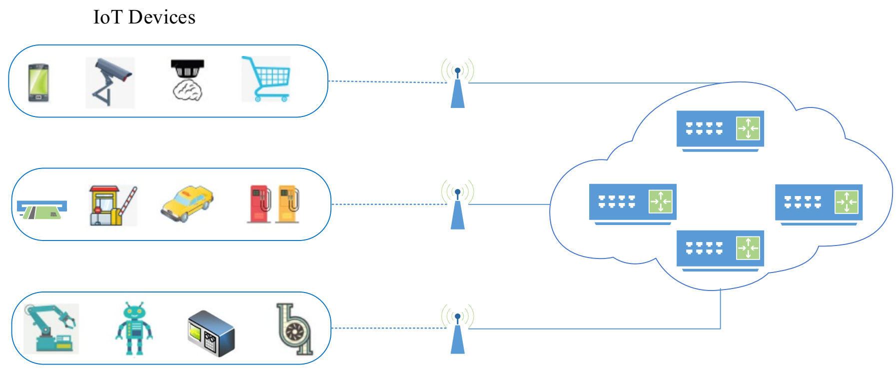

Fig. 1 Overview of the IoT architecture

it achieves an acceptable tradeoff between computational cost and anticipated security guarantees.

The widely used construction paradigm for group signatures is the modular Sign-Encrypt-Prove (SEP) paradigm, which typically consists of three steps. Firstly, the issuer and group members play an interactive protocol to generate the signing key, namely a certificate associated with the identity of a member. Then group member generates a digital signature on the message and the encryption of her identity. Finally, a Non-Interactive Zero-Knowledge proof (NIZK) is provided to prove that the user takes possession of knowledge of a legitimate certificate. Unfortunately, the main drawback of the SEP paradigm is inefficiency due to the complexity of NIZK proof and encryption. To solve this issue, Bichsel et al. [16] explored an efficient alternative called Sign-Randomize-Proof (SRP) paradigm and creatively removed explicit encryption by employing re-randomizable signatures during the group signature generation phase, guarantying that multiple randomized counterparts originated from the identical signature are not linkable. In particular, they improved efficiency by proving a Signature of Knowledge (SoK) on the message instead of NIZK proof. Following this novel paradigm, Derler and Slamanig [17] contributed a highly-efficient dynamic group signature construction that employed structure-preserving signatures on equivalence classes (SPS-EQ) [18, 19]. SPS-EQ defines a relation $\mathcal{R}$ to establish partitions of the message space, which indicates that the signer virtually signs the whole partition as long as signing one representative of a partition. Especially, the SPS-EQ signature can be transformed to any different representative of the partition, without knowing any information of the secret key. It is also noteworthy that the scheme of Derler and Slamanig is especially fit for resource-constrained devices because the signature size of their CPA-fully anonymous instantiation is shorter than the classical BBS scheme [20].

It is indispensable for practical purpose to provide revocation functionality, however, which usually cause the degradation of efficiency. Spontaneously, it is significant to speed up revocation checks, especially in the resource-constrained setting. More precisely, the revocation check (RC) is classified into implicit and explicit revocation. The implicit revocation indicates that a revoked signer cannot compute signatures that passing the verification check, and she needs to prove both that she is unrevoked and enrolled in the group. Hence, in the implicit revocation [21–24], the signing algorithm is computationally expensive, whereas the verification expense is relatively low. Inversely, in the explicit case, all signers can create signatures passing the verification check, but a verifier needs to further run the RC procedure to check if the signer has been revoked or not. Thus, in the explicit revocation [25–28], a signer only proves her membership. Accordingly, the signing algorithm is at a relatively low computational cost, while the cost of the verification part is computationally expensive because of the supplementary RC procedure. Therefore, explicit RC has lower power consumption than the implicit case, for the IoT devices as the data producers.

Typically, the computational cost of RC increases linearly with the scale of the revocation list. Therefore, there is a high demand for a flexible revocation approach to downsizing the revocation list. Libert et al. [22] put forward the classical paradigm for revocable group signature (RGS) solution in the standard model based on a complete subtree algorithm. Unfortunately, the solution fails to achieve sufficient efficiency in the practical setting, due to adopting the complex standard model and Groth–Sahai proofs. We additionally remark that the construction in the asymmetric pairing setting and the random oracle model (ROM) is highly desirable in view of efficiency and suitability for practical resource-constrained context, although that in the standard model or based on lattices are quite attractive. Ohara et al. [24] showed an RGS scheme called parallel BBS group signature and the costs of which are asymptotically identical with that of the LPY scheme [22]. Nonetheless, the cost of computing the signing process in [22] is relatively high due to implicit revocation. Emura and Hayashi [23] proposed an RGS scheme under the simple assumption by employing the methodology in [24]. They modify their proposal to support weak opening soundness since the LPY model is incapable of ownership proof. Ishida et al. [27] came up with a fully anonymous group signature, where revocation component is achieved using additional key pairs of a key-private public key encryption scheme. Their design is not fully dynamic due to following BMW construction and also fail to provide instantiation and efficiency evaluation. Very recently, Yue et al. [29] offered a distributed RGS scheme with backward security by introducing a trusted authority.

Besides, time-bound keys (TBK) management techniques [30], which means that secret key is embedded with a timestamp, are usually combined with group key management, broadcast encryption, group signature, attribute-based encryption for efficient revocation, access control, and anonymous authentication on the time dimension [5, 31–34]. It is crucial to highlight that, for the sake of downsizing the expense of RC in practical settings, Chu et al. [31] detailed a feasible method called group signature with time-bound keys (GS-TBK). In GS-TBK, the signing key of each member is closely related to expiry time, and the verifiers check whether the signers produced group signatures based on expired keys. The proposal could be regarded as a solution possessing the simultaneous properties of both "natural" and "premature" revocation types. The "natural" revocation means that only signers having non-expired keys can create signatures that pass the verification check whereas the "premature" revocation indicates that it is able to revoke signers in advance even expiry times have not passed and thus verifiers need to run the RC procedure. The number of prematurely revoked signers is merely a small proportion of all revoked members, thus, the size of the revocation list and the cost of RC is significantly cut down [10, 34]. Subsequently, Emura et al. [34] revisited the definition the traceability in GS-TBK by offering the unforgeability of expiry time for signing keys [24]. The forgeability attack refers to an adversary may forge a valid signature after expiry time τ. Specifically, [34] defined a complete subtree algorithm for time-bound keys (CS-TBK) similar to the CS method and proposed a novel group signature in the proposed model. Assuming that T represents the maxlength of time and the number of the leaf node belong to the binary tree BT, the subtree covering all nodes that are non-revoked could be found. The underlying primitives of the proposal are BBS + signature [20]. Regarding security properties, the scheme provides backward unlinkability-anonymity, traceability, and non-frameablity. Constant signing cost was provided, unlike the earlier solution where the efficiency of signing depends entirely on the length of bits representing the time. Similarly, Malina et al. [33] and Perera et al. [28] provided group signatures with time-bound membership but not consider premature revocation and fail to resist forgeability signing time and expiry time.

1.2 Motivation and our contribution

To sum up, it is necessary to propose an efficient fully dynamic group signature that provides minimize the revocation verification cost following the SRP paradigm. Our construction is based on a tailored combination of dynamic group signatures scheme following the SRP model in [17] and GS-TBK scheme in [34]. The main contributions are summarized below.

- Efficient flexible registration and revocation functionality is realized by combining with novel dynamic group signatures scheme and GS-TBK scheme.
- The design realizes security against forgery of expiry time and the backward attack, which prohibits revoked signers from generating group signatures associating future periods.
- Relatively low and constant computational cost at the signing stage is provided by employing re-randomizable structure-preserving signatures on equivalence classes (SPS-EQ).
- The cost of verification algorithms, which is linearly correlated with the number of signers prematurely revoked rather than that of total revoked signers, is significantly cut down.
- BU-anonymity, traceability, non-frameability, and weak opening soundness are fully guaranteed.

The remainder of this article is structured as follows. Some related definitions are recalled in Sect. 2. Then Sect. 3 focuses on the formal description of the proposed scheme and security model. Section 4 describes the details of the construction and analysis of security. In Sect. 5, the comparison with the related solutions will be discussed. Finally, Sect. 6 concludes the article.

2 Preliminaries

Next, some cryptographic preliminaries used in this article are recalled. The detailed definitions about the digital signature, public key encryption (PKE), NIZK proof systems, and SoK could refer to [17].

Notation $z \xleftarrow{R} Z$ means that z is chosen randomly from a finite set Z uniformly. Let $z \leftarrow \psi(x)$ denote that is the randomized function ψ with input x and output z.

All algorithms are assumed that be run in polynomial time and output $\perp$ if any error happens. $\Pr[\Omega : E]$ means that the probability of an event E over the probability space Ω. A negligible function $\epsilon : \mathbb{N} \to \mathbb{R}^+$ states that there exists a contain constant $k_0 \in \mathbb{N}$ satisfying $\epsilon(k) < 1/k^c$ for any $k > k_0$ and any positive number c.

Assume $\mathrm{BGGen}(1^\kappa)$ *generates a bilinear group* $\mathrm{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$ *in the Type-3 setting* where $\mathbb{G}_1 \neq \mathbb{G}_2$ and no computable isomorphism $\varphi : \mathbb{G}_2 \to \mathbb{G}_1$.

2.1 Assumptions

Definition 1 The Decisional Diffie–Hellman (DDH) assumption states that any adversary $\mathcal{A}$ is infeasible to break DDH assumption with a negligible function $\epsilon(\kappa)$.That is

$$\Pr\left[\begin{array}{c} m \leftarrow \{0,1\}, x, y, z \leftarrow \mathbb{Z}_p \\ m^* \leftarrow \mathcal{A}(P, xP, yP, (m \cdot (xy) + (1-m) \cdot z)P \end{array} : m = m^*\right] - 1/2 \leq \epsilon(\kappa)$$

where $\log_2 \mathrm{P} = \kappa$ and P is the prime-order of group $\mathbb{G}$.

Definition 2 The Symmetric External Diffie–Hellman (SXDH) assumption states that the DDH assumption holds in $\mathbb{G}_1$ and $\mathbb{G}_2$.

Definition 3 The Computational co-Diffie–Hellman Inversion (co-CDHI) assumption means that any adversary $\mathcal{A}$ is infeasible to break co-CDHI assumption with non-negligible probability during polynomial-time. That is

$$\Pr\left[\begin{array}{c} BG \leftarrow \mathrm{BGGen}(1^\kappa), c \leftarrow \mathbb{Z}_p \\ C \leftarrow \mathcal{A}(BG, cP, 1/cP) \end{array} : C = 1/cP\right] \leq \epsilon(\kappa)$$

2.2 SPS-EQ

Definition 4 An SPS-EQ on $\mathbb{G}_i^*$ $(i \in \{1,2\})$ is defined based on the below algorithms:

$\mathrm{BGGen}(1^\kappa)$: input a security parameter κ, and outputs a bilinear group BG.
$\mathrm{KGen}_{\mathcal{R}}(BG, \ell)$: input BG as well as a vector length ℓ, and outputs a key pair $(sk_{\mathcal{R}}, pk_{\mathcal{R}})$.
$Sign_{\mathcal{R}}(M, sk_{\mathcal{R}})$: input a representative $E \in (\mathbb{G}_i^*)^\ell$ of equivalence classes $[E]_{\mathcal{R}}$ and a secret key $sk_{\mathcal{R}}$, and outputs an SPS-EQ signature.

