Group Signatures with Message-Dependent Opening

Yusuke Sakai[1,⋆], Keita Emura[2,⋆⋆], Goichiro Hanaoka[3], Yutaka Kawai[4,⋆⋆⋆], Takahiro Matsuda[3,†], and Kazumasa Omote[5]

[1] The University of Electro-Communications, Japan
yusuke.sakai@uec.ac.jp
[2] National Institute of Information and Communications Technology, Japan
[3] National Institute of Advanced Industrial Science and Technology, Japan
[4] Mitsubishi Electric, Japan
[5] Japan Advanced Institute of Science and Technology, Japan

Abstract. This paper introduces a new capability of the group signature, called *message-dependent opening*. It is intended to weaken the higher trust put on an opener, that is, no anonymity against an opener is provided by ordinary group signature. In a group signature system with message-dependent opening (GS-MDO), in addition to the opener, we set up the *admitter* which is not able to open any user's identity but *admits* the opener to open signatures by specifying messages whose signatures should be opened. For any signature whose corresponding message is not specified by the admitter, the opener cannot extract the signer's identity from it. In this paper, we present formal definitions and constructions of GS-MDO. Furthermore, we also show that GS-MDO implies identity-based encryption, and thus for designing a GS-MDO scheme, identity-based encryption is crucial. Actually, we propose a generic construction of GS-MDO from identity-based encryption and adaptive NIZK proofs, and its specific instantiation from the Groth-Sahai proof system by constructing a new (k-resilient) identity-based encryption scheme which is compatible to the Groth-Sahai proof.

1 Introduction

Group signature [20] is a kind of anonymous signatures, which allows members of a group to sign a message anonymously. Signatures are verified with a single group public key, but the verification process does not reveal the identity of the signer. In some exceptional case, a designated authority, called the opener, identifies the actual signer. However, ordinary group signature puts extremely strong

⋆ The first author is supported by a JSPS Fellowship for Young Scientists.
⋆⋆ This work was done when the second author was a postdoctoral researcher at Center for Highly Dependable Embedded Systems Technology, Japan Advanced Institute of Science and Technology (JAIST).
⋆⋆⋆ This work was done when the fourth author was a doctoral student in The University of Tokyo, Japan.
† The fifth author is supported by a JSPS Fellowship for Young Scientists.

M. Abdalla and T. Lange (Eds.): Pairing 2012, LNCS 7708, pp. 270–294, 2013.

privilege on the opener, i.e., the opener can freely identify the originator of any signature of his choice. In other words, ordinary group signature schemes provide no assurance on privacy against the opener at all. For example, in anonymous auction (which will later be explained in more detail), the opener can extract all bidders' identities.

This paper investigates a way of decentralizing this strong power of the opener. To this end, we propose a new kind of group signatures, group signature with a message-dependent opening capability. It divides (or "decentralizes") the strong power of the opener by introducing another authority, called the *admitter*. In an exceptional case in which, for example, a signature on a problematic message is found, the admitter issues a *token* which corresponds to the message (not the whole signed message). By using this token, the opener extracts the signer's identity from the signature while without the token, he is not able to do so. For instance, if the admitter decides that a message "Mr. XXX is fool!" should not be publicized as a signed message by an anonymous group member, he issues a token on this message. Then, by using it, the opener can immediately open the signer's identity of any signature if it corresponds to the above message.

At a first glance, one may think that for achieving the above functionality, the popular thresholding technique (i.e. thresholding the opener into multiple less-trusted openers) would be already sufficient. However, this is not true. Namely, in our context, the token is generated based on *the message which the admitter chooses* but not the signature for such messages. Therefore, once a token under a message (which is chosen by the admitter) is issued, for all signatures of this message, the signer's identity can be immediately extracted by the opener without interacting with any other party. Consequently, for a message which has already been specified as problematic, the opener can non-interactively open the signer's identity, and furthermore, if the admitter considers that there is no need to specify further messages which should be opened anymore, then he can erase his memory for avoiding leaking his secret. Notice that even when the admitter erases his secret, the opener can still open the signer's identity of any signature provided that its corresponding message was specified by the admitter before.

Contributions. In this paper, we propose group signature with a new additional capability, called *group signature with message-dependent opening* (GS-MDO). In GS-MDO, as mentioned above, we introduce the *admitter* which issues tokens for specific messages, and by using these tokens, the opener can extract signers' identities from signatures only if their corresponding messages are those specific ones. Due to this functionality, we can flexibly restrict the ability of the opener without any complicated interactive procedure (e.g. threshold decryption).

We first give a security definition of GS-MDO. Our security definition is an extension of the Bellare-Micciancio-Warinschi model [7] which is considered as the basic security definition for group signatures in the static setting, and more specifically, our security model is a natural modification of this model according to the difference between the standard group signature and ours which introduces the functionality of the message-dependent opening. Next, we discuss technical

hurdles for constructing GS-MDO which satisfies the above security requirement. Especially, we show that *it is possible to derive identity-based encryption (IBE) from any GS-MDO scheme in a black-box manner* if the underlying GS-MDO is secure in the above sense. In other words, IBE is crucial for constructing GS-MDO, and thus, it is impossible to construct GS-MDO without using IBE as a building block. Then, based on this observation, we present a generic construction of GS-MDO from IBE and adaptive non-interactive zero-knowledge (NIZK) proofs. Notice that in our generic construction, simulation-soundness [39] for NIZK is not required while the generic construction of the (standard) group signature [7] requires this strong property. Lastly, we propose an efficient instantiation of GS-MDO by applying the Groth-Sahai proof [29] to our generic construction. For utilizing the Groth-Sahai proof in our generic construction, we see that an IBE scheme which is compatible to the Groth-Sahai proof (like "structure preserving signatures" [4]) is necessary since our generic construction requires IBE. Unfortunately, there is no known such primitive, and thus we also construct a new IBE scheme which satisfies this requirement. By using our new IBE together with the Groth-Sahai proof, a fairly practical GS-MDO can be constructed. Specifically, the size of a signature is approximately 16 kilobytes when 256-bit prime order group is used. However, we should also honestly mention that our IBE has only k-resilient security [30], and consequently, the resulting GS-MDO scheme inherits this restriction (i.e. the admitter can issues at most k tokens).

Applications. As mentioned before, a straightforward application of GS-MDO schemes is detecting the originator of inappropriate messages in an anonymous bulletin board system. We further discuss more other potential applications of message-dependent opening systems in the following.

The first application we discuss is *anonymous auction*. In this application, the bidders form the group of anonymous signers. Each bidder produces a group signature on his bidding price. To detect the winner(s), the admitter issues the token for opening signatures on the highest price. Then the opener is only able to open the signatures on the highest price.

The advantage (of the message-dependent opening approach) over the threshold approach becomes clear in this application. Suppose that there are *many* winners who all bid the highest price in a tie. In the threshold approach, an interaction will be needed for each winner, hence the total communication cost will be proportional to the number of winners. In contrast, if one takes the message-dependent opening approach, only a small communication from the admitter to the opener will be needed. The communication cost does not depend on the number of winners.

Another application in which the message-dependent opening capability is useful is *identity escrow*. Let us consider an automated parking garage [34], in which when a customer enters the garage, he generates a group signature on a

message which encodes the date when he enters the garage (say, the string "2012-02-20"). Suppose a case in which there is an accident (a person is murdered, for example) in the garage. In this case the opener will open the signatures on the date when the accident occurs, in order to identify who is there at that day.

In this application, the opener needs to open *many* signatures on the *same* message. If one adopts the threshold technique to decentralize the authority, a large amount of interactions is required to open all the signatures. The message-dependent opening capability removes interactions between authorities, that is, the admitter issues a token for the day, and the opener opens all the signatures without interaction.

Related Works. Since the first proposal of group signature by Chaum and van Heyst [20], many efficient constructions have been proposed, most of which are relying on the random oracle model [6,11,18,33,25,23,10]. Many initial schemes were based on the strong-RSA assumption. Group signature schemes based on assumptions of the discrete-logarithm type were achieved, to name a few, by Camenisch and Lysyanskaya [18] and by Boneh, Boyen, and Shacham [11]. The former scheme is based on the LRSW assumption, while the latter is based on the q-strong Diffie-Hellman assumption.

Except generic constructions from general NIZK techniques, group signature schemes without relying on the random oracles are only very recently achieved. Ateniese, Camenisch, Hohenberger, and de Medeiros first proposed a group signature scheme from interactive assumptions avoiding random oracles [5]. Following to this scheme, Groth proposed a group signature scheme which avoids random oracles and interactive assumptions [27], but the scheme has a very large signature size. Boyen and Waters proposed highly efficient constructions [14,15], although the security guarantee of their schemes are not very strong, i.e. they only achieve so-called CPA-anonymity. Groth proposed another group signature scheme [28], which is almost as efficient as the Boyen-Waters schemes and satisfies higher security guarantee of the Bellare-Shi-Zhang model [8].

As for decentralizing and distributing the power of the group manager, separability of a cryptographic protocol was introduced by Kilian and Petrank [34] in the context of identity escrow. Lately, this notion was refined and adopted to the context of group signature by Camenisch and Michels [19]. The separability notion demands that keys of several entities involved in the cryptographic primitive need to be generated independently each other. In their setting, the power of a group manager is separated into two authorities. The first authority is able to allow a new group member to join the group, but not able to identify the originator of a group signature, and the other authority is vice versa. More formal modeling of these separated authorities is put forward by Bellare, Shi, and Zhang [8] and Kiayias and Yung [32].

Traceable signature is an extended notion of group signature, introduced by Kiayias, Tsiounis, and Yung [31]. This primitive allows the group manager to specify a group member as "misbehaving". Once a member was specified by the manager, anyone becomes able to detect the signatures of the specified user without interacting with the manager. In this time signatures of other group

members continue to be anonymous. In our terminology, this primitive achieves somewhat "signer-dependent opening" capability, but no message-dependent opening is achieved. A contractual anonymity system [40] has been proposed based on group signatures with verifier-local revocation [13]. In this system, when a user breaks a contract, an accountability server revokes anonymity of the user and notices the identity of the user to the service provider (In the contractual anonymity system, a user is said to *break the contract* when the user sends a message specified by the contract policy of the service provider). Since this scheme uses the conventional open algorithm, this system also differs from message-dependent opening.

Paper Organization. The rest of the paper is structured as follows. Sect. 2 describes definitions and security notions of several building blocks briefly. Sect. 3 presents the notion of GS-MDO and its syntax and security definitions. Sect. 4 discusses difficulties behind constructing efficient GS-MDO schemes. Specifically we argue that use of IBE in a construction of GS-MDO is essential by showing a generic construction of IBE from GS-MDO. In Sect. 5 and 6, we propose a generic construction of GS-MDO and its fairly efficient instantiation.

