Leakage-Resilient Group Signature: Definitions and Constructions

Jianye Huang, Qiong Huang*

College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China.

Abstract

Group signature scheme provides group members a way to sign messages without revealing their identities. Anonymity and traceability are two essential properties in a group signature system. However, these two security properties hold based on the assumption that all the signing keys are perfectly secret and leakage-free. On the another hand, on account of the physical imperfection of cryptosystems in practice, malicious attackers can learn fraction of secret state (including secret keys and intermediate randomness) of the cryptosystem via side-channel attacks, and thus breaking the security of whole system.

To address this issue, Ono *et al.* introduced a new security model of group signature, which captures randomness exposure attacks. They proved that their proposed construction satisfies the security requirements of group signature scheme. Nevertheless, their scheme is only provably secure against randomness exposure and supposes the secret keys remains leakage-free. In this work, we focus on the security model of leakage-resilient group signature based on bounded leakage setting and propose three new black-box constructions of leakage-resilient group signature secure under the proposed security models.

Key words: group signature, full anonymity, full traceability, black-box construction, leakage resilience

1. Introduction

Group Signature. Group signature, introduced in the seminal work of Chaum and Heyst [16], provides group members with anonymity while signing messages. That is, every group members is able to anonymously sign messages on behalf of their group. For example, Alice is an employee (group member) of a large company. She is able to submit an anonymous tip and convinces all verifiers that it is signed by one of employees in Alice's company without revealing her identity. On the other hand, if Alice abuses her anonymity power to sign bogus messages, her identity can be traced by her employer Bob (group manager). Briefly, a group signature scheme must satisfy the following basic security requirements [3].

Correctness. Signatures honestly generated by group members can always be verified while invalid signatures always failing verification process.

*Corresponding Author.

Email addresses: jianye_Huang@stu.scau.edu.cn (Jianye Huang), qhuang@scau.edu.cn (Qiong Huang)

Unforgeability. Only group members can sign messages on behalf of their group. In other word, non-members cannot efficiently forge valid group signatures.

Anonymity. Given any message/signature pair, the identity of the real signer cannot be revealed without group manager's help.

Traceability. Given any valid message/signature pair, the group manager is able to trace the real signer's identity.

Unlinkability. Given any two message/signature pairs, no probabilistic polynomial-time distinguisher can decide if the signatures are both from the same signer or not.

Exculpability. Even if all other group members and the managers collude, they cannot forge a signature for an honest group member.

Coalition Resistance. A colluding subset of group members cannot generate a valid signature that is traced to other non-participants.

A formal definition was proposed by Bellare, Micciancio and Warinschi [8]. In their work, they proposed three security properties *Correctness*, *Full-Anonymity* and *Full-Traceability.* They showed that a group signature fulfilling these three properties implies the satisfaction of all the security requirements described above. More details are presented in Sect. 3.3.

Side-Channel Attacks and Leakage Models. Unfortunately, when we consider physical attacks (e.g. side-channel attack) in practice, most existing group signature schemes does not satisfy the aforementioned security requirements any more. For instance, Alice's co-worker, say Eve, measures the power consumption of her computer while Alice is sending an anonymous tip and signing it with her secret key. Then the resulting power traces may subsequently lead to the recovery of Alice's secret key.

Over the last twenty years, researchers have discovered a wide range of side channel attacks, including running-time attacks [34], fault detection [11], differential power analysis (DPA) [33, 18], electromagnetic radiation analysis [23, 39], and etc. In 2009, Halderman *et al.* [25] proposed the well-known *cold-boot attack* on private keys stored in the memory of devices. Such memory leakage showed that obtaining secret information (even not being used) is possible. In 2017, Craig Ramsay [40] proposed an efficient tempest attack against AES encryption algorithms. Ramsay showed how to recover the encryption keys from two realistic AES-256 implementations within one meter attack distance and few minutes. Furthermore, their equipment is only pocket-size and low-cost. Nowadays, it is feasible to obtain secret keys (or other sensitive information like intermediate randomness) from the cryptosystem via such physical attacks. On the other hand, traditional cryptography does not capture the side channel attacks that focus the weakness of devices themselves and steer clear of the intractability of mathematical hard problems. Therefore, it is a great threat for many cryptographic systems and countermeasures for protecting cryptographic systems from such physical attacks are imperative.

For this purpose, in 2008, Dziembowski *et al.* first introduced the notion of leakage-resilient cryptography (LRC) [21]. Micali and Reyzin [37] proposed the well-known atom, *Only Computation Leaks Information*. They assumed that only the part of secret information that is being operated will be possibly leaked to attackers. It is not trivial, however, providing a heuristic solution to formalize the leakage in practice. Inspired by Akavia, Goldwasser and Vaikuntanathan's work [1], leakage is defined as an adversarially chosen function of private state. The adversary is given an additional leakage oracle, which on input an adversarially chosen function and return the leakage (function value) of private state of cryptographic system. In terms of the restriction of adversaries' power, there are different leakage models. For example, in *bounded leakage model* [2, 1], there must be sufficient entropy left ensuring the system is not fully ruined. That is, the leakage amount is bounded. In *noisy leakage model* [20, 21], instead of restricting the amount of leakage to a concrete bound, arbitrarily large leakage is allowed as long as the secret information remains sufficient min-entropy. It is more accurate to model leakage in the reality, and thus can capture more practice attacks comparing with bounded leakage model. In *continual leakage model* [13, 30], the secret keys should be periodically updated without modifying the corresponding public keys, which allows the adversary to obtain in total arbitrarily many bits of the cryptosystem's private state as long as the leakage between two invocations of the secret key refreshing algorithms is bounded. Namely, the leakage rate is bounded.

1.1. Related Works

Group Signature. Subsequent to the introduction of Chaum and Heysts seminal work [16], many proposed schemes, including [7, 15, 10], focused on improving the performance of Chaum-Heysts seminal construction. [4, 14, 6] investigated the dynamics property to support the member revocation. [14, 6] studied the independent-generation of group member keys with member revocation to support large group. [9] considered the forward security of group signatures. Chen and Pedersen [17] proposed the first group signature scheme with dynamical group size. Camenisch and Stadler [15] presented the first group signature scheme for large groups, in which the size of group public key and signatures is independent of the group size. To support efficient member revocation, Kim *et al.* [32] constructed a group signature scheme from *traitor tracing schemes*. In 2003, Ateniese and de Medeiros [5] proposed another group scheme supporting trapdoor privacy. That is, no third party knows any trapdoor secret, and therefore a same cryptographic domain can be shared in different groups without security loss. It is a great advantage over other group signature schemes.

Leakage-Resilient Signature Primitives. There has been impressive progress in leakage-resilient cryptography. In 2009, Katz and Vaikuntanathan [31] proposed a secure black-box construction of signature scheme in the bounded leakage model. Boyle *et al.* [12] and Yuen *et al.* [45] considered the fully leakage-resilient unforgeability, which takes account of leakage on both signing keys and private randomness. Wang *et al.*

[43, 44] and Huang *et al.* [27, 28] studied the signature schemes with *strong unforgeability* in different leakage settings. Huang *et al.* [26] proposed black-box constructions of identity-based signature scheme and certificateless signature scheme. [29] proposed another efficient method to construct a leakage-resilient signature via introducing the notation of *leakage-resilient dual form signature.* Many other leakage-resilient models are studied in literatures, for example, [36, 22].

Leakage-Resilient Group Signature. To the best of our knowledge, however, there are seldom works on leakage-resilient group signature. Ono *et al.* [38] proposed a new security model of group signature for capturing full randomness exposure and proved their proposed scheme satisfies the security requirements. Nevertheless, their scheme is only provably secure against randomness exposure and the secret keys is still assumed to be leakage-free. How to construct a (fully) leakage-resilient group signature scheme remains an open problem.

1.2. Our Contributions

In this work, we propose three black-box constructions of group signature scheme that satisfies *correctness*, *full anonymity*, *full traceability* and *leakage resilience* at the same time. To be more concrete, we make the following contributions in this paper.

1. First of all, we revisit the security models of group signature and give a formal definition of leakage-resilient group signature scheme.
2. Second, we present a new construction of group signature scheme via black-box method, constructing from a leakage-resilient identity-based signature scheme and a CCA-secure leakage-resilient encryption scheme. We further prove that the leakage bound of resulting construction is the same as the lower-bound of the IBS scheme and the encryption scheme.
3. Third, we show how to construct a group signature scheme with leakage resilience from a leakage-resilient signature scheme and a CCA-secure leakage-resilient encryption scheme. The leakage bound of resulting construction is the same as the lower-bound of the underlying signature scheme and encryption scheme.
4. Forth, we further propose a black-box construction of group signature scheme from a leakage-resilient signature scheme and a CPA-secure leakage-resilient scheme. The leakage bound of resulting construction is the same as the lower-bound of the underlying signature scheme and encryption scheme.

2. Preliminaries and Definitions

2.1. Notations and Abbreviations

Definition 1 (Negligible Function [35]) We say a function $\mu(x) : \mathbb{N} \rightarrow \mathbb{R}$ is negligible if for every positive polynomial $\mathsf{poly}(\cdot)$ there exists a positive integer $K \in \mathbb{N}$ s.t. $\forall k > K, |\mu(k)| < \frac{1}{\mathsf{poly}(k)}$ holds.

Let $k \in \mathbb{N}$ be the security parameter and 1^k be its unary representation. We assume that all the algorithms are given the unary coding of the security parameter k as input, which will not be explicitly involved as input of the algorithms when it is clear from the context. Let $\mathsf{negl}(k)$ denotes a negligible function of security parameter k. Denote $\mathcal{B}(x_1, x_2, \cdots, x_n; r)$ by the output of evaluating PPT algorithm $\mathcal{B}$ on input $x_1, x_2, \cdots, x_n$ with randomness r. For any finite set S, we denote $\alpha \overset{\$}{\leftarrow} S$ by the operation of uniformly sampling a random element α from S. Let $\|\alpha\|$ be the size of element α. We write $[n]$ to denote the set of positive integers from 1 to n, i.e. $\{1, 2, \cdots, n\}$. We use the abbreviation PPT to mean probabilistic polynomial-time. The abbreviation KGC signifies key generation center.

