IEICE TRANS. FUNDAMENTALS, VOL.E107–A, NO.3 MARCH 2024

PAPER Special Section on Information Theory and Its Applications

Short DL-Based Blacklistable Ring Signatures from DualRing*

Toru NAKANISHI[†a)], ***Member***, **Atsuki IRIBOSHI**[†], ***Nonmember***, ***and*** **Katsunobu IMAI**[†], ***Member***

SUMMARY As one of privacy-enhancing authentications suitable for decentralized environments, ring signatures have intensively been researched. In ring signatures, each user can choose any ad-hoc set of users (specified by public keys) called a ring, and anonymously sign a message as one of the users. However, in applications of anonymous authentications, users may misbehave the service due to the anonymity, and thus a mechanism to exclude the anonymous misbehaving users is required. However, in the existing ring signature scheme, a trusted entity to open the identity of the user is needed, but it is not suitable for the decentralized environments. On the other hand, as another type of anonymous authentications, a decentralized blacklistable anonymous credential system is proposed, where anonymous misbehaving users can be detected and excluded by a blacklist. However, the DL-based instantiation needs $O(N)$ proof size for the ring size N. In the research line of the DL-based ring signatures, an efficient scheme with $O(\log N)$ signature size, called *DualRing*, is proposed. In this paper, we propose a DL-based blacklistable ring signature scheme extended from DualRing, where in addition to the short $O(\log N)$ signature size for N, the blacklisting mechanism is realized to exclude misbehaving users. Since the blacklisting mechanism causes additional costs in our scheme, the signature size is $O(\log N + \ell)$, where ℓ is the blacklist size.

key words: *ring signatures, DualRing, blacklist, decentralized anonymous credentials*

1. Introduction

As privacy-enhancing authentications, group signatures [7] and ring signatures [14] have been researched with significant effort. In group signatures, a trusted group manager issues a certificate to each user in the group, and the user can anonymously sign a message as the group member. On the other hand, in ring signatures, each user can choose any ad-hoc set of users (specified by public keys) called a ring, and anonymously sign a message as one of the users. The ring signatures do not need any trusted manager, and thus are suitable in the decentralized environment such as blockchain.

In the anonymous authentications, misbehaving users have to be addressed. For example, in an online forum, an anonymous user, who is authorized by an anonymous authentication, may write a message that violates the code of conduct. In the group signatures, the manager (or a designated opener) can trace the identity of the misbehaving user, and revoke the membership. On the other hand, in the original ring signature scheme [14], such a tracing function is not equipped, since the trusted tracing entity does not exist. In [3], to address this issue, an accountable ring signature scheme is proposed. In the scheme, similarly to the group signatures, an opener is introduced, where only the opener can identify the signer from a signature. Thus, if a misbehaving user is detected, the user is identified from the signature, and the user is excluded from the ring. However, in the accountable ring signatures, the trust of the opener is needed. Thus, the accountable ring signatures are not suitable for decentralized environment.

As another approach to the privacy-enhancing anonymous authentications, anonymous credential systems have been researched (e.g., [5], [8]). In the anonymous credential systems, similar to the group signatures, a trusted entity called an issuer issues a certificate to each user, where the certificate is a proof of membership or privilege, and furthermore certifies the user's attributes. In the authentication, the user can anonymously prove the certified ones to verifiers. In the setting of the anonymous credential systems, misbehaving users also have to be addressed. Thus, as an extension of the anonymous credential systems, a blacklistable anonymous credential system (BLAC) is proposed in [15]. Compared to the conventional approach using the trusted opener, the characteristic of BLAC is that it does not need such a trusted entity. Instead, a *blacklist* is used, as follows. In each authentication, the user generates a (anonymous) ticket which is generated from the user's secret key. When a misbehavior in the service use linking to the ticket is detected, the ticket is added to the blacklist. In the authentication, in addition, the user has to prove that each ticket in the blacklist is NOT generated from the user's secret key. This is achieved using a zero-knowledge proofs, and thus the misbehaving user can be detected and excluded while the anonymity holds. However, the basic mechanism of the anonymously proving the certificate is similar to the group signatures, and thus the trust of the issuer is needed.

On the background in the approach of anonymous credential systems, in [16], a decentralized blacklistable anonymous credential system is proposed. In the system, similarly to the ring signatures, a user can anonymously prove that the user is one of a ring without any certificate from a trusted entity, and the misbehaving user can be excluded by the ticket-based blacklisting approach similarly to BLAC. Furthermore, the blacklisting based on reputation is available, where each session is scored, and a user is blacklisted based

Manuscript received February 22, 2023.
Manuscript revised June 12, 2023.
Manuscript publicized September 6, 2023.

[†]The authors are with Graduate School of Advanced Science and Engineering, Hiroshima University, Higashi-hiroshima-shi, 739-8527 Japan.

*A preliminary version of this paper was presented at CANDAR 2022 [12].

a) E-mail: t-nakanishi@hiroshima-u.ac.jp

DOI: 10.1587/transfun.2023TAP0008

on the scores of each user, i.e., reputation. In [16], several instantiations are shown. The RSA-based instantiation has the $O(1)$ proof size for the ring size N, by using an RSA accumulator. Two types of DL-based instantiations are shown in [16]. One type is the combination of a decentralized anonymous credential system [10] and the BLAC mechanism. In this type, an RSA-based accumulator is needed (i.e., the strong RSA assumption is needed), and a proof of knowledge for double discrete logs [4] is used. However, the proof of knowledge is inefficient, since the prover has to conduct 80 to 128 iterations of 3 move protocols. The second type is the combination of the classical OR proof of knowledge [9] and the BLAC mechanism. However, the type requires $O(N)$ proof size.

On the other hand, in the research line of ring signatures, the efficiency has been improved. In the original scheme [14], the signature size is $O(N)$. With the advance of the OR proofs of knowledge, DL-based ring signature schemes with $O(\log N)$ signature size have been proposed [3], [11], [17], where *DualRing* [17] achieves the better concrete signature size. Furthermore, the schemes in [3], [11] has $O(N \log N)$ signing cost, but the signing cost in DualRing is $O(N)$ (verification costs are $O(N)$ in all the schemes). However, in these short ring signature schemes, only the accountable ring signature scheme [3] has the function that misbehaving users can be identified, but the function needs a trusted entity, which implies that all the short ring signature schemes cannot prevent users from misbehaving in the decentralized environment.

Our Contributions: In this paper, we propose a short blacklistable ring signature scheme based on DualRing. In our scheme, due to DualRing, the signature size for N is $O(\log N)$, and the signing cost for N is $O(N)$ instead of $O(N \log N)$. Furthermore, our scheme has the blacklisting mechanism similarly to BLAC [15], and thus a misbehaving user is blacklisted and the signature issued by the user after blacklisted can be detected (can be excluded from the service). However, the mechanism causes additional costs in our scheme, and the signature size is $O(\log N + \ell)$ and the signing cost is $O(N + \ell)$, where ℓ is the blacklist size (i.e., the number of tickets in the blacklist). Our scheme adopts the blacklisting mechanism in BLAC, and this is why the security of our scheme is shown under the DDH assumption instead of the DL assumption. Since a blacklistable ring signature scheme has not been known, we newly define the model and security requirements. The blacklistable ring signature scheme can be considered as a DL-based blacklistable anonymous credential system, where the proof size is $O(\log N)$ for N. Therefore, we improve the proof size in the previous decentralized blacklistable anonymous credential system [16], without using the RSA-based accumulator. The extension to a reputation-based blacklisting mechanism that the previous system [16] has is one of our future works.

On decentralized blacklist managements: The management of blacklists is one of issue to be considered in the applications. The Service Provider (SP), who is the verifier of authentications with users, may generate arbitrary blacklists at any time, and thus sends a faked blacklist to the user in the authentication. In the centralized setting of BLAC, the trust of the SP is needed for the blacklist. In the decentralized setting, blockchain can be used for the blacklist management, as follows. Blacklists can be stored in blockchain, and the shared blacklists are used in each authentication, where the correctness of the blacklists are verified by blockchain nodes. Thus, for the blacklist management, the trust of the SP is not needed.

2. Preliminaries

We use notation $a \xleftarrow{R} A$ as randomly selecting an element a from a set A, and the bold letter such as $\boldsymbol{a}$ for a vector.

2.1 Assumptions

We adopt the DL assumption and DDH assumption, where the DDH assumption implies DL assumption. Here, let $\mathcal{G}(\lambda)$ be a generator to output $(\mathbb{G}, p, g)$ of a cyclic group $\mathbb{G}$ of a prime order $p > 2^\lambda$ and a generator g, given security parameter λ.

Definition 1 (DL assumption)**.** *The Discrete Logarithm (DL) assumption holds if for any PPT (Probabilistic Polynomial-Time) adversary $\mathcal{A}$,*

$$\Pr[(\mathbb{G}, p, g) \leftarrow \mathcal{G}(\lambda); x \xleftarrow{R} \mathbb{Z}_p; h = g^x : \mathcal{A}(\mathbb{G}, p, g, h) = x]$$

is negligible for λ.

Definition 2 (DDH assumption)**.** *The Decisional Diffie-Hellman (DDH) assumption holds if for any PPT adversary $\mathcal{A}$,*

$$\begin{aligned}&|\Pr[(\mathbb{G}, p, g) \leftarrow \mathcal{G}(\lambda); x, y \xleftarrow{R} \mathbb{Z}_p;\\ &\quad u = g^x; v = g^y; w = g^{xy} : \mathcal{A}(\mathbb{G}, p, g, u, v, w) = 1]\\ &-|\Pr[(\mathbb{G}, p, g) \leftarrow \mathcal{G}(\lambda); x, y, z \xleftarrow{R} \mathbb{Z}_p;\\ &\quad u = g^x; v = g^y; w = g^z : \mathcal{A}(\mathbb{G}, p, g, u, v, w) = 1]|\end{aligned}$$

is negligible for λ.

2.2 Pedersen Commitments

In this paper, we use Pedersen commitments [13], as follows. The sender computes a commitment value to an input message using a random value. The receiver cannot guess the message from the commitment, but the sender can later open the commitment by revealing the message and random value. For public parameters $(\mathbb{G}, p, g) \leftarrow \mathcal{G}(\lambda)$ and $h \xleftarrow{R} \mathbb{G}$, the commitment to a message $m \in \mathbb{Z}_p$ is computed as $C = g^m h^r$ for a random $r \xleftarrow{R} \mathbb{Z}_p$. The security requirements of the commitments are *hiding* and *binding*. The hiding means that any adversary cannot guess any information on m. The binding means that any PPT adversary cannot open different m' from $C = g^m h^r$. The hiding of Pedersen commitment

is information-theoretical, and the binding is computational under the DL assumption.

2.3 DualRing-EC

In [17], an efficient ring signature scheme with $O(\log N)$ signature size for ring size N, called *DualRing*, is proposed. The DL-based instantiation of DualRing is *DualRing-EC*[†].

The algorithms of a ring signature scheme are as follows.

- $\mathsf{Setup}(\lambda)$: Given security parameter λ, this algorithm outputs the public parameters param.
- $\mathsf{KeyGen}(\mathsf{param})$: Given param, this algorithm outputs a pair $(\mathsf{pk}, \mathsf{sk})$ of public key pk and secret key sk.
- $\mathsf{Sign}(\mathsf{param}, M, \mathbf{pk}, \mathsf{sk})$: Given $\mathsf{param}, M, \mathbf{pk}, \mathsf{sk}$, this algorithm outputs a ring signature σ on message M and ring $\mathbf{pk}$ that is a vector of public keys, where the corresponding public key to sk belongs to $\mathbf{pk}$.
- $\mathsf{Verify}(\mathsf{param}, M, \mathbf{pk}, \sigma)$: Given $\mathsf{param}, M, \mathbf{pk}, \sigma$, this algorithm outputs 1 if the signature σ on message M and ring $\mathbf{pk}$ is valid, and otherwise outputs 0.

Then, two security requirements (unforgeability w.r.t. insider corruption and anonymity against full key exposure) are defined in [17], and in addition we define the perfect correctness to show the validity of algorithms, which is shown in [3], [11].

