International Journal of Networking and Computing – www.ijnc.org, ISSN 2185-2847
Volume 13, Number 2, pages 173–194, July 2023

Dynamic Group Signatures with Message Dependent Opening and Non-Interactive Signing

Hiroaki Anada
Department of Software and Information Technology, Faculty of Software and Information Technology, Aomori University
2-3-1 Kobata, Aomori-shi, Aomori, 030-0943 Japan
anada@aomori-u.ac.jp

Masayuki Fukumitsu
Department of Information Security, Faculty of Information Systems, University of Nagasaki
1-1-1 Manabino, Nagayo-cho, Nishisonogi-gun, Nagasaki, 851-2195 Japan
fukumitsu@sun.ac.jp

Shingo Hasegawa
Center for Data-driven Science and Artificial Intelligence, Tohoku University
Multimedia Education and Research Complex, Tohoku University, Kawauchi 41, Aoba-ku, Sendai, 980-8576 Japan
shingo.hasegawa.b7@tohoku.ac.jp

Received: February 14, 2023
Revised: April 28, 2023
Accepted: June 6, 2023
Communicated by Yasuyuki Nogami

Abstract

The group signature with message dependent opening (GS-MDO) is a variant of the group signature in the sense that the opening authority is split into two parties called the opener and the admitter. Most known constructions of GS-MDO consider the *static* model. The only scheme using the *dynamic* model by Sun and Liu has a problem of the anonymity against the admitter in the real-world usage because the signing process requires the interaction between the signer and the admitter. In this paper, we restart the line of research of GS-MDO in the dynamic setting. We introduce the definition of the *dynamic group signature with message dependent opening* (DGS-MDO) with the security requirements and propose a generic construction. By instantiating our construction with appropriate primitives, we can obtain a DGS-MDO scheme with the standard model security, constant signature size and non-interactive signing process.

Keywords: Dynamic Group Signature, Message Dependent Opening, Standard Model

1 Introduction

The group signature (GS) scheme [8] allows members of a group to sign messages on behalf of the group. Although the actual signer is generally intended to be anonymous, the central authority, such as the group manager, can identify the signer by using a trapdoor. Since the power of such an authority is too strong in some applications, it should be restricted appropriately.

Table 1: Comparison of GS-MDO schemes

	GS model	Signature Size	Security model	Assumption	MDO	Signing
[19]	Static	$O(1)$	STD	DLIN, SFP	Bounded	Non-interactive
[17]	Static	$O(1)$	ROM	DBDH, DLIN, q-SDH	Unbounded	Non-interactive
[15]	Static	$O(\log N)$	STD	DLIN, D3DH	Unbounded	Non-interactive
[16]	Static	$O(\log N)$	ROM	LWE, SIS	Unbounded	Non-interactive
[21]	Fully dynamic	$O(\log N)$	ROM	LWE, SIS	Unbounded	Interactive
[ours]	Dynamic	$O(1)$	STD	DLIN, SFP	Bounded	Non-interactive

In the column Signature Size, N denotes the number of members of the group.

As a solution to the above problem, the notion of the *group signature with message dependent opening* (GS-MDO) is proposed by Sakai *et al.* [19]. In GS-MDO, the opening functionality is separated into two authorities called the *opener* and the *admitter*. On the opening operation, the admitter first issues a message-specific token for the target message. The opener then opens the signature corresponding to the message to identify the actual signer by using his secret key and the message-specific token. Once the opener has received a message-specific token, it can open all signatures corresponding to the same message by itself. Note that neither of them can open signatures by itself.

Sakai *et al.* [19] proposed the first GS-MDO scheme with a generic construction from the identity-based encryption (IBE), the tag-based encryption and the non-interactive zero knowledge proof (NIZK). As an instantiation, they also presented the concrete construction from the bilinear group. The security of their scheme is proven in the standard model and its signature size is independent of the group size N. However, the scheme is restricted in the sense that there is a limit to the number of tokens that the admitter can issue, that is, the MDO property is bounded. This is because the scheme uses the Groth-Sahai proof [13] as NIZK, and the GS-proof-compatible IBE only satisfies a slightly weaker security notion [14].

To address the above problem, Ohara *et al.* [17] proposed the GS-MDO scheme which has the unbounded MDO property in the random oracle model (ROM) [4]. Libert and Joye [15] also proposed the unbounded GS-MDO scheme whose security is proven in the standard model. However, the signature size of their scheme is $O(\log N)$, where N is the size of the group. Note that all of the above GS-MDO schemes use the bilinear map. For GS-MDO schemes without the bilinear map, Libert, Mouhartemm and Nguyen [16] constructed the lattice-based GS-MDO scheme. Their scheme is based on the LWE assumption and the SIS assumption, and the security is proven in the random oracle model.

Another point of view is that the GS-MDO schemes above are constructed based on the model of the *static* group signature [3], which fixes the group and its members when the scheme is set up. On the other hand, there are more realistic group signature models, namely the *dynamic* group signature (DGS) [5] and the *fully dynamic* group signature (FDGS) [7]. The DGS allows a group to add members and the FDGS allows a group to add and remove members, respectively, while the static GS cannot change the group.

For the dynamic-type scheme of GS-MDO, Sun and Liu [21] proposed the fully dynamic GS-MDO scheme. Their scheme is a lattice-based scheme and the security is proven under the LWE assumption and the SIS assumption in the random oracle model. While the scheme of [21] achieves the provable security, especially including the anonymity, there is a problem in real-world usage. This is because a signer must send a message to the admitter in the signing process. Then the admitter can infer the actual signer from the pair of the message-signature pair. This means that this GS-MDO scheme does not seem to have adequate anonymity against the admitter, even if the anonymity defined in the model of [21] is attained. Thus it is considered that GS-MDO schemes in the dynamic model are still open.

1.1 Contribution

In this paper, we restart the research line of GS-MDO in the dynamic setting. As the first step, we focus on the dynamic group signature by [5]. We introduce the definition of the *dynamic group sig-*

nature with message dependent opening (DGS-MDO) and the security requirements of DGS-MDO. We also propose a generic construction of DGS-MDO. Our generic construction consists of standard cryptographic primitives such as the digital signature (DS), the key encapsulation mechanism (KEM), the identity-based KEM (ID-KEM) and the non-interactive zero knowledge proof (NIZK). By instantiating each primitive with an appropriate scheme, our construction has competitive properties over existing GS-MDO schemes, as shown in Table 1. Our construction can achieve the standard model security, constant signature size and non-interactive signing process. However, same as the scheme of [19], our scheme has one disadvantage, namely the bounded message opening. The DGS-MDO scheme that supports unbounded message opening is an interesting open question.

1.2 Difference from the Conference Proceeding

The earlier version of this paper appeared in [2]. We refine the definitions of security notions and add the full security proofs.

2 Preliminaries

For any algorithm $\mathcal{A}$, we denote by $y \leftarrow \mathcal{A}(x)$ that $\mathcal{A}$ outputs y on input x. When $\mathcal{A}$ is probabilistic, $\mathcal{A}(x)$ stands for the random variable of $\mathcal{A}$'s output on input x, where the probability is taken over the internal coin flips of $\mathcal{A}$. In particular, we explicitly express that $\mathcal{A}$ outputs y on input x with the random coin r by $y \leftarrow \mathcal{A}(x; r)$. We abbreviate the word "probabilistic polynomial-time" as PPT.

For a finite set X, $x \overset{\$}{\leftarrow} X$ means that x is chosen from X uniformly at random. $|x|$ and $|X|$ denotes the length of the element x and the size of the set X, respectively. $\langle\cdot\rangle$ is some encoding function which takes strings as input.

A positive function ϵ in λ is said to be *negligible* if for any positive polynomial p, there exists a natural number λ_0 such that for any $\lambda \geq \lambda_0$, $\epsilon(\lambda) < 1/p(\lambda)$.

3 Dynamic Group Signature with Message Dependent Opening

We introduce the notion of the dynamic group signature with message dependent opening (DGS-MDO). We first define the syntax of DGS-MDO and consider the security requirements with several oracles which express the ability of adversaries.

3.1 Syntax

A DGS-MDO scheme $\mathcal{DGS}\text{-}\mathcal{MDO}$ consists of a tuple $(\mathsf{GKg}, \mathsf{UKg}, \mathsf{Join}, \mathsf{Iss}, \mathsf{GSig}, \mathsf{Td}, \mathsf{GVf}, \mathsf{Open}, \mathsf{Judge})$. There are five parties on DGS-MDO: the trust, the issuer, the opener, the admitter and the user. The syntax of DGS-MDO is defined based on the dynamic group signature [5] and the (static) group signature with message dependent opening [11]. The formal description is as follows.

$\mathsf{GKg}(1^\lambda) \rightarrow (\mathsf{gpk}, \mathsf{ik}, \mathsf{ok}, \mathsf{ak})$:

GKg is the *group-key generation* algorithm performed by the trusted party. On input security parameter 1^λ, GKg generates the *group public key* gpk, the *issuer key* ik, the *opener key* ok and the *admitter key* ak, respectively. ik, ok, ak are sent to the corresponding party via a secure channel, and gpk is made public.

$\mathsf{UKg}(1^\lambda, \mathsf{gpk}) \rightarrow (\mathsf{upk}_i, \mathsf{usk}_i)$:

UKg is the *user key generation* algorithm performed by each user. On input security parameter 1^λ and gpk, the user i runs UKg and obtains the *personal public and secret key pair* $(\mathsf{upk}_i, \mathsf{usk}_i)$. The list $\mathbf{upk} = \{\mathsf{upk}_i\}$ of public keys of all users is made public.

$\mathsf{Join}(\mathsf{gpk}, \mathsf{upk}_i, \mathsf{usk}_i) \rightarrow \mathsf{gsk}_i$ and $\mathsf{Iss}(\mathsf{gpk}, \mathsf{ik}, \mathbf{reg}) \rightarrow \mathbf{reg}$:

Join and Iss are the *two-party interactive group-joining* protocol between the user i and the issuer. Join is performed by the user and Iss is performed by the issuer, respectively. After the interaction,

Join outputs gsk_i which is the group signing key for the user i if the protocol succeeds to the end. On the other hand, Iss outputs the *registration table* $\mathbf{reg}$ which records the registration information of legitimate users. Namely, $\mathbf{reg}$ is updated by Iss with the i-th entry $\mathbf{reg}[i]$ of $\mathbf{reg}$ if the issuer accepts the user i.

$\mathsf{GSig}(\mathsf{gpk}, \mathsf{gsk}_i, m) \to \sigma$:
GSig is the *group signing* algorithm performed by each user. On input gpk, gsk_i and the *message* m, GSig generates the *group signature* σ.

$\mathsf{Td}(\mathsf{gpk}, \mathsf{ak}, m) \to \mathsf{t}_m$:
Td is the *message-specific token generation* algorithm performed by the admitter. On input $(\mathsf{gpk}, \mathsf{ak}, m)$, Td generates the *token* t_m for the message m.

$\mathsf{GVf}(\mathsf{gpk}, m, \sigma) \to 1/0$:
GVf is the *deterministic group signature verification* algorithm. We say that the signature σ is *valid* for m under gpk when GVf outputs 1.

$\mathsf{Open}(\mathsf{gpk}, \mathsf{ok}, \mathbf{reg}, m, \sigma, \mathsf{t}_m) \to (i, \pi)$:
Open is the *deterministic opening* algorithm performed by the opener. The output i identifies the signer of the group signature σ and π is the proof for the fact, respectively. Note that when $i = 0$, it means that there is no legitimate group member which produces σ.

$\mathsf{Judge}(\mathsf{gpk}, i, \mathsf{upk}_i, m, \sigma, \pi) \to 1/0$:
Judge is the *deterministic judgement* algorithm. It checks whether or not π is a valid proof that the user i generates σ for m.

3.2 Security Definitions

We first introduce oracles which is used in the security definitions. These oracles give adversaries various capabilities and functionalities. We can represent various attack scenarios by combining them. The definitions of oracles and securities are based on [5] and [21]. In the descriptions of oracles below, $\mathbf{HU}$ is the set of honest users, $\mathbf{CU}$ is the set of corrupted users, $\mathbf{RU}$ is the set of users which are revealed with their signing keys, $\mathbf{QL_{gs}}$ is the set of signing queries, $\mathbf{TL}$ is the set of messages whose corresponding tokens are generated, $\mathbf{CL}$ is the set of challenge signatures, respectively.

$\mathsf{AddU}(i)$: AddU adds the user i as an honest user. AddU adds i into the set $\mathbf{HU}$, generates $(\mathsf{upk}_i, \mathsf{usk}_i)$ and then gsk_i, updates $\mathbf{reg}[i]$, respectively, via operating Join and Iss internally. Finally, AddU returns upk_i.

$\mathsf{CrptU}(i, \mathsf{upk})$: The adversary corrupts the user i by calling this oracle. CrptU updates upk_i of the user i as upk, and adds i into the set $\mathbf{CU}$.

$\mathsf{SndToI}(i)$: The adversary calls this oracle to run the group-joining protocol on behalf of the corrupted user i with the honest issuer. When the oracle accepts the protocol, it updates $\mathbf{reg}[i]$ as Iss does.

$\mathsf{SndToU}(i)$: The adversary calls this oracle to run the group-joining protocol on behalf of the corrupted issuer with the honest user i. When the oracle accepts the protocol, it updates the user i's signing key gsk_i.

$\mathsf{USK}(i)$: USK returns the signing key gsk_i and secret key usk_i of the user i, and adds i into the set $\mathbf{RU}$.

$\mathsf{RReg}(i)$: RReg returns the registration information $\mathbf{reg}[i]$ of the user i.

$\mathsf{WReg}(i, \rho)$: WReg writes/changes the registration information $\mathbf{reg}[i]$ of the user i into ρ.

$\mathsf{GSig}(i, m)$: GSig returns the signature σ on the message m with respect to the signing key gsk_i, and adds (i, m) into the set $\mathbf{QL_{gs}}$.

$\mathsf{Ch}_b(i_0, i_1 .m)$: Ch_b returns the challenge signature σ on the message m with respect to the signing key of the user i_b for $b \in \{0, 1\}$. Ch_b adds (m, σ) into the set $\mathbf{CL}$.

$\mathsf{Td}(m)$: Td returns the message-specific token t_m for the message m. If m is not in $\mathbf{TL}$, Td adds m into in $\mathbf{TL}$.

$\mathsf{Open}(m,\sigma)$: Open returns the identity i of the signer of σ with the proof π when the input (m,σ) is not in $\mathbf{CL}$.

We now define the security notions of DGS-MDO.

Correctness: The correctness states that honestly generated signatures are always valid, opened to the correct signers, and the proofs produced by Open pass Judge. For a DGS-MDO scheme $\mathcal{DGS}\text{-}\mathcal{MDO}$ and any adversary $\mathcal{A}$, let us consider the following experiment.