2 Preliminaries

Signatures. A signature scheme consists of the following three algorithms: A key generation algorithm $\mathsf{SigKg}(1^\lambda)$ outputs a pair (vk, sk). A signing algorithm $\mathsf{Sign}_{sk}(M)$ generates a signature s for a message M. A verification algorithm $\mathsf{Verify}_{vk}(M, s)$ outputs $\top$ or $\bot$, which respectively indicate "accept" or "reject". As a correctness, for all $\lambda \in \mathbb{N}$, all pairs (vk, sk) in the range of $\mathsf{SigKg}(1^\lambda)$, and all messages M, it is required to be satisfied that $\Pr[\mathsf{Verify}_{vk}(M, \mathsf{Sign}_{sk}(M)) = \top] = 1$. A signature scheme is *existentially unforgeable under chosen-message attack (EUF-CMA)* if all PPT adversaries, given vk generated from $\mathsf{SigKg}(1^\lambda)$ and an access to a signing oracle, which gives the adversary a signature of his choice, have negligible probability of outputting a pair (M, s) where M was never queried to the signing oracle and $\mathsf{Verify}_{vk}(M, s) = \top$. A signature scheme is said to be *strongly unforgeable one-time signature* when no adversary, given vk and allowed to access to a signing oracle *only at most once*, can output a valid message-signature pair (M, s) (i.e. $\mathsf{Verify}_{vk}(M, s) = \top$) which is different from the message-signature pair obtained from the signing oracle.

Tag-Based Key Encapsulation Mechanism. A tag-based key encapsulation mechanism (tag-based KEM)[1] [36,35] consists of the following three algorithms: A key generation algorithm $\mathsf{TKg}(1^\lambda)$ outputs a pair (pk, dk). An encapsulation

[1] Tag-based encryption, an encryption analogue of tag-based KEM, is originally introduced as "encryption with labels" by Shoup and Gennaro [42]. Tag-based KEM is different from "tag-KEM", introduced by Abe, Gennaro, Kurosawa, and Shoup [3]. However, any CCA-secure tag-KEM scheme can be immediately converted to a tag-based KEM scheme which is sufficiently secure for our purpose.

algorithm $\mathsf{TEnc}_{pk}(t)$ outputs (C, K) where a ciphertext C for a tag t encapsulates a session key $K \in \mathcal{K}_{\mathrm{PKE}}$, where $\mathcal{K}_{\mathrm{PKE}}$ is the session key space associated with the scheme. A decapsulation algorithm $\mathsf{TDec}_{dk}(t, C)$ outputs a decapsulated session key K or a special symbol $\perp$ indicating an invalid ciphertext. A tag-based KEM is said to be *selective-tag weakly chosen-ciphertext secure* when no PPT adversary has non-negligible advantage in the following game: The adversary is given a security parameter 1^λ and output a target tag t^*, then the challenger gives a public key pk. After receiving the public key, the adversary, in an arbitrary order, issues decryption queries (t, C), to which the challenger responds with the decryption result of C under the tag t. Here the adversary is not allowed to issue queries with $t = t^*$. At some point the adversary requests a challenge. The challenger flips a fair coin b' and sends (C^*, K^*) where C^* is a ciphertext generated under the tag t^* and K^* is either the session key encapsulated in C^* when $b' = 0$ or a random session key when $b' = 1$. After receiving the challenge the adversary is again allowed to issue decryption queries. The same restriction for queries is applied as before. Finally the adversary outputs a bit b. The advantage of the adversary is defined by the probability that $b = b'$ minus $1/2$.

Identity-Based KEM and Its k-resilient Variant. A k-resilient identity-based KEM [30] consists of the following four algorithms: A setup algorithm $\mathsf{ISetup}(1^\lambda, 1^k)$ outputs a pair (par, mk). A key extraction algorithm $\mathsf{IExt}_{mk}(ID)$ outputs a user decapsulation key dk_{ID}. An encapsulation algorithm $\mathsf{IEnc}_{par}(ID)$ outputs (C, K) where a ciphertext C for an identity ID encapsulates a session key $K \in \mathcal{K}_{\mathrm{IBE}}$, where $\mathcal{K}_{\mathrm{IBE}}$ is the session key space associated with the scheme. A decapsulation algorithm $\mathsf{IDec}_{dk_{ID}}(C)$ outputs a decapsulated session key K or a special symbol $\perp$ indicating an invalid ciphertext. A k-resilient identity-based KEM is said to be *k-resilient* if no PPT adversary has non-negligible (in λ) advantage in the following game: The adversary first receives a public parameter par. After receiving the parameter the adversary, in an arbitrary order, issues extraction queries ID, to which the challenger responds with the user decapsulation key for the user ID. At some point the adversary requests a challenge with an identity ID^*. The challenger flips a fair coin b' and sends a pair (C^*, K^*) where C^* is a ciphertext for the user ID^* and K^* is either the session key encapsulated in C^* when $b' = 0$ or a random session key when $b' = 1$. The adversary is not allowed to request a challenge with an identity whose user decapsulation key is queried before. After receiving the challenge the adversary is again allowed to issue extraction queries. This time querying the user decapsulation key for ID^* is disallowed. The adversary is also restricted that the total number of queries before and after the challenge is at most k. Finally the adversary outputs a bit b. The advantage of the adversary is defined by the probability that $b = b'$ minus $1/2$. We also say that an identity-based KEM is *fully secure* when any PPT adversary has non-negligible advantage in the same game even when the number of extraction queries is unbounded.

Non-Interactive Zero-Knowledge Proofs. A non-interactive proof system for a polynomial-time computable relation R consists of three probabilistic algorithms K, P, and V. The common reference string generation algorithm K produces a common reference string Σ. The proof algorithm P takes a common reference string Σ, a theorem x, and a witness w, where $R(x,w) = \top$, and produces a proof π. The verification algorithm V takes as input (Σ, x, π), and outputs either $\top$ or $\bot$. We say that a non-interactive proof system (K,P,V) has perfect completeness, when we have $\Pr[\Sigma \leftarrow K(1^\lambda); (x,w) \leftarrow \mathcal{A}(\Sigma); \pi \leftarrow P(\Sigma,x,w) : V(\Sigma,x,\pi) = \top \vee R(x,w) = \bot] = 1$ for any adversary $\mathcal{A}$. We say that a non-interactive proof system (K,P,V) has perfect soundness, when we have $\Pr[\Sigma \leftarrow K(1^\lambda); (x,\pi) \leftarrow \mathcal{A}(\Sigma) : V(\Sigma,x,\pi) = \bot \vee x \in L] = 1$ for all adversary $\mathcal{A}$, where L denotes the set of all x that has at least one w such that $R(x,w) = \top$. We say that a non-interactive proof system (K,P,V) is zero-knowledge when there exists a pair of probabilistic algorithms (S_1, S_2) such that we have $\Pr[\Sigma \leftarrow K(1^\lambda); (x,w) \leftarrow \mathcal{A}(\Sigma); \pi \leftarrow P(\Sigma,x,w) : \mathcal{A}(\pi) = 1] - \Pr[(\Sigma,\tau) \leftarrow S_1(1^\lambda); (x,w) \leftarrow \mathcal{A}(\Sigma); \pi \leftarrow S_2(\Sigma,\tau,x) : \mathcal{A}(\pi) = 1]$ is negligible for all PPT adversaries $\mathcal{A}$ that do not output (x,w) with $R(x,w) = \bot$.

3 Group Signatures with Message-Dependent Opening

Firstly we give an explanation of the scenario in which group signature with message-dependent opening is used. As ordinary group signatures, a GS-MDO scheme allows group members to sign a message anonymously, that is, without revealing their identities but only showing that one of the group members actually signed. In exceptional cases, a designated third party, called the opener, can "open" exceptional signatures, to identify the originator of signatures. In contrast to ordinary group signature schemes, a GS-MDO scheme requires the opener to cooperate with another authority, called the admitter, to open signatures. The admitter issues a message-specific token, and the opener is able to open signature on some message *only when a token for the message is issued from the admitter*.

A formal model of this scenario is given by the following definition. A GS-MDO scheme consists of the following five algorithms:

GKg: This algorithm takes as an input $(1^\lambda, 1^n, 1^k)$ where λ is a security parameter, n is the number of group members, and k is the maximum number of message-specific tokens that can be issued, and returns a group public key gpk, a message specification key msk, an opening key ok, and n group signing keys $\{gsk_i\}_{i\in[n]}$.

GSig: This algorithm takes as inputs gpk, gsk_i, and a message M, and returns a group signature σ.

Td: This algorithm takes as inputs gpk, msk, and M, and returns the token t_M for M.

GVf: This algorithm takes as inputs gpk, σ, and M, and returns $\top$ or $\bot$.

Open: This algorithm takes as inputs gpk, ok, M, σ, and t_M, and returns $i \in \mathbb{N}$ or $\bot$.

As a correctness, it is required that for all λ, n, k and for all $(gpk, msk, ok, \{gsk_i\}_{i\in[n]})$ in the range of $\mathsf{GKg}(1^\lambda, 1^n, 1^k)$, $\mathsf{GVf}(gpk, M, \mathsf{GSig}(gpk, gsk_i, M)) = \top$ for all $M \in \{0,1\}^*$ and $i \in [n]$, and $\mathsf{Open}(gpk, ok, M, \mathsf{GSig}(gpk, gsk_i, M), \mathsf{Td}(gpk, msk, M)) = i$ for all $M \in \{0,1\}^*$ and $i \in [n]$.

As in ordinary group signature, we need to ensure anonymity and traceability. However, in contrast to ordinary group signature, we have to further ensure two types of anonymity. It is related to the original motivation of the introduction of the admitter. The introduction of the admitter is intended to strengthen signers' anonymity against the authorities as strong as possible. To capture this intention, we define the indistinguishability of the originator of the signature in the strong setting that the opening key is given to the adversary. As a counterpart of this, we also define the indistinguishability in the setting that the message-specification key is given to the adversary.

For traceability, we just use the same definition to the ordinary group signature, in which the authorities are entirely corrupted by the adversary, since even ordinary group signature schemes has ensured that traceability against entirely corrupted openers.

Opener Anonymity. Here we give a formal definition of anonymity against the opener, called *opener anonymity*. It is formalized as the indistinguishability of signatures of two different signers of the adversary's choice. In the indistinguishability game, the adversary is given the opening key, and is asked to distinguish signatures of two different signers of its own choice. Opener anonymity is defined by requiring that no adversary has non-negligible advantage in distinguishing signatures.

We again remark that contrary to ordinary group signatures, the adversary is allowed to have the opening key. This is intended for modeling "anonymity against the opener."