Definition 3 (Perfect Correctness)**.** *A ring signature scheme is perfectly correct if for any PPT adversary $\mathcal{A}$,*

$$\Pr[\mathsf{param} \leftarrow \mathsf{Setup}(\lambda);\ (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{param});\ (M, \mathbf{pk}) \leftarrow \mathcal{A}(\mathsf{param}, \mathsf{pk}, \mathsf{sk});\ \sigma \leftarrow \mathsf{Sign}(\mathsf{param}, M, \mathbf{pk}, \mathsf{sk}) :\ \text{If } \mathsf{pk} \in \mathbf{pk} \text{ then } \mathsf{Verify}(\mathsf{param}, M, \mathbf{pk}, \sigma) = 1] = 1.$$

Definition 4 (Unforgeability w.r.t. Insider Corruption)**.** *A ring signature scheme is unforgeable if for any PPT adversary $\mathcal{A}$ and some integer $\mathsf{num}_{\mathsf{key}}$ polynomial in λ,*

$$\Pr[\mathsf{param} \leftarrow \mathsf{Setup}(\lambda);\ (\hat{\mathsf{pk}}_i, \hat{\mathsf{sk}}_i) \leftarrow \mathsf{KeyGen}(\mathsf{param})\ for\ all\ i \in [1, \mathsf{num}_{\mathsf{key}}];\ Q_{\mathsf{KeyGen}} := \{\hat{\mathsf{pk}}_i\}_{i=1}^{\mathsf{num}_{\mathsf{key}}};\ (M^*, \mathbf{pk}^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Reveal},\mathsf{HSign}}(\mathsf{param}, Q_{\mathsf{KeyGen}}) :\ \mathsf{Verify}(\mathsf{param}, M^*, \mathbf{pk}^*, \sigma^*) = 1 \wedge \mathbf{pk}^* \subseteq Q_{\mathsf{KeyGen}} \setminus Q_{\mathsf{Reveal}} \wedge (M^*, \mathbf{pk}^*) \notin Q_{\mathsf{Sig}}]$$

is negligible for λ, where the following oracles are used by $\mathcal{A}$.

- Reveal*: This oracle is queried on $\mathsf{pk}_k \in Q_{\mathsf{KeyGen}}$, returns the corresponding secret key sk_k, and adds pk_k to Q_{Reveal}.*
- HSign*: This oracle is queried on $(M_k, \mathbf{pk}_k, \mathsf{pk}_k)$. If $\mathsf{pk}_k \notin Q_{\mathsf{KeyGen}} \setminus Q_{\mathsf{Reveal}}$, or pk_k is not an element in $\mathbf{pk}_k$, abort. For secret key sk_k corresponding to pk_k, run $\mathsf{Sign}(\mathsf{param}, M_k, \mathbf{pk}_k, \mathsf{sk}_k)$ to obtain σ_k which is returned to $\mathcal{A}$, and add $(M_k, \mathbf{pk}_k)$ to Q_{Sig}.*

Definition 5 (Anonymity against Full Key Exposure)**.** *A ring signature scheme is anonymous if for any PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and some integer $\mathsf{num}_{\mathsf{key}}$ polynomial in λ,*

$$|\Pr[\mathsf{param} \leftarrow \mathsf{Setup}(\lambda);\ (\hat{\mathsf{pk}}_i, \hat{\mathsf{sk}}_i) \leftarrow \mathsf{KeyGen}(\mathsf{param}; \omega_i)\ for\ all\ i \in [1, \mathsf{num}_{\mathsf{key}}];\ Q_{\mathsf{KeyGen}} := \{\hat{\mathsf{pk}}_i\}_{i=1}^{\mathsf{num}_{\mathsf{key}}};\ (M^*, \mathbf{pk}^*, i_0, i_1, St) \leftarrow \mathcal{A}_1^{\mathsf{HSign}}(\mathsf{param}, Q_{\mathsf{KeyGen}});\ b \leftarrow \{0,1\};\ \sigma^* \leftarrow \mathsf{Sign}(\mathsf{param}, M^*, \mathbf{pk}^*, \hat{\mathsf{sk}}_{i_b});\ b' \leftarrow \mathcal{A}_2(\sigma^*, \{\omega_i\}_{i=1}^{\mathsf{num}_{\mathsf{key}}}, St) :\ b' = b \wedge \hat{\mathsf{pk}}_{i_0}, \hat{\mathsf{pk}}_{i_1} \in Q_{\mathsf{KeyGen}} \cap \mathbf{pk}^*] - 1/2|$$

is negligible for λ, where $\mathsf{KeyGen}(\mathsf{param}; \omega_i)$ means that randomness ω_i is used in KeyGen, HSign is the same oracle as in the unforgeability, and $\mathbf{pk}^*$ can include adversarially generated public keys.

Then, the algorithms of *DualRing-EC* are as follows, where NISA.Proof and NISA.Verify that are shown later are used. Here, NISA is the abbreviation of Non-Interactive Sum Argument.

$\mathsf{Setup}(\lambda)$:

1. Generate $(\mathbb{G}, p, g) \leftarrow \mathcal{G}(\lambda)$. Select $u \xleftarrow{R} \mathbb{G}$.
2. Output $\mathsf{param} = (\mathbb{G}, p, g, u)$.

$\mathsf{KeyGen}(\mathsf{param})$:

1. Select $\mathsf{sk} \xleftarrow{R} \mathbb{Z}_p$, and compute $\mathsf{pk} = g^{\mathsf{sk}}$.
2. Output $(\mathsf{pk}, \mathsf{sk})$.

$\mathsf{Sign}(\mathsf{param}, M, \mathbf{pk}, \mathsf{sk})$:

1. Parse $\mathbf{pk} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_N)$. Let the corresponding public key pk of sk be pk_j for $j \in [1, N]$.
2. Select $r, c_i \xleftarrow{R} \mathbb{Z}_p$ for all $i \neq j$. Compute $R = g^r \cdot \prod_{i \neq j} \mathsf{pk}_i^{c_i}$, $c = H(M, \mathbf{pk}, R)$, $c_j = c - \sum_{i \neq j} c_i$, and $z = r - c_j \cdot \mathsf{sk}$, where H is a hash function.
3. Set $\boldsymbol{a} = (c_1, \ldots, c_N)$. Compute $P = R \cdot g^{-z}$. Then, conduct $\mathsf{NISA.Proof}(\mathsf{param}, \mathbf{pk}, u, P, c, \boldsymbol{a})$ to obtain π, as

[†]EC means Elliptic Curve which is used for the DL-based implementation

the proof of $P = \prod_{i=1}^{N} \mathsf{pk}_i^{c_i}$ and $c = \sum_{i=1}^{N} c_i$.

4. Output $\sigma = (z, R, \pi)$.

Verify(param, $M, \mathbf{pk}, \sigma$):

1. Parse $\mathbf{pk} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_N)$, and $\sigma = (z, R, \pi)$.
2. Compute $c = H(M, \mathbf{pk}, R)$ and $P = R \cdot g^{-z}$.
3. Conduct NISA.Verify(param, $\mathbf{pk}, u, P, c, \pi$). If the output of NISA.Verify is 0, output 0. Otherwise, output 1.

The algorithms of NISA.Proof and NISA.Verify are as follows. For simplicity of description, as in [17], assume $N = 2^\kappa$ for some integer κ.

NISA.Proof(param, $\boldsymbol{g}, u, P, c, \boldsymbol{a}$): Given param $= (\mathbb{G}, p, g)$, $\boldsymbol{g} = (g_1, \ldots, g_N)$ for $g_i \in \mathbb{G}$, $u, P \in \mathbb{G}$, $c \in \mathbb{Z}_p$, and $\boldsymbol{a} = (a_1, \ldots, a_N)$ for $a_i \in \mathbb{Z}_p$, this algorithm output a proof π to prove that $P = \prod_{i=1}^{N} g_i^{a_i}$ and $c = \sum_{i=1}^{N} a_i$.

1. Output $\pi \leftarrow \mathsf{Pf}(\boldsymbol{g}, u^{H(P,u,c)}, \boldsymbol{a}, \mathbf{1}^n)$ for the following sub-algorithm Pf.

$\mathsf{Pf}(\boldsymbol{g}, \hat{u}, \boldsymbol{a}, \boldsymbol{b})$: Given $\boldsymbol{g} = (g_1, \ldots, g_N)$ for $g_i \in \mathbb{G}$, $\hat{u} \in \mathbb{G}$, and $\boldsymbol{a} = (a_1, \ldots, a_N)$ and $\boldsymbol{b} = (b_1, \ldots, b_N)$ for $a_i, b_i \in \mathbb{Z}_p$, do the following.

1. If $N = 1$, output $\pi = (\boldsymbol{L}, \boldsymbol{R}, a, b)$. Otherwise, go to the next step.
2. Compute $N' = N/2$, $c_L = \sum_{i=1}^{N'} a_i \cdot b_{i+N'}$, and $c_R = \sum_{i=1}^{N'} a_{i+N'} \cdot b_i$.
3. Compute $L = \prod_{i=1}^{N'} g_{i+N'}^{a_i} \hat{u}^{c_L}$ and $R = \prod_{i=1}^{N'} g_i^{a_{i+N'}} \hat{u}^{c_R}$. Add L to $\boldsymbol{L}$ and R to $\boldsymbol{R}$, and compute $x = H(L, R)$.
4. Compute $\boldsymbol{g'} = (g_1^{x^{-1}} \cdot g_{1+N'}^{x}, \ldots, g_{N'}^{x^{-1}} \cdot g_{N'+N'}^{x})$, $\boldsymbol{a'} = (x \cdot a_1 + x^{-1} \cdot a_{1+N'}, \ldots, x \cdot a_{N'} + x^{-1} \cdot a_{N'+N'})$, and $\boldsymbol{b'} = (x^{-1} \cdot b_1 + x \cdot b_{1+N'}, \ldots, x^{-1} \cdot b_{N'} + x \cdot b_{N'+N'})$.
5. Run algorithm $\mathsf{Pf}(\boldsymbol{g'}, \hat{u}, \boldsymbol{a'}, \boldsymbol{b'})$.

NISA.Verify(param, $\boldsymbol{g}, u, P, c, \pi = (\boldsymbol{L}, \boldsymbol{R}, a, b)$):

1. Parse $\boldsymbol{L} = (L_1, \ldots, L_{\log_2 N})$ and $\boldsymbol{R} = (R_1, \ldots, R_{\log_2 N})$. Compute $P' = P \cdot u^{c \cdot H(P,u,c)}$.
2. Compute $x_j = H(L_j, R_j)$ for all $j \in [1, \log_2 N]$, and $y_i = \prod_{j \in [1, \log_2 n]} x_j^{f(i,j)}$ for all $i \in [1, N]$, where $f(i, j) = 1$ if $(i-1)$'s j-th bit is 1, and otherwise $f(i, j) = -1$.
3. If

$$L_1^{x_1^2} R_1^{x_1^{-2}} \cdots L_{\log_2 N}^{x_{\log_2 N}^2} R_{\log_2 N}^{x_{\log_2 N}^{-2}} \cdot P'$$
$$= g_1^{a y_1} \cdots g_N^{a y_N} \cdot u^{ab \cdot H(P,u,c)}.$$

output 1. Otherwise, output 0.

NISA.Proof and NISA.Verify are obtained from an argument of knowledge called NISA protocol. An argument of knowledge consists of three PPT algorithms $(\mathsf{S}, \mathcal{P}, \mathcal{V})$. CRS (Common Reference String) generator S is given security parameter λ, and output CRS $\hat{\sigma}$. Prover $\mathcal{P}$ on input s and verifier $\mathcal{V}$ on input t executes an interactive protocol produces a transcript $\mathsf{tr} \leftarrow \langle \mathcal{P}(s), \mathcal{V}(t) \rangle$. If $\mathcal{V}$ accepts tr, for notation $b = \langle \mathcal{P}(s), \mathcal{V}(t) \rangle$, set $b = 1$, and otherwise set $b = 0$. Let $\mathcal{R}$ be a polynomial time decidable binary relation. Consider the language $\mathcal{L} = \{x | \exists w : (\hat{\sigma}, x, w) \in \mathcal{R}\}$, where w is a witness for a statement x if $(\hat{\sigma}, x, w) \in \mathcal{R}$. Then, the security of the argument is defined by *perfect completeness* and *statistical witness-extended emulation*, as in [17].

Definition 6 (Perfect completeness)**.** $(\mathsf{S}, \mathcal{P}, \mathcal{V})$ *has* perfect completeness *if for all PPT adversary* $\mathcal{A}$*,*

$$\Pr[\hat{\sigma} \leftarrow \mathsf{S}(\lambda); (x, w) \leftarrow \mathcal{A}(\hat{\sigma}) : (\hat{\sigma}, x, w) \notin \mathcal{R} \vee \langle \mathcal{P}(\hat{\sigma}, x, w), \mathcal{V}(\hat{\sigma}, x) \rangle = 1] = 1.$$

Definition 7 (Statistical witness-extended emulation)**.** $(\mathsf{S}, \mathcal{P}, \mathcal{V})$ *has* statistical witness-extended emulation *if for all deterministic polynomial time prover* $\mathcal{P}^*$*, there exists a polynomial time emulator* $\mathcal{E}$ *s.t. for all pairs of adversaries* $(\mathcal{A}_1, \mathcal{A}_2)$*,*

$$\begin{aligned} &|\Pr[\hat{\sigma} \leftarrow \mathsf{S}(\lambda); (x, s) \leftarrow \mathcal{A}_1(\hat{\sigma}); \\ &\quad \mathsf{tr} \leftarrow \langle \mathcal{P}^*(\hat{\sigma}, x, s), \mathcal{V}(\hat{\sigma}, x) \rangle : \mathcal{A}_2(\mathsf{tr}) = 1] \\ &- \Pr[\hat{\sigma} \leftarrow \mathsf{S}(\lambda); (x, s) \leftarrow \mathcal{A}_1(\hat{\sigma}); \\ &\quad (\mathsf{tr}, w) \leftarrow \mathcal{E}^{O}(\hat{\sigma}, x) : \mathcal{A}_2(\mathsf{tr}) = 1 \\ &\quad \wedge \mathsf{tr} \text{ is accepting then } (\hat{\sigma}, x, w) \in \mathcal{R}]| \end{aligned}$$

is negligible for λ*, where* O *is an oracle* $\langle \mathcal{P}^*(\hat{\sigma}, x, s), \mathcal{V}(\hat{\sigma}, x) \rangle$ *which can rewind to a specific point and resume with fresh randomness for the verifier from this point onward.*

In this definition, s represents the internal state of $\mathcal{P}^*$ including randomness. Thus, $\mathcal{E}$ can extract a witness w if $\mathcal{P}^*$ on s is accepted.