$\mathsf{Exp}^{\mathsf{corr}}_{\mathcal{DGS}\text{-}\mathcal{MDO},\mathcal{A}}(\lambda)$:

1. $(\mathsf{gpk},\mathsf{ik},\mathsf{ok},\mathsf{ak}) \leftarrow \mathsf{GKg}(1^\lambda)$; $\mathbf{HU}=\emptyset$;
2. $(i,m) \leftarrow \mathcal{A}^{\mathsf{AddU},\mathsf{RReg}}(\mathsf{gpk})$;
3. If $i \notin \mathbf{HU}$ then return 0; If $\mathsf{gsk}_i = \epsilon$ then return 0;
4. $\sigma \leftarrow \mathsf{GSig}(\mathsf{gpk},\mathsf{gsk}_i,m)$;
   If $\mathsf{GVf}(\mathsf{gpk},m,\sigma)=0$ then return 1;
5. $\mathsf{t}_m \leftarrow \mathsf{Td}(\mathsf{gpk},\mathsf{ak},m)$; $(j,\pi) \leftarrow \mathsf{Open}(\mathsf{gpk},\mathsf{ok},\mathbf{reg},m,\sigma,\mathsf{t}_m)$;
   If $i \neq j$ then return 1;
6. If $\mathsf{Judge}(\mathsf{gpk},i,\mathsf{upk}_i,m,\sigma,\pi)=0$ then return 1, else return 0

The advantage of $\mathcal{A}$ for the experiment is defined by

$$\mathsf{Adv}^{\mathsf{corr}}_{\mathcal{DGS}\text{-}\mathcal{MDO},\mathcal{A}}(\lambda) = \Pr[\mathsf{Exp}^{\mathsf{corr}}_{\mathcal{DGS}\text{-}\mathcal{MDO},\mathcal{A}}(\lambda)=1].$$

We say that a DGS-MDO scheme $\mathcal{DGS}\text{-}\mathcal{MDO}$ is *correct* if $\mathsf{Adv}^{\mathsf{corr}}_{\mathcal{DGS}\text{-}\mathcal{MDO},\mathcal{A}}(\lambda)=0$ for any unbounded adversary $\mathcal{A}$ and $\lambda \in \mathbb{N}$.

On DGS-MDO scheme, we consider two types of anonymity, the *opener anonymity* and the *admitter anonymity*. This is because identifying the signer of a group signature requires two steps: generating a message-specific token by the admitter, and opening the identity by the opener. These two notions imply that the anonymity cannot be broken by either the opener or the admitter alone.

Opener Anonymity: The opener anonymity states that a PPT adversary $\mathcal{A}$ cannot distinguish signatures generated by two distinct legitimate users even though $\mathcal{A}$ knows the opener key ok and the issuer key ik, corrupts any user and accesses the Td oracle. For a DGS-MDO scheme $\mathcal{DGS}\text{-}\mathcal{MDO}$ and any adversary $\mathcal{A}$, let us consider the following experiment.

$\mathsf{Exp}^{\mathsf{op\text{-}anon},b}_{\mathcal{DGS}\text{-}\mathcal{MDO},\mathcal{A}}(\lambda)$:

1. $(\mathsf{gpk},\mathsf{ik},\mathsf{ok},\mathsf{ak}) \leftarrow \mathsf{GKg}(1^\lambda)$; $\mathbf{HU},\mathbf{CU},\mathbf{RU},\mathbf{TL},\mathbf{CL}=\emptyset$;
2. $b' \leftarrow \mathcal{A}^{\mathsf{CrptU},\mathsf{SndToU},\mathsf{USK},\mathsf{WReg},\mathsf{Td},\mathsf{Ch}_b}(\mathsf{gpk},\mathsf{ik},\mathsf{ok})$;
3. return b'

The advantage of $\mathcal{A}$ for the experiment is defined by

$$\mathsf{Adv}^{\mathsf{op\text{-}anon}}_{\mathcal{DGS}\text{-}\mathcal{MDO},\mathcal{A}}(\lambda) = |\Pr[\mathsf{Exp}^{\mathsf{op\text{-}anon},0}_{\mathcal{DGS}\text{-}\mathcal{MDO},\mathcal{A}}(\lambda)=1] - \Pr[\mathsf{Exp}^{\mathsf{op\text{-}anon},1}_{\mathcal{DGS}\text{-}\mathcal{MDO},\mathcal{A}}(\lambda)=1]|.$$

We say that a DGS-MDO scheme $\mathcal{DGS}\text{-}\mathcal{MDO}$ has the *opener anonymity* if $\mathsf{Adv}^{\mathsf{op\text{-}anon}}_{\mathcal{DGS}\text{-}\mathcal{MDO},\mathcal{A}}(\lambda)$ is negligible in λ for any PPT adversary $\mathcal{A}$. When $\mathcal{A}$ calls Td oracle at most k times, we say that $\mathcal{DGS}\text{-}\mathcal{MDO}$ has the opener anonymity *with k-bounded tokens*.

Admitter Anonymity: The admitter anonymity states that a PPT adversary $\mathcal{A}$ cannot distinguish signatures generated by two distinct legitimate users even though $\mathcal{A}$ knows the admitter key ak and the issuer key ik, corrupts any user and accesses the Open oracle. For a DGS-MDO scheme $\mathcal{DGS}\text{-}\mathcal{MDO}$ and any adversary $\mathcal{A}$, let us consider the following experiment.

$\mathsf{Exp}^{\mathsf{ad\text{-}anon},b}_{\mathcal{DGS}\text{-}\mathcal{MDO},\mathcal{A}}(\lambda)$:

1. $(\mathsf{gpk}, \mathsf{ik}, \mathsf{ok}, \mathsf{ak}) \leftarrow \mathsf{GKg}(1^\lambda)$; $\mathbf{HU}, \mathbf{CU}, \mathbf{RU}, \mathbf{CL} = \emptyset$;
2. $b' \leftarrow \mathcal{A}^{\mathsf{CrptU}, \mathsf{SndToU}, \mathsf{USK}, \mathsf{WReg}, \mathsf{Open}, \mathsf{Ch}_b}(\mathsf{gpk}, \mathsf{ik}, \mathsf{ak})$;
3. return b'

The advantage of $\mathcal{A}$ for the experiment is defined by

$$\mathsf{Adv}^{\mathsf{ad\text{-}anon}}_{\mathcal{DGS\text{-}MDO},\mathcal{A}}(\lambda) = |\Pr[\mathsf{Exp}^{\mathsf{ad\text{-}anon},0}_{\mathcal{DGS\text{-}MDO},\mathcal{A}}(\lambda) = 1] - \Pr[\mathsf{Exp}^{\mathsf{ad\text{-}anon},1}_{\mathcal{DGS\text{-}MDO},\mathcal{A}}(\lambda) = 1]|.$$

We say that a DGS-MDO scheme $\mathcal{DGS\text{-}MDO}$ has the *admitter anonymity* if $\mathsf{Adv}^{\mathsf{ad\text{-}anon}}_{\mathcal{DGS\text{-}MDO},\mathcal{A}}(\lambda)$ is negligible in λ for any PPT adversary $\mathcal{A}$.

We note that AddU oracle does not appear in the two definitions of anonymity. This is because $\mathcal{A}$ is given the issuer key ik and then $\mathcal{A}$ can add a user to the group arbitrarily by itself.

Traceability: The traceability states that a PPT adversary $\mathcal{A}$ cannot generate a valid signature which is traced to an illicit user even though $\mathcal{A}$ knows the opener key ok and the admitter key ak and corrupts any user. For a DGS-MDO scheme $\mathcal{DGS\text{-}MDO}$ and any adversary $\mathcal{A}$, let us consider the following experiment.

$\mathsf{Exp}^{\mathsf{trace}}_{\mathcal{DGS\text{-}MDO},\mathcal{A}}(\lambda)$:

1. $(\mathsf{gpk}, \mathsf{ik}, \mathsf{ok}, \mathsf{ak}) \leftarrow \mathsf{GKg}(1^\lambda)$; $\mathbf{HU}, \mathbf{CU}, \mathbf{RU} = \emptyset$;
2. $(m, \sigma) \leftarrow \mathcal{A}^{\mathsf{AddU}, \mathsf{CrptU}, \mathsf{SndToI}, \mathsf{USK}, \mathsf{RReg}}(\mathsf{gpk}, \mathsf{ok}, \mathsf{ak})$;
3. If $\mathsf{GVf}(\mathsf{gpk}, m, \sigma) = 0$ then return 0;
4. $\mathsf{t}_m \leftarrow \mathsf{Td}(\mathsf{gpk}, \mathsf{ak}, m)$; $(i, \pi) \leftarrow \mathsf{Open}(\mathsf{gpk}, \mathsf{ok}, \mathbf{reg}, m, \sigma, \mathsf{t}_m)$;
5. If $i = 0$ or $\mathsf{Judge}(\mathsf{gpk}, i, \mathsf{upk}_i, m, \sigma, \pi) = 0$ then return 1, else return 0

The advantage of $\mathcal{A}$ for the experiment is defined by

$$\mathsf{Adv}^{\mathsf{trace}}_{\mathcal{DGS\text{-}MDO},\mathcal{A}}(\lambda) = \Pr[\mathsf{Exp}^{\mathsf{trace}}_{\mathcal{DGS\text{-}MDO},\mathcal{A}}(\lambda) = 1].$$

We say that a DGS-MDO scheme $\mathcal{DGS\text{-}MDO}$ has the *traceability* if $\mathsf{Adv}^{\mathsf{trace}}_{\mathcal{DGS\text{-}MDO},\mathcal{A}}(\lambda)$ is negligible in λ for any PPT adversary $\mathcal{A}$.

Non-Frameability: The non-frameability states that a PPT adversary $\mathcal{A}$ cannot generate a valid signature which is traced to a legitimate user even though $\mathcal{A}$ knows the opener key ok, the admitter key ak and the issuer key ik and corrupts any user. For a DGS-MDO scheme $\mathcal{DGS\text{-}MDO}$ and any adversary $\mathcal{A}$, let us consider the following experiment.

$\mathsf{Exp}^{\mathsf{nf}}_{\mathcal{DGS\text{-}MDO},\mathcal{A}}(\lambda)$:

1. $(\mathsf{gpk}, \mathsf{ik}, \mathsf{ok}, \mathsf{ak}) \leftarrow \mathsf{GKg}(1^\lambda)$; $\mathbf{HU}, \mathbf{CU}, \mathbf{RU}, \mathbf{QL_{gs}} = \emptyset$;
2. $(m, \sigma, i, \pi) \leftarrow \mathcal{A}^{\mathsf{CrptU}, \mathsf{SndToU}, \mathsf{USK}, \mathsf{WReg}, \mathsf{GSig}}(\mathsf{gpk}, \mathsf{ok}, \mathsf{ak}, \mathsf{ik})$;
3. If $\mathsf{GVf}(\mathsf{gpk}, m, \sigma) = 0$ then return 0;
4. If all following conditions are satisfied then return 1;
   (a) $i \in \mathbf{HU}$ and $i \notin \mathbf{RU}$;
   (b) $\mathsf{Judge}(\mathsf{gpk}, i, \mathsf{upk}_i, m, \sigma, \pi) = 1$;
   (c) $(i, m) \notin \mathbf{QL_{gs}}$;
5. Else return 0

The advantage of $\mathcal{A}$ for the experiment is defined by

$$\mathsf{Adv}^{\mathsf{nf}}_{\mathcal{DGS\text{-}MDO},\mathcal{A}}(\lambda) = \Pr[\mathsf{Exp}^{\mathsf{nf}}_{\mathcal{DGS\text{-}MDO},\mathcal{A}}(\lambda) = 1].$$

We say that a DGS-MDO scheme $\mathcal{DGS\text{-}MDO}$ has the *non-frameability* if $\mathsf{Adv}^{\mathsf{nf}}_{\mathcal{DGS\text{-}MDO},\mathcal{A}}(\lambda)$ is negligible in λ for any PPT adversary $\mathcal{A}$.

4 Generic Construction of DGS-MDO

We propose a generic construction of DGS-MDO in this section. Before describing our construction, we introduce several primitives which are required in the construction.

4.1 Building Blocks

Digital Signature: The digital signature scheme $\mathcal{DS}$ consists of three algorithms $(\mathsf{SKg}, \mathsf{Sig}, \mathsf{Vf})$ defined as follows.

$\mathsf{SKg}(1^\lambda) \to (\mathsf{pk}_s, \mathsf{sk}_s)$:
SKg is the key generation algorithm that takes a security parameter 1^λ as input and outputs a pair of a public key pk_s and a secret key sk_s.

$\mathsf{Sig}(\mathsf{sk}_s, m) \to \sigma$:
Sig is the signing algorithm that takes a secret key sk_s and a message m as input and outputs a signature σ.

$\mathsf{Vf}(\mathsf{pk}_s, m, \sigma) \to 1/0$:
Vf is the deterministic verification algorithm that takes a public key pk_s, a message m and a signature σ as input and outputs 1 or 0.

As the security of digital signatures, we consider the ordinary existential unforgeability against the chosen message attack (EUF-CMA) [12] defined by the following experiment. Let Sig be the signing oracle provided to an adversary. Sig returns a signature $\sigma \leftarrow \mathsf{Sig}(\mathsf{sk}_s, m)$ for a queried message m. Let $\mathbf{QL_s}$ be the set of signing queries.

$\mathsf{Exp}^{\mathsf{euf\text{-}cma}}_{\mathcal{DS},\mathcal{A}}(\lambda)$:

1. $(\mathsf{pk}_s, \mathsf{sk}_s) \leftarrow \mathsf{SKg}(1^\lambda)$; $\mathbf{QL_s} = \emptyset$;
2. $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sig}}(\mathsf{pk}_s)$;
3. If all following conditions are satisfied then return 1;
   (a) $\mathsf{Vf}(\mathsf{pk}_s, m^*, \sigma^*) = 1$;
   (b) $m^* \notin \mathbf{QL_s}$;
4. Else return 0

The advantage of $\mathcal{A}$ for the experiment is defined by

$$\mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathcal{DS},\mathcal{A}}(\lambda) = \Pr[\mathsf{Exp}^{\mathsf{euf\text{-}cma}}_{\mathcal{DS},\mathcal{A}}(\lambda) = 1].$$

We say that a digital signature scheme $\mathcal{DS}$ is EUF-CMA if $\mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathcal{DS},\mathcal{A}}(\lambda)$ is negligible in λ for any PPT adversary $\mathcal{A}$.

Key Encapsulation Mechanism: The key encapsulation mechanism (KEM) $\mathcal{KEM}$ consists of three algorithms $(\mathsf{EKg}, \mathsf{Enc}, \mathsf{Dec})$ defined as follows.

$\mathsf{EKg}(1^\lambda) \to (\mathsf{pk}_e, \mathsf{sk}_e)$:
EKg is the key generation algorithm that takes a security parameter 1^λ as input and outputs a pair of a public key pk_e and a secret key sk_e.

$\mathsf{Enc}(\mathsf{pk}_e) \to (C, K)$:
Enc is the encapsulation algorithm that takes a public key pk_e as input and outputs a ciphertext C and a session key $K \in \mathcal{K}_{\mathsf{KEM}}$, where $\mathcal{K}_{\mathsf{KEM}}$ is the space of session keys associated with the scheme.

$\mathsf{Dec}(\mathsf{sk}_e, C) \to K/\perp$:
Dec is the deterministic decapsulation algorithm that takes a secret key sk_e and a ciphertext C as input and outputs a session key K or a special symbol $\perp$ which indicates that the ciphertext is invalid.