Definition 1. *A GS-MDO scheme has* opener anonymity *if the advantage of any PPT adversary $\mathcal{A}$ in the following game between a challenger and the adversary is negligible in the security parameter λ:*

Setup. *The challenger runs $\mathsf{GKg}(1^\lambda, 1^n, 1^k)$ to obtain $(gpk, ok, msk, \{gsk_i\}_{i\in[n]})$ and sends $(gpk, ok, \{gsk_i\}_{i\in[n]})$ to $\mathcal{A}$.*

Token Query (Phase I). *$\mathcal{A}$ adaptively issues token queries. For a token query for a message M, the challenger responds with t_M which is obtained by running $\mathsf{Td}(gpk, msk, M)$.*

Challenge. *At some point $\mathcal{A}$ requests a challenge for i_0, $i_1 \in [n]$ and a message M^*. The challenger chooses a random bit b, and responds with $\mathsf{GSig}(gpk, gsk_{i_b}, M^*)$. In this phase $\mathcal{A}$ is forbidden to submit M^* whose token is previously queried in Phase I.*

Token Query (Phase II). *$\mathcal{A}$ continues to query tokens. In this phase $\mathcal{A}$ is forbidden to query M^*, which is submitted in Challenge phase.*

Guess. *Finally $\mathcal{A}$ outputs a bit b'. The advantage of $\mathcal{A}$ is defined by the absolute difference between the probability that b' is equal to b and $1/2$.*

We also say that a GS-MDO scheme has opener anonymity with k-bounded tokens *if any PPT adversary $\mathcal{A}$ which issues at most k token queries in total has negligible advantage.*

Admitter Anonymity. We then give a definition of anonymity against the admitter, called *admitter anonymity*. It is formalized in a similar manner to opener anonymity. That is, admitter anonymity requires signatures of two different signers are indistinguishable even when the adversary is given the message-specification key. The formal definition is as follows:

Definition 2. *A GS-MDO scheme has* admitter anonymity *if the advantage of any PPT adversary $\mathcal{A}$ in the following game between a challenger and the adversary is negligible in the security parameter λ:*

Setup. *The challenger runs* $\mathsf{GKg}(1^\lambda, 1^n, 1^k)$ *to obtain* $(gpk, ok, msk, \{gsk_i\}_{i\in[n]})$ *and sends* $(gpk, msk, \{gsk_i\}_{i\in[n]})$ *to* $\mathcal{A}$.

Open Query (Phase I). $\mathcal{A}$ *adaptively issues open queries. For an open query for a message-signature pair* (M, σ)*, the challenger generates* t_M *by running* $\mathsf{Td}(gpk, msk, M)$ *and responds with* $\mathsf{Open}(gpk, ok, M, \sigma, t_M)$.

Challenge. *At some point* $\mathcal{A}$ *requests a challenge for* i_0, $i_1 \in [n]$ *and a message* M^**. The challenger chooses a random bit* b*, and responds with* $\sigma^* \leftarrow \mathsf{GSig}(gpk, gsk_{i_b}, M^*)$.

Open Query (Phase II). $\mathcal{A}$ *continues to submit open queries. In this phase* $\mathcal{A}$ *is forbidden to query* σ^**, which is same as the signature produced in Challenge phase.*

Guess. *Finally* $\mathcal{A}$ *outputs a bit* b'*. The advantage of* $\mathcal{A}$ *is defined by the absolute difference between the probability that* b' *is equal to* b *and* $1/2$.

Notice that the number of opening queries the adversary issues is unbounded (but of course polynomially many).

Traceability. The last notion is *traceability*, which requires that even if the opener and the admitter collude and they further adaptively corrupt some group members, the corrupted parties can produce neither forged signatures nor untraceable signatures. In contrast to the case of the anonymity notions, this case considers a collusion of two authorities.

Definition 3. *A GS-MDO scheme has* traceability *if the advantage of any PPT adversary $\mathcal{A}$ in the following game between a challenger and the adversary is negligible in the security parameter λ:*

Setup. *The challenger runs* $\mathsf{GKg}(1^\lambda, 1^n, 1^k)$ *to obtain* $(gpk, ok, msk, \{gsk_i\}_{i\in[n]})$ *and sends* (gpk, ok, msk) *to* $\mathcal{A}$.

Query. $\mathcal{A}$ *adaptively issues following two types of queries:*

1. *The first type of queries is* key revealing query*, in which $\mathcal{A}$ requests for revealing the group signing key of the group member i. For this type of queries the challenger responds with* gsk_i.

$\mathsf{ISetup}(1^\lambda)$:
$(gpk, ok, msk, \{gsk_1, gsk_2\}) \leftarrow \mathsf{GKg}(1^\lambda, 1^2)$;
$par \leftarrow (gpk, ok, gsk_1, gsk_2)$; $mk \leftarrow msk$;
Output (par, mk).

$\mathsf{IExt}_{mk}(ID)$:
$dk_{ID} \leftarrow \mathsf{Td}(gpk, mk, ID)$;
Output dk_{ID}.

$\mathsf{IEnc}_{par}(ID)$:
For $i \in \{1, \ldots, \lambda\}$:
  $K_i \leftarrow \{0, 1\}$;
  $\sigma_i \leftarrow \mathsf{GSig}(gpk, gsk_{K_i+1}, ID)$;
$C \leftarrow (\sigma_1, \ldots, \sigma_\lambda)$;
$K \leftarrow K_1 \cdots K_\lambda$;
Output (C, K).

$\mathsf{IDec}_{dk_{ID}}(C)$:
Parse C as $(\sigma_1, \ldots, \sigma_\lambda)$;
For $i \in \{1, \ldots, \lambda\}$:
  $K_i \leftarrow \mathsf{Open}(gpk, ok, ID, \sigma_i, dk_{ID})$;
If $K_i = \perp$ for some i
  then Output $\perp$;
Else Output $K_1 \ldots K_\lambda$.

Fig. 1. The black-box construction of identity-based KEM from group signature with message-dependent opening

2. *The second type of queries is* signing query*, in which $\mathcal{A}$ requests for a signature on some message by some group member. For a query (i, M) of this type, the challenger responds with $\mathsf{GSig}(gpk, gsk_i, M)$.*

Forge. *Finally the challenger outputs a forgery (M^*, σ^*). $\mathcal{A}$ wins if $\mathsf{GVf}(gpk, M^*, \sigma^*) = \top$ and one of the following two conditions holds: (1) $\mathsf{Open}(gpk, ok, M^*, \sigma^*, \mathsf{Td}(gpk, msk, M^*)) = \perp$, or (2) $\mathsf{Open}(gpk, ok, M^*, \sigma^*, \mathsf{Td}(gpk, msk, M^*)) = i^* \neq \perp$, and neither a key revealing query for the user i^* nor a signing query for (i^*, M^*) is submitted. The advantage of $\mathcal{A}$ is defined by the probability that $\mathcal{A}$ wins.*

4 Difficulty in Having Efficient Constructions

In this section we discuss several difficulties in designing efficient GS-MDO schemes. We firstly investigate relationships between GS-MDO and other cryptographic primitives, and then we discuss the difficulty that lies in designing efficient constructions.

As for the relationship to other primitives, we show that the existence of a GS-MDO scheme implies that of an IBE scheme. In other words, we will present a black-box construction of IBE from any GS-MDO scheme. The same holds for the k-resilient versions.

The formal theorems are as follows:

Theorem 1. *If the underlying GS-MDO scheme satisfies opener anonymity, the identity-based KEM in Fig. 1 is fully secure.*

Theorem 2. *If the underlying GS-MDO scheme satisfies opener anonymity with k-bounded tokens, the identity-based KEM in Fig. 1 is k-resilient.*

Formal proofs can be given by a straightforward modification from the proof by Abdalla and Warinschi [1] or the similar technique used by Ohtake, Fujii, Hanaoka, and Ogawa [37], hence we omit detailed proofs.

We also note that Fig. 1 only shows a construction of identity-based *key encapsulation mechanism* rather than identity-based *encryption*. However, it suffices for the theorems since we can obtain a secure encryption scheme by combining the construction with an appropriate data encapsulation mechanism.

These theorems suggest that to use IBE is crucial for constructing a GS-MDO scheme. Considering the fact that a black-box construction of IBE from trapdoor permutation is impossible [12], we should conclude that it is almost unavoidable for a GS-MDO scheme to relying on an IBE scheme or its equivalence, not only on trapdoor permutation and NIZK proof. Otherwise one would construct an IBE scheme from surprisingly weaker primitives.

Another important aspect to establish an *efficient* GS-MDO scheme is realizing a "Groth-Sahai compatible" IBE scheme. This is because the only known construction of non-interactive zero-knowledge proof with reasonable efficiency is limited to the Groth-Sahai proof system. Also note that a non-interactive zero-knowledge proof system has been an important building block for almost all group signature schemes ever.

However, no currently known IBE scheme is Groth-Sahai compatible in the sense that the Groth-Sahai proof system cannot prove a kind of well-formedness of an IBE ciphertext in a zero-knowledge manner.

To overcome this gap, we adopt k-resilient IBE instead of fully secure IBE. In particular we design a k-resilient IBE scheme from the decision linear assumption by modifying the Heng-Kurosawa scheme [30] for this purpose (We also note that a similar construction can be obtained from a key-insulated encryption scheme by Dodis, Katz, Xu, and Yung [24]). The modification is needed since the original Heng-Kurosawa scheme is based on the decision Diffie-Hellman (DDH) assumption, which does not hold in groups with a bilinear map, and the Groth-Sahai proof system relies on a bilinear map in an essential way.

5 Generic Construction

In this section, we give a construction of a GS-MDO scheme. The construction is built on an EUF-CMA secure signature scheme, a strongly unforgeable one-time signature scheme, a selective-tag weakly chosen-ciphertext secure tag-based KEM, a k-resilient identity-based KEM, and an adaptive NIZK proof system.

At a first glance there are various building blocks. However, our generic construction is only relying on the existence of an IBE scheme and that of an NIZK proof system. Indeed signature schemes and a chosen-ciphertext secure tag-based encryption scheme can be constructed from a fully secure IBE.

The proposed construction shares an underlying idea with the generic construction by Bellare, Micciancio, and Warinschi (the BMW construction) [7] except the use of "simulation-sound" NIZK proofs. The proposed construction no longer relies on such a strong security requirement of simulation-soundness, which was exploited by the BMW construction [7]. Instead of the strong security requirement of simulation-soundness, we combine (ordinary) NIZK proofs with a strongly unforgeable one-time signature scheme. We remark that essentially same techniques have been used in a variety of contexts. To name a few,

Groth [28] used this technique for an efficient group signature scheme in a very similar manner. Camenisch, Chandran, and Shoup [17,16] used this to construct simulation-sound NIZK proofs, improving the result of Groth [27].

$\mathsf{GKg}(1^\lambda, 1^n, 1^k)$:
$(vk_{\mathrm{issue}}, sk_{\mathrm{issue}}) \leftarrow \mathsf{SigKg}(1^\lambda)$;
$(pk, dk) \leftarrow \mathsf{TKg}(1^\lambda)$;
$(par, mk) \leftarrow \mathsf{ISetup}(1^\lambda, 1^k)$;
$\Sigma \leftarrow K(1^\lambda)$;
$gpk \leftarrow (vk_{\mathrm{issue}}, pk, par, \Sigma)$;
$ok \leftarrow dk$;
$msk \leftarrow mk$;
For all $i \in [1, n]$:
  $(vk_i, sk_i) \leftarrow \mathsf{SigKg}(1^\lambda)$;
  $cert_i \leftarrow \mathsf{Sign}_{sk_{\mathrm{issue}}}(\langle i, sk_i \rangle)$;
  $gsk_i \leftarrow (i, vk_i, cert_i, sk_i)$;
Output $(gpk, ok, msk, \{gsk_i\}_i)$.