2.2. Unbounded Simulation-Sound Non-Interactive Zero Knowledge

Definition 2 (Unbounded Simulation-Sound NIZK [41, 19]) An unbounded simulation-sound NIZK proof system $\Pi = (l := l(k), \mathcal{P}, \mathcal{V}, \mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2))$ for a $\mathcal{NP}$-language L with relation R involves four PPT algorithms $\mathcal{P}, \mathcal{V}, \mathcal{S}_1, \mathcal{S}_2$ that satisfy the following properties.

Completeness. It holds that $\Pr[\mathcal{V}(x, \mathcal{P}(x, w, r), r) = 1 : (x, w) \in \{(x, w)|R(x, w) = 1\} \wedge r \in \{0,1\}^l] = 1$.

Simulation Soundness. Let T be the transcripts generated by the simulation algorithm $\mathcal{S}_2$ and Succ be the event $(x, \pi) \notin \mathsf{T} \wedge x \notin L \wedge \mathcal{V}(x, \pi, r) = 1$. Then we call Π satisfies *simulation-soundness* if the advance of any PPT adversary $\mathcal{A}$ succeeding in generating a valid proof on a false statement x is negligible, i.e. $\Pr[\mathsf{Succ} : (r, \rho) \leftarrow \mathcal{S}_1(1^k), (x, \pi) \leftarrow \mathcal{A}^{\mathcal{S}_2(\cdot, r, \rho)}(r)] \leq \mathsf{negl}(k)$.

Unbounded Zero Knowledge. We call Π satisfies *unbounded zero knowledge* if for all PPT adversaries $\mathcal{A}$,

$$|\Pr[\mathsf{Expt}_0^{\mathcal{A}}(k) = 1] - \Pr[\mathsf{Expt}_1^{\mathcal{A}}(k) = 1]| \leq \mathsf{negl}(k)$$

always holds, where $\mathsf{Expt}_0(k)$ and $\mathsf{Expt}_1(k)$ is defined as below.

```
$\mathsf{Expt}_0(k)$
    $r \overset{\$}{\leftarrow} \{0,1\}^l$
    $\mathcal{S}'_r(x, w) := \mathcal{P}(x, w, r)$
    Return $\mathcal{A}^{\mathcal{S}'_r(\cdot,\cdot)}(r)$.
```

```
$\mathsf{Expt}_1(k)$
    $(r, \rho) \overset{\$}{\leftarrow} \mathcal{S}_1(1^k)$
    $\mathcal{S}'_{r,\rho}(x, w) := \mathcal{S}_2(x, r, \rho)$
    Return $\mathcal{A}^{\mathcal{S}'_{r,\rho}(\cdot,\cdot)}(r)$.
```

3. Definitions and Security Models of Signature Schemes

3.1. Digital Signature

Definition 3 (Digital Signature [24]) A digital signature scheme Σ consists of the following three polynomial time algorithms.

Key Generation Algorithm. On input a security parameter k represented in unary, the key generation algorithm produces a verification/signing key pair $(vk, sk) \leftarrow \mathsf{Kg}(1^k)$.

Signing Algorithm. On input a signing key sk and a message m, the algorithm returns a valid signature $\sigma \leftarrow \mathsf{Sig}(sk, m)$.

Verification Algorithm. On input a verification key vk, a message m and a signature σ, the algorithm returns 1/0 (acceptance/rejection) $\leftarrow \mathsf{Ver}(vk, m, \sigma)$.

Security Model. The standard security notation of digital signature is *existential unforgeability under adaptive chosen-message attack* (UF-CMA, for short). Consider the following unforgeability experiment for a signature scheme $\Sigma = (\mathsf{Kg}, \mathsf{Sig}, \mathsf{Ver})$, where $\mathcal{C}$ is a challenger and $\mathcal{F}$ is a PPT forger who tries to produce a valid signature on a new message.

UF-CMA Experiment $\mathsf{UF\text{-}CMA}_{\mathcal{F},\Sigma}(k)$:

1. $\mathcal{C}$ invokes $(vk, sk) \leftarrow \mathsf{Kg}(1^k)$ and gives the verification key vk to $\mathcal{F}$ while keeping the signing key sk secret. $\mathcal{C}$ maintains an initially empty query list $\mathcal{L}_s$.
2. $\mathcal{F}$ adaptively accesses to a signing oracle $\mathcal{O}_s(\cdot)$ for polynomially many times. Given a message m, the signing oracle returns a valid signature σ on m. Update the signing query list $\mathcal{L}_s = \mathcal{L}_s \cup \{m\}$.
3. Finally, $\mathcal{F}$ outputs a forgery $(\hat{m}, \hat{\sigma})$ and wins iff. (a) $\mathsf{Ver}(\hat{m}, \hat{\sigma}) = 1$ and (b) $\hat{m} \notin \mathcal{L}_s$.

Definition 4 (Unforgeability under Adaptive Chosen-Message Attack, UF-CMA [24]) A signature scheme Σ is existentially unforgeable under adaptive chosen-message attack (or is UF-CMA secure) if the advantage of any PPT forger $\mathcal{F}$ in the experiment $\mathsf{UF\text{-}CMA}_{\mathcal{F},\Sigma}(k)$ is negligible, i.e. for all PPT forger $\mathcal{F}$,

$$\Pr\left[\mathsf{Ver}(vk, \hat{m}, \hat{\sigma}) = 1 \wedge \hat{m} \notin \mathcal{L}_s : (\hat{m}, \hat{\sigma}) \leftarrow \mathcal{F}^{\mathcal{O}_s(\cdot)}(vk)\right] \leq \mathsf{negl}(k).$$

Leakage-Resilient Unforgeability. To model a forger $\mathcal{F}$ who is allowed to launch side-channel attacks against signature scheme Σ and obtain a fraction of signing sk, $\mathcal{F}$ is given a leakage oracle $\mathcal{O}_L(\cdot)$. On input the ith adversarially chosen leakage function f_i, the leakage oracle $\mathcal{O}_L(f_i)$ returns $\Lambda_i := f_i(sk)$ where $\|\Lambda_i\| \leq \lambda_i$. Without loss of generality, we suppose that the $\mathcal{F}$ makes leakage queries at most q_L times. We have following definition.

Definition 5 (Leakage-Resilient Unforgeability [27, 29]) A signature scheme Σ is λ-leakage-resilient and existentially unforgeable under adaptive chosen-message attacks (or is λ-LR-UF-CMA secure) if no PPT forger $\mathcal{F}$ has non-negligible advantage in the modified experiment, i.e. for all PPT forger $\mathcal{F}$,

$$\Pr\left[\mathsf{Ver}(vk, \hat{m}, \hat{\sigma}) = 1 \wedge \hat{m} \notin \mathcal{L}_s \wedge \sum_{i=1}^{q_L} \lambda_i \leq \lambda : (\hat{m}, \hat{\sigma}) \leftarrow \mathcal{F}^{\mathcal{O}_s(\cdot), \mathcal{O}_L(\cdot)}(vk)\right] \leq \mathsf{negl}(k).$$

3.2. ID-based Signature

Definition 6 (Identity-Based Signature, IBS [42]) An identity-based signature scheme $\mathcal{IBS}$ consists of four PPT algorithms $\mathsf{Setup}, \mathsf{Extract}, \mathsf{IB\text{-}Sig}, \mathsf{IB\text{-}Ver}$, respectively called setup algorithm, key extraction algorithm, signing algorithm and verification algorithm.

Setup Algorithm. On input a security parameter k represented in unary, KGC invokes setup algorithm Setup to generate a master public/secret key pair $(mpk, msk) \leftarrow \mathsf{Setup}(1^k)$.

Extraction Algorithm. On input a master key msk and a user identify id, KGC invokes extraction algorithm $\mathsf{Extract}$ to generate a signing key for user id, i.e. $sk_{id} \leftarrow \mathsf{Extract}(msk, id)$.

Signing Algorithm. On input a user signing key sk_{id} and a message m, the user invokes signing algorithm $\mathsf{IB\text{-}Sig}$ to produce a signature $\sigma \leftarrow \mathsf{IB\text{-}Sig}(sk_{id}, m)$.

Verification Algorithm. On input a user identity id, a message m and a signature σ, the verifier invokes verification algorithm $\mathsf{IB\text{-}Ver}$ to test the validity the purported signature, i.e. 1/0 (acceptance/rejection) $\leftarrow$ $\mathsf{IB\text{-}Ver}(id, m, \sigma)$.

Remark 1 The master public key mpk is implicitly involved in extraction algorithm, signing algorithm and verification algorithm and therefore it is omitted in this paper for simplicity.

Security Model. The standard security notion of IBS scheme is *existential unforgeability under adaptive chosen-message and chosen-identity attack.* Let $\mathcal{IBS} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{IB\text{-}Sig}, \mathsf{IB\text{-}Ver})$ be an IBS scheme. Consider the following unforgeability experiment where $\mathcal{C}$ is challenger and $\mathcal{F}$ is a PPT forger.

UF-CMIA Experiment $\mathsf{CMIA\text{-}IDA}_{\mathcal{F},\mathcal{IBS}}(k)$:

1. $\mathcal{C}$ generates master public/secret keys by invoking $(mpk, msk) \leftarrow \mathsf{Setup}(k)$. Extraction query list $\mathcal{L}_e$ and signing query list $\mathcal{L}_s$ are initialized to be empty. Initially set system state $\mathsf{State} = \{(KGC, msk)\}$ and leakage amount $\mathsf{L} = 0$.
2. Invoke $\mathcal{F}(mpk)$ and answer queries from $\mathcal{F}$ as follows.

   CreateUser Oracle $\mathcal{CO}$. Given a user identity id, if $(id, *) \in \mathsf{State}$, none of the process takes place. Otherwise, generate user signing key by invoking $sk_{id} \leftarrow \mathsf{Extract}(msk, id)$. Update the system state $\mathsf{State} = \mathsf{State} \cup \{(id, sk_{id})\}$.