As the security of KEM, we consider the indistinguishability against the chosen ciphertext attack (IND-CCA) [10] defined by the following experiment. Let Dec be the decapsulation oracle provided to an adversary. Dec returns a session key $K \leftarrow \mathsf{Dec}(\mathsf{sk}_e, C)$ for a queried ciphertext C. Let $\mathbf{QL_c}$ be the set of queried ciphertexts.

$\mathsf{Exp}^{\mathsf{ind\text{-}cca},b}_{\mathcal{KEM},\mathcal{A}}(\lambda)$:

1. $(\mathsf{pk}_e, \mathsf{sk}_e) \leftarrow \mathsf{EKg}(1^\lambda)$; $\mathbf{QL_c} = \emptyset$;
2. $b' \leftarrow \mathcal{A}^{\mathsf{Dec},\mathsf{Ch}_{\mathcal{KEM},b}}(\mathsf{pk}_e)$;
3. If $C^* \notin \mathbf{QL_c}$ then return b' else return $\perp$

$\mathsf{Ch}_{\mathcal{KEM},b}(\mathsf{pk}_e)$:

1. $(C^*, K_0) \leftarrow \mathsf{Enc}(\mathsf{pk}_e)$;
2. $K_1 \xleftarrow{\$} \mathcal{K}_{\mathsf{KEM}}$;
3. return (C^*, K_b)

The advantage of $\mathcal{A}$ for the experiment is defined by

$$\mathsf{Adv}^{\mathsf{ind\text{-}cca}}_{\mathcal{KEM},\mathcal{A}}(\lambda) = |\Pr[\mathsf{Exp}^{\mathsf{ind\text{-}cca},0}_{\mathcal{KEM},\mathcal{A}}(\lambda) = 1] - \Pr[\mathsf{Exp}^{\mathsf{ind\text{-}cca},1}_{\mathcal{KEM},\mathcal{A}}(\lambda) = 1]|.$$

We say that a KEM $\mathcal{KEM}$ is IND-CCA if $\mathsf{Adv}^{\mathsf{ind\text{-}cca}}_{\mathcal{KEM},\mathcal{A}}(\lambda)$ is negligible in λ for any PPT adversary $\mathcal{A}$.

ID-based KEM: The identity-based key encapsulation mechanism (ID-based KEM) [6] $\mathcal{ID\text{-}KEM}$ consists of four algorithms $(\mathsf{ISetup}, \mathsf{IExt}, \mathsf{IEnc}, \mathsf{IDec})$ defined as follows.

$\mathsf{ISetup}(1^\lambda) \rightarrow (\mathsf{pp}, \mathsf{msk})$:
ISetup is the parameter setup algorithm that takes a security parameter 1^λ as input and outputs a public parameter pp and a master secret key msk.

$\mathsf{IExt}(\mathsf{msk}, ID) \rightarrow (\mathsf{sk}_{ID})$:
IExt is the user key generation algorithm that takes a master secret key msk and a user's identity ID as input and outputs a user's secret key sk_{ID}.

$\mathsf{IEnc}(\mathsf{pp}, ID) \rightarrow (C, K)$:
IEnc is the encapsulation algorithm that takes a public parameter pp and an identity ID as input and outputs a ciphertext C and a session key $K \in \mathcal{K}_{\mathsf{IDKEM}}$, where $\mathcal{K}_{\mathsf{IDKEM}}$ is the space of session keys associated with the scheme.

$\mathsf{IDec}(\mathsf{sk}_{ID}, C, ID) \rightarrow K/\perp$:
IDec is the deterministic decapsulation algorithm that takes a user's secret key sk_{ID}, a ciphertext C and an identity ID as input and outputs a session key K or a special symbol $\perp$ which indicates that the ciphertext is invalid.

As the security of ID-based KEM, we consider the k-resilient security [14] defined by the following experiment. Let IExt be the key extraction oracle provided to an adversary. IExt returns a user's secret key $\mathsf{sk}_{ID} \leftarrow \mathsf{IExt}(\mathsf{msk}, ID)$ for a queried identity ID. Let $\mathbf{QL_{id}}$ be the set of queried identities.

$\mathsf{Exp}^{\mathsf{k\text{-}resi},b}_{\mathcal{ID\text{-}KEM},\mathcal{A}}(\lambda)$:

1. $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{ISetup}(1^\lambda)$; $\mathbf{QL_{id}} = \emptyset$;
2. $b' \leftarrow \mathcal{A}^{\mathsf{IExt},\mathsf{Ch}_{\mathcal{ID\text{-}KEM},b}}(\mathsf{pp})$;
3. If $ID^* \notin \mathbf{QL_{id}}$ and $|\mathbf{QL_{id}}| \leq k$ then return b' else return $\perp$

$\mathsf{Ch}_{\mathcal{ID\text{-}KEM},b}(ID^*)$:

1. $(C^*, K_0) \leftarrow \mathsf{IEnc}(\mathsf{pp}, ID^*)$;

2. $K_1 \overset{\$}{\leftarrow} \mathcal{K}_{\mathsf{IDKEM}}$;

3. return (C^*, K_b)

The advantage of $\mathcal{A}$ for the experiment is defined by

$$\mathsf{Adv}^{\mathsf{k\text{-}resi}}_{\mathcal{ID}\text{-}\mathcal{KEM},\mathcal{A}}(\lambda) = |\Pr[\mathsf{Exp}^{\mathsf{k\text{-}resi},0}_{\mathcal{ID}\text{-}\mathcal{KEM},\mathcal{A}}(\lambda) = 1] - \Pr[\mathsf{Exp}^{\mathsf{k\text{-}resi},1}_{\mathcal{ID}\text{-}\mathcal{KEM},\mathcal{A}}(\lambda) = 1]|.$$

We say that an ID-based KEM $\mathcal{ID}\text{-}\mathcal{KEM}$ is k-resilient if $\mathsf{Adv}^{\mathsf{k\text{-}resi}}_{\mathcal{ID}\text{-}\mathcal{KEM},\mathcal{A}}(\lambda)$ is negligible in λ for any PPT adversary $\mathcal{A}$. When k is a polynomial in λ, we say that $\mathcal{ID}\text{-}\mathcal{KEM}$ is IND-ID-CPA.

Non-Interactive Zero Knowledge Proof: Let $R \subseteq \{0,1\}^* \times \{0,1\}^*$ be a polynomial-time verifiable relation. Namely we can check whether or not $(x, w) \in R$ in polynomial time. Note that we estimate the time efficiency in the length of the first component x. We consider a non-interactive zero-knowledge proof in the common reference string model.

The non-interactive proof system Π consists of the three algorithm $(\mathsf{K}_{\mathsf{crs}}, \mathsf{P}, \mathsf{V})$ defined as follows.

$\mathsf{K}_{\mathsf{crs}}(1^\lambda) \to \mathsf{crs}$:
$\mathsf{K}_{\mathsf{crs}}$ is the common reference string (CRS) generation algorithm that takes a security parameter 1^λ as input and outputs a CRS crs.

$\mathsf{P}(\mathsf{crs}, (x, w)) \to \tau$:
P is the prover algorithm that takes a CRS crs, a pair (x, w) of a statement and a witness such that $(x, w) \in R$ as input and outputs a proof τ.

$\mathsf{V}(\mathsf{crs}, x, \tau) \to 1/0$:
V is the verifier algorithm that takes a CRS crs, a statement x and a proof τ as input and outputs 1 or 0.

We say that $\Pi = (\mathsf{K}_{\mathsf{crs}}, \mathsf{P}, \mathsf{V})$ is a non-interactive zero knowledge proof (NIZK) for the relation R if Π satisfies the following three properties.

Completeness: Let p be a polynomial. For all $\lambda \in \mathbb{N}$ and $(x, w) \in R$ such that $|x| \leq p(\lambda)$,

$$\Pr[\mathsf{crs} \leftarrow \mathsf{K}_{\mathsf{crs}}(1^\lambda); \tau \leftarrow \mathsf{P}(\mathsf{crs}, (x, w)); \mathsf{V}(\mathsf{crs}, x, \tau) \to 1] = 1.$$

Soundness: Let $\bar{L}_R$ be the set of all x that has no corresponding witness. For all $\lambda \in \mathbb{N}$, $x \in \bar{L}_R$ and any polynomial-time algorithm $\bar{\mathsf{P}}$, the probability

$$\Pr[\mathsf{crs} \leftarrow \mathsf{K}_{\mathsf{crs}}(1^\lambda); \tau \leftarrow \bar{\mathsf{P}}(\mathsf{crs}, x); \mathsf{V}(\mathsf{crs}, x, \tau) \to 1]$$

is negligible in λ.

Computational Zero Knowledge: Let $\mathsf{Sim} = (S_1, S_2)$ be a simulator algorithm. S_1 outputs a *simulated* CRS for the input security parameter, and S_2 outputs a *simulated* proof for the input statement and simulated CRS. Then we consider the following experiments.

$\mathsf{Exp}^{\mathsf{zk}}_{\Pi,\mathsf{Sim},\mathcal{D}}(\lambda)$:

1. $b \overset{\$}{\leftarrow} \{0, 1\}$;

2. $\mathsf{crs}_0 \leftarrow \mathsf{K}_{\mathsf{crs}}(1^\lambda)$; $(\mathsf{crs}_1, \mathsf{st}) \leftarrow S_1(1^\lambda)$; $\mathsf{crs} = \mathsf{crs}_b$;

3. $b' \leftarrow \mathcal{D}^{\mathsf{PO}_b}(\mathsf{crs})$;

4. If $b = b'$ then return 1 else return 0

$\mathsf{PO}_0(x, w)$:

1. $\tau \leftarrow \mathsf{P}(\mathsf{crs}, x, w)$;

2. Return τ

$\mathsf{PO}_1(x, w)$:

1. $\tau \leftarrow S_2(\mathsf{crs}, \mathsf{st}, x)$;
2. Return τ

The advantage of $\mathcal{D}$ for the experiment is defined by

$$\mathsf{Adv}^{\mathsf{zk}}_{\Pi,\mathsf{Sim},\mathcal{D}}(\lambda) = |\Pr[\mathsf{Exp}^{\mathsf{zk}}_{\Pi,\mathsf{Sim},\mathcal{D}}(\lambda) = 1] - 1/2|.$$

We say that Π is computational zero knowledge if there exists a PPT simulator Sim such that $\mathsf{Adv}^{\mathsf{zk}}_{\Pi,\mathsf{Sim},\mathcal{D}}(\lambda)$ is negligible in λ for any PPT algorithm $\mathcal{D}$.

We consider additional property for a NIZK Π, called the *simulation-soundness* [18] defined by the following experiment.

$\mathsf{Exp}^{\mathsf{ss}}_{\Pi,\mathsf{Sim},\mathcal{A}}(\lambda)$:

1. $(\mathsf{crs}, \mathsf{st}) \leftarrow S_1(1^\lambda)$;
2. $(x, \tau) \leftarrow \mathcal{A}^{S_2(\mathsf{st})}(\mathsf{crs})$;
3. If all following conditions are satisfied then return 1;
   (a) $x \in \bar{L}_R$;
   (b) τ is not returned from the oracle when $\mathcal{A}$ queries x;
   (c) $\mathcal{A}$ calls S_2 oracle exactly one time;
   (d) $\mathsf{V}(\mathsf{crs}, x, \tau) = 1$;
4. Else return 0

The advantage of $\mathcal{A}$ for the experiment is defined by

$$\mathsf{Adv}^{\mathsf{ss}}_{\Pi,\mathsf{Sim},\mathcal{A}}(\lambda) = \Pr[\mathsf{Exp}^{\mathsf{ss}}_{\Pi,\mathsf{Sim},\mathcal{A}}(\lambda) = 1].$$

We say that Π is simulation-sound if $\mathsf{Adv}^{\mathsf{ss}}_{\Pi,\mathsf{Sim},\mathcal{A}}(\lambda)$ is negligible in λ for any PPT adversary $\mathcal{A}$.

4.2 Construction

We describe our generic construction. Let $\mathcal{DS} = (\mathsf{SKg}, \mathsf{Sig}, \mathsf{Vf})$ be a digital signature scheme, $\mathcal{KEM} = (\mathsf{EKg}, \mathsf{Enc}, \mathsf{Dec})$ be a KEM, $\mathcal{ID\text{-}KEM} = (\mathsf{ISetup}, \mathsf{IExt}, \mathsf{IEnc}, \mathsf{IDec})$ be an ID-based KEM, respectively. Assume that $\mathcal{K}_{\mathsf{KEM}}$ and $\mathcal{K}_{\mathsf{IDKEM}}$ are the same group with the operation $\odot$. We employ a polynomial-time computable and invertible encoding function $\langle\cdot\rangle$ which maps strings to elements in $\mathcal{K}_{\mathsf{KEM}} = \mathcal{K}_{\mathsf{IDKEM}}$.

Let $\mathsf{pk}_s, \mathsf{pk}_i$ be public keys of $\mathcal{DS}$, cert_i and s_i be signatures by $\mathcal{DS}$, m be a message, i be a user index, $(\mathsf{pk}_e, \mathsf{sk}_e)$ be a pair of a public key and a secret key of $\mathcal{KEM}$, $(C_{\mathsf{KEM}}, K_{\mathsf{KEM}})$ be an output of Enc, pp be a public parameter of $\mathcal{ID\text{-}KEM}$, $(C_{\mathsf{IDKEM}}, K_{\mathsf{IDKEM}})$ be an output of IEnc, t_m be an output of IExt, κ, r_e, $\mathsf{r}_{\mathsf{KEM}}$, $\mathsf{r}_{\mathsf{IDKEM}}$ be strings in $\{0,1\}^\lambda$, respectively.

Let $x = (\mathsf{pk}_s, \mathsf{pk}_e, \mathsf{pp}, m, C_{\mathsf{KEM}}, C_{\mathsf{IDKEM}}, \kappa)$ and $w = (i, \mathsf{pk}_i, \mathsf{cert}_i, s_i, \mathsf{r}_{\mathsf{KEM}}, \mathsf{r}_{\mathsf{IDKEM}})$. Then let R_1 be the relation such that $(x, w) \in R_1$ if and only if $\mathsf{Vf}(\mathsf{pk}_s, \langle i, \mathsf{pk}_i\rangle, \mathsf{cert}_i) = 1$ and $\mathsf{Vf}(\mathsf{pk}_i, m, s_i) = 1$ and $\mathsf{Enc}(\mathsf{pk}_e; \mathsf{r}_{\mathsf{KEM}}) = (C_{\mathsf{KEM}}, K_{\mathsf{KEM}})$ and $\mathsf{IEnc}(\mathsf{pp}, m; \mathsf{r}_{\mathsf{IDKEM}}) = (C_{\mathsf{IDKEM}}, K_{\mathsf{IDKEM}})$ and $\kappa = \langle i, \mathsf{pk}_i, \mathsf{cert}_i, s_i\rangle \odot K_{\mathsf{KEM}} \odot K_{\mathsf{IDKEM}}$.