The NISA protocol is transformed to non-interactive argument (NISA.Proof, NISA.Verifier), using Fiat-Shamir heuristic in the random oracle model.

In [17], the security of NISA protocol is proved, and based on it, it is proved that the *DualRing-EC* is unforgeable and anonymous under the DL assumption in the random oracle model.

2.4 Signatures of Knowledge

As in BLAC [15], we adopt signatures of knowledge [4], [6] on discrete logs. The proofs are non-interactive proofs transformed from interactive proofs of knowledge via Fiat-Shamir heuristic. The prover can prove the secret knowledge $x_1, \ldots, x_\ell$ s.t. $C = g_1^{x_1} \cdots g_\ell^{x_\ell}$, where $C, g_1, \ldots, g_\ell \in \mathbb{G}$. Furthermore, the prover with the secret knowledge can sign a

message. The proof of knowledge consists of three moves, as follows. The prover sends an initial message to the verifier, the verifier returns a random challenge, and the prover sends a response message to the verifier. In the signature of knowledge, the challenge is computed as a hashed value on the initial message, public parameters, and the signed message. The signatures of knowledge satisfy the following properties in the random oracle model.

Simulatability: Given the public parameters, it is able to simulate a signature of knowledge on a message without the secret knowledge.

Extractability: From two transcripts of a signature of knowledge where the initial message are the same but the challenges are different, we can extract the secret knowledge. In the extraction, we rewind the prover to a hash query to the random oracle and resume with a fresh random.

3. Model

3.1 Syntax

The blacklistable ring signature scheme consists of the following algorithms, which are derived from *DualRing-EC* and added the BLAC mechanism to. In the scheme, a session ID sid is used to specify each authentication of Sign/Verify. Since a ring signature is anonymous, any information to specify the signer such as user ID cannot be used. But, in the blacklistable ring signature scheme, it is necessary to distinguish each authentication, for blacklisting the user in the service use linked to the authentication. This is why the session ID is used to generate the ticket that is used for blacklisting.

- $\mathsf{Setup}(\lambda)$: This PPT algorithm, given security parameter λ, outputs public parameters param.
- $\mathsf{KeyGen}(\mathsf{param})$: This PPT algorithm, given param, outputs a key pair $(\mathsf{pk}, \mathsf{sk})$ of a user, where pk (resp., sk) is the public key (resp., secret key) of the user.
- $\mathsf{Sign}(\mathsf{param}, M, \mathsf{sid}, \mathbf{pk}, \mathsf{sk}, \mathbf{BL})$: This PPT algorithm, given param, message M, session ID sid, a vector of public keys $\mathbf{pk}$, secret key sk, and blacklist $\mathbf{BL}$ of tickets, outputs the signature σ and a ticket τ including sid.
- $\mathsf{Verify}(\mathsf{param}, M, \mathsf{sid}, \mathbf{pk}, \mathbf{BL}, \sigma, \tau)$: This deterministic algorithm, given $\mathsf{param}, M, \mathsf{sid}, \mathbf{pk}, \mathbf{BL}$, σ, and τ, outputs the validity 1 if accepting the signature, or 0 otherwise.
- $\mathsf{AddBL}(\mathsf{param}, \tau, \mathbf{BL})$: This deterministic algorithm, given param, τ, and $\mathbf{BL}$, outputs new blacklist $\mathbf{BL}'$.

We use notation $\mathbf{pk}$ as a vector of public keys $(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$, and $\mathbf{BL}$ as a vector of tickets $(\tau_1, \ldots, \tau_\ell)$.

3.2 Security Requirements

We define the following security requirements, which are derived from the requirements in the ring signature scheme *DualRing* [17], which are shown in Sect. 2.3.

3.2.1 Correctness

The perfect correctness defined in Sect. 2.3 is extended to the model of the blacklistable ring signature scheme, as follows, where the blacklisting mechanism is added to the conventional ring signature scheme. In the extension, to specify tickets of the target honest user, HSign oracle is used by adversary $\mathcal{A}$ instead of giving $\mathcal{A}$ the secret key sk of the user.

Definition 8 (Perfect Correctness)**.** *A blacklistable ring signature scheme is* perfectly correct *if for any PPT adversary $\mathcal{A}$,*

$$
\begin{aligned}
&\Pr[\mathsf{param} \leftarrow \mathsf{Setup}(\lambda);\\
&\quad (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{param});\\
&\quad (M, \mathsf{sid}, \mathbf{pk}, \mathbf{BL}) \leftarrow \mathcal{A}^{\mathsf{HSign}}(\mathsf{param}, \mathsf{pk});\\
&\quad (\sigma, \tau) \leftarrow \mathsf{Sign}(\mathsf{param}, M, \mathsf{sid}, \mathbf{pk}, \mathsf{sk}, \mathbf{BL}) :\\
&\quad \text{If } \mathsf{sid} \notin \mathrm{Q}_{\mathsf{sid}}, \mathsf{pk} \in \mathbf{pk},\\
&\qquad \text{and for all } \tau \in Q_{\mathsf{Htickets}}, \tau \notin \mathbf{BL},\\
&\quad \text{then } \mathsf{Verify}(\mathsf{param}, M, \mathsf{sid}, \mathbf{pk}, \mathbf{BL}, \sigma, \tau) = 1] = 1,
\end{aligned}
$$

where the following oracle is queried by $\mathcal{A}$.

- HSign*: This oracle is queried on* $(M_k, \mathsf{sid}_k, \mathbf{pk}_k, \mathbf{BL}_k)$*. If* $\mathsf{sid}_k \in Q_{\mathsf{sid}}$, $\mathsf{pk} \notin \mathbf{pk}_k$*, or for* $\tau \in Q_{\mathsf{Htickets}}$, $\tau \in \mathbf{BL}_k$*, then abort. Run* $\mathsf{Sign}(\mathsf{param}, M_k, \mathsf{sid}_k, \mathbf{pk}_k, \mathsf{sk}, \mathbf{BL}_k)$ *to obtain* (σ_k, τ_k)*, which is returned to* $\mathcal{A}$*. Add* sid_k *to* Q_{sid}*, and* τ_k *to* Q_{Htickets}.

3.2.2 Unforgeability

Definition 9 (Unforgeability)**.** *A blacklistable ring signature scheme is* unforgeable *if for any PPT adversary $\mathcal{A}$ which can access oracles O and some integer* $\mathsf{num}_{\mathsf{key}}$ *polynomial in λ,*

$$
\begin{aligned}
&\Pr[\mathsf{cnt}_{\mathbf{BL}} := 0; \mathsf{param} \leftarrow \mathsf{Setup}(\lambda);\\
&\quad (\hat{\mathsf{pk}}_i, \hat{\mathsf{sk}}_i) \leftarrow \mathsf{KeyGen}(\mathsf{param}) \ for\ all\ i \in [1, \mathsf{num}_{\mathsf{key}}];\\
&\quad Q_{\mathsf{KeyGen}} := \{\hat{\mathsf{pk}}_i\}_{i=1}^{\mathsf{num}_{\mathsf{key}}};\\
&\quad (M^*, \mathsf{sid}^*, \mathbf{pk}^*, j^*, \sigma^*, \tau^*) \leftarrow \mathcal{A}^{O}(\mathsf{param}, Q_{\mathsf{KeyGen}}) :\\
&\quad \mathsf{Verify}(\mathsf{param}, M^*, \mathsf{sid}^*, \mathbf{pk}^*, \mathbf{BL}_{j^*}, \sigma^*, \tau^*) = 1\\
&\quad \wedge \mathsf{sid}^* \notin Q_{\mathsf{sid}} \wedge j^* \in [1, \mathsf{cnt}_{\mathbf{BL}}]\\
&\quad \wedge (\mathbf{pk}^* \subseteq Q_{\mathsf{KeyGen}} \setminus Q_{\mathsf{reveal}}\\
&\qquad \vee \mathsf{CntBListedCUser}_{j^*} \geq |Q_{\mathsf{Reveal}}|)]
\end{aligned}
$$

is negligible for λ, where the following oracles are used as O queried by $\mathcal{A}$. The history of oracle queries is kept.

- Reveal: *This oracle is queried on* $\mathsf{pk}_k \in Q_{\mathsf{KeyGen}}$*, returns the corresponding secret key* sk_k*, and adds* pk_k *to* Q_{Reveal}*. When the secret key is revealed, the corresponding user is controlled by* $\mathcal{A}$*.*

- HSign: *In this oracle, a signature and the ticket of issued by a honest (not adversarially controlled) user are requested. This oracle is queried on* $(M_k, \mathsf{sid}_k, \mathbf{pk}_k, \mathsf{pk}_k, j_k)$*. If* $\mathsf{pk}_k \notin Q_{\mathsf{KeyGen}} \setminus Q_{\mathsf{Reveal}}$*,* $\mathsf{sid}_k \in Q_{\mathsf{sid}}$*,* pk_k *is not an element in* $\mathbf{pk}_k$*,* $j_k \notin [1, \mathsf{cnt}_{\mathbf{BL}}]$*, or for* $(\mathsf{pk}_k, \mathsf{tickets}_k)$ *in* $Q_{\mathsf{UserTickets}}$*, some element in* $\mathsf{tickets}_k$ *is in* $\mathbf{BL}_{j_k}$*, then abort. For secret key* sk_k *corresponding to* pk_k*, run* $\mathsf{Sign}(\mathsf{param}, M_k, \mathsf{sid}_k, \mathbf{pk}_k, \mathsf{sk}_k, \mathbf{BL}_{j_k})$ *to obtain* (σ_k, τ_k)*, which is returned to* $\mathcal{A}$*. Update the element* $(\mathsf{pk}_k, \mathsf{tickets}_k)$ *in* $Q_{\mathsf{UserTickets}}$ *to* $(\mathsf{pk}_k, \mathsf{tickets}_k \cup \{\tau_k\})$*. If there is not the element in* $Q_{\mathsf{UserTickets}}$*, add new element* $(\mathsf{pk}_k, \mathsf{tickets}_k)$ *in* $Q_{\mathsf{UserTickets}}$*, where* $\mathsf{tickets}_k = \{\tau_k\}$*. Add* sid_k *to* Q_{sid}*, and* τ_k *to* Q_{Htickets}*.*

- CSign: *In this oracle, a ticket of a signature issued by a corrupted (adversarially controlled) user is outputted as the oracle query. This oracle is queried on* $(M_k, \mathsf{sid}_k, \mathbf{pk}_k, j_k, \sigma_k, \tau_k)$*, where the queried values* $(M_k, \mathsf{sid}_k, \mathbf{pk}_k, j_k, \sigma_k, \tau_k)$ *are sent from* $\mathcal{A}$ *to this oracle which keeps the needed values for the following* CAddBL *oracle, as follows. If* $\mathsf{Verify}(\mathsf{param}, M_k, \mathsf{sid}_k, \mathbf{pk}_k, \mathbf{BL}_{j_k}, \sigma_k, \tau_k) = 0$*,* $\mathsf{sid}_k \in Q_{\mathsf{sid}}$*,* $\mathbf{pk}_k \setminus Q_{\mathsf{KeyGen}} \neq \emptyset$*, or* $j_k \notin [1, \mathsf{cnt}_{\mathbf{BL}}]$*, abort. Add* sid_k *to* Q_{sid}*, and* τ_k *to* Q_{Ctickets}*. In this oracle, nothing is returned to* $\mathcal{A}$*.*

- NewBL : *For this oracle, increment counter* $\mathsf{cnt}_{\mathbf{BL}}$*, and initializes* $\mathbf{BL}_{\mathsf{cnt}_{\mathbf{BL}}}$ *as empty and* $\mathsf{CntBListedCUser}_{\mathsf{cnt}_{\mathbf{BL}}} = 0$*.*

- HAddBL: *In this oracle, a ticket of a honest user is added to a blacklist. This oracle is queried on* (τ_k, j_k)*. If* $\tau_k \notin Q_{\mathsf{Htickets}}$ *or* $j_k \notin [1, \mathsf{cnt}_{\mathbf{BL}}]$*, abort. Otherwise, add* τ_k *to* $\mathbf{BL}_{j_k}$*, using* $\mathsf{AddBL}(\tau_k, \mathbf{BL}_{j_k})$*.*