$\mathsf{GVf}(gpk, M, \sigma)$:
Parse gpk as $(vk_{\mathrm{issue}}, pk, par, \Sigma)$;
Parse σ as $(vk_{\mathrm{OT}}, C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi, \pi, \sigma_{\mathrm{OT}})$;
If $\mathsf{Verify}^{\mathrm{OT}}_{vk_{\mathrm{OT}}}(\langle C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi, \pi \rangle, \sigma_{\mathrm{OT}}) = 1$
and $V_{\mathrm{NIZK}}(\ldots) = 1$ then Output $\top$;
Else Output $\bot$.

$\mathsf{Td}(gpk, msk, M)$:
Parse gpk as $(vk_{\mathrm{issue}}, pk, par, \Sigma)$;
$t_M \leftarrow \mathsf{IExt}(par, msk, M)$;
Output t_M.

$\mathsf{GSig}(gpk, gsk_i, M)$:
Parse gpk as $(vk_{\mathrm{issue}}, pk, par, \Sigma)$;
Parse gsk_i as $(i, vk_i, cert_i, sk_i)$;
$s \leftarrow \mathsf{Sign}_{sk_i}(M)$;
$(vk_{\mathrm{OT}}, sk_{\mathrm{OT}}) \leftarrow \mathsf{SigKg}^{\mathrm{OT}}(1^\lambda)$;
$(C_{\mathrm{PKE}}, K_{\mathrm{PKE}}) \leftarrow \mathsf{TEnc}_{pk}(vk_{\mathrm{OT}})$;
$(C_{\mathrm{IBE}}, K_{\mathrm{IBE}}) \leftarrow \mathsf{IEnc}_{par}(M)$;
$\chi \leftarrow \langle i, vk_i, cert_i, s \rangle \odot K_{\mathrm{PKE}} \odot K_{\mathrm{IBE}}$;
$\pi \leftarrow P_{\mathrm{NIZK}}(\cdots)$;
$\sigma_{\mathrm{OT}} \leftarrow \mathsf{Sign}^{\mathrm{OT}}_{sk_{\mathrm{OT}}}(\langle C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi, \pi \rangle)$;
$\sigma \leftarrow (vk_{\mathrm{OT}}, C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi, \pi, \sigma_{\mathrm{OT}})$;
Output σ.

$\mathsf{Open}(gpk, ok, M, \sigma, t_M)$:
Parse gpk as $(vk_{\mathrm{issue}}, pk, par, \Sigma)$;
Parse σ as $(vk_{\mathrm{OT}}, C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi, \pi, \sigma_{\mathrm{OT}})$;
$K_{\mathrm{PKE}} \leftarrow \mathsf{TDec}_{ok}(vk_{\mathrm{OT}}, C_{\mathrm{PKE}})$;
$K_{\mathrm{IBE}} \leftarrow \mathsf{IDec}_{t_M}(M, C_{\mathrm{IBE}})$;
$\langle i, vk_i, cert_i, s \rangle \leftarrow \chi \odot K^{-1}_{\mathrm{IBE}} \odot K^{-1}_{\mathrm{PKE}}$;
If $\mathsf{Verify}^{\mathrm{OT}}_{vk_{\mathrm{OT}}}(\langle C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi, \pi \rangle, \sigma_{\mathrm{OT}}) = 1$
and $V_{\mathrm{NIZK}}(\ldots) = 1$
then Output i;
else Output $\bot$;

Fig. 2. The brief overview of the proposed GS-MDO scheme. The operator $\odot$ denotes some group operation. In the concrete instantiation, $\langle \cdots \rangle$ denotes a tuple consisting of all group elements that appear in the bracket, and the operator $\odot$ is the component-wise group multiplication. The non-interactive proof system $(K, P_{\mathrm{NIZK}}, V_{\mathrm{NIZK}})$ is for demonstrating the existence of a satisfying assignment of Eq. (1).

The Construction. In the construction, a group member has a key pair (vk_i, sk_i) of the signature scheme in which vk_i is authorized by another verification key vk_{issue} at the setup time. When a member makes a group signature, the member simply signs a message by sk_i. To be anonymous, the member further encrypts the signature together with the certificate (of the member), which authorizes the verification key vk_i, and attaches a non-interactive proof that demonstrates that a signature of an authorized member is encrypted. To encrypt a signature, the member uses a multiple encryption technique to ensure

neither the opener nor the admitter can reveal the identity as long as the admitter does not issue a token to the opener. The complete description of the scheme is shown in Fig. 2.

Let us explain the non-interactive proof that appears in the construction. The signature of the proposed scheme is of the form as $(vk_{\mathrm{OT}}, C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi, \pi, \sigma_{\mathrm{OT}})$, and, as mentioned above, the proof π demonstrates a valid signature of an authorized group member is encrypted into $(C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi)$ in a kind of a "multiple encryption" manner. In detail, the proof π proves that there exists a randomness r (for tag-based KEM), another randomness ρ (for identity-based KEM), a group member i, and the verification key vk_i, the certificate $cert_i$, and the signature s on a message M, such that

$$\begin{aligned}
&(C_{\mathrm{PKE}}, K_{\mathrm{PKE}}) = \mathsf{TEnc}_{pk}(vk_{\mathrm{OT}}; r),\\
&(C_{\mathrm{IBE}}, K_{\mathrm{IBE}}) = \mathsf{IEnc}_{par}(M; \rho),\\
&\chi = \langle i, vk_i, cert_i, s\rangle \odot K_{\mathrm{PKE}} \odot K_{\mathrm{IBE}},\\
&\mathsf{Verify}_{vk_{\mathrm{issue}}}(\langle i, vk_i\rangle, cert_i) = \top,\\
&\mathsf{Verify}_{vk_i}(M, s) = \top.
\end{aligned} \tag{1}$$

Technically speaking, we need several requirements on the session key spaces of tag-based KEM and k-resilient IBE. The requirements are: (i) The tag-based KEM scheme and the k-resilient IBE scheme share the same session key space $\mathcal{K}_{\mathrm{PKE}} = \mathcal{K}_{\mathrm{IBE}}$ and (ii) this session key space forms a finite group. These requirements are needed because we do a one-time pad to encrypt a signature of the group member. This group operation also needs to fall into the class of relations that the used non-interactive proof system can represent.

Finally, there are two encoding functions needed for completing the generic construction. The first is used to encode the identity of a group member and his verification key into the message space of the signature scheme when generating certificates of group members. The second one is used to encode $(i, vk_i, cert_i, s)$ into $\mathcal{K}_{\mathrm{PKE}}$, where i is the identity of a group member and vk_i, $cert_i$, and s are his verification key, certificate, and signature, respectively. It is used when issuing group signatures, especially encrypting his signature in order to hide his identity.

As below, the generic construction will have desirable security properties when all building blocks satisfy appropriate security properties.

Theorem 3. *The proposed scheme satisfies opener anonymity with k-bounded tokens if the identity-based KEM is k-resilient and the non-interactive proof system is zero-knowledge.*

Theorem 4. *The proposed scheme satisfies admitter anonymity when the tag-based KEM is selective-tag weakly chosen-ciphertext secure, the non-interactive proof system is zero-knowledge, and the one-time signature scheme is strongly unforgeable.*

Theorem 5. *The proposed scheme satisfies traceability when the non-interactive proof system is sound and the signature scheme is EUF-CMA secure.*

All the proofs of the theorems will appear in Appendix B.

6 Efficient Instantiation

Toward an efficient scheme, we will discuss how to instantiate the building blocks used in the generic constructions of the previous section.

As for the non-interactive proof, an obvious choice is the Groth-Sahai proof system, since there is no known fairly practical construction of a NIZK proof system except the Groth-Sahai proof system. However, to adopt the Groth-Sahai proof system, other building blocks are subjected to restrictions, due to the limitation of the type of theorems that the Groth-Sahai proof system can prove. In other words, other building blocks need to be *structure preserving* [2], and especially, the theorem should not involve elements of $\mathbb{G}_T$, where $\mathbb{G}_T$ is the target group of the underlying bilinear mapping. Hence, we have to choose an IBE scheme which fulfills this requirement as a building block, but unfortunately, there is no known such scheme. This means that *it is not straightforward to construct an efficient instantiation of our generic construction from the Groth-Sahai proof.*

In this section, we give an efficient instantiation by constructing a *structure preserving* IBE scheme and choosing other appropriate building blocks. However, we must also honestly mention that our IBE does not provide full security but only k-resilience [30]. It is also worth noting that constructing a structure-preserving IBE scheme is already an important open problem.

Our structure-preserving k-resilient IBE scheme is obtained by means of modifying the Heng-Kurosawa scheme [30] which is secure under the decision Diffie-Hellman (DDH) assumption in the sense of k-resilient security. Since the DDH assumption does not hold in a bilinear group, it is not possible to utilize it as it is, and thus, we construct a modified version of this scheme which is secure under the decision linear (DLIN) assumption.

6.1 k-Resilient IBE from the Decision Linear Assumption

As mentioned above, our proposed k-resilient IBE scheme can be obtained by applying several modifications to the original Heng-Kurosawa scheme [30, Sect. 3.2] which are as follows: (1) Basing on the DLIN assumption instead of the DDH assumption[2], (2) designing it as a key encapsulation mechanism instead of an encryption scheme, and (3) modifying it to encapsulate a sufficiently long session key in a constant size ciphertext (Indeed our proposed scheme encapsulates a session key of l group elements in a ciphertext of three group elements). Our proposed scheme is as follows:

[2] If we adopt the SXDH assumption, we can plug in the original Heng-Kurosawa scheme to the generic construction. However, in this case we need to set up two instances of the original Heng-Kurosawa scheme for two different groups $\mathbb{G}_1$ and $\mathbb{G}_2$, over which the bilinear map is defined. This is because the Abe-Haralambiev-Ohkubo signature scheme contains elements of both groups in its signature value. The same thing holds for the tag-based KEM.

Setup. Let $par = (u, v, h, \{D_{i,j} = u^{d_{i,j}} h^{d''_{i,j}}, \tilde{D}_{i,j} = v^{d'_{i,j}} h^{d''_{i,j}}\}_{i\in[l],j\in[k]})$ and $mk = (d_i(X), d'_i(X), d''_i(X))_{i\in[l]}$ where d_i, d'_i, and d''_i are the polynomials defined as follows: $d_i(X) = \sum_{j=0}^{k} d_{i,j}X^j$, $d'_i(X) = \sum_{j=0}^{k} d'_{i,j}X^j$, and $d''_i(X) = \sum_{j=0}^{k} d''_{i,j}X^j$ for all $i \in [l]$.

Key Extract. The decryption key for the user ID is derived as $dk_{ID} = \{d_i(ID), d'_i(ID), d''_i(ID)\}_{i\in[l]}$.