Let $x = (\mathsf{pk}_e, C_{\mathsf{KEM}}, C_{\mathsf{IDKEM}}, m, \kappa, i, \mathsf{pk}_i, \mathsf{cert}_i, s_i)$ and $w = (\mathsf{sk}_e, \mathsf{r}_e, \mathsf{t}_m)$. Then let R_2 be the relation such that $(x, w) \in R_2$ if and only if $\mathsf{EKg}(1^\lambda; \mathsf{r}_e) = (\mathsf{pk}_e, \mathsf{sk}_e)$ and $\mathsf{Dec}(\mathsf{sk}_e, C_{\mathsf{KEM}}) \to K_{\mathsf{KEM}}$ and $\mathsf{IDec}(\mathsf{t}_m, C_{\mathsf{IDKEM}}, m) \to K_{\mathsf{IDKEM}}$ and $\kappa = \langle i, \mathsf{pk}_i, \mathsf{cert}_i, s_i\rangle \odot K_{\mathsf{KEM}} \odot K_{\mathsf{IDKEM}}$.

Let $\Pi_1 = (\mathsf{K}_{(\mathsf{crs},1)}, \mathsf{P}_1, \mathsf{V}_1)$ and $\Pi_2 = (\mathsf{K}_{(\mathsf{crs},2)}, \mathsf{P}_2, \mathsf{V}_2)$ be the proof systems for R_1 and R_2, respectively.

The DGS-MDO scheme $\mathcal{DGS\text{-}MDO}$ consists of $(\mathsf{GKg}, \mathsf{UKg}, \mathsf{Join}, \mathsf{Iss}, \mathsf{GSig}, \mathsf{Td}, \mathsf{GVf}, \mathsf{Open}, \mathsf{Judge})$ defined as follows.

$\mathsf{GKg}(1^\lambda) \to (\mathsf{gpk}, \mathsf{ik}, \mathsf{ok}, \mathsf{ak})$:

1. $(\mathsf{pk}_s, \mathsf{sk}_s) \leftarrow \mathsf{SKg}(1^\lambda)$;
2. $\mathsf{r}_e \xleftarrow{\$} \{0,1\}^\lambda$; $(\mathsf{pk}_e, \mathsf{sk}_e) \leftarrow \mathsf{EKg}(1^\lambda; \mathsf{r}_e)$;
3. $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{ISetup}(1^\lambda)$;
4. $\mathsf{crs}_1 \leftarrow \mathsf{K}_{(\mathsf{crs},1)}(1^\lambda)$; $\mathsf{crs}_2 \leftarrow \mathsf{K}_{(\mathsf{crs},2)}(1^\lambda)$
5. $\mathsf{gpk} = (\mathsf{pk}_s, \mathsf{pk}_e, \mathsf{pp}, \mathsf{crs}_1, \mathsf{crs}_2)$;
6. $\mathsf{ik} = \mathsf{sk}_s$; $\mathsf{ok} = (\mathsf{sk}_e, \mathsf{r}_e)$; $\mathsf{ak} = \mathsf{msk}$;
7. Return $(\mathsf{gpk}, \mathsf{ik}, \mathsf{ok}, \mathsf{ak})$

$\mathsf{UKg}(1^\lambda, \mathsf{gpk}) \rightarrow (\mathsf{upk}_i, \mathsf{usk}_i)$:

1. $(\mathsf{upk}_i, \mathsf{usk}_i) \leftarrow \mathsf{SKg}(1^\lambda)$;
2. Return $(\mathsf{upk}_i, \mathsf{usk}_i)$

$\mathsf{Join}(\mathsf{gpk}, \mathsf{upk}_i, \mathsf{usk}_i) \rightarrow \mathsf{gsk}_i$ and $\mathsf{Iss}(\mathsf{gpk}, \mathsf{ik}, \mathbf{reg}) \rightarrow \mathbf{reg}$:
Join and Iss are the three round interactive protocol between the user i and the issuer operated as follows.
Round1: $\mathsf{Join}(\mathsf{gpk}, \mathsf{upk}_i, \mathsf{usk}_i)$ by the user i:

1. $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{SKg}(1^\lambda)$;
2. $\sigma_i \leftarrow \mathsf{Sig}(\mathsf{usk}_i, \mathsf{pk}_i)$;
3. send $(\mathsf{pk}_i, \sigma_i)$ to the issuer.

Round2: $\mathsf{Iss}(\mathsf{gpk}, \mathsf{ik}, \mathbf{reg})$ by the issuer:

1. If $\mathsf{Vf}(\mathsf{upk}_i, \mathsf{pk}_i, \sigma_i) = 1$ then $\mathsf{cert}_i \leftarrow \mathsf{Sig}(\mathsf{sk}_s, \langle i, \mathsf{pk}_i \rangle)$ and $\mathbf{reg}[i] \leftarrow (\mathsf{pk}_i, \sigma_i)$;
2. Else $\mathsf{cert}_i = \epsilon$;
3. send cert_i to the user i.
4. Return $\mathbf{reg}$

Round3: $\mathsf{Join}(\mathsf{gpk}, \mathsf{upk}_i, \mathsf{usk}_i)$ by the user i:

1. $\mathsf{gsk}_i \leftarrow (i, \mathsf{pk}_i, \mathsf{sk}_i, \mathsf{cert}_i)$;
2. Return gsk_i

$\mathsf{GSig}(\mathsf{gpk}, \mathsf{gsk}_i, m) \rightarrow \sigma$:

1. $s_i \leftarrow \mathsf{Sig}(\mathsf{sk}_i, m)$;
2. $\mathsf{r}_{\mathsf{KEM}} \xleftarrow{\$} \{0,1\}^\lambda$; $(C_{\mathsf{KEM}}, K_{\mathsf{KEM}}) \leftarrow \mathsf{Enc}(\mathsf{pk}_e; \mathsf{r}_{\mathsf{KEM}})$;
3. $\mathsf{r}_{\mathsf{IDKEM}} \xleftarrow{\$} \{0,1\}^\lambda$; $(C_{\mathsf{IDKEM}}, K_{\mathsf{IDKEM}}) \leftarrow \mathsf{IEnc}(\mathsf{pp}, m; \mathsf{r}_{\mathsf{IDKEM}})$;
4. $\kappa = \langle i, \mathsf{pk}_i, \mathsf{cert}_i, s_i \rangle \odot K_{\mathsf{KEM}} \odot K_{\mathsf{IDKEM}}$;
5. $x_1 = (\mathsf{pk}_s, \mathsf{pk}_e, \mathsf{pp}, m, C_{\mathsf{KEM}}, C_{\mathsf{IDKEM}}, \kappa)$;
6. $w_1 = (i, \mathsf{pk}_i, \mathsf{cert}_i, s_i, \mathsf{r}_{\mathsf{KEM}}, \mathsf{r}_{\mathsf{IDKEM}})$;
7. $\tau_1 = \mathsf{P}_1(\mathsf{crs}_1, x_1, w_1)$;

```
8. Return $\sigma = (C_{\mathsf{KEM}}, C_{\mathsf{IDKEM}}, \kappa, \tau_1)$
```

$\mathsf{Td}(\mathsf{gpk}, \mathsf{ak}, m) \rightarrow \mathsf{t}_m$:

```
1. $\mathsf{t}_m \leftarrow \mathsf{IExt}(\mathsf{msk}, m)$;
2. Return $\mathsf{t}_m$
```

$\mathsf{GVf}(\mathsf{gpk}, m, \sigma) \rightarrow 1/0$:

```
1. $x_1 = (\mathsf{pk}_s, \mathsf{pk}_e, \mathsf{pp}, m, C_{\mathsf{KEM}}, C_{\mathsf{IDKEM}}, \kappa)$;
2. If $\mathsf{V}_1(\mathsf{crs}_1, x_1, \tau_1) = 1$ then return 1, else return 0
```

$\mathsf{Open}(\mathsf{gpk}, \mathsf{ok}, \mathbf{reg}, m, \sigma, \mathsf{t}_m) \rightarrow (i, \pi)$:

```
1. $K_{\mathsf{KEM}} \leftarrow \mathsf{Dec}(\mathsf{sk}_e, C_{\mathsf{KEM}})$;
2. $K_{\mathsf{IDKEM}} \leftarrow \mathsf{IDec}(\mathsf{t}_m, C_{\mathsf{IDKEM}}, m)$;
3. If $K_{\mathsf{KEM}} = \perp$ or $K_{\mathsf{IDKEM}} = \perp$ then return $(0, \epsilon)$;
4. $\langle i, \mathsf{pk}_i, \mathsf{cert}_i, s_i \rangle = \kappa \odot K_{\mathsf{KEM}}^{-1} \odot K_{\mathsf{IDKEM}}^{-1}$;
5. $x_1 = (\mathsf{pk}_s, \mathsf{pk}_e, \mathsf{pp}, m, C_{\mathsf{KEM}}, C_{\mathsf{IDKEM}}, \kappa)$;
6. If $\mathsf{V}_1(\mathsf{crs}_1, x_1, \tau_1) = 0$ then return $(0, \epsilon)$;
7. $x_2 = (\mathsf{pk}_e, C_{\mathsf{KEM}}, C_{\mathsf{IDKEM}}, m, \kappa, i, \mathsf{pk}_i, \mathsf{cert}_i, s_i)$;
8. $w_2 = (\mathsf{sk}_e, \mathsf{r}_e, \mathsf{t}_m)$;
9. $\tau_2 = \mathsf{P}_2(\mathsf{crs}_2, x_2, w_2)$;
10. $\pi = (\sigma_i, i, \mathsf{pk}_i, \mathsf{cert}_i, s_i, \tau_2)$;
11. Return $(i, \pi)$
```

$\mathsf{Judge}(\mathsf{gpk}, i, \mathsf{upk}_i, m, \sigma, \pi) \rightarrow 1/0$:

```
1. $x_1 = (\mathsf{pk}_s, \mathsf{pk}_e, \mathsf{pp}, m, C_{\mathsf{KEM}}, C_{\mathsf{IDKEM}}, \kappa)$;
2. If $(i, \pi) = (0, \epsilon)$ then return $1 - \mathsf{V}_1(\mathsf{crs}_1, x_1, \tau_1)$;
3. Parse $\pi = (\sigma', i', \mathsf{pk}', \mathsf{cert}', s', \tau_2)$;
4. $x_2' = (\mathsf{pk}_e, C_{\mathsf{KEM}}, C_{\mathsf{IDKEM}}, m, \kappa, i', \mathsf{pk}', \mathsf{cert}', s')$;
5. If $\mathsf{V}_2(\mathsf{crs}_2, x_2', \tau_2) = 0$ return 0;
6. If all following conditions are satisfied then return 1;
    (a) $i = i'$;
    (b) $\mathsf{pk}_i = \mathsf{pk}'$;
    (c) $\mathsf{Vf}(\mathsf{upk}_i, \mathsf{pk}', \sigma') = 1$;
7. Else return 0
```

4.3 Security

For the security of the proposed DGS-MDO scheme $\mathcal{DGS}\text{-}\mathcal{MDO}$, we have the following theorems.

Theorem 1. *$\mathcal{DGS}\text{-}\mathcal{MDO}$ is correct.*

Proof. Let $\sigma = (C_{\mathsf{KEM}}, C_{\mathsf{IDKEM}}, \kappa, \tau_1)$ be a group signature generated by a legitimate user i with honestly generated parameters.

Then $\mathsf{V}_1(\mathsf{crs}_1, x_1, \tau_1) = 1$ always follows with $x_1 = (\mathsf{pk}_s, \mathsf{pk}_e, \mathsf{pp}, m, C_{\mathsf{KEM}}, C_{\mathsf{IDKEM}}, \kappa)$ by the completeness of Π_1. Namely $\mathsf{GVf}(\mathsf{gpk}, m, \sigma) \to 1$ always holds.

Let t_m be a token for the message m. t_m is computed by $\mathsf{t}_m \leftarrow \mathsf{IExt}(\mathsf{msk}, m)$. Then the opener can decrypt C_{KEM} to K_{KEM} and C_{IDKEM} to K_{IDKEM} by using the opener key $\mathsf{ok} = (\mathsf{sk}_e, \mathsf{r}_e)$ and the token t_m. Thus the opener can retrieve the signer from $\langle i, \mathsf{pk}_i, \mathsf{cert}_i, s\rangle = \kappa \odot K_{\mathsf{KEM}}^{-1} \odot K_{\mathsf{IDKEM}}^{-1}$. $\mathsf{V}_2(\mathsf{crs}_2, x_2, \tau_2)$ also always outputs 1 by the completeness of Π_2. □

Theorem 2. *Assume that $\mathcal{ID}\text{-}\mathcal{KEM}$ is k-resilient and Π_1 is an NIZK. Then $\mathcal{DGS}\text{-}\mathcal{MDO}$ has the opener anonymity with k-bounded tokens.*

Proof. We consider the sequence of games Game between the challenger $\mathcal{C}_{\mathsf{op}}$ and the adversary $\mathcal{A}_{\mathsf{op}}$. For each $0 \leq \ell \leq 3$, let Win_ℓ be the probability of the event where $\mathcal{C}_{\mathsf{op}}$ outputs 1 in Game_ℓ.

Game_0.

Game_0 coincides with the experiment $\mathsf{Exp}^{\mathsf{op\text{-}anon},0}_{\mathcal{DGS}\text{-}\mathcal{MDO},\mathcal{A}_{\mathsf{op}}}(\lambda)$. Thus we have

$$\mathsf{Win}_0 = \Pr[\mathsf{Exp}^{\mathsf{op\text{-}anon},0}_{\mathcal{DGS}\text{-}\mathcal{MDO},\mathcal{A}_{\mathsf{op}}}(\lambda) = 1].$$

Game_1.

Game_1 coincides with Game_0 except that the proof τ_1 generated in GSig is computed by the simulator $\mathsf{Sim}_1 = (S_{1,1}, S_{1,2})$ for the proof system Π_1. Since Π_1 is an NIZK, the difference between Game_0 and Game_1 is bounded by some negligible function $\mathsf{negl}_{\mathsf{op},01}$.

$$|\,\mathsf{Win}_1 - \mathsf{Win}_0\,| \leq \mathsf{negl}_{\mathsf{op},01}(\lambda).$$

Game_2.

Game_2 coincides with Game_1 except that the challenge oracle Ch_0 is replaced with Ch_1. For the difference between Win_1 and Win_2, we have the following lemma.

Lemma 1.

$$|\,\mathsf{Win}_2 - \mathsf{Win}_1\,| \leq \mathsf{negl}_{op,12}(\lambda),$$

for some negligible function $\mathsf{negl}_{op,12}$.

Proof. Let $\mathcal{A}_{(\mathsf{op},12)}$ be an adversary which participates in Game_1 or Game_2. We aim to construct a PPT algorithm $\mathcal{B}_{(\mathsf{op},12)}$ such that if $\mathcal{B}_{(\mathsf{op},12)}$ breaks the k-resilience of $\mathcal{ID}\text{-}\mathcal{KEM}$ with $\mathcal{A}_{(\mathsf{op},12)}$, the difference between Win_1 and Win_2 can be non-negligible. The description of the algorithm $\mathcal{B}_{(\mathsf{op},12)}$ is given in Figure 1.