$\mathrm{ChgRep}_{\mathcal{R}}(E, \sigma, \mu, pk_{\mathcal{R}})$: input a representative $E \in (\mathbb{G}_i^*)^\ell$ of class $[E]_{\mathcal{R}}$, a signature σ for E, a scalar μ, and a public key $pk_{\mathcal{R}}$, finally outputs a fresh message-signature pair (E', σ'), where $E'{=}\mu \cdot E$ is the new representative and σ' is the corresponding updated signature.

$\mathrm{Vrf}_{\mathcal{R}}(E, \sigma, pk_{\mathcal{R}})$: input a representative $E \in (\mathbb{G}_i^*)^\ell$, a signature σ, and a public key $pk_{\mathcal{R}}$, finally outputs 0 or 1.

$Vkey_{\mathcal{R}}(sk_{\mathcal{R}}, pk_{\mathcal{R}})$: input a secret key $sk_{\mathcal{R}}$ and a public key $pk_{\mathcal{R}}$, finally outputs 0 or 1.

Definition 5 Correctness is achieved for an SPS-EQ scheme on $(\mathbb{G}_i^*)^\ell$, if

$$\begin{aligned} Vkey_{\mathcal{R}}(sk_{\mathcal{R}}, pk_{\mathcal{R}}) &= 1 \wedge \Pr[\mathrm{Vrf}_{\mathcal{R}}(E, \mathrm{Sign}_{\mathcal{R}}(E, sk_{\mathcal{R}}), pk_{\mathcal{R}}) = 1] \\ &= 1 \wedge \Pr[\mathrm{Vrf}_{\mathcal{R}}(\mathrm{ChgRep}_{\mathcal{R}}(E, \mathrm{Sign}_{\mathcal{R}}(E, sk_{\mathcal{R}}), \mu, pk_{\mathcal{R}}), pk_{\mathcal{R}}) = 1] \\ &= 1 \end{aligned}$$

where $\kappa \in \mathbb{N}$, $\ell > 1$, $\mathrm{BG} \leftarrow \mathrm{BGGen}_R(1^\kappa)$, $(sk_{\mathcal{R}}, pk_{\mathcal{R}}) \leftarrow \mathrm{KGen}_{\mathcal{R}}(BG, \ell)$, $E \in (\mathbb{G}_i^*)^\ell$, and $\mu \in \mathbb{Z}_p^*$.

Definition 6 Existential unforgeablity under adaptive chosen-message attacks (EUF-CMA) is achieved for an SPS-EQ scheme on $(\mathbb{G}_i^*)^\ell$, if a negligible function $\epsilon(\cdot)$ exits for any PPT adversary able to access to a signing oracle $\mathcal{O}^{\mathrm{Sign}_{\mathcal{R}}}$ such that:

$$\Pr\left[\begin{array}{c} \mathrm{BG} \leftarrow \mathrm{BGGen}_R(1^\kappa) \\ (sk_{\mathcal{R}}, pk_{\mathcal{R}}) \leftarrow \mathrm{KGen}_{\mathcal{R}}(\mathrm{BG}, \ell)L \\ (E^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}^{\mathrm{Sign}_{\mathcal{R}}(sk, \cdot)}}(pk_{\mathcal{R}}) \end{array} \begin{array}{c} [E*]_{\mathcal{R}} \neq [E]_{\mathcal{R}} \, \forall E \in Q^{\mathrm{Sign}_{\mathcal{R}}} \\ \wedge \mathrm{Vrf}_{\mathcal{R}}(E^*, \sigma^*, pk_{\mathcal{R}}) = 1 \end{array}\right] \leq \epsilon(\kappa)$$

where $Q^{\mathrm{Sign}_{\mathcal{R}}}$ is the set of queries that adversary has sent to the signing oracle $\mathcal{O}^{\mathrm{Sign}_{\mathcal{R}}}$.

Definition 7 Perfect adaption is achieved for an SPS-EQ scheme on $(\mathbb{G}_i^*)^\ell$ if $(\rho E, \mathrm{Sign}_{\mathcal{R}}(\rho E, sk_{\mathcal{R}}))$ and $\mathrm{ChgRep}_{\mathcal{R}}(E, \sigma, \mu, pk_{\mathcal{R}})$ are identically distribute for any tuple $(sk_{\mathcal{R}}, pk_{\mathcal{R}}, E, \sigma, \mu)$ as long as

$$E \in (\mathbb{G}_i^*)^\ell \wedge \mu \in \mathbb{Z}_p^* \wedge Vkey_{\mathcal{R}}(sk_{\mathcal{R}}, pk_{\mathcal{R}}) = 1 \wedge \mathrm{Vrf}_{\mathcal{R}}(E, \sigma, pk_{\mathcal{R}}) = 1.$$

2.3 CS-TBK

The CS-TBK algorithm detailed in [34] is used for finds subtrees containing all non-revoked nodes. Let T denotes the maximum size of expiry time τ, and thus the number of leaf nodes in the binary tree BT is T. Both current time t and expiry time τ are mapped to the corresponding leaf nodes.

If τ is allocated to a leaf node η, the issuer produces a signature for each node of $Path(\eta)$ via the CS-TBK algorithm and then publishes these signatures to signers with τ. Although a bunch of signers with the same expiry times share a common leaf node η, the signatures of those signers are dissimilar for randomness. All leaves located left side of the leaf node related to a certain time t are revoked for their expiry times ahead of t. Expiration information $info_t$ at time t is essentially signatures of non-revoked nodes generated according to the CS-TBK algorithm.

For example, let $T = 8$ and τ corresponds to node 11, signers with τ obtain signatures of nodes 1, 2, 5, and 11. As shown in Fig. 2a, nodes 8, 9 are revoked when $t < \tau$, namely current time t is before expiry time τ, whereas nodes 3 and 5 are chosen as root nodes of subtrees of all non-revoked nodes, that is node 10–15. Thus, $info_t$ contains signatures of non-revoked subtrees of root node 3 and node5. Later, the signers can prove in a zero-knowledge manner that they own a signature of node 5, which is also contained in corresponding $info_t$. As shown in Fig. 2b when $t > \tau$, nodes 8,9,10, and11 are revoked. Thus, $info_t$ contains signatures of subtrees of root node 3 but no 5.

3 Scheme and security model

3.1 Scheme

In this part, we provide the established model for the revocable group signatures with time-bound keys and unforgeability of expiry time (RGS-TBK-UET). There are several entities involved in our scheme: a trusted party responsible for initial key generation and distribution, two authorities called the issuer and the opener, a bunch of users trying to join the group. The proposed proposal consists of the following algorithms.

GS.$GKGen(1^{\kappa}) \rightarrow (gpk, ik, ok)$: The algorithm takes in the security parameter 1^{κ}, and finally outputs the group public key gpk, the issuing key ik, and the opening key ok. The algorithm also initializes the user registration table **reg**.

GS.$UKGen(1^{\kappa}) \rightarrow (usk_i, upk_i)$: The algorithm takes in the public parameters and outputs secret/public key pair (usk_i, upk_i) for user i.

$\langle$GS.Join(gpk, usk_i, upk_i), GS.Issue$(gpk, ik, i, upk_i, \mathbf{reg}, \tau_i)\rangle$: In order to add user to the group, the issuer and a user executes the interactive protocol, which is usually assumed to communicate over secure channels. The joining algorithm is implemented by a user with (usk_i, upk_i), whereas the issuing algorithm is run by the issuer with inputs gpk, ik, upk_i,**reg** and τ_i. On success, the joining algorithm outputs (gsk_i, τ_i) and the issuing algorithm outputs registration table **reg** which user i is added an entry.

GS.Revoke$(i, ik, t, \mathcal{R}_t, \mathbf{reg}) \rightarrow (RL_t, info_t)$: The issuer runs the algorithm to generate expiration information and revocation list RL_t for revoked signers at time t. Upon input i, ik, t, $\mathcal{R}_t$ and **reg**, the algorithm outputs $(RL_t, info_t)$. The algorithm computes revocation token $grt_{i,t}$ for each i provided $t < \tau_i$ and stores $grt_{i,t}$ to RL_t. Besides, expiration information $info_t$ is computed.

GS.Sign$(gpk, gsk_i, m, t, info_t) \rightarrow \sigma$: The algorithm takes in the group public key gpk, group signing key gsk_i, a message m, current time t, and group information $info_t$, and outputs a group signature σ.

GS.Verify$(gpk, m, \sigma, t, RL_t) \rightarrow 0/1$: The deterministic algorithm is able to be run by anyone holding group public key gpk to check given σ is a valid group signature on m.

GS.Open$(gpk, ok, \mathbf{reg}, m, \sigma) \rightarrow (i, \pi)$: Provided that the group public key gpk, the opening key ok, a registration table **reg**, a message m, and a signature σ are input, the opener may extract the identity of the signer and the proof of signature, and finally return a pair (i, π), where integer i is nonnegative. The algorithm output a

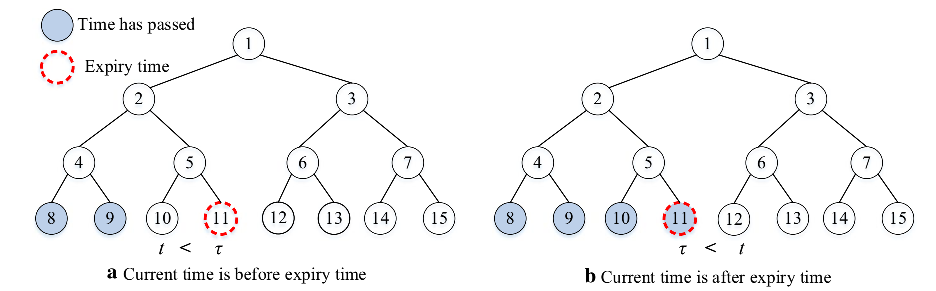

Fig. 2 The example of the CS-TBK

pair $(0, \pi)$ to indicate that the opener fails to attribute the signature to a certain group member. If $i > 0$, the opener could allege that the group member with identity i who produced the signature because the group member produced a proof π as corresponding evidence to demonstrate the above-mentioned fact.

GS.Judge$(gpk, m, \sigma, i, upk_i, \pi) \rightarrow 0/1$: Anyone in possession gpk can deterministically judge the validity of π given the group public key gpk, a message m, a signature σ, a user i, its public key upk_i, and a proof π. If π is a valid proof demonstrating that group member with identity i produced signature σ, the deterministic algorithm outputs 1 and outputs 0 otherwise.

3.2 Security model

Generally, the attack capabilities of adversaries are formalized via accessing certain subsets of oracles, which are detailed in Fig. 3. The security experiments corresponding to different requirements of the group signature as showed in Fig. 4. The following lists used in oracles are assumed to be global and maintained by the environment.

$\mathcal{H}$: Honest users. $\mathcal{C}$: Corrupted users. $\mathcal{B}$: Bad users. $\mathcal{R}_t$: Revoked users at time t.
SL: List of message-signature tuples. CL: List of challenge signatures obtained.
RL_t: List of revocation information.

Notably, $\mathcal{A}$ is capable of choosing personal secret keys of corrupted users, yet obtaining both personal and group signing keys of bad users.

A**ddU**: $\mathcal{A}$ adds an honest user with an identity $i \in \mathbb{N}$ and expiry time τ_i to the group.
RReg: $\mathcal{A}$ can read the information from the registration table.
WReg: $\mathcal{A}$ is allowed to modify the specified value of the registration table.
USK: $\mathcal{A}$ inputs an identity $i \in \mathbb{N}$ and then returns the personal private key usk_i and the private signing key gsk_i to the user.
Sign: $\mathcal{A}$ obtains a signature on behalf of an honest user by the signing oracle.
Chal$_\mathbf{b}$: $\mathcal{A}$ chooses two non-revoked honest members (i_0, i_1) as challenging users, and obtains challenge signature by calling the challenge oracle in the anonymity experi-

ment. Additionally, the adversary adds (m, σ, t) to the challenge signature set CL. The adversary is restricted to call the challenge oracle only once.

