論文 / 著書情報
Article / Book Information

題目(和文)	集約署名の研究
Title(English)	Studies on Aggregate Signature
著者(和文)	手塚真徹
Author(English)	Masayuki Tezuka
出典(和文)	学位:博士(理学), 学位授与機関:東京工業大学, 報告番号:甲第11711号, 授与年月日:2022年3月26日, 学位の種別:課程博士, 審査員:田中 圭介,伊東 利哉,尾形 わかは,鹿島 亮,安永 憲司
Citation(English)	Degree:Doctor (Science), Conferring organization: Tokyo Institute of Technology, Report number:甲第11711号, Conferred date:2022/3/26, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Studies on Aggregate Signature

Masayuki Tezuka
Supervisor: Keisuke Tanaka

Department of Mathematical and Computing Science
School of Computing
Tokyo Institute of Technology

February 1, 2022

Acknowledgements

First and foremost, I would express my great deal of gratitude to my supervisor Professor Keisuke Tanaka for support and encouragement. I appreciate that he spent a lot of time for our discussion and gave us valuable advice. I am very honored to belong to his laboratory.

I would like to thank Prof. Toshiya Ito, Prof. Wakaha Ogata, Prof. Ryo Kashima, and Prof. Kenji Yasunaga, undertaking the referee of my doctoral thesis. Based on their feedbacks, we would be more than happy to improve my work.

I would also like to thank all members and related members of Tanaka Laboratory. Especially, I would thanks to Dr. Masahiro Ishii, Dr. Yuyu Wang, Dr. Kazumasa Shinagawa, Dr. Cid Reyes Bustos, Keisuke Hara, Yusuke Yoshida, Xiangyu Su, Shinji Yoshino, Daiki Hiraga, Kyohei Sudo, and Hiroki Yamamuro. I am happy to have a fruitful time in collaboration studies with them.

At last, I would express my gratitude to my parents for their support. Also I would like to thank my beloved dogs' Pudding, Candy, and Karin.

Contents

Chapter 1

Introduction

This thesis consists of two studies which are related to aggregate signature. These studies summarized as follows.

1.1 A Study on the Synchronized Aggregate Signature Scheme

Aggregate Signature. Aggregate signature schemes originally introduced by Boneh, Gentry, Lynn, and Shacham [11] allow anyone to convert n individual signatures $(\sigma_1, \ldots, \sigma_n)$ produced by different n signers on different messages into the aggregate signature Σ whose size is much smaller than a concatenation of the individual signatures. This feature leads significant reductions of bandwidth and storage space in BGP (Border Gateway Protocol) routing [11, 40, 8], bundling software updates [1], sensor network data [1], authentication [47], and blockchain protocol [54, 29, 59]. After the introduction of aggregate signature schemes, various aggregate signature schemes have been proposed: sequential aggregate signature schemes [41], identity-based aggregate signature schemes [23], synchronized aggregate signature schemes [23, 1], and fault-tolerant aggregate signature schemes [25].

Synchronized Aggregate Signature. Synchronized aggregate signature schemes are a special type of aggregate signature schemes. The concept of the synchronized setting aggregate signature scheme was introduced by Gentry and Ramzan [23]. Ahn, Green, and Hohenberger [1] revisited the Gentry-Ramzan model and formalized the synchronized aggregate signature scheme. In this scheme, all of the signers have a synchronized time period t and each signer can sign a message at most once for each period t. A set of signatures that are all generated for the same period t can be aggregated into a short signature.

It is useful to adopt synchronized aggregate signature schemes to systems which have a natural reporting period, such as log or sensor data. As mentioned in [29], synchronized aggregate signature schemes are also useful for blockchain protocols. For instance, we consider a blockchain protocol that records several signed transactions in each new block creation. The creation of an additional block is a natural synchronization event. These signed transactions could use a synchronized aggregate signature scheme with a block number as a time period number. This reduces the signature overhead from one per transaction to just one synchronized signature per block iteration.

Several provable secure synchronized aggregate signature schemes with bilinear groups have been proposed (see Fig. 1.1). Ahn, Green, and Hohenberger [1] constructed two synchronized aggregate signature schemes based on the Hohenberger-Waters [28] short signature scheme. One is constructed in the random oracle model and the other is constructed in the standard model. The security of both schemes relies on the computational Diffie-Hellman (CDH) assumption. Lee, Lee, and Yung [38] proposed a synchronized aggregate signature scheme based on the Camenisch-Lysyanskaya signature (CL) scheme [16]. This is the efficient synchronized aggregate signature scheme with bilinear groups in that the number of pairing operations in the verification of an aggregate signature and the number of group elements in an aggregate signature is smaller than those of [23, 1]. The security of this scheme relies on the one-time Lysyanskaya-Rivest-Sahai-Wolf (OT-LRSW) assumption [42] in the random oracle model. As the provable secure synchronized aggregate signature schemes without bilinear groups, Hohenberger and Waters [29] proposed the synchronized aggregate signature scheme based on the RSA assumption.

Camenisch-Lysyanskaya Signature Scheme. Camenisch and Lysyanskaya [16] proposed the CL scheme which has a useful feature called randomizability. This property allows anyone to randomize a valid signature σ to σ' where σ and σ' are valid signatures on the same message. The CL scheme is widely used to construct various schemes: anonymous credentials [16], anonymous attestation [5], divisible E-cash [17], batch verification [15], group signatures [6], ring signatures [4], and aggregate signatures [55]. The security of the CL scheme relies on the Lysyanskaya-Rivest-Sahai-Wolf (LRSW) assumption which is an interactive assumption. An interactive assumption allows us to design an efficient scheme, however, these are not preferable.

Scheme	Assumption	Security	pp size	vk size	Agg size	Agg Ver (in Pairings)
GR [23]	CDH + ROM	EUF-CMA*	$O(1)$	ID	3	3
AGH [1] §4	CDH	EUF-CMA in CK	$O(k)$	1	3	$k+3$
AGH [1] §A	CDH + ROM	EUF-CMA in CK	$O(1)$	1	3	4
LLY [38]	OT-LRSW + ROM (interactive assumption)	EUF-CMA in CK	$O(1)$	1	2	3
LLY [38] (New proof)	1-MSDH-2 + ROM (static assumption)	EUF-CMA in CK	$O(1)$	1	2	3

In our work, we prove that the scheme LLY [38] satisfies the EUF-CMA security in the certified-key model under the 1-MSDH-2 assumption in the random oracle model.

Figure 1.1: Summary of synchronized aggregate signature schemes with bilinear groups. In the column of "Assumption", "ROM" means the random oracle model. In the column of "Security", "CK" means the certified-key model. "pp size", "vk size", "Agg size", "Agg Ver" mean the number of group elements in a public parameter pp, a verification key vk, an aggregate signature, and the number of pairing operations in aggregate signatures verification respectively. The scheme GR [23] is an identity-based scheme that has a verification key size of "ID". In the scheme AGH [1], k is a special security parameter. As mentioned in [1], k could be five in practice. * Note that Gentry and Ramzan [23] only provided heuristic security arguments.

Modified Camenisch-Lysyanskaya Signature Scheme. Pointcheval and Sanders [50] proposed the Modified q-Strong Diffie-Hellman-2 (q-MSDH-2) assumption which is defined on a type 1 bilinear group. This assumption is a q-type assumption [9] where the number of input elements depends on the number of adversarial queries. They proved that the q-MSDH-2 assumption holds in the generic bilinear group model [10] and the CL scheme satisfies the weak-existentially unforgeable under chosen message attacks (weak-EUF-CMA) security under the q-MSDH-2 assumption. Moreover, they proposed the modified Camenisch-Lysyanskaya signature (MCL) scheme which has randomizability. Then, they showed that the MCL scheme satisfies the existentially unforgeable under chosen message attacks (EUF-CMA) security under the q-MSDH-2 assumption. Their modification from the CL scheme to the MCL scheme incurs a slight increase in the complexity.*

Our Results. To our knowledge, the most efficient synchronized aggregate signature scheme with bilinear groups is Lee et al.'s [38] scheme. However, the security of this scheme relies on the interactive assumption (the OT-LRSW assumption). Even if interactive assumptions hold in the generic group model or bilinear group model, the concerns about these assumptions arise in a cryptographic community. This fact causes a barrier to the use of this scheme. Also, it is not desired that the security of the scheme depends on q-type assumptions. Because the size of these assumptions grows dynamically and this fact leads to inefficiency of the scheme. Hence, it is desirable to prove the security of this scheme under the non-q-type (static) assumptions or construct another efficient synchronized aggregate signature scheme whose security does not rely on interactive assumptions or q-type assumptions.

Security Proof under the Static Assumption. In this paper, we give a new security proof for Lee et al.'s synchronized aggregate scheme under the static assumption in the random oracle model. More specifically, we convert from the MCL scheme to Lee et al.'s [38] synchronized aggregate signature scheme. Then, we reduce the security of Lee et al.'s scheme to the one-time EUF-CMA (OT-EUF-CMA) security of the MCL scheme in the random oracle model. We refer the reader to Section 3.5 for details about these techniques. Since the OT-EUF-CMA security of the MCL scheme is implied by the 1-MSDH-2 assumption, the security of Lee et al.'s scheme can be proven under the 1-MSDH-2 assumption. We can regard the 1-MSDH-2 assumption as the static assumption. Therefore, we can see that the security of Lee et al.'s scheme relies on the static assumption. Notably, while the EUF-CMA security of the MCL scheme is proved under the q-type assumption, the security of Lee et al.'s synchronized aggregate signature scheme can be proven under the static assumption in the random oracle model.

In general, there is a trade-off that efficiency is reduced when we design a scheme based on weaker computational assumptions. Surprisingly, we can change the assumptions underlying the security of Lee et al.'s [38] scheme from the interactive assumption (OT-LRSW) to the static assumption (1-MSDH-2) with almost the same reduction loss. Specifically, the size of verification key vk, the size of aggregate signature Σ, and the number of pairing operations in an aggregate signature verification do not increase at all.

*Their modification from the CL scheme to the MCL scheme increases the number of group elements in a signature and an aggregate signature from 2 to 3.

Related Works Boneh et al. [11] proposed the first full aggregate signature scheme which allows any user to aggregate signatures of different signers. Furthermore, this scheme allows us to aggregate individual signatures as well as already aggregated signatures in any order. They constructed a full aggregate signature scheme in the random oracle model. Hohenberger, Sahai, and Waters [27] firstly constructed a full aggregate signature scheme in the standard model by using multilinear maps. Hohenberger, Koppula, and Waters [26] constructed a full aggregate signature scheme in the standard model by using the indistinguishability obfuscation.

Several variants of aggregate signature schemes have been proposed. One major variant is a sequential aggregate signature scheme which was firstly proposed by Lysyanskaya, Micali, Reyzin, and Shacham [41]. In this scheme, an aggregate signature is constructed sequentially, with each signer modifying the aggregate signature in turn. They constructed a sequential aggregate signature scheme in the random oracle model by using families of trapdoor permutations. Lu, Rafail Ostrovsky, Sahai, Shacham, and Waters [40] firstly constructed the sequential aggregate signature scheme in the standard model based on the Waters signature scheme.

Furthermore, Lee et al. [38] proposed a combined aggregate signature scheme. In this scheme, a signer can use two modes of aggregation (sequential aggregation or synchronized aggregation) dynamically. They constructed a combined aggregate signature scheme in the random oracle model based on the CL scheme.

1.2 A Study on the T-out-of-N Redactable Signature Scheme

Redactable Signature. Recently, due to the development of IoT devices, the number of electronic data is steadily increasing. It is indispensable for future information society to make use of these data. When we use data, it is important to prove that the data has not been modified in any way. A digital signature enables a verifier to verify the authenticity of M by checking that σ is a legitimate signature on M. However, in our real-world scenario, when we use data, the confidential information should be deleted from the original data. A digital signature cannot verify the validity of a message with parts of the message removed.

A redactable signature scheme (RSS) is a useful cryptographic scheme for such a situation. This scheme consists of a signer, a redactor, and a verifier. A signer signs a message M with a secret key sk and generates a valid signature σ. A redactor who can become anyone removes some parts of a signed message from M, generate a redacted message M', and updates the corresponding signature σ' without the secret key sk. A verifier still verifies the validity of the signature σ' on message M' using vk.

An idea of a redactable signature scheme was introduced by Steinfeld, Bull, and Zheng [57] as a content extraction signature scheme (CES). This scheme allows generating an extracted signature on selected portions of the signed original document while hiding removed parts of portions. Johnson, Molnar, Song, and Wagner [35] proposed a redactable signature scheme (RSS) which is similar to a content extraction signature scheme.

Security of Redactable Signature. Security of a redactable signature scheme was argued in many works. Brzuska, Busch, Dagdelen, Fischlin, Franz, Katzenbeisser, Manulis, Onete, Peter, Poettering, and Schröder [13] formalized three security notions of a redactable signature for tree-structured messages in the game-based definition.

- Unforgeability: Without the secret key sk it is hard to generate a valid signature σ' on a message M' except to redact a signed message (M, σ).
- Privacy: Except for a signer and redactors, it is hard to derive any information about removed parts of the original message M from the redacted message M'.
- Transparency: It is hard to distinguish whether (M, σ) directly comes from the signer or has been processed by a redactor.

Derler, Pöhls, Samelin, and Slamanig [21] gave a general framework of a redactable signature scheme for arbitrary data structures and defined its security.

Camenisch, Dubovitskaya, Haralambiev, and Kohlweiss [14] proposed unlinkable redactable signature. This signature satisfies unforgeability and unlinkability which is a variant security notion of privacy. They used an unlinkable redactable signature scheme to construct an anonymous credential scheme [18]. Later, Sanders [52] constructed an unlinkable redactable signature scheme to obtain an efficient anonymous credential scheme. Moreover, Sanders [53] constructed a revokable group signature scheme based on an unlinkable redactable signature scheme.

Additional Functionalities. Following additional functionalities for a redactable signature scheme were proposed.

- Disclosure Control [46, 44, 45, 24, 32, 33, 30, 51, 43]: Miyazaki, Iwamura, Matsumoto, Sasaki, Yoshiura, Tezuka, and Imai [46] proposed the disclosure control. The signer or intermediate redactors can control to prohibit further redactions for parts of the message.

- Identification of a Redactor [31, 34]: Izu, Kanaya, Takenaka, and Yoshioka [31] proposed the redactable signature scheme called "Partial Information Assuring Technology for Signature" (PIATS). PIATS allows a verifier to identify the redactor of the signed message.

- Accountability [49]: Pöhls and Samelin proposed an accountable redactable signature scheme that allows deriving the accountable party of a signed message.

- Update and Marge [39, 48]: Lim, Lee, and Park [39] proposed the redactable signature scheme where a signer can update signature by adding new parts of a message. Moreover, Pöhls and Samelin [48] proposed the updatable redactable signature scheme that can update a signature and marge signatures derived from the same signer.

- Compactness [58]: Most redactable signature schemes, to remove parts of the signed message, we need pieces of information for each part we want to remove. That is, if a signed message consists of l elements, the number of elements in an original signature is at least linear in l. Tezuka and Tanaka [58] introduce compactness for redactable signature schemes. Compactness requires that the size of an original signature and signature for a subdocument (redacted message) are indipendent regardless of the number of elements in messages.

Motivation. Consider the case where a citizen requests the signed secret document disclosure to the government. To disclose the secret signed document, the government must remove sensitive data from it. A decision of deletion for confidential information of a document is performed by multiple officers in the government meeting.

One of the simple solutions is that the signer of the secret document gives the signing key sk to the meeting chair. The chair takes a vote on removing sensitive information and removes it from the secret document and signed it using sk. However, if the meeting chair is malicious, it is risky for the secret document signer to give the meeting chair a signing key sk. Therefore, the secret document signer wants to avoid giving a signing key sk to others.

If we try to adapt the original RSS on this situation, we suffer from the following problem. RSS allows anyone to redact message parts and even removes the necessary information.

Moreover, a malicious chair can redact message parts form the signed document regardless of the decision of the officers.

Our Contributions. We introduce the new notion of t-out-of-n redactable signature scheme to overcome this problem. This scheme is composed of a signer, n redactors, a combiner, and a verifier. The signer designates n redactors and a combiner, generates a key pair $(\mathsf{vk}, \mathsf{sk})$ and redactor's secret key $\{\mathsf{rk}[i]\}_{i=1}^{n}$ and sends $\mathsf{rk}[i]$ to the redactor i. Then signer decides parts of a message that redaction is allowed, signs the message, and sends its signature to n redactor and a combiner. Each redactor i selects parts of the signed message that he or she wants to remove, generates a piece of redaction information RI_i, and sends it to the combiner. The combiner collects all redaction information $\{\mathsf{RI}_i\}_{i=1}^{n}$, extracts signed message parts which at least t redactors want to remove using $\{\mathsf{RI}_i\}_{i=1}^{n}$, generates the redactable signature. The verifier can verify the validities of signatures.

Now, we reconsider applying the t-out-of-n redactable signature scheme to the above redaction problem. Let the secret document signer be a signer of the t-out-of-n redactable signature scheme, officers be redactors, and the meeting chair be a combiner. The secret document signer does not have to give the signing key sk to the chair. Our t-out-of-n redactable signature only allows the chair to redact parts of message which at least t officers wants to remove.

We consider the one-time redaction model which allows redacting signed message only one time for each signature and gives the unforgeability, privacy, and transparency security of the t-out-of-n redactable signature scheme in the one-time redaction model. Also, we give a concrete construction of the t-out-of-n redactable signature scheme which satisfies the unforgeability, privacy, and transparency security.

Our construction is based on the (t, n)-Shamir's secret sharing scheme and the redactable signature scheme proposed by Miyazaki, Hanaoka, and Imai [44] which use the aggregate signature scheme proposed by Boneh, Gentry, Lynn, and Shacham [11] based on the BLS signature scheme [12]. Our technical point is to adapt (t, n)-Shamir's secret share scheme and compute Lagrangian interpolation at the exponent part of the group element to reconstruct information for the redaction. Security of our scheme is based on the computational co-CDH assumption in the random oracle model.

Related Works. We present several signatures that allow editing signed message. More details of the overview of related works, see [20, 7].

- Append-Only Signature [36]: Kiltz, Mityagin, Panjwani, and Raghavan [36] introduce the notion of the append-only signature scheme. In this scheme, we can only publicly append message blocks to a signed message and update the signature correspondingly.

- Sanitizable Signature [2]: Ateniese, Chou, de Medeiros, and Tsudik [2] introduce the notion of the sanitizable signature scheme. In this scheme, a signer selects a sanitizer who can modify the signed message and generate a signature. The sanitizer can modify some parts of message blocks of the signed document, but he or she cannot remove message blocks. In the redactable signature, anyone can redact parts of the signed message without the secret key. However, in the sanitizable signature scheme, each sanitizer has the sanitizer's secret key and the only the sanitizer designated by signer can sanitize parts of the message using own sanitizer's secret key.

- Protean Signature [37]: Krenn, Pöhls, Samelin, and Slamanig [37] introduce the notion of the protean signature scheme. This scheme allows removing and editing some parts of message blocks. They give the construction of the protean signature scheme from a sanitizable signature scheme and a redactable signature scheme in the black box way.

1.3 Road Map

We describe the road map of this thesis. In Chapter 2, we introduce notations and review bilinear groups, digital signature and the random oracle model (ROM). These primitives play an important role in both our studies.