Let pp^* be an instance given to the k-resilience adversary $\mathcal{B}_{(\mathsf{op},12)}$. Then $\mathcal{B}_{(\mathsf{op},12)}$ invokes Game_{1+b} for $b \in \{0,1\}$ with the adversary $\mathcal{A}_{(\mathsf{op},12)}$. $\mathcal{B}_{(\mathsf{op},12)}$ generates public and secret keys $(\mathsf{pk}_s, \mathsf{sk}_s)$ and $(\mathsf{pk}_e, \mathsf{sk}_e)$, a simulated CRS crs_1, and a honestly generated CRS crs_2. Then $\mathcal{B}_{(\mathsf{op},12)}$ sets $\mathsf{gpk} = (\mathsf{pk}_s, \mathsf{pk}_e, \mathsf{pp}^*, \mathsf{crs}_1, \mathsf{crs}_2)$, $\mathsf{ik} = \mathsf{sk}_s$, $\mathsf{ok} = (\mathsf{sk}, \mathsf{r}_e)$, and gives $(\mathsf{gpk}, \mathsf{ok}, \mathsf{ik})$ to $\mathcal{A}_{(\mathsf{op},12)}$ as input.

For oracle queries from $\mathcal{A}_{(\mathsf{op},12)}$, $\mathcal{B}_{(\mathsf{op},12)}$ can answer queries honestly except Ch_b oracle and Td oracle. For a challenge oracle query, $\mathcal{B}_{(\mathsf{op},12)}$ calls its challenge oracle $\mathsf{Ch}_{\mathcal{ID}\text{-}\mathcal{KEM},b}$ to obtain $(C^*_{\mathsf{IDKEM}}, K^*_{\mathsf{IDKEM}})$ and then computes a group signature. Note that $\mathcal{B}_{(\mathsf{op},12)}$ does not know the master secret key corresponding to pp^*. Thus $\mathcal{B}_{(\mathsf{op},12)}$ uses the simulator for the proof system Π_1 to compute the proof τ_1^*.

```
$\mathcal{B}^{O_{\mathcal{ID}\text{-}\mathcal{KEM},\mathsf{IExt}}}_{(\mathsf{op},12),b}(\mathsf{pp}^*)$
$\mathbf{HU}, \mathbf{CU}, \mathbf{RU}, \mathbf{TL}, \mathbf{MTL}, \mathbf{CL} = \emptyset$
$\mathsf{count} = 0$
$(\mathsf{pk}_s, \mathsf{sk}_s) \leftarrow \mathsf{SKg}(1^\lambda)$
$\mathsf{r}_e \xleftarrow{\$} \{0,1\}^\lambda, (\mathsf{pk}_e, \mathsf{sk}_e) \leftarrow \mathsf{EKg}(1^\lambda; \mathsf{r}_e)$
$(\mathsf{crs}_1, \mathsf{st}_1) \leftarrow S_{1,1}(1^\lambda), \mathsf{crs}_2 \leftarrow \mathsf{K}_{(\mathsf{crs},2)}(1^\lambda)$
$\mathsf{gpk} = (\mathsf{pk}_s, \mathsf{pk}_e, \mathsf{pp}^*, \mathsf{crs}_1, \mathsf{crs}_2)$
$\mathsf{ik} = \mathsf{sk}_s, \mathsf{ok} = (\mathsf{sk}_e, \mathsf{r}_e)$
$b' \leftarrow \mathcal{A}^{\mathsf{CrptU},\mathsf{SndToU},\mathsf{USK},\mathsf{WReg},\mathsf{Td},\mathsf{Ch}_b}_{(\mathsf{op},12)}(\mathsf{gpk}, \mathsf{ok}, \mathsf{ik})$
$\mathbf{return}\ b'$
```

```
$\mathsf{Ch}_b(m, i_0, i_1)$
$s_{i_b} \leftarrow \mathsf{Sig}(\mathsf{sk}_{i_b}, m)$
$(C^*_{\mathsf{IDKEM}}, K^*_{\mathsf{IDKEM}}) \leftarrow \mathsf{Ch}_{\mathcal{ID}\text{-}\mathcal{KEM},b}(m)$
$\mathsf{r}_{\mathsf{KEM}} \xleftarrow{\$} \{0,1\}^\lambda$
$(C^*_{\mathsf{KEM}}, K^*_{\mathsf{KEM}}) \leftarrow \mathsf{Enc}(\mathsf{pk}_e; \mathsf{r}_{\mathsf{KEM}})$
$\kappa^* = \langle i_b, \mathsf{pk}_{i_b}, \mathsf{cert}_{i_b}, s_{i_b} \rangle \odot K^*_{\mathsf{KEM}} \odot K^*_{\mathsf{IDKEM}}$
$x^*_1 = (\mathsf{pk}_s, \mathsf{pk}_e, \mathsf{pp}^*, m, C^*_{\mathsf{KEM}}, C^*_{\mathsf{IDKEM}}, \kappa^*)$
$\tau^*_1 = S_{1,2}(\mathsf{crs}_1, \mathsf{st}_1, x^*_1)$
$\mathbf{CL} \leftarrow \mathbf{CL} \cup \{(m, \sigma^*)\}$
$\mathbf{return}\ \sigma^* = (C^*_{\mathsf{KEM}}, C^*_{\mathsf{IDKEM}}, \kappa^*, \tau^*_1)$
```

```
$\mathsf{Td}(m)$
$\mathbf{if}\ m \in \mathbf{TL}\ \mathbf{then\ return}\ \mathsf{t}_m$ s.t. $(m, \mathsf{t}_m) \in \mathbf{MTL}$
$\mathbf{if}\ \mathsf{count} \geq k\ \mathbf{then\ return}\ \perp$
$\mathsf{count} \leftarrow \mathsf{count} + 1$
$\mathsf{t}_m \leftarrow O_{\mathcal{ID}\text{-}\mathcal{KEM},\mathsf{IExt}}(m)$
$\mathbf{TL} \leftarrow \mathbf{TL} \cup \{m\}, \mathbf{MTL} \leftarrow \mathbf{MTL} \cup \{(m, \mathsf{t}_m)\}$
$\mathbf{return}\ \mathsf{t}_m$
```

Figure 1: The k-resilient adversary $\mathcal{B}_{(\mathsf{op},12)}$ with $\mathcal{A}_{(\mathsf{op},12)}$

For a queried message m to Td oracle, $\mathcal{B}_{(\mathsf{op},12)}$ computes the message-specific token t_m by querying m to the key extraction oracle IExt provided to a k-resilient attacker $\mathcal{B}_{(\mathsf{op},12)}$. Namely the message-specific token t_m is a user's secret key of $\mathcal{ID}\text{-}\mathcal{KEM}$ when the message is considered as an ID.

By the description of $\mathcal{B}_{(\mathsf{op},12)}$ and the explanation above, the game between $\mathcal{B}_{(\mathsf{op},12)}$ and $\mathcal{A}_{(\mathsf{op},12)}$ coincides with Game_1 when $b = 0$ (Ch_0 is given), and it also coincides with Game_2 when $b = 1$ (Ch_1 is given).

Then it follows that

$$\mathsf{Win}_1 = \Pr[\mathcal{B}_{(\mathsf{op},12)} \text{ outputs } 1 | b = 0] \text{ and } \mathsf{Win}_2 = \Pr[\mathcal{B}_{(\mathsf{op},12)} \text{ outputs } 1 | b = 1].$$

Moreover, we have

$$\begin{aligned}\Pr[\mathcal{B}_{(\mathsf{op},12)} \text{ outputs } 1 | b = 0] &= \Pr[\mathsf{Exp}^{\mathsf{k\text{-}resi},0}_{\mathcal{KEM},\mathcal{B}_{(\mathsf{op},12)}}(\lambda) = 1],\\ \Pr[\mathcal{B}_{(\mathsf{op},12)} \text{ outputs } 1 | b = 1] &= \Pr[\mathsf{Exp}^{\mathsf{k\text{-}resi},1}_{\mathcal{KEM},\mathcal{B}_{(\mathsf{op},12)}}(\lambda) = 1].\end{aligned}$$

Then,

$$\begin{aligned}\mathsf{Adv}^{\mathsf{k\text{-}resi}}_{\mathcal{ID}\text{-}\mathcal{KEM},\mathcal{B}_{(\mathsf{op},12)}}(\lambda) &= |\Pr[\mathsf{Exp}^{\mathsf{k\text{-}resi},0}_{\mathcal{ID}\text{-}\mathcal{KEM},\mathcal{B}_{(\mathsf{op},12)}}(\lambda) = 1] - \Pr[\mathsf{Exp}^{\mathsf{k\text{-}resi},1}_{\mathcal{ID}\text{-}\mathcal{KEM},\mathcal{B}_{(\mathsf{op},12)}}(\lambda) = 1]|\\ &= |\,\mathsf{Win}_1 - \mathsf{Win}_2\,|,\end{aligned}$$

follows. Since $\mathcal{ID}\text{-}\mathcal{KEM}$ is k-resilient by the assumption on the theorem, the statement holds. □

Game_3.

Game_3 coincides with Game_2 except that the proof τ_1 in GSig is generated honestly. Since Π_1 is an NIZK, the difference between Game_3 and Game_2 is bounded by some negligible function $\mathsf{negl}_{\mathsf{op},23}$.

$$|\,\mathsf{Win}_3 - \mathsf{Win}_2\,| \leq \mathsf{negl}_{\mathsf{op},23}(\lambda).$$

Game_3 is equivalent with $\mathsf{Exp}^{\mathsf{op\text{-}anon},1}_{\mathcal{DGS}\text{-}\mathcal{MDO},\mathcal{A}_{\mathsf{op}}}(\lambda)$ by its description. Thus we have

$$\mathsf{Win}_3 = \Pr[\mathsf{Exp}^{\mathsf{op\text{-}anon},1}_{\mathcal{DGS}\text{-}\mathcal{MDO},\mathcal{A}_{\mathsf{op}}}(\lambda) = 1].$$

Finally, we have

$$|\Pr[\mathsf{Exp}^{\mathsf{op\text{-}anon},0}_{\mathcal{DGS}\text{-}\mathcal{MDO},\mathcal{A}_{\mathsf{op}}}(\lambda) = 1] - \Pr[\mathsf{Exp}^{\mathsf{op\text{-}anon},1}_{\mathcal{DGS}\text{-}\mathcal{MDO},\mathcal{A}_{\mathsf{op}}}(\lambda) = 1]| \leq \mathsf{negl}_{\mathsf{op}}(\lambda),$$

for a negligible function $\mathsf{negl}_{\mathsf{op}}$ and the statement holds.

□

Corollary 1. *Assume that $\mathcal{ID}$-$\mathcal{KEM}$ is IND-ID-CPA and Π_1 is an NIZK. Then $\mathcal{DGS}$-$\mathcal{MDO}$ has the opener anonymity.*

Theorem 3. *Assume that $\mathcal{KEM}$ is IND-CCA, Π_1 is a simulation-sound NIZK and Π_2 is an NIZK. Then $\mathcal{DGS}$-$\mathcal{MDO}$ has the admitter anonymity.*

Proof. We consider the sequence of games Game between the challenger $\mathcal{C}_{\mathsf{ad}}$ and the adversary $\mathcal{A}_{\mathsf{ad}}$. For each $0 \leq \ell \leq 3$, let Win_ℓ be the probability of the event where $\mathcal{C}_{\mathsf{ad}}$ outputs 1 in Game_ℓ.

Game_0.

Game_0 coincides with the experiment $\mathsf{Exp}^{\mathsf{ad\text{-}anon},0}_{\mathcal{DGS}\text{-}\mathcal{MDO},\mathcal{A}_{\mathsf{ad}}}(\lambda)$. Thus we have

$$\mathsf{Win}_0 = \Pr[\mathsf{Exp}^{\mathsf{ad\text{-}anon},0}_{\mathcal{DGS}\text{-}\mathcal{MDO},\mathcal{A}_{\mathsf{ad}}}(\lambda) = 1].$$

Game_1.

Game_1 coincides with Game_0 except that the proof τ_1 generated in GSig and the proof τ_2 generated in Open are computed by the simulator $\mathsf{Sim}_1 = (S_{1,1}, S_{1,2})$ for the proof system Π_1 and the simulator $\mathsf{Sim}_2 = (S_{2,1}, S_{2,2})$ for the proof system Π_2, respectively. Since Π_1 and Π_2 are NIZKs, the difference between Game_0 and Game_1 is bounded by some negligible function $\mathsf{negl}_{\mathsf{ad},01}$.

$$|\,\mathsf{Win}_1 - \mathsf{Win}_0\,| \leq \mathsf{negl}_{\mathsf{ad},01}(\lambda).$$

Game_2.

Game_2 coincides with Game_1 except that the challenge oracle Ch_0 is replaced with Ch_1. For the difference between Win_1 and Win_2, we have the following lemma.

Lemma 2.

$$|\,\mathsf{Win}_2 - \mathsf{Win}_1\,| \leq \mathsf{negl}_{\mathit{ad},12}(\lambda),$$

for some negligible function $\mathsf{negl}_{\mathit{ad},12}$.

Proof. Let $\mathcal{A}_{(\mathsf{ad},12)}$ be an adversary which participates in Game_1 or Game_2. We aim to construct a PPT algorithm $\mathcal{B}_{(\mathsf{ad},12)}$ such that if $\mathcal{B}_{(\mathsf{ad},12)}$ breaks the IND-CCA of $\mathcal{KEM}$ with $\mathcal{A}_{(\mathsf{ad},12)}$, the difference between Win_1 and Win_2 can be non-negligible. The description of the algorithm $\mathcal{A}_{(\mathsf{ad},12)}$ is given in Figure 2.

Let pk^*_e be an instance given to the IND-CCA adversary $\mathcal{B}_{(\mathsf{ad},12)}$. Then $\mathcal{B}_{(\mathsf{ad},12)}$ invokes Game_{1+b} for $b \in \{0,1\}$ with the adversary $\mathcal{A}_{(\mathsf{ad},12)}$. $\mathcal{B}_{(\mathsf{ad},12)}$ generates public and secret keys $(\mathsf{pk}_s, \mathsf{sk}_s)$ and $(\mathsf{pp}, \mathsf{msk})$, simulated CRSs crs_1 and crs_2. Then $\mathcal{B}_{(\mathsf{ad},12)}$ sets $\mathsf{gpk} = (\mathsf{pk}_s, \mathsf{pk}^*_e, \mathsf{pp}, \mathsf{crs}_1, \mathsf{crs}_2)$, $\mathsf{ik} = \mathsf{sk}_s$, $\mathsf{ak} = \mathsf{msk}$, and gives $(\mathsf{gpk}, \mathsf{ak}, \mathsf{ik})$ to $\mathcal{A}_{(\mathsf{ad},12)}$ as input.

For oracle queries from $\mathcal{A}_{(\mathsf{ad},12)}$, $\mathcal{B}_{(\mathsf{ad},12)}$ can answer queries honestly except Ch_b oracle and Open oracle. For a challenge oracle query, $\mathcal{B}_{(\mathsf{ad},12)}$ calls its challenge oracle $\mathsf{Ch}_{\mathcal{KEM},b}$ to obtain $(C^*_{\mathsf{KEM}}, K^*_{\mathsf{KEM}})$ and then computes a group signature. Note that $\mathcal{B}_{(\mathsf{ad},12)}$ does not know the secret key corresponding to pk^*_e. Thus $\mathcal{B}_{(\mathsf{ad},12)}$ uses the simulator for the proof system Π_1 to compute the proof τ^*_1.