Open: $\mathcal{A}$ receives the identity and proof on a signature by running the opening algorithm as long as signature σ is not part of the challenge set CL.

Revoke: $\mathcal{A}$ removes users by calling revocation oracle.

CrptU: $\mathcal{A}$ parse the certain value pk as personal public key upk_i of newly corrupted user i before calling the **SndToU** oracle.

SndToU: $\mathcal{A}$ interacts with an honest user on behalf of the corrupted issuer.

SndToI: $\mathcal{A}$ communicate with an honest issuer in the user role.

3.3 Security notions

The definitions of correctness and main security attribute of the scheme are focused in this subsection. Let $\mathrm{Adv}_{\mathcal{GS}\cdot\mathcal{A}}(k) = \Pr[\mathrm{Exp}_{\mathcal{GS}\cdot\mathcal{A}}(k) = 1] \leq \epsilon(k)$ formally denote that the advantage of adversary $\mathcal{A}$ to win the respective experiment during polynomial-time in Fig. 4 is negligible.

Definition 8 The scheme achieves correctness if $\mathrm{Adv}_{\mathcal{GS}\cdot\mathcal{A}}^{\mathrm{correctness}}(k) = \Pr[\mathrm{Exp}_{\mathcal{GS}\cdot\mathcal{A}}^{\mathrm{correctness}}(k) = 1] \leq \epsilon(k)$.

The correctness states that any honest non-revoked group member should be able to issue valid signatures on any message. Once the message and signature are given, the opening algorithm ought to correctly recover the identity of the original signer. The opening algorithm produces the publicly verifiable proof that should be accepted by the judging algorithm.

Definition 9 The scheme achieves BU-Anonymity if $\mathrm{Adv}_{\mathcal{GS}\cdot\mathcal{A}}^{\mathrm{Anonymity}}(k) = \left|\Pr[\mathrm{Exp}_{\mathcal{GS}\cdot\mathcal{A}}^{0\text{ - Anonymity}}(k) = 1] - \Pr[\mathrm{Exp}_{\mathcal{GS}\cdot\mathcal{A}}^{1\text{ - Anonymity}}(k) = 1]\right| \leq \epsilon(k)$

The anonymity with backward unlinkability (BU-anonymity) means that $\mathcal{A}$ is infeasible to distinguish the identities of signers from signatures even if signatures are created by revoked signers. Specifically, if the real value of the bit b in the $\mathbf{Chal_b}$ oracle is guessed perfectly, $\mathcal{A}$ will break the anonymity. Moreover, $\mathcal{A}$ is allowed to access **Open** oracle and obtain signing keys excluding that of the challenge users.

Definition 10 The scheme achieves non-frameability if $\mathrm{Adv}_{\mathcal{GS}\cdot\mathcal{A}}^{\mathrm{Non\text{ - }Frame}}(k) = \Pr[\mathrm{Exp}_{\mathcal{GS}\cdot\mathcal{A}}^{\mathrm{Non\text{ - }Frame}}(k) = 1] \leq \epsilon(k)$.

Non-frameability guarantees that $\mathcal{A}$ is incapable of enforcing the honest opener to ascribe a certain valid signature to a specific user via creating a judge-accepted proof if this honest user indeed did not create this signature.

Definition 11 The scheme achieves traceability if $\mathrm{Adv}_{\mathcal{GS}\cdot\mathcal{A}}^{\mathrm{Traceability}}(k) = \Pr[\mathrm{Exp}_{\mathcal{GS}\cdot\mathcal{A}}^{\mathrm{Traceability}}(k) = 1] \leq \epsilon(k)$.

AddU(i)
Return $\perp$ If $i \in \mathcal{H} \cup \mathcal{C}$
$(usk_i, upk_i) \leftarrow \mathrm{GS.UKGen}(1^\kappa)$
$\mathcal{H} := \mathcal{H} \cup \{i\}$, $gsk_i := \perp$, $\mathrm{dec}^i_{Iss} := cont$
$\mathrm{st}^i_{Jn} := (\tau_i, gpk, i, usk_i)$
$\mathrm{st}^i_{Iss} := (\tau_i, gpk, ik, i, upk_i)$
$(\mathrm{st}^i_{Jn}, M_{Iss}, \mathrm{dec}^i_{Jn}) \leftarrow \mathrm{GS.Join}(\mathrm{st}^i_{Jn}, \perp)$
While $(\mathrm{dec}^i_{Iss} = cont$ and $\mathrm{dec}^i_{Jn} = cont)$ Do
  $(\mathrm{st}^i_{Iss}, M_{Jn}, \mathrm{dec}^i_{Iss}) \leftarrow \mathrm{GS.Issue}(\mathrm{st}^i_{Iss}, M_{Iss})$
  $(\mathrm{st}^i_{Jn}, M_{Iss}, \mathrm{dec}^i_{Jn}) \leftarrow \mathrm{GS.Join}(\mathrm{st}^i_{Jn}, M_{Jn})$
If $\mathrm{dec}^i_{Iss} = accept$ Then $\mathbf{reg}[i] := \mathrm{st}^i_{Iss}$
If $\mathrm{dec}^i_{Jn} = accept$ Then $gsk_i := \mathrm{st}^i_{Jn}$
Return (upk_i, τ_i)

CrptU(i, pk)
Return $\perp$ If $i \in \mathcal{H} \cup \mathcal{C}$
$\mathcal{C} := \mathcal{C} \cup \{i\}$
$upk_i := pk$, $\mathrm{dec}^i_{Iss} := cont$
Return *accept*

USK(i)
Return $\perp$ If $i \notin \mathcal{H} \setminus (\mathcal{C} \cup \mathcal{B})$
$\mathcal{B} := \mathcal{B} \cup \{i\}$
Return (usk_i, gsk_i)

SndToU(i, M_{in})
Return $\perp$ If $i \in \mathcal{C} \cup \mathcal{B}$
If $i \notin \mathcal{H}$ Then
  $\mathcal{H} := \mathcal{H} \cup \{i\}$
  $(usk_i, upk_i) \leftarrow \mathrm{GS.UKGen}(1^\kappa)$
  $gsk_i := \perp$, $M_{in} := \perp$
Return $\perp$ If $\mathrm{dec}^i_{Jn} \neq cont$
If st^i_{Jn} is undefined
  $\mathrm{st}^i_{Jn} := (t, gpk, i, usk_i)$
  $(\mathrm{st}^i_{Jn}, M_{out}, \mathrm{dec}^i_{Jn}) \leftarrow \mathrm{GS.Join}(\mathrm{st}^i_{Jn}, M_{in})$
If $\mathrm{dec}^i_{Jn} = accept$ Then $gsk_i := \mathrm{st}^i_{Jn}$
Return $(M_{out}, \mathrm{dec}^i_{Jn})$

SndToI(i, M_{in})
Return $\perp$ If $i \notin \mathcal{C}$
Return $\perp$ If $\mathrm{dec}^i_{Jn} \neq cont$
$\mathrm{st}^i_{Iss} := (t, ik, i, upk_i)$
$(\mathrm{st}^i_{Iss}, M_{out}, \mathrm{dec}^i_{Iss}) \leftarrow \mathrm{GS.Issue}(\mathrm{st}^i_{Iss}, M_{Iss})$
If $\mathrm{dec}^i_{Iss} = accept$ Then $\mathbf{reg}_i := \mathrm{st}^i_{Iss}$
Return $(M_{out}, \mathrm{dec}^i_{Iss})$

WReg(i, val)
$\mathbf{reg}[i] := val$

RReg(i)
Return $\mathbf{reg}[i]$

Sign(i, m, t)
Return $\perp$ If $i \notin \mathcal{H}$ or $i \in \mathcal{R}_t$ or $gsk_i = \perp$ or $info_t = \perp$
$\sigma \leftarrow \mathrm{GS.Sign}(gpk, gsk_i, m, t, info_t)$
$\mathrm{SL} := \mathrm{SL} \cup \{(i, m, \sigma, t)\}$
Return σ

Chal$_b$(i_0, i_1, m, t)
Return $\perp$ If $\exists b \in \{0,1\}$ s.t. $\notin \mathcal{H}$ or
  $\mathbf{gsk}_{i_b} = \perp$ or $i_b \in \mathcal{R}_t$ or $t \geq \tau_{i_b}$
$\sigma \leftarrow \mathrm{GS.Sign}(gpk, gsk_i, m, t, info_t)$
$\mathrm{CL} := \mathrm{CL} \cup \{(m, \sigma, t)\}$
Return σ

Open ($m, \sigma, info_t$)
Return $(\perp, \perp)$ If $\mathrm{GS.Verify}(gpk, m, \sigma, info_t, \mathrm{RL}_t) = 0$
Return $(\perp, \perp)$ If $(m, \sigma, t) \in \mathrm{CL}$
Return $\mathrm{GS.Open}(gpk, ok, \mathbf{reg}, m, \sigma, info_t)$

Revoke($\mathcal{R}_t$)
$\mathrm{RL}_t \leftarrow \mathrm{GS.Revoke}(i, ik, t, \mathcal{R}_t, \mathbf{reg})$
Return $(info_t, \mathrm{RL}_t)$

Fig. 3 Details of the oracles

Traceability essentially defines that $\mathcal{A}$ is infeasible to counterfeit a signature result. In other words, the honest opener is either incapable of identifying the signer of the forgery signature or generating a judge-accepted proof of its claim even if the signer of a signature has been identified.

Definition 12 The scheme achieves weak opening soundness if $\mathrm{Adv}^{\mathrm{WeakOS}}_{\mathcal{GS}\cdot\mathcal{A}}(k) = \Pr[\mathrm{Exp}^{\mathrm{WeakOS}}_{\mathcal{GS}\cdot\mathcal{A}}(k) = 1] \leq \epsilon(k)$.

Weak opening soundness actually means that a malicious user is infeasible to allege ownership of signatures generated originally by honest users via a counterfeited opening proof as long as the opener behaves honestly [15].