In Chapter 3, we study for the Camenish-Lysyanskaya signature based synchronized aggregate signature scheme. At first, we review the definition of synchronized aggregate signature scheme and its security. Second, we review the modified Camenisch-Lysyanskaya (MCL) signature scheme [50] whose security is used to prove the synchronized aggregate signature by Lee et al. [38]. Then, we review the synchronized aggregate signature by Lee et al. [38] and provide the high-level idea of our new conversion technique from a MCL signature to a synchronized aggregate signature by Lee et al. Finally, we provide the improved security proof for the aggregate signature by Lee et al. by using our conversion technique.

In Chapter 4, we study for the extension of redactable signature. At first, we propose a t-out-of-n redactable signature signature scheme and its security notions. Second, we review the BGLS aggregate signature scheme and the Shamir's secret sharing scheme. Then, based on these primitives, we propose our t-out-of-n redactable signature scheme construction. Finally, we prove security for our construction of t-out-of-n redactable signature scheme.

In Chapter 5, we summarize our result and describe some open problems.

Chapter 2

Preliminaries

2.1 Notations

Let 1^λ be the security parameter. A function $f(\lambda)$ is negligible in λ if $f(\lambda)$ tends to 0 faster than $\frac{1}{\lambda^c}$ for every constant $c > 0$. PPT stands for probabilistic polynomial time. For an integer n, $[n]$ denotes the set $\{1, \ldots, n\}$. For a finite set S, $s \xleftarrow{\$} S$ denotes choosing an element s from S uniformly at random. For a group $\mathbb{G}$, we define $\mathbb{G}^* := \mathbb{G} \backslash \{1_{\mathbb{G}}\}$. For an algorithm A, $y \leftarrow \mathsf{A}(x)$ denotes that the algorithm A outputs y on input x.

2.2 Bilinear Groups

We introduce a bilinear group generator. Let G be a bilinear group generator that takes as an input a security parameter 1^λ and outputs the descriptions of multiplicative groups $\mathcal{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ where $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are groups of prime order p and e is an efficient computable, non-degenerating bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$.

1. Bilinear: for all $u \in \mathbb{G}_1$, $v \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}_p$, then $e(u^a, v^b) = e(u, v)^{ab}$.

2. Non-degenerate: for any $g_1 \in \mathbb{G}^*$ and $g_2 \in \mathbb{G}_2^*$, $e(g_1, g_2) \neq 1_{\mathbb{G}_T}$.

Bilinear groups are classified into following three types [22]: Type 1 pairings: $\mathbb{G}_1 = \mathbb{G}_2$; Type 2 pairings: $\mathbb{G}_1 \neq \mathbb{G}_2$ but there exists an efficient homomorphism $\psi : \mathbb{G}_2 \to \mathbb{G}_1$; Type 3 pairings: $\mathbb{G}_1 \neq \mathbb{G}_2$ and there are no efficiently computable homomorphisms between $\mathbb{G}_1$ and $\mathbb{G}_2$.

2.3 Computational Assumptions

Boneh, Gentry, Lynn, and Shacham [11] introduced the computational co-Diffie-Hellman assumption (co-CDH) which is the variant of the computational Diffie-Hellman (CDH) assumption. They used this assumption to prove the security of their aggregate signature scheme.*

Assumption 2.1 (Computational co-Diffie-Hellman Assumption [11]). Let G be a type-2 pairing-group generator. The computational co-Diffie-Hlleman (co-CDH) assumption over G is that for all $\lambda \in \mathbb{N}$, for all $\mathcal{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathsf{G}(1^\lambda)$, given $(\mathcal{G}, g_1, g_1^\alpha, h)$ where $g_1, h \leftarrow \mathbb{G}_2$ and $\alpha \xleftarrow{\$} \mathbb{Z}_p$ as an input, no PPT adversary can, without non-negligible probability, outputs h^α. We write the advantage of co-Diffie-Hlleman assumption for A as

$$\mathsf{Adv}^{\text{co-CDH}}_{\mathcal{G},\mathsf{A}} = \Pr\left[\mathsf{A}(g_1, g_1^\alpha, h) = h^\alpha \middle| (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathsf{G}(1^\lambda), \alpha \xleftarrow{\$} \mathbb{Z}_q, g_1 \xleftarrow{\$} \mathbb{G}_1, h \xleftarrow{\$} \mathbb{G}_2\right].$$

Pointcheval and Sanders [50] introduced the new q-type assumption which is called the Modified q-Strong Diffie-Hellman-2 (q-MSDH-2) assumption. This is a variant of the q-Strong Diffie-Hellman (q-SDH) assumption and defined on a type 1 bilinear group. By using this assumption, the weak EUF-CMA security for CL signature scheme and EUF-CMA security The q-MSDH-2 assumption holds in the generic bilinear group model [10]. In this work, we fix the value to $q = 1$ and only use 1-MSDH-2 assumption in a static way. We can regard 1-MSDH-2 as a static assumption.

Assumption 2.2 (Modified 1-Strong Diffie-Hellman-2 Assumption [50]). Let G be a type-1 pairing-group generator. The Modified 1-Strong Diffie-Hellman-2 (1-MSDH-2) assumption over G is that for all $\lambda \in \mathbb{N}$, for all $\mathcal{G} = (p, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathsf{G}(1^\lambda)$, given $(\mathcal{G}, g, g^x, g^{x^2}, g^b, g^{bx}, g^{bx^2}, g^a, g^{abx})$ where $g \leftarrow \mathbb{G}^*$ and $a, b, x \xleftarrow{\$} \mathbb{Z}_p^*$ as an input, no PPT adversary can, without non-negligible probability, output a tuple $(w, P, h^{\frac{1}{x+w}}, h^{\frac{a}{x \cdot P(x)}})$ with $h \in \mathbb{G}$, P a polynomial in $\mathbb{Z}_p[X]$ of degree at most 1, and $w \in \mathbb{Z}_p^*$ such that $X + w$ and $P(X)$ are relatively prime.†

2.4 Digital Signature

Definition 2.3 (Digital Signature Scheme). A digital signature scheme DS is composed of following four algorithms ($\mathsf{DS.Setup}, \mathsf{DS.KeyGen}, \mathsf{DS.Sign}, \mathsf{DS.Verify}$).

*In the case of $\mathbb{G}_1 = G_2$, the co-CDH assumption reduces to the CDH assumption.

†In the q-MSDH-2 assumption, an input is changed to $(\mathcal{G}, g, g^x, \ldots, g^{x^{q+1}}, g^b, g^{bx}, \ldots, g^{bx^{q+1}}, g^a, g^{abx})$ and the condition of the order of $P(x)$ is changed to at most q.

- $\mathsf{DS.Setup}(1^\lambda)$: Given a security parameter λ, return the public parameter pp. We assume that pp defines the message space $\mathcal{M}_{pp}$.

- $\mathsf{DS.KeyGen}(pp)$: Given a public parameter pp, return a verification key vk and a signing key sk.

- $\mathsf{DS.Sign}(pp, \mathsf{sk}, m)$: Given a public parameter pp, a signing key sk, and a message $m \in \mathcal{M}_{pp}$, return the signature σ.

- $\mathsf{DS.Verify}(pp, \mathsf{vk}, m, \sigma)$: Given a public parameter pp, a verification key vk, a message $m \in \mathcal{M}_{pp}$, and a signature σ, return either 1 (Accept) or 0 (Reject).

For DS, we require the following correctness.

- Correctness: A digital signature scheme DS is correct if for all $\lambda \in \mathbb{N}$, $pp \leftarrow \mathsf{DS.Setup}(1^\lambda)$, for all $m \in \mathcal{M}_{pp}$, $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{DS.KeyGen}(pp)$, $\sigma \leftarrow \mathsf{DS.Sign}(pp, \mathsf{sk}, m)$, then $\mathsf{DS.Verify}(pp, \mathsf{vk}, m, \sigma) = 1$ holds.

Definition 2.4 (EUF-CMA)**.** Existentially unforgeable under chosen-message attacks (EUF-CMA) security for a digital signature scheme DS is defined by the following unforgeability game between a challenger and an adversary A.

- The challenger computes $pp \leftarrow \mathsf{DS.Setup}(1^\lambda)$, $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{DS.KeyGen}(pp)$ initializes $Q \leftarrow \{\}$, and sends (pp, vk) to A.

- A is given access to a signing oracle $\mathcal{O}^{\mathsf{Sign}}(\cdot)$. Given an input m, $\mathcal{O}^{\mathsf{Sign}}$ computes $\sigma \leftarrow \mathsf{DS.Sign}(pp, \mathsf{sk}, m)$, update $Q \leftarrow Q \cup \{m\}$ and returns σ to A.

- Finally, A outputs a forgery (m^*, σ^*).

DS is EUF-CMA secure if for all $\lambda \in \mathbb{N}$ and all PPT adversaries A, the advantage $\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{DS,A}} := \Pr\left[\mathsf{DS.Verify}(pp, \mathsf{vk}, m^*, \sigma^*) = 1 \land m^* \notin Q\right]$ is negligible in λ.

If the number of signing oracle $\mathcal{O}^{\mathsf{Sign}}$ query is restricted to the one-time in the unforgeability security game, we call DS satisfies the one-time EUF-CMA (OT-EUF-CMA) security.

2.5 Random Oracle Model

In this thesis, security of signature schemes is proved in the random oracle model (ROM) [3]. In this model, a hash function is regarded as an ideal random function. Instead of computing a hash value, all the parties can obtain hash values by querying the random oracle with an input.

In a security reduction in the ROM, the reduction simulates security the random oracle in the security game. The reduction can set(program) and output hash values as long as the distribution of output value is indistinguishable from a uniform distribution. This property is called programmability and widely used in security reductions. Compared to the standard model, it allows us to construct efficient signature schemes with the provable security.

Chapter 3

Synchronized Aggregate Signature

In this cahpter, at first, we review the definition of a synchronized aggregate signature scheme and its security model, review the $\mathsf{DS}_{\mathsf{MCL}}$ scheme proposed by Pointcheval and Sanders [50]. This is used for security proof for the synchronized aggregate signature scheme proposed by Lee et al. Then, we describe Lee et al.'s aggregate signature scheme construction. Finally, we give a new security proof for Lee et al.'s scheme under the 1-MSDH-2 assumption in the random oracle model.

3.1 Syntax

Synchronized aggregate signature schemes [23, 1] are a special type of aggregate signature schemes. In this scheme, all of the signers have a synchronized time period t and each signer can sign a message at most once for each period t. A set of signatures that are all generated for the same period t can be aggregated into a short signature. The size of an aggregate signature is the same size as an individual signature. Now, we review the definition of synchronized aggregate signature schemes.

Definition 3.1 (Synchronized Aggregate Signature Schemes [23, 1])**.** A synchronized aggregate signature scheme SAS for a bounded number of periods is a tuple of algorithms ($\mathsf{SAS.Setup}$, $\mathsf{SAS.KeyGen}$, $\mathsf{SAS.Sign}$, $\mathsf{SAS.Verify}$, $\mathsf{SAS.Aggregate}$, $\mathsf{SAS.AggVerify}$).

- $\mathsf{SAS.Setup}(1^\lambda, 1^T)$: Given a security parameter λ and the time period bound T, return the public parameter pp. We assume that pp defines the message space $\mathcal{M}_{pp}$.

- $\mathsf{SAS.KeyGen}(pp)$: Given a public parameter pp, return a verification key vk and a signing key sk.

- $\mathsf{SAS.Sign}(pp, \mathsf{sk}, t, m)$: Given a public parameter pp, a signing key sk, a time period $t \leq T$, and a message $m \in \mathcal{M}_{pp}$, return the signature σ.

- $\mathsf{SAS.Verify}(pp, \mathsf{vk}, m, \sigma)$: Given a public parameter pp, a verification key vk, a message $m \in \mathcal{M}_{pp}$, and a signature σ, return either 1 (Accept) or 0 (Reject).

- $\mathsf{SAS.Aggregate}(pp, (\mathsf{vk}_1, \ldots, \mathsf{vk}_r), (m_1, \ldots, m_r), (\sigma_1, \ldots, \sigma_r))$: Given a public parameter pp, a list of verification keys $(\mathsf{vk}_1, \ldots, \mathsf{vk}_r)$, a list of messages $(m_1, \ldots, m_r)$, and a list of signatures $(\sigma_1, \ldots, \sigma_r)$, return either the aggregate signature Σ or $\perp$.

- $\mathsf{SAS.AggVerify}(pp, (\mathsf{vk}_1, \ldots, \mathsf{vk}_r), (m_1, \ldots, m_r), \Sigma)$: Given a public parameter pp, a list of verification keys $(\mathsf{vk}_1, \ldots, \mathsf{vk}_r)$, a list of messages $(m_1, \ldots, m_r)$, and an aggregate signature, return either 1 (Accept) or 0 (Reject).

Correctness: Correctness is satisfied if for all $\lambda \in \mathbb{N}$, $T \in \mathbb{N}$, $pp \leftarrow \mathsf{SAS.Setup}(1^\lambda, 1^T)$, for any finite sequence of key pairs $(\mathsf{vk}_1, \mathsf{sk}_1), \ldots (\mathsf{vk}_r, \mathsf{sk}_r) \leftarrow \mathsf{SAS.KeyGen}(pp)$ where vk_i are all distinct, for any time period $t \leq T$, for any sequence of messages $(m_1, \ldots m_r) \in \mathcal{M}_{pp}$, $\sigma_i \leftarrow \mathsf{SAS.Sign}(pp, \mathsf{sk}_i, t, m_i)$ for $i \in [r]$, $\Sigma \leftarrow \mathsf{SAS.Aggregate}(pp, (\mathsf{vk}_1, \ldots, \mathsf{vk}_r), (m_1, \ldots, m_r), (\sigma_1, \ldots, \sigma_r))$, we have

$$\mathsf{SAS.Verify}(pp, \mathsf{vk}_i, m_i, \sigma_i) = 1 \text{ for all } i \in [r] \\ \wedge\, \mathsf{SAS.AggVerify}(pp, (\mathsf{vk}_1, \ldots, \mathsf{vk}_r), (m_1, \ldots, m_r), \Sigma) = 1.$$

In a signature aggregation, it is desirable to confirm that each signature is valid. This is because if there is at least one invalid signature, the generated aggregate signature will be invalid.* In this work, before aggregating signatures, $\mathsf{SAS.Aggregate}$ checks the validity of each signature.

3.2 Security

We introduce the security notion of synchronized aggregate signature schemes. The EUF-CMA security of synchronized aggregate signature schemes proposed by Gentry and Ramzan [23] captures that it is hard for adversaries to forge an aggregate signature without signing key

*Fault-tolerant aggregate signature schemes [25] allow us to determine the subset of all messages belonging to an aggregate signature that were signed correctly. However, this scheme has a drawback that the aggregate signature size depends on the number of signatures to be aggregated into it.

sk^*. However, they only provided heuristic security arguments in their synchronized aggregate signature scheme.

Ahn, Green, and Hohrnberger [1] introduced the certified-key model for the EUF-CMA security of synchronized aggregate signature schemes. In this model, signers must certify their verification key vk by proving knowledge of their signing key sk. In other words, no verification key vk is allowed except those correctly generated by the $\mathsf{SAS.KeyGen}$ algorithm. In certified-key model, to ensure the correct generation of a verification key $\mathsf{vk}_i \neq \mathsf{vk}^*$, EUF-CMA adversaries must submit $(\mathsf{vk}_i, \mathsf{sk}_i)$ to the certification oracle $\mathcal{O}^{\mathsf{Cert}}$. As in [1, 38], we consider the EUF-CMA security in the certified-key model.

Definition 3.2 (EUF-CMA Security in the Certified-Key Model [1, 38])**.** The EUF-CMA security of a sequential aggregate signature scheme SAS in the certified-key model is defined by the following unforgeability game between a challenger C and a PPT adversary A.

- C runs $pp^* \leftarrow \mathsf{SAS.Setup}(1^\lambda, 1^T)$, $(\mathsf{vk}^*, \mathsf{sk}^*) \leftarrow \mathsf{SAS.KeyGen}(pp^*)$, sets $Q \leftarrow \{\}$, $L \leftarrow \{\}$, $t_{ctr} \leftarrow 1$, and gives (pp, vk^*) to A.
- A is given access (throughout the entire game) to a certification oracle $\mathcal{O}^{\mathsf{Cert}}(\cdot,\cdot)$. Given an input $(\mathsf{vk}, \mathsf{sk})$, $\mathcal{O}^{\mathsf{Cert}}$ performs the following procedure.
  - If the key pair $(\mathsf{vk}, \mathsf{sk})$ is valid, $L \leftarrow L \cup \{\mathsf{vk}\}$ and return "`accept`".
  - Otherwise return "`reject`".

  (A must submit key pair $(\mathsf{vk}, \mathsf{sk})$ to $\mathcal{O}^{\mathsf{Cert}}$ and get "`accept`" before using vk.)
- A is given access (throughout the entire game) to a sign oracle $\mathcal{O}^{\mathsf{Sign}}(\cdot,\cdot)$. Given an input ("`inst`", m), $\mathcal{O}^{\mathsf{Sign}}$ performs the following procedure.
  ("`inst`" $\in$ {"`skip`", "`sign`"} represent the instruction for $\mathcal{O}^{\mathsf{Sign}}$ where "`skip`" implies that A skips the concurrent period t_{ctr} and "`sign`" implies that A require the signature on message m.)
  - If $t_{ctr} \notin [T]$, return $\perp$.
  - If "`inst`" = "`skip`", $t_{ctr} \leftarrow t_{ctr} + 1$.
  - If "`inst`" = "`sign`", $Q \leftarrow Q \cup \{m\}$, $\sigma \leftarrow \mathsf{SAS.Sign}(pp^*, \mathsf{sk}^*, t, m)$, $t_{ctr} \leftarrow t_{ctr} + 1$, return σ.
- A outputs a forgery $((\mathsf{vk}_1^*, \ldots, \mathsf{vk}_{r^*}^*), (m_1^*, \ldots, m_{r^*}^*), \Sigma^*)$.

A sequential aggregate signature scheme SAS satisfies the EUF-CMA security in the certified-key model if for all PPT adversaries A, the following advantage

$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{SAS,A}} := \Pr\left[\begin{array}{l}\mathsf{SAS.AggVerify}(pp^*, (\mathsf{vk}^*_1, \ldots, \mathsf{vk}^*_{r^*}), (m^*_1, \ldots, m^*_{r^*}), \Sigma^*) = 1\\ \wedge \text{ For all } j \in [r^*] \text{ such that } \mathsf{vk}^*_j \neq \mathsf{vk}^*, \mathsf{vk}^*_j \in L\\ \wedge \text{ For some } j^* \in [r^*] \text{ such that } \mathsf{vk}^*_{j^*} = \mathsf{vk}^*, m^*_{j^*} \notin Q\end{array}\right]$$

is negligible in λ.

3.3 Modified Camenisch-Lysyanskaya Signature Scheme

Pointcheval and Sanders [50] proposed the modified Camenisch-Lysyanskaya signature scheme which supports a multi-message (vector message) signing. In this work, we only need a single-message signing scheme. Here, we review the single-message modified Camenisch-Lysyanskaya signature scheme $\mathsf{DS_{MCL}} = (\mathsf{DS_{MCL}.Setup}, \mathsf{DS_{MCL}.KeyGen}, \mathsf{DS_{MCL}.Sign}, \mathsf{DS_{MCL}.Verify})$ as follows.

- $\mathsf{DS_{MCL}.Setup}(1^\lambda)$:
  $\mathcal{G} = (p, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathsf{G}(1^\lambda)$. ($\mathsf{G}$ is a type-1 pairing-group generator)
  Return $pp \leftarrow \mathcal{G}$.