For queries to Open oracle, $\mathcal{B}_{(\mathsf{ad},12)}$ uses its decryption oracle $O_{\mathcal{KEM},\mathsf{Dec}}$ to decapsulate C_{KEM} since $\mathcal{B}_{(\mathsf{ad},12)}$ is now a CCA adversary. Then $\mathcal{B}_{(\mathsf{ad},12)}$ computes the proof τ_2 by using the simulator

```
$\mathcal{B}^{O_{\mathcal{KEM},\mathsf{Dec}}}_{(\mathsf{ad},12),b}(\mathsf{pk}^*_e)$
$\mathbf{HU}, \mathbf{CU}, \mathbf{RU}, \mathbf{CL}, \mathbf{CT} = \emptyset$
$(\mathsf{pk}_s, \mathsf{sk}_s) \leftarrow \mathsf{SKg}(1^\lambda), (\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{ISetup}(1^\lambda)$
$(\mathsf{crs}_1, \mathsf{st}_1) \leftarrow S_{1,1}(1^\lambda), (\mathsf{crs}_2, \mathsf{st}_2) \leftarrow S_{2,1}(1^\lambda)$
$\mathsf{gpk} = (\mathsf{pk}_s, \mathsf{pk}^*_e, \mathsf{pp}, \mathsf{crs}_1, \mathsf{crs}_2), \mathsf{ik} = \mathsf{sk}_s, \mathsf{ak} = \mathsf{msk}$
$b' \leftarrow \mathcal{A}^{\mathsf{CrptU},\mathsf{SndToU},\mathsf{USK},\mathsf{WReg},\mathsf{Open},\mathsf{Ch}_b}_{(\mathsf{ad},12)}(\mathsf{gpk}, \mathsf{ak}, \mathsf{ik})$
$\mathbf{return}\ b'$
```

```
$\mathsf{Ch}_b(m, i_0, i_1)$
$s_{i_b} \leftarrow \mathsf{Sig}(\mathsf{sk}_{i_b}, m)$
$(C^*_{\mathsf{KEM}}, K^*_{\mathsf{KEM}}) \leftarrow \mathsf{Ch}_{\mathcal{KEM},b}(\mathsf{pk}^*_e)$
$\mathbf{CT} \leftarrow \mathbf{CT} \cup \{C^*_{\mathsf{KEM}}\}$
$\mathsf{r}_{\mathsf{IDKEM}} \xleftarrow{\$} \{0,1\}^\lambda$
$(C^*_{\mathsf{IDKEM}}, K^*_{\mathsf{IDKEM}}) \leftarrow \mathsf{IEnc}(\mathsf{pp}, m; \mathsf{r}_{\mathsf{IDKEM}})$
$\kappa^* = \langle i_b, \mathsf{pk}_{i_b}, \mathsf{cert}_{i_b}, s_{i_b} \rangle \odot K^*_{\mathsf{KEM}} \odot K^*_{\mathsf{IDKEM}}$
$x^*_1 = (\mathsf{pk}_s, \mathsf{pk}^*_e, \mathsf{pp}, m, C^*_{\mathsf{KEM}}, C^*_{\mathsf{IDKEM}}, \kappa^*)$
$\tau^*_1 = S_{1,2}(\mathsf{crs}_1, \mathsf{st}_1, x^*_1)$
$\mathbf{CL} \leftarrow \mathbf{CL} \cup \{(m, \sigma^*)\}$
$\mathbf{return}\ \sigma^* = (C^*_{\mathsf{KEM}}, C^*_{\mathsf{IDKEM}}, \kappa^*, \tau^*_1)$
```

```
$\mathsf{Open}(m, \sigma)$
$\mathbf{if}\ (m, \sigma) \in \mathbf{CL}\ \mathbf{then\ return}\ \perp$
$x_1 = (\mathsf{pk}_s, \mathsf{pk}^*_e, \mathsf{pp}, m, C_{\mathsf{KEM}}, C_{\mathsf{IDKEM}}, \kappa)$
$\mathbf{if}\ \mathsf{V}_1(\mathsf{crs}_1, x_1, \tau_1) = 0\ \mathbf{then\ return}\ (0, \epsilon)$
$K_{\mathsf{KEM}} \leftarrow O_{\mathcal{KEM},\mathsf{Dec}}(C_{\mathsf{KEM}}), \mathbf{CT} \leftarrow \mathbf{CT} \cup \{C_{\mathsf{KEM}}\}$
$\mathsf{t}_m \leftarrow \mathsf{IExt}(\mathsf{msk}, m), K_{\mathsf{IDKEM}} \leftarrow \mathsf{IDec}(\mathsf{t}_m, C_{\mathsf{IDKEM}}, m)$
$\mathbf{if}\ K_{\mathsf{KEM}} = \perp \text{ or } K_{\mathsf{IDKEM}} = \perp\ \mathbf{then\ return}\ (0, \epsilon)$
$\langle i, \mathsf{pk}_i, \mathsf{cert}_i, s_i \rangle = \kappa \odot K^{-1}_{\mathsf{KEM}} \odot K^{-1}_{\mathsf{IDKEM}}$
$x_2 = (\mathsf{pk}^*_e, C_{\mathsf{KEM}}, C_{\mathsf{IDKEM}}, m, \kappa, i, \mathsf{pk}_i, \mathsf{cert}_i, s_i)$
$\tau_2 = S_{2,2}(\mathsf{crs}_2, \mathsf{st}_2, x_2)$
$\pi = (\sigma_i, i, \mathsf{pk}_i, \mathsf{cert}_i, s_i, \tau_2)$
$\mathbf{return}\ (i, \pi)$
```

Figure 2: The IND-CCA adversary $\mathcal{B}_{(\mathsf{ad},12)}$ with $\mathcal{A}_{(\mathsf{ad},12)}$

for the proof system Π_2. We note the case where $\mathcal{A}_{(\mathsf{ad},12)}$ queries a group signature σ' which is the same as the challenged signature σ^* except for the attached proof. Namely $\mathcal{A}_{(\mathsf{ad},12)}$ aims to obtain the user index i_b of the challenge group signature by queried it to Open oracle with another proof τ'_1. According to the definition of games, such σ' is not considered as the challenged group signature σ^* since the proof part τ_1 of the group signature differs. However, $\mathcal{A}_{(\mathsf{ad},12)}$ cannot make such τ'_1 which passes the verification of V_1 because Π_1 is assumed to have the simulation-soundness. Thus $\mathcal{B}_{(\mathsf{ad},12)}$ correctly simulates Open oracle.

By the description of $\mathcal{B}_{(\mathsf{ad},12)}$ and the explanation above, the game between $\mathcal{B}_{(\mathsf{ad},12)}$ and $\mathcal{A}_{(\mathsf{ad},12)}$ coincides with Game_1 when $b = 0$ (Ch_0 is given), and it also coincides with Game_2 when $b = 1$ (Ch_1 is given).

Then it follows that

$$\mathsf{Win}_1 = \Pr[\mathcal{B}_{(\mathsf{ad},12)} \text{ outputs } 1 | b = 0] \text{ and } \mathsf{Win}_2 = \Pr[\mathcal{B}_{(\mathsf{ad},12)} \text{ outputs } 1 | b = 1].$$

Moreover, we have

$$\Pr[\mathcal{B}_{(\mathsf{ad},12)} \text{ outputs } 1 | b = 0] = \Pr[\mathsf{Exp}^{\mathsf{ind\text{-}cca},0}_{\mathcal{KEM},\mathcal{B}_{(\mathsf{ad},12)}}(\lambda) = 1],$$
$$\Pr[\mathcal{B}_{(\mathsf{ad},12)} \text{ outputs } 1 | b = 1] = \Pr[\mathsf{Exp}^{\mathsf{ind\text{-}cca},1}_{\mathcal{KEM},\mathcal{B}_{(\mathsf{ad},12)}}(\lambda) = 1].$$

Then,

$$\begin{aligned}\mathsf{Adv}^{\mathsf{ind\text{-}cca}}_{\mathcal{KEM},\mathcal{B}_{(\mathsf{ad},12)}}(\lambda) &= |\Pr[\mathsf{Exp}^{\mathsf{ind\text{-}cca},0}_{\mathcal{KEM},\mathcal{B}_{(\mathsf{ad},12)}}(\lambda) = 1] - \Pr[\mathsf{Exp}^{\mathsf{ind\text{-}cca},1}_{\mathcal{KEM},\mathcal{B}_{(\mathsf{ad},12)}}(\lambda) = 1]| \\ &= |\mathsf{Win}_1 - \mathsf{Win}_2|,\end{aligned}$$

```
$\mathcal{B}_{\mathsf{trace}}^{O_{\mathcal{DS},\mathsf{Sig}}}(\mathsf{pk}_s^*)$
    $\mathbf{HU}, \mathbf{CU}, \mathbf{RU} = \emptyset$
    $((\mathsf{pk}_s, \mathsf{pk}_e, \mathsf{pp}, \mathsf{crs}_1, \mathsf{crs}_2), \mathsf{sk}_s, (\mathsf{sk}_e, \mathsf{r}_e), \mathsf{msk}) \leftarrow \mathsf{GKg}(1^\lambda)$
    $\mathsf{gpk} = (\mathsf{pk}_s^*, \mathsf{pk}_e, \mathsf{pp}, \mathsf{crs}_1, \mathsf{crs}_2), \mathsf{ok} = (\mathsf{sk}_e, \mathsf{r}_e), \mathsf{ak} = \mathsf{msk}$
    $(m^*, \sigma^*) \leftarrow \mathcal{A}_{\mathsf{trace}}^{\mathsf{AddU},\mathsf{CrptU},\mathsf{SndToI},\mathsf{USK},\mathsf{RReg}}(\mathsf{gpk}, \mathsf{ok}, \mathsf{ak})$
    if $\mathsf{GVf}(\mathsf{gpk}, m^*, \sigma^*) = 0$ then return $0$
    $\mathsf{t}_{m^*} \leftarrow \mathsf{Td}(\mathsf{gpk}, \mathsf{ak}, m^*)$
    $(i^*, \pi^*) \leftarrow \mathsf{Open}(\mathsf{gpk}, \mathsf{ok}, \mathbf{reg}, m^*, \sigma^*, \mathsf{t}_{m^*})$
    if $i \neq 0$ then return $0$
    $K_{\mathsf{KEM}}^* \leftarrow \mathsf{Dec}(\mathsf{sk}_e, C_{\mathsf{KEM}}^*), K_{\mathsf{IDKEM}}^* \leftarrow \mathsf{IDec}(\mathsf{t}_{m^*}, C_{\mathsf{IDKEM}}^*, m^*)$
    $\langle 0, \mathsf{pk}_0, \mathsf{cert}_0, s_0 \rangle = \kappa^* \odot {K_{\mathsf{KEM}}^*}^{-1} \odot {K_{\mathsf{IDKEM}}^*}^{-1}$
    return $(\langle 0, \mathsf{pk}_0 \rangle, \mathsf{cert}_0)$
```

```
$\mathsf{CrptU}(i, \mathsf{upk})$
    $\mathbf{CU} \leftarrow \mathbf{CU} \cup \{i\}$
    $\mathsf{upk}_i \leftarrow \mathsf{upk}$
    return $1$
```

```
$\mathsf{USK}(i)$
    $\mathbf{RU} \leftarrow \mathbf{RU} \cup \{i\}$
    return $(\mathsf{usk}_i, \mathsf{gsk}_i)$
```

```
$\mathsf{RReg}(i)$
    return $\mathbf{reg}[i]$
```

```
$\mathsf{AddU}(i)$
    $\mathbf{HU} \leftarrow \mathbf{HU} \cup \{i\}$
    $(\mathsf{upk}_i, \mathsf{usk}_i) \leftarrow \mathsf{UKg}(1^\lambda, \mathsf{gpk})$
    $\mathsf{gsk}_i \leftarrow \mathsf{Join}(\mathsf{gpk}, \mathsf{upk}_i, \mathsf{usk}_i)$ with $\mathsf{SndToI}$
    return $\mathsf{upk}_i$
```

```
$\mathsf{SndToI}(i, \mathsf{pk}_i, \sigma_i)$
    $\mathsf{cert}_i \leftarrow O_{\mathcal{DS},\mathsf{Sig}}(\langle i, \mathsf{pk}_i \rangle)$
    $\mathbf{reg}[i] \leftarrow (\mathsf{pk}_i, \sigma_i)$
    return $\mathsf{cert}_i$
```

Figure 3: The EUF-CMA adversary $\mathcal{B}_{\mathsf{trace}}$ with the traceablity adversary $\mathcal{A}_{\mathsf{trace}}$

follows. Since $\mathcal{KEM}$ is IND-CCA by the assumption on the theorem, the statement holds. □

Game_3.

Game_3 coincides with Game_2 except that the proof τ_1 in GSig and the proof τ_2 generated in Open are generated honestly. Since Π_1 and Π_2 are NIZKs, the difference between Game_3 and Game_2 is bounded by some negligible function $\mathsf{negl}_{\mathsf{ad},23}$.

$$|\,\mathsf{Win}_3 - \mathsf{Win}_2\,| \leq \mathsf{negl}_{\mathsf{ad},23}(\lambda).$$

Game_3 is equivalent with $\mathsf{Exp}_{\mathcal{DGS}\text{-}\mathcal{MDO},\mathcal{A}_{\mathsf{ad}}}^{\mathsf{ad\text{-}anon},1}(\lambda)$ by its description. Thus we have

$$\mathsf{Win}_3 = \Pr[\mathsf{Exp}_{\mathcal{DGS}\text{-}\mathcal{MDO},\mathcal{A}_{\mathsf{ad}}}^{\mathsf{ad\text{-}anon},1}(\lambda) = 1].$$

Finally, we have

$$|\Pr[\mathsf{Exp}_{\mathcal{DGS}\text{-}\mathcal{MDO},\mathcal{A}_{\mathsf{ad}}}^{\mathsf{ad\text{-}anon},0}(\lambda) = 1] - \Pr[\mathsf{Exp}_{\mathcal{DGS}\text{-}\mathcal{MDO},\mathcal{A}_{\mathsf{ad}}}^{\mathsf{ad\text{-}anon},1}(\lambda) = 1]| \leq \mathsf{negl}_{\mathsf{ad}}(\lambda),$$

for a negligible function $\mathsf{negl}_{\mathsf{ad}}$ and the statement holds. □

Theorem 4. *Assume that $\mathcal{DS}$ is EUF-CMA and Π_1 and Π_2 are NIZKs. Then $\mathcal{DGS}\text{-}\mathcal{MDO}$ has the traceability.*

Proof. Let $\mathcal{A}_{\mathsf{trace}}$ be an adversary which breaks the traceability of $\mathcal{DGS}\text{-}\mathcal{MDO}$ with non-negligible probability. We aim to construct a PPT algorithm $\mathcal{B}_{\mathsf{trace}}$ which breaks the EUF-CMA of $\mathcal{DS}$ with non-negligible probability with the help of $\mathcal{A}_{\mathsf{trace}}$. The description of the algorithm $\mathcal{B}_{\mathsf{trace}}$ is given in Figure 3.