4 Construction and security analysis

4.1 Detailed construction

As previously mentioned, the scheme [17] essentially only allow members to enroll at all times but cannot leave freely, which inspired us to added revocation functionality to

Experiment:$\mathrm{Exp}_{GS\text{-}\mathcal{A}}^{\mathrm{correctness}}(k)$

$(gpk, ik, ok) \leftarrow \mathrm{GS.GKGen}(1^{\kappa}); \mathcal{H} := \varnothing; \ \mathcal{C} := \varnothing$

$(i, m, t) \leftarrow \mathcal{A}^{\mathrm{AddU,RReg,Revoke}}(gpk)$

If $i \notin \mathcal{H}$ or $gsk_i = \perp$ or $i \in \mathcal{R}_t$ or $t \geq \tau_i$

or $info_t = 0$ then return 0

$\sigma \leftarrow \mathrm{GS.Sign}(gpk, gsk_i, m, t, info_t)$

If $\mathrm{GS.Verify}(gpk, m, \sigma, t, \mathrm{RL}_t) = 0$ then return 1

$(j, \pi) \leftarrow \mathrm{GS.Open}(gpk, ok, \mathbf{reg}, m, \sigma)$

If $i \neq j$ then return 1

If $\mathrm{GS.GS.Judge}(gpk, m, \sigma, i, upk_i, \pi) = 0$

then return 1 else return 0

Experiment:$\mathrm{Exp}_{GS\text{-}\mathcal{A}}^{\mathrm{Traceability}}(k)$

$(gpk, ik, ok) \leftarrow \mathrm{GS.GKGen}(1^{\kappa}); \mathcal{H}, \mathcal{C}, \mathcal{B}, \mathcal{R}_t := \varnothing$

$(m, \sigma, t) \leftarrow \mathcal{A}^{\mathrm{AddU,CrptU,USK,SndToI,Sign,RReg,Revoke}}(gpk, ok)$

If $\mathrm{GS.Verify}(gpk, m, \sigma, t, \mathrm{RL}_t) = 0$

or $(i, m, \sigma, t) \notin \mathrm{SL}$ then return 0

$(i, \pi) \leftarrow \mathrm{GS.Open}(gpk, ok, \mathbf{reg}, m, \sigma)$

If $i = 0$ or $(i \in \mathcal{C} \backslash \mathcal{R}_t \wedge t \geq \tau_i)$

or $\mathrm{GS.Judge}(gpk, m, \sigma, i, upk_i, \pi) = 0$

then return 1 else return 0

Experiment:$\mathrm{Exp}_{GS\text{-}\mathcal{A}}^{\mathrm{Anonymity}}(k) // b \in \{0, 1\}$

$(gpk, ik, ok) \leftarrow \mathrm{GS.GKGen}(1^{\kappa}); \mathcal{H}, \mathcal{C}, \mathcal{R}_t, \mathrm{SL}, \mathrm{CL} := \varnothing$

$d \leftarrow \mathcal{A}^{\mathrm{Chal}_b, \mathrm{WReg,CrptU,Open,AddU,Sign,Revoke}}(gpk)$

Return d

Experiment:$\mathrm{Exp}_{GS\text{-}\mathcal{A}}^{\mathrm{Non\text{-}Frame}}(k)$

$(gpk, ik, ok) \leftarrow \mathrm{GS.GKGen}(1^{\kappa}); \mathcal{H}, \mathcal{C}, \mathcal{B}, \mathcal{R}_t, \mathrm{SL} := \varnothing$

$(i, m, \sigma, t, \pi) \leftarrow \mathcal{A}^{\mathrm{SndToU,WReg,Sign,USK,Crpt,Revoke}}(gpk, ik, ok)$

If $\mathrm{GS.Verify}(gpk, m, \sigma, t, \mathrm{RL}_t) = 0$ then return 0

If $i \in \mathcal{H} \backslash (\mathcal{C} \cup \mathcal{B})$ and $gsk_i \neq \perp$ and $(i, m, \sigma, t) \notin \mathrm{SL}$

and $\mathrm{GS.Judge}(gpk, m, \sigma, i, upk_i, \pi) = 1$

then return 1 else return 0

Experiment:$\mathrm{Exp}_{GS\text{-}\mathcal{A}}^{\mathrm{WeakOpengingSoundness}}(k)$

$(gpk, ik, ok) \leftarrow \mathrm{GS.GKGen}(1^{\kappa}); \mathcal{H}, \mathcal{C}, \mathcal{B} := \varnothing$

$(m, i, j, st) \leftarrow \mathcal{A}^{\mathrm{AddU}}(gpk)$

$\sigma \leftarrow \mathrm{GS.Sign}(gpk, gsk_i, m, t, info_t)$

$\pi^{*} \leftarrow \mathcal{A}^{\mathrm{AddU}}(st, \sigma, gsk_j)$

If $i \neq j$ and $i, j \in i \in \mathcal{H} \backslash (\mathcal{C} \cup \mathcal{B})$ *and* $gsk_i, gsk_j \neq \perp$

and $\mathrm{GS.Judge}(gpk, m, \sigma, j, upk_j, \pi^{*}) = 1$

then return 1else return 1

Fig. 4 Security experiments

the construction of [17] using the methodology of [34]. Our construction is detailed in Fig. 5, and system parameters are illustrated in Table 1.

The natural revocation is achieved by CS-TBK algorithm and is details as follows. Assumed that a leaf node η is selected to an expiry time τ, the issuer produces SPS-EQ signature $\sigma'_{\mathrm{A}_j} \leftarrow \mathrm{Sign}_{\mathcal{R}}(\xi_j(U_i, Q), sk_{\mathcal{R}})$ for each node $\xi_j \in \mathrm{path}(\eta)$ and sends $(\{\sigma'_{\mathrm{A}_j}\}_{j \in [nt]}, \tau_i)$ to the signer i. Then, the signer computers her owning secret signing key by re-randomization property of SPS-EQ $gsk_i := \{(\xi_j(rP, P), \sigma_{Aj})\}_{j \in [nt]} \leftarrow \{\mathrm{ChgRep}_{\mathcal{R}}(\xi_j(U_i, Q), \sigma'_{A_j}, q^{-1}, pk_{\mathcal{R}})\}_{j \in [nt]}$.For the current time t, the issuer firstly outputs $Y =: (\vartheta_1, \vartheta_2, \ldots, \vartheta_{\mathrm{num}})$ running the CS - TBK(BT,t) algorithm and chooses a secret vector $(T_i, Q) \leftarrow (\mu P, P) \in (\mathbb{G}_1^*)^2$ (where $\mu \xleftarrow{\$} \mathbb{Z}_p^*$). Next, the issuer computes SPS-EQ signature $\sigma_{Bk} \leftarrow \mathrm{Sign}_{\mathcal{R}}(\vartheta_k(T_i, Q), sk_{\mathcal{R}})$, which is contained in expiration information $info_t$. Apparently, the gsk_i is the SPS-EQ signature of the message $\xi_j \in \mathrm{path}(\eta) := (\xi_1, \xi_2, \ldots, \xi_{nt})$ and the equivalence class of signer secret identity, whereas $info_t$ is the SPS-EQ signature of another message $\vartheta_i \in Y =: (\vartheta_1, \vartheta_2, \ldots, \vartheta_{num})$ and the equivalence class corresponding to the current time. In the light of the CS method, such a common node both $\xi \in Path(\eta) \cap Y$ exists if $\tau < t$. That is the non-revoked signers can prove in a zero-knowledge manner, that the node ξ possesses two signatures in their own the private signing key gsk and expiry information $info_t$ respectively. Note that unless unforgeability of the SPS-EQ signature scheme is broken, that is the signer creates SPS-EQ signature σ_{Bk}, it is infeasible to generate a valid group signature for an expired signer. Consequently, the unforgeability of expiry time for signing keys is guaranteed.

The way of premature revocation is described as below: At the stage of GS.Issue, the issuer stores a revocation token $grt_i := U_i$. At the time t, the issuer picks randomly $y_t \xleftarrow{\$} \mathbb{Z}_p^*$ and lets $h_t{=}y_t P, \hat{h}_t{=}y_t \hat{P}$, then $e(h_t, \hat{P}) = e(P, \hat{h}_t)$ hold. A group signature is composed of $\beta y_t P$,$\alpha(rq{+}\beta)P$ and $\alpha\hat{P}$, where α, β are picked randomly by the signer. If $i \in \mathcal{R}_t$, namely a signer i is revoked in the premature manner then the issuer computes $grt_{i,t} := y_t grt_i$ and stores $grt_{i,t}$ to RL_t. By checking whether the equation $e(grt_{i,t} + \beta y_t P, \alpha\hat{P}) = e(\alpha(rq + \beta)P, y_t\hat{P})$ holds for each $grt_{i,t}$ successively, the verifier is able to verify whether i is a premature revoked signer.

4.2 Security analysis

Theorem 1 *The proposal achieves correctness if SPS-EQ and SoK achieve correctness.*

Proof (*Sketch*) Correctness is straightly originated from the correctness of the proposal [17, 34].

Theorem 2 *The proposal achieves BU-anonymity if Π achieves adaptive zero-knowledge, SoK achieves simulatability (simulatability and straight-line f-extractability), Ω achieves IND-CPA (IND-CCA2) security and the DDH assumption holds.*

Proof (*Sketch*) Usually, BU-anonymity indicates that signers of two group signatures cannot be distinguished without an opening secret key. Thus, the attack on anonymity is essentially equivalent to that on encryption, which producing membership certificates and proofs. Finally, the anonymity is reduced to the security of the PKE scheme and NIZK Proof. Naturally, a signer maintains anonymity because two randomized user secret keys are difficult to distinguish under DDH assumption in $\mathbb{G}_1$. Therefore, the output distributions of the $\mathbf{Chal_b}$ oracle and the input bit b are mutually independent.

Proof Let $\mathrm{N}_{ch}, \mathrm{N}_o, \mathrm{N}_{\mathrm{AddU}} \leq \mathrm{poly}(\kappa)$ denote the number of queries to $\mathbf{Chal_b}$, **Open**, and A**ddU** respectively.

G$_0$: original anonymity experiment.
G$_1$: As **G**$_0$, except that executing $(crs_J, td_J) \leftarrow \Pi.\mathcal{S}_1(1^\kappa)$ rather than $crs_J \leftarrow \Pi.Setup(1^\kappa)$ at the stage of GS.*GKGen* algorithm, and the information td_J is stored. Next, each call to Π.Proof that executed at the stage of GS.Join algorithm is simulated via the simulator $\Pi.\mathcal{S}_1$. According to adaptive zero-knowledge of Π, the probability of winning the game that $\mathcal{A}$ successfully distinguishes **G**$_0$ and **G**$_1$ is negligible, i.e. $|\Pr[\mathcal{G}_1] - \Pr[\mathcal{G}_0]| \leq \epsilon_{zk_J}(k)$.
G$_2$: As **G**$_1$, except that executing $(crs_O, td_O) \leftarrow \Pi.\mathcal{S}_1(1^\kappa)$ rather than $crs_O \leftarrow \Pi.Setup(1^\kappa)$ at the stage of GS.*GKGen* algorithm, and the information td_O is stored. Next, all zero-knowledge proofs Π.Proof at the stage of GS.Open algorithm is simulated via the simulator $\Pi.\mathcal{S}_1$. According to adaptive zero-knowledge of Π, the probability of winning the game that $\mathcal{A}$ successfully distinguishes **G**$_1$ and **G**$_2$ is negligible, i.e. $|\Pr[\mathcal{G}_2] - \Pr[\mathcal{G}_1]| \leq \epsilon_{zk_o}(k)$.
G$_3$: As **G**$_2$, except that executing $(crs_s, td_s) \leftarrow$ Sok.$Setup(1^\kappa)$ rather than $crs_s \leftarrow$ Sok.$Setup(1^\kappa)$ at the stage of GS.*GKGen* algorithm, and the information td_s is stored. Next, each call to Sok.Sign is simulated via the simulator (without a witness). According to simulatability of Sok, the probability of winning the game that

$\mathcal{A}$ successfully distinguishes $\mathbf{G_3}$ and $\mathbf{G_2}$ is negligible, i.e., $|\Pr[\mathcal{G}_3] - \Pr[\mathcal{G}_2]| \leq \epsilon_{\mathrm{SIM}}(k)$.