- CAddBL: *In this oracle, a ticket of a corrupted user is added to a blacklist. This oracle is queried on* (τ_k, j_k)*. If* $\tau_k \notin Q_{\mathsf{Ctickets}}$ *or* $j_k \notin [1, \mathsf{cnt}_{\mathbf{BL}}]$*, abort. Otherwise, add* τ_k *to* $\mathbf{BL}_{j_k}$*, using* $\mathsf{AddBL}(\tau_k, \mathbf{BL}_{j_k})$*. Check the past* CAddBL *and* CSign *queries for* j_k*. If* τ_k *is produced by* CSign *query after the last* CAddBL *query, increment* $\mathsf{CntBListedCUser}_{j_k}$*. As shown in the following paragraph, the variable* $\mathsf{CntBListedCUser}_{j_k}$ *means that the number of corrupted users blacklisted in blacklist* $\mathbf{BL}_{j_k}$*. Thus, the condition* $\mathsf{CntBListedCUser}_{j^*} \geq |Q_{\mathsf{Reveal}}|$ *in the unforgeability game implies that the number of corrupted users blacklisted in blacklist* $\mathbf{BL}_{j^*}$ *is not less than the number of the total number of corrupted users, i.e., all corrupted users are blacklisted.*

To the original definition of unforgeability, we add the functions of blacklistable authentication [15]. In addition to HSign that is the oracle to return signatures issued by honest (not adversarially controlled) signers, CSign oracle is added, where tickets of the signatures issued by corrupted (adversarially controlled) signers are outputted. Note that in this oracle, adversary $\mathcal{A}$ is returned nothing, but instead the queried values are checked and the ticket τ_k is stored in Q_{Ctickets} for CAddBL. In this definition, multiple blacklists are available. By NewBL oracle, a new blacklists $\mathbf{BL}_j$ with $j = \mathsf{cnt}_{\mathbf{BL}}$ is initialized for counter $\mathsf{cnt}_{\mathbf{BL}}$ that shows the number of currently used blacklists. In this oracle, counter $\mathsf{CntBListedCUser}_j$ is also initialized. $\mathsf{CntBListedCUser}_j$ shows the number of the currently blacklisted and corrupted users. By HAddBL oracle, a ticket τ_k of a signature issued by an honest signer is added to blacklist $\mathbf{BL}_{j_k}$. By CAddBL oracle, a ticket τ_k of a signature issued by a corrupted signer is added to blacklist $\mathbf{BL}_{j_k}$. In this oracle, $\mathsf{CntBListedCUser}_j$ is incremented if τ_k is produced by a corrupted user after the last addition of a ticket issued by a corrupted user to the same blacklist. In this case, the anonymous user producing τ_k is not blacklisted in the previous $\mathbf{BL}_{j_k}$, but after this oracle, the user is blacklisted, i.e., the number of the currently blacklisted and corrupted users is incremented. Thus, the condition $\mathsf{CntBListedCUser}_{j^*} \geq |Q_{\mathsf{Reveal}}|$ in the unforgeability game means that all corrupted users are blacklisted in blacklist $\mathbf{BL}_{j^*}$.

3.2.3 Anonymity

The following anonymity definition is weaker than that of DualRing, which is shown in Definition 5. The original definition is the strong anonymity model against the full key exposure in [2], where an adversary is given all randomness to the secret keys of honest users. However, in the proposed blacklistable scheme, similarly to BLAC, the ticket can be computed from the secret key without any random due to the blacklisting mechanism, and thus the strong adversary with all randomness to the secret keys can identify the signer. This is why we weaken the anonymity definition to the basic anonymity[†]. Furthermore, the same oracles for blacklist as in the unforgeability definition are added.

Definition 10 (Anonymity)**.** *A blacklistable ring signature scheme is* anonymous *if for any PPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *which can access oracles* O *and some integer* $\mathsf{num}_{\mathsf{key}}$ *polynomial in* λ*,*

$$
\begin{aligned}
&|\Pr[\mathsf{cnt}_{\mathbf{BL}} := 0; \mathsf{param} \leftarrow \mathsf{Setup}(\lambda);\\
&\quad (\hat{\mathsf{pk}}_i, \hat{\mathsf{sk}}_i) \leftarrow \mathsf{Keygen}(\mathsf{param})\ for\ all\ i \in [1, \mathsf{num}_{\mathsf{key}}];\\
&\quad Q_{\mathsf{KeyGen}} := \{\hat{\mathsf{pk}}_i\}_{i=1}^{\mathsf{num}_{\mathsf{key}}};\\
&\quad (M^*, \mathsf{sid}^*, \mathbf{pk}^*, j^*, i_0, i_1, St) \leftarrow \mathcal{A}_1^{O}(\mathsf{param}, Q_{\mathsf{KeyGen}});\\
&\quad b \leftarrow \{0, 1\};\\
&\quad (\sigma^*, \tau^*) \leftarrow \mathsf{Sign}(\mathsf{param}, M^*, \mathsf{sid}^*, \mathbf{pk}^*, \hat{\mathsf{sk}}_{i_b}, \mathbf{BL}_{j^*});
\end{aligned}
$$

[†]In our preliminary version [12] of this work, we were not aware of this, and adopted the strong anonymity definition. In this journal version, it is corrected, and the basic anonymity of the proposed scheme is formally proved.

$$b' \leftarrow \mathcal{A}_2(\sigma^*, \tau^*, St) :$$
$$b' = b \wedge \hat{\mathsf{pk}}_{i_0}, \hat{\mathsf{pk}}_{i_1} \in (Q_{\mathsf{KeyGen}} \setminus Q_{\mathsf{Reveal}}) \cap \mathbf{pk}^*$$
$$\wedge \mathsf{tickets}_{i_b} \cap \mathbf{BL}_{j^*} = \emptyset$$
$$for\ (\mathsf{pk}_{i_b}, \mathsf{tickets}_{i_b}) \in Q_{\mathsf{UserTickets}}\ with\ b \in \{0,1\}]$$
$$-1/2|$$

is negligible for λ, where each oracle as O is the same as in the unforgeability.

4. Proposed Short Blacklistable Ring Signature Scheme

4.1 Construction Idea

We construct a blacklistable ring signature scheme, where *DualRing-EC* is combined with the blacklistable anonymous credential system BLAC [15].

In BLAC, a central IdP (Identity Provider) issues a credential to each user, where the credential is a digital signature on user's secret with the IdP's public key. Let $\mathsf{Sig}(\mathsf{sk})$ be the signature on user's secret sk. In each authentication between the user and a verifier with a session ID sid, the user computes a ticket $t = H(s, \mathsf{sid})^{\mathsf{sk}}$ for $s \xleftarrow{R} \mathbb{Z}_p$ and a signature of knowledge to prove the knowledge of (S, sk) s.t. $S = \mathsf{Sig}(\mathsf{sk})$ and $t = H(s, \mathsf{sid})^{\mathsf{sk}}$. In case that the user's activity based on the authentication is judged as misbehavior, the ticket $\tau = (\mathsf{sid}, s, t)$ is added to a blacklist $\mathbf{BL}$. In fact, in each authentication, the user also proves that the user's secret key sk certified by $\mathsf{Sig}(\mathsf{sk})$ is not used to compute each ticket $\tau_i = (\mathsf{sid}_i, s_i, t_i)$ with session ID's sid_i in $\mathbf{BL} = (\tau_1, \ldots, \tau_\ell)$, using a signature of knowledge to prove $t_i \neq H(s_i, \mathsf{sid}_i)^{\mathsf{sk}}$ for all $i \in [1, \ell]$.

In the proposed blacklistable ring signature scheme, the signature of knowledge to prove $\mathsf{Sig}(\mathsf{sk})$ is replaced by $\mathsf{DualRing\text{-}EC.Sign}$. In the ring signature, the signer proves that the signer's public key $\mathsf{pk} = g^{\mathsf{sk}}$ is included in a ring of public keys $\{\mathsf{pk}_1, \ldots, \mathsf{pk}_N\}$. The *DualRing-EC* achieves the $O(\log N)$-size of signature. In the combination of the blacklistable authentication and *DualRing-EC*, we need to verify that the same sk is used in the blacklistable authentication part and $\mathsf{DualRing\text{-}EC.Sign}$. But, due to the anonymity of the ring signature, the verifier cannot check which pk_j (resp., sk_j s.t. $\mathsf{pk}_j = g^{\mathsf{sk}_j}$) is used as the signer's pk (resp., sk) in $\mathsf{DualRing\text{-}EC.Sign}$. Thus, in our proposed scheme, the commitment $C = g^{\mathsf{sk}} h^\rho$ is used. For all pk_i in the ring, $\mathsf{pk}'_i = C/\mathsf{pk}_i$ is computed, where $\mathsf{pk}'_j = g^{\mathsf{sk}_j} h^\rho / \mathsf{pk}_j = h^\rho$ for j, and $\mathsf{pk}'_i = g^{\mathsf{sk}_j} h^\rho / \mathsf{pk}_i = g^{\mathsf{sk}_j - \mathsf{sk}_i} h^\rho$ for all other i with $i \neq j$, where $\mathsf{pk}'_i = h^{(\mathsf{sk}_j - \mathsf{sk}_i)\theta + \rho}$ for θ s.t. $g = h^\theta$. Thus, for $\{\mathsf{pk}'_1, \ldots, \mathsf{pk}'_N\}$ instead of $\{\mathsf{pk}_1, \ldots, \mathsf{pk}_N\}$, h instead of g, and $\mathsf{sk}'_j = \rho$ instead of sk_j, we can use $\mathsf{DualRing\text{-}EC.Sign}$ to show that the singer knows the secret key sk'_j of a public key pk'_j in $\{\mathsf{pk}'_1, \ldots, \mathsf{pk}'_N\}$. Furthermore, for sk bound by ρ s.t. $C = g^{\mathsf{sk}} h^\rho$, we can use the signature of knowledge to prove $t = H(s, \mathsf{sid})^{\mathsf{sk}}$ and $t_i \neq H(s_i, \mathsf{sid}_i)^{\mathsf{sk}}$ for all $i \in [1, \ell]$. This is why we can achieve the blacklistable authentication in *DualRing-EC*.

4.2 Proposed Algorithms

$\mathsf{Setup}(\lambda)$:

1. Generate $(\mathbb{G}, p, g) \leftarrow \mathcal{G}(\lambda)$. Select $u, h \xleftarrow{R} \mathbb{G}$.
2. Output $\mathsf{param} = (\mathbb{G}, p, g, u, h)$.

$\mathsf{KeyGen}(\mathsf{param})$:

1. Select $\mathsf{sk} \xleftarrow{R} \mathbb{Z}_p$, and compute $\mathsf{pk} = g^{\mathsf{sk}}$.
2. Output $(\mathsf{pk}, \mathsf{sk})$.

$\mathsf{Sign}(\mathsf{param}, M, \mathsf{sid}, \mathbf{pk}, \mathsf{sk}, \mathbf{BL})$:

1. Parse $\mathbf{pk} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_N)$, $\mathbf{BL} = (\tau_1, \ldots, \tau_\ell)$, and $\tau_i = (\mathsf{sid}_i, s_i, t_i)$ for $i \in [1, \ell]$. Let the corresponding public key pk of sk be pk_j for $j \in [1, N]$.
2. Select $\rho \xleftarrow{R} \mathbb{Z}_p$, and compute $C = \mathsf{pk}_j \cdot h^\rho$ which is $g^{\mathsf{sk}_j} h^\rho$. For all $i \in [1, N]$, compute $\mathsf{pk}'_i = C/\mathsf{pk}_i$. Then, $\mathsf{pk}'_j = g^{\mathsf{sk}_j} h^\rho / \mathsf{pk}_j = h^\rho$ for j, and $\mathsf{pk}'_i = g^{\mathsf{sk}_j} h^\rho / \mathsf{pk}_i = g^{\mathsf{sk}_j - \mathsf{sk}_i} h^\rho$ for all other i with $i \neq j$, where $\mathsf{pk}'_i = h^{(\mathsf{sk}_j - \mathsf{sk}_i)\theta + \rho}$ for θ s.t. $g = h^\theta$.
3. For $\mathbf{pk}' = (\mathsf{pk}'_1, \ldots, \mathsf{pk}'_N)$ instead of $\mathbf{pk}$, h instead of g, and ρ instead of sk_j, conduct $\mathsf{DualRing\text{-}EC.Sign}$, as follows.
   a. Select $r, c_i \xleftarrow{R} \mathbb{Z}_p$ for all $i \neq j$. Compute $R = h^r \cdot \prod_{i \neq j} \mathsf{pk}'^{c_i}_i$, $c = H(M, \mathbf{pk}', R)$, $c_j = c - \sum_{i \neq j} c_i$, and $z = r - c_j \cdot \rho$.
   b. Set $\boldsymbol{a} = (c_1, \ldots, c_N)$. Compute $P = R \cdot h^{-z}$. Then, for $\mathsf{param}' = (\mathbb{G}, p, h)$ and $\mathbf{pk}'$, conduct $\mathsf{NISA.Proof}(\mathsf{param}', \mathbf{pk}', u, P, c, \boldsymbol{a})$ to obtain π, as the proof of $P = \prod_{i=1}^N \mathsf{pk}'^{c_i}_i$ and $c = \sum_{i=1}^N c_i$.
   c. The output of $\mathsf{DualRing\text{-}EC.Sign}$ is (z, R, π).
4. For $i \in [1, \ell]$, compute $B_i = H(s_i, \mathsf{sid}_i)$. Select $s \xleftarrow{R} \mathbb{Z}_p$, and compute $B = H(s, \mathsf{sid})$ and $t = B^{\mathsf{sk}_j}$.