- $\mathsf{DS_{MCL}.KeyGen}(pp)$:
  $g \xleftarrow{\$} \mathbb{G}^*$, $x \xleftarrow{\$} \mathbb{Z}_p^*$, $y \xleftarrow{\$} \mathbb{Z}_p^*$, $z \xleftarrow{\$} \mathbb{Z}_p^*$, $X \leftarrow g^x$, $Y \leftarrow g^y$, $Z \leftarrow g^z$.
  Return $(\mathsf{vk}, \mathsf{sk}) \leftarrow ((g, X, Y, Z), (x, y, z))$.

- $\mathsf{DS_{MCL}.Sign}(pp, \mathsf{sk}, m)$:
  Parse sk as (x, y, z)
  $w \xleftarrow{\$} \mathbb{Z}_p$, $A \xleftarrow{\$} \mathbb{G}^*$, $B \leftarrow A^y$, $C \leftarrow A^z$, $D \leftarrow C^y$, $E \leftarrow A^x B^{mx} D^{wx}$.
  Return $\sigma \leftarrow (w, A, B, C, D, E)$.

- $\mathsf{DS_{MCL}.Verify}(pp, \mathsf{vk}, m, \sigma)$:
  Parse vk as (g, X, Y, Z), σ as (w, A, B, C, D, E).
  If $(e(A, Y) \neq e(B, g)) \vee (e(A, Z) \neq e(C, g)) \vee (e(C, Y) \neq e(D, g))$, return 0.
  If $e(AB^m D^w, \tilde{X}) = e(E, g)$, return 1.
  Otherwise return 0.

Pointcheval and Sanders [50] proved that if the q-MSDH-2 assumption holds, then the $\mathsf{DS}_{\mathsf{MCL}}$ scheme satisfies the EUF-CMA security where q is a bound on the number of adaptive signing queries. In this work, we only need the OT-EUF-CMA security for the $\mathsf{DS}_{\mathsf{MCL}}$ scheme.

Theorem 3.3 ([50]). If the 1-MSDH-2 assumption holds, then the $\mathsf{DS}_{\mathsf{MCL}}$ scheme satisfies the OT-EUF-CMA security.

3.4 Synchronized Aggregate Signature Scheme by Lee et al.

We describe the synchronized aggregate signature scheme by Lee et al. [38]. Let T be a bounded number of periods which is a polynomial in λ. The synchronized aggregate signature scheme by Lee et al. $\mathsf{SAS}_{\mathsf{LLY}} = (\mathsf{SAS}_{\mathsf{LLY}}.\mathsf{Setup}, \mathsf{SAS}_{\mathsf{LLY}}.\mathsf{KeyGen}, \mathsf{SAS}_{\mathsf{LLY}}.\mathsf{Sign}, \mathsf{SAS}_{\mathsf{LLY}}.\mathsf{Verify}, \mathsf{SAS}_{\mathsf{LLY}}.\mathsf{Aggregate}, \mathsf{SAS}_{\mathsf{LLY}}.\mathsf{AggVerify})$ [38] is given as follows.[†]

- $\mathsf{SAS}_{\mathsf{LLY}}.\mathsf{Setup}(1^\lambda, 1^T)$:
  1. $\mathcal{G} = (p, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathsf{G}(1^\lambda)$, $g \xleftarrow{\$} \mathbb{G}^*$. ($\mathsf{G}$ is a type-1 pairing-group generator)
  2. Choose hash functions:
     $H_1 : [T] \to \mathbb{G}$, $H_2 : [T] \to \mathbb{G}^*$, $H_3 : [T] \times \{0,1\}^* \to \mathbb{Z}_p$.
  3. Return $pp \leftarrow (\mathcal{G}, g, H_1, H_2, H_3)$.
- $\mathsf{SAS}_{\mathsf{LLY}}.\mathsf{KeyGen}(pp)$:
  1. $x \xleftarrow{\$} \mathbb{Z}_p^*$, $X \leftarrow g^x$.
  2. Return $(\mathsf{vk}, \mathsf{sk}) \leftarrow (X, x)$.
- $\mathsf{SAS}_{\mathsf{LLY}}.\mathsf{Sign}(pp, \mathsf{sk}, t, m)$:
  1. $m' \leftarrow H_3(t, m)$, $E \leftarrow H_1(t)^{\mathsf{sk}} H_2(t)^{m'\mathsf{sk}}$.
  2. Return (E, t).
- $\mathsf{SAS}_{\mathsf{LLY}}.\mathsf{Verify}(pp, \mathsf{vk}, m, \sigma)$:

[†]The $\mathsf{SAS}_{\mathsf{LLY}}$ scheme described here is slightly different from the original ones [38] in that the range of H_2 is changed from $\mathbb{G}$ to $\mathbb{G}^*$.

1. $m' \leftarrow H_3(t, m)$, parse σ as (E, t),.
2. If $e(E, g) = e(H_1(t)H_2(t)^{m'}, \mathsf{vk})$, return 1.
3. Otherwise return 0.

- $\mathsf{SAS}_{\mathsf{LLY}}.\mathsf{Aggregate}(pp, (\mathsf{vk}_1, \ldots, \mathsf{vk}_r), (m_1, \ldots, m_r), (\sigma_1, \ldots, \sigma_r))$:

1. For $i = 1$ to r, parse σ_i as (E_i, t_i).
2. If there exists $i \in \{2, \ldots, r\}$ such that $t_i \neq t_1$, return $\perp$.
3. If there exists $(i, j) \in [r] \times [r]$ such that $i \neq j \wedge \mathsf{vk}_i = \mathsf{vk}_j$, return $\perp$.
4. If there exists $i \in [r]$ suth that $\mathsf{SAS}_{\mathsf{LLY}}.\mathsf{Verify}(pp, \mathsf{vk}_i, m_i, \sigma_i) \neq 0$, return $\perp$.
5. $E' \leftarrow \prod_{i=1}^{r} E_i$.
6. Return $\Sigma \leftarrow (E', w)$.

- $\mathsf{SAS}_{\mathsf{LLY}}.\mathsf{AggVerify}(pp, (\mathsf{vk}_1, \ldots, \mathsf{vk}_r), (m_1, \ldots, m_r), \Sigma)$:

1. There exists $(i, j) \in [r] \times [r]$ such that $i \neq j \wedge \mathsf{vk}_i = \mathsf{vk}_j$, return 0.
2. For $i = 1$ to r, $m'_i \leftarrow H_3(t, m_i)$.
3. Parse Σ as (E', w).
4. If $e(E', g) = e\left(H_1(t), \prod_{i=1}^{r} \mathsf{vk}_i\right) \cdot e\left(H_2(t), \prod_{i=1}^{r} \mathsf{vk}_i^{m'_i}\right)$, return 1.
5. Otherwise, return 0.

Now, we confirm the correctness. Let $(\mathsf{vk}_i, \mathsf{sk}_i) \leftarrow \mathsf{SAS}_{\mathsf{LLY}}.\mathsf{KeyGen}(pp)$ and $\sigma_i \leftarrow \mathsf{SAS}_{\mathsf{LLY}}.\mathsf{Sign}(pp, \mathsf{sk}_i, t, m_i)$ for $i \in [r]$ where vk_i are all distinct. Then, for all $i \in [r]$, $E_i \leftarrow H_1(t)^{\mathsf{sk}_i} H_2(t)^{m'_i \mathsf{sk}_i}$ holds where $m'_i \leftarrow H_3(t, m_i)$ and $\sigma_i = (E_i, t)$. This fact implies that $\mathsf{SAS}_{\mathsf{LLY}}.\mathsf{Verify}(pp, \mathsf{vk}_i, m_i, \sigma_i) = 1$. Furthermore, let $\Sigma \leftarrow \mathsf{SAS}_{\mathsf{LLY}}.\mathsf{Aggregate}(pp, (\mathsf{vk}_1, \ldots, \mathsf{vk}_r), (m_1, \ldots, m_r), (\sigma_1, \ldots, \sigma_r))$. Then,

$$E' = \prod_{i=1}^{r} E_i = H_1(t)^{\sum_{i=1}^{n} \mathsf{sk}_i} H_2(t)^{\sum_{i=1}^{n} m'_i \mathsf{sk}_i}$$

holds where $\Sigma = (E', t)$ and $m'_i \leftarrow H_3(t, m_i)$ for all $i \in [r]$. This fact implies that $\mathsf{SAS}_{\mathsf{LLY}}.\mathsf{AggVerify}(pp, (\mathsf{vk}_1, \ldots, \mathsf{vk}_r), (m_1, \ldots, m_r), \Sigma) = 1$.

3.5 Conversion from MCL Signature to Aggregate Signature by Lee et al.

Before security analysis the synchronized aggregate signature proposed by Lee et al, we explain an intuition that there is a relationship between the $\mathsf{DS}_{\mathsf{MCL}}$ scheme and Lee et al.'s aggregate signature scheme. Concretely, we explain that there is a conversion from the $\mathsf{DS}_{\mathsf{MCL}}$ scheme to Lee et al.'s aggregate signature scheme.

Our idea of conversion is a similar technique in [38] which converts the Camenisch-Lysyanskaya signature CL scheme to the synchronized aggregate signature scheme. However, the form of signatures in CL and $\mathsf{DS}_{\mathsf{MCL}}$, we cannot immediately convert $\mathsf{DS}_{\mathsf{MCL}}$ scheme to the synchronized aggregate signature scheme. Thus, we need to modify the conversion technique in [38].

Now, we explain an intuition of our conversion. We start from the $\mathsf{DS}_{\mathsf{MCL}}$ scheme in Section 3.3. A signature of the $\mathsf{DS}_{\mathsf{MCL}}$ scheme on a message m is formed as

$$\sigma = (w, A, B = A^y, C = A^z, D = C^y, E = A^x B^{mx} D^{wx}).$$

where $w \xleftarrow{\$} \mathbb{Z}_p$ and $A \xleftarrow{\$} \mathbb{G}_1^*$. If we can force signers to use same w, A, $B = A^y$, $C = A^z$, and $D = C^y$, we can obtain an aggregate signature

$$\Sigma = \left(w, A, B, C, D, E' = \prod_{i=1}^{r} E_i = A^{\sum_{i=1}^{r} x_i} B^{\sum_{i=1}^{r} m_i x_i} D^{\sum_{i=1}^{r} w x_i} \right)$$

on a message list $(m_1, \ldots, m_r)$ from valid signatures $(\sigma_1, \ldots \sigma_r)$ where $\sigma_i = (w, A, B, C, D, E_i)$ is a signature on a message m_i generated by each signer. If we regard E' as $E' = (AD^w)^{\sum_{i=1}^{r} x_i} B^{\sum_{i=1}^{r} m_i x_i}$, verification of the aggregate signature Σ on the message list $(m_1, \ldots, m_r)$ can be done by checking the following equation.

$$e(E', g) = e\left(AD^w, \prod_{i=1}^{r} \mathsf{vk}_i \right) \cdot e\left(B, \prod_{i=1}^{r} \mathsf{vk}_i^{m_i} \right)$$

Then, required elements to verify the aggregate signature Σ are $F = AD^w$, B, and E'. Similar to Lee et al.'s conversion, the three verification equations $e(A, Y) = e(B, g)$, $e(A, Z) = e(C, g)$, $e(C, Y) = e(D, g)$ in $\mathsf{DS}_{\mathsf{MCL}}.\mathsf{Verify}$ is discarded in this conversion. We use hash functions to force signers to use the same F and B for each period t. We choose hash functions H_1 and H_2 and set $F \leftarrow H_1(t)$ and $B \leftarrow H_2(t)$. Then, we can derive Lee et al.'s aggregate signature scheme. In this derived aggregate signature scheme, a signature on a message m and period t is formed as

$$\sigma = (E = H_1(t)^x H_2(t)^{mx}, t).$$

An aggregate signature Σ' on a message list $(m_1, \ldots, m_r)$ and period t is formed as

$$\Sigma = \left(E' = \prod_{i=1}^{r} E_i = H_1(t)^{\sum_{i=1}^{r} x_i} H_2(t)^{\sum_{i=1}^{r} m_i x_i}, t \right)$$

where $\sigma_i = (E_i = H_1(t)^{x_i} H_2(t)^{m_i x_i}, t)$ is a signature on a message m_i generated by each signer. In our conversion, we need to hash a message with a time period for the security proof. This conversion technique is used for our security proof in next section.

3.6 New Security Proof

We reassess the EUF-CMA security of the $\mathsf{SAS_{LLY}}$ scheme. In particular, we newly prove the EUF-CMA security of the $\mathsf{SAS_{LLY}}$ scheme under the 1-MSDH-2 assumption. By using the conversion technique described previous section, we simulate a signature in the aggregate signature scheme Lee et al. in the reduction.

Theorem 3.4. If the $\mathsf{DS_{MCL}}$ scheme satisfies the OT-EUF-CMA security, then, in the random oracle model, the $\mathsf{SAS_{LLY}}$ scheme satisfies the EUF-CMA security in the certified-key model.

proof. We give an overview of our security proof. Similar to the work in [38], we reduce the EUF-CMA security of the $\mathsf{SAS_{LLY}}$ scheme to the OT-EUF-CMA security of the $\mathsf{DS_{MCL}}$ scheme. We construct a reduction algorithm according to the following strategy. First, the reduction algorithm chooses a message $m_{\mathsf{DS_{MCL}}}$ at random, make signing query on $m_{\mathsf{DS_{MCL}}}$, and obtains its signature $\sigma_{\mathsf{DS_{MCL}}} = (w_{\mathsf{DS_{MCL}}}, A_{\mathsf{DS_{MCL}}}, B_{\mathsf{DS_{MCL}}}, C_{\mathsf{DS_{MCL}}}, D_{\mathsf{DS_{MCL}}}, E_{\mathsf{DS_{MCL}}})$ of the $\mathsf{DS_{MCL}}$ scheme. Then, the reduction algorithm guesses the time period t' of a forged aggregate signature and an index $k' \in [q_{H_3}]$ at random where q_{H_3} be the maximum number of H_3 hash queries. Then reduction algorithm programs hash values as $H_1(t') = A_{\mathsf{DS_{MCL}}} D_{\mathsf{DS_{MCL}}}^{w_{\mathsf{DS_{MCL}}}}$, $H_2(t') = B_{\mathsf{DS_{MCL}}}$, and $H_3(t', m_{k'}) = m_{\mathsf{DS_{MCL}}}$. For a signing query on period $t \neq t'$, the reduction algorithm generate the signature by programmability of hash functions H_1, H_2, and H_3. For a signing query on period $t \neq t'$, if the query index j of H_3 is equal to the index k', the reduction algorithm can compute a valid signature by using $\sigma_{\mathsf{DS_{MCL}}}$ (This can be done by using the conversion technique in Section 3.5.). Otherwise, the algorithm should abort the simulation. Finally, the reduction algorithm extracts valid forgery of the $\mathsf{DS_{MCL}}$ scheme from a forged aggregate signature on time period t' of the $\mathsf{SAS_{LLY}}$ scheme.

Now, we give the security proof. Let A be an EUF-CMA adversary of the $\mathsf{SAS_{LLY}}$ scheme, C be the OT-EUF-CMA game challenger of the $\mathsf{DS_{MCL}}$ scheme, and q_{H_3} be the maximum number of H_3 hash queries. We construct the algorithm B against the OT-EUF-CMA game of the $\mathsf{DS_{MCL}}$ scheme. The construction of B is given as follow.

- **Initial setup:** Given an input $pp = \mathcal{G}_{\mathsf{DS_{MCL}}}$ and $\mathsf{vk} = (g_{\mathsf{DS_{MCL}}}, X_{\mathsf{DS_{MCL}}}, Y_{\mathsf{DS_{MCL}}}, Z_{\mathsf{DS_{MCL}}})$ from C, B performs the following procedure.
  - $\mathcal{G} \leftarrow \mathcal{G}_{\mathsf{DS_{MCL}}}$, $g \leftarrow g_{\mathsf{DS_{MCL}}}$, $pp^* \leftarrow (\mathcal{G}, g)$, $\mathsf{vk}^* \leftarrow X_{\mathsf{DS_{MCL}}}$. $t' \xleftarrow{\$} [T]$, $k' \xleftarrow{\$} [q_{H_3}]$, $t_{ctr} \leftarrow 1$, $L \leftarrow \{\}$, $\mathbb{T}_1 \leftarrow \{\}$, $\mathbb{T}_2 \leftarrow \{\}$, $\mathbb{T}_3 \leftarrow \{\}$, $Q \leftarrow \{\}$.
  - $m_{\mathsf{DS_{MCL}}} \xleftarrow{\$} \mathbb{Z}_p$, query C for the signature on the message $m_{\mathsf{DS_{MCL}}}$ and get its signature $\sigma_{\mathsf{DS_{MCL}}} = (w_{\mathsf{DS_{MCL}}}, A_{\mathsf{DS_{MCL}}}, B_{\mathsf{DS_{MCL}}}, C_{\mathsf{DS_{MCL}}}, D_{\mathsf{DS_{MCL}}}, E_{\mathsf{DS_{MCL}}})$,
  - Give (pp^*, vk^*) to A as an input.
- $\mathcal{O}^{\mathsf{Cert}}(\mathsf{vk}, \mathsf{sk})$: If $\mathsf{vk} = g^{\mathsf{sk}}$, update a list $L \leftarrow L \cup \{\mathsf{vk}\}$ and return "`accept`" to A. Otherwise return "`reject`" to A.
- $\mathcal{O}^{H_1}(t_i)$: Given an input t_i, B responds as follows.
  - If there is an entry $(t_i, \cdot, F_i)$ ('$\cdot$' represents an arbitrary value or $\perp$) for some $F_i \in \mathbb{G}_1$ in $\mathbb{T}_1$, return F_i.
  - If $t_i \neq t'$, $r_{(1,i)} \xleftarrow{\$} \mathbb{Z}_p$, $F_i \leftarrow g^{r_{(1,i)}}$, $\mathbb{T}_1 \leftarrow \mathbb{T}_1 \cup \{(t_i, r_{(1,i)}, F_i)\}$, return F_i.
  - If $t_i = t'$, $\mathbb{T}_1 \leftarrow \mathbb{T}_1 \cup \{(t_i, \perp, A_{\mathsf{DS_{MCL}}} D_{\mathsf{DS_{MCL}}}^{w_{\mathsf{DS_{MCL}}}}\}$, return $A_{\mathsf{DS_{MCL}}} D_{\mathsf{DS_{MCL}}}^{w_{\mathsf{DS_{MCL}}}}$.
- $\mathcal{O}^{H_2}(t_i)$: Given an input t_i, B responds as follows.
  - If there is an entry $(t_i, \cdot, B_i)$ ('$\cdot$' represents an arbitrary value or $\perp$) for some $B_i \in \mathbb{G}_1^*$ in $\mathbb{T}_3$, return B_i.
  - If $t_i \neq t'$, $r_{(2,i)} \xleftarrow{\$} \mathbb{Z}_p^*$, $B_i \leftarrow g^{r_{(2,i)}}$, $\mathbb{T}_2 \leftarrow \mathbb{T}_2 \cup \{(t_i, r_{(2,i)}, B_i)\}$, return D_i.
  - If $t_i = t'$, $\mathbb{T}_2 \leftarrow \mathbb{T}_2 \cup \{(t_i, \perp, B_{\mathsf{DS_{MCL}}})\}$, return $B_{\mathsf{DS_{MCL}}}$.
- $\mathcal{O}^{H_3}(t_i, m_j)$: Given an input (t_i, m_j), B responds as follows.
  - If there is an entry $(t_i, m_j, m'_{(i,j)})$ for some $m'_{(i,j)} \in \mathbb{Z}_p$ in $\mathbb{T}_3$, return $m'_{(i,j)}$.
  - If $t_i \neq t' \vee j \neq k'$, $m'_{(i,j)} \xleftarrow{\$} \mathbb{Z}_p$, $\mathbb{T}_3 \leftarrow \mathbb{T}_3 \cup \{(t_i, m_j, m'_{(i,j)})\}$, return $m'_{(i,j)}$.
  - If $t_i = t' \wedge j = k'$, $\mathbb{T}_3 \leftarrow \mathbb{T}_3 \cup \{(t_i, m_j, m_{\mathsf{DS_{MCL}}})\}$, return $m_{\mathsf{DS_{MCL}}}$.
- $\mathcal{O}^{\mathsf{Sign}}($"`inst`"$, m_j)$: Given an input $($"`inst`"$, m_j)$, B performs the following procedure.