$\mathbf{G_4}$: As $\mathbf{G_3}$, except that pk_o is obtained from an IND-CPA or IND-CCA2 challenger rather than $(sk_o, pk_o) \leftarrow \Omega.\mathrm{KGen}(1^\kappa)$ at the stage of GS.*GKGen* algorithm, and $sk_o = \perp$ is set. In the CCA2 case, it next uses the **Open** oracle to decrypt the ciphertext $\hat{C}_{J_i}$ stored in **reg** for all users and obtain simulates the proof via the straight-line f-extractor. In case of CCA2, a witness ρ can be extracted in each call to the **Open** oracle with overwhelm $1 - \epsilon_{\mathrm{EXT}}(k)$ extraction probability according to the straight-line f-extractability of the SoK. Therefore, both games proceed same unless there is a extraction fail, i.e., $|\Pr[\mathcal{G}_4] - \Pr[\mathcal{G}_3]| \leq N_o \cdot \epsilon_{\mathrm{EXT}}(k)$. In case of CPA, the **Open** oracle do not have to be simulated and the opening key is only obtained from the IND-CPA challenger. Therefore, $\mathbf{G_4}$ is conceptually identical to $\mathbf{G_3}$, i.e., $\Pr[\mathcal{G}_4] = \Pr[\mathcal{G}_3]$.

$\mathbf{G_5}$: As $\mathbf{G_4}$, except that the ciphertext $\hat{C}_{J_i}$ is computed in the GS.Join algorithm (actually executed via the A**ddU** oracle) as $\hat{C}_{J_i} = \Omega.\mathrm{Enc}(pk_o, \hat{P})$ rather than $\hat{C}_{J_i} = \Omega.\mathrm{Enc}(pk_o, r\hat{P}, \omega)$, namely the random parameters associated with identity is removed in message. According to the IND-CCA2 security of Ω, the probability of winning the game for $\mathcal{A}$, i.e., $|\Pr[\mathcal{G}_5] - \Pr[\mathcal{G}_4]| \leq N_{\mathrm{AddU}} \cdot \epsilon_{CCA2}(k)$.

$\mathbf{G_6}$: As $\mathbf{G_5}$, except that sk_o is re-added, namely, $(sk_o, pk_o) \leftarrow \Omega.\mathrm{KGen}(1^\kappa)$ is obtained again. We decrypt ourselves with in the WReg simulation rather than via the decryption oracle in the CCA2 case. Therefore, $\mathbf{G_6}$ is conceptually identical to $\mathbf{G_5}$, i.e., $\Pr[\mathcal{G}_6] = \Pr[\mathcal{G}_5]$.

$\mathbf{G_7}$: As $\mathbf{G_6}$, except that i^* is revoked by computing $grt_{i^*,t} := y_t r_{i^*} qP$ while $t \neq t^*$. Remark that $\mathcal{A}$ is not need to compute grt_{i^*,t^*} because at the challenge time t^*, i^* is unrevoked, which induced backward unlinkability. Therefore, $\mathbf{G_7}$ is conceptually identical to $\mathbf{G_6}$. i.e., $\Pr[\mathcal{G}_7] = \Pr[\mathcal{G}_6]$.

$\mathbf{G_8}$: As $\mathbf{G_7}$, except that the $\mathbf{Chal_b}$ oracle is modified as follows. Instead of $\mathrm{ChgRep}_{\mathcal{R}}(M, \rho, pk_{\mathcal{R}})$, $(\Phi, \Psi) \xleftarrow{\$} \mathbb{G}_1$ is chosen, and $\mathrm{ChgRep}_{\mathcal{R}}((\Phi, \Psi), \rho, pk_{\mathcal{R}})$ is computed to answers to the $\mathbf{Chal_b}$ query. According to DDH assumption, the winning probability of $\mathcal{A}$ is negligible, i.e.$|\Pr[\mathcal{G}_8] - \Pr[\mathcal{G}_7]| \leq N_{\mathrm{Chal}_b} \cdot \epsilon_{\mathrm{DDH}}(k)$. In $\mathbf{G_8}$, the advantage of $\mathcal{A}$ can then only be 0 and the simulation is irrelevant the bit b, i.e., $\Pr[\mathcal{G}_8] = 1/2$.

$\mathbf{G_9}$: As $\mathbf{G_9}$, except that $\psi_3^* \xleftarrow{\$} \mathbb{G}_2$ is randomly choose.

$\mathbf{G_{10}}$: As $\mathbf{G_9}$, except that randomly choose $\psi_2^* \xleftarrow{\$} \mathbb{G}_1$ is randomly choose.

The bound of success probability in $\mathbf{G_0}$ is $\Pr[\mathcal{G}_0] \leq 1/2 + N_{\mathrm{AddU}} \cdot \epsilon_{CCA2}(k) + N_{Chalb} \cdot \epsilon_{\mathrm{DDH}}(k) + \epsilon_{\mathrm{zk}_J}(k) + \epsilon_{\mathrm{zk}_o}(k) + \epsilon_{\mathrm{SIM}}(k)$ (in the CPA case) or $\Pr[\mathcal{G}_0] \leq 1/2 + N_{\mathrm{AddU}} \cdot \epsilon_{CCA2}(k) + N_{\mathrm{Chal}} \cdot \epsilon_{\mathrm{DDH}}(k) + \epsilon_{\mathrm{zk}_J}(k) + \epsilon_{\mathrm{zk}_o}(k) + \epsilon_{\mathrm{SIM}}(k) + N_o \cdot \epsilon_{\mathrm{EXT}}(k)$ (in the CCA2 case), which proves Theorem 2.

Theorem 3 *The proposal achieves non-frameablity if Π achieves soundness and adaptive zero-knowledge, SoK achieves simulatability and extractability, $\sum$ achieves EUF-CMA security, Ω achieves perfect correctness, and the co-CDHI assumption holds.*

$\mathrm{GS.GKGen}(1^\kappa) \to (gpk, ik, ok)$

- $BG := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P}) \leftarrow \mathrm{BGGen}(1^\kappa)$
- $crs_J \leftarrow \Pi.\mathrm{Setup}(1^\kappa),\ crs_o \leftarrow \Pi.\mathrm{Setup}(1^\kappa),\ crs_s \leftarrow \mathrm{Sok.Setup}(1^\kappa)$
- $(sk_\mathcal{R}, pk_\mathcal{R}) \leftarrow \mathrm{KGen}_\mathcal{R}(BG, 2),\ (sk_o, pk_o) \leftarrow \Omega\mathrm{KGen}(1^\kappa),\ (sk_\varepsilon, pk_\varepsilon) \leftarrow \Sigma.\mathrm{KGen}(1^\kappa)$
- $gpk := (pk_\mathcal{R}, pk_o, crs_J, crs_o, crs_s),\ ik := (sk_\varepsilon, sk_\mathcal{R}),\ ok = sk_o,\ \mathbf{reg} := 0$

$\mathrm{GS.UKGen}(1^\kappa) \to (usk_i, upk_i)$

- $(usk_i, upk_i) \leftarrow \Sigma\mathrm{KGen}(1^\kappa)$

$\mathrm{GS.Join}^1(gpk, usk_i, upk_i) \to (M_J, st)$

- Choose $q, r \xleftarrow{\$} \mathbb{Z}_p^*$, set $(U_i, Q) \leftarrow (r \cdot qP, qP)$
- $\hat{C}_{J_i} \leftarrow \Omega.Enc(pk_o, r\hat{P}; \omega),\ \sigma_{J_i} \leftarrow \Sigma.Sign(usk_i, \hat{C}_{J_i}),\ \pi_{J_i} \leftarrow \Pi.\mathrm{Proof}(crs_J, (U_i, Q, \hat{C}_{J_i}, pk_o), (r, \omega))$
- Return $M_J \leftarrow (\ (U_i, Q), \hat{C}_{J_i}, \sigma_{J_i}, \pi_{J_i})$, and $st \leftarrow (gpk, q, U_i, Q)$

$\mathrm{GS.Issue}(gpk, ik, i, upk_i, \mathbf{reg}, \tau_i) \to (\mathbf{reg}, \sigma'_{A_j})$

- Receive $M_J \leftarrow (\ (U_i, Q), \hat{C}_{J_i}, \sigma_{J_i}, \pi_{J_i})$
- If $(\hat{C}_{J_i}, \sigma_{J_i}, \cdot, \cdot) \notin \mathbf{reg}$ and $\Pi.\mathrm{Vrf}(crs_J, (U_i, Q, \hat{C}_{J_i}, pk_o), \pi_{J_i}) = 1 \wedge \Sigma.\mathrm{Vrf}(upk_i, \hat{C}_{J_i}, \sigma_{J_i}) = 1$
- Assign a leaf node η to τ_i, for each $\xi_j \in \mathrm{path}(\eta) := (\xi_1, \xi_2, \cdots, \xi_{nt})$ ($nt = \log(\mathrm{T})$)
  compute $\sigma'_{A_j} \leftarrow \mathrm{Sign}_\mathcal{R}(\xi_j(U_i, Q), sk_\mathcal{R}), \mathbf{reg}_i := (\hat{C}_{J_i}, \sigma_{J_i}, \tau_i, \{\sigma'_{A_j}\}_{j\in[nt]}), grt_i := U_i$
- Return $\mathbf{reg}$ and send $(\{\sigma'_{A_j}\}_{j\in[nt]}, \tau_i)$ to the user i

$\mathrm{GS.Revoke}(i, ik, t, \mathcal{R}_t, \mathbf{reg}) \to (RL_t, info_t)$

- For the current time t, obtain $(\vartheta_1, \vartheta_2, \cdots, \vartheta_{num}) \leftarrow \mathrm{CS\text{-}TBK}(\mathrm{BT}, t)$
- Choose $\mu \xleftarrow{\$} \mathbb{Z}_p^*$, set $(T_i, Q) \leftarrow (\mu P, P)$, compute $\sigma_{Bk} \leftarrow \mathrm{Sign}_\mathcal{R}(\vartheta_k(T_i, Q), sk_\mathcal{R})$
- Choose $y_t \xleftarrow{\$} \mathbb{Z}_p^*$, compute $h_t = y_t P, \hat{h}_t = y_t \hat{P}$, set expiration information $info_t := (h_t, \{\sigma_{Bk}\}_{k\in[num]})$
- For all $i \in \mathcal{R}_t$, compute $grt_{i,t} := y_t grt_i$, set revocation list $RL_t = (h_t, \hat{h}_t, \{grt_{i,t}\}_{i\in\mathcal{R}_t})$
- Return $(RL_t, info_t)$

$\mathrm{GS.Sign}(gpk, gsk_i, m, t, info_t) \to \sigma$

- Choose $\rho \xleftarrow{\$} Z_p^*$, *compute signature*
  $\sigma_{1A} \leftarrow \mathrm{ChgRep}_\mathcal{R}(\{\sigma_{Aj}\}_{j\in[nt]}, \rho, pk_\mathcal{R}), \sigma_{1B} \leftarrow \mathrm{ChgRep}_\mathcal{R}(\{\sigma_{Bk}\}_{k\in[num]}, \rho, pk_\mathcal{R})$
  $\sigma_{2A} \leftarrow \mathrm{Sok.Sign}(crs_s, (P, \sigma_{1A}[1][2]), \rho, \sigma_{1A} \| m), \sigma_{2B} \leftarrow \mathrm{Sok.Sign}(crs_s, (P, \sigma_{1B}[1][2]), \rho, \sigma_{1B} \| t)$
- Choose $\alpha, \beta \xleftarrow{\$} Z_p^*$ set $\psi_1 = \beta y_t P$, $\psi_2 = \alpha(rq + \beta)P$, $\psi_3 = \alpha\hat{P}$, $\psi_4 = \alpha P$
- Return $\sigma \leftarrow (\psi_1, \psi_2, \psi_3, \psi_4, \sigma_{1A}, \sigma_{1B}, \sigma_{2A}, \sigma_{2B})$