Encrypt. To encapsulate a session key, choose ρ and $\tilde{\rho}$ from $\mathbb{Z}_p$ randomly and compute $C_{\mathrm{IBE}} = (u^{\rho}, v^{\tilde{\rho}}, h^{\rho+\tilde{\rho}})$, which encapsulates the session key $K_{\mathrm{IBE}} = ((\prod_{j=0}^{k} D_{1,j}^{ID^j})^{\rho}(\prod_{j=0}^{k} \tilde{D}_{1,j}^{ID^j})^{\tilde{\rho}}, \ldots, (\prod_{j=0}^{k} D_{l,j}^{ID^j})^{\rho}(\prod_{j=0}^{k} \tilde{D}_{l,j}^{ID^j})^{\tilde{\rho}})$.

Decrypt. To decapsulate a session key from a ciphertext $C_{\mathrm{IBE}} = (C_1, C_2, C_3)$, compute $(C_1^{d_1(ID)} C_2^{d'_1(ID)} C_3^{d''_1(ID)}, \ldots, C_1^{d_l(ID)} C_2^{d'_l(ID)} C_3^{d''_l(ID)})$.

The security of this scheme is proved under the DLIN assumption, which says that given a tuple $(u, v, h, u^r, v^{\tilde{r}}, h^{\tilde{\tilde{r}}})$ it is hard to efficiently decide $r + \tilde{r} = \tilde{\tilde{r}}$ or not. Formal statements of the assumption and the theorem are as follows.

Definition 4. *We say that the* decision linear assumption *on* $\mathbb{G}$ *holds if for any polynomial-time algorithm* $\mathcal{D}$, $|\Pr[\mathcal{D}(u, v, h, u^r, v^{\tilde{r}}, h^{\tilde{\tilde{r}}}) \to 1 | r + \tilde{r} = \tilde{\tilde{r}}] - \Pr[\mathcal{D}(u, v, h, u^r, v^{\tilde{r}}, h^{\tilde{\tilde{r}}}) \to 1 | r + \tilde{r} \neq \tilde{\tilde{r}}]|$ *is negligible.*

Theorem 6. *The above construction is an adaptively secure* k*-resilient identity-based KEM if the decision linear assumption on* $\mathbb{G}$ *holds.*

Proof. Given an adversary $\mathcal{A}$ which attacks adaptive security against the above scheme, we bound its advantage by constructing the simulator below:

Setup. The simulator $\mathcal{B}$ receives an instance $(u, v, h, u^r, v^{\tilde{r}}, h^{\tilde{\tilde{r}}})$ of the decision linear problem, where $\tilde{\tilde{r}}$ is either $r + \tilde{r}$ or an independently random element of $\mathbb{Z}_p$. The simulator generates random polynomials $\{d_i(x) = d_{i,0} + \cdots + \alpha_{i,k}x^k, d'_i(x) = d'_{i,0} + \cdots + d'_{i,k}x^k, d''_i(x) = d''_{i,0} + \cdots + d''_{i,k}x^k\}_{i\in[l]}$ of degree k, sets $D_{i,j} \leftarrow u^{d_{i,j}} h^{d''_{i,j}}$ and $\tilde{D}_{i,j} \leftarrow v^{d'_{i,j}} h^{d''_{i,j}}$ for all $i \in [1, l]$ and $j \in [0, k]$, and runs $\mathcal{A}$ with input $par = (u, v, h, \{D_{i,j}, \tilde{D}_{i,j}\}_{i\in[1,l],j\in[0,k]})$.

Key Extraction Query (Phase I). When $\mathcal{A}$ queries an identity ID, $\mathcal{B}$ returns $dk_{ID} = \{d_i(ID), d'_i(ID), d''_i(ID)\}_{i\in[l]}$.

Challenge. When $\mathcal{A}$ requests a challenge for an identity ID^*, $\mathcal{B}$ computes $C^* = (u^r, v^{\tilde{r}}, h^{\tilde{\tilde{r}}})$ and $K^* = (K_1^*, \ldots, K_l^*) = ((u^r)^{d_1(ID^*)}(v^{\tilde{r}})^{d'_1(ID^*)}(h^{\tilde{\tilde{r}}})^{d''_1(ID^*)}, \ldots, (u^r)^{d_l(ID^*)}(v^{\tilde{r}})^{d'_l(ID^*)}(h^{\tilde{\tilde{r}}})^{d''_l(ID^*)})$. This C^* and K^* are given to $\mathcal{A}$ as a challenge.

Key Extraction Query (Phase II). Again, $\mathcal{A}$ may request a decryption key for ID and $\mathcal{B}$ responds as before.

Guess. Finally $\mathcal{A}$ outputs a bit b' and $\mathcal{B}$ outputs the same bit.

When $\tilde{\tilde{r}} = r + \tilde{r}$, a simple calculation shows that K^* is the real session key encapsulated in C^*. Otherwise when $\tilde{\tilde{r}} \neq r + \tilde{r}$, we will show that K^* distributes independently from all other values seen by $\mathcal{A}$. To see this, let ID_1, …, ID_k be the decapsulation key queries issued by $\mathcal{A}$ during the simulation, and observe

that queries reveal function values $d_i(ID_j)$, $d'_i(ID_j)$, $d''_i(ID_j)$ to $\mathcal{A}$, but $d_i(ID^*)$, $d'_i(ID^*)$, and $d''_i(ID^*)$ are not revealed. However, *par* further reveals the value $d_i(ID^*) + \alpha d''_i(ID^*)$ and $d'_i(ID^*) + \beta d''_i(ID_*)$, where $u = g^\alpha$ and $v = g^\beta$. The equations $\mathcal{A}$ can observe is represented as

$$\begin{pmatrix} \begin{pmatrix} dk_{ID_1} \end{pmatrix} \\ \vdots \\ \begin{pmatrix} dk_{ID_k} \end{pmatrix} \\ \log_g u^{x_i(ID^*)} h^{x''_i(ID^*)} \\ \log_g v^{x'_i(ID^*)} h^{x''_i(ID^*)} \\ \log_g K^*_i \end{pmatrix} = \begin{pmatrix} 1 & & & & & & & & \\ & 1 & & & & & & & \\ & & 1 & & & & & & \\ & & & \ddots & & & & & \\ & & & & 1 & & & & \\ & & & & & 1 & & & \\ & & & & & & 1 & & \\ & & & & & & & \alpha & 1 \\ & & & & & & & \beta & 1 \\ & & & & & & r\alpha & \tilde{r}\beta & \tilde{\tilde{r}} \end{pmatrix} \begin{pmatrix} d_i(ID_1) \\ d'_i(ID_1) \\ d''_i(ID_1) \\ \vdots \\ d_i(ID_k) \\ d'_i(ID_k) \\ d''_i(ID_k) \\ d_i(ID^*) \\ d'_i(ID^*) \\ d''_i(ID^*) \end{pmatrix},$$

where this matrix is non-singular, and hence K^* is uniformly distributed. All these facts justify the fact that the quantity

$$\Pr[\mathcal{A} \to 1 \mid K^* \text{ is real}] - \Pr[\mathcal{A} \to 1 \mid K^* \text{ is random}]$$

is equal to

$$\Pr[\mathcal{B}(u, v, h, u^r, v^{\tilde{r}}, h^{\tilde{\tilde{r}}}) \to 1 \mid r + \tilde{r} = \tilde{\tilde{r}}] \\ - \Pr[\mathcal{B}(u, v, h, u^r, v^{\tilde{r}}, h^{\tilde{\tilde{r}}}) \to 1 \mid r + \tilde{r} \neq \tilde{\tilde{r}}].$$

The decision linear assumption says that the latter is negligible, and so is the former, which is what we wanted. □

6.2 Other Building Blocks

Other building blocks are instantiated as follows.

Groth-Sahai Proofs [29,26]. This is an efficient non-interactive proof system for groups with a bilinear map. This proof system is able to demonstrate quite broad types of algebraic equations hold in a zero-knowledge manner, and is useful to avoid an expensive blowup from general NIZK techniques.

Abe-Haralambiev-Ohkubo Signature [4,2]. This is a structure-preserving signature, in the sense that the signing and verification procedure has no use of non-algebraic operation. This property is essential when the scheme is used together with Groth-Sahai proofs, due to the restriction that Groth-Sahai proofs are unable to treat a kind of non-algebraic relation such as hashing.

The Decision Linear Variant of Cramer-Shoup [41]. Groth-Sahai proofs are highly relying on its use of pairing, and thus we can no longer expect the DDH assumption to hold in our setting. This is why we need to modify the Cramer-Shoup encryption to use the DLIN assumption instead of the classical DDH assumption. Such a DLIN variant of the Cramer-Shoup encryption was proposed by Shacham [41], but we further modify the Shacham's scheme to be tag-based for adopting the one-time signature technique and to be a key encapsulation mechanism for further efficiency than in a direct use of public-key encryption[3].

Encoding Functions. The first encoding function has to encode (i, vk_i) into the message space of the Abe-Haralambiev-Ohkubo scheme. The verification key vk_i is already represented by sixteen elements of $\mathbb{G}$. The identity i of a signer is an integer, but it can be efficiently encoded as g^i. Notice that decoding is also efficient, because the number of group members is polynomial, and so is i. The same thing holds for the second encoding function. In this case, $(i, vk_i, cert_i, s)$ can be encoded as thirty-one group elements of $\mathbb{G}$.[4] Because the Shacham PKE, as well as the Heng-Kurosawa IBE, can be modified to have the session key space $\mathbb{G}^{31}$, the identity encoding function suffices for this purpose. Another important point is that $\langle i, vk_i, cert_i, s\rangle$ is masked by a session key via the group operation of $\mathbb{G}$ for keeping the structure-preserving property, which enables us to adopt Groth-Sahai proofs.

Theorem 7. *When instantiating our construction in Fig. 2 with our decision linear variant of the Heng-Kurosawa k-resilient IBE, the Groth-Sahai proof, the decision linear variant of the Cramer-Shoup encryption, the Abe-Haralambiev-Ohkubo signature scheme, and the one-time signature scheme from the Okamoto identification scheme [38] via the transformation due to Bellare and Shoup [9], the resulting scheme satisfies opener anonymity with k-bounded tokens, admitter anonymity, and traceability.*

6.3 Efficiency

Finally we give a brief efficiency comparison between the proposed scheme and previous group signatures (without message-dependent opening capability).

In the instantiation in Theorem 7, a signature contains 501 elements of $\mathbb{G}$ and 2 elements of $\mathbb{Z}_p$. For a reference, we remark that the group signature of Groth [28] has a signature that consists of 52 elements of $\mathbb{G}$ and 1 elements of $\mathbb{Z}_p$. The message-dependent opening capability is achieved by roughly 10 times blowup

[3] A possible alternative choice here is Kiltz's tag-based encryption [35], which could reduce the size of NIZK proofs due to its public verifiability. One drawback of this scheme is that, to the best of the authors' knowledge, Kiltz's encryption does not allow encrypting multiple group elements with constant ciphertext overhead, while the Cramer-Shoup scheme (and its DLIN variant by Shacham) allow such a modification. See Sect. A.2 for details of this modification.