Let pk_s^* be an instance given to $\mathcal{B}_{\mathsf{trace}}$ in the EUF-CMA experiment $\mathsf{Exp}_{\mathcal{DS},\mathcal{B}_{\mathsf{trace}}}^{\mathsf{euf\text{-}cma}}$. Then $\mathcal{B}_{\mathsf{trace}}$ invokes the traceability experiment $\mathsf{Exp}_{\mathcal{DGS}\text{-}\mathcal{MDO},\mathcal{A}_{\mathsf{trace}}}^{\mathsf{trace}}(\lambda)$ with the adversary A_{trace}. $\mathcal{B}_{\mathsf{trace}}$ performs the group-key generation algorithm GKg and obtain $((\mathsf{pk}_s, \mathsf{pk}_e, \mathsf{pp}, \mathsf{crs}_1, \mathsf{crs}_2), \mathsf{sk}_s, (\mathsf{sk}_e, \mathsf{r}_e), \mathsf{msk})$. Then

$\mathcal{B}_{\mathsf{trace}}$ sets $\mathsf{gpk} = (\mathsf{pk}_s^*, \mathsf{pk}_e, \mathsf{pp}, \mathsf{crs}_1, \mathsf{crs}_2)$, $\mathsf{ok} = (\mathsf{sk}_e, \mathsf{r}_e)$, $\mathsf{ak} = \mathsf{msk}$, and gives $(\mathsf{gpk}, \mathsf{ok}, \mathsf{ak})$ to $\mathcal{A}_{\mathsf{trace}}$ as input.

$\mathcal{B}_{\mathsf{trace}}$ can answer all oracle queries from $\mathcal{A}_{\mathsf{trace}}$ honestly except queries to AddU oracle and SndToI oracle. This is because $\mathcal{B}_{\mathsf{trace}}$ embeds pk_s^* into gpk and hence $\mathcal{B}_{\mathsf{trace}}$ does not know the corresponding secret key. However, $\mathcal{B}_{\mathsf{trace}}$ is required to sign $\langle i, \mathsf{pk}_i \rangle$ to create the certification cert_i which is verifiable with pk_s^*, to answer AddU or SndToI oracle queries. To address this problem, $\mathcal{B}_{\mathsf{trace}}$ uses the provided signing oracle $O_{\mathcal{DS},\mathsf{Sig}}$ since $\mathcal{B}_{\mathsf{trace}}$ is now an EUF-CMA adversary. $\mathcal{B}_{\mathsf{trace}}$ queries $\langle i, \mathsf{pk}_i \rangle$ to the oracle $O_{\mathcal{DS},\mathsf{Sig}}$ and obtains $\mathsf{cert}_i \leftarrow \mathsf{Sig}(\mathsf{sk}_s^*, \langle i, \mathsf{pk}_i \rangle)$. Thus $\mathcal{B}_{\mathsf{trace}}$ can answer all oracle queries from $\mathcal{A}_{\mathsf{trace}}$ correctly.

Let $(m^*, \sigma^*) = (m^*, (C^*_{\mathsf{KEM}}, C^*_{\mathsf{IDKEM}}, \kappa^*, \tau_1^*))$ be an output of $\mathcal{A}_{\mathsf{trace}}$. By the assumption, $\mathcal{A}_{\mathsf{trace}}$ wins the experiment with the output (m^*, σ^*) with the non-negligible probability. Then we have GVf outputs 1, and (i) Open outputs $i^* = 0$ or (ii) Open outputs $i^* \neq 0$ and Judge outputs 0.

We first consider the case (i). In this case $\mathsf{Open}(\mathsf{gpk}, \mathsf{ok}, \mathbf{reg}, m^*, \sigma^*, \mathsf{Td}(\mathsf{gpk}, \mathsf{ak}, m^*))$ outputs $(0, \pi^*)$. This means that the group signature σ^* is not issued on behalf of legitimate users although (m^*, σ^*) passes the verification algorithm GVf. Now, $\mathsf{GVf}(\mathsf{gpk}, m^*, \sigma^*) = 1$ implies $\mathsf{V}_1(\mathsf{crs}_1, x_1^*, \tau_1^*) = 1$ with $x_1^* = (\mathsf{pk}_s^*, \mathsf{pk}_e, \mathsf{pp}, m^*, C^*_{\mathsf{KEM}}, C^*_{\mathsf{IDKEM}}, \kappa^*)$. τ_1^* is a proof generated by the proof system Π_1 which proves the honest executions of GSig. Since Π_1 is an NIZK by the statement, we can decrypt C^*_{KEM} and C^*_{IDKEM} correctly, and then can retrieve $\langle 0, \mathsf{pk}_0, \mathsf{cert}_0, s_0 \rangle$ from κ^*. In this case (i), pk_0 is not recorded to the registration table $\mathbf{reg}$ since $i^* = 0$ means an illicit user. Namely AddU oracle or SndToI oracle does not invoked for this user, and hence $\langle 0, \mathsf{pk}_0 \rangle$ is not queried to the CMA oracle. Then $(\langle 0, \mathsf{pk}_0 \rangle, \mathsf{cert}_0)$ can be a valid forgery of $\mathcal{DS}$ and $\mathcal{B}_{\mathsf{trace}}$ wins the EUF-CMA experiment by the forgery $(\langle 0, \mathsf{pk}_0 \rangle, \mathsf{cert}_0)$ with the non-negligible probability.

We next consider the case (ii). In this case, $\mathsf{Open}(\mathsf{gpk}, \mathsf{ok}, \mathbf{reg}, m^*, \sigma^*, \mathsf{Td}(\mathsf{gpk}, \mathsf{ak}, m^*))$ outputs (i^*, π^*) for some $i^* \neq 0$ and $\mathsf{Judge}(\mathsf{gpk}, i, \mathsf{upk}_{i^*}, m^*, \sigma^*, \pi^*) = 0$. Since $\mathsf{GVf}(\mathsf{gpk}, m^*, \sigma^*) = 1$ as in the case (i), it is ensured that the signing procedure is honestly done by the soundness of Π_1. Then C^*_{KEM} and C^*_{IDKEM} can be decrypted correctly and $\langle i^*, \mathsf{pk}_{i^*}, \mathsf{cert}_{i^*}, s_{i^*} \rangle$ is recovered from κ^*. $i^* \neq 0$ means that the user i^* is a legitimate user, hence $\mathcal{B}_{\mathsf{trace}}$ can obtain $\mathbf{reg}[i^*] = (pk_{i^*}, \sigma_{i^*})$. Therefore, if $\mathcal{B}_{\mathsf{trace}}$ honestly generates $\pi^* = (\sigma_{i^*}, i^*, \mathsf{pk}_{i^*}, \mathsf{cert}_{i^*}, s_{i^*}, \tau_2^*)$ with $\tau_2^* = \mathsf{P}_2(\mathsf{crs}_2, x_2^*, w_2^*)$ for $x_2^* = (\mathsf{pk}_e, C^*_{\mathsf{KEM}}, C^*_{\mathsf{IDKEM}}, m^*, \kappa^*, i^*, \mathsf{pk}_{i^*}, \mathsf{cert}_{i^*}, s_{i^*})$ and $w_2^* = (\mathsf{sk}_e, \mathsf{r}_e, \mathsf{t}_{m^*})$, $\mathsf{V}_2(\mathsf{crs}_2, x_2^*, \tau_2^*)$ always outputs 1 by the completeness of Π_2 and then $\mathsf{Judge}(\mathsf{gpk}, i, \mathsf{upk}_{i^*}, m^*, \sigma^*, \pi^*) = 1$ always holds. This means that the case (ii) never occurs.

The case (i) is the only case when $\mathcal{A}_{\mathsf{trace}}$ breaks the traceability of $\mathcal{DGS}$-$\mathcal{MDO}$ by the discussion above. Thus if $\mathcal{A}_{\mathsf{trace}}$ breaks the traceability of $\mathcal{DGS}$-$\mathcal{MDO}$ with non-negligible probability, $\mathcal{B}_{\mathsf{trace}}$ breaks the EUF-CMA of $\mathcal{DS}$ with non-negligible probability. □

Theorem 5. *Assume that $\mathcal{DS}$ is EUF-CMA and Π_1 and Π_2 are NIZKs. Then $\mathcal{DGS}$-$\mathcal{MDO}$ has the non-frameability.*

Proof. Let $\mathcal{A}_{\mathsf{nf}}$ be an adversary which breaks the non-frameability of $\mathcal{DGS}$-$\mathcal{MDO}$ with non-negligible probability. We aim to construct a PPT algorithm $\mathcal{B}_{\mathsf{nf}}$ which breaks the EUF-CMA of $\mathcal{DS}$ with non-negligible probability with the help of $\mathcal{A}_{\mathsf{nf}}$. The description of the algorithm $\mathcal{B}_{\mathsf{nf}}$ is given in Figure 4.

Let pk_s^* be an instance given to $\mathcal{B}_{\mathsf{nf}}$ in the EUF-CMA experiment $\mathsf{Exp}^{\mathsf{euf\text{-}cma}}_{\mathcal{DS},\mathcal{B}_{\mathsf{nf}}}$. Then $\mathcal{B}_{\mathsf{nf}}$ invokes the non-frameability experiment $\mathsf{Exp}^{\mathsf{nf}}_{\mathcal{DGS}\text{-}\mathcal{MDO},\mathcal{A}_{\mathsf{nf}}}(\lambda)$ with the adversary $\mathcal{A}_{\mathsf{nf}}$. We suppose that the adversary $\mathcal{A}_{\mathsf{nf}}$ creates $n(\lambda)$ users for a polynomial n. The fundamental strategy of $\mathcal{B}_{\mathsf{nf}}$ is as follows. $\mathcal{B}_{\mathsf{nf}}$ guesses the user u with which $\mathcal{A}_{\mathsf{nf}}$ outputs a forgery. When $\mathcal{A}_{\mathsf{nf}}$ register the user u via the SndToU oracle, $\mathcal{B}_{\mathsf{nf}}$ embeds the instance public key pk_s^* into registration information $\mathbf{reg}[u]$. For group signature queries with respect to the user u, $\mathcal{B}_{\mathsf{nf}}$ answers queries by using the provided signing oracle $O_{\mathcal{DS},\mathsf{Sig}}$ since $\mathcal{B}_{\mathsf{nf}}$ is now an EUF-CMA adversary. If $\mathcal{A}_{\mathsf{nf}}$ wins the non-frameability experiment by an output with respect to the user u, $\mathcal{B}_{\mathsf{nf}}$ can extract a valid signature with respect to pk_s^* and it can be regarded as a valid forgery of $\mathcal{DS}$.

However, we must note that $\mathcal{A}_{\mathsf{nf}}$ may not use the registration information which is recorded via SndToU oracle. Namely $\mathcal{A}_{\mathsf{nf}}$ can change the registration table $\mathbf{reg}$ via the WReg oracle. In this case, the $\mathcal{B}_{\mathsf{nf}}$'s strategy above does not work and $\mathcal{B}_{\mathsf{nf}}$ should take another approach. If $\mathcal{B}_{\mathsf{nf}}$ guesses that

$\mathcal{A}_{\mathsf{nf}}$ will change $\mathbf{reg}[u]$, $\mathcal{B}_{\mathsf{nf}}$ signs the public key when it is recorded to $\mathbf{reg}[u]$ under the instance public key pk_s^* by using the oracle $O_{\mathcal{DS},\mathsf{Sig}}$. Since the output of $\mathcal{A}_{\mathsf{nf}}$ must pass Judge, $\mathcal{B}_{\mathsf{nf}}$ can extract a valid signature with respect to pk_s^* and it can be regarded as a valid forgery of $\mathcal{DS}$. Of course $\mathcal{B}_{\mathsf{nf}}$ does not know whether or not $\mathcal{A}_{\mathsf{nf}}$ uses WReg. Thus $\mathcal{B}_{\mathsf{nf}}$ first guess the behavior of $\mathcal{A}_{\mathsf{nf}}$ and $\mathcal{B}_{\mathsf{nf}}$ wins the EUF-CMA experiment if the guess is right.

Let $(m^*, \sigma^*) = (m^*, (C^*_{\mathsf{KEM}}, C^*_{\mathsf{IDKEM}}, \kappa^*, \tau_1^*))$ be an output of $\mathcal{A}_{\mathsf{nf}}$. By the assumption, $\mathcal{A}_{\mathsf{nf}}$ wins the experiment with the output (m^*, σ^*) with the non-negligible probability. Then we have GVf outputs 1. According to the guess on the behavior of $\mathcal{A}_{\mathsf{nf}}$, $\mathcal{B}_{\mathsf{nf}}$ extracts different parts as his forgery from $\mathcal{A}_{\mathsf{nf}}$'s outputs. The details are as follows.

We first consider the case where $\mathcal{A}_{\mathsf{nf}}$ uses the public key stored in $\mathbf{reg}$ without any change when it generates the forgery. Such a case corresponds to the case $b = 0$ in Figure 4. In this case, the public key recorded in the registration table $\mathbf{reg}[u]$ for the user u is pk_s^*. If $\mathcal{A}_{\mathsf{nf}}$ wins the non-frameability experiment, π^* output by $\mathcal{A}_{\mathsf{nf}}$ passes Judge. Then the signature σ_{i^*} included in π^* can be verified for the output message m^* under the public key pk_s^*. Now, this (i^*, m^*) is not queried to GSig by the winning condition of the non-frameability. This means that m^* is not queried to the signing oracle of the EUF-CMA experiment and (m^*, σ_{i^*}) is a valid forgery for $\mathcal{DS}$. Then $\mathcal{B}_{\mathsf{nf}}$ wins the EUF-CMA experiment with the forgery (m^*, σ_{i^*}) if $\mathcal{A}_{\mathsf{nf}}$ wins the non-frameability experiment.

We next consider the another case, namely $\mathcal{A}_{\mathsf{nf}}$ uses a different public key pk'_{i^*} in group signing than pk_{i^*} which is first recorded to $\mathbf{reg}$ via SndToU. This case corresponds to the case $b = 1$ in Figure 4. In this case, $\mathcal{B}_{\mathsf{nf}}$ cannot embed the public key pk_s^* into the oracle GSig since $\mathcal{A}_{\mathsf{nf}}$ change the registration table via calling WReg. Hence $\mathcal{B}_{\mathsf{nf}}$ aims to embed pk_s^* elsewhere. Note that $\mathcal{A}_{\mathsf{nf}}$ is assumed to win the non-frameability experiment, especially the output of $\mathcal{A}_{\mathsf{nf}}$ must pass Judge. This means that pk_{i^*} and σ_{i^*} included in π^* must be verified by upk_{i^*} by the description of Judge. Thus $\mathcal{B}_{\mathsf{nf}}$ sets pk_s^* as upk_u on t-th invocation of SndToU oracle. By the winning condition of $\mathcal{A}_{\mathsf{nf}}$ on the non-frameability experiment, $(\mathsf{pk}_{i^*}, \sigma_{i^*})$ must be accepted by Vf with respect to the public key $\mathsf{upk}_{i^*} = \mathsf{pk}_s^*$. Moreover, pk_{i^*} is not queried the signing oracle $O_{\mathcal{DS},\mathsf{Sig}}$ since we now suppose that $\mathcal{A}_{\mathsf{nf}}$ change the registration information $\mathbf{reg}[u]$ via WReg. Hence $(\mathsf{pk}_{i^*}, \sigma_{i^*})$ is a valid forgery for $\mathcal{DS}$ and $\mathcal{B}_{\mathsf{nf}}$ wins the EUF-CMA experiment.