$\mathrm{GS.Verify}(gpk, m, \sigma, t, RL_t) \to 0/1$

- If $\mathrm{Vrf}_\mathcal{R}(\sigma_{1A}, pk_\mathcal{R}) = 0 \vee \mathrm{Sok.Vrf}(crs_s, (P, \sigma_{1A}[1][2]), \sigma_{1A} \| m) = 0$
  $\vee\, \mathrm{Vrf}_\mathcal{R}(\sigma_{1B}, pk_\mathcal{R}) = 0 \vee \mathrm{Sok.Vrf}(crs_s, (P, \sigma_{1B}[1][2]), \sigma_{1B} \| t_s) = 0$ then return 0
- If $e(grt_{i,t} + \psi_1, \psi_3) = e(\psi_2, \hat{h}_t)$ return 1

$\mathrm{GS.Open}(gpk, ok, \mathbf{reg}, m, \sigma) \to (i, \pi)$

- If $\hat{R} \leftarrow \Omega.\mathrm{Dec}(sk_o, \hat{C}_{J_i}) \wedge e(\sigma_{1A}[1][1], \hat{P}) = e(\sigma_{1A}[1][2], \hat{R})$ then
- $\pi = (\pi_o, \hat{C}_{J_i}, \sigma_{J_i})$ and $\pi_o \leftarrow \Pi.\mathrm{Proof}(crs_o, (\hat{C}_{J_i}, pk_o, \sigma), (sk_o, \hat{R}))$
- Return (i, π)

$\mathrm{GS.Judge}(gpk, m, \sigma, i, upk_i, \pi) \to 0/1$

- Return 1 if the following holds and 0 otherwise
- If $\pi \leftarrow (\pi_o, \hat{C}_{J_i}, \sigma_{J_i}) \wedge \Sigma\mathrm{Vrf}(upk_i, \hat{C}_{J_i}, \sigma_{J_i}) = 1 \wedge \Pi.\mathrm{Vrf}(crs_o, (\hat{C}_{J_i}, pk_o, \sigma), \pi_o) = 1$

Fig. 5 The details of construction

Proof (*Sketch*) Equivalence class related to each group member is chosen as the secret vector of the membership certificate, and this secret information is only known by the signer. The encryption of $\hat{R} \in \mathbb{G}_2$ and digital signature are used an identity proof for providing means to open signatures. The signer issues a group signature, which consists of the randomized group signing key and the signature of knowledge. The unforgeability of $\sum$ and perfect correctness of Ω ensure that all valid signatures can be correctly opened. Moreover, the impossibility to unblind a user secret key under co-CDHI, ensures the

Table 1 Explanation of parameters

Parameters	Description	Parameters	Description
κ	The security parameter	σ	The signature
BG	A type-3 bilinear group	π	The proof
P	Generator of group $\mathbb{G}_1$	i	The user
$\hat{p}$	Generator of group $\mathbb{G}_2$	m	The message
gpk	Group public key	Π	The non-interactive zero-knowledge proofs system
ik	The issuing key	Ω	The encryption algorithm
ok	The opening key	Σ	The digital signature algorithm
reg	The user registration table	crs	Common reference string
$\mathcal{R}_t$	Revoked users at time t	$(sk_{\mathcal{R}}, pk_{\mathcal{R}})$	The key pair from the SPS-EQ algorithm
gsk	The group signing key	(sk_o, pk_o)	The key pair from the encryption algorithm
grt	The revocation token	$(sk_\epsilon, pk_\epsilon)$	The key pair from the signature algorithm
$info$	The group information	(usk,upk)	The user secret/public key pair
t	The current time	(U_i, Q)	The equivalence classes representative on users
τ	The expiry time	(T_i, Q)	The equivalence classes representative on time

impossibility to counterfeit a signature owned by an honest group member. Additionally, the *SoK* guarantees that unblinded user secret keys can be extracted even if $\mathcal{A}$ has succeeded.

Proof Let $n \leq \text{poly}(\kappa)$ denote the number of users.

$\mathbf{G_0}$: The original non-frameability experiment.

$\mathbf{G_1}$: As $\mathbf{G_0}$, except that we guess $\mathcal{A}$ will attack the user i^* and if $\mathcal{A}$ attacks another user, we abort. The winning probability in $\mathbf{G_1}$ is in common with that in $\mathbf{G_0}$ unless an abortion happens, i.e., $\Pr[\mathcal{G}_1] = \Pr[\mathcal{G}_0] \cdot 1/n$.

$\mathbf{G_2}$: As $\mathbf{G_1}$, except that executing $(crs_J, td_J) \leftarrow \Pi.\mathcal{S}_1(1^\kappa)$ rather than $crs_J \leftarrow \Pi.Setup(1^\kappa)$ at the stage of GS.*GKGen* algorithm and the trapdoor information td_J is stored. Next, each call to Π.Proof at the stage of GS.Join algorithm is simulated by the simulator $\Pi.\mathcal{S}_1$. According to adaptive zero-knowledge of Π, the probability of winning the game that $\mathcal{A}$ successfully distinguishes $\mathbf{G_1}$ and $\mathbf{G_2}$ is negligible, i.e., $|\Pr[\mathcal{G}_2] - \Pr[\mathcal{G}_1]| \leq \epsilon_{\text{zk}_J}(k)$

$\mathbf{G_3}$: As $\mathbf{G_2}$, except that crs_o is obtained from a soundness challenger at the stage of GS.*GKGen* algorithm. Therefore, $\mathbf{G_3}$ is conceptually identical to $\mathbf{G_2}$, i.e., $\Pr[\mathcal{G}_3] = \Pr[\mathcal{G}_2]$.

$\mathbf{G_4}$: As $\mathbf{G_3}$, except that we setup the SoK via $(crs_s, td_s) \leftarrow$ Sok.$Setup(1^\kappa)$ rather than $crs_s \leftarrow$ Sok.$Setup(1^\kappa)$ at the stage of GS.*GKGen* algorithm, and the information td_s is stored. Next, each call to Sok.Sign is simulated by the simulator. According to simulatability of Sok, the probability of winning the game that $\mathcal{A}$ successfully distinguishes $\mathbf{G_4}$ and $\mathbf{G_3}$ is negligible, i.e.,$|\Pr[\mathcal{G}_4] - \Pr[\mathcal{G}_3]| \leq \epsilon_{\text{SIM}}(k)$.

$\mathbf{G_5}$: As $\mathbf{G_4}$, except that we pick $q, r \xleftarrow{\$} \mathbb{Z}_p^*$ while queried for user i^* and let (U_{i^*}, Q_{i^*}) denote at the stage of GS.Join algorithm (actually executed via the **SndToU**) $(r \cdot qP, qP)$. Next, on each Join for any user $i \neq i^*$, it need to check that if the same class have been chosen for user i^* incidentally. The check process is performed via checking whether $U_{i^*} = r_i \cdot Q_{i^*}$ using r_i that is the value for r selected for the user i when Joining. The check above does not need to acquire r for user i^* or the discrete logarithms q.Both $\mathbf{G_4}$ and $\mathbf{G_5}$ proceed identically unless an abortion happens, the probability of which is $\epsilon_{guess}(k) = n/(p-1)$, i.e., $|\Pr[\mathcal{G}_5] - \Pr[\mathcal{G}_4]| \leq \epsilon_{guess}(k)$.

$\mathbf{G_6}$: As $\mathbf{G_5}$ except that a co-CDHI instance $(aP, 1/a\hat{P})$ in relation to BG is obtained and pick $\hbar \xleftarrow{\$} \mathbb{Z}_p^*$. Next, we adjust the GS.Join algorithm (actually executed via the SndToU) while queried for i^* as below. Let $(U_{i^*}, Q_{i^*}) = (\hbar P, aP)$, compute $\hat{C}_{J_{i^*}} \leftarrow \Omega.\text{Enc}(pk_o, \hbar{\cdot}1/a\hat{P})$ and store $\hbar$. Once execution is successful, let group signing key $gsk_i := \{(U_{i^*}, Q_{i^*}, \sigma_{Aj})\}_{j\in[nt]}$ and revocation token $grt_{i^*} = U_{i^*}$ Because $\hbar$ is uniformly random. Therefore, $\mathbf{G_6}$ is conceptually identical to $\mathbf{G_5}$, i.e.,$\Pr[\mathcal{G}_6] = \Pr[\mathcal{G}_5]$.

$\mathbf{G_7}$: As $\mathbf{G_6}$, except that for each forgery output by the $\mathcal{A}$, $\rho = \text{Sok.Extract}(crs_s, td_s, ((P, \sigma_{1A}[1][2]), \sigma_{1A}||m, \sigma_{2A})))[2]), \sigma_{1A}||m, \sigma_{2A})))$ is extracted and abort if the extraction fails. According to the extractability of the SoK, the unsuccessful probability of extracting a witness ρ is negligible. Therefore, both $\mathbf{G_7}$ and $\mathbf{G_6}$ proceed identically unless an extracting failure happens, i.e.,$|\Pr[\mathcal{G}_7] - \Pr[\mathcal{G}_6]| \leq \epsilon_{\text{EXT}}(k)$.

$\mathbf{G_8}$: As $\mathbf{G_7}$, except that we further adjust the GS.Join algorithm while queried for user i^* (actually executed via the **SndToU** oracle) as below. Rather than obtaining (usk_{i^*}, upk_{i^*}) from GS.UK$Gen(1^\kappa)$, we set $usk_{i^*} \leftarrow \emptyset$ and obtain upk_{i^*} by interacting with an EUF-CMA challenger. Moreover, we obtain all required signatures via the oracle offered by the EUF-CMA challenger. Therefore, $\mathbf{G_8}$ is conceptually identical to $\mathbf{G_7}$, so $\Pr[\mathcal{G}_8] = \Pr[\mathcal{G}_7]$.

Now there are three possibilities when $\mathcal{A}$ creates a valid forgery.

1. In the case that a signature for $\hat{C}_{J_{i^*}}$ was never requested, thus an EUF-CMA forger for $\sum$ is $\mathcal{A}$ and the forgery is $(\hat{C}_{J_{i^*}}, \sigma_{J_{i^*}})$. The upper bound of the probability of this event is $\epsilon_f(k)$.
2. Otherwise, according to the perfect correctness of Ω, $\hat{C}_{J_{i^*}}$ is deemed to honestly computed by the environment thus contains $\hbar/a\hat{P}$. Furthermore, there are two following possibilities:

   - If $e(\sigma_{1\text{A}}[1][1], \hat{P}) = e(\sigma_{1\text{A}}[1][2], \hbar/a\hat{P})$, $\mathcal{A}$ is an adversary breaking co-CDHI, because we can obtain $((\hbar{\cdot}1/aP, P), \sigma'_{\text{A}_j}) \leftarrow \text{ChgRep}_{\mathcal{R}}(\sigma_{1A}, \rho^{-1}, pk_{\mathcal{R}})$ and use $\hbar$ to output $\hbar^{-1} \cdot (\hbar{\cdot}1/aP) = 1/aP$. The upper bound of the probability of this event is $\epsilon_{co-CDHI}(k)$.
   - (Otherwise,$\mathcal{A}$ has created an opening proof of a statement that does not belong to L_{RO}. The upper bound of the probability of this event is $\epsilon_S(k)$.

The result of merging the above upper bound is $\epsilon_{nf8}(k) \leq \epsilon_f(k) + \epsilon_{co-CDHI}(k) + \epsilon_S(k)$. Therefore, the upper probabilistic bound of the success of $\mathcal{A}$ in $\mathbf{G_1}$ is negligible, i.e.,$\Pr[\mathcal{G}_0] \leq n \cdot (\epsilon_{nf8}(k) + \epsilon_{\text{zk}_J}(k) + \epsilon_{\text{SIM}}(k) + \epsilon_{\text{guess}}(k) + \epsilon_{\text{EXT}}(k))$.