[4] These thirty-one elements come form one element for g^i, sixteen elements for the verification key vk_i, seven elements for $cert_i$, and seven elements for s.

of the signature size (The Groth scheme allows dynamic joining of members, whereas ours does not, though). From this evaluation, we see that our scheme is fairly practical, or at least implementable in a real system.

Acknowledgment. The authors would like to thank anonymous reviewers for their invaluable comments.

References

1. Abdalla, M., Warinschi, B.: On the Minimal Assumptions of Group Signature Schemes. In: López, J., Qing, S., Okamoto, E. (eds.) ICICS 2004. LNCS, vol. 3269, pp. 1–13. Springer, Heidelberg (2004)
2. Abe, M., Fuchsbauer, G., Groth, J., Haralambiev, K., Ohkubo, M.: Structure-Preserving Signatures and Commitments to Group Elements. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 209–236. Springer, Heidelberg (2010)
3. Abe, M., Gennaro, R., Kurosawa, K., Shoup, V.: Tag-KEM/DEM: A New Framework for Hybrid Encryption and A New Analysis of Kurosawa-Desmedt KEM. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 128–146. Springer, Heidelberg (2005)
4. Abe, M., Haralambiev, K., Ohkubo, M.: Signing on elements in bilinear groups for modular protocol design. Cryptology ePrint Archive, Report 2010/133 (2010), `http://eprint.iacr.org/`
5. Ateniese, G., Camenisch, J., Hohenberger, S., de Medeiros, B.: Practical group signatures without random oracles. Cryptology ePrint Archive, Report 2005/385 (2005), `http://eprint.iacr.org/`
6. Ateniese, G., Camenisch, J.L., Joye, M., Tsudik, G.: A Practical and Provably Secure Coalition-Resistant Group Signature Scheme. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 255–270. Springer, Heidelberg (2000)
7. Bellare, M., Micciancio, D., Warinschi, B.: Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 614–629. Springer, Heidelberg (2003)
8. Bellare, M., Shi, H., Zhang, C.: Foundations of Group Signatures: The Case of Dynamic Groups. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 136–153. Springer, Heidelberg (2005)
9. Bellare, M., Shoup, S.: Two-tier signatures from the Fiat–Shamir transform, with applications to strongly unforgeable and one-time signatures. IET Information Security 2(2), 47–63 (2008)
10. Bichsel, P., Camenisch, J., Neven, G., Smart, N.P., Warinschi, B.: Get Shorty via Group Signatures without Encryption. In: Garay, J.A., De Prisco, R. (eds.) SCN 2010. LNCS, vol. 6280, pp. 381–398. Springer, Heidelberg (2010)
11. Boneh, D., Boyen, X., Shacham, H.: Short Group Signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
12. Boneh, D., Papakonstantinou, P.A., Rackoff, C., Vahlis, Y., Waters, B.: On the impossibility of basing identity based encryption on trapdoor permutations. In: 49th Annual IEEE Symposium on Foundations of Computer Science, pp. 283–292 (2008)

13. Boneh, D., Shacham, H.: Group signatures with verifier-local revocation. In: 11th ACM Conference on Computer and Communications Security, pp. 168–177. ACM, New York (2004)
14. Boyen, X., Waters, B.: Compact Group Signatures Without Random Oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 427–444. Springer, Heidelberg (2006)
15. Boyen, X., Waters, B.: Full-Domain Subgroup Hiding and Constant-Size Group Signatures. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 1–15. Springer, Heidelberg (2007)
16. Camenisch, J., Chandran, N., Shoup, V.: A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. Cryptology ePrint Archive, Report 2008/375 (2008), http://eprint.iacr.org/
17. Camenisch, J., Chandran, N., Shoup, V.: A Public Key Encryption Scheme Secure against Key Dependent Chosen Plaintext and Adaptive Chosen Ciphertext Attacks. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 351–368. Springer, Heidelberg (2009)
18. Camenisch, J., Lysyanskaya, A.: Signature Schemes and Anonymous Credentials from Bilinear Maps. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 56–72. Springer, Heidelberg (2004)
19. Camenisch, J., Michels, M.: Separability and Efficiency for Generic Group Signature Schemes (Extended Abstract). In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 413–785. Springer, Heidelberg (1999)
20. Chaum, D., van Heyst, E.: Group Signatures. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (1991)
21. Cramer, R., Shoup, V.: A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998)
22. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM Journal on Computing 33(1), 167–226 (2003)
23. Delerablée, C., Pointcheval, D.: Dynamic Fully Anonymous Short Group Signatures. In: Nguyên, P.Q. (ed.) VIETCRYPT 2006. LNCS, vol. 4341, pp. 193–210. Springer, Heidelberg (2006)
24. Dodis, Y., Katz, J., Xu, S., Yung, M.: Key-Insulated Public Key Cryptosystems. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 65–82. Springer, Heidelberg (2002)
25. Furukawa, J., Imai, H.: An Efficient Group Signature Scheme from Bilinear Maps. In: Boyd, C., Nieto, J.G. (eds.) ACISP 2005. LNCS, vol. 3574, pp. 455–467. Springer, Heidelberg (2005)
26. Ghadafi, E., Smart, N.P., Warinschi, B.: Groth–Sahai Proofs Revisited. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 177–192. Springer, Heidelberg (2010)
27. Groth, J.: Simulation-Sound NIZK Proofs for a Practical Language and Constant Size Group Signatures. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 444–459. Springer, Heidelberg (2006)
28. Groth, J.: Fully Anonymous Group Signatures Without Random Oracles. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 164–180. Springer, Heidelberg (2007)
29. Groth, J., Sahai, A.: Efficient Non-interactive Proof Systems for Bilinear Groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)

30. Heng, S.-H., Kurosawa, K.: k-Resilient Identity-Based Encryption in the Standard Model. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 67–80. Springer, Heidelberg (2004)
31. Kiayias, A., Tsiounis, Y., Yung, M.: Traceable Signatures. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 571–589. Springer, Heidelberg (2004)
32. Kiayias, A., Yung, M.: Group signatures: Provable security, efficient constructions and anonymity from trapdoor-holders. Cryptology ePrint Archive, Report 2004/076 (2004), `http://eprint.iacr.org/`
33. Kiayias, A., Yung, M.: Group Signatures with Efficient Concurrent Join. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 198–214. Springer, Heidelberg (2005)
34. Kilian, J., Petrank, E.: Identity Escrow. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 169–185. Springer, Heidelberg (1998)
35. Kiltz, E.: Chosen-Ciphertext Security from Tag-Based Encryption. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 581–600. Springer, Heidelberg (2006)
36. MacKenzie, P.D., Reiter, M.K., Yang, K.: Alternatives to Non-malleability: Definitions, Constructions, and Applications (Extended Abstract). In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 171–190. Springer, Heidelberg (2004)
37. Ohtake, G., Fujii, A., Hanaoka, G., Ogawa, K.: On the Theoretical Gap between Group Signatures with and without Unlinkability. In: Preneel, B. (ed.) AFRICACRYPT 2009. LNCS, vol. 5580, pp. 149–166. Springer, Heidelberg (2009)
38. Okamoto, T.: Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 31–53. Springer, Heidelberg (1993)
39. Sahai, A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: 40th Annual Symposium on Foundations of Computer Science, pp. 543–553. IEEE Computer Society (1999)
40. Schwartz, E.J., Brumley, D., McCune, J.M.: A contractual anonymity system. In: NDSS 2010. The Internet Society (2010)
41. Shacham, H.: A Cramer-Shoup encryption scheme from the linear assumption and from progressively weaker linear variants. Cryptology ePrint Archive, Report 2007/074 (2007), `http://eprint.iacr.org/`
42. Shoup, V., Gennaro, R.: Securing Threshold Cryptosystems against Chosen Ciphertext Attack. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 1–16. Springer, Heidelberg (1998)

A Building Blocks and Their Security Proofs

In the following, let $\mathbb{G}$ and $\mathbb{G}_T$ be groups of a prime order p, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map.

A.1 Abe-Haralambiev-Ohkubo Signature

The Abe-Haralambiev-Ohkubo signature scheme is as follows [2,4]:

Key Generation. The verification key is $vk = (g'', h'', g', h', \{g_i, h_i\}_{i\in[l]}, a_0, \tilde{a}_0, b_0, \tilde{b}_0, a_1, \tilde{a}_1, b_1, \tilde{b}_1)$, where g', $h' \in \mathbb{G} \setminus \{1\}$, $g_i \leftarrow g'^{\gamma_i}$ and $h_i \leftarrow h'^{\delta_i}$ for random $\gamma_i, \delta_i \leftarrow \mathbb{Z}_p^*$ for $i \in [l]$, $g'' \leftarrow g'^{\gamma''}$, $h'' \leftarrow h'^{\delta''}$ for random $\gamma'', \delta'' \leftarrow \mathbb{Z}_p^*$, $(a_0, \tilde{a}_0, a_1, \tilde{a}_1) \leftarrow \mathsf{Extend}(g', g^\alpha)$ for a random $\alpha \leftarrow \mathbb{Z}_p^*$, and $(b_0, \tilde{b}_0, b_1, \tilde{b}_1) \leftarrow \mathsf{Extend}(h', g^\beta)$ for a random $\beta \leftarrow \mathbb{Z}_p^*$. The signing key is $sk = (\alpha, \beta, \gamma'', \delta'', \{\gamma_i, \delta_i\}_{i\in[l]})$.

Signing. For a message $(m_1, \ldots, m_l) \in \mathbb{G}^l$, choose randomly ζ, ρ, τ, φ, ω from $\mathbb{Z}_p$, compute $z = \tilde{g}^\zeta$, $r = \tilde{g}^{\alpha-\rho\tau-\gamma_z\zeta} \prod_{i=1}^l m_i^{-\gamma_i}$, $s = g'^\rho$, $t = \tilde{g}^\tau$, $u = \tilde{g}^{\beta-\varphi\omega-\delta_z\zeta} \prod_{i=1}^l m_i^{-\delta_i}$, $v = h_r^\varphi$, and $w = \tilde{g}^\omega$, and output (z, r, s, t, u, v, w) as a signature.

Verification. For a pair $(m, \sigma) = ((m_1, \ldots, m_l), (z, r, s, t, u, v, w))$ of a signature and a message, verify two equations $e(a_0, \tilde{a}_0)e(a_1, \tilde{a}_1) = e(g'', z)\, e(g', r)e(s, t) \prod_{i=1}^k e(g_i, m_i)$ and $e(b_0, \tilde{b}_0)e(b_1, \tilde{b}_1) = e(h'', z)e(h', u)e(v, w)\, \prod_{i=1}^l e(h_i, m_i)$. If both equations hold, output $\top$. Otherwise output $\bot$.