   Generate a signature of knowledge for (sk_j, ρ) s.t.

$$C = g^{\mathsf{sk}_j} h^\rho \wedge t = B^{\mathsf{sk}_j} \wedge \Big(\bigwedge_{i \in [1,\ell]} t_i \neq B_i^{\mathsf{sk}_j} \Big),$$

   as follows.
   a. For $i \in [1, \ell]$, select $\rho_i \xleftarrow{R} \mathbb{Z}_p$, compute $\tilde{A}_i = (B_i^{\mathsf{sk}_j} / t_i)^{\rho_i}$, and $\mu_i = \rho_i \cdot \mathsf{sk}_j$, where $1 = B^{\mu_i} t^{-\rho_i}$ and $\tilde{A}_i = B_i^{\mu_i} t_i^{-\rho_i}$ hold.
   b. Select $r_{\mathsf{sk}_j}, r_\rho, r_{\rho_1}, \ldots, r_{\rho_\ell}, r_{\mu_1}, \ldots, r_{\mu_\ell} \xleftarrow{R} \mathbb{Z}_p$, and compute

$$R_1 = g^{r_{\mathsf{sk}_j}} h^{r_\rho}, \quad R_2 = B^{r_{\mathsf{sk}_j}},$$
$$R_{3,1} = B^{r_{\mu_1}} t^{-r_{\rho_1}}, \ldots, R_{3,\ell} = B^{r_{\mu_\ell}} t^{-r_{\rho_\ell}},$$
$$R_{4,1} = B_1^{r_{\mu_1}} t_1^{-r_{\rho_1}}, \ldots, R_{4,\ell} = B_\ell^{r_{\mu_\ell}} t_\ell^{-r_{\rho_\ell}}.$$

c. Compute $\tilde{c} = H(B, t, \{B_i, t_i, \tilde{A}_i\}_{i\in[1,\ell]}, R_1, R_2, \{R_{3,i}, R_{4,i}\}_{i\in[1,\ell]}, M)$, and

$$s_{\mathsf{sk}_j} = r_{\mathsf{sk}_j} - \tilde{c}\cdot \mathsf{sk}_j, \quad s_\rho = r_\rho - \tilde{c}\cdot\rho,$$
$$s_{\rho_1} = r_{\rho_1} - \tilde{c}\cdot\rho_1, \ldots, s_{\rho_\ell} = r_{\rho_\ell} - \tilde{c}\cdot\rho_\ell,$$
$$s_{\mu_1} = r_{\mu_1} - \tilde{c}\cdot\mu_1, \ldots, s_{\mu_\ell} = r_{\mu_\ell} - \tilde{c}\cdot\mu_\ell.$$

d. The output is $\tilde{\pi} = (\tilde{A}_1, \ldots, \tilde{A}_\ell, \tilde{c}, s_{\mathsf{sk}_j}, s_\rho, s_{\rho_1}, \ldots, s_{\rho_\ell}, s_{\mu_1}, \ldots, s_{\mu_\ell})$.

5. Output $\sigma = (C, z, R, \pi, \tilde{\pi})$ and $\tau = (\mathsf{sid}, s, t)$.

Verify($\mathsf{param}, M, \mathsf{sid}, \mathbf{pk}, \mathbf{BL}, \sigma, \tau$):

1. Parse $\mathbf{pk} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_N)$, $\mathbf{BL} = (\tau_1, \ldots, \tau_\ell)$, $\tau_i = (\mathsf{sid}_i, s_i, t_i)$ for $i \in [1,\ell]$, $\sigma = (C, z, R, \pi, \tilde{\pi})$, and $\tau = (\mathsf{sid}, s, t)$.
2. For all $i \in [1, N]$, compute $\mathsf{pk}'_i = C/\mathsf{pk}_i$.
3. For $\mathbf{pk}' = (\mathsf{pk}'_1, \ldots, \mathsf{pk}'_N)$ instead $\mathbf{pk}$, h instead of g, conduct DualRing-EC.Verify, as follows.
   a. Compute $c = H(M, \mathbf{pk}', R)$ and $P = R \cdot h^{-z}$.
   b. For $\mathsf{param}' = (\mathbb{G}, p, h)$ and $\mathbf{pk}'$, conduct NISA.Verify($\mathsf{param}', \mathbf{pk}', u, P, c, \pi$). If the output of NISA.Verify is 0, output 0.
4. Verify the signature of knowledge $\tilde{\pi}$ as follows. If $\tilde{\pi}$ is valid, output 1, and otherwise output 0.
   a. Parse $\tilde{\pi} = (\tilde{A}_1, \ldots, \tilde{A}_\ell, \tilde{c}, s_{\mathsf{sk}_j}, s_\rho, s_{\rho_1}, \ldots, s_{\rho_\ell}, s_{\mu_1}, \ldots, s_{\mu_\ell})$.
   b. Compute

$$R'_1 = g^{s_{\mathsf{sk}_j}} h^{s_\rho} C^{\tilde{c}}, \quad R'_2 = B^{s_{\mathsf{sk}_j}} t^{\tilde{c}},$$
$$R'_{3,1} = B^{s_{\mu_1}} t^{-s_{\rho_1}}, \ldots, R'_{3,\ell} = B^{s_{\mu_\ell}} t^{-s_{\rho_\ell}},$$
$$R'_{4,1} = B_1^{s_{\mu_1}} t_1^{-s_{\rho_1}} \tilde{A}_1^{\tilde{c}}, \ldots, R'_{4,\ell} = B_\ell^{s_{\mu_\ell}} t_\ell^{-s_{\rho_\ell}} \tilde{A}_\ell^{\tilde{c}},$$

   where $B = H(s, \mathsf{sid})$ and $B_i = H(s_i, \mathsf{sid}_i)$ for all $i \in [1,\ell]$.
   c. $\tilde{\pi}$ is valid, if and only if $\tilde{c} = H(B, t, \{B_i, t_i, \tilde{A}_i\}_{i\in[1,\ell]}, R'_1, R'_2, \{R'_{3,i}, R'_{4,i}\}_{i\in[1,\ell]}, M)$ and $\tilde{A}_i$ is not the identity element $1_\mathbb{G}$ of $\mathbb{G}$ for all $i \in [1,\ell]$.

AddBL($\mathsf{param}, \tau, \mathbf{BL}$):

1. Parse $\mathbf{BL} = (\tau_1, \ldots, \tau_\ell)$, and output $\mathbf{BL}' = (\tau_1, \ldots, \tau_\ell, \tau)$.

5. Security

In this section, we show the security of the proposed scheme.

Theorem 1. *The proposed blacklistable ring signature scheme is perfectly correct.*

Proof. For correctly computed param and $(\mathsf{sk}, \mathsf{pk})$, and $(M, \mathsf{sid}, \mathbf{pk} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_N), \mathbf{BL} = (\tau_1, \ldots, \tau_\ell))$ given by $\mathcal{A}$, consider C, pk'_i that are correctly computed in Sign. As the experiment with the adversary $\mathcal{A}$ in the definition of the perfect correctness, assume that $\mathsf{pk} \in \mathbf{pk}$ and for all $\tau \in Q_{\mathsf{Htickets}}$, $\tau \notin \mathbf{BL}$. Let $\mathsf{pk} = \mathsf{pk}_j$, and thus $\mathsf{sk} = \mathsf{sk}_j$. Then, $\mathsf{pk}'_j = g^{\mathsf{sk}_j} h^\rho / \mathsf{pk}_j = h^\rho$ for j, and $\mathsf{pk}'_i = g^{\mathsf{sk}_j} h^\rho / \mathsf{pk}_i = g^{\mathsf{sk}_j - \mathsf{sk}_i} h^\rho$ for all other i with $i \neq j$, where $\mathsf{pk}'_i = h^{(\mathsf{sk}_j - \mathsf{sk}_i)\theta + \rho}$ for θ s.t. $g = h^\theta$. Thus, due to $\mathsf{pk}'_j \in \mathbf{pk}' = (\mathsf{pk}'_1, \ldots, \mathsf{pk}'_N)$, the values (z, R, π) outputted by DualRing-EC.Sign for $\mathbf{pk}'$ instead of $\mathbf{pk}$, h instead of g, and ρ instead of sk_j are accepted by DualRing-EC.Verify with probability 1. Next, consider B_i for $i \in [1,\ell]$, and $B, t, \tilde{\pi}$ that are correctly computed in Sign. The condition that for all $\tau \in Q_{\mathsf{Htickets}}$, $\tau \notin \mathbf{BL}$ implies that all $\tau_i = (\mathsf{sid}_i, s_i, t_i)$ is not computed from $\mathsf{sk} = \mathsf{sk}_j$. Then, it holds that $\bigwedge_{i\in[1,\ell]} t_i \neq B_i^{\mathsf{sk}_j}$, where $B_i = H(s_i, \mathsf{sid}_i)$. Thus, due to the completeness of the signature of knowledge, $\tilde{\pi}$ is accepted with probability 1. □

Theorem 2. *The proposed blacklistable ring signature scheme is unforgeable under the DL assumption in the random oracle model.*

Proof. In the definition of the unforgeability, the experiment with the adversary $\mathcal{A}$ is called unforgeability game, and $\mathcal{A}$ satisfying the condition $\mathsf{Verify}(\mathsf{param}, M^*, \mathsf{sid}^*, \mathbf{pk}^*, \mathbf{BL}_{j^*}, \sigma^*, \tau^*) = 1 \wedge \mathsf{sid}^* \notin Q_{\mathsf{sid}} \wedge j^* \in [1, \mathsf{cnt}_{\mathbf{BL}}] \wedge (\mathbf{pk}^* \subseteq Q_{\mathsf{KeyGen}} \setminus Q_{\mathsf{reveal}} \vee \mathsf{CntBListedCUser}_{j^*} \geq |Q_{\mathsf{Reveal}}|)$ is called winning adversary.

Assume there is a winning adversary $\mathcal{A}$ in the unforgeability game for the proposed scheme. We will construct adversaries to break the DL assumption. In the unforgeability game, $\mathcal{A}$ finally outputs a forged signature $\sigma^* = (C^*, z^*, R^*, \pi^*, \tilde{\pi}^*)$ on $M^*, \mathsf{sid}^*, \mathbf{pk}^*, \mathbf{BL}_{j^*}$ s.t. $\mathsf{Verify}(\mathsf{param}, M^*, \mathsf{sid}^*, \mathbf{pk}^*, \mathbf{BL}_{j^*}, \sigma^*, \tau^*) = 1$. Let $\mathbf{pk}^* = (\hat{\mathsf{pk}}_{i_1}, \ldots, \hat{\mathsf{pk}}_{i_N})$, where $\hat{\mathsf{pk}}_{i_u} = g^{\hat{\mathsf{sk}}_{i_u}}$. The winning adversary $\mathcal{A}$ satisfies $\mathbf{pk}^* \subseteq Q_{\mathsf{KeyGen}} \setminus Q_{\mathsf{reveal}}$ or $\mathsf{CntBListedCUser}_{j^*} \geq |Q_{\mathsf{Reveal}}|$. $\mathcal{A}$ also outputs a signature $\sigma_k = (C_k, z_k, R_k, \pi_k, \tilde{\pi}_k)$ on $M_k, \mathsf{sid}_k, \mathbf{pk}_k, \mathbf{BL}_{j_k}$ in each CSign query. Similarly, we denote $\mathbf{pk}_j = (\hat{\mathsf{pk}}_{i_1}, \ldots, \hat{\mathsf{pk}}_{i_N})$. As shown in the following reductions, we can extract the committed value $\tilde{\mathsf{sk}}^*$ in the commitment $C^* = g^{\tilde{\mathsf{sk}}^*} h^{\rho^*}$ from σ^*. Also, similarly, we can extract the committed value $\tilde{\mathsf{sk}}_k$ in the commitment $C_k = g^{\tilde{\mathsf{sk}}_k} h^{\rho_k}$ from σ_k. Then, we distinguish three cases of $\mathcal{A}$.

[Case 1: $\tilde{\mathsf{sk}}^* \neq \hat{\mathsf{sk}}_{i_u}$ (or $\tilde{\mathsf{sk}}_k \neq \hat{\mathsf{sk}}_{i_u}$) for all $u \in [1, N]$]: In this case, we show an adversary $\mathcal{B}_{\mathrm{DL}}$ as follows. Here, we describe only the case of $\tilde{\mathsf{sk}}^* \neq \hat{\mathsf{sk}}_{i_u}$ in the final signature σ^*, but the case of $\tilde{\mathsf{sk}}_k \neq \hat{\mathsf{sk}}_{i_u}$ in a CSign query is similar.