   Extraction Oracle $\mathcal{EO}$. Given a user identity id, retrieve the corresponding sk_{id} s.t. $(id, sk_{id}) \in \mathsf{State}$ and return sk_{id}. Update the extraction query list $\mathcal{L}_e = \mathcal{L}_e \cup \{id\}$.

   Signing Oracle $\mathcal{SO}$. Given a user identity id and a message m, retrieve the corresponding sk_{id} from State and return $\sigma \leftarrow \mathsf{IB\text{-}Sig}(sk_{id}, m)$. Update signing query list $\mathcal{L}_s = \mathcal{L}_s \cup \{(id, m)\}$.

   Leakage Oracle $\mathcal{LO}$. Given a function f, compute $\Lambda := f(\mathsf{State})$. If $\mathsf{L} + \|\Lambda\| > \lambda$, abort. Otherwise, return Λ and update the leakage amount $\mathsf{L} = \mathsf{L} + \|\Lambda\|$.
3. Finally, $\mathcal{F}$ outputs a forgery $(\hat{id}, \hat{m}, \hat{\sigma})$. Denote by Forge the event: (a) $\mathsf{IB\text{-}Ver}(\hat{id}, \hat{m}, \hat{\sigma}) = 1$, (b) $\hat{id} \notin \mathcal{L}_e$ and (c) $(\hat{id}, \hat{m}) \notin \mathcal{L}_s$.

Remark 2 Without loss of generality, suppose that each id used in other oracle queries has already been created by $\mathcal{CO}(id)$.

Definition 7 (Leakage-Resilient Identity-Based Signature, LR-IBS [26]) An IBS scheme $\mathcal{IBS}$ is λ-leakage-resilient and existentially unforgeable under adaptive chosen-message and chosen-ID attack (or is λ-LR-UF-CMIA secure) if no probabilistic polynomial-timeforger $\mathcal{F}$ has a non-negligible advantage in the Experiment $\mathsf{CMIA\text{-}IDA}_{\mathcal{F},\mathcal{IBS}}(k)$, i.e.

$$\Pr\left[\begin{array}{l}\mathsf{IB\text{-}Ver}(\hat{id},\hat{m},\hat{\sigma})=1\\ \hat{id}\notin\mathcal{L}_e,(\hat{id},\hat{m})\notin\mathcal{L}_s:(\hat{id},\hat{m},\hat{\sigma})\leftarrow\mathcal{F}^{\mathcal{CO}(\cdot),\mathcal{EO}(\cdot),\mathcal{SO}(\cdot),\mathcal{LO}(\cdot)}(vk)\\ \qquad\quad\mathsf{L}\leq\lambda\end{array}\right]\leq\mathsf{negl}(k).$$

3.3. Group Signature

Definition 8 (Group Signature [16]) A group signature scheme $\mathcal{GS}$ consists of the following four polynomial-time algorithms.

$\mathsf{Setup}(1^k)\rightarrow(\mathsf{PP},msk,tsk)$. The setup algorithm takes as input the security parameter k (in unary representation) and outputs a group parameter PP, a master key msk for member authorization and a tracing key tsk for identifying signers.

$\mathsf{Join}(\mathsf{PP},msk,id)\rightarrow sk_{id}$. The join algorithm takes as input the group parameter PP, the master key msk and an identity id, output signing key sk_{id}, which is sent to user via a secure channel.

$\mathsf{GSig}(\mathsf{PP},sk_{id},m)\rightarrow\sigma$. On input public parameter PP, a signing key sk_{id} of user id and a message m, the signing algorithm outputs a valid group signature σ.

$\mathsf{GVer}(\mathsf{PP},m,\sigma)\rightarrow 0/1$. On input the public parameter PP, a message m and a signature σ, if σ is a valid signature on m signed by one of the group members, the verification algorithm outputs 1 and 0 otherwise.

$\mathsf{Trace}(\mathsf{PP},tsk,m,\sigma)\rightarrow id/\perp$. The trace algorithm takes as input the public parameter PP, a tracing key tsk and the message/signature pair (m,σ), output the real signer id who produces the signature σ on message m. If id is not a group member, output $\perp$.

We follow the security definition presented in Bellare, Micciancio and Warinschis work [8]. That is, a group signature must satisfy three properties: *correctness*, *full anonymity* and *full traceability*. Details are as follows.

Correctness. Generally, we say that a group signature scheme satisfies correctness if any honestly generated signature is always verified. I.e.

$$\Pr\left[\mathsf{GVer}(\mathsf{PP},m,\mathsf{GSig}(\mathsf{PP},sk_{id},m))=1:\begin{array}{r}(\mathsf{PP},msk,tsk)\leftarrow\mathsf{Setup}(1^k)\\ (m,id)\xleftarrow{\$}\{0,1\}^*\\ sk_{id}\leftarrow\mathsf{Join}(\mathsf{PP},msk,id)\end{array}\right]=1.$$

Security Model. Let $\mathsf{EList},\mathsf{SList},\mathsf{CList},\mathsf{TList}$ be lists of enrollment queries, signing queries, corrupt queries and tracing queries respectively and L be the leakage amount. Initially set $\mathsf{EList}=\mathsf{SList}=\mathsf{CList}=\mathsf{TList}=\mathsf{State}=\phi$ and $\mathsf{L}=0$. Consider the following oracles.

$\mathsf{OEnroll}(\cdot)$ The enrollment oracle takes as input an index of specific user identity id and runs $sk_{id} \leftarrow \mathsf{Join}(\mathsf{PP}, msk, id)$. Notice that if $(id, *) \in \mathsf{EList}$, none of the process takes place. Update the enrollment query list $\mathsf{EList} = \mathsf{EList} \cup \{(id, sk_{id})\}$ and system state $\mathsf{State} = \mathsf{State} \cup \{(id, sk_{id})\}$

$\mathsf{OSig}(\cdot, \cdot)$ The signing oracle takes as input an index of specific user identity id and a message m, returns the corresponding signature σ. Update the signing query list $\mathsf{SList} = \mathsf{SList} \cup \{(id, m, \sigma)\}$.

$\mathsf{OCorrupt}(\cdot)$ The corruption oracle takes as input a specific user identity id and retrieve the users signing key sk_{id} s.t. $(id, sk_{id}) \in \mathsf{EList}$. Return sk_{id} and update the corruption query list $\mathsf{CList} = \mathsf{CList} \cup \{id\}$.

$\mathsf{OTrace}(\cdot, \cdot)$ The tracing oracle takes as input a valid message/signature pair (m, σ), return the identity id of the real signer. Update the tracing query list $\mathsf{TList} = \mathsf{TList} \cup \{(m, \sigma)\}$.

$\mathsf{OLeak}(\cdot)$ The leakage oracle takes as input a leakage function $f(\cdot)$ and computes $\Lambda := f(\mathsf{State})$ where State contains the master signing key and all signing keys of group members. If $\mathsf{L} + \|\Lambda\| \leq \lambda$, return Λ and update $\mathsf{L} = \mathsf{L} + \|\Lambda\|$. Otherwise, return abort symbol $\perp$ and aborts.

Remark 3 We assume all the identities submitted to signing oracle and corruption oracle have been enrolled in enrollment oracle and therefore their signing is already stored in enrollment list EList.

Full Anonymity. We say a group signature satisfies *full anonymity* if any PPT verifier learns nothing about the information of the actual signers identity from a given group signature. That is, even t group members collude together, they cannot correctly identify the authentic signer with probability better than $\frac{1}{n-t}$, where n is the group size. Formally, let $\mathcal{D}$ be a distinguisher who tries to break the anonymity of $\mathcal{GS}$. Consider the following full anonymity experiment.

Full Anonymity Experiment $\mathsf{FA\text{-}Expt}_{\mathcal{D},\mathcal{GS}}$:

Setup. The challenger $\mathcal{C}$ initiates the system by running $(\mathsf{PP}, msk, tsk) \leftarrow \mathsf{Setup}(1^k)$. Public parameter PP is given to $\mathcal{D}$.

Query 1. $\mathcal{D}$ adaptively accesses to oracles $\mathsf{OEnroll}, \mathsf{OSig}, \mathsf{OCorrupt}, \mathsf{OTrace}$ and OLeak for polynomially many times.

Challenge. $\mathcal{D}$ outputs a challenge message m^* and two distinct challenge identifies id_0, id_1. $\mathcal{C}$ tosses a random coin $b \xleftarrow{\$} \{0,1\}$ and returns a challenge signature $\sigma^* \leftarrow \mathsf{GSig}(\mathsf{PP}, sk_{id_b}, m)$ to $\mathcal{D}$.

Query 2. $\mathcal{D}$ is allowed to adaptively access to oracles $\mathsf{OEnroll}, \mathsf{OSig}, \mathsf{OCorrupt}, \mathsf{OTrace}$ and OLeak for polynomially many times with restriction that $(m^*, \sigma^*) \notin \mathsf{TList}$ must hold.

Guess. Finally, $\mathcal{D}$ outputs a bit b' and wins iff. $b' = b$.

Anonymity with Bounded Leakage Resilience. We call a group signature scheme $\mathcal{IBS}$ achieves λ-*leakage resilience* and *full anonymity* if the probability that any PPT distinguisher $\mathcal{D}$ wins in the Full Anonymity Experiment $\mathsf{FA\text{-}Expt}_{\mathcal{D},\mathcal{IBS}}$ is

$$\Pr\left[b' = b : b' \leftarrow \mathcal{D}^{\mathsf{OEnroll},\mathsf{OSig},\mathsf{OCorrupt},\mathsf{OTrace},\mathsf{OLeak}}(\mathsf{PP})\right] \leq \frac{1}{2} + \mathsf{negl}(k).$$

Full Traceability. We say a group signature satisfies full traceability if no one, even collusive/compromised users and group manager, can forge a signature that is traced to any honest group members. That is, it requires that it is computationally difficult for any PPT forger to produce a valid signature on a new message on behalf of an honest group member. It is formally described in the experiment as follows.

Full Traceability Experiment $\mathsf{FT\text{-}Expt}_{\mathcal{F},\mathcal{GS}}$:

Setup. The challenger $\mathcal{C}$ initiates the system by running $(\mathsf{PP}, msk, tsk) \leftarrow \mathsf{Setup}(1^k)$. Public parameter PP is given to $\mathcal{F}$.