- If $t_{ctr} \notin [T]$, return $\perp$.
- If "inst" = "skip", $t_{ctr} \leftarrow t_{ctr} + 1$.
- If "inst" = "sign",
  * If $t_{ctr} \neq t'$, $E \leftarrow X_{\mathsf{DS_{MCL}}}^{r_{(1,ctr)}} X_{\mathsf{DS_{MCL}}}^{r_{(2,ctr)} m'_{(ctr,j)}}$ where $r_{(1,i)}$, $r_{(2,i)}$, and $m'_{(i,j)}$ are retreived from $(t_{ctr}, r_{(1,ctr)}, F_{ctr}) \in \mathbb{T}_1$, $(t_{ctr}, r_{(2,ctr)}, B_{ctr}) \in \mathbb{T}_2$, and $(t_{ctr}, m_j, m'_{(ctr,j)}) \in \mathbb{T}_3$ respectively. $Q \leftarrow Q \cup \{m_j\}$, return $\sigma_{ctr,j} \leftarrow (E, t_{ctr})$, then update $t_{ctr} \leftarrow t_{ctr}+1$.
  * If $t_{ctr} = t' \wedge j = k'$, $Q \leftarrow Q \cup \{m_j\}$, return $\sigma_{ctr,j} \leftarrow (E_{\mathsf{DS_{MCL}}}, t_i)$, then update $t_{ctr} \leftarrow t_{ctr} + 1$
  * If $t_{ctr} = t' \wedge j \neq k'$, abort the simulation.

- **Output procedure:** B receives a forgery $((\mathsf{vk}_1^*, \ldots, \mathsf{vk}_{r^*}^*), (m_1^*, \ldots, m_{r^*}^*), \Sigma^*)$ outputted by A. Then B proceeds as follows.

  1. If $\mathsf{SAS_{LLY}}.\mathsf{AggVerify}(pp^*, (\mathsf{vk}_1^*, \ldots, \mathsf{vk}_{r^*}^*), (m_1^*, \ldots, m_{r^*}^*), \Sigma^*) \neq 1$, then abort.
  2. If there exists $j \in [r^*]$ such that $\mathsf{vk}_j^* \neq \mathsf{vk}^* \wedge \mathsf{vk}_j^* \notin L$, then abort.
  3. If there is no $j^* \in [r^*]$ such that $\mathsf{vk}_{j^*}^* = \mathsf{vk}^* \wedge m_{j^*}^* \notin Q$, then abort.
  4. Set $j^* \in [r^*]$ such that $\mathsf{vk}_{j^*}^* = \mathsf{vk}^* \wedge m_{j^*}^* \notin Q$.
  5. Parse Σ^* as $(E^{*\prime}, t^*)$.
  6. If $t^* \neq t'$, then abort.
  7. $m_{j^*}^{*\prime} \leftarrow H_3(t^*, m_{j^*}^*)$
  8. If $m_{j^*}^{*\prime} = m_{\mathsf{DS_{MCL}}}$, then abort.
  9. For $i \in [r^*] \backslash \{j^*\}$, retrieve $x_i \leftarrow \mathsf{sk}_i^*$ of vk_i^* from L.
  10. $F' \leftarrow H_1(t^*)$, $B' \leftarrow H_2(t^*)$, $m'_i \leftarrow H_3(t^*, m_i^*)$ for $i \in [r^*] \backslash \{j^*\}$,
  $E' \leftarrow E^{*\prime} \cdot \left(F'^{\sum_{i \in [r^*] \backslash \{j^*\}} x_i} B'^{\sum_{i \in [r^*] \backslash \{j^*\}} x_i m'_i} \right)^{-1}$.
  11. Return $(m_{\mathsf{DS_{MCL}}}^*, \sigma_{\mathsf{DS_{MCL}}}^*) \leftarrow (m_{j^*}^*, (w_{\mathsf{DS_{MCL}}}, A_{\mathsf{DS_{MCL}}}, B', C_{\mathsf{DS_{MCL}}}, D_{\mathsf{DS_{MCL}}}, E'))$.

We confirm that if B does not abort, B can simulate the EUF-CMA game of the $\mathsf{SAS_{LLY}}$ scheme.

- **Initial setup:** First, we discuss the distribtuon of pp^*. In the original EUF-CMA game of the $\mathsf{SAS_{LLY}}$ scheme, $pp^* = (\mathcal{G}, g)$ is constructed by $\mathcal{G} = (p, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathsf{G}(1^\lambda)$ and $g \xleftarrow{\$} \mathbb{G}^*$. In the simulation of B, pp^* is a tuple $(\mathcal{G}_{\mathsf{DS_{MCL}}}, g_{\mathsf{DS_{MCL}}})$. This tuple is constructed by C as $\mathcal{G}_{\mathsf{DS_{MCL}}} = (p, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathsf{G}(1^\lambda)$ and $g_{\mathsf{DS_{MCL}}} \xleftarrow{\$} \mathbb{G}^*$. Therefore, B simulates pp^* perfectly. Next, we discuss the distribution of vk^*. In the original EUF-CMA game of the $\mathsf{SAS_{LLY}}$ scheme, vk is computed by $x \xleftarrow{\$} \mathbb{Z}_p^*$ and $\mathsf{vk}^* \leftarrow g^x$. In the simulation of B, vk^* is set by $X_{\mathsf{DS_{MCL}}}$. Since $X_{\mathsf{DS_{MCL}}}$ is computed by C as $x_{\mathsf{DS_{MCL}}} \xleftarrow{\$} Z_p$ and $X_{\mathsf{DS_{MCL}}} \leftarrow g^{x_{\mathsf{DS_{MCL}}}}$, distributions of vk between the original game and simulation of B are identical. Hence, the distributions of (pp^*, vk^*) are identical.

- **Output of $\mathcal{O}^{\mathsf{Cert}}$:** This is clearly that B can simulate the original EUF-CMA game of the $\mathsf{SAS_{LLY}}$ scheme perfectly.

- **Output of $\mathcal{O}^{H_1}$:** In the original game, hash values of H_1 is chosen from $\mathbb{G}$ uniformly at random. In the simulation of B, if $t_i \neq t'$, the hash value $H(t_i)$ is set by $g^{r_{(1,i)}}$ where $r_{(1,i)} \xleftarrow{\$} \mathbb{Z}_p$. Obviously, in this case, B can simulate $\mathcal{O}^{H_1}$ perfectly. If $t_i = t'$, the hash value $H(t_i)$ is set by $F = A_{\mathsf{DS_{MCL}}} D_{\mathsf{DS_{MCL}}}^{w_{\mathsf{DS_{MCL}}}} = A_{\mathsf{DS_{MCL}}}^{1+y_{\mathsf{DS_{MCL}}} z_{\mathsf{DS_{MCL}}} w_{\mathsf{DS_{MCL}}}}$ where $Y_{\mathsf{DS_{MCL}}} = g_{\mathsf{DS_{MCL}}}^{y_{\mathsf{DS_{MCL}}}}$, $Z_{\mathsf{DS_{MCL}}} = g_{\mathsf{DS_{MCL}}}^{z_{\mathsf{DS_{MCL}}}}$, and $w_{\mathsf{DS_{MCL}}}$ is chosen by C as $w_{\mathsf{DS_{MCL}}} \leftarrow \mathbb{Z}_p$. For fixed $y_{\mathsf{DS_{MCL}}} \in \mathbb{Z}_p^*$ and $z_{\mathsf{DS_{MCL}}} \in \mathbb{Z}_p^*$, the distribution α where $\alpha \xleftarrow{\$} \mathbb{Z}_p$ and $w_{\mathsf{DS_{MCL}}} \xleftarrow{\$} \mathbb{Z}_p$, $\alpha \leftarrow 1 + y_{\mathsf{DS_{MCL}}} z_{\mathsf{DS_{MCL}}} w_{\mathsf{DS_{MCL}}}$ are identical. This fact implies that B also simulate $\mathcal{O}^{H_1}$ perfectly in the case of $t_i = t'$. Therefore, B simulates $\mathcal{O}^{H_1}$ perfectly.

- **Output of $\mathcal{O}^{H_2}$:** As the same argument of $\mathcal{O}^{H_1}$, if $t_i \neq t'$, B can simulate hash values $H(t_i)$ perfectly. In the case of $t_i = t'$, the hash value $H(t_i)$ is set by $B_{\mathsf{DS_{MCL}}} = A^{y_{\mathsf{DS_{MCL}}}} = g^{x_{\mathsf{DS_{MCL}}} y_{\mathsf{DS_{MCL}}}}$. For fixed $x_{\mathsf{DS_{MCL}}} \in \mathbb{Z}_p^*$, the distributions of B where $y_{\mathsf{DS_{MCL}}} \xleftarrow{\$} \mathbb{Z}_p^*$, $B \leftarrow g^{x_{\mathsf{DS_{MCL}}} y_{\mathsf{DS_{MCL}}}}$ and $B \xleftarrow{\$} \mathbb{G}^*$ are identical. Therefore, B simulates $\mathcal{O}^{H_2}$ perfectly.

- **Output of $\mathcal{O}^{H_3}$:** If $t_i \neq t' \vee j \neq k'$, clearly B can simulate $\mathcal{O}^{H_3}$ perfectly. If $t_i = t' \wedge j = k'$, the hash value $H_3(t_i, m_j)$ is set by $m_{\mathsf{DS_{MCL}}}$. Since $m_{\mathsf{DS_{MCL}}}$ is chosen by B as $m_{\mathsf{DS_{MCL}}} \xleftarrow{\$} \mathbb{Z}_p$, B simulates $\mathcal{O}^{H_3}$ perfectly.

- **Output of $\mathcal{O}^{\mathsf{Sign}}$:** For the sake of argument, we denote $X_{\mathsf{DS_{MCL}}} = g_{\mathsf{DS_{MCL}}}^{x_{\mathsf{DS_{MCL}}}}$ $(x_{\mathsf{DS_{MCL}}} \in \mathbb{Z}_p^*)$. If $t_i \neq t'$, B sets $E \leftarrow X_{\mathsf{DS_{MCL}}}^{r_{(1,i)}} X_{\mathsf{DS_{MCL}}}^{r_{(2,i)} m'_{(i,j)}}$ and output the signature $\sigma = (E, t_i)$. Now we confirm that σ is a valid signature on the message m_j. The following equation

$$
\begin{aligned}
E = X_{\mathsf{DS_{MCL}}}^{r_{(1,i)}} X_{\mathsf{DS_{MCL}}}^{r_{(2,i)} m'_{(i,j)}} &= (g_{\mathsf{DS_{MCL}}}^{x_{\mathsf{DS_{MCL}}}})^{r_{(1,i)}} (g_{\mathsf{DS_{MCL}}}^{x_{\mathsf{DS_{MCL}}}})^{r_{(2,i)} m'_{(i,j)}} \\
&= H_1(t_i)^{x_{\mathsf{DS_{MCL}}}} H_2(t_i)^{x_{\mathsf{DS_{MCL}}} m'_{(i,j)}}
\end{aligned}
$$

holds where $m'_{(i,j)} = H_3(t_i, m_j)$. This fact implies that

$$
e(E, g) = e(H_1(t_i) H_2(t_i)^{m'_{(i,j)}}, \mathsf{vk}^*)
$$

holds. Therefore, σ is valid signature on the message m_j.

If $t_i \neq t' \wedge j = k'$, B sets $E \leftarrow E_{\mathsf{DS}_{\mathsf{MCL}}}$, return $\sigma_{i,j} \leftarrow (E, t_i)$ to A. We also confirm that σ is a valid signature on the message m_j. In the case, $H_1(t_i) = A_{\mathsf{DS}_{\mathsf{MCL}}} D_{\mathsf{DS}_{\mathsf{MCL}}}^{w_{\mathsf{DS}_{\mathsf{MCL}}}}$, $H_2(t_i) = B_{\mathsf{DS}_{\mathsf{MCL}}}$, and $H_3(t_i, m_j) = m'_{(i,j)} = m_{\mathsf{DS}_{\mathsf{MCL}}}$ hold. Since $E_{\mathsf{DS}_{\mathsf{MCL}}}$ is the valid signature of the $\mathsf{DS}_{\mathsf{MCL}}$ scheme on message $m_{\mathsf{DS}_{\mathsf{MCL}}}$,

$$\begin{aligned} e(E_{\mathsf{DS}_{\mathsf{MCL}}}, g) &= e(A_{\mathsf{DS}_{\mathsf{MCL}}} B_{\mathsf{DS}_{\mathsf{MCL}}}^{m_{\mathsf{DS}_{\mathsf{MCL}}}} D_{\mathsf{DS}_{\mathsf{MCL}}}^{w_{\mathsf{DS}_{\mathsf{MCL}}}}, X_{\mathsf{DS}_{\mathsf{MCL}}}) \\ &= e((A_{\mathsf{DS}_{\mathsf{MCL}}} D_{\mathsf{DS}_{\mathsf{MCL}}}^{w_{\mathsf{DS}_{\mathsf{MCL}}}}) B_{\mathsf{DS}_{\mathsf{MCL}}}^{m_{\mathsf{DS}_{\mathsf{MCL}}}}, X_{\mathsf{DS}_{\mathsf{MCL}}}) \end{aligned}$$

holds. This implies that $e(E, g) = e(H_1(t_i) H_2(t_i)^{m'_{(i,j)}}, \mathsf{vk}^*)$ where $m'_{(i,j)} = H_3(t_i, m_j)$.

By the above discussion, we can see that B does not abort, B can simulate the EUF-CMA game of the $\mathsf{SAS}_{\mathsf{LLY}}$ scheme.

Second, we confirm that when A successfully output a valid forgery $((\mathsf{vk}_1^*, \ldots, \mathsf{vk}_{r^*}^*), (m_1^*, \ldots, m_{r^*}^*), \Sigma^*)$ of the $\mathsf{SAS}_{\mathsf{LLY}}$ scheme, B can forge a signature of the $\mathsf{DS}_{\mathsf{MCL}}$ scheme. Let $((\mathsf{vk}_1^*, \ldots, \mathsf{vk}_{r^*}^*), (m_1^*, \ldots, m_{r^*}^*), \Sigma^*)$ be a valid forgery output by A. Then there exists $j^* \in [r^*]$ such that $\mathsf{vk}_{j^*}^* = \mathsf{vk}^*$. By the verification equation of $\mathsf{SAS}_{\mathsf{LLY}}.\mathsf{Verify}$,

$$e(E^{*\prime}, g) = e\left(H_1(t^*), \prod_{i=1}^{r^*} \mathsf{vk}_i^*\right) \cdot e\left(H_2(t^*), \prod_{i=1}^{r^*} (\mathsf{vk}_i^*)^{m_i^*}\right)$$

holds where $\Sigma^* = (E^{*\prime}, t^*)$ and $H_3(t^*, m_i^*) = m_i^{*\prime}$ for $i \in [r^*]$. If B does not abort in Step 6 of **Output procedure**, $t^* = t'$ holds. This means that $H_1(t^*) = A_{\mathsf{DS}_{\mathsf{MCL}}} D_{\mathsf{DS}_{\mathsf{MCL}}}^{w_{\mathsf{DS}_{\mathsf{MCL}}}}$ and $H_2(t^*) = B_{\mathsf{DS}_{\mathsf{MCL}}}$ hold. These facts imply that

$$\begin{aligned} E^{*\prime} &= H_1(t^*)^{\sum_{i=1}^{r^*} \mathsf{sk}_i^*} H_2(t^*)^{\sum_{i=1}^{r^*} m_i^{*\prime} \mathsf{sk}_i^*} \\ &= \left(A_{\mathsf{DS}_{\mathsf{MCL}}} D_{\mathsf{DS}_{\mathsf{MCL}}}^{w_{\mathsf{DS}_{\mathsf{MCL}}}}\right)^{\sum_{i=1}^{r^*} x_i^*} B_{\mathsf{DS}_{\mathsf{MCL}}}^{\sum_{i=1}^{r^*} m_i^{*\prime} x_i^*} \end{aligned}$$

holds where $\mathsf{sk}_i^* = x_i^*$ is a secret key corresponding to vk_i^*.

By setting $F' \leftarrow A_{\mathsf{DS}_{\mathsf{MCL}}} D_{\mathsf{DS}_{\mathsf{MCL}}}^{w_{\mathsf{DS}_{\mathsf{MCL}}}}$ and $B' \leftarrow B_{\mathsf{DS}_{\mathsf{MCL}}}$,

$$\begin{aligned} E' &= E^{*\prime} \cdot \left(F'^{\sum_{i \in [r^*] \setminus \{j^*\}} x_i} B'^{\sum_{i \in [r^*] \setminus \{j^*\}} x_i m_i'}\right)^{-1} \\ &= (A_{\mathsf{DS}_{\mathsf{MCL}}} D_{\mathsf{DS}_{\mathsf{MCL}}}^{w_{\mathsf{DS}_{\mathsf{MCL}}}})^{x_{j^*}^*} B_{\mathsf{DS}_{\mathsf{MCL}}}^{m_{j^*}^{*\prime} x_{j^*}^*} \end{aligned}$$

Moreover, $e(A_{\mathsf{DS}_{\mathsf{MCL}}}, Y_{\mathsf{DS}_{\mathsf{MCL}}}) = e(B_{\mathsf{DS}_{\mathsf{MCL}}}, g_{\mathsf{DS}_{\mathsf{MCL}}})$, $e(A_{\mathsf{DS}_{\mathsf{MCL}}}, Z_{\mathsf{DS}_{\mathsf{MCL}}}) = e(C_{\mathsf{DS}_{\mathsf{MCL}}}, g_{\mathsf{DS}_{\mathsf{MCL}}})$, and $e(C_{\mathsf{DS}_{\mathsf{MCL}}}, Y_{\mathsf{DS}_{\mathsf{MCL}}}) = e(D_{\mathsf{DS}_{\mathsf{MCL}}}, g_{\mathsf{DS}_{\mathsf{MCL}}})$ holds. If B does not abort in Step 8 of **Output procedure**, $m_{j^*}^*$ is a not queried message for the signing of the OT-EUF-CMA game of the $\mathsf{DS}_{\mathsf{MCL}}$

scheme. Therefore, if B does not abort and outputs $(m^*_{\mathsf{DS}_{\mathsf{MCL}}}, \sigma^*_{\mathsf{DS}_{\mathsf{MCL}}}) \leftarrow (m^*_{j^*}, (w_{\mathsf{DS}_{\mathsf{MCL}}}, A_{\mathsf{DS}_{\mathsf{MCL}}}, B', C_{\mathsf{DS}_{\mathsf{MCL}}}, D_{\mathsf{DS}_{\mathsf{MCL}}}, E'))$, B can forge a signature of the $\mathsf{DS}_{\mathsf{MCL}}$ scheme.