Given $(\mathbb{G}, p, g) \leftarrow \mathcal{G}(\lambda)$ and h s.t. the discrete log of h to base g is unknown, run $(\hat{\mathsf{pk}}_i, \hat{\mathsf{sk}}_i) \leftarrow \mathsf{KeyGen}(\mathsf{param})$ for all $i \in [1, \mathsf{num}_{\mathsf{key}}]$. Generate the other parameters in param as

in Setup, and run $\mathcal{A}$ with param and $Q_{\mathsf{KeyGen}} := \{\hat{\mathsf{pk}}_i\}_{i=1}^{\mathsf{num_{key}}}$. In the oracles, simulate H as a random oracle. For Reveal oracle, return the corresponding $\hat{\mathsf{sk}}_i$. For HSign oracle, return the signature and ticket using the corresponding sk_k. CSign, NewBL, HAddBL, CAddBL queries are conducted as in the descriptions of the oracles. $\mathcal{A}$ finally outputs the signature $\sigma^* = (C^*, z^*, R^*, \pi^*, \tilde{\pi}^*)$ on $M^*, \mathsf{sid}^*, \mathbf{pk}^*, \mathbf{BL}_{j^*}$ s.t. $\mathsf{Verify}(\mathsf{param}, M^*, \mathsf{sid}^*, \mathbf{pk}^*, \mathbf{BL}_{j^*}, \sigma^*, \tau^*) = 1$. Parse $\sigma^* = (C^*, z^*, R^*, \pi^*, \tilde{\pi}^*)$, where $\pi^* = (\boldsymbol{L}^*, \boldsymbol{R}^*, a^*, b^*)$, $\tilde{\pi}^* = (\tilde{A}_1^*, \ldots, \tilde{A}_\ell^*, \tilde{c}^*, s_{\mathsf{sk}_j}^*, s_\rho^*, s_{\rho_1}^*, \ldots, s_{\rho_\ell}^*, s_{\mu_1}^*, \ldots, s_{\mu_\ell}^*)$, and $\tau^* = (\mathsf{sid}^*, s^*, t^*)$ with $B^* = H(s^*, \mathsf{sid}^*)$. Rewind the point that $H(B^*, t^*, \{B_i^*, t_i^*, \tilde{A}_i^*\}_{i\in[1,\ell]}, R_1^*, R_2^*, \{R_{3,i}^*, R_{4,i}^*\}_{i\in[1,\ell]}, M^*)$ is queried, and return a different $\tilde{c}'$ instead. Then, $\mathcal{A}$ outputs $\sigma' = (C^*, z^*, R^*, \pi^*, \tilde{\pi}')$, where $\tilde{\pi}' = (\tilde{A}_1^*, \ldots, \tilde{A}_\ell^*, \tilde{c}', s'_{\mathsf{sk}_j}, s'_\rho, s'_{\rho_1}, \ldots, s'_{\rho_\ell}, s'_{\mu_1}, \ldots, s'_{\mu_\ell})$. Thus, applying the forking lemma [1] to the H, we can successfully extract $(\tilde{\mathsf{sk}}^*, \rho^*)$ s.t. $C^* = g^{\tilde{\mathsf{sk}}^*} h^{\rho^*}$ with some non-negligible probability. We have $\mathsf{pk}'^*_{i_u} = C^*/\hat{\mathsf{pk}}_{i_u} = g^{\tilde{\mathsf{sk}}^* - \hat{\mathsf{sk}}_{i_u}} h^{\rho^*}$ for all $u \in [1, N]$.

Using the statistical witness-extended emulation of NISA for π^*, run an extractor to obtain $(c_1^*, \ldots, c_N^*, z^*)$, where $P^* = R^* h^{-z^*} = \prod_{u=1}^N \mathsf{pk}'^{*\,c_u^*}_{i_u}$, and $c^* = \sum_{u=1}^N c_u^*$ is responded by the random oracle as the query on $H(M^*, \mathbf{pk}'^*, R^*)$. Then, rewind the point that $H(M^*, \mathbf{pk}'^*, R^*)$ is queried, and return a different c'' instead. Then, $\mathcal{A}$ output $\sigma'' = (C^*, z'', R^*, \pi'', \tilde{\pi}'')$, where $\pi'' = (\boldsymbol{L}'', \boldsymbol{R}'', a'', b'')$. Using the statistical witness-extended emulation of NISA for π'', run an extractor to obtain $(c_1'', \ldots, c_N'', z'')$, where $P'' = R^* h^{-z''} = \prod_{u=1}^N \mathsf{pk}'^{*\,c_u''}_{i_u}$, and $c'' = \sum_{u=1}^N c_u''$. Then, we have

$$\begin{aligned} R^* &= h^{z^*} \cdot \prod_{u=1}^N \mathsf{pk}'^{*\,c_u^*}_{i_u} \\ &= g^{\sum_{u=1}^N c_u^* \cdot (\tilde{\mathsf{sk}}^* - \hat{\mathsf{sk}}_{i_u})} h^{z^* + \sum_{u=1}^N c_u^* \cdot \rho^*}. \end{aligned}$$

Similarly, we have $R^* = g^{\sum_{u=1}^N c_u'' \cdot (\tilde{\mathsf{sk}}^* - \hat{\mathsf{sk}}_{i_u})} h^{z'' + \sum_{u=1}^N c_u'' \cdot \rho^*}$. Therefore, we obtain the following equations, where $1_{\mathbb{G}}$ is the identity element of $\mathbb{G}$.

$$\begin{aligned} & g^{\sum_{u=1}^N c_u^* \cdot (\tilde{\mathsf{sk}}^* - \hat{\mathsf{sk}}_{i_u})} h^{z^* + \sum_{u=1}^N c_u^* \cdot \rho^*} \\ &= g^{\sum_{u=1}^N c_u'' \cdot (\tilde{\mathsf{sk}}^* - \hat{\mathsf{sk}}_{i_u})} h^{z'' + \sum_{u=1}^N c_u'' \cdot \rho^*} \\ & g^{\sum_{u=1}^N (c_u^* - c_u'') \cdot (\tilde{\mathsf{sk}}^* - \hat{\mathsf{sk}}_{i_u})} h^{z^* - z'' + \sum_{u=1}^N (c_u^* - c_u'') \cdot \rho^*} = 1_{\mathbb{G}} \end{aligned}$$

Namely, for $\alpha = \sum_{u=1}^N (c_u^* - c_u'') \cdot (\tilde{\mathsf{sk}}^* - \hat{\mathsf{sk}}_{i_u})$ and $\beta = z^* - z'' + \sum_{u=1}^N (c_u^* - c_u'') \cdot \rho^*$, we have $g^\alpha h^\beta = 1_{\mathbb{G}}$.

Due to the random oracle, both $c^* = \sum_{u=1}^N c_u^*$ and $c'' = \sum_{u=1}^N c_u''$ are selected independently and uniformly from $\mathbb{Z}_p$. Because of $c^* \neq c''$, we have uniformly random $c_{u^*}^*$ or c_{u^*}'' s.t. $c_{u^*}^* \neq c_{u^*}''$ for some $u^* \in [1, N]$. Namely, $\Delta c_{u^*} = c_{u^*}^* - c_{u^*}''$ is selected independently and uniformly from $\mathbb{Z}_p$. Here, consider polynomial

$$P(x) = (\tilde{\mathsf{sk}}^* - \hat{\mathsf{sk}}_{i_{u^*}})x + \sum_{u=1, u\neq u^*}^N (c_u^* - c_u'') \cdot (\tilde{\mathsf{sk}}^* - \hat{\mathsf{sk}}_{i_u})$$

In this case, we have $\tilde{\mathsf{sk}}^* \neq \hat{\mathsf{sk}}_{i_{u^*}}$, and thus P is not a zero polynomial. Therefore, by using Schwartz-Zippel lemma, we have $P(\Delta c_{u^*}) \neq 0$, i.e., $\alpha = \sum_{u=1}^N (c_u^* - c_u'') \cdot (\tilde{\mathsf{sk}}^* - \hat{\mathsf{sk}}_{i_u}) \neq 0$ with some non-negligible probability. Thus, from $g^\alpha h^\beta = 1_{\mathbb{G}}$, it holds that $\beta \neq 0$. This means that we have $h = g^{-\alpha/\beta}$. Output $-\alpha/\beta$ as $\mathcal{B}_{\mathrm{DL}}$.

[Case 2: $\tilde{\mathsf{sk}}^* = \hat{\mathsf{sk}}_{i_{u^*}}$ (and $\tilde{\mathsf{sk}}_k = \hat{\mathsf{sk}}_{i_{u^*}}$) for some $u^* \in [1, N]$ and $\hat{\mathsf{pk}}_{i_{u^*}} \notin Q_{\mathsf{reveal}}$ in σ^* (or σ_k)]: In this case, we can construct another adversary $\mathcal{B}'_{\mathrm{DL}}$ to break the DL assumption using $\mathcal{A}$ of this case, as follows. Here, we describe only the case of $\hat{\mathsf{pk}}_{i_{u^*}} \notin Q_{\mathsf{reveal}}$ in the final signature σ^*, but the case in a CSign query is similar.

Given $(\mathbb{G}, p, g) \leftarrow \mathcal{G}(\lambda)$ and $\mathsf{pk}^* = g^{\mathsf{sk}^*}$ for unknown sk^*, pick a random index $i^* \in [1, \mathsf{num_{key}}]$, and run $(\hat{\mathsf{pk}}_i, \hat{\mathsf{sk}}_i) \leftarrow \mathsf{KeyGen}(\mathsf{param})$ for all $i \in [1, \mathsf{num_{key}}]$ except $i = i^*$. Set $\hat{\mathsf{pk}}_{i^*} = \mathsf{pk}^*$. Generate the other parameters in param as in Setup, and run $\mathcal{A}$ with param and $Q_{\mathsf{KeyGen}} := \{\hat{\mathsf{pk}}_i\}_{i=1}^{\mathsf{num_{key}}}$. In the oracles, simulate H as a random oracle. For Reveal oracle, if $\mathsf{pk}_k \neq \hat{\mathsf{pk}}_{i^*}$, return the corresponding sk_k. If $\mathsf{pk}_k = \hat{\mathsf{pk}}_{i^*}$, abort. For HSign oracle, if $\mathsf{pk}_k \neq \hat{\mathsf{pk}}_{i^*}$, return the signature and ticket using the corresponding sk_k. If $\mathsf{pk}_k = \hat{\mathsf{pk}}_{i^*}$, compute the simulated signature $\sigma_k = (C, z, R, \pi, \tilde{\pi})$, as follows. Select $C \xleftarrow{R} \mathbb{G}$, which is indistinguishable from the original commitment due to the hiding property. Pick $c_1, \ldots, c_n, z \xleftarrow{R} \mathbb{Z}_p$, and compute $R = h^z \cdot \prod_{u=1}^N \mathsf{pk}'^{\,c_u}_u$. Set $H(M, \mathbf{pk}', R) = \sum_{u=1}^N c_u$ as the response of the random oracle. If the value has been set in the random oracle, abort (As shown in the original proof [17], the probability of aborting is negligible). Then, using NISA.Proof as in the proposed scheme, obtain π. $\tilde{\pi}$ is simulated as follows. Select $\tilde{A}_1, \ldots, \tilde{A}_\ell \xleftarrow{R} \mathbb{G}$, and $\tilde{c}, s_{\mathsf{sk}_j}, s_\rho, s_{\rho_1}, \ldots, s_{\rho_\ell}, s_{\mu_1}, \ldots, s_{\mu_\ell} \xleftarrow{R} \mathbb{Z}_p$. Compute $R_1 = g^{s_{\mathsf{sk}_j}} h^{s_\rho} C^{\tilde{c}}, R_2 = B^{s_{\mathsf{sk}_j}} t^{\tilde{c}}, R_{3,1} = B^{s_{\mu_1}} t^{-s_{\rho_1}}, \ldots, R_{3,\ell} = B^{s_{\mu_\ell}} t^{-s_{\rho_\ell}}, R_{4,1} = B_1^{s_{\mu_1}} t_1^{-s_{\rho_1}} \tilde{A}_1^{\tilde{c}}, \ldots, R_{4,\ell} = B_\ell^{s_{\mu_\ell}} t_\ell^{-s_{\rho_\ell}} \tilde{A}_\ell^{\tilde{c}}$. Then, set $H(B, t, \{B_i, t_i, \tilde{A}_i\}_{i\in[1,\ell]}, R_1, R_2, \{R_{3,i}, R_{4,i}\}_{i\in[1,\ell]}, M) = \tilde{c}$ as the response of the random oracle. If the value has been set in the random oracle, abort. For $\tau_k = (\mathsf{sid}, s, t)$, select a random $\theta, s \xleftarrow{R} \mathbb{Z}_p$, compute $t = \mathsf{pk}^{*\theta}$, and set $H(s, \mathsf{sid}) = g^\theta$ as the response of the random oracle, where $t = B^{\mathsf{sk}^*}$ holds for $B = H(s, \mathsf{sid})$. If the value for $H(s, \mathsf{sid})$ has been set in the random oracle, abort. CSign, NewBL, HAddBL, CAddBL queries are conducted as in the descriptions of the oracles.