Query $\mathcal{F}$ adaptively accesses to oracles $\mathsf{OEnroll}, \mathsf{OSig}, \mathsf{OCorrupt}, \mathsf{OTrace}$ and OLeak for polynomially many times.

Challenge. Finally, $\mathcal{F}$ outputs a forgery $(\hat{m}, \hat{\sigma})$, which is traced to identity $\widehat{id} \leftarrow \mathsf{Trace}(tsk, \hat{m}, \hat{\sigma})$. $\mathcal{F}$ wins iff. (a) $\mathsf{GVer}(\mathsf{PP}, \hat{m}, \hat{\sigma}) = 1$, (b) $(\widehat{id}, \hat{m}, \hat{\sigma}) \notin \mathsf{SList}$, and (c) $\widehat{id} \notin \mathsf{CList}$.

$\mathcal{GS}$ achieves λ-leakage-resilient full anonymity if for any PPT forger $\mathcal{F}$, the probability that $\mathcal{F}$ wins in the Full Traceability Experiment $\mathsf{FT\text{-}Expt}_{\mathcal{F},\mathcal{GS}}$, i.e.

$$\Pr\left[\begin{array}{l} \mathsf{GVer}(\mathsf{PP}, \hat{m}, \hat{\sigma}) = 1 \\ \quad (\widehat{id}, \hat{m}, *) \notin \mathsf{SList} \quad : (\widehat{id}, \hat{m}, \hat{\sigma}) \leftarrow \mathcal{F}^{\mathsf{OEnroll},\mathsf{OSig},\mathsf{OCorrupt},\mathsf{OTrace},\mathsf{OLeak}}(\mathsf{PP}) \\ \qquad \widehat{id} \notin \mathsf{CList} \end{array}\right] \leq \mathsf{negl}(k).$$

Full Leakage Resilience. In the aforementioned full anonymity experiment and full traceability experiment, If the system state involves all signing keys of the group members as well as all randomness used in the signing process we say group signature scheme $\mathcal{GS}$ satisfies *full leakage resilience.*

4. Generic Constructions of Leakage-Resilient Group Signature

4.1. Construction 1: From Leakage-Resilient IBS and CCA-secure Encryption

Intuition. In this subsection, we present a black-box construction of leakage-resilient group signature scheme from a leakage-resilient public-key encryption scheme and a leakage-resilient IBS scheme. Let $\Pi = (\mathsf{EKg}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme, $\mathcal{IBS} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{IB\text{-}Sig}, \mathsf{IB\text{-}Ver})$ be an identity-based signature scheme, and $\mathcal{NIZK} := (l, \mathcal{P}, \mathcal{V}, \mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2))$ be an unbounded simulation-sound NIZK proof argument for $\mathcal{NP}$-language

$$L := \{(mpk, m, ek, C) : \exists (id, \sigma), \text{ s.t.} \mathsf{IB\text{-}Ver}(mpk, id, m, \sigma) = 1 \wedge C = \Pi.\mathsf{Enc}(ek, id \| \sigma)\}.$$

Put things differently, an authenticated signer possesses a valid signing key to sign any messages. Therefore, the signer is able to generate a witness (valid signature) σ on message m and a ciphertext C of the combination of his/her identity id and the signature σ. Consider the following group signature scheme $\sigma_{\mathsf{G}} = (\mathsf{Setup}, \mathsf{Join}, \mathsf{GSig}, \mathsf{GVer}, \mathsf{Trace})$.

Construction 1

$\mathsf{Setup}(1^k)$. Given the security parameter 1^k, the system is initialized as follows.

1. Invoke $(mpk, msk) \leftarrow \mathcal{IBS}.\mathsf{Setup}(1^k)$ and $(ek, dk) \leftarrow \Pi.\mathsf{EKg}(1^k)$.
2. Randomly pick a random string $r \xleftarrow{\$} \{0,1\}^{l(k)}$.
3. Output the public parameter $\mathsf{PP} := (mpk, ek, r)$, the master key $\mathsf{msk} := msk$ and the tracing key $\mathsf{tsk} := dk$.

$\mathsf{Join}(\mathsf{PP}, msk, id)$. Given the public parameter PP, the master key msk and a user identity id, return the corresponding user signing key $sk_{id} \leftarrow \mathcal{IBS}.\mathsf{Extract}(\mathsf{msk}, id)$ via secure channel.

$\mathsf{GSig}(\mathsf{PP}, sk_{id}, m)$. Given a public parameter PP, a signing key sk_{id} and a message m, compute a group signature as follows.

1. Sign message m by invoking $\sigma \leftarrow \mathcal{IBS}.\mathsf{Sig}(sk_{id}, m)$.
2. Encrypt identity id and signature σ, i.e. $C := \pi.\mathsf{Enc}(ek, id \| \sigma)$.
3. Generate the proof $\pi \leftarrow \mathcal{P}((mpk, m, ek, C), (id, \sigma); r)$.
4. Output the signature $\sigma_{\mathsf{G}} := (C, \pi)$.

$\mathsf{GVer}(\mathsf{PP}, m, \sigma_{\mathsf{G}})$. Given the public parameter PP, a message m and a purported group signature $\sigma_{\mathsf{G}} := (C, \pi)$, the verification algorithm outputs $\mathcal{V}((mpk, m, ek, C), \pi; r)$.

$\mathsf{Trace}(\mathsf{PP}, \mathsf{tsk}, m, \sigma)$. Given the public parameter PP, a tracing key tsk, message m and signature σ_{G}, if $\mathsf{GVer}(\mathsf{PP}, m, \sigma_{\mathsf{G}}) = 0$, return $\perp$. Otherwise, compute $(id \| \sigma) := \Pi.\mathsf{Dec}(\mathsf{tsk}, C)$. Return the corresponding binding identity id.

Security Analysis..

Theorem 1 If Π is CCA-secure and λ_1-leakage-resilient, $\mathcal{IBS}$ is UF-CMA secure with λ_2-bounded leakage resilience, and $\mathcal{NIZK}$ is an unbounded simulation-sound NIZK proof argument as described above, then group signature scheme Σ_{G} satisfies *full anonymity* and *full traceability* with λ-bounded leakage resilience where $\lambda = \min(\lambda_1, \lambda_2)$.

Proof (of Theorem 1). First we prove the full anonymity of Σ_{G}. Consider the following hopping experiments and denote the probability that $\mathcal{D}$ outputs 0 in Full Anonymity Experiment i by p_i.

$\mathsf{FA\text{-}Expt}_0$. It is exactly the same as the full anonymity experiment in Sect. 3.3 with $b = 0$. $C \leftarrow \Pi.\mathsf{Enc}(ek, id_0 \| \sigma_0)$ where $\sigma_0 \leftarrow \Sigma.\mathsf{Sig}(sk_{id_0}, m^*)$. Then we have

$$p_0 = \Pr[0 \leftarrow \mathcal{D}^{\mathsf{OEnroll},\mathsf{OSig},\mathsf{OCorrupt},\mathsf{OTrace},\mathsf{OLeak}}(\mathsf{PP}) | b = 0]. \qquad (1)$$

$\mathsf{FA\text{-}Expt}_1$. It is the same as Experiment 0 except that the common reference string r of the $\mathcal{NIZK}$ is generated by running $(r, \tau) \leftarrow \mathcal{S}_1(1^k)$. Furthermore, to sign message m, the proof π is now generated as

$\pi \leftarrow \mathcal{S}_2((mpk, m, ek, C), \tau)$. Following the (unbounded) zero-knowledge property of $\mathcal{NIZK}$, we have

$$|p_1 - p_0| \leq \mathsf{negl}(k). \tag{2}$$

$\mathsf{FA\text{-}Expt}_2$. It is the same as Experiment 1 except that in the challenge phase, to generate the challenge signature, compute $C \leftarrow \Pi.\mathsf{Enc}(ek, id_1 \| \sigma_1)$ and $\sigma_1 \leftarrow \Sigma.\mathsf{Sig}(sk_{id_1}, m^*)$. Then compute π same as in Experiment 1. CCA-security of the encryption scheme implies that

$$|p_2 - p_1| \leq \mathsf{negl}(k). \tag{3}$$

$\mathsf{FA\text{-}Expt}_3$. It is the same as Experiment 2 except that in the challenge phase, we sample the random common string $r \overset{\$}{\leftarrow} \{0,1\}^l$ instead of invoking simulation algorithm $\mathcal{S}$. Then, to sign a quired message m, generate C same as in Experiment 2, but compute π by running $\mathcal{P}((vk, m, ek, C), (id_1, \sigma_1); r)$. From the zero-knowledge property of $\mathcal{NIZK}$, we have

$$|p_3 - p_2| \leq \mathsf{negl}(k). \tag{4}$$

Furthermore, this experiment is exactly the same as full anonymity experiment in Sect. 3.3 with $b = 1$. That is,

$$p_3 = \Pr[0 \leftarrow \mathcal{D}^{\mathsf{OEnroll},\mathsf{OSig},\mathsf{OCorrupt},\mathsf{OTrace},\mathsf{OLeak}}(\mathsf{PP}) | b = 1]. \tag{5}$$

From eqs. (1) to (5), we have

$$\begin{aligned} &\Pr[b' = b : b' \leftarrow \mathcal{D}^{\mathsf{OEnroll},\mathsf{OSig},\mathsf{OCorrupt},\mathsf{OTrace},\mathsf{OLeak}}(\mathsf{PP})] \\ =& p_0 \cdot \Pr[b = 0] + (1 - p_3) \cdot \Pr[b = 1] \\ =& \frac{1}{2} + (p_0 - p_3) \cdot \frac{1}{2} \\ \leq& \frac{1}{2} + \mathsf{negl}(k). \end{aligned}$$

This shows that Σ_{G} satisfies full anonymity property. Next, we prove the full traceability of Σ_{G}. Assume that there exists a PPT forger $\mathcal{F}$ that breaks the full traceability of Σ_{G} with non-negligible probability, then we can construct another PPT forger $\mathcal{F}'$ that breaks the unforgeability of $\mathcal{IBS}$ with non-negligible probability as well. Let ϵ be the probability of $\mathcal{F}'$ wins in the UF-CMIA Experiment $\mathsf{CMIA\text{-}IDA}_{\mathcal{F},\mathcal{IBS}}(k)$. Details are as follows.