Here, $\mathsf{Extend}(g, h)$ is the algorithm that takes two group elements g and h, picks random $x \in \mathbb{G}$ and $r \in \mathbb{Z}_p$, and outputs $(\mathsf{Rand}(gx^r, h), \mathsf{Rand}(x, h^{-r}))$. Algorithm $\mathsf{Rand}(g, h)$, when $g \neq 1$ and $h \neq 1$, outputs $(g^s, h^{1/s})$ for random $s \in \mathbb{Z}_p^*$. When $g = 1$ or $h = 1$, it outputs $(1, 1)$ with probability $1/(2p-1)$, otherwise outputs one of $(1, x)$ or $(x, 1)$ with probability $1/2$ for random $x \in \mathbb{G} \setminus \{1\}$.

A.2 Shacham's Variant of Cramer-Shoup Encryption

Shacham [41] proposed a variant of the Cramer-Shoup encryption scheme [21,22] modified to be based on the decision linear assumption. The scheme below further modifies the Shacham's variants in two points: (1) Used as a key encapsulation mechanisms and (2) modified to encapsulate a long session key in a constant-size ciphertext. This modified Shacham's variant is as follows:

Key Generation. The public key is $pk = (u, v, h, X = u^x h^{x''}, \tilde{X} = v^{x'} h^{x''}, Y = u^y h^{y''}, \tilde{Y} = v^{y'} h^{y''}, \{Z_i = u^{z_i} h^{z'_i}, \tilde{Z}_i = v^{z'_i} h^{z''_i}\}_{i\in[l]})$, whose corresponding secret key is $dk = (x, x', x'', y, y', y'', \{z_i, z'_i, z''_i\}_{i\in[l]})$.

Encrypt. To encapsulate a session key with a tag t, choose random r and $\tilde{r}$ from $\mathbb{Z}_p$ and compute a ciphertext as $C_{\mathrm{PKE}} = (u^r, v^{\tilde{r}}, h^{r+\tilde{r}}, (XY^t)^r(\tilde{X}\tilde{Y}^t)^{\tilde{r}})$. The session key is $(Z_1^r \tilde{Z}_1^{\tilde{r}}, \ldots, Z_l^r \tilde{Z}_l^{\tilde{r}})$.

Decrypt. To decapsulate a ciphertext (c_1, c_2, c_3, c_4) with a tag t, verify whether $c_1^{x+ty} c_2^{x'+ty'} c_3^{x''+ty''} = c_4$ holds. If it does not hold, output $\bot$, and otherwise output $(c_1^{z_1} c_2^{z'_1} c_3^{z''_1}, \ldots, c_1^{z_l} c_2^{z'_l} c_3^{z''_l})$.

B Security Proofs for the Construction in Sect. 5

B.1 Proof of Theorem 3

Proof. Let $\mathcal{A}$ be an opener anonymity adversary against the proposed scheme. Let $\mathbf{OAnonym}_{\mathcal{A}}$ be the random variable that is 1 when $\mathcal{A}$ correctly guesses the

bit b in the opener anonymity game and is 0 when it does not. Let $\mathbf{OAnonym}'_{\mathcal{A}}$ be a similar random variable with one exception that the common reference string used in the scheme is generated with the zero-knowledge simulator $\mathcal{S}_1$. This change does not affect the probability that the adversary $\mathcal{A}$ wins the game, that is, $|\Pr[\mathbf{OAnonym}_{\mathcal{A}} = 1] - \Pr[\mathbf{OAnonym}'_{\mathcal{A}} = 1]|$ is negligible, due to the zero-knowledge property of the underlying non-interactive proof system. We then show that $|\Pr[\mathbf{OAnonym}'_{\mathcal{A}} = 1] - 1/2|$ is negligible, which concludes the proof.

We construct an adversary $\mathcal{B}$ which attacks the underlying (k-resilient) IBE scheme, and then we relate its success probability to that of $\mathcal{A}$ (in the experiment $\mathbf{OAnonym}'_{\mathcal{A}}$) to obtain the desired bound. The construction of $\mathcal{B}$ is as follows:

Setup. The adversary $\mathcal{B}$ is given as input the master public key par for the identity-based KEM. The adversary $\mathcal{B}$ then generates the rest of a group public key gpk as $(vk_{\mathrm{issue}}, sk_{\mathrm{issue}}) \leftarrow \mathsf{SigKg}(1^\lambda)$, $(pk, dk) \leftarrow \mathsf{TKg}(1^\lambda)$, $(\Sigma, \tau) \leftarrow \mathcal{S}_1(1^\lambda)$, generates user signing keys $(vk_i, sk_i) \leftarrow \mathsf{SigKg}(1^\lambda)$ and their certificates $cert_i \leftarrow \mathsf{Sign}_{sk_{\mathrm{issue}}}(\langle i, vk_i \rangle)$ for all $i \in [n]$. The adversary then sets gpk to $(vk_{\mathrm{issue}}, par, pk, \Sigma)$, sets gsk_i to $(i, vk_i, cert_i, sk_i)$, and run $\mathcal{A}$ with input $(gpk, dk, \{gsk_i\}_{i\in[n]})$.

Token Query (Phase I). When $\mathcal{A}$ makes a token query for a message M, $\mathcal{B}$ makes a key extraction query for M (as an identity) to obtain a decryption key dk_M, and responds $\mathcal{A}$ with dk_M.

Challenge. When $\mathcal{A}$ requests a challenge for (i_0, i_1, M^*), $\mathcal{B}$ proceeds as follows: $\mathcal{B}$ computes two signatures $s_0 \leftarrow \mathsf{Sign}_{vk_{i_0}}(M^*)$ and $s_1 \leftarrow \mathsf{Sign}_{vk_{i_1}}(M^*)$ of the group members i_0 and i_1, generates a one-time signature key pair $(vk^*_{\mathrm{OT}}, sk^*_{\mathrm{OT}})$, and requests a challenge for an identity M^*. Then $\mathcal{B}$ receives a challenge $(C^*_{\mathrm{IBE}}, K^*_{\mathrm{IBE}})$, computes a ciphertext $(C_{\mathrm{PKE}}, K_{\mathrm{PKE}}) \leftarrow \mathsf{TEnc}_{pk}(vk^*_{\mathrm{OT}})$, further computes $\chi^* \leftarrow \langle i_b, vk_{i_b}, cert_{i_b}, s_b \rangle \odot K_{\mathrm{PKE}} \odot K^*_{\mathrm{IBE}}$ for a random bit b, generates a simulated proof π^* with a token τ, and signs $\langle C^*_{\mathrm{PKE}}, C^*_{\mathrm{IBE}}, \chi^*, \pi^* \rangle$ with the one-time signing key sk^*_{OT} to obtain σ^*_{OT}. Finally $\mathcal{B}$ sends $(vk^*_{\mathrm{OT}}, C^*_{\mathrm{PKE}}, C^*_{\mathrm{IBE}}, \chi^*, \pi^*, \sigma^*_{\mathrm{OT}})$ to $\mathcal{A}$ as a challenge.

Token Query (Phase II). The adversary $\mathcal{A}$ continue to issue token queries, and they are answered by $\mathcal{B}$ as before.

Guess. At last $\mathcal{A}$ outputs a bit b', and $\mathcal{B}$ outputs 1 if and only if $b' = b$, otherwise outputs 0.

When the challenger gives the real session key (i.e. C^*_{IBE} is decrypted into K^*_{IBE}), $\mathcal{B}$ perfectly simulates the experiment $\mathbf{OAnonym}'_{\mathcal{A}}$, whereas when a random session key is given, the information on the bit b is perfectly hidden from $\mathcal{A}$. This fact justifies the equality below:

$$\Pr[\mathbf{OAnonym}'_{\mathcal{A}} = 1] - \frac{1}{2} = \Pr[\mathcal{B} \to 1 \mid K^*_{\mathrm{PKE}} \text{ is real}] - \Pr[\mathcal{B} \to 1 \mid K^*_{\mathrm{PKE}} \text{ is random}].$$

Due to the security of the k resilient identity-based KEM, the right hand side is negligible, and it completes the proof. □

B.2 Proof of Theorem 4

Proof. Let $\mathcal{A}$ be an admitter anonymity adversary against the proposed scheme. Let $\mathbf{AAnonym}_{\mathcal{A}}$ be a random variable that indicates $\mathcal{A}$ correctly guesses the bit in the admitter anonymity game. Let $\mathbf{AAnonym}'_{\mathcal{A}}$ be a similar random variable with the exception that the common reference string is replaced to that for simulation. Due to the zero-knowledge property of the proof system, $\mathcal{A}$'s success probability does not change non-negligibly, that is, $|\Pr[\mathbf{AAnonym}_{\mathcal{A}} = 1] - \Pr[\mathbf{AAnonym}'_{\mathcal{A}} = 1]|$ is negligible. We then show that $|\Pr[\mathbf{AAnonym}'_{\mathcal{A}} = 1] - 1/2|$ is negligible, which concludes the actual proof.

We construct an adversary $\mathcal{B}$ which attacks the underlying tag-based KEM. The construction of $\mathcal{B}$ is as follows:

Setup. The adversary $\mathcal{B}$ first runs $\mathsf{SigKg}^{\mathrm{OT}}(1^\lambda)$ to generate a verification/signing key pair $(vk^*_{\mathrm{OT}}, sk^*_{\mathrm{OT}})$, outputs vk^*_{OT} as a target tag, and then receives the public key pk of the tag-based KEM. The adversary $\mathcal{B}$ then generates the rest of a group public key as $(vk_{\mathrm{issue}}, sk_{\mathrm{issue}}) \leftarrow \mathsf{SigKg}(1^\lambda)$, $(par, mk) \leftarrow \mathsf{ISetup}(1^\lambda)$, $(\Sigma, \tau) \leftarrow \mathcal{S}_1(1^\lambda)$, user signing keys $(vk_i, sk_i) \leftarrow \mathsf{SigKg}(1^\lambda)$ for all $i \in [n]$, and their certificates $cert_i \leftarrow \mathsf{Sign}_{sk_{\mathrm{issue}}}(\langle i, vk_i\rangle)$ for all $i \in [n]$. The adversary $\mathcal{B}$ then sets $gpk \leftarrow (vk_{\mathrm{issue}}, pk, par, \Sigma)$ and $gsk_i \leftarrow (i, vk_i, cert_i, sk_i)$ and runs $\mathcal{A}$ with input $(gpk, mk, \{gsk_i\}_{i\in[n]})$.