Finally, we analyze the probability that B succeeds in forging a signature of the $\mathsf{DS}_{\mathsf{MCL}}$ scheme. First, we consider the probability that B does not abort at the simulation of signatures. B aborts the simulation of $\mathcal{O}^{\mathsf{Sign}}$ if $t_{ctr} = t' \wedge j \neq k'$. The probability that B succeeds in simulating $\mathcal{O}^{\mathsf{Sign}}$ is at least $1/q_{H_3}$. Next, we consider the probability that B aborts in Step 6 of **Output procedure**. Since B chooses the target period $t' \leftarrow [T]$, the probability $t^* \neq t'$ is $1/[T]$. Finally, the probability that B aborts in Step 8 of **Output procedure** is $1/p$. Let $\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{SAS}_{\mathsf{LLY}},\mathsf{A}}$ be the advantage of the EUF-CMA game for the $\mathsf{SAS}_{\mathsf{LLY}}$ scheme of A. The advantage of the OT-EUF-CMA game for the $\mathsf{DS}_{\mathsf{MCL}}$ scheme of B is

$$\mathsf{Adv}^{\mathsf{OT\text{-}EUF\text{-}CMA}}_{\mathsf{DS}_{\mathsf{MCL}},\mathsf{B}} \geq \frac{\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{SAS}_{\mathsf{LLY}},\mathsf{A}}}{T \times q_{H_3}} \left(1 - \frac{1}{p}\right).$$

Therefore, we can conclude the proof of Theorem 3.4. □ □

By combining Theorem 3.3 and Theorem 3.4, we have the following corollary.

Corollary 3.5. If the 1-MSDH-2 assumption holds, then, in the ROM, the $\mathsf{SAS}_{\mathsf{LLY}}$ scheme satisfies the EUF-CMA security in the certified-key model.

Chapter 4

T-out-of-N Redactable Signature

In this chapter, at first, we introduce the notion of t-out-of-n redactable signature, define its security. Second, we review the aggregate signature scheme by Boneh, Gentry, Lynn, and Shacham [11] and Shamir's secret sharing scheme [56]. Then, we give a t-out-of-n redactable signature scheme by using these primitives. Finally, we prove that our scheme satisfies unforgeability and transparency.

4.1 Syntax

We explain the t-out-of-n redactable signature scheme in the one-time redaction model. A t-out-of-n redactable signature scheme in the one-time redaction model (t, n)-RS is a signature scheme that has a signer, n redactors, a combiner, and a verifier. The signer designates n redactors and the combiner.

The signer selects a threshold t and the number of redactors n. Then, he or she runs key generation algorithm and gets $(\mathsf{vk}, \mathsf{sk}, \{\mathsf{rk}[i]\}_{i=1}^{n})$. The vk is published and the redactor's key $\mathsf{rk}[i]$ is sent to the redactor i.

The signer signs a message M with an admissible description ADM which represents parts of the message that redactors cannot remove from the message M. In the processing of the signing, a random document ID (DID) is added to the message M, then the signature σ is generated. $(M, \mathsf{ADM}, \mathsf{DID}, \sigma)$ generated by the signer is sent to n redactors and the combiner.

Each redactor i checks whether DID has never been seen before. If he or she has seen it, then aborts. Also, if the signature is invalid, then aborts. Otherwise, he or she selects parts of the message that he or she wants to remove and makes the redaction information RI_i and sends it to the combiner. The protocol works only once for DID which redactors have not seen before.

The combiner collects pieces of redaction information $\{\mathsf{RI}_i\}_{i=1}^n$. From $\{\mathsf{RI}_i\}_{i=1}^n$, the combiner extracts parts which at least t redactor want to remove. Finally, the combiner outputs the redacted message M', ADM, DID, and its updated valid signature σ'.

The signature is verified using the signer's public key vk. In the verification, it is possible to prove the validity of the $(M, \mathsf{ADM}, \mathsf{DID}, \sigma)$ made by a legitimate signer or redacted by the redaction protocol for that signature while keeping redactors anonymity.

Now, we formalize the t-out-of-n redactable signature scheme in the one-time redaction model for set. In the following, we assume that a message M is a set and use following notations. An admissible description ADM is a set containing all elements which must not be redacted.
A modification instruction MOD is a set containing all elements which a redactor want to redact from M. $\mathsf{ADM} \preceq M$ means that ADM is a valid description. (i.e.,$\mathsf{ADM} \cap M = \mathsf{ADM}$.) $\mathsf{MOD} \overset{\mathsf{ADM}}{\preceq} M$ means that MOD is valid redaction description respect to ADM and M. (i.e., $\mathsf{MOD} \cap \mathsf{ADM} = \emptyset \wedge \mathsf{MOD} \subset M$.) A redaction $M' \overset{\mathsf{MOD}}{\leftarrow} M$ would be $M' \leftarrow M \backslash \mathsf{MOD}$. In the following definition, we explicit ADM and DID in the syntax.

Definition 4.1. A t-out-of-n redactable signature scheme in the one-time redaction model (t, n)-RS is composed of four components ($\mathsf{RS.Setup}$, $\mathsf{RS.KeyGen}$, $\mathsf{RS.Sign}$, $\mathsf{RS.Redact}$, $\mathsf{RS.Verify}$).

- $\mathsf{RS.Setup}(1^\lambda)$: A setup algorithm is a randomized algorithm. Given a security parameter λ, return the public parameter pp. We assume that pp defines the message space $\mathcal{M}_{pp}$.
- $\mathsf{RS.KeyGen}(pp, t, n)$: A key generation algorithm is a randomized algorithm that a signer runs. Given a public parameter 1^λ, a threshold t and the number of redactors n, return a signer's public key vk, a signer's secret key sk, and redactor's secret keys $\{\mathsf{rk}[i]\}_{i=1}^n$.
- $\mathsf{RS.Sign}(pp, \mathsf{sk}, M, \mathsf{ADM})$: A signing algorithm is a randomized algorithm that a signer runs. Given a public parameter pp, signer's secret key sk, a message M and an admissible description ADM, return a message M, an admissible description ADM, a document ID (DID), and a signature σ.
- $\mathsf{RS.Redact}$: A redact protocol is a 1-round interactive protocol between the combiner and n redactors. Each redactor i generates redaction information RI_i and sends to the combiner. The combiner collects all redaction informations $\{\mathsf{RI}_i\}_{i=1}^n$ and finally outputs the redacted signature $(M', \mathsf{ADM}, \mathsf{DID}, \sigma')$. We describe the protocol as follows:

– Given an input $(M, \mathsf{ADM}, \mathsf{DID}, \sigma)$ from the signer, each redactor i selects a modification instruction MOD_i and runs a redact information algorithm $\mathsf{RS.RedInf}$ with $(pp, \mathsf{vk}, \mathsf{rk}[i], M, \mathsf{ADM}, \mathsf{DID}, \sigma, \mathsf{MOD}_i, \mathbb{L}_i^{u-1})$. $\mathbb{L}_i^{u-1}$ is the list which stores on DID sent from the signer. It is used for t-th input of the $\mathsf{RS.RedInf}$ by redactor i and $L^0 = \emptyset$. In the processing in $\mathsf{RS.RedInf}$, if DID is previous input to $\mathsf{RS.RedInf}$ then redactor i stop interacting with a combiner. Otherwise, output the redact information RI_i and the updated list $\mathbb{L}_i^u$. Each redactor i sends RI_i to the combiner.

– The combiner runs a deterministic threshold redact algorithm $\mathsf{RS.ThrRed}$ with $(pp, \mathsf{vk}, M, \mathsf{ADM}, \mathsf{DID}, \sigma, \{\mathsf{RI}_i\}_{i=1}^n)$ as an input. In the algorithm $\mathsf{RS.ThrRed}$, MOD is derived from $\{\mathsf{RI}_i\}_{i=1}^n$ and it redacts a message M based on MOD. $\mathsf{RS.ThrRed}$ outputs a redacted message M', ADM, DID and the updated signature σ'. Finally, the combiner outputs $(M', \mathsf{ADM}, \mathsf{DID}, \sigma')$ as an output of $\mathsf{RS.Redact}$ protocol.

- $\mathsf{RS.Verify}(pp, \mathsf{vk}, M, \mathsf{ADM}, \mathsf{DID}, \sigma)$: Given an input $(pp, \mathsf{vk}, M, \mathsf{ADM}, \mathsf{DID}, \sigma)$, return either 1 (Accept) or 0 (Reject).

Correctness

We require the correctness that all honestly computed and redacted signatures are accepted.

Definition 4.2 (Correctness)**.** A t-out-of-n redactable signature scheme in the one-time redaction model (t, n)-RS is correct, $\forall \lambda \in \mathbb{N}$, $\forall k \in \mathbb{N}$,
$\forall M_0^u$, $\forall \mathsf{ADM}^u \preceq M_0^u$, $\forall \mathsf{MOD}_i^u \overset{\mathsf{ADM}^u}{\preceq} M_0^u$ for $u \in [k]$ and $i \in [n]$,
$pp \leftarrow \mathsf{RS.Setup}(1^\lambda)$, $(\mathsf{vk}, \mathsf{sk}, \{\mathsf{rk}[i]\}_{i=1}^n) \leftarrow \mathsf{RS.KeyGen}(pp, t, n)$,
For $u = 1$ to k,
$(M_0, \mathsf{ADM}^u, \mathsf{DID}^u, \sigma_0^u) \leftarrow \mathsf{RS.Sign}(pp, \mathsf{sk}, M_0^u, \mathsf{ADM}^u)$,
For $i \in [n]$,
$(\mathsf{RI}_i^u, \mathbb{L}_i^u) \leftarrow \mathsf{RS.RedInf}(pp, \mathsf{vk}, \mathsf{rk}[i], M_0^u, \mathsf{ADM}^u, \mathsf{DID}^u, \sigma_0^u, \mathsf{MOD}_i^u, \mathbb{L}_i^{u-1})$,
$(M_1^u, \mathsf{ADM}^u, \mathsf{DID}, \sigma_1^u) \leftarrow \mathsf{RS.ThrRed}(pp, \mathsf{vk}, M_0^u, \mathsf{ADM}^u, \mathsf{DID}^u, \sigma_0^u, \{\mathsf{RI}_i^u\}_{i=1}^n)$,
we require the following for all $u \in [k]$:

- If $\mathsf{DID}^k \notin \mathbb{L}_i^{u-1}$, $\mathsf{RS.Verify}(pp, \mathsf{vk}, M_t^u, \mathsf{ADM}^u, \mathsf{DID}^u, \sigma_b^u) = 1$ for all $b \in \{0, 1\}$.

- If $\mathsf{DID}^k \in \mathbb{L}_i^{u-1}$, $\mathsf{RS.Verify}(pp, \mathsf{vk}, M_0^u, \mathsf{ADM}^u, \mathsf{DID}^u, \sigma_0^u) = 1$.

4.2 Security

We give the security notion of unforgeability, privacy, and transparency for a redactable signature scheme in the one-time redaction model.

Unforgeability. Unforgeability requires that without a signer's secret key sk, it should be infeasible to compute a valid signature σ' on $(M', \mathsf{ADM}, \mathsf{DID})$ except to redact a signed message $(M, \mathsf{ADM}, \mathsf{DID}, \sigma)$ even if $t-1$ redactors keys are corrupted.

Definition 4.3 (Unforgeability)**.** The unforgeability against redactors security of a t-out-of-n redactable signature scheme in the one-time redaction model (t, n)-RS Π is defined by the following unforgeability game between a challenger C and a PPT adversary A.

1. C runs $pp \leftarrow \mathsf{RS.Setup}(1^\lambda)$, $(\mathsf{vk}, \mathsf{sk}, \{\mathsf{rk}[i]\}_{i=1}^n) \leftarrow \mathsf{RS.KeyGen}(pp, t, n)$, and gives (pp, vk) to an adversary A.

2. A is given access (throughout the entire game) to a sign oracle $\mathcal{O}^{\mathsf{Sign}}(\cdot, \cdot)$ such that $\mathcal{O}^{\mathsf{Sign}}(M, \mathsf{ADM})$, returns $(M, \mathsf{ADM}, \mathsf{DID}, \sigma) \leftarrow \mathsf{RS.Sign}(pp, \mathsf{sk}, M, \mathsf{ADM})$.

3. A is given access (throughout the entire game) to a redact oracle $\mathcal{O}^{\mathsf{Redact}}(\cdot, \cdot, \cdot, \cdot, \cdot)$. $\mathcal{O}^{\mathsf{Redact}}$ is defined as follows:
   For an u-th query $(M, \mathsf{ADM}, \mathsf{DID}, \sigma, \mathsf{MOD})$:
   (a) $(\mathsf{RI}_i, \mathbb{L}_i^u) \leftarrow \mathsf{RS.RedInf}(pp, \mathsf{vk}, \mathsf{rk}[i], M, \mathsf{ADM}, \mathsf{DID}, \sigma, \mathsf{MOD}, \mathbb{L}_i^{u-1})$ for $i = 1, ..., n$.
   (b) $(M', \mathsf{ADM}, \mathsf{DID}, \sigma') \leftarrow \mathsf{RS.ThrRed}(pp, \mathsf{vk}, M, \mathsf{ADM}, \mathsf{DID}, \sigma, \{\mathsf{RI}_i\}_{i=1}^n)$.
   (c) Return $(M', \mathsf{ADM}, \mathsf{DID}, \sigma')$.

4. A is given up to $t-1$ times access (throughout the entire game) to a corrupt oracle $\mathcal{O}^{\mathsf{Corrupt}}(\cdot)$, where $\mathcal{O}^{\mathsf{Corrupt}}(i)$ outputs a $\mathsf{rk}[i]$ of a redactor i.

5. A outputs $(M^*, \mathsf{ADM}^*, \mathsf{DID}^*, \sigma^*)$.

A t-out-of-n redactable signature scheme in the one-time redaction model (t, n)-RS satisfies the unforgeability security if for all PPT adversaries A, the advantage $\mathsf{Adv}^{\mathsf{Uf\text{-}(t,n)\text{-}RSS}}_{(t,n)\text{-}\mathsf{RS},\mathsf{A}} = \Pr[\mathsf{RS.Verify}(pp, \mathsf{vk}, M^*, \mathsf{ADM}^*\mathsf{DID}^*, \sigma^*) = 1 \wedge (M^*, \mathsf{ADM}^*, \mathsf{DID}^*) \notin (Q_{\mathsf{Sign}} \cup Q_{\mathsf{Redact}})]$ is negligible in λ.

Here, q_s is the total number of queries to $\mathcal{O}^{\mathsf{Sign}}$, (M_i, ADM_i) is an i-th input for $\mathcal{O}^{\mathsf{Sign}}$, $(M^i, \mathsf{ADM}^i, \mathsf{DID}^i, \sigma^i)$ is an i-th output of $\mathcal{O}^{\mathsf{Sign}}$ and $Q_{\mathsf{Sign}} := \bigcup_{i=1}^{q_s}\{(M^i, \mathsf{ADM}^i, \mathsf{DID}^i)\}$. Also, q_r

is the total number of queries to $\mathcal{O}^{\mathsf{Redact}}$, $(M^i, \mathsf{ADM}^i, \mathsf{DID}^i, \sigma^i, \mathsf{MOD}^i)$ is an i-th input for $\mathcal{O}^{\mathsf{Redact}}$, $(M'^i, \mathsf{ADM}^i, \mathsf{DID}^i, \sigma'^i)$ is an i-th output of $\mathcal{O}^{\mathsf{Redact}}$ and $Q_{\mathsf{Redact}} := \bigcup_{i=1}^{q_r} \{(M'^i, \mathsf{ADM}^i, \mathsf{DID}^i)\}$.

Privacy. Privacy requires that except for a signer, n redactors, and a combiner, it is infeasible to derive information on redacted message parts when given a message-ADM-DID-signature pair.

Definition 4.4 (Privacy)**.** The privacy of a t-out-of-n redactable signature scheme in the one-time redaction model (t, n)-RS Π is defined by the following weak privacy game between a challenger C and a PPT adversary A.

1. C runs $pp \leftarrow \mathsf{RS.Setup}(1^\lambda)$, $(\mathsf{vk}, \mathsf{sk}, \{\mathsf{rk}[i]\}_{i=1}^n) \leftarrow \mathsf{RS.KeyGen}(pp, t, n)$, and gives (pp, vk) to an adversary A.

2. A is given access (throughout the entire game) to a sign oracle $\mathcal{O}^{\mathsf{Sign}}(\cdot, \cdot)$ such that $\mathcal{O}^{\mathsf{Sign}}(M, \mathsf{ADM})$, returns $(M, \mathsf{ADM}, \mathsf{DID}, \sigma) \leftarrow \mathsf{RS.Sign}(pp, \mathsf{sk}, M, \mathsf{ADM})$.

3. A is given access (throughout the entire game) to a redact oracle $\mathcal{O}^{\mathsf{Redact}}(\cdot, \cdot, \cdot, \cdot, \cdot)$. $\mathcal{O}^{\mathsf{Redact}}$ is defined as follows:
   For an u-th query $(M, \mathsf{ADM}, \mathsf{DID}, \sigma, \mathsf{MOD})$:
   Let w be the number of queries to $\mathcal{O}^{\mathsf{LoRredact}}$ when A makes an u-th query to $\mathcal{O}^{\mathsf{Redact}}$.

   (a) $(\mathsf{RI}_i, \mathbb{L}_i^{u+2w}) \leftarrow \mathsf{RS.RedInf}(pp, \mathsf{vk}, \mathsf{rk}[i], M, \mathsf{ADM}, \mathsf{DID}, \sigma, \mathsf{MOD}, \mathbb{L}_i^{u+2w-1})$ for $i = 1, ..., n$.

   (b) $(M', \mathsf{ADM}, \mathsf{DID}, \sigma') \leftarrow \mathsf{RS.ThrRed}(pp, \mathsf{vk}, M, \mathsf{ADM}, \mathsf{DID}, \sigma, \{\mathsf{RI}_i\}_{i=1}^n)$.

   (c) Return $(M', \mathsf{ADM}, \mathsf{DID}, \sigma')$.

4. A is given access (throughout the entire game) to a left-or-right redact oracle $\mathcal{O}^{\mathsf{LoRredact}}(\cdot, \cdot, \cdot, \cdot, \cdot, \cdot)$. $\mathcal{O}^{\mathsf{LoRredact}}$ is defined as follows:
   For an w-th query $(M^0, \mathsf{ADM}^0, \mathsf{MOD}^0, M^1, \mathsf{ADM}^1, \mathsf{MOD}^1)$:
   Let u be the number of queries to $\mathcal{O}^{\mathsf{Redact}}$ when A makes an w-th query to $\mathcal{O}^{\mathsf{LoRredact}}$.

   (a) Compute $(M^c, \mathsf{ADM}^c, \mathsf{DID}^c, \sigma^c) \leftarrow \mathsf{Sign}(pp, \mathsf{sk}, M^c, \mathsf{ADM}^c)$ for $c \in \{0, 1\}$.