From these discussion above, if $\mathcal{A}_{\mathsf{nf}}$ breaks the non-frameability of $\mathcal{DGS}$-$\mathcal{MDO}$ with non-negligible probability, $\mathcal{B}_{\mathsf{nf}}$ breaks the EUF-CMA of $\mathcal{DS}$ with non-negligible probability. □

Remark. Our construction can be converted to the DGS-MDO scheme which is identical to the GS-MDO scheme presented in [11] by replacing the KEM part in our scheme with a tag-based KEM. Moreover, the security proofs for our scheme can also be applied to the converted scheme as well.

4.4 Instantiation

We present concrete cryptographic primitives to instantiate our generic construction.

First, we employ the Groth-Sahai (GS) proof [13] which is based on the bilinear group as NIZK proof. Its security, including the simulation soundness, is proven in the standard model and it is known as one of the most efficient NIZK proofs.

Since we use the GS proof, we should choose other primitives as *structure-preserving.* This is because the verification formulas of the GS proof are pairing equations. From this viewpoint, we use the structure-preserving signature by Abe *et al.* [1] which is EUF-CMA in the standard model under the SFP assumption.

We require the *partially structure-preserving* property [15] for KEM and ID-KEM, which means that ciphertexts are in the *source group* $\mathbb{G}$ of the bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, not in the target group $\mathbb{G}_T$ to fit these primitives into the GS proof. Thus we choose the DLIN variant [20] of the Cramer-Shoup (CS) encryption [9] as KEM. It is partially structure-preserving and IND-CCA KEM in the standard model under the DLIN assumption.

As ID-KEM, we consider the DLIN variant [11] of the Heng-Kurosawa (HK) ID-KEM [14] which is partially structure-preserving and k-resilient in the standard model under the DLIN assumption.

By using these cryptographic schemes above, we can obtain a DGS-MDO scheme from our generic construction, which is secure in the standard model under the DLIN assumption and the

SFP assumption. However, it has only the opener anonymity with k-bounded tokens due to the k-resilience of the ID-KEM. It is an open question to find an ID-KEM which is partially structure-preserving and IND-ID-CPA in the standard model, and then to find a DGS-MDO scheme whose opener anonymity is satisfied with unbounded tokens.

5 Concluding Remarks

In this paper, We have introduced the definition of the *dynamic group signature with message dependent opening* (DGS-MDO) with the associated security requirements. We have also proposed a generic construction of DGS-MDO. from standard cryptographic primitives. Our construction can achieve the standard model security, constant signature size and non-interactive signing process. However, like the the scheme of [19], our scheme has one disadvantage, namely the bounded message opening. The DGS-MDO scheme that supports unbounded message opening is an interesting open question.

Acknowledgment

We would like to thank the anonymous reviewers for their valuable comments and suggestions. This work is supported by JSPS KAKENHI Grant Numbers JP19K20272 and JP22K12023, and the Telecommunication Advancement Foundation (TAF).

References

[1] Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Structure-preserving signatures and commitments to group elements. In Tal Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, pages 209–236, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[2] Hiroaki Anada, Masayuki Fukumitsu, and Shingo Hasegawa. Dynamic group signatures with message dependent opening and non-interactive signing. In *2022 Tenth International Symposium on Computing and Networking (CANDAR)*, pages 76–82, 2022.

[3] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In Eli Biham, editor, *Advances in Cryptology — EUROCRYPT 2003*, pages 614–629, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[4] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, CCS '93, page 62–73, New York, NY, USA, 1993. Association for Computing Machinery.

[5] Mihir Bellare, Haixia Shi, and Chong Zhang. Foundations of group signatures: The case of dynamic groups. In Alfred Menezes, editor, *Topics in Cryptology – CT-RSA 2005*, pages 136–153, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[6] K. Bentahar, P. Farshim, J. Malone-Lee, and N. P. Smart. Generic constructions of identity-based and certificateless kems. Cryptology ePrint Archive, Paper 2005/058, 2005. `https://eprint.iacr.org/2005/058`.

[7] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Essam Ghadafi, and Jens Groth. Foundations of fully dynamic group signatures. In Mark Manulis, Ahmad-Reza Sadeghi, and Steve Schneider, editors, *Applied Cryptography and Network Security*, pages 117–136, Cham, 2016. Springer International Publishing.

[8] David Chaum and Eugène Van Heyst. Group signatures. In *Proceedings of the 10th Annual International Conference on Theory and Application of Cryptographic Techniques*, EUROCRYPT'91, page 257–265, Berlin, Heidelberg, 1991. Springer-Verlag.

[9] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *Advances in Cryptology — CRYPTO '98*, pages 13–25, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[10] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003.

[11] Keita Emura, Goichiro Hanaoka, Yutaka Kawai, Takahiro Matsuda, Kazuma Ohara, Kazumasa Omote, and Yusuke Sakai. Group signatures with message-dependent opening: Formal definitions and constructions. *Sec. and Commun. Netw.*, 2019, jan 2019.

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

[13] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In *Proceedings of the Theory and Applications of Cryptographic Techniques 27th Annual International Conference on Advances in Cryptology*, EUROCRYPT'08, pages 415–432, Berlin, Heidelberg, 2008. Springer-Verlag.

[14] Swee-Huay Heng and Kaoru Kurosawa. k-resilient identity-based encryption in the standard model. In Tatsuaki Okamoto, editor, *Topics in Cryptology – CT-RSA 2004*, pages 67–80, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[15] Benoît Libert and Marc Joye. Group signatures with message-dependent opening in the standard model. In Josh Benaloh, editor, *Topics in Cryptology – CT-RSA 2014*, pages 286–306, Cham, 2014. Springer International Publishing.

[16] Benoît Libert, Fabrice Mouhartem, and Khoa Nguyen. A lattice-based group signature scheme with message-dependent opening. In Mark Manulis, Ahmad-Reza Sadeghi, and Steve Schneider, editors, *Applied Cryptography and Network Security*, pages 137–155, Cham, 2016. Springer International Publishing.

[17] Kazuma Ohara, Yusuke Sakai, Keita Emura, and Goichiro Hanaoka. A group signature scheme with unbounded message-dependent opening. In *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, ASIA CCS '13, page 517–522, New York, NY, USA, 2013. Association for Computing Machinery.

[18] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA*, pages 543–553. IEEE Computer Society, 1999.

[19] Yusuke Sakai, Keita Emura, Goichiro Hanaoka, Yutaka Kawai, Takahiro Matsuda, and Kazumasa Omote. Group signatures with message-dependent opening. In Michel Abdalla and Tanja Lange, editors, *Pairing-Based Cryptography – Pairing 2012*, pages 270–294, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[20] Hovav Shacham. A cramer-shoup encryption scheme from the linear assumption and from progressively weaker linear variants. Cryptology ePrint Archive, Paper 2007/074, 2007. `https://eprint.iacr.org/2007/074`.

[21] Yiru Sun and Yanyan Liu. An efficient fully dynamic group signature with message dependent opening from lattice. *Cybersecurity*, 4(1):15, May 2021.

```
$\mathcal{B}_{\mathsf{nf}}^{O_{\mathcal{DS},\mathsf{Sig}}}(\mathsf{pk}_s^*)$

$\mathbf{HU}, \mathbf{CU}, \mathbf{RU}, \mathbf{QL_{gs}} = \emptyset$
$b \xleftarrow{\$} \{0,1\}, t \xleftarrow{\$} \{1, \dots, n(\lambda)\}, \mathsf{count} = 0, u = \perp$
$((\mathsf{pk}_s, \mathsf{pk}_e, \mathsf{pp}, \mathsf{crs}_1, \mathsf{crs}_2), \mathsf{sk}_s, (\mathsf{sk}_e, \mathsf{r}_e), \mathsf{msk}) \leftarrow \mathsf{GKg}(1^\lambda)$
$\mathsf{gpk} = (\mathsf{pk}_s, \mathsf{pk}_e, \mathsf{pp}, \mathsf{crs}_1, \mathsf{crs}_2)$
$\mathsf{ik} = \mathsf{sk}_s, \mathsf{ok} = (\mathsf{sk}_e, \mathsf{r}_e), \mathsf{ak} = \mathsf{msk}$
$(m^*, \sigma^*, i^*, \pi^*) \leftarrow \mathcal{A}_{\mathsf{nf}}^{\mathsf{CrptU},\mathsf{SndToU},\mathsf{USK},\mathsf{WReg},\mathsf{GSig}}(\mathsf{gpk}, \mathsf{ok}, \mathsf{ak}, \mathsf{ik})$
$\mathbf{if}\ \mathsf{GVf}(\mathsf{gpk}, m^*, \sigma^*) = 0\ \mathbf{then\ return}\ 0$
$\mathbf{if}\ \mathsf{Judge}(\mathsf{gpk}, i^*, \mathsf{upk}_{i^*}, m^*, \sigma^*, \pi^*) = 0\ \mathbf{then\ return}\ 0$
$\mathbf{if}\ (i^*, m^*) \in \mathbf{QL_{gs}}\ \mathbf{then\ return}\ 0$
$\mathbf{if}\ i^* \neq u\ \mathbf{then\ return}\ 0$
$\mathbf{parse}\ \pi^* = (\sigma_{i^*}, i^*, \mathsf{pk}_{i^*}, \mathsf{cert}_{i^*}, s_{i^*}, \tau_2^*)$
$\mathbf{if}\ b = 0$
    $\mathbf{return}\ (m^*, s_{i^*})$
$\mathbf{else}\ b = 1$
    $\mathbf{return}\ (\mathsf{pk}_{i^*}, \sigma_{i^*})$
```

```
$\mathsf{GSig}(i, m)$

$\mathbf{QL_{gs}} \leftarrow \mathbf{QL_{gs}} \cup \{(i, m)\}$
$\mathbf{if}\ (b = 0)\ \&\ (i = u)\ \mathbf{then}$
    $s_i \leftarrow O_{\mathcal{DS},\mathsf{Sig}}(m)$
$\mathbf{else}$
    $s_i \leftarrow \mathsf{Sig}(\mathsf{sk}_i, m)$
$\mathsf{r_{KEM}} \xleftarrow{\$} \{0,1\}^\lambda, \mathsf{r_{IDKEM}} \xleftarrow{\$} \{0,1\}^\lambda$
$(C_{\mathsf{KEM}}, K_{\mathsf{KEM}}) \leftarrow \mathsf{Enc}(\mathsf{pk}_e; \mathsf{r_{KEM}})$
$(C_{\mathsf{IDKEM}}, K_{\mathsf{IDKEM}}) \leftarrow \mathsf{IEnc}(\mathsf{pp}, m; \mathsf{r_{IDKEM}})$
$\kappa = \langle i, \mathsf{pk}_i, \mathsf{cert}_i, s_i \rangle \odot K_{\mathsf{KEM}} \odot K_{\mathsf{IDKEM}}$
$x_1 = (\mathsf{pk}_s, \mathsf{pk}_e, \mathsf{pp}, m, C_{\mathsf{KEM}}, C_{\mathsf{IDKEM}}, \kappa)$
$w_1 = (i, \mathsf{pk}_i, \mathsf{cert}_i, s_i, \mathsf{r_{KEM}}, \mathsf{r_{IDKEM}})$
$\tau_1 \leftarrow \mathsf{P}_1(\mathsf{crs}_1, x_1, w_1)$
$\mathbf{return}\ \sigma = (C_{\mathsf{KEM}}, C_{\mathsf{IDKEM}}, \kappa, \tau_1)$
```

```
$\mathsf{CrptU}(i, \mathsf{upk})$

$\mathbf{CU} \leftarrow \mathbf{CU} \cup \{i\}$
$\mathsf{upk}_i \leftarrow \mathsf{upk}$
$\mathbf{return}\ 1$
```

```
$\mathsf{USK}(i)$

$\mathbf{if}\ i = u\ \mathbf{then\ return}\ 0$
$\mathbf{RU} \leftarrow \mathbf{RU} \cup \{i\}$
$\mathbf{return}\ (\mathsf{usk}_i, \mathsf{gsk}_i)$
```

```
$\mathsf{WReg}(i, \rho)$

$\mathbf{reg}[i] \leftarrow \rho$
$\mathbf{return}\ 1$
```

```
$\mathsf{SndToU}(i)\ (b = 0)$

$\mathbf{HU} \leftarrow \mathbf{HU} \cup \{i\},$
$\mathsf{count} \leftarrow \mathsf{count} + 1$
$(\mathsf{upk}_i, \mathsf{usk}_i) \leftarrow \mathsf{UKg}(1^\lambda, \mathsf{gpk})$
$\mathbf{if}\ \mathsf{count} = t\ \mathbf{then}$
    $u \leftarrow i, \mathsf{pk}_i \leftarrow \mathsf{pk}_s^*, \mathsf{sk}_i \leftarrow \perp$
$\mathbf{else}$
    $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{SKg}(1^\lambda)$
$\sigma_i \leftarrow \mathsf{Sig}(\mathsf{usk}_i, \mathsf{pk}_i)$
$\mathbf{return}\ (\mathsf{pk}_i, \sigma_i)$
$\mathsf{gsk}_i \leftarrow (i, \mathsf{pk}_i, \mathsf{sk}_i, \mathsf{cert}_i)$
```

```
$\mathsf{SndToU}(i)\ (b = 1)$

$\mathbf{HU} \leftarrow \mathbf{HU} \cup \{i\},$
$\mathsf{count} \leftarrow \mathsf{count} + 1$
$(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{SKg}(1^\lambda)$
$\mathbf{if}\ \mathsf{count} = t\ \mathbf{then}$
    $u \leftarrow i, \mathsf{upk}_i \leftarrow \mathsf{pk}_s^*, \mathsf{usk}_i \leftarrow \perp$
    $\sigma_i \leftarrow O_{\mathcal{DS},\mathsf{Sig}}(\mathsf{pk}_i)$
$\mathbf{else}$
    $(\mathsf{upk}_i, \mathsf{usk}_i) \leftarrow \mathsf{UKg}(1^\lambda, \mathsf{gpk})$
    $\sigma_i \leftarrow \mathsf{Sig}(\mathsf{usk}_i, \mathsf{pk}_i)$
$\mathbf{return}\ (\mathsf{pk}_i, \sigma_i)$
$\mathsf{gsk}_i \leftarrow (i, \mathsf{pk}_i, \mathsf{sk}_i, \mathsf{cert}_i)$
```

Figure 4: The EUF-CMA adversary $\mathcal{B}_{\mathsf{nf}}$ with the non-frameability adversary $\mathcal{A}_{\mathsf{nf}}$