Algorithm $\mathcal{F}'$. Given a master public key mpk^* along with a user creation oracle $\mathcal{CO}(\cdot)$, a extraction oracle $\mathcal{EO}(\cdot)$, a signing oracle $\mathcal{SO}(\cdot,\cdot)$ and a leakage oracle $\mathcal{LO}(\cdot)$, $\mathcal{F}'$ does as follows.

Setup. Sample random common reference string $r \overset{\$}{\leftarrow} \{0,1\}^{l}(k)$ and set the leakage amount $\mathsf{L} = 0$. Invoke $(ek, dk) \leftarrow \Pi.\mathsf{EKg}(1^k)$. Set the public parameter $\mathsf{PP} := (mpk^*, ek, r)$, the master key $\mathsf{msk} :=\perp$ and the tracing key $\mathsf{tsk} := dk$. The public parameter PP is given to $\mathcal{F}$. Initially set $\mathsf{EList} = \mathsf{SList} = \mathsf{CList} = \phi$, $\mathsf{L} = 0$ and $\mathsf{State} := \{\mathsf{msk} = \perp, \mathsf{tsk} = dk\}$.

Query. Answer queries from $\mathcal{F}$ as follows.

$\mathsf{OEnroll}$. On input an index of specific user identity id, generate a user signing key $sk_{id} \leftarrow \mathcal{EO}(id)$. Update the enrollment query list $\mathsf{EList} = \mathsf{EList} \cup \{(id, sk_{id})\}$ and the system state $\mathsf{State} := \mathsf{State} \cup \{sk_{id}\}$.

OSig. On input a specific user identity id and a message m, retrieve the enrollment list to obtain the user signing key sk_{id} s.t. $(id, sk_{id}) \in \mathsf{EList}$. Compute $\sigma \leftarrow \mathsf{Sig}(sk_{id}, m)$, $C := \Pi.\mathsf{Enc}(ek, id \| \sigma)$ and $\pi \leftarrow \mathcal{P}((mpk^*, m, ek, C), (id, \sigma); r)$. Update the signing query list $\mathsf{SList} = \mathsf{SList} \cup \{(id, m, \sigma)\}$.

$\mathsf{OCorrupt}$. On input a user identity id, return sk_{id} and update the corruption query list $\mathsf{CList} = \mathsf{CList} \cup \{id\}$.

OTrace. On input a valid message/signature pair $(m, (C, \pi))$, if $\mathsf{GVer}(m, (C, \pi)) = 0$, output $\perp$. Otherwise, obtain $id \| \sigma \leftarrow \Pi.\mathsf{Dec}(\mathsf{tsk}, C)$ and return id. Update the tracing query list $\mathsf{TList} = \mathsf{TList} \cup \{(id, m, (C, \pi))\}$.

OLeak. On input a leakage function $f(\cdot)$, construct another equivalent leakage function f' that embeds all terms of $\mathsf{State} \setminus \{\mathsf{msk}\}$ and issues a query f' to $\mathcal{LO}(\cdot)$. Return $\Lambda \leftarrow \mathcal{LO}(f')$ and update $\mathsf{L} = \mathsf{L} + \Lambda$.

Challenge. Finally, $\mathcal{F}$ outputs a forgery $(\hat{m}, (\hat{C}, \hat{\pi}))$. $\mathcal{F}'$ opens $\hat{C}$ by running $(\hat{id}, \hat{\sigma}) \leftarrow \Pi.\mathsf{Dec}(dk, \hat{C})$, and outputs $(\hat{id}, \hat{m}, \hat{\sigma})$.

Let Succ be the event that the forgery $(\hat{C}, \hat{\pi})$ is a valid signature of message $\hat{m}$. This implies that (a) $(\hat{id}, \hat{m}, *) \notin \mathsf{SList}$, (b) $\mathcal{V}((mpk^*, \hat{m}, ek, \hat{C}), \hat{\pi}; r)) = 1$, and (c) $\hat{id} \notin \mathsf{CList}$. Let Ext be the event that Succ occurs and furthermore, $\mathsf{GVer}(mpk^*, \hat{id}, \hat{m}, \hat{\sigma}) = 1$ hold. Unbounded simulation soundness of the NIZK proof system implies that $|\Pr[\mathsf{Succ}] - \Pr[\mathsf{Ext}]| \leq \mathsf{negl}(k)$. Then we have

$$\begin{aligned}
\Pr[\mathsf{Succ}] &= (1 - \mathsf{negl}(k)) \cdot \Pr[\mathsf{Ext}] \\
&= (1 - \mathsf{negl}(k)) \cdot \Pr[\mathsf{GVer}(mpk^*, \hat{id}, \hat{m}, \hat{\sigma}) = 1 \wedge \mathcal{O}(\hat{id}, \hat{m}) \text{ is not quired} \wedge \mathcal{L}(\hat{id}) \text{ is not quired}] \\
&= (1 - \mathsf{negl}(k))\epsilon \\
&= \mathsf{negl}'(k)
\end{aligned}$$

which results from the fact that $\mathcal{IBS}$ is λ_2-leakage-resilient. □

4.2. Construction 2: From Leakage-Resilient Signature and CCA-secure Encryption

Let $\Pi = (\mathsf{EKg}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme, $\Sigma = (\mathsf{Kg}, \mathsf{Sig}, \mathsf{Ver})$ be a standard existentially unforgeable signature scheme, and $\mathcal{NIZK} := (l, \mathcal{P}, \mathcal{V}, \mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2))$ be an unbounded simulation-sound NIZK proof argument for $\mathcal{NP}$-language

$$L := \left\{ (vk, m, ek, C) : \exists (uvk, Cert, \sigma), \text{ s.t. } \begin{array}{r} \mathsf{Ver}(vk, uvk, Cert) = 1 \wedge \mathsf{Ver}(uvk, m, \sigma) = 1 \\ \wedge C = \Pi.\mathsf{Enc}(ek, uvk \| Cert \| m \| \sigma) \end{array} \right\}.$$

Consider the following group signature scheme $\sigma_{\mathsf{G}} = (\mathsf{Setup}, \mathsf{Join}, \mathsf{GSig}, \mathsf{GVer}, \mathsf{Trace})$.

Construction 2

$\mathsf{Setup}(1^k)$. Given the security parameter 1^k, the system is initialized as follows.

1. Invoke $(vk, sk) \leftarrow \Sigma.\mathsf{Kg}(1^k)$.
2. $(ek, dk) \leftarrow \Pi.\mathsf{EKg}(1^k)$.
3. Randomly pick a random string $r \xleftarrow{\$} \{0,1\}^{l(k)}$.
4. Output the public parameter $\mathsf{PP} := (vk, ek, r)$, the master key $\mathsf{msk} := sk$ and the tracing key $\mathsf{tsk} := dk$.

$\mathsf{Join}(\mathsf{PP}, msk, id)$.

1. User with identity uid generates a verification/signing key pair by invoking $(uvk, usk) \leftarrow \Sigma.\mathsf{Kg}(1^k)$ and submit $id = (uid, uvk)$ to group manager.
2. Group manager signs $uid\|uvk$ with its master key msk, i.e. $Cert \leftarrow \Sigma.\mathsf{Sig}(\mathsf{msk}, uid\|uvk)$.
3. Return $Cert$ to user via public channel.
4. User sets the signing key $sk_{id} := usk$ along with the group certificate signed by group manager.

$\mathsf{GSig}(\mathsf{PP}, sk_{id}, m)$. Given a public parameter PP, a signing key sk_{id} along with a group certificate $Cert$ and a message m, compute a group signature as follows.

1. Compute $\sigma \leftarrow \Sigma.Sig(sk_{id}, m)$.
2. Compute $C := \pi.\mathsf{Enc}(ek, uvk\|Cert\|m\|\sigma)$.
3. Generate the proof $\pi \leftarrow \mathcal{P}((vk, m, ek, C), (uvk, Cert, \sigma); r)$.
4. Output the signature $\sigma_{\mathsf{G}} := (C, \pi)$.

$\mathsf{GVer}(\mathsf{PP}, m, \sigma_{\mathsf{G}})$. Given a public parameter PP, a message m and a purported group signature $\sigma_{\mathsf{G}} := (C, \pi)$, the verification algorithm outputs $\mathcal{V}((vk, m, ek, C), \pi; r)$.

$\mathsf{Trace}(\mathsf{PP}, \mathsf{tsk}, m, \sigma)$. Given the public parameter PP, a tracing key tsk, message m and signature σ_{G}, if $\mathsf{GVer}(\mathsf{PP}, m, \sigma_{\mathsf{G}}) = 0$, return $\perp$. Otherwise, compute $(uvk\|Cert\|m\|\sigma) \leftarrow \Pi.\mathsf{Dec}(\mathsf{tsk}, C)$. Return the corresponding binding identity id.

Theorem 2 If Π is CCA-secure and λ_1-leakage-resilient, Σ is UF-CMA secure with λ_2-bounded leakage resilience, and $\mathcal{NIZK}$ is an unbounded simulation-sound NIZK proof argument as described above, then group signature scheme Σ_{G} satisfies *full anonymity* and *full traceability* with λ-bounded leakage resilience where $\lambda = \min(\lambda_1, \lambda_2)$.

Proof (of Theorem 2). First we prove the full anonymity of Σ_{G}. Consider the following hopping experiments and denote the probability that $\mathcal{D}$ outputs 0 in Full Anonymity Experiment i by p_i.