   (b) For $i = 1, \cdots n$, compute

$$(\mathsf{RI}_i^0, \mathbb{L}^{u+2w-1}) \leftarrow \mathsf{RS.RedInf}(pp, \mathsf{vk}, \mathsf{rk}[i], M^0, \mathsf{ADM}^0, \mathsf{DID}^0, \sigma^0, \mathsf{MOD}^0, \mathbb{L}_i^{u+2w-2})$$
$$(\mathsf{RI}_i^1, \mathbb{L}^{u+2w}) \leftarrow \mathsf{RS.RedInf}(pp, \mathsf{vk}, \mathsf{rk}[i], M^1, \mathsf{ADM}^1, \mathsf{DID}^1, \sigma^1, \mathsf{MOD}^1, \mathbb{L}_i^{u+2w-1}).$$

(c) For $i = 1, ..., n$, compute

$$(M^{c\prime}, \mathsf{ADM}^c, \mathsf{DID}^c, \sigma^{c\prime}) \leftarrow \mathsf{RS.ThrRed}(pp, \mathsf{vk}, M^c, \mathsf{ADM}^c, \mathsf{DID}^c, \sigma^c, \{\mathsf{RI}_i^c\}_{i=1}^n).$$

(d) If $M^{0\prime} \neq M^{1\prime} \vee \mathsf{ADM}_0 \neq \mathsf{ADM}_1$, return $\perp$.

(e) Return $(M^{b\prime}, \mathsf{ADM}^b, \mathsf{DID}^b, \sigma^{b\prime})$. (b is chosen by C in step 1.)

5. A outputs b^*.

A t-out-of-n redactable signature scheme in the one-time redaction model (t, n)-RS satisfies the privacy security if for all PPT adversaries A, the following advantage $\mathsf{Adv}_{(t,n)\text{-}\mathsf{RS},\mathsf{A}}^{\mathsf{Priv\text{-}(t,n)\text{-}RSS}} = |\Pr[b = b^*] - 1/2|$ is negligible in λ.

Transparency. Transparency requires that except for a signer, n redactors, and a combiner, it is infeasible to distinguish whether a signature directly comes from the signer or has been redacted by redactors.

Definition 4.5 (Transparency)**.** The privacy of a t-out-of-n redactable signature scheme in the one-time redaction model (t, n)-RS Π is defined by the following weak privacy game between a challenger C and a PPT adversary $\mathcal{A}$.

1. C chooses a bit $b \xleftarrow{\$} \{0, 1\}$, runs $pp \leftarrow \mathsf{RS.Setup}(1^\lambda)$, $(\mathsf{vk}, \mathsf{sk}, \{\mathsf{rk}[i]\}_{i=1}^n) \leftarrow \mathsf{RS.KeyGen}(pp, t, n)$, and gives (pp, vk) to an adversary A.

2. A is given access (throughout the entire game) to a sign oracle $\mathcal{O}^{\mathsf{Sign}}(\cdot, \cdot)$ such that $\mathcal{O}^{\mathsf{Sign}}(M, \mathsf{ADM})$, returns $(M, \mathsf{ADM}, \mathsf{DID}, \sigma) \leftarrow \mathsf{RS.Sign}(pp, \mathsf{sk}, M, \mathsf{ADM})$.

3. A is given access (throughout the entire game) to a redact oracle $\mathcal{O}^{\mathsf{Redact}}(\cdot, \cdot, \cdot, \cdot, \cdot)$. $\mathcal{O}^{\mathsf{Redact}}$ is defined as follows:
For an u-th query $(M, \mathsf{ADM}, \mathsf{DID}, \sigma, \mathsf{MOD})$:
Let w be the number of queries to $\mathcal{O}^{\mathsf{Sign/Redact}}$ when A makes an u-th query to $\mathcal{O}^{\mathsf{Redact}}$.

(a) $(\mathsf{RI}_i, \mathbb{L}_i^{u+2w}) \leftarrow \mathsf{RS.RedInf}(pp, \mathsf{vk}, \mathsf{rk}[i], M, \mathsf{ADM}, \mathsf{DID}, \sigma, \mathsf{MOD}, \mathbb{L}_i^{u+2w-1})$ for $i = 1, ..., n$.

(b) $(M', \mathsf{ADM}, \mathsf{DID}, \sigma') \leftarrow \mathsf{RS.ThrRed}(pp, \mathsf{vk}, M, \mathsf{ADM}, \mathsf{DID}, \sigma, \{\mathsf{RI}_i\}_{i=1}^n)$.

(c) Return $(M', \mathsf{ADM}, \mathsf{DID}, \sigma')$.

4. A is given access (throughout the entire game) to a sign or redact oracle $\mathcal{O}^{\mathsf{Sign/Redact}}(\cdot,\cdot,\cdot)$. $\mathcal{O}^{\mathsf{Sign/Redact}}$ is defined as follows:
   For an w-th query $(M, \mathsf{ADM}, \mathsf{MOD})$:
   Let u be the number of queries to $\mathcal{O}^{\mathsf{Redact}}$ when A makes an w-th query to $\mathcal{O}^{\mathsf{Sign/Redact}}$.

   (a) Compute $(M, \mathsf{ADM}, \mathsf{DID}_0, \sigma) \leftarrow \mathsf{RS.Sign}(pp, \mathsf{sk}, M, \mathsf{ADM})$.

   (b) For $i = 1, \ldots n$, compute

   $$(\mathsf{RI}_i, \mathbb{L}_i^{u+2w-1}) \leftarrow \mathsf{RS.RedInf}(pp, \mathsf{vk}, \mathsf{rk}[i], M, \mathsf{ADM}, \mathsf{DID}^0, \sigma, \mathsf{MOD}, \mathbb{L}_i^{u+2w-2}).$$

   (c) Compute $(M', \mathsf{ADM}, \mathsf{DID}^0, \sigma^0) \leftarrow \mathsf{RS.ThrRed}(pp, \mathsf{vk}, M, \mathsf{ADM}, \mathsf{DID}^0, \sigma, \{\mathsf{RI}_i\}_{i=1}^n)$.

   (d) Compute $(M', \mathsf{ADM}, \mathsf{DID}^1, \sigma^1) \leftarrow \mathsf{RS.Sign}(pp, \mathsf{sk}, M', \mathsf{ADM})$.

   (e) For $i = 1, \ldots n$, $\mathbb{L}_i^{u+2w} \leftarrow \mathbb{L}_i^{u+2w-1} \cup \{\mathsf{DID}^1\}$.

   (f) Return $(M', \mathsf{ADM}, \mathsf{DID}^b, \sigma^b)$.

5. A outputs b^*.

A t-out-of-n redactable signature scheme in the one-time redaction model (t,n)-RS Π satisfies the transparency security if for all PPT adversaries A, the following advantage $\mathsf{Adv}^{\mathsf{Tran\text{-}(t,n)\text{-}RSS}}_{(t,n)\text{-}\mathsf{RS},\mathsf{A}} = |\Pr[b = b^*] - 1/2|$ is negligible in λ.

Theorem 4.6. If t-out-of-n redactable signature scheme in the one-time redaction model (t,n)-RS Π satisfies transparency, then it satisfies privacy.

We prove Theorem 4.6 in a similar way of [13, 21].

proof. Assume that PPT adversary $\mathsf{A}^{\mathsf{Priv}}$ that wins the privacy game with probability $1/2 + \epsilon_{\mathsf{Priv}}$ where ϵ_{Priv} is non-negligible in λ. Let $\mathsf{C}^{\mathsf{Tran}}$ be the challenger in transparency game. Now we construct a PPT adversary $\mathsf{B}^{\mathsf{Tran}}$ that wins the transparency game with probability $1/2 + \epsilon_{\mathsf{Priv}}/2$ using $\mathsf{A}^{\mathsf{Priv}}$. The operation of $\mathsf{B}^{\mathsf{Tran}}$ is following.

- $\mathsf{B}^{\mathsf{Tran}}$ receives (pp, vk) from $\mathsf{C}^{\mathsf{Tran}}$, chooses a bit $c \leftarrow \{0, 1\}$ and sends vk to $\mathsf{A}^{\mathsf{Priv}}$.

- For each query (M, ADM) of $\mathsf{A}^{\mathsf{Priv}}$ to $\mathcal{O}^{\mathsf{Sign}}$, $\mathsf{B}^{\mathsf{Tran}}$ queries (M, ADM) to $\mathcal{O}^{\mathsf{Sign}}$ and gets $(M, \mathsf{ADM}, \mathsf{DID}, \sigma)$ and sends it to $\mathsf{A}^{\mathsf{Priv}}$.

- For each query $(M^0, \mathsf{ADM}^0, \mathsf{MOD}^0, M^1, \mathsf{ADM}^1, \mathsf{MOD}^1)$ of $\mathsf{A}^{\mathsf{Priv}}$ to $\mathcal{O}^{\mathsf{LoRredact}}$, $\mathsf{B}^{\mathsf{Tran}}$ checks $M^{0\prime} = M^{1\prime}$ where $M^{0\prime} = M^0/\mathsf{MOD}^0$ and $M^{1\prime} = M^1/\mathsf{MOD}^1$. If so, $\mathsf{B}^{\mathsf{Tran}}$ queries $(M^c, \mathsf{ADM}^c, \mathsf{MOD}^c)$ to $\mathcal{O}^{\mathsf{Sign/Redact}}$ and $\mathsf{B}^{\mathsf{Tran}}$ returns its result to $\mathsf{A}^{\mathsf{Priv}}$. Otherwise, $\mathsf{B}^{\mathsf{Tran}}$ returns $\perp$ to $\mathsf{A}^{\mathsf{Priv}}$.

- $\mathsf{B}^{\mathsf{Tran}}$ receives a guess b^* from $\mathsf{A}^{\mathsf{Priv}}$. If $b^* = c$, $\mathsf{B}^{\mathsf{Tran}}$ outputs 0, otherwise $\mathsf{B}^{\mathsf{Tran}}$ outputs 1.

If $b = 0$, $\mathcal{O}^{\mathsf{Sign/Redact}}$ always redacts and the view of $\mathsf{A}^{\mathsf{Priv}}$ is the same as in the privacy game. However, if $b = 1$, each signature is fresh and the output of $\mathsf{A}^{\mathsf{Priv}}$ is useless to win the transparency game. Therefore, the win probability of $\mathsf{B}^{\mathsf{Tran}}$ in transparency game is $\epsilon_{\mathsf{Tran}} = 1/2(1/2 + \epsilon_{\mathsf{Priv}}) + 1/2 \cdot 1/2 = 1/2 + \epsilon_{\mathsf{Priv}}/2$. Therefore, the advantage of $\mathsf{B}^{\mathsf{Tran}}$ in transparency game is non-negligible in λ. □

4.3 BGLS Aggregate Signature Scheme

Boneh, Gentry, Lynn, and Shacham [11] proposed the aggregate signature scheme which is based on the Boneh-Lynn-Shacham (BLS) signature scheme [12]. Our construction of t-out-of-n redactable signature is based on the BGLS aggregate signature scheme. Here, we review the the BGLS aggregate signature scheme $\mathsf{AS_{BGLS}} = (\mathsf{AS_{BGLS}.Setup}, \mathsf{AS_{BGLS}.KeyGen}, \mathsf{AS_{BGLS}.Sign}, \mathsf{AS_{BGLS}.Verify}, \mathsf{AS_{BGLS}.Aggregate}, \mathsf{AS_{BGLS}.AggVerify})$ is given as follows.*

- $\mathsf{AS_{BGLS}.Setup}(1^\lambda)$:

  1. $\mathcal{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathsf{G}(1^\lambda)$, $g_1 \xleftarrow{\$} \mathbb{G}_1^*$, $g_2 \xleftarrow{\$} \mathbb{G}_2^*$. ($\mathsf{G}$ is a type-2 pairing-group generator)
  2. Choose hash functions: $H : \{0,1\}^* \rightarrow \mathbb{G}_1$.
  3. Return $pp \leftarrow (\mathcal{G}, g_1, g_2, H)$.

- $\mathsf{AS_{BGLS}.KeyGen}(pp)$:

  1. $x \xleftarrow{\$} \mathbb{Z}_p$, $X \leftarrow g_2^x$.
  2. Return $(\mathsf{vk}, \mathsf{sk}) \leftarrow (X, x)$.

- $\mathsf{AS_{BGLS}.Sign}(pp, \mathsf{sk}, m)$:

*If we remove two algorithms $\mathsf{AS_{BGLS}.Aggregate}$ and $\mathsf{AS_{BGLS}.AggVerify}$ from $\mathsf{AS_{BGLS}}$, then this scheme correspond to the BLS signature scheme.

1. $h \leftarrow H(m)$, $\sigma \leftarrow h^{\mathsf{sk}}$.
2. Return σ.

- $\mathsf{AS}_{\mathsf{BGLS}}.\mathsf{Verify}(pp, \mathsf{vk}, m, \sigma)$:

  1. $h \leftarrow H(m)$.
  2. If $e(\sigma, g_2) = e(h, \mathsf{vk})$, return 1.
  3. Otherwise return 0.

- $\mathsf{AS}_{\mathsf{BGLS}}.\mathsf{Aggregate}(pp, (\mathsf{vk}_1, \ldots, \mathsf{vk}_r), (m_1, \ldots, m_r), (\sigma_1, \ldots, \sigma_r))$:

  1. If there exists $(i, j) \in [r] \times [r]$ such that $i \neq j \wedge m_i = m_j$, return $\perp$.
  2. If there exists $i \in [r]$ suth that $\mathsf{AS}_{\mathsf{BGLS}}.\mathsf{Verify}(pp, \mathsf{vk}_i, m_i, \sigma_i) \neq 0$, return $\perp$.
  3. $\Sigma \leftarrow \prod_{i=1}^{r} \sigma_i$.
  4. Return Σ.

- $\mathsf{AS}_{\mathsf{BGLS}}.\mathsf{AggVerify}(pp, (\mathsf{vk}_1, \ldots, \mathsf{vk}_r), (m_1, \ldots, m_r), \Sigma)$:

  1. If there exists $i \in [r]$ suth that $\mathsf{AS}_{\mathsf{BGLS}}.\mathsf{Verify}(pp, \mathsf{vk}_i, m_i, \sigma_i) \neq 0$, return 0.
  2. For $i = 1$ to r, $h_i \leftarrow H(m_i)$.
  3. If $e(\Sigma, g_2) = \prod_{i=1}^{r} e\left(H_i, \mathsf{vk}_i\right)$, return 1.
  4. Otherwise, return 0.

Boneh et al prove that the EUF-CMA security of [11] the BGLS aggregate signature scheme under the co-CDH assumption in the ROM.

4.4 Shamir's Secret Sharing Scheme

In order to construct a t-out-of-n redactable signature scheme, we use the (t, n)-Shamir's secret sharing scheme [56]. The (t, n)-secret sharing scheme is composed of a dealer and n users. The dealer decides a secret s, computes secret shares $\{s_i\}_{i=1}^{n}$, and gives the secret share s_i to the user i. If any t of n secret shares or more shares are collected, we can reconstruct the secret s from them. While, with less than t secret shares, we cannot recover the secret s.

We refer to the (t, n)-shamir's secret sharing scheme in [19].

1. The dealer chooses the secret $s \in \mathbb{Z}$ and sets $a_0 \leftarrow s$.

2. The dealer chooses $a_1, \cdots, a_{t-1} \in \{0, \cdots, p-1\}$ independently at random and gets the polynomial $f(X) = \sum_{i=0}^{t-1} a_i X^i$.

3. The dealer computes $f(i)$, sets $s_i \leftarrow (i, f(i))$, and sends the secret share s_i to the user i.

If we collect t or more secret shares, we can reconstruct the secret s by the Lagrange interpolation. Let $J \subset \{1, \cdots, n\}$ and $|J| = t$. If we have secret shares $\{s_j\}_{j\in J} = \{(j, f(j))\}_{j\in J}$, we can compute $s = \sum_{i\in J} \Big(f(i) \prod_{j\in J, j\neq i} j(j-i)^{-1} \Big)$.

4.5 Our Construction

We give a concrete construction of t-out-of-n redactable signature scheme in one-time redaction model (t,n)-RS Π_1. Let ℓ, d be polynomials in λ and M a message having a set data structure (i.e., $M = \{m_1, ..., m_\ell\}$) and $\#M \leq \ell$.

$\mathsf{RS.Setup}(1^\lambda)$:

1. Run $\mathcal{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathsf{G}(1^\lambda)$.

2. Choose $g_1 \xleftarrow{\$} \mathbb{G}_1^*$, $g_2 \xleftarrow{\$} \mathbb{G}_2^*$.

3. Choose a hash function $H : \{0,1\}^* \rightarrow \mathbb{G}_1$.

4. Return $pp = (\mathcal{G}, g_1, g_2, H)$

$\mathsf{RS.KeyGen}(pp, t, n)$:

1. Choose $\tilde{x} \xleftarrow{\$} \mathbb{Z}_q$, compute $\tilde{y} \leftarrow g_2^{\tilde{x}}$, and set $(\mathsf{vk}_{\mathsf{Fix}}, \mathsf{sk}_{\mathsf{Fix}}) \leftarrow (\tilde{y}, \tilde{x})$.

2. Choose $a_0, a_1, \cdots, a_{t-1} \xleftarrow{\$} \mathbb{Z}_q$ independently at random and gets the polynomial $f(X) = \sum_{i=0}^{t-1} a_i X^i$.

3. For $i = 0$ to n, compute $x_i \leftarrow f(i),\ y_i \leftarrow g_2^{f(i)}$.

4. Set $(\mathsf{vk}_{\mathsf{Agg}}, \mathsf{sk}_{\mathsf{Agg}}) \leftarrow (y_0, x_0)$, $\mathsf{rk}[i] \leftarrow (i, x_i)$ for all $i \in [n]$.

5. Set $(\mathsf{vk}, \mathsf{sk}) \leftarrow ((\mathsf{vk}_{\mathsf{Fix}}, \mathsf{vk}_{\mathsf{Agg}}, t, n), (\mathsf{sk}_{\mathsf{Fix}}, \mathsf{sk}_{\mathsf{Agg}}))$.

6. Return $(\mathsf{vk}, \mathsf{sk}, \{\mathsf{rk}[i]\}_{i=1}^n)$.

$\mathsf{RS.Sign}(pp, \mathsf{sk}, M, \mathsf{ADM})$: (Note that ADM is a set containing all blocks which must not be redacted.):

1. Parse sk as $(\mathsf{sk}_{\mathsf{Fix}}, \mathsf{sk}_{\mathsf{Agg}})$.
2. If $\mathsf{ADM} \not\preceq M$, (i.e., $\mathsf{ADM} \cap M \neq \mathsf{ADM}$.) then abort.
3. Choose document ID $\mathsf{DID} \xleftarrow{\$} \{0,1\}^d$.
4. Compute $h_{\mathsf{ADM}} \leftarrow H(\mathsf{DID}||\mathrm{ord}(\mathsf{ADM}))$.
   $\mathrm{ord}(\mathsf{ADM})$ denotes a lexicographic ordering to the elements in ADM.
5. For $m_j \in M$, compute $h_{m_j} \leftarrow H(\mathsf{DID}||m_j)$.
6. Compute $\sigma_{\mathsf{Fix}} \leftarrow h_{\mathsf{ADM}}^{\mathsf{sk}_{\mathsf{Fix}}}$.
7. Compute $\sigma_{\mathsf{ADM}} \leftarrow h_{\mathsf{ADM}}^{\mathsf{sk}_{\mathsf{Agg}}}$, $\sigma_{m_j} \leftarrow h_{m_j}^{\mathsf{sk}_{\mathsf{Agg}}}$ for $m_j \in M$.
8. Compute $\Sigma_{\mathsf{agg}} \leftarrow \sigma_{\mathsf{ADM}} \cdot \prod_{m_j \in M} \sigma_{m_j}$.
9. Set $\sigma \leftarrow (\sigma_{\mathsf{Fix}}, \Sigma_{\mathsf{agg}})$.
10. Return $(M, \mathsf{ADM}, \mathsf{DID}, \sigma)$.