Theorem 4 *The proposal achieves traceability if SPS-EQ achieves EUF-CMA security and Π achieves soundness.*

Proof (*Sketch*) The adversary is essentially concerned with two forgeries in $\mathrm{Exp}_{\mathcal{GS}\cdot\mathcal{A}}^{\mathrm{Traceability}}(k)$: the forgery of the current group membership certificate and that of non-revoked users' tokens. The first type of forgery can be `reduced to the EUF-CMA security of SPS-EQ and the soundness of NIZK proof because group membership certificates are created based on SPS-EQ and NIZK proof system. The second type of forgery, if an adversary can forge a valid signature after expiry time, then there exist a valid SPS-EQ signatures which is not contained in the revocation list RL_{t^*}. Thus, the second forgery attack is also reduced to EUF-CMA security of the SPS-EQ. Therefore, traceability is guaranteed both by the EUF-CMA security of the SPS-EQ scheme and the soundness of the NIZK proof system.

Proof Use $q \leq \mathrm{poly}(\kappa)$ to denote the number of queries to the **SndToI** oracle.

$\mathbf{G}_0$: The original traceability experiment.
$\mathbf{G}_1$: As $\mathbf{G}_0$, except that crs_J is obtained from a soundness challenger of Π. Therefore, $\mathbf{G}_1$ is conceptually identical to $\mathbf{G}_0$, i.e., $\Pr[\mathcal{G}_1] = \Pr[\mathcal{G}_0]$.
$\mathbf{G}_2$: As $\mathbf{G}_1$ except that BG and $pk_{\mathcal{R}}$ is obtained from an EUF-CMA challenger of the SPS-EQ. $\mathbf{G}_2$ is conceptually identical to $\mathbf{G}_1$, i.e.$\Pr[\mathcal{G}_2] = \Pr[\mathcal{G}_1]$
$\mathbf{G}_3$: According to the winning condition $i \notin \mathcal{C}\backslash\mathcal{R}_t$, $\mathcal{A}$ needs to inquiry the signing oracle of the SPS-EQ to generate a counterfeit σ_{Aj} (namely type I forger) which is not published via the **SndToI** oracle. The Type II forger can be considered as in real game. As $\mathbf{G}_0$ except that upon successful execution of **SndToI**, we obtain $\hat{R} = \Omega.Dec(sk_o, \hat{C}_{J_i})$ and abort when $e(U_i, \hat{P}) \neq e(Q, \hat{R})$. If abortion happens, we obtain a valid proof π_{J_i} attesting that $(U_i, Q, \hat{C}_{J_i}, pk_o) \in L_{R_J}$, but by the perfect correctness of Ω there exists no ω so that $\hat{C}_{J_i} = \Omega.\mathrm{Enc}(pk_o, r \cdot \hat{P}; \omega) \wedge U_i = r \cdot Q$, i.e., $(U_i, Q, \hat{C}_{J_i}, pk_o)$ is actually not in L_{R_J}. Therefore, both $\mathbf{G}_3$ and $\mathbf{G}_2$ proceed identically as long as $\mathcal{A}$ does not break the soundness of NIZK in one oracle query, i.e., $|\Pr[\mathcal{G}_3] = \Pr[\mathcal{G}_2]| \leq q \cdot \epsilon_s(k)$.
$\mathbf{G}_4$: According to the winning condition $i \notin \mathcal{C}\backslash\mathcal{R}_t$, $\mathcal{A}$ needs to inquiry the signing oracle of the SPS-EQ signature to create a forged non-revoked certificate σ_{Bk} while the **Revoke** oracle is called. The Type I forger can be considered as in real game. As $\mathbf{G}_3$, but obtain σ_{Bk} from an EUF-CMA challenger of the SPS-EQ. Therefore, $\mathbf{G}_4$ is conceptually identical to $\mathbf{G}_3$, i.e., $\Pr[\mathcal{G}_4] = \Pr[\mathcal{G}_3]$
$\mathbf{G}_5$: According to the winning condition $i \in \mathcal{C}\backslash\mathcal{R}_t \wedge t \geq \tau_i$, obviously, if $\mathcal{A}$ is able to create a valid signature when $\tau_i < t^*$, there definitely exists a valid σ_{Bk} not contained in RL_{t^*}.The unforgeability of expiry time is finally reduced to unforgeability of the SPS-EQ scheme. Therefore, $\mathbf{G}_5$ is conceptually identical to $\mathbf{G}_4$, i.e.,$\Pr[\mathcal{G}_5] = \Pr[\mathcal{G}_4]$.

If $\mathcal{A}$ finally produces a valid forgery signatures σ, which contains an SPS-EQ signature $\sigma_{1\mathrm{A}}$ for some (rP, P) so that the registration table exits no entry i for corresponding $r\hat{P}$ s.t. $e(\sigma_{1\mathrm{A}}[1][1], \hat{P}) = e(\sigma_{1\mathrm{A}}[1][2], r\hat{P})$ holds. Consequently, $\sigma_{1\mathrm{A}}$ is a valid SPS-EQ signature for an unqueried equivalence class and we can conclude that $\Pr[\mathcal{G}_3] \leq \epsilon_F(k)$ and then $\Pr[\mathcal{G}_0] \leq \epsilon_F(k) + q \cdot \epsilon_S(k)$ which proves the theorem.

Theorem 5 *The proposal achieves weak opening soundness if Ω achieves perfect correctness and $\sum$ achieves EUF-CMA security.*

Proof (*Sketch*) $\mathcal{A}$ breaks weak opening soundness when he can forge an opening proof to eliminate the uniqueness of group members, which indicates that he can resist against the soundness of Π in the phase of GS.Judge. The EUF-CMA security of digital signatures and the perfect correctness of the PKE scheme guarantee that user i signed σ uniquely. Once GS.Join is honestly executed for users i and j, the probability that r (resp.$\hat{R}$) values of users i and j are identical is negligible.

5 Results and discussion

In Table 2, we summarize the characteristics of our scheme and other revocable group signatures schemes [23, 24, 27–29, 34] that are security in ROM and asymmetric pairing-based. Our proposal can resist the attack of the forgeability of expiry time for signing keys for following the way of GS-TBK in [34], but the counterparts are not taken into account that attack. Moreover, we employed re-randomizable SPS-EQ instead of traditional ones based on BBS+signature. The benefit of this method is gaining efficiency following the SRP paradigm, and thus avoiding the assumption of q-SDH assumption and the knowledge of secret key (KOSK) that employed in [24, 29, 34]. The substantial drawback of the KOSK assumption is difficult to realize by existing infrastructure [35], and the q-type assumption leads to the Cheon attack [36].

Weak opening soundness is reasonable in many scenarios, where it needs to reward signers or prevent the abusers from transferring blame to someone else. The schemes of [23, 24, 29] and our proposal capture weak opening soundness, but others fail to possess the property. As mentioned before, introducing revocation functionality inevitably leads to the scheme fail to satisfy the anonymity because the revocation token could be derived from the signing key. In other words, it fails to prevent the leakages of group signing keys. Although our proposal can only achieve BU- and selfless anonymity, it seems to be a reasonable price considering the benefits. The scheme of [27] shows the construction of the VLR-GS with a fully anonymous, which is desirable but rather strict for reasonable application areas of group signature schemes. As a result, a slightly weaker notion of anonymity was suitable for more general use cases.

Table 2 Characteristic comparison

Schemes	BU	TBK	UET	WOS	Full-Dynamic	Assumption
[34]	√	√	√	×	√	KOSK, q-SDH
[28]	×	×	×	×	√	–
[27]	√	√	×	×	×	–
[23]	×	×	×	√	×	DDH, SXDH,SDL
[29]	√	×	×	√	×	DDH, XDH, DL, q-SDH, KOSK
[24]	×	×	×	√	×	KOSK, q-SDH, DLIN
Our Scheme	√	√	√	√	√	DDH, SXDH, co-CDHI

UET Unforgeablity of expiry time, *BU* backward unlinkability, *TBK* Time-bound key, *WOS* weak opening soundness

Table 3 Performance comparison

Schemes	Signature size	Sign cost	Verify cost
[34]	$12\mathbb{G}_1 + 4\mathbb{Z}_p$	$28\mathrm{mul}\mathbb{G}_1 + 2\exp\mathbb{G}_2 + 4P$	$20\mathrm{mul}\mathbb{G}_1 + 12\exp\mathbb{G}_2 + 2\mathrm{mul}\mathbb{G}_T + 4P$
[23]	$12\mathbb{G}_1 + 4\mathbb{Z}_p$	$32\exp\mathbb{G}_1 + 10P + 5\exp\mathbb{G}_T$	$10\exp\mathbb{G}_1 + 20P + 7\exp\mathbb{G}_T$
[24]	$20\mathbb{G}_1 + 4\mathbb{Z}_p$	$18\exp\mathbb{G}_1 + 2P + 15\exp\mathbb{G}_T$	$14\exp\mathbb{G}_1 + 4P + 17\exp\mathbb{G}_T$
[29]	$5\mathbb{G}_1 + 8\mathbb{Z}_p$	$2\mathrm{mul}\mathbb{G}_1 + 7\exp\mathbb{G}_T + 2P$	$5\mathrm{mul}\mathbb{G}_1 + 7exp\mathbb{G}_T + 5P$
Our scheme	$10\mathbb{G}_1 + 3\mathbb{G}_2 + 4\mathbb{Z}_p$	$16\mathrm{mul}\mathbb{G}_1 + 3\mathrm{mul}\mathbb{G}_2$	$4\mathrm{mul}\mathbb{G}_1 + 12P$

Table 3 shows an evaluation of the signature size, computational costs for the signing, and verification of revocable group signature schemes. The schemes [27, 28] have not provided instantiations. The scheme [24] achieves scalability in ROM, but the costs are asymptotically equal to that of the scheme [22]. As shown in Table 3, our proposal has the lowest cost in the signatures generation and verification processes due to avoiding complex $\mathbb{G}_T$ operation. Regarding signature size, our group signature contains respectively 10, 3, and 4 group elements in $\mathbb{G}_1$ $\mathbb{G}_2$ and $\mathbb{Z}_p$, which benefits from the SEP paradigm. Besides, the complexity of the CS method is acceptable. Consequently, our proposal is efficient on the computation cost and is suited for resource-constrained systems.

6 Conclusion

In this article, we presented a revocable group signature that realizes the unforgeability of expiry time for signing keys, BU-anonymity, non-frameability, traceability, weak opening soundness, and backward security. Moreover, the results showed that it is feasible in resource-constrained settings for constant and efficient computational cost of signing algorithm. Our scheme essentially follows the BSZ model, which places reliance heavily on the monopolistic issuer and opener. In other words, there are no strategies against either corrupt opener disclosing privacy illegally or corrupt issuer counterfeiting credentials in the BSZ model. Those imperfections are bound to be barriers in the future. Thus, it is attractive to adopt group signatures with multiple issuers and openers for distributed applications in the future.

Abbreviations

IoT: Internet of things; SEP: Sign-encrypt-prove; SRP: Sign-randomize-proof; ROM: Random oracle model; BU: Backward unlinkability; SPS-EQ: Structure preserving signatures on equivalence classes; SoK: Signature of knowledge; NIZK: Non-interactive zero-knowledge proof; PKE: Public key encryption; RGS: Revocable group signature; RC: Revocation check; KOSK: The knowledge of secret key; TBK: Time-bound key; GS-TBK: Group signature with time-bound keys.