$\mathsf{FA\text{-}Expt}_0$. It is exactly the same as the full anonymity experiment in Sect. 3.3 with $b = 0$. That is,

$C \leftarrow \Pi.\mathsf{Enc}(ek, uvk_0 \| Cert_0 \| m^* \| \sigma_0)$ where $Cert_0 \leftarrow \Sigma.\mathsf{Kg}(\mathsf{msk}, id_0)$ and $\sigma_0 \leftarrow \Sigma.\mathsf{Sig}(usk_0, m^*)$. Then we have

$$p_0 = \Pr[0 \leftarrow \mathcal{D}^{\mathsf{OEnroll},\mathsf{OSig},\mathsf{OCorrupt},\mathsf{OTrace},\mathsf{OLeak}}(\mathsf{PP}) | b = 0]. \tag{6}$$

$\mathsf{FA\text{-}Expt}_1$. It is the same as Experiment 0 except that the common reference string r of the $\mathcal{NIZK}$ is generated by running $(r, \tau) \leftarrow \mathcal{S}_1(1^k)$. Furthermore, to sign message m^*, the proof π is now generated as $\pi \leftarrow \mathcal{S}_2((vk, m^*, ek, C), \tau)$. Following the (unbounded) zero-knowledge property of $\mathcal{NIZK}$, we have

$$|p_1 - p_0| \leq \mathsf{negl}(k). \tag{7}$$

$\mathsf{FA\text{-}Expt}_2$. It is the same as Experiment 1 except that in the challenge phase, to generate the challenge signature, compute $C \leftarrow \Pi.\mathsf{Enc}(ek, uvk_1 \| Cert_1 \| m^* \| \sigma_1)$ where $Cert_1 \leftarrow \Sigma.\mathsf{Kg}(\mathsf{msk}, id_1)$ and $\sigma_1 \leftarrow \Sigma.\mathsf{Sig}(usk_1, m^*)$. Then compute π same as in Experiment 1. CCA-security of the encryption scheme implies that

$$|p_2 - p_1| \leq \mathsf{negl}(k). \tag{8}$$

$\mathsf{FA\text{-}Expt}_3$. It is the same as Experiment 2 except that in the challenge phase, we sample the random common string $r \overset{\$}{\leftarrow} \{0,1\}^l$ instead of invoking simulation algorithm $\mathcal{S}$. Then, to sign a quired message m, generate C same as in Experiment 2, but compute π by running $\mathcal{P}((vk, m, ek, C), (uvk, Cert, \sigma); r)$. From the zero-knowledge property of $\mathcal{NIZK}$, we have

$$|p_3 - p_2| \leq \mathsf{negl}(k). \tag{9}$$

Furthermore, this experiment is exactly the same as full anonymity experiment in Sect. 3.3 with $b = 1$. That is,

$$p_3 = \Pr[0 \leftarrow \mathcal{D}^{\mathsf{OEnroll},\mathsf{OSig},\mathsf{OCorrupt},\mathsf{OTrace},\mathsf{OLeak}}(\mathsf{PP}) | b = 1]. \tag{10}$$

From eqs. (6) to (10), we have

$$\begin{aligned} &\Pr[b' = b : b' \leftarrow \mathcal{D}^{\mathsf{OEnroll},\mathsf{OSig},\mathsf{OCorrupt},\mathsf{OTrace},\mathsf{OLeak}}(\mathsf{PP})] \\ =& p_0 \cdot \Pr[b = 0] + (1 - p_3) \cdot \Pr[b = 1] \\ =& \frac{1}{2} + (p_0 - p_3) \cdot \frac{1}{2} \\ \leq& \frac{1}{2} + \mathsf{negl}(k). \end{aligned}$$

Therefore, Σ_{G} satisfies full anonymity property. Next, we prove the full traceability of Σ_{G}. Assume that there exists a PPT forger $\mathcal{F}$ that breaks the full traceability of Σ_{G} with non-negligible probability, then we can construct another PPT forger $\mathcal{F}'$ that breaks the unforgeability of Σ with non-negligible probability as well. We consider following two types of forgers, Type-I forger $\mathcal{F}'_{\text{I}}$ and Type-II forger $\mathcal{F}'_{\text{II}}$ who succeed in forging a group signature.

Type-I. The valid signature is traced to a signer id^* who is not a group member. That is, $(id^*, *) \notin EList$, which implies that $\mathcal{F}'_{\text{I}}$ succeeds in forging a new group certificate (signature) on id^*, which breaks the unforgeability of Σ on master signing key msk.

Type-II. The valid signature is traced to an honest signer id^*. In other words, the signing key $\mathsf{OCorrupt}(id^*)$ is not quired before and $\mathcal{F}'_{\text{II}}$ succeeds in forging a signature σ^* on new message m^*, which breaks the unforgeability of Σ on user signing key usk.

Let ϵ_1, ϵ_2 respectively be the probabilities of $\mathcal{F}'_{\text{I}}$ and $\mathcal{F}'_{\text{II}}$ wins in the UF-CMIA Experiment. Details are as follows.

Algorithm $\mathcal{F}'_{\text{I}}$. Given a verification key vk^* along with a signing oracle $\mathcal{O}_s(\cdot)$ and a leakage oracle $\mathcal{O}_L(\cdot)$, $\mathcal{F}'_{\text{I}}$ does as follows.

Setup. Sample random common reference string $r \xleftarrow{\$} \{0,1\}^l(k)$ and set the leakage amount $\mathsf{L} = 0$. Invoke $(ek, dk) \leftarrow \Pi.\mathsf{EKg}(1^k)$. Set the public parameter $\mathsf{PP} := (vk^*, ek, r)$, the master key $\mathsf{msk} :=\perp$ and the tracing key $\mathsf{tsk} := dk$. The public parameter PP is given to $\mathcal{F}$. Initially set $\mathsf{EList} = \mathsf{SList} = \mathsf{CList} = \phi$, $\mathsf{L} = 0$ and $\mathsf{State} := \{\mathsf{msk} =\perp, \mathsf{tsk} = dk\}$.

Query. Answer queries from $\mathcal{F}$ as follows.

$\mathsf{OEnroll}$. On input an index of specific user identity uid, generate a user signing key by running $(usk, uvk) \leftarrow \Sigma.\mathsf{Kg}(1^k)$ and obtain $Cert \leftarrow \mathcal{O}_s(uid \| uvk)$. Update the enrollment query list $\mathsf{EList} = \mathsf{EList} \cup \{(uid, uvk, usk, Cert)\}$ and the system state $\mathsf{State} := \mathsf{State} \cup \{usk\}$.

OSig. On input a specific user identity $id := (uid, uvk)$ and a message m, retrieve the enrollment list to obtain the user signing key $sk_{id} := (usk, Cert)$ s.t. $(uid, uvk, usk, Cert) \in \mathsf{EList}$. Compute $\sigma \leftarrow \mathsf{Sig}(usk, m)$, $C := \Pi.\mathsf{Enc}(ek, uvk \| Cert \| m \| \sigma)$ and $\pi \leftarrow \mathcal{P}((vk^*, m, ek, C), (uvk, Cert, \sigma); r)$. Update the signing query list $\mathsf{SList} = \mathsf{SList} \cup \{(id := (uid, uvk), m, \sigma)\}$.

$\mathsf{OCorrupt}$. On input a user identity id, return sk_{id} and update the corruption query list $\mathsf{CList} = \mathsf{CList} \cup \{id\}$.

OTrace. On input a valid message/signature pair $(m, (C, \pi))$, if $\mathsf{GVer}(m, (C, \pi)) = 0$, output $\perp$. Otherwise, obtain $uvk \| Cert \| m \| \sigma \leftarrow \Pi.\mathsf{Dec}(\mathsf{tsk}, C)$, retrieve enrollment list to find uid s.t. $(uid, usk, *, *) \in \mathsf{EList}$ and return uid. Update the tracing query list $\mathsf{TList} = \mathsf{TList} \cup \{(id, m, (C, \pi))\}$.

OLeak. On input a leakage function $f(\cdot)$, construct another equivalent leakage function f' that embeds all terms of $\mathsf{State} \setminus \{\mathsf{msk}\}$ and issues a query f' to $\mathcal{LO}(\cdot)$. Return $\Lambda \leftarrow \mathcal{LO}(f')$ and update $\mathsf{L} = \mathsf{L} + \Lambda$.

Challenge. Finally, $\mathcal{F}$ outputs a forgery $(\hat{m}, (\hat{C}, \hat{\pi}))$. $\mathcal{F}'_{\text{I}}$ opens $\hat{C}$ by running $(\hat{id}, \hat{\sigma}) \leftarrow \Pi.\mathsf{Dec}(dk, \hat{C})$, and outputs $(\hat{uvk}, \hat{Cert})$.

Let Succ be the event that the forgery $(\hat{C}, \hat{\pi})$ is a valid signature of message $\hat{m}$. Again, forger $\mathcal{F}$ is able to produce a valid signature on a new message that is traced to a non-existent group member. This implies that (a) $(*, \hat{uvk}) \notin \mathsf{EList}$, (b) $((*, \hat{uvk}), \hat{m}, *) \notin \mathsf{SList}$, (c) $\mathcal{V}((vk^*, \hat{m}, ek, \hat{C}), \hat{\pi}; r)) = 1$, and (d) $\hat{uvk} \notin \mathsf{CList}$. Let Ext be the event that Succ occurs and furthermore, $\mathsf{Ver}(vk^*, \hat{uvk}, \hat{m}, \hat{Cert}) = 1$ and $\mathsf{Ver}(\hat{uvk}, \hat{m}, \hat{\Sigma}) = 1$ hold. Unbounded simulation soundness of the NIZK proof system implies that $|\Pr[\mathsf{Succ}] - \Pr[\mathsf{Ext}]| \leq \mathsf{negl}(k)$. Then we have

$$\begin{aligned}
\Pr[\mathsf{Succ}] &= (1 - \mathsf{negl}(k)) \cdot \Pr[\mathsf{Ext}] \\
&= (1 - \mathsf{negl}(k)) \cdot \Pr[\mathsf{Ver}(vk^*, \hat{m}, \hat{Cert}) = 1 \wedge \mathcal{O}_s(\hat{uvk}) \text{ is not quired}] \\
&= (1 - \mathsf{negl}(k))\epsilon_1 \\
&= \mathsf{negl}'(k)
\end{aligned}$$

which results from the fact that Σ is λ_2-leakage-resilient. Next, we consider forger $\mathcal{F}'_{\mathrm{II}}$ as follows.