$\mathsf{RS.Redact}$: $\mathsf{RS.Redact}$ is an interactive protocol between the combiner and n redactor. The combiner interacts with the n redactors and finally outputs the redacted signature.

1. Each redactor i selects a modifiction instruction MOD_i. Let $\mathbb{L}_i$ be the list which stores DIDs, $\mathbb{L}_i^0 = \emptyset$, and $\mathbb{L}_i^{u-1}$ the list which used in the input of u-th running of the PPT algorithm $\mathsf{RS.RedInf}$ by the redactor i.
   The redactor i runs $\mathsf{RS.RedInf}(pp, \mathsf{vk}, \mathsf{rk}[i], M, \mathsf{ADM}, \mathsf{DID}, \sigma, \mathsf{MOD}_i, \mathbb{L}_i^{u-1})$.

   $\mathsf{RS.RedInf}(pp, \mathsf{vk}, \mathsf{rk}[i], M, \mathsf{ADM}, \mathsf{DID}, \sigma, \mathsf{MOD}_i, \mathbb{L}_i^{u-1})$:

   (a) Parse vk as $(\mathsf{vk}_{\mathsf{Fix}}, \mathsf{vk}_{\mathsf{Agg}}, t, n)$ and σ as $(\sigma_{\mathsf{Fix}}, \Sigma_{\mathsf{agg}})$.
   (b) If $\mathsf{DID} \in \mathbb{L}_i^{u-1}$ then abort.
   (c) Update $\mathbb{L}_i^u \leftarrow \mathbb{L}_i^{u-1} \cup \{\mathsf{DID}\}$.
   (d) Check $\mathsf{MOD}_i \overset{\mathsf{ADM}}{\preceq} M$. (i.e., $\mathsf{MOD}_i \cap \mathsf{ADM} = \emptyset \wedge \mathsf{MOD}_i \subset M$.)

(e) Compute $h_{\mathsf{ADM}} \leftarrow H(\mathsf{DID}||\mathrm{ord}(\mathsf{ADM}))$.
ord(ADM) denotes a lexicographic ordering to the elements in ADM.

(f) For $m_j \in M$, compute $h_{m_j} \leftarrow H(\mathsf{DID}||m_j)$.

(g) If $e(\sigma_{\mathsf{Fix}}, g_2) \neq e(h_{\mathsf{ADM}}, \mathsf{vk}_{\mathsf{Fix}})$ then abort.

(h) If $e(\Sigma_{\mathsf{agg}}, g_2) \neq e\left(h_{\mathsf{ADM}}, \mathsf{vk}_{\mathsf{Agg}}\right) \cdot \prod_{m_j \in M} e(h_{m_j}, \mathsf{vk}_{\mathsf{Agg}})$ then abort.

(i) For $m_j \in \mathsf{MOD}_i$, compute $\mathsf{RI}_{i,m_j} \leftarrow h_{m_j}^{\mathsf{rk}[i]}$.

(j) For $m_j \notin \mathsf{MOD}_i$, set $\mathsf{RI}_{i,m_j} \leftarrow \emptyset$.

(k) Set a redaction information RI_i of redactor i as $\mathsf{RI}_i \leftarrow \{\mathsf{RI}_{i,m_j}\}_{m_j \in M}$

(l) Output $(\mathsf{RI}_i, \mathbb{L}_i^u)$.

For one DID, redactor i runs RS.RedInf only once. This can be done by introducing a table $\mathbb{L}_i$.

2. Each redactor i sends (i, RI_i) to the combiner.

3. The combiner collects all n redaction information $\{\mathsf{RI}_i\}_{i=1}^n$.

4. The combiner runs the PPT algorithm $\mathsf{RS.ThrRed}(pp, \mathsf{vk}, M, \mathsf{ADM}, \mathsf{DID}, \sigma, \{\mathsf{RI}_i\}_{i=1}^n)$.

$\mathsf{RS.ThrRed}(\mathsf{vk}, M, \mathsf{ADM}, \mathsf{DID}, \sigma, \{\mathsf{RI}_i\}_{i=1}^n)$:

(a) Parse vk as $(\mathsf{vk}_{\mathsf{Fix}}, \mathsf{vk}_{\mathsf{Agg}}, t, n)$ and σ as $(\sigma_{\mathsf{Fix}}, \Sigma_{\mathsf{agg}})$.

(b) Parse RI_i as $\{\mathsf{RI}_{i,m_j}\}_{m_j \in M}$.

(c) For $m_j \in M$, define $\mathsf{RI}_{m_j} = \{\mathsf{RI}_{i,m_j}\}_{i=1}^n$.

(d) Define $\mathsf{MOD} = \{m_j | m_j \in M \wedge \#\mathsf{RI}_{m_j} \geq t\}$

(e) For $m_j \in \mathsf{MOD}$, define $\mathsf{InRI}_{m_j} \leftarrow \{i \in \mathbb{N} | \{\mathsf{RI}_{i,m_j}\} \neq \emptyset\}$.

(f) For $m_j \in \mathsf{MOD}$, choose subset $J_{m_j} \subset \mathsf{InRI}_{m_j}$ such that $\#J_{m_j} = t$.

(g) For $m_j \in \mathsf{MOD}$, compute $\sigma_{m_j} \leftarrow \prod_{i \in J_{m_j}} \left(\mathsf{RI}_{i,m_j}\right)^{\gamma_{i,J_{m_j}}}$,
where $\gamma_{i,J_{m_j}} = \prod_{j \in J_{m_j}, j \neq i} j(j-i)^{-1}$.

(h) Compute $\sigma_{\mathsf{MOD}} \leftarrow \prod_{m_j \in \mathsf{MOD}} \sigma_{m_j}$, $\Sigma'_{\mathsf{agg}} \leftarrow \Sigma_{\mathsf{agg}} / \sigma_{\mathsf{MOD}}$.

(i) Set $M' \leftarrow M \backslash \{\mathsf{MOD}\}$, $\sigma' \leftarrow (\sigma_{\mathsf{Fix}}, \Sigma'_{\mathsf{agg}})$.

(j) Return $(M', \mathsf{ADM}, \mathsf{DID}, \sigma')$.

5. The combiner outputs $(M', \mathsf{ADM}, \mathsf{DID}, \sigma')$.

$\mathsf{RS.Verify}(pp, \mathsf{vk}, M, \mathsf{ADM}, \mathsf{DID}, \sigma)$:

1. Parse vk as $(\mathsf{vk}_{\mathsf{Fix}}, \mathsf{vk}_{\mathsf{Agg}}, t, n)$ and σ as $(\sigma_{\mathsf{Fix}}, \Sigma_{\mathsf{agg}})$.

2. If $\mathsf{ADM} \cap M \neq \mathsf{ADM}$, return 0.

3. Compute $h_{\mathsf{ADM}} \leftarrow H(\mathsf{DID}||\mathrm{ord}(\mathsf{ADM}))$.
   $\mathrm{ord}(\mathsf{ADM})$ denotes a lexicographic ordering to the elements in ADM.

4. For $m_j \in M$, compute $h_{m_j} \leftarrow H(\mathsf{DID}||m_j)$.

5. If $e(\sigma_{\mathsf{Fix}}, g_2) \neq e(h_{\mathsf{ADM}}, \mathsf{vk}_{\mathsf{Fix}})$, return 0

6. If $e(\Sigma_{\mathsf{agg}}, g_2) = e\left(h_{\mathsf{ADM}}, \mathsf{vk}_{\mathsf{Agg}}\right) \cdot \prod_{m_j \in M} e(h_{m_j}, \mathsf{vk}_{\mathsf{Agg}})$, return 1. Otherwise output 0.

Correctness

If $pp \leftarrow \mathsf{RS.Setup}(1^\lambda)$, $\mathsf{RS.KeyGen}(pp, t, n)$, and $(M, \mathsf{ADM}, \mathsf{DID}, \sigma)$ is honestly generated by the $\mathsf{RS.Sign}$ and has not been processed by the $\mathsf{RS.Redact}$ protocol, $\mathsf{RS.Verify}(M, \mathsf{ADM}, \mathsf{DID}, \sigma) = 1$ always holds. If $(M, \mathsf{ADM}, \mathsf{DID}, \sigma)$ is honestly generated the $\mathsf{RS.Sign}$ and $(M', \mathsf{ADM}, \mathsf{DID}, \sigma')$ is honestly redacted from $(M, \mathsf{ADM}, \mathsf{DID}, \sigma)$ by $\mathsf{RS.Redact}$ protocol, $(M', \mathsf{ADM}, \mathsf{DID}, \sigma')$ passes the verification in the $\mathsf{RS.Verify}$. Therefore, our construction of t-out-of-n redactable signature scheme in the one-time redaction model satisfies correctness.

4.6 Security Proof for Unforgeability

Overview of Unforgeability Security Proof. Before describing unforgeability security proof for our proposed scheme, we explain the outline of the proof. For convenience of our security proof, we introduce new notations. Let q_s be the total number of queries from an adversary to $\mathcal{O}^{\mathsf{Sign}}$, (M_i, ADM_i) an i-th input for $\mathcal{O}^{\mathsf{Sign}}$, $(M^i, \mathsf{ADM}^i, \mathsf{DID}^i, \sigma^i)$ the i-th output of $\mathcal{O}^{\mathsf{Sign}}$. We denote

$$Q_{\mathsf{Sign}} := \bigcup_{i=1}^{q_s} \{(M^i, \mathsf{ADM}^i, \mathsf{DID}^i)\}, \quad Q_{\mathsf{Sign}}^{\mathsf{AD}} := \bigcup_{i=1}^{q_s} \{(\mathsf{ADM}^i, \mathsf{DID}^i)\}.$$

Also, let q_r be the total number of queries from an adversary to $\mathcal{O}^{\mathsf{Redact}}$, $(M^i, \mathsf{ADM}^i, \mathsf{DID}^i, \sigma^i, \mathsf{MOD}^i)$ an i-th input for $\mathcal{O}^{\mathsf{Redact}}$, $(M'^i, \mathsf{ADM}^i, \mathsf{DID}^i, \sigma'^i)$ the i-th output of $\mathcal{O}^{\mathsf{Redact}}$. We denote

$$Q_{\mathsf{Redact}} := \bigcup_{i=1}^{q_r} \{(M'^i, \mathsf{ADM}^i, \mathsf{DID}^i)\}, \quad Q^{\mathsf{AD}}_{\mathsf{Redact}} := \bigcup_{i=1}^{q_r} \{(\mathsf{ADM}^i, \mathsf{DID}^i)\}.$$

We assume the following three types of PPT adversaries that breaks the unforgeability security in our proposed scheme.

- An adversary A_1 that outputs a forgery $(M^*, \mathsf{DID}^*, \mathsf{ADM}^*, \sigma^*)$ such that $(\mathsf{ADM}^*, \mathsf{DID}^*) \notin (Q^{\mathsf{AD}}_{\mathsf{Sign}} \cup Q^{\mathsf{AD}}_{\mathsf{Redact}})$.
- An adversary A_2 that outputs a forgery $(M^*, \mathsf{DID}^*, \mathsf{ADM}^*, \sigma^*)$ which satisfies $(\mathsf{ADM}^*, \mathsf{DID}^*) \in (Q^{\mathsf{AD}}_{\mathsf{Sign}} \cup Q^{\mathsf{AD}}_{\mathsf{Redact}})$. Moreover, there is $\tilde{M}$ such that $(\tilde{M}, \mathsf{ADM}^*, \mathsf{DID}^*) \in (Q_{\mathsf{Sign}} \cup Q_{\mathsf{Redact}})$.
- An adversary A_3 that outputs a forgery $(M^*, \mathsf{DID}^*, \mathsf{ADM}^*, \sigma^*)$ which satisfies $(\mathsf{ADM}^*, \mathsf{DID}^*) \in (Q^{\mathsf{AD}}_{\mathsf{Sign}} \cup Q^{\mathsf{AD}}_{\mathsf{Redact}})$. Moreover, there are no $\tilde{M}$ such that $(\tilde{M}, \mathsf{ADM}^*, \mathsf{DID}^*) \in (Q_{\mathsf{Sign}} \cup Q_{\mathsf{Redact}})$ and $\tilde{M} \nsubseteq M$.

To prove the theorem, for each A_i, we consider a sequential of games from the original unforgeability game to game which is directly related to solving the co-CDH problem. Then, We construct B_i which breaking the co-CDH assumption using A_i. B_1 breaks the co-CDH assumption using the forgery σ^*_{Fix}. In the case of B_2 and B_3, they use the forgery Σ^*_{agg} to break the co-CDH assumption. One difference between B_2 and B_3 is how to program the hash value.

Theorem 4.7. In the random oracle model, if the computational co-Diffie-Hellman problem assumption holds, then our proposed t-out-of-n redactable signature scheme in the one-time redaction model (t, n)-RS Π_1 satisfies the unforgeability property.

proof. We consider three types of adversary described above.

Case 1**:**

We consider an adversary A_1 that can generate a valid forgery with ϵ_{Uf1} against our proposal redactable signature scheme. Let $\mathbf{Game}_{1-0}$ be the original unforgeability game in a redactable signature scheme and $\mathbf{Game}_{1-5}$ be directly related to solving the computational co-Diffie-Hellman problem. Define $\mathsf{Adv}_{\mathsf{A}_1}[\mathbf{Game}_{1-X}]$ as the advantage of an adversary A_1 in $\mathbf{Game}_{1-X}$.

- **Game**$_{1-0}$: Original unforgeability game in a redactable signature scheme.

$$\mathsf{Adv}_{\mathsf{A}_1}[\mathbf{Game}_{1-0}] = \epsilon_{\mathsf{Uf1}}$$

- **Game**$_{1-1}$: We change a key generation algorithm $\mathsf{RS.KeyGen}$ in Step 1.
Choose $\tilde{x} \xleftarrow{\$} \mathbb{Z}_q$, $\tilde{r} \xleftarrow{\$} \mathbb{Z}_q$ and compute $u \leftarrow g_2^{\tilde{x}}$, $\tilde{y} \leftarrow g_2^{\tilde{x}+\tilde{r}}$.
Set $(\mathsf{vk_{Fix}}, \mathsf{sk_{Fix}}) \leftarrow (\tilde{y}, \tilde{x}+\tilde{r})$.

- **Game**$_{1-2}$:We change a setting of the random oracle $\mathcal{O}^H$. Fix $h \xleftarrow{\$} \mathbb{G}_2$ and let $\mathbb{T}$ be a table that maintains a list of tuples $\langle v, w, b, c\rangle$ as explain below. We refer to this list for the query to $\mathcal{O}^h$. The initial state of $\mathbb{T}$ is empty. For queries $v^{(i)}$ to $\mathcal{O}^H$:

  - If $\langle v^{(i)}, w^{(i)}, \cdot, \cdot\rangle$ (Here, '$\cdot$' represents an arbitrary value) already appears in $\mathbb{T}$, then return $w^{(i)}$.
  - Choose $s^{(i)} \xleftarrow{\$} \mathbb{Z}_q$.
  - Flip a biased coin $c^{(i)} \in \{0, 1\}$ such that $\Pr[c^{(i)} = 0] = 1 - 1/(q_s + 1)$ and $\Pr[c^{(i)} = 1] = 1/(q_s + 1)$.
  - If $c^{(i)} = 0$, compute $w^{(i)} = \phi(g_2)^{b^{(i)}}$.
  - If $c^{(i)} = 1$, compute $w^{(i)} = h \cdot \phi(g_2)^{b^{(i)}}$.
  - Insert $\langle v^{(i)}, w^{(i)}, s^{(i)}, c^{(i)}\rangle$ in $\mathbb{T}$ and return $w^{(i)}$.

- **Game**$_{1-3}$: We modify the signing algorithm $\mathsf{RS.Sign}$ in Step 4 as follows:

  - Set $v^{(0)} \leftarrow (\mathsf{DID}||\mathrm{ord}(\mathsf{ADM}))$.
  - Query $v^{(0)}$ to $\mathcal{O}^H$. We assume $\langle v^{(0)}, w^{(0)}, b^{(0)}, c^{(0)}\rangle$ to be the tuple in $\mathbb{T}$ for $v^{(0)}$.
  - If $c^{(0)} = 1$, return $\perp$ and abort.

- **Game**$_{1-4}$: We modify the signing algorithm $\mathsf{RS.Sign}$ in Step 6 as follows:

  - Compute $\sigma_{\mathsf{Fix}} \leftarrow \phi(u)^{b^{(0)}} \cdot \phi(g_2)^{\tilde{r}b^{(0)}}$.

  (A signature σ_{Fix} can be generated without a knowledge of $\mathsf{sk_{Fix}}$.)

- **Game**$_{1-5}$: We receive a valid forgery $(M^*, \mathsf{ADM}^*\mathsf{DID}^*, \sigma^*)$ from the adversary A_1, we operate as follows:

  - Set $v^{(0)} \leftarrow (\mathsf{DID}^*||\mathrm{ord}(\mathsf{ADM}^*))$.

- Query $v^{(0)}$ to $\mathcal{O}^H$. We assume $\langle v^{(0)}, w^{(0)}, s^{(0)}, c^{(0)} \rangle$ to be the tuple in $\mathbb{T}$ for each $v^{(0)}$.
- If $c^{(0)} = 0$, then abort.

Lemma 4.8. The following equation holds.

$$\mathsf{Adv}_{\mathsf{A}_1}[\mathbf{Game}_{1-1}] = \mathsf{Adv}_{\mathsf{A}_1}[\mathbf{Game}_{1-0}].$$

Since the distribution of $(\mathsf{vk}_{\mathsf{Fix}}, \mathsf{sk}_{\mathsf{Fix}})$ in $\mathbf{Game}_{1-0}$ and $\mathbf{Game}_{1-1}$ are same.

Lemma 4.9. If H is the random oracle model, the following eqauation holds.

$$\mathsf{Adv}_{\mathsf{A}_1}[\mathbf{Game}_{1-2}] = \mathsf{Adv}_{\mathsf{A}_1}[\mathbf{Game}_{1-1}]$$

Since the distribution of outputs of $\mathcal{O}^H$ in $\mathbf{Game}_{1-1}$ and $\mathbf{Game}_{1-2}$ are identical.

Lemma 4.10. The following inequality holds.

$$\mathsf{Adv}_{\mathsf{A}_1}[\mathbf{Game}_{1-3}] \geq (1 - 1/(q_s + 1))^{q_s} \times \mathsf{Adv}_{\mathsf{A}_1}[\mathbf{Game}_{1-2}].$$

Since the probability that each signing query does not abort at least $1 - 1/(q_s + 1)$.

Lemma 4.11. The following equation holds.

$$\mathsf{Adv}_{\mathsf{A}_1}[\mathbf{Game}_{1-4}] = \mathsf{Adv}_{\mathsf{A}_1}[\mathbf{Game}_{1-3}].$$

Since outputs of Sign in $\mathbf{Game}_{1-3}$ and $\mathbf{Game}_{1-4}$ are same.