Acknowledgements

The authors would like to thank the reviewers for their meticulous and constructive suggestions, for improving this article.

Authors' contributions

JF proposed the scheme and is the main writer of the manuscript. TF is the corresponding author and gave important advices with the respect to the construction, result, and writing. Both authors read and approved the final manuscript.

Funding

This work is supported by the National Natural Science Foundation of China (Grant No. 61762060), Educational Commission of Gansu Province, China (Grant No. 2017C-05), Natural Science Foundation of Gansu Province, China (Grant No. 20JR5RA467), Foundation for the Key Research and Development Program of Gansu Province, China (Grant No. 20YF3GA016).

Availability of data and materials

Data sharing is not applicable to this article as no datasets are generated or analyzed during the current study.

Declarations

Competing interests
The authors declared no potential conflicts of interest with respect to this article.

Received: 16 November 2020 Accepted: 11 March 2021
Published online: 03 May 2021

References
1. A. Zanella, N. Bui, A. Castellani, L. Vangelista, M. Zorzi, Internet of things for smart cities. IEEE Internet Things J. **1**(1), 22–32 (2014). https://doi.org/10.1109/JIOT.2014.2306328
2. L. Zhao, X. Dong, An industrial internet of things feature selection method based on potential entropy evaluation criteria. IEEE Access **6**, 4608–4617 (2018). https://doi.org/10.1109/ACCESS.2018.2800287
3. R. Liu, Y. Wang, M. Shu, H. Zhao, C. Chen, Power control with nearest neighbor nodes distribution for coexisting wireless body area network based on stochastic geometry. KSII Trans. Internet Inf. Syst. **12**(11), 5218–5233 (2018). https://doi.org/10.3837/tiis.2018.11.003
4. Y. Hu, Y. Zheng, X. Wu, H. Liu, A rendezvous node selection and routing algorithm for mobile wireless sensor network. KSII Trans. Internet Inf. Syst. **12**(10), 4738–4753 (2018). https://doi.org/10.3837/tiis.2018.10.007
5. A. Anand, M. Conti, P. Kaliyar, C. Lal, TARE: topology adaptive re-keying scheme for secure group communication in IoT networks. Wirel. Netw. **26**(4), 2449–2463 (2020). https://doi.org/10.1007/s11276-019-01975-y
6. A. Castiglione, P. D'Arco, A.D. Santis, R. Russo, Secure group communication schemes for dynamic heterogeneous distributed computing. Future Gener. Comput. Syst. **74**, 313–324 (2017). https://doi.org/10.1016/j.future.2015.11.026
7. L.U. Khan, I. Yaqoob, N.H. Tran, S.M.A. Kazmi, T.N. Dang, C.S. Hong, Edge-computing-enabled smart cities: a comprehensive survey. IEEE Internet Things J. **7**(10), 10200–10232 (2020). https://doi.org/10.1109/JIOT.2020.2987070
8. J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, W. Zhao, A survey on internet of things: architecture, enabling technologies, security and privacy, and applications. IEEE Internet Things J. **4**(5), 1125–1142 (2017). https://doi.org/10.1109/JIOT.2017.2683200
9. D. Chaum, E. van Heyst, in *EUROCRYPT'91: Advances in Cryptology—EUROCRYPT'91*, ed. by DW Davies. Workshop on the Theory and Application of of Cryptographic Techniques, Brighton, April 1991. Lecture Notes in Computer Science, vol. 547 (Springer, Heidelberg, 1991), p. 257.
10. J.K. Liu, C. Chu, S.S.M. Chow, X. Huang, M.H. Au, J. Zhou, Time-bound anonymous authentication for roaming networks. IEEE Trans. Inf. Forensics Secur. **10**(1), 178–189 (2015). https://doi.org/10.1109/TIFS.2014.2366300
11. K. Kluczniak, J. Wang, X. Chen, M. Kutylowski, Multi-device anonymous authentication. Int. J. Inf. Secur. **18**, 181–197 (2019). https://doi.org/10.1007/s10207-018-0406-4
12. T. Feng, X. Chen, C. Liu, X. Feng, Research on privacy enhancement scheme of blockchain transactions. Secur. Privacy **2**(6), e89 (2019). https://doi.org/10.1002/spy2.89
13. M. Bellare, D. Micciancio, B. Warinschi, in *EUROCRYPT'03: Advances in Cryptology-EUROCRYPT* 2003, ed. by E Biham. 22th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, May 2003. Lecture Notes in Computer Science, vol. 2656 (Springer, Heidelberg, 2003), p. 614.
14. M. Bellare, H. Shi, C. Zhang, in *CT-RSA'05: Topics in Cryptology*, ed. by A Menezes. The Cryptographers' Track at the RSA Conference 2005, San Francisco, CA, February 2005. Lecture Notes in Computer Science, vol. 3376 (Springer, Heidelberg, 2005), p. 136.
15. Y. Sakai, J.C.N. Schuldt, K. Emura, G. Hanaoka, K. Ohta, in *PKC'12: Public Key Cryptography—PKC 2012*, ed. by M Fischlin, J Buchmann, M Manulis. 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, May 2012. Lecture Notes in Computer Science, vol. 7293 (Springer, Heidelberg, 2012), p. 715.
16. P. Bichsel, J. Camenisch, G. Neven, N.P. Smart, B. Warinschi, in *SCN'10: Security and Cryptography for Networks*, ed. by JA Garay, R De Prisco. 4th Security and Cryptography for Networks, Amalfi, September 2010. Lecture Notes in Computer Science, vol. 6280 (Springer, Heidelberg, 2010), p. 381.
17. D. Derler, D. Slamanig, in *AsiaCCS'18: Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, ed. by J Kim, GJ Ahn, S Kim, Y Kim, J López, T Kim. The 13th ACM Asia Conference on Computer and Communications Security, Incheon, June 2018. Lecture Notes in Computer Science, (ACM, 2018), p. 551.
18. C. Hanser, D. Slamanig, in *ASIACRYPT'14: Advances in Cryptology-ASIACRYPT 2014*, ed. by P. Sarkar, T Iwata. 20th International Conference on the Theory and Application of Cryptology and Information Security, Taiwan, December 2014. Lecture Notes in Computer Science, vol. 8873 (Springer, Heidelberg, 2014), p. 491.
19. G. Fuchsbauer, C. Hanser, D. Slamanig, Structure-preserving signatures on equivalence classes and constant-size anonymous credentials. J. Cryptol. **32**, 498–546 (2019). https://doi.org/10.1007/s00145-018-9281-4
20. D. Boneh, X. Boyen, H. Shacham, in *CRYPTO'04: Advances in Cryptology-CRYPTO 2004*, ed. by M Franklin. 24th Annual International Cryptology Conference, Santa Barbara, August 2004. Lecture Notes in Computer Science, vol. 3152 (Springer, Heidelberg, 2004), p. 41.
21. T. Nakanishi, H. Fujii, Y. Hira, N. Funabiki, in *PKC'09: Public Key Cryptography—PKC* 2009, ed. by S Jarecki, G Tsudik. 12th International Conference on Practice and Theory in Public Key Cryptography, Irvine, March 2009. Lecture Notes in Computer Science, vol. 5443 (Springer, Heidelberg, 2009), p. 463.
22. B. Libert, T. Peters, M. Yung, in *CRYPTO'12: Advances in Cryptology-CRYPTO 2012*, ed. by R. Safavi-Naini, R. Canetti. 32nd Annual Cryptology Conference, Santa Barbara, August 2012. Lecture Notes in Computer Science, vol. 7417, (Springer, Heidelberg, 2012), p. 571.
23. T.H. Emura, in *ISC'18: Information Security-ISC 2018*, ed. by L. Chen, M. Manulis, S. Schneider. 21st International Conference, Guildford, September 2018. Lecture Notes in Computer Science, vol. 11060 (Springer, Cham, 2018), p. 442.

24. K. Ohara, K. Emura, G. Hanaoka, A. Ishida, K. Ohta, Y. Sakai, Shortening the Libert–Peters–Yung revocable group signature scheme by using the random oracle methodology. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **E102**(9), 1101–1117 (2019). https://doi.org/10.1587/transfun.E102.A.1101
25. J. Camenisch, A. Lysyanskaya, in *CRYPTO'02: Advances in Cryptology-CRYPTO 2002*, ed. by M Yung. 22nd Annual International Cryptology Conference, Santa Barbara, August 2002. Lecture Notes in Computer Science, vol. 2442 (Springer, Heidelberg, 2002), p. 61.
26. T. Nakanishi, N. Funabiki, in *IWSEC'06: Advances in Information and Computer Security*, ed. by H Yoshiura, K Sakurai, K Rannenberg, Y Murayama, S Kawamura. First International Workshop on Security, Kyoto, October 2006. Lecture Notes in Computer Science, vol. 4266 (Springer, Heidelberg, 2006), p.17.
27. A. Ishida, Y. Sakai, K. Emura, G. Hanaoka, K. Tanaka, in *SCN'18: Security and Cryptography for Networks-SCN 2018*, ed. by D Catalano, R De Prisco. 11th International Conference on Security and Cryptography for Networks, Amalfi, September 2018. Lecture Notes in Computer Science, vol. 11035 (Springer, Cham, 2018), p.23.
28. M.N.S. Perera, T. Koshiba, in *IDCS'18: Internet and Distributed Computing Systems-IDCS 2018*, ed. by Y Xiang, J Sun, G Fortino, A Guerrieri, J Jung. 11th International Conference on Internet and Distributed Computing Systems, Tokyo, October 2018. Lecture Notes in Computer Science, vol. 11226 (Springer, Cham, 2018), p. 134.
29. X. Yue, M. Xi, B. Chen, M. Gao, J. Xu, A revocable group signatures scheme to provide privacy-preserving authentications. Mob. Netw. Appl. (2020). https://doi.org/10.1007/s11036-019-01459-5
30. W.T. Zhu, R.H. Deng, J. Zhou, F. Rao, Time-bound hierarchical key assignment: an Overview. Ice Trans. Inf. Syst. **93**(5), 1044–1052 (2010). https://doi.org/10.1587/transinf.E93.D.1044
31. C.K. Chu, J.K. Liu, X. Huang, in *ASIACCS'12: Information, Computer and Communications Security*, ed. by N Foo, R Goebel. 7th ACM Symposium on Information, Computer and Communications Security, Seoul, May 2012 Lecture Notes in Computer Science, vol. 1114 (ACM, 2012), p. 26.
32. R. Zhang, L. Liu, R. Xue, Role-based and time-bound access and management of EHR data. Secur. Commun. Netw. **7**(6), 1–22 (2014). https://doi.org/10.1002/sec.817
33. L. Malina, J. Hajny, V. Zeman, Light-weight group signatures with time-bound membership. Secur. Commun. Netw. **9**(7), 599–612 (2016). https://doi.org/10.1002/sec.1383
34. K. Emura, T. Hayashi, A. Ishida, Group signatures with time-bound keys revisited: a new model, an efficient construction, and its implementation. IEEE Trans. Dependable Secure Comput. **17**(2), 292–305 (2020). https://doi.org/10.1109/TDSC.2017.2754247
35. T. Ristenpart, S. Yilek, in *EUROCRYPT'07: Advances in Cryptology -EUROCRYPT 2007*, ed. by M Naor. 26th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Barcelona, May 2007. Lecture Notes in Computer Science, vol. 4515 (Springer, Heidelberg, 2007), p. 228.
36. J.H. Cheon, Discrete logarithm problems with auxiliary inputs. J. Cryptol. **23**(3), 457–547 (2010)

Publisher's Note