Algorithm $\mathcal{F}'_{\mathrm{II}}$. Given a verification key vk^* along with a signing oracle $\mathcal{O}_s(\cdot)$ and a leakage oracle $\mathcal{O}_L(\cdot)$, $\mathcal{F}'_{\mathrm{II}}$ does as follows.

Setup. Sample random common reference string $r \xleftarrow{\$} \{0,1\}^l(k)$ and set the leakage amount $\mathsf{L} = 0$. Invoke $(vk, sk) \leftarrow \Sigma.\mathsf{Kg}(1^k)$ and $(ek, dk) \leftarrow \Pi.\mathsf{EKg}(1^k)$. Set the public parameter $\mathsf{PP} := (vk, ek, r)$, the master key $\mathsf{msk} := sk$ and the tracing key $\mathsf{tsk} := dk$. The public parameter PP is given to $\mathcal{F}$. Randomly pick $i^* \leftarrow [n]$ where $n := n(k)$ is the group size. Initially set $\mathsf{EList} = \mathsf{SList} = \mathsf{CList} = \phi$, $\mathsf{L} = 0$ and $\mathsf{State} := \{\mathsf{msk} = sk, \mathsf{tsk} = dk\}$.

Query. Answer queries from $\mathcal{F}$ as follows.

$\mathsf{OEnroll}$. Denote the ith user identity by uid_i. Obtain a verification/signing key pair

$$(uvk_i, usk_i) := \begin{cases} \Sigma.\mathsf{Kg}(1^k) & i \neq i^* \\ (vk^*, \perp) & i = i^* \end{cases}$$

and generate the group certificate by running $Cert \leftarrow \Sigma.\mathsf{Sig}(\mathsf{msk}, uid_i \| uvk_i)$. Update the enrollment query list $\mathsf{EList} = \mathsf{EList} \cup \{(uid_i, uvk_i, usk_i, Cert_i)\}$ and the system state $\mathsf{State} := \mathsf{State} \cup \{usk_i\}$. Notice that usk_{i^*} is unknown.

OSig. On input a specific user identity $id := (uid, uvk)$ and a message m, retrieve the enrollment list to obtain the user signing key $sk_{id} := (usk, Cert)$ s.t. $(uid, uvk, usk, Cert) \in \mathsf{EList}$. Compute $C := \Pi.\mathsf{Enc}(ek, uvk \| Cert \| m \| \sigma)$ and $\pi \leftarrow \mathcal{P}((vk, m, ek, C), (uvk, Cert, \sigma); r)$ where

$$\sigma := \begin{cases} \Sigma.\mathsf{Sig}(usk, m) & uvk \neq vk^* \\ \mathcal{O}_s(m) & uvk = vk^* \end{cases}$$

Update the signing query list $\mathsf{SList} = \mathsf{SList} \cup \{(id := (uid, uvk), m, \sigma)\}$.

$\mathsf{OCorrupt}$. On input a user identity $id := (uid, uvk)$, if $uvk = vk^*$, abort. Otherwise, return sk_{id} and update the corruption query list $\mathsf{CList} = \mathsf{CList} \cup \{id\}$.

OTrace. On input a valid message/signature pair $(m, (C, \pi))$, if $\mathsf{GVer}(m, (C, \pi)) = 0$, output $\perp$. Otherwise, obtain $uvk\|Cert\|m\|\sigma \leftarrow \Pi.\mathsf{Dec}(\mathsf{tsk}, C)$, retrieve enrollment list to find uid s.t. $(uid, usk, *, *) \in \mathsf{EList}$ and return uid. Update the tracing query list $\mathsf{TList} = \mathsf{TList} \cup \{(id, m, (C, \pi))\}$.

OLeak. On input a leakage function $f(\cdot)$, construct another equivalent leakage function f' that embeds all terms of $\mathsf{State} \setminus \{usk_{i^*}\}$ and issues a query f' to $\mathcal{LO}(\cdot)$. Return $\Lambda \leftarrow \mathcal{LO}(f')$ and update $\mathsf{L} = \mathsf{L} + \Lambda$.

Challenge. Finally, $\mathcal{F}$ outputs a forgery $(\hat{m}, (\hat{C}, \hat{\pi}))$. $\mathcal{F}'_{\text{II}}$ opens $\hat{C}$ by running $(\hat{uvk}\|\hat{Cert}\|\hat{m}\|\hat{\sigma}) \leftarrow \Pi.\mathsf{Dec}(dk, \hat{C})$, and outputs $(\hat{m}, \hat{\sigma})$.

Let Succ be the event that the forgery $(\hat{C}, \hat{\pi})$ is a valid signature of message $\hat{m}$. Again, forger $\mathcal{F}$ is able to produce a valid signature on a new message that is traced to an honest group member. This implies that (a) $(*, \hat{uvk}) \notin \mathsf{EList}$, (b) $((*, \hat{uvk}), \hat{m}, *) \notin \mathsf{SList}$, (c) $\mathcal{V}((vk, \hat{m}, ek, \hat{C}), \hat{\pi}; r)) = 1$, and (d) $\hat{uvk} \notin \mathsf{CList}$. Let Ext be the event that Succ occurs and furthermore, $\mathsf{Ver}(vk, \hat{uvk}, \hat{m}, \hat{Cert}) = 1$ and $\mathsf{Ver}(\hat{uvk}, \hat{m}, \hat{\Sigma}) = 1$ hold. Unbounded simulation soundness of the NIZK proof system implies that $|\Pr[\mathsf{Succ}] - \Pr[\mathsf{Ext}]| \leq \mathsf{negl}(k)$. Then we have

$$\begin{aligned}
\Pr[\mathsf{Succ}] &= (1 - \mathsf{negl}(k)) \cdot \Pr[\mathsf{Ext}] \\
&\leq (1 - \mathsf{negl}(k)) \cdot \Pr[\mathsf{Ver}(\hat{uvk}, \hat{m}, \hat{Cert}) = 1 \wedge \mathcal{O}_s(\hat{m}) \text{ is not quired}] \\
&\leq (1 - \mathsf{negl}(k)) \cdot \Pr[\mathsf{Ver}(\hat{vk^*}, \hat{m}, \hat{Cert}) = 1 \wedge \mathcal{O}_s(\hat{m}) \text{ is not quired}] \cdot \Pr[vk^* = \hat{uvk}] \\
&= (1 - \mathsf{negl}(k))\epsilon_2 \cdot \frac{1}{n} \\
&= \mathsf{negl}'(k),
\end{aligned}$$

which results from the fact that Σ is λ_2-leakage-resilient. □

4.3. Construction 3: From Leakage-Resilient Signature and CPA-Secure Encryption

Let $\Pi = (\mathsf{EKg}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme, $\Sigma = (\mathsf{Kg}, \mathsf{Sig}, \mathsf{Ver})$ be a standard existentially unforgeable signature scheme, and $\mathcal{NIZK} := (l, \mathcal{P}, \mathcal{V}, \mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2))$ be an unbounded simulation-sound NIZK proof argument for $\mathcal{NP}$-language

$$L := \left\{ (vk, m, ek_1, ek_2, C_1, C_2) : \exists (uvk, Cert, \sigma, \omega_1, \omega_2), \text{ s.t. } \begin{array}{r} \mathsf{Ver}(vk, uvk, Cert) = 1 \wedge \mathsf{Ver}(uvk, m, \sigma) = 1 \\ \wedge C_1 = \Pi.\mathsf{Enc}(ek_1, uvk\|Cert\|m\|\sigma; \omega_1) \\ \wedge C_2 = \Pi.\mathsf{Enc}(ek_2, uvk\|Cert\|m\|\sigma; \omega_2) \end{array} \right\}.$$

Consider the following group signature scheme $\sigma_{\mathsf{G}} = (\mathsf{Setup}, \mathsf{Join}, \mathsf{GSig}, \mathsf{GVer}, \mathsf{Trace})$.

Construction 3

$\mathsf{Setup}(1^k)$. Given the security parameter 1^k, the system is initialized as follows.

1. Invoke $(vk, sk) \leftarrow \Sigma.\mathsf{Kg}(1^k)$.
2. $(ek_1, dk_1) \leftarrow \Pi.\mathsf{EKg}(1^k)$ and $(ek_2, dk_2) \leftarrow \Pi.\mathsf{EKg}(1^k)$.
3. Randomly pick a random string $r \overset{\$}{\leftarrow} \{0,1\}^{l(k)}$.
4. Output the public parameter $\mathsf{PP} := (vk, ek_1, ek_2, r)$, the master key $\mathsf{msk} := sk$ and the tracing key $\mathsf{tsk} := dk_1$.

$\mathsf{Join}(\mathsf{PP}, msk, id)$.

1. User with identity uid generates a verification/signing key pair by invoking $(uvk, usk) \leftarrow \Sigma.\mathsf{Kg}(1^k)$ and submit $id = (uid, uvk)$ to group manager.
2. Group manager signs $uid\|uvk$ with its master key msk, i.e. $Cert \leftarrow \Sigma.\mathsf{Sig}(\mathsf{msk}, uid\|uvk)$.
3. Return $Cert$ to user via public channel.
4. User sets the signing key $sk_{id} := usk$ along with the group certificate signed by group manager.

$\mathsf{GSig}(\mathsf{PP}, sk_{id}, m)$. Given a public parameter PP, a signing key sk_{id} along with a group certificate $Cert$ and a message m, compute a group signature as follows.

1. Compute $\sigma \leftarrow \Sigma.Sig(sk_{id}, m)$.
2. Compute $C_1 := \pi.\mathsf{Enc}(ek_1, uvk\|Cert\|m\|\sigma; \omega_1)$ and $C_2 := \pi.\mathsf{Enc}(ek_2, uvk\|Cert\|m\|\sigma; \omega_1)$.
3. Generate the proof $\pi \leftarrow \mathcal{P}((vk, m, ek_1, ek_2, C_1, C_2), (uvk, Cert, \sigma, \omega_1, \omega_2); r)$.
4. Output the signature $\sigma_{\mathsf{G}} := (C, \pi)$.