$\mathcal{A}$ finally outputs the signature $\sigma^* = (C^*, z^*, R^*, \pi^*, \tilde{\pi}^*)$ on $M^*, \mathsf{sid}^*, \mathbf{pk}^*, \mathbf{BL}_{j^*}$ s.t. $\mathsf{Verify}(\mathsf{param}, M^*, \mathsf{sid}^*, \mathbf{pk}^*, \mathbf{BL}_{j^*}, \sigma^*, \tau^*) = 1$. If $\mathsf{pk}^* \notin \mathbf{pk}^*$, abort. Otherwise, where this happens with some non-negligible probability as shown in the original paper [17], let $\tilde{u}^*$ be u s.t. $\hat{\mathsf{pk}}_{i_u} = \mathsf{pk}^*$ for $\mathbf{pk}^* = (\hat{\mathsf{pk}}_{i_1}, \ldots, \hat{\mathsf{pk}}_{i_N})$ with $i_u \in [1, \mathsf{num_{key}}]$. Parse $\sigma^* = (C^*, z^*, R^*, \pi^*, \tilde{\pi}^*)$, where $\tilde{\pi}^* = (\tilde{A}_1^*, \ldots, \tilde{A}_\ell^*, \tilde{c}^*, s_{\mathsf{sk}_j}^*, s_\rho^*, s_{\rho_1}^*, \ldots, s_{\rho_\ell}^*, s_{\mu_1}^*, \ldots, s_{\mu_\ell}^*)$, and $\tau^* = (\mathsf{sid}^*, s^*, t^*)$ with $B^* = H(s^*, \mathsf{sid}^*)$. Rewind the point that $H(B^*, t^*, \{B_i^*, t_i^*, \tilde{A}_i^*\}_{i\in[1,\ell]}, R_1^*, R_2^*, \{R_{3,i}^*, R_{4,i}^*\}_{i\in[1,\ell]}, M^*)$ is queried, and return a different $\tilde{c}'$ instead. Then, $\mathcal{A}$ outputs $\sigma' = (C^*, z^*, R^*, \pi^*, \tilde{\pi}')$, where $\tilde{\pi}' =$

$(\tilde{A}_1^*, \ldots, \tilde{A}_\ell^*, \tilde{c}', s'_{\mathsf{sk}_j}, s'_\rho, s'_{\rho_1}, \ldots, s'_{\rho_\ell}, s'_{\mu_1}, \ldots, s'_{\mu_\ell})$. Thus, applying the forking lemma [1] to the H, we can successfully extract $(\tilde{\mathsf{sk}}^*, \rho^*)$ s.t. $C^* = g^{\tilde{\mathsf{sk}}^*} h^{\rho^*}$ with some non-negligible probability. In this case, we have $\tilde{\mathsf{sk}}^* = \hat{\mathsf{sk}}_{i_{u^*}}$ for some $u^* \in [1, N]$. If $u^* \neq \tilde{u}^*$, abort. Otherwise, where this happens with probability at least $1/N$, output $\tilde{\mathsf{sk}}^* = \hat{\mathsf{sk}}_{i_{u^*}} = \mathsf{sk}^*$ as $\mathcal{B}'_{\mathrm{DL}}$.

[**Case 3:** $\tilde{\mathsf{sk}}^* = \hat{\mathsf{sk}}_{i_{u^*}}$ **(and** $\tilde{\mathsf{sk}}_k = \hat{\mathsf{sk}}_{i_{u^*}}$**) for some** $u^* \in [1, N]$ **and** $\hat{\mathsf{pk}}_{i_{u^*}} \in Q_{\mathsf{reveal}}$ **in** σ^* **(and** σ_k**)**]: This remaining case does not happen except some negligible probability, as follows. The winning conditions of $\mathcal{A}$ are $\mathbf{pk}^* \subseteq Q_{\mathsf{KeyGen}} \setminus Q_{\mathsf{reveal}}$ or $\mathsf{CntBListedCUser}_{j^*} \geq |Q_{\mathsf{Reveal}}|$. In this case, for $\mathbf{pk}^* = (\hat{\mathsf{pk}}_{i_1}, \ldots, \hat{\mathsf{pk}}_{i_N})$ of the final output σ^*, we have $\hat{\mathsf{pk}}_{i_{u^*}} \in Q_{\mathsf{reveal}}$. Thus, since $\mathbf{pk}^* \notin Q_{\mathsf{KeyGen}} \setminus Q_{\mathsf{reveal}}$, we have $\mathsf{CntBListedCUser}_{j^*} \geq |Q_{\mathsf{Reveal}}|$.

In the final output σ^*, we have $t^* = B^{*\tilde{\mathsf{sk}}^*}$, $1 = B^{*\mu_i} t^{*-\rho_i}$ and $\tilde{A}_i^* = B_i^{*\mu_i} t_i^{*-\rho_i}$ for all $i \in [1, \ell]$. $t^* = B^{*\tilde{\mathsf{sk}}^*}$ and $1 = B^{*\mu_i} t^{*-\rho_i}$ imply $B^{*\mu_i} = B^{*\tilde{\mathsf{sk}}^* \cdot \rho_i}$. Thus, we have $\mu_i = \tilde{\mathsf{sk}}^* \cdot \rho_i$. Then, from $\tilde{A}_i^* = B_i^{*\mu_i} t_i^{*-\rho_i}$, we obtain $\tilde{A}_i^* = B_i^{*\tilde{\mathsf{sk}}^* \cdot \rho_i} t_i^{*-\rho_i}$. Since Verify outputs 1, $\tilde{A}_i^* \neq 1_{\mathbb{G}}$. Thus, $B_i^{*\tilde{\mathsf{sk}}^* \cdot \rho_i} t_i^{*-\rho_i} \neq 1_{\mathbb{G}}$, i.e., $t_i^* \neq B_i^{*\tilde{\mathsf{sk}}^*}$. On the other hand, $\tilde{\mathsf{sk}}^*$ is a revealed key (i.e., $\hat{\mathsf{sk}}_{i_{u^*}}$ s.t. $\hat{\mathsf{pk}}_{i_{u^*}} \in Q_{\mathsf{reveal}}$) in this case. In each CSign query, we have similar relations. Therefore, in each CSign query and the final output, the signature σ_k and σ^* ensures that all tickets in the blacklist are not produced using the secret key that is a revealed key (i.e., $\tilde{\mathsf{sk}}_k$ or $\tilde{\mathsf{sk}}^*$). $\mathcal{A}$ can generate a fake signature that does not ensure it (This is because for a randomly chosen $\tilde{c}$ in $\tilde{\pi}$, $\mathcal{A}$ can generate simulated other values in $\tilde{\pi}$), but the probability is negligible (This is because the probability that the randomly chosen $\tilde{c}$ is equal to $H(B, t, \{B_i, t_i, \tilde{A}_i\}_{i\in[1,\ell]}, R'_1, R'_2, \{R'_{3,i}, R'_{4,i}\}_{i\in[1,\ell]}, M)$ is negligible). On the other hand, in the unforgeability game, a ticket τ_k produced by CSign using a revealed secret key is added to the blacklist in CAddBL, and it is checked that the added ticket is not produced using a revealed secret key in the following CSign or the final output (A faked signature slipping through this check can be created, but the probability is negligible, as shown above). In the later check, the same revealed secret key cannot be used. Thus, counter $\mathsf{CntBListedCUser}_{j^*}$ means the number of revealed secret keys s.t. the relations $t_i^* \neq B_i^{*\tilde{\mathsf{sk}}^*}$ in Verify does not pass. Then, the condition $\mathsf{CntBListedCUser}_{j^*} \geq |Q_{\mathsf{Reveal}}|$ implies that all revealed keys do not pass the relations $t_i^* \neq B_i^{*\tilde{\mathsf{sk}}^*}$, i.e., no revealed secret key s.t. Verify outputs 1. Therefore, this case does not happen except some negligible probability. □

Theorem 3. *The proposed blacklistable ring signature scheme is anonymous under the DDH assumption in the random oracle model.*

Proof. This proof is based on a sequence of games. Consider the following games. Let S_i be the event that the adversary in **game** i successfully guesses b.

Game 0: This is the anonymity game (i.e., the experiment with the adversary $\mathcal{A}$ in the definition of the anonymity) for the proposed scheme, where H is simulated as a random oracle.

Game 1: **Game 1** is modified from **Game 0**, as follows. In $(\sigma^*, \tau^*) \leftarrow \mathsf{Sign}(\mathsf{param}, M^*, \mathsf{sid}^*, \mathbf{pk}^*, \hat{\mathsf{sk}}_{i_b}, \mathbf{BL}_{j^*})$, $\sigma^* = (C^*, z^*, R^*, \pi^*, \tilde{\pi}^*)$ is replaced by simulated values without $\hat{\mathsf{sk}}_{i_b}$. Select $C^* \xleftarrow{R} \mathbb{G}$, $c_1^*, \ldots, c_n^*, z^* \xleftarrow{R} \mathbb{Z}_p$, and compute $R^* = h^{z^*} \cdot \prod_{u=1}^N \mathsf{pk}'^{*\,c_u^*}_u$. Set $H(M^*, \mathbf{pk}'^*, R^*) = \sum_{u=1}^N c_u^*$ as the response of the random oracle. If the value has been set in the random oracle, abort. Then, using $\mathsf{NISA.Proof}$ as in the proposed scheme, obtain π^*. $\tilde{\pi}^*$ is simulated as follows. Select $\tilde{A}_1^*, \ldots, \tilde{A}_\ell^* \xleftarrow{R} \mathbb{G}$, and $\tilde{c}^*, s^*_{\mathsf{sk}_j}, s^*_\rho, s^*_{\rho_1}, \ldots, s^*_{\rho_\ell}, s^*_{\mu_1}, \ldots, s^*_{\mu_\ell} \xleftarrow{R} \mathbb{Z}_p$. Compute $R_1^* = g^{s^*_{\mathsf{sk}_j}} h^{s^*_\rho} C^{*\tilde{c}^*}, R_2^* = B^{*s^*_{\mathsf{sk}_j}} t^{*\tilde{c}^*}, R_{3,1}^* = B^{*s^*_{\mu_1}} t^{*-s^*_{\rho_1}}, \ldots, R_{3,\ell}^* = B^{*s^*_{\mu_\ell}} t^{*-s^*_{\rho_\ell}}$, $R_{4,1}^* = B_1^{*s^*_{\mu_1}} t_1^{*-s^*_{\rho_1}} \tilde{A}_1^{*\tilde{c}^*}, \ldots, R_{4,\ell}^* = B_\ell^{*s^*_{\mu_\ell}} t_\ell^{*-s^*_{\rho_\ell}} \tilde{A}_\ell^{*\tilde{c}^*}$. Then, set

$$H(B^*, t^*, \{B_i^*, t_i^*, \tilde{A}_i^*\}_{i\in[1,\ell]}, R_1^*, R_2^*, \{R_{3,i}^*, R_{4,i}^*\}_{i\in[1,\ell]}, M^*) = \tilde{c}^*$$

as the response of the random oracle. If the value has been set in the random oracle, abort.

Game 2: **Game 2** is modified from **Game 1**, as follows. In $(\sigma^*, \tau^*) \leftarrow \mathsf{Sign}(\mathsf{param}, M^*, \mathsf{sid}^*, \mathbf{pk}^*, \hat{\mathsf{sk}}_{i_b}, \mathbf{BL}_{j^*})$, for $\tau^* = (\mathsf{sid}^*, s^*, t^*)$ and $B^* = H(s^*, \mathsf{sid}^*)$, the value $t^* = B^{*\hat{\mathsf{sk}}_{i_b}}$ is modified to $t^* = B^{*z}$ with $z \xleftarrow{R} \mathbb{Z}_p$.

In **Game 2**, (σ^*, τ^*) consists of uniformly random values and zero-knowledge simulations. Thus, since the adversary has no information on b, we have $\Pr[S_2] = 1/2$. On the other hand, from the following lemmas, $|\Pr[S_0] - \Pr[S_1]|$ and $|\Pr[S_1] - \Pr[S_2]|$ are negligible. Therefore, $|\Pr[S_0] - 1/2|$ is negligible, which means that the proposed scheme is anonymous. □

Here, we show the lemmas.

Lemma 1. $|\Pr[S_0] - \Pr[S_1]|$ *is negligible.*

Proof. In **Game 1**, C^* is randomly generated, which is indistinguishable from the original, due to the information-theoretically hiding of the commitment. Other modifications are the zero-knowledge simulations, which have the same probability distributions from the original except aborting. The aborting probability is negligible. Thus, since **Game 0** are **game 1** are indistinguishable, $|\Pr[S_0] - \Pr[S_1]|$ is negligible. □

Lemma 2. $|\Pr[S_1] - \Pr[S_2]|$ *is negligible.*

Proof. Assume that $|\Pr[S_1] - \Pr[S_2]|$ is not negligible. We will construct the following adversary $\mathcal{B}_{\mathrm{DDH}}$ to break the

DDH assumption, which contradicts the DDH assumption.