Open Query (Phase I). When the adversary $\mathcal{A}$ submits an open query for a signature $(vk_{\mathrm{OT}}, C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi, \pi, \sigma_{\mathrm{OT}})$ and a message M, the adversary $\mathcal{B}$ responds as follows: (i) when $vk_{\mathrm{OT}} \neq vk^*_{\mathrm{OT}}$, $\mathcal{B}$ makes a decapsulation query for the ciphertext C_{PKE} with a tag vk_{OT} to obtain a session key K_{PKE} (note that this query is legitimate), and then extracts a user decryption key dk_M (of an identity-based KEM) from mk, decrypts C_{IBE} with dk_M to obtain a session key K_{IBE}, and verifies whether $\mathsf{Verify}^{\mathrm{OT}}_{vk_{\mathrm{OT}}}(\langle C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi, \pi\rangle, \sigma_{\mathrm{OT}}) = 1$ and $V_{\mathrm{NIZK}}(\ldots) = 1$ hold. If both of them hold, $\mathcal{B}$ further computes $\langle i, vk, cert, s\rangle \leftarrow \chi \odot K^{-1}_{\mathrm{IBE}} \odot K^{-1}_{\mathrm{PKE}}$ and responds with i. Otherwise $\mathcal{B}$ responds with $\perp$. (ii) When $vk_{\mathrm{OT}} = vk^*_{\mathrm{OT}}$, if $\mathsf{Verify}_{vk^*_{\mathrm{OT}}}(\langle C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi, \pi\rangle) = \perp$, $\mathcal{B}$ responds with $\perp$. Otherwise $\mathcal{B}$ aborts and outputs a random b'.

Challenge. At some time $\mathcal{A}$ requests a challenge for (i_0, i_1, M^*), $\mathcal{B}$ computes a challenge as follows: $\mathcal{B}$ generates signatures $s_b \leftarrow \mathsf{Sign}_{sk_{i_b}}(M^*)$ for a random bit b, requests a challenge to obtain $(C^*_{\mathrm{PKE}}, K^*_{\mathrm{PKE}})$, generates a ciphertext and a session key as $(C^*_{\mathrm{IBE}}, K_{\mathrm{IBE}}) \leftarrow \mathsf{IEnc}_{par}(M^*)$, computes $\chi^* \leftarrow \langle i_b, vk_{i_b}, cert_{i_b}, s_{i_b}\rangle \odot K^*_{\mathrm{PKE}} \odot K_{\mathrm{IBE}}$, and generates a fake proof π^*. Finally $\mathcal{B}$ signs $\langle vk^*_{\mathrm{OT}}, C^*_{\mathrm{PKE}}, C^*_{\mathrm{IBE}}, \chi^*, \pi^*\rangle$ with the one-time signing key sk^*_{OT} to obtain σ^*_{OT} and sends $(vk^*_{\mathrm{OT}}, C^*_{\mathrm{PKE}}, C^*_{\mathrm{IBE}}, \chi^*, \pi^*, \sigma^*_{\mathrm{OT}})$ to $\mathcal{A}$.

Open Query (Phase II). Again $\mathcal{A}$ submits more open queries and $\mathcal{B}$ responds as before.

Guess. When $\mathcal{A}$ outputs a bit b, $\mathcal{B}$ outputs 1 if and only if $b' = b$, otherwise outputs 0.

Let F denote the event that the adversary $\mathcal{A}$ submits an open query $(vk_{\mathrm{OT}}, C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \pi, \sigma_{\mathrm{OT}})$ where $vk_{\mathrm{OT}} = vk^*_{\mathrm{OT}}$ and $\mathsf{Verify}_{vk_{\mathrm{OT}}}(\langle C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \pi\rangle) = \top$. Due to its perfect simulation of the experiment $\mathbf{AAnonym}'_{\mathcal{A}}(k, n)$ (with only

exception of aborting in the event F), when given the real session key $K^*_{\mathcal{PKE}}$, $\mathcal{B}$ outputs 1 whenever $\mathcal{A}$ successfully guesses a bit and the event F does not occur. In addition, when given a random session key, $\mathcal{B}$ gives no information on the bit b to $\mathcal{A}$. The inequality below can be obtained from these two facts:

$$\begin{aligned}
&\left|\Pr[\mathbf{AAnonym}'_{\mathcal{A}} = 1] - \frac{1}{2}\right| \\
&\quad = \left|\Pr[\mathbf{AAnonym}'_{\mathcal{A}} = 1 \wedge F] + \Pr[\mathbf{AAnonym}'_{\mathcal{A}} = 1 \wedge \neg F] - \frac{1}{2}\right| \\
&\quad \leq \left|\Pr[F] + \Pr[\mathbf{AAnonym}'_{\mathcal{A}} = 1 \wedge \neg F] - \frac{1}{2}\right| \\
&\quad \leq \Pr[F] + |\Pr[\mathcal{B} \to 1 \mid K^*_{\mathrm{PKE}} \text{ is real}] - \Pr[\mathcal{B} \to 1 \mid K^*_{\mathrm{PKE}} \text{ is random}]|
\end{aligned}$$

Finally we prove $\Pr[F]$ is negligible to complete the proof.

To prove $\Pr[F]$ is negligible, we will construct another adversary $\mathcal{F}$, which attacks strong unforgeability of the one-time signature scheme and relate its success probability with the probability of the event F. The construction of $\mathcal{F}$ is as follows:

Setup. The adversary $\mathcal{F}$ first receives a verification key vk^*_{OT} for the one-time signature scheme. The adversary then runs $\mathsf{GKg}(1^\lambda, 1^n, 1^k)$ to obtain a group public key $gpk = (vk_{\mathrm{issue}}, pk, par, \Sigma)$, the opening key ok, the message-specification key msk, and user signing keys $gsk_i = (i, vk_i, cert_i, sk_i)$ for all $i \in [n]$.

Open Query (Phase I). Queries are answered with the opening key ok and the message-specifying key msk. In addition, when $\mathcal{A}$ queries a group signature $(vk_{\mathrm{OT}}, C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi, \pi, \sigma_{\mathrm{OT}})$ in which $vk_{\mathrm{OT}} = vk^*_{\mathrm{OT}}$ and $\mathsf{Verify}_{vk_{\mathrm{OT}}}(\langle C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi, \pi\rangle, \sigma_{\mathrm{OT}}) = \top$, $\mathcal{F}$ stops the simulation and outputs this $(\langle C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi, \pi\rangle, \sigma_{\mathrm{OT}})$ as a forgery.

Challenge. To respond to the challenge request (i_0, i_1, M^*), $\mathcal{F}$ chooses a random bit b and generate a group signature $(vk^*_{\mathrm{OT}}, C^*_{\mathrm{PKE}}, C^*_{\mathrm{IBE}}, \chi^*, \pi^*, \sigma^*_{\mathrm{OT}})$ in the way exactly same to the construction with one exception that σ^*_{OT} is obtained by querying $\langle C^*_{\mathrm{PKE}}, C^*_{\mathrm{IBE}}, \chi^*, \pi^*\rangle$ to the signing oracle.

Open Query (Phase II). Further open queries are answered as in the phase I.

Guess. If $\mathcal{A}$ outputs a guess and halts, $\mathcal{F}$ halts without outputting a forgery.

Whenever the event F happens, this adversary $\mathcal{F}$ successfully outputs a forgery and wins the game (Because $(C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi, \pi, \sigma_{\mathrm{OT}})$ must be different from $(C^*_{\mathrm{PKE}}, C^*_{\mathrm{IBE}}, \chi^*, \pi^*, \sigma^*_{\mathrm{OT}})$, and it consists a legitimate strong forgery). Then we can conclude $\Pr[F]$ is negligible, because of the security of the underlying one-time signature scheme. □

B.3 Proof of Theorem 5

Proof. Let $\mathcal{A}$ be a traceability adversary against the proposed scheme. We first classify successful forgery that $\mathcal{A}$ may produce.

The forgery is opened to $\perp$: In this case, either C_{PKE} or C_{IBE} is invalid (decapsulated to $\perp$) or $\chi \odot K_{\mathrm{IBE}}^{-1} \odot K_{\mathrm{PKE}}^{-1}$ cannot be parsed. In all of these case, π is an invalid proof for a false statement.

The forgery is opened to $i \in \mathbb{N}$: In this case all C_{PKE}, C_{IBE}, and χ have been correctly decrypted, and when $\chi \odot K_{\mathrm{IBE}}^{-1} \odot K_{\mathrm{PKE}}^{-1}$ is parsed into $\langle i', vk', cert', s' \rangle$, either one of the following three cases will hold: (i) $\mathsf{Verify}_{vk_{\mathrm{issue}}}(\langle i', vk' \rangle, cert') = \perp$ or $\mathsf{Verify}_{vk'}(M, s) = \perp$, (ii) $cert'$ is a valid signature, but $\langle i', vk' \rangle$ was not singed at the setup phase, or (iii) $(\langle i, vk' \rangle, cert')$ is the same one generated at the setup phase. Note that in case (i) the proof π is a valid proof for the false statement, in case (ii) $cert'$ is a forgery for the verification key vk_{isuue}, and in case (iii) s' is a forgery for the user verification key vk_i.

To bound the probability that $\mathcal{A}$ outputs a forgery of case (iii), we construct a forger $\mathcal{B}$ against the underlying EUF-CMA signature scheme. The construction of $\mathcal{B}$ is as follows: The forger $\mathcal{B}$ receives a verification key vk, and $\mathcal{B}$ uses this verification key as a user verification key vk_{i^*}, where i^* is randomly chosen by $\mathcal{B}$. Other components of the public key and the secret keys for the group members and the authorities are honestly generated by $\mathcal{B}$. Then $\mathcal{B}$ runs $\mathcal{A}$ with input the group public key gpk, the opener key ok, and the admitter key msk. Signing queries for the user i^* can be simulated with the signing oracle of the underlying scheme, group signing key revealing query for the user i^* cannot be simulated, in which case $\mathcal{B}$ aborts. Other signing queries and key revealing queries can be answered by $\mathcal{B}$ itself. When $\mathcal{A}$ outputs a forgery (M, σ), $\mathcal{B}$ verifies the one-time signature and the non-interactive proof in σ, decrypts ciphertexts in σ to obtains the plaintext $\langle i', vk', cert', s' \rangle$, verifies $cert'$ by vk_{issue} and s' by vk', confirms that $(i', vk') = (i^*, vk)$, and finally outputs (M, s) as a forgery. If one of these procedures fails, $\mathcal{B}$ stops. Since $\mathcal{B}$ is a legitimate forger of the signature scheme and whenever $\mathcal{A}$ produces a forgery of case (iii) also $\mathcal{B}$ does a successful forgery, we obtains a bound for the case (iii).

To bound the probability that $\mathcal{A}$ outputs a forgery of case (ii), we construct another forger $\mathcal{B}'$. This time $\mathcal{B}$ receives a key vk and uses it as a certificate verification key vk_{issue}, and obtains certificates $cert_i$ for all group members $i \in [n]$ by querying the signing oracle of the underlying scheme. Signing queries and key revealing queries issued by $\mathcal{A}$ are correctly answered by $\mathcal{B}$ itself for all group members. When $\mathcal{A}$ outputs (M, σ), $\mathcal{B}$ verifies the validity of σ, confirms that σ contains a certificate forgery, and outputs this forged certificate, otherwise stops. Also in this case $\mathcal{B}$ is a legitimate forger against the underlying scheme, and thus the probability that $\mathcal{A}$ produces a forgery of the case (ii) is bounded.

Since the other cases of forgeries $\mathcal{A}$ may produce contains a valid proof of a false statement, the probability that $\mathcal{A}$ produces such a forgery is bounded due to the underlying non-interactive proof system. □