$\mathsf{GVer}(\mathsf{PP}, m, \sigma_{\mathsf{G}})$. Given a public parameter PP, a message m and a purported group signature $\sigma_{\mathsf{G}} := (C, \pi)$, the verification algorithm outputs $\mathcal{V}((vk, m, ek_1, ek_2, C), \pi; r)$.

$\mathsf{Trace}(\mathsf{PP}, \mathsf{tsk}, m, \sigma)$. Given the public parameter PP, a tracing key tsk, message m and signature σ_{G}, if $\mathsf{GVer}(\mathsf{PP}, m, \sigma_{\mathsf{G}}) = 0$, return $\perp$. Otherwise, compute $(uvk\|Cert\|m\|\sigma) \leftarrow \Pi.\mathsf{Dec}(\mathsf{tsk}, C)$. Return the corresponding binding identity id.

Theorem 3 If Π is CPA-secure and λ_1-leakage-resilient, Σ is UF-CMA secure with λ_2-bounded leakage resilience, and $\mathcal{NIZK}$ is an unbounded simulation-sound NIZK proof argument as described above, then group signature scheme Σ_{G} satisfies *full anonymity* and *full traceability* with λ-bounded leakage resilience where $\lambda = \min(\lambda_1, \lambda_2)$.

The proof of theorem 3 is similar to that of theorem 2 and thus we omit it here. The main difference is that the way to answer tracing queries from the adversary. We are able to correctly simulate the tracing process if the decryption oracle of CCA-secure encryption scheme is given. For a CPA-secure encryption, we generate two uniform key pairs (ek_1, dk_1) and (ek_2, dk_2) where dk_2 is erased at once in the setup process. In the simulation, the tracing key dk_1 is unknown. To answer tracing query, we do not erase dk_2 while the view of forger being not changed. Therefore, we are able to correctly to simulate the tracing query.

5. Conclusion

In this work we formalize the definition of leakage-resilient group signature scheme. Based on this model, we present three black-box constructions of group signature scheme. They are constructed from constructing from the combination of an IBS scheme and a CCA-secure PKE scheme, the combination of a signature scheme and a CCA-secure PKE scheme, the combination of a signature scheme and a CPA-secure PKE scheme, respectively. We prove that the leakage bound of resulting construction is the same as the lowest leakage bound of underlying primitives. Intuitively, the buckets effect is inescapable when the private state generated from different cryptographic primitives and the weakest part of the system is always more vulnerable.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61472146, 61872152), Guangdong Natural Science Funds for Distinguished Young Scholar (No. 2014A030306021), Guangdong Program for Special Support of Top-notch Young Professionals (No. 2015TQ01X796), and Pearl River Nova Program of Guangzhou (No. 201610010037).

References

[1] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In *Theory of Cryptography*, pages 474–495. Springer, 2009.

[2] Joël Alwen, Yevgeniy Dodis, and Daniel Wichs. Leakage-resilient public-key cryptography in the bounded-retrieval model. In *Advances in Cryptology-CRYPTO 2009*, pages 36–54. Springer, 2009.

[3] Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *Annual International Cryptology Conference*, pages 255–270. Springer, 2000.

[4] Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *International Cryptology Conference on Advances in Cryptology*, pages 255–270, 2000.

[5] Giuseppe Ateniese and Breno de Medeiros. Efficient group signatures without trapdoors. *Cryptology Eprint Archive*, 2002:173, 2002.

[6] Giuseppe Ateniese, Dawn Song, and Gene Tsudik. Quasi-efficient revocation of group signatures. In *International Conference on Financial Cryptography*, pages 183–197. Springer, 2002.

[7] Giuseppe Ateniese and Gene Tsudik. Group signatures á la carte. In *Symposium on Discrete Algorithms: Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, pages 848–849. ACM/SIAM, 1999.

[8] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 614–629. Springer, 2003.

[9] Mihir Bellare and Sara K Miner. A forward-secure digital signature scheme. In *Annual International Cryptology Conference*, pages 431–448. Springer, 1999.

[10] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *Annual International Cryptology Conference*, pages 41–55. Springer, 2004.

[11] Dan Boneh, Richard A DeMillo, and Richard J Lipton. On the importance of checking cryptographic protocols for faults. In *Advances in Cryptology-EUROCRYPT'97*, pages 37–51. Springer, 1997.

[12] Elette Boyle, Gil Segev, and Daniel Wichs. Fully leakage-resilient signatures. *Journal of cryptology*, 26(3):513–558, 2013.

[13] Zvika Brakerski, Yael Tauman Kalai, Jonathan Katz, and Vinod Vaikuntanathan. Overcoming the hole in the bucket: Public-key cryptography resilient to continual memory leakage. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 501–510. IEEE, 2010.

[14] Emmanuel Bresson and Jacques Stern. Efficient revocation in group signatures. In *International Workshop on Public Key Cryptography*, pages 190–206. Springer, 2001.

[15] Jan Camenisch and Markus Stadler. Efficient group signature schemes for large groups. In *Annual International Cryptology Conference*, pages 410–424. Springer, 1997.

[16] David Chaum and Eugène Van Heyst. Group signatures. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 257–265. Springer, 1991.

[17] L. Chen and T. P. Pedersen. New group signature schemes. In *Workshop on the Theory & Application of of Cryptographic Techniques*, 1994.

[18] Jean-Sébastien Coron. Resistance against differential power analysis for elliptic curve cryptosystems. In *Cryptographic Hardware and Embedded Systems*, pages 292–302. Springer, 1999.

[19] Alfredo De Santis, Giovanni Di Crescenzo, Rafail Ostrovsky, Giuseppe Persiano, and Amit Sahai. Robust non-interactive zero knowledge. In *Annual International Cryptology Conference*, pages 566–598. Springer, 2001.

[20] Yevgeniy Dodis, Kristiyan Haralambiev, Adriana López-Alt, and Daniel Wichs. Efficient public-key cryptography in the presence of key leakage. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 613–631. Springer, 2010.

[21] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*, pages 293–302. IEEE, 2008.

[22] Sebastian Faust, Carmit Hazay, Jesper Buus Nielsen, Peter Sebastian Nordholt, and Angela Zottarel. Signature schemes secure against hard-to-invert leakage. *Journal of Cryptology*, 29(2):422–455, 2016.

[23] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. In *Cryptographic Hardware and Embedded SystemsCHES 2001*, pages 251–261. Springer, 2001.

[24] Shafi Goldwasser, Silvio Micali, and Ronald L Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988.

[25] J Alex Halderman, Seth D Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph A Calandrino, Ariel J Feldman, Jacob Appelbaum, and Edward W Felten. Lest we remember: cold-boot attacks on encryption keys. *Communications of the ACM*, 52(5):91–98, 2009.

[26] Jianye Huang and Qiong Huang. Black-box constructions of signature schemes in the bounded leakage setting. *Information Sciences*, 423:313–325, 2018.

[27] Jianye Huang, Qiong Huang, and Chunhua Pan. A black-box construction of strongly unforgeable signature schemes in the bounded leakage model. In *Provable Security*, pages 320–339. Springer Nature, 2016.

[28] Jianye Huang, Qiong Huang, and Chunhua Pan. A black-box construction of strongly unforgeable signature scheme in the leakage setting. *Int. J. Found. Comput. Sci.*, 28(6):761–780, 2017.

[29] Jianye Huang, Qiong Huang, and Willy Susilo. Leakage-resilient dual-form signatures. *The Computer Journal*, 2018.

[30] Yael Tauman Kalai, Bhavana Kanukurthi, and Amit Sahai. Cryptography with tamperable and leaky memory. In *Annual Cryptology Conference*, pages 373–390. Springer, 2011.

[31] Jonathan Katz and Vinod Vaikuntanathan. Signature schemes with bounded leakage resilience. In *Advances in Cryptology–ASIACRYPT 2009*, pages 703–720. Springer, 2009.

[32] Aggelos Kiayias and Moti Yung. Extracting group signatures from traitor tracing schemes. In *International Conference on Advances in Cryptology-Eurocrypt*, 2003.

[33] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *Advances in CryptologyCRYPTO99*, pages 388–397. Springer, 1999.

[34] Paul C Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *Advances in Cryptology-CRYPTO'96*, pages 104–113. Springer, 1996.

[35] Yehuda Lindell and Jonathan Katz. *Introduction to modern cryptography*. Chapman and Hall/CRC, 2014.

[36] Tal Malkin, Isamu Teranishi, Yevgeniy Vahlis, and Moti Yung. Signatures resilient to continual leakage on memory and computation. In *Theory of Cryptography Conference*, pages 89–106. Springer, 2011.

[37] Silvio Micali and Leonid Reyzin. Physically observable cryptography. In *Theory of Cryptography*, pages 278–296. Springer, 2004.

[38] Tomoyoshi Ono and Kazuki Yoneyama. On randomness exposure resilience of group signatures. *IEICE TRANSACTIONS on Information and Systems*, 100(10):2357–2367, 2017.

[39] Jean-Jacques Quisquater and David Samyde. Electromagnetic analysis (ema): Measures and counter-measures for smart cards. In *Smart Card Programming and Security*, pages 200–210. Springer, 2001.

[40] Craig Ramsay. *TEMPEST attacks against AES Covertly stealing keys for €200*, 2017.

[41] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 543–553. IEEE, 1999.

[42] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Workshop on the theory and application of cryptographic techniques*, pages 47–53. Springer, 1984.

[43] Huaqun Wang, Qianhong Wu, Bo Qin, Futai Zhang, and Josep Domingo-Ferrer. A provably secure ring signature scheme with bounded leakage resilience. In *International Conference on Information Security Practice and Experience*, pages 388–402. Springer, 2014.

[44] Yuyu Wang and Keisuke Tanaka. Generic transformations for existentially unforgeable signature schemes in the bounded leakage model. *Security and Communication Networks*, 9(12):1829–1842, 2016.

[45] Tsz Hon Yuen, Siu Ming Yiu, and Lucas CK Hui. Fully leakage-resilient signatures with auxiliary inputs. In *Australasian Conference on Information Security and Privacy*, pages 294–307. Springer, 2012.