Given $(\mathbb{G}, p, g) \leftarrow \mathcal{G}(\lambda)$, $u = g^x, v = g^y$, and $w = g^{xy}$ or $w = g^z$ where $x, y, z \xleftarrow{R} \mathbb{Z}_p$. Pick a random index $i^* \in [1, \mathsf{num_{key}}]$, and run $(\hat{\mathsf{pk}}_i, \hat{\mathsf{sk}}_i) \leftarrow \mathsf{KeyGen}(\mathsf{param})$ for all $i \in [1, \mathsf{num_{key}}]$ except $i = i^*$. Set $\hat{\mathsf{pk}}_{i^*} = u = g^x$ and define $\hat{\mathsf{sk}}_{i^*} = x$ which is unknown. Generate the other parameters in param as in Setup, and run the adversary $\mathcal{A}$ in the anonymity game with param and $Q_{\mathsf{KeyGen}} := \{\hat{\mathsf{pk}}_i\}_{i=1}^{\mathsf{num_{key}}}$. In the oracles, simulate H as a random oracle. For Reveal oracle, if $\mathsf{pk}_k \neq \hat{\mathsf{pk}}_{i^*}$, return the corresponding $\hat{\mathsf{sk}}_i$. If $\mathsf{pk}_k = \hat{\mathsf{pk}}_{i^*}$, abort (Note that for the winning adversary, this case does not happen for $i^* = i_b$, since $\hat{\mathsf{pk}}_{i_0}, \hat{\mathsf{pk}}_{i_1} \in (Q_{\mathsf{KeyGen}} \setminus Q_{\mathsf{Reveal}}) \cap \mathbf{pk}^*$). For HSign oracle, if $\mathsf{pk}_k \neq \hat{\mathsf{pk}}_{i^*}$, return the signature and ticket using the corresponding sk_k. If $\mathsf{pk}_k = \hat{\mathsf{pk}}_{i^*}$, compute the simulated signature $\sigma_k = (C, z, R, \pi, \tilde{\pi})$, as the case 2 in the proof of the unforgeability. CSign, NewBL, HAddBL, CAddBL queries are conducted as in the descriptions of the oracles. Then, $\mathcal{A}_1$ outputs $M^*, \mathbf{pk}^*, i_0, i_1, St$. Pick $b \leftarrow \{0,1\}$. If $i_b \neq i^*$, abort. Otherwise, as $(\sigma^*, \tau^*) \leftarrow \mathsf{Sign}(\mathsf{param}, M^*, \mathbf{pk}^*, \hat{\mathsf{sk}}_{i_b})$, set $H(s^*, \mathsf{sid}^*) = v = g^y$ as the response of the random oracle. If the value for $H(s^*, \mathsf{sid}^*)$ has been set in the random oracle, abort. Otherwise, set $t^* = w$, where $t^* = B^{*\hat{\mathsf{sk}}_{i^*}}$ holds for $B^* = H(s^*, \mathsf{sid}^*)$ in case of $w = g^{xy}$. Given (σ^*, τ^*), the adversary $\mathcal{A}$ outputs the guess b'. If $b = b'$, $\mathcal{B}_{\mathrm{DDH}}$ outputs 1, and otherwise outputs 0. Then, $\Pr[\mathcal{B}_{\mathrm{DDH}}(\mathbb{G}, p, g, u = g^x, v = g^y, w = g^{xy}) = 1] = \Pr[\neg\mathsf{abort}]\Pr[S_1]$ where $\Pr[\neg\mathsf{abort}]$ is the probability that $\mathcal{B}_{\mathrm{DDH}}$ does not abort. Also, we have $\Pr[\mathcal{B}_{\mathrm{DDH}}(\mathbb{G}, p, g, u = g^x, v = g^y, w = g^z) = 1] = \Pr[\neg\mathsf{abort}]\Pr[S_2]$. Since the aborting probability due to the failure of setting the hash oracles is negligible, $\Pr[\neg\mathsf{abort}]$ is the probability of successfully guessing i^*, which is at least $1/\mathsf{num_{key}}$. Thus, we have

$$
\begin{aligned}
&|\Pr[\mathcal{B}_{\mathrm{DDH}}(\mathbb{G}, p, g, u = g^x, v = g^y, w = g^{xy}) = 1] \\
&\qquad - \Pr[\mathcal{B}_{\mathrm{DDH}}(\mathbb{G}, p, g, u = g^x, v = g^y, w = g^z) = 1]| \\
&= |\Pr[\neg\mathsf{abort}]\Pr[S_1] - \Pr[\neg\mathsf{abort}]\Pr[S_2]| \\
&= \Pr[\neg\mathsf{abort}]|\Pr[S_1] - \Pr[S_2]| \\
&\geq 1/\mathsf{num_{key}} \cdot |\Pr[S_1] - \Pr[S_2]|
\end{aligned}
$$

Since $|\Pr[S_1] - \Pr[S_2]|$ is non-negligible and $\mathsf{num_{key}}$ is polynomial in λ, $\mathcal{B}_{\mathrm{DDH}}$ breaks the the DDH assumption. □

6. Efficiency Considerations

In Table 1, the comparison of the signature size between the original *DualRing-EC* and the proposed scheme is shown, where the numbers of $\mathbb{G}$-elements and $\mathbb{Z}_p$-elements are counted. The proposed scheme still achieves the $O(\log N)$ signature size for the ring size N, although the added blacklisting mechanism increases the size by $O(\ell)$ for the blacklist size ℓ, as in BLAC [15]. Specifically, in case that p is a 256-bit prime, $N = 1,024$, and $\ell = 100$, the signature size is about 10KBytes.

As for the computational costs, the signing and verification costs in our scheme is both $O(N + \ell)$, because, in both signing and verification algorithms, *DualRing-EC* part needs $O(N)$ exponentiations and BLAC part needs $O(\ell)$ exponentiations. Since the other ring signature scheme [3], [11] with $O(\log N)$ signature size needs $O(N \log N)$ signing cost, the adoption of DualRing provides a more efficient signing algorithm.

Table 1 Signature sizes.

	#$\mathbb{G}$-elements	#$\mathbb{Z}_p$-elements
DualRing-EC [17]	$2\log N + 1$	3
Proposed	$2\log N + \ell + 2$	$2\ell + 6$

7. Applications to Decentralized Blacklistable Anonymous Credentials

Our blacklistable ring signatures can easily be applied to decentralized blacklistable anonymous credentials, as follows. The anonymous credential system consists of a registration protocol and an authentication protocol. In the registration protocol, a user registers himself to the system, where the user put the information to confirm the validity of the user to a public ledger such as blockchain. For example, the information includes some proof and attributes to aid the Service Provider (SP) in deciding whether the user is valid. Then, using the authentication protocol, the user can be anonymously authenticated by the SP in each service usage of session ID sid, and in addition a user in a service usage can be blacklisted. In the application of our ring signatures, a user registers with the system by sending his public key to the public ledger. Then, in the authentication, the user sends his ring signature on session sid to the SP, where a misbehaving user on the session is blacklisted by adding the ticket of the authentication to a blacklist managed by the SP. Compared to the previous DL-based system [16] with $O(N)$ proof size, our system achieves $O(\log N)$ proof size.

On the other hand, the previous system has a reputation-based blacklisting mechanism. In the reputation-based system, the behaviors of the user in the service are rated by the SP. The rated scores are either positive ones or negative ones, and belong to different categories. The scores are accumulated as the reputation of the user in the blacklist. In each authentication, the user also proves that the user's reputation satisfies some complex policies on the accumulated scores in the categories. Thus, compared to the simple blacklist where a user who misbehaves once is added, the reputation-based authentication becomes more flexible. The extension of our scheme to the reputation-based mechanism is one of our future works.

8. Conclusion

In this paper, we have proposed a DL-based blacklistable ring signature scheme from DualRing, where the signature size is $O(\log N)$ for ring size N. The scheme can be applied to a DL-based decentralized blacklistable anonymous credentials, where the authentication data size is $O(\log N)$. We defined the security model, and proved the security based on

the DDH assumption (implying the DL assumption) in the random oracle model.

Our future works includes evaluations on implementations, and the extension to the reputation-based authentication.

Acknowledgments

This work was partially supported by JSPS KAKENHI Grant Number 22K12027.

References

[1] M. Bellare and G. Neven, "Multi-signatures in the plain public-key model and a general forking lemma," Proc. 13th ACM Conference on Computer and Communications Security (CCS 2006), pp.390–399, 2006.
[2] A. Bender, J. Katz, and R. Morselli, "Ring signatures: Stronger definitions, and constructions without random oracles," J. Cryptol., vol.22, no.1, pp.114–138, 2009.
[3] J. Bootle, A. Cerulli, P. Chaidos, E. Ghadafi, J. Groth, and C. Petit, "Short accountable ring signatures based on DDH," Proc. 20th European Symposium on Research in Computer Security (ESORICS 2015), Part I, LNCS 9326, pp.243–265, Springer-Verlag, 2015.
[4] J. Camenisch and M. Stadler, "Efficient group signature schemes for large groups," Advances in Cryptology — CRYPTO'97, LNCS 1294, pp.410–424, Springer-Verlag, 1997.
[5] J. Camenisch and A. Lysyanskaya, "An efficient system for non-transferable anonymous credentials with optional anonymity revocation," Advances in Cryptology — EUROCRYPT 2001, LNCS 2045, pp.93–118, Springer-Verlag, 2001.
[6] M. Chase and A. Lysyanskaya, "On signatures of knowledge," Advances in Cryptology — CRYPTO 2006, LNCS 4117, pp.78–96, Springer-Verlag, 2006.
[7] D. Chaum and E. van Heijst, "Group signatures," Advances in Cryptology — EUROCRYPT'91, LNCS 547, pp.241–246, Springer-Verlag, 1991.
[8] D. Chaum, "Security without identification: Transaction systems to make big brother obsolete," Commun. ACM, vol.28, no.10, pp.1030–1044, 1985.
[9] R. Cramer, Damgård, and B. Schoenmakers, "Proofs of partial knowledge and simplified design of witness hiding protocols," Advances in Cryptology — CRYPTO'94, LNCS 839, pp.174–187, Springer-Verlag, 1994.
[10] C. Garman, M. Green, and I. Miers, "Decentralized anonymous credentials," Proc. 21st Annual Network and Distributed System Security Symposium, (NDSS 2014), 2014.
[11] J. Groth and M. Kohlweiss, "One-out-of-many proofs: Or how to leak a secret and spend a coin," Advances in Cryptology - EUROCRYPT 2015, Part II, LNCS 9057, pp.253–280, Springer-Verlag, 2015.
[12] T. Nakanishi, A. Iriboshi, and K. Imai, "Short DL-based blacklistable ring signatures from DualRing," Proc. Tenth International Symposium on Computing and Networking (CANDAR 2022), Track 5, pp.137–143, 2022.
[13] T.P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," Advances in Cryptology - CRYPTO'91, LNCS 576, pp.129–140, Springer-Verlag, 1992.
[14] R.L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," Advances in Cryptology — ASIACRYPT 2001, LNCS 2248, pp.552–565, Springer-Verlag, 2001.
[15] P.P. Tsang, M.H. Au, A. Kapadia, and S.W. Smith, "Blacklistable anonymous credentials: Blocking misbehaving users without TPPs," Proc. ACM Conference on Computer and Communications Security 2007 (ACM-CCS'07), pp.72–81, 2007.
[16] R. Yang, M.H. Au, Q. Xu, and Z. Yu, "Decentralized blacklistable anonymous credentials with reputation," Comput. Secur., vol.85, pp.353–371, 2019. Conference version appeared in ACISP2018.
[17] T.H. Yuen, M.F. Esgin, J.K. Liu, M.H. Au, and Z. Ding, "DualRing: Generic construction of ring signatures with efficient instantiations," Advances in Cryptology - CRYPTO 2021, Part I, LNCS 12825, pp.251–281, Springer-Verlag, 2021.

Toru Nakanishi received the M.S. and Ph.D. degrees in information and computer sciences from Osaka University, Japan, in 1995 and 2000 respectively. He joined the Department of Information Technology at Okayama University, Japan, as a research associate in 1998, and moved to the Department of Communication Network Engineering in 2000, where he became an assistant professor and an associate professor in 2003 and 2006 respectively. In 2014, he moved to the Department of Information Engineering at Hiroshima University as a professor. His research interests include cryptography and information security. He is a member of the IPS of Japan.

Atsuki Iriboshi received B.E. degree in Information Engineering from Hiroshima University, Japan in 2021. His research interests include cryptography and information security.

Katsunobu Imai received the B.E., M.E. and Dr. E. degrees from Osaka University in 1990, 1992, and 1999 respectively. From 1993 to 2007, he was a research associate of the Graduate School of Engineering, Hiroshima University. He is presently an assistant professor of the Graduate School of Advanced Science and Engineering, Hiroshima University. His present interests include cellular automata and natural computing.