Lemma 4.12. The following inequality holds.

$$\mathsf{Adv}_{\mathsf{A}_1}[\mathbf{Game}_{1-5}] \geq (1/(q_s + 1)) \times \mathsf{Adv}_{\mathsf{A}_1}[\mathbf{Game}_{1-4}].$$

Since the probability that the forged signature satisfies $c^{(0)} = 1$ at least $1/(q_s + 1)$.

To summarize from Lemma 4.8 to Lemma 4.12, the following holds.
(In the following equation, e represents the Napier's constant.)

$$\begin{aligned} \mathsf{Adv}_{\mathsf{A}_1}[\mathbf{Game}_{1-5}] &\geq (1 - 1/(q_s + 1))^{q_s} \times (1/(q_s + 1)) \times \mathsf{Adv}_{\mathsf{A}_1}[\mathbf{Game}_{1-0}] \\ &\geq (1/e) \times (1/(q_s + 1)) \times \mathsf{Adv}_{\mathsf{A}_1}[\mathbf{Game}_{1-0}] \end{aligned}$$

Now we construct the algorithm B_1 which breaking the computational co-Diffie-Hellman assumption using the algorithm A_1. The operation of B_1 for the input co-Diffie-Hellman problem instance (g_2, g_2^{α}, h^*) is changed to h to h^* and u to g_2^{α} in $\mathbf{Game}_{1-5}$. Suppose B_1 does not abort receiving a forgery $(M^*, \mathsf{ADM}^*, \mathsf{DID}^*, \sigma^*)$ from A_1.

B_1 parses σ^* as $(\sigma^*_{\mathsf{Fix}^*}, \Sigma^*_{\mathsf{agg}})$, sets $v^{(0)} \leftarrow (\mathsf{DID}^*||\mathrm{ord}(\mathsf{ADM}^*))$ and computes $w^{(0)} \leftarrow h^* \cdot \phi(g_2)^{b^{(0)}}$. Since $(M^*, \mathsf{ADM}^*, \mathsf{DID}^*, \sigma^*)$ is valid and $\mathsf{vk}_{\mathsf{Fix}} = g_2^{\alpha+\tilde{r}}$, $e(\sigma^*_{\mathsf{Fix}}, g_2) = e((w^{(0)})^{\alpha+\tilde{r}}, g_2)$ holds. It implies that $\sigma^*_{\mathsf{Fix}} = (w^{(0)})^{\alpha+\tilde{r}} = (h^* \cdot \phi(g_2)^{b^{(0)}})^{\alpha+\tilde{r}}$. Therefore, B_1 computes $(h^*)^\alpha = \sigma^*_{\mathsf{Fix}} \cdot (\phi(u)^{b^{(0)}} \cdot (h^*)^{\tilde{r}} \cdot \phi(g_2)^{\tilde{r}b^{(0)}})^{-1}$ and outputs the solution $(h^*)^\alpha$ of the computational co-Diffie-Hellman problem instance (g_2, g_2^α, h^*).

Let $\epsilon_{\mathsf{co\text{-}cdh}}$ is the probability that B_1 break the computational co-Diffie-Hellman assumption. We can bound the probability $\epsilon_{\mathsf{co\text{-}cdh1}} \geq \mathsf{Adv}_{\mathsf{A}_1}[\textbf{Game } 1{-}5]$ and $\epsilon_{\mathsf{co\text{-}cdh1}} \geq (1/e) \times (1/q_s + 1) \times \epsilon_{\mathsf{uf1}}$ holds. (e represents the Napier's constant.) Hence, if ϵ_{uf1} is non-negligiable in λ, B_1 breaks the computational co-Diffie-Hellman assumption with non-negligiable in $\epsilon_{\mathsf{co\text{-}cdh1}}$.

Case 2:

We consider an adversary A_2 that can generate a valid forgery with ϵ_{uf2} against our proposal redactable signature scheme. Let $\textbf{Game}_{2-0}$ be the original unforgeability game in a redactable signature scheme and $\textbf{Game}_{2-6}$ be directly related to solve the computational co-Diffie-Hellman problem. Define $\mathsf{Adv}_{\mathsf{A}_2}[\textbf{Game}_{2-X}]$ as the advantage of an adversary A_2 in $\textbf{Game}_{2-X}$.

- $\textbf{Game}_{2-0}$: Original unforgeability game in a redactable signature scheme.

$$\mathsf{Adv}_{\mathsf{A}_2}[\textbf{Game}_{2-0}] = \epsilon_{\mathsf{uf2}}$$

- $\textbf{Game}_{2-1}$: We change a setting of $\mathcal{O}^{\mathsf{Redact}}$.
  We introduce a table $\mathbb{L}^u$ that store DIDs and $\mathbb{L}^0 = \emptyset$.
  For a u-th query $(M, \mathsf{ADM}, \mathsf{DID}, \sigma, \mathsf{MOD})$ to $\mathcal{O}^{\mathsf{Redact}}$:
  - Parse vk as $(\mathsf{vk}_{\mathsf{Fix}}, \mathsf{vk}_{\mathsf{Agg}}, t, n)$ and σ as $(\sigma_{\mathsf{Fix}}, \Sigma_{\mathsf{agg}})$.
  - If $\mathsf{DID} \in \mathbb{L}^{u-1}$, then abort.
  - Set $\mathbb{L}^u \leftarrow \mathbb{L}^{u-1} \cup \{\mathsf{DID}\}$.
  - If $\mathsf{MOD} \not\subseteq M \vee \mathsf{MOD} \cap \mathsf{ADM} \neq \emptyset$, then abort.
  - Compute $h_{\mathsf{ADM}} \leftarrow H(\mathsf{DID}||\mathrm{ord}(\mathsf{ADM}))$.
    $\mathrm{ord}(\mathsf{ADM})$ denotes a lexicographic ordering to the elements in ADM.
  - For $m_j \in M$, compute $h_{m_j} \leftarrow H(\mathsf{DID}||m_j)$.
  - If $e(\sigma_{\mathsf{Fix}}, g_2) \neq e(h_{\mathsf{ADM}}, \mathsf{vk}_{\mathsf{Fix}})$, then abort.

- If $e(\Sigma_{\mathsf{agg}}, g_2) \neq e\left(h_{\mathsf{ADM}}, \mathsf{vk}_{\mathsf{Agg}}\right) \cdot \prod_{m_j \in M} e(h_{m_j}, \mathsf{vk}_{\mathsf{Agg}})$, then abort.
- For $m_j \in \mathsf{MOD}$, compute $\sigma_{m_j} \leftarrow H(\mathsf{DID}||m_j)^{\mathsf{sk}_{\mathsf{Agg}}}$.
- Compute $\sigma_{\mathsf{MOD}} \leftarrow \prod_{m_j \in \mathsf{MOD}} \sigma_{m_j}$, $\Sigma'_{\mathsf{agg}} \leftarrow \Sigma_{\mathsf{agg}}/\sigma_{\mathsf{MOD}}$.
- Set $M' \leftarrow M \backslash \mathsf{MOD}$, $\sigma' \leftarrow (\sigma_{\mathsf{Fix}}, \Sigma'_{\mathsf{agg}})$.
- Return $(M', \mathsf{ADM}, \mathsf{DID}, \sigma')$.

(Redactions are done using $\mathsf{sk}_{\mathsf{Agg}}$ instead of using $\{\mathsf{rk}[i]\}_{i=1}^{n}$.)

- **Game**$_{2-2}$: We change settings of $\mathsf{RS.KeyGen}$ and $\mathcal{O}^{\mathsf{Corrupt}}$.
  - We change a key generation algorithm $\mathsf{RS.KeyGen}$ in Step 2 to 6.
    - Choose $x \xleftarrow{\$} \mathbb{Z}_q$, $r \xleftarrow{\$} \mathbb{Z}_q$, compute $u \leftarrow g^x$, $y \leftarrow g_2^{x+r}$.
    - Set $\mathsf{vk}_{\mathsf{Agg}} \leftarrow y$, $\mathsf{sk}_{\mathsf{Agg}} \leftarrow x + r$.
    - Return $(\mathsf{vk}, \mathsf{sk}) \leftarrow ((\mathsf{vk}_{\mathsf{Fix}}, \mathsf{vk}_{\mathsf{Agg}}, t, n), (\mathsf{sk}_{\mathsf{Fix}}, \mathsf{sk}_{\mathsf{Agg}}))$.

    (Redactor's keys $\{\mathsf{rk}[i]\}_{i=1}^{n}$ are not generated in the KeyGen.)
  - We change the setting of $\mathcal{O}^{\mathsf{Corrupt}}$ as follows:
    Let CR is a list to store a redactor's key information $(i, \mathsf{rk}[i])$
    For a query i to $\mathcal{O}^{\mathsf{Corrupt}}$,
    - If $(i, \mathsf{rk}[i])$ already appears in CR, then return $\mathsf{rk}[i]$.
    - Choose $f(i) \xleftarrow{\$} \mathbb{Z}_q$, set $CR \leftarrow CR \cup \{(i, f(i))\}$.
    - Return $\mathsf{rk}[i] \leftarrow (i, f(i))$.
- **Game**$_{2-3}$: We change a setting of the random oracle $\mathcal{O}^H$. Fix $h \xleftarrow{\$} \mathbb{G}_2$ and let $\mathbb{T}$ be a table that maintains a list of tuples $\langle v, w, b, c\rangle$ as explain below. We refer to this list for the query to $\mathcal{O}^h$. The initial state of $\mathbb{T}$ is empty. For queries $v^{(i)}$ to $\mathcal{O}^H$:
  - If $\langle v^{(i)}, w^{(i)}, \cdot, \cdot\rangle$ (Here, '$\cdot$' represents an arbitrary value) already appears in $\mathbb{T}$, then return $w^{(i)}$.
  - Choose $s^{(i)} \xleftarrow{\$} \mathbb{Z}_q$.
  - Flip a biased coin $c^{(i)} \in \{0, 1, 2\}$ such that such that $\Pr[c^{(i)} = 1] = 1 - 1/((\ell+1)(q_s + q_r)+1)$, $\Pr[c^{(i)} = 1] = 1/(2(\ell+1)(q_s+q_r)+2)$, $\Pr[c^{(i)} = 2] = 1/(2(\ell+1)(q_s+q_r)+2)$.
  - If $c^{(i)} = 0$, compute $w^{(i)} = \phi(g_2)^{b^{(i)}}$.
  - If $c^{(i)} = 1$, compute $w^{(i)} = h \cdot \phi(g_2)^{b^{(i)}}$.

– If $c^{(i)} = 2$, compute $w^{(i)} = h^{-1} \cdot \phi(g_2)^{b^{(i)}}$.

– Insert $\langle v^{(i)}, w^{(i)}, s^{(i)}, c^{(i)} \rangle$ in $\mathbb{T}$ and return $w^{(i)}$.

- **Game**$_{2-4}$:We modify the signing algorithm $\mathsf{RS.Sign}$ in Step 6 as follows:

  – Set $v^{(0)} \leftarrow (\mathsf{DID}||\mathrm{ord}(\mathsf{ADM}))$, $v^{(j)} \leftarrow (\mathsf{DID}||m_j)$ $(1 \leq j \leq \#M)$.

  – Query $v^{(j)}$ $(0 \leq j \leq \#M)$ to $\mathcal{O}^H$. We assume $\langle v^{(j)}, w^{(j)}, b^{(j)}, c^{(j)} \rangle$ to be the tuple in $\mathbb{T}$ for each $v^{(j)}$ $(1 \leq j \leq \#M)$.

  – If $c^{(0)} = 2$, $c^{(1)} = 1$, $c^{(j)} = 0$ $(2 \leq \forall j \leq \#M)$ or $c^{(j)} = 0$ $(0 \leq \forall j \leq \#M)$, go to Step 6 of Sign. Otherwise return $\perp$ and abort.

- **Game**$_{2-5}$: We modify the signing algorithm $\mathsf{RS.Sign}$ in Step 7, 8 as follows:

  – If $c^{(0)} = 2$, $c^{(1)} = 1$, $c^{(j)} = 0$ $(2 \leq \forall j \leq \#M)$,

    * Compute $\sigma_{\mathsf{ADM}m_1} \leftarrow \phi(u)^{b^{(0)}+b^{(1)}} \cdot \phi(g_2)^{r(b^{(0)}+b^{(1)})}$.
    * For all $m_j \in M \backslash \{m_1\}$, compute $\sigma_{m_j} \leftarrow \phi(u)^{b^{(j)}} \cdot \phi(g_2)^{rb^{(j)}}$.
    * Compute $\Sigma_{\mathsf{agg}} \leftarrow \sigma_{\mathsf{ADM}m_1} \cdot \prod_{m_j \in M \backslash \{m_1\}} \sigma_{m_j}$.

  – If $c^{(j)} = 0$ $(0 \leq \forall j \leq \#M)$,

    * Compute $\sigma_{\mathsf{ADM}} \leftarrow \phi(u)^{b^{(0)}} \cdot \phi(g_2)^{rb^{(0)}}$.
    * For all $m_j \in M$, compute $\sigma_{m_j} \leftarrow \phi(u)^{b^{(j)}} \cdot \phi(g_2)^{rb^{(j)}}$.
    * Compute $\Sigma_{\mathsf{agg}} \leftarrow \sigma_{\mathsf{ADM}} \cdot \prod_{m_j \in M} \sigma_{m_j}$.

  (By above modification, a signature Σ_{agg} can be generated without a knowledge of the $\mathsf{sk}_{\mathsf{Agg}}$.)

- **Game**$_{2-6}$: We change a setting of $\mathcal{O}^{\mathsf{Redact}}$.

  – Parse vk as $(\mathsf{vk}_{\mathsf{Fix}}, \mathsf{vk}_{\mathsf{Agg}}, t, n)$ and σ as $(\sigma_{\mathsf{Fix}}, \Sigma_{\mathsf{agg}})$.

  – If $\mathsf{DID} \in \mathbb{L}^{u-1}$, then abort.

  – Set $\mathbb{L}^u \leftarrow \mathbb{L}^{u-1} \cup \{\mathsf{DID}\}$.

  – If $\mathsf{MOD} \nsubseteq M \vee \mathsf{MOD} \cap \mathsf{ADM} \neq \emptyset$, then abort.

  – Set $v^{(0)} \leftarrow (\mathsf{DID}||\mathrm{ord}(\mathsf{ADM}))$, $v^{(j)} \leftarrow (\mathsf{DID}||m_j)$ $(1 \leq j \leq \#M)$.

  – Query $v^{(j)}$ $(0 \leq j \leq \#\mathsf{MOD})$ to $\mathcal{O}^H$. We assume $\langle v^{(j)}, w^{(j)}, b^{(j)}, c^{(j)} \rangle$ to be the tuple in $\mathbb{T}$ for each $v^{(j)}$ $(1 \leq j \leq \#\mathsf{MOD})$.

- If $e(\sigma_{\mathsf{Fix}}, g_2) \neq e(w^{(0)}, \mathsf{vk}_{\mathsf{Fix}})$, then abort.
- If $e(\Sigma_{\mathsf{agg}}, g_2) \neq \prod_{0 \leq j \leq \# M} e(w^{(j)}, \mathsf{vk}_{\mathsf{Agg}})$, then abort.
- If $c^{(j)} = 0$ ($\forall m_j \in \mathsf{MOD}$), go to next step. Otherwise return $\perp$ and abort.
- For all $m_j \in \mathsf{MOD}$, compute $\sigma_{m_j} \leftarrow \phi(u)^{b^{(j)}} \cdot \phi(g_2)^{rb^{(j)}}$.
- Compute $\sigma_{\mathsf{MOD}} \leftarrow \prod_{m_j \in \mathsf{MOD}} \sigma_{m_j}$, $\Sigma'_{\mathsf{agg}} \leftarrow \Sigma_{\mathsf{agg}} / \sigma_{\mathsf{MOD}}$.
- Set $M' \leftarrow M \backslash \mathsf{MOD}$, $\sigma' \leftarrow (\sigma_{\mathsf{Fix}}, \Sigma'_{\mathsf{agg}})$.
- Return $(M', \mathsf{ADM}, \mathsf{DID}, \sigma')$.

(Redactions can be done without the knowledge of the $\mathsf{sk}_{\mathsf{Agg}}$.)

- **Game**$_{2-7}$: We receiving the output forgery $(M^*, \mathsf{ADM}^*, \mathsf{DID}^*, \sigma^*)$ from the adversary A_3,
  - Set $v^{(0)} \leftarrow (\mathsf{DID}^* || \mathrm{ord}(\mathsf{ADM}^*))$, $v^{(j)} \leftarrow (\mathsf{DID} || m_j^*)$ $(1 \leq j \leq \# M^*)$.
  - Query $v^{(j)}$ $(0 \leq j \leq \# M^*)$ to $\mathcal{O}^H$. We assume $\langle v^{(j)}, w^{(j)}, s^{(j)}, c^{(j)} \rangle$ to be the tuple in $\mathbb{T}$ for each $v^{(j)}$ $(0 \leq j \leq \# M^*)$.
  - If $c^{(0)} = 1$ and $c^{(j)} = 0$ $(1 \leq j \leq \# M^*)$, then accept. Otherwise reject and abort.

Lemma 4.13. The following equation holds.

$$\mathsf{Adv}_{\mathsf{A}_2}[\mathbf{Game}_{2-1}] = \mathsf{Adv}_{\mathsf{A}_2}[\mathbf{Game}_{2-0}].$$

Since outputs of $\mathcal{O}^{\mathsf{Redact}}$ in **Game**$_{2-0}$ and **Game**$_{2-1}$ are same.

Lemma 4.14. The following equation holds.

$$\mathsf{Adv}_{\mathsf{A}_2}[\mathbf{Game}_{2-2}] = \mathsf{Adv}_{\mathsf{A}_2}[\mathbf{Game}_{2-1}].$$

To simplify the discussion, let A_2 get $\mathsf{rk}[i], \ldots, \mathsf{rk}[t-1]$ from $\mathcal{O}^{\mathsf{Corrupt}}$. In [**Game**$_{2-1}$], the following equation holds.

$$V \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{t-1} \end{pmatrix} = \begin{pmatrix} f(0) \\ f(1) \\ f(2) \\ \vdots \\ f(t-1) \end{pmatrix} \text{ where } V = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 2^2 & \cdots & 2^{t-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & t-1 & (t-1)^2 & \cdots & (t-1)^{t-1} \end{pmatrix}.$$

Since V is the Vandermonde matrix, V is the regular matrix. Distributions of $(a_0, a_1, \cdots, a_{t-1})$ and $(f(0), f(1), \ldots, f(t-1))$ are identical. Therefore, distributions of $(\mathsf{sk}_{\mathsf{Agg}}, \mathsf{rk}[1], \ldots, \mathsf{rk}[t-1])$ in $[\mathbf{Game}_{2-1}]$ and $[\mathbf{Game}_{2-2}]$ are same.

Lemma 4.15. If H is the random oracle model, the following equation holds.

$$\mathsf{Adv}_{\mathsf{A}_2}[\mathbf{Game}_{2-3}] = \mathsf{Adv}_{\mathsf{A}_2}[\mathbf{Game}_{2-2}].$$

Since the distribution of outputs of $\mathcal{O}^H$ in $\mathbf{Game}_{2-3}$ and $\mathbf{Game}_{2-2}$ is identical.

Lemma 4.16. The following inequality holds.

$$\mathsf{Adv}_{\mathsf{A}_2}[\mathbf{Game}_{2-4}] \geq (1 - 1/((\ell+1)(q_s+q_r)+1))^{(\ell+1)q_s} \times \mathsf{Adv}_{\mathsf{A}_2}[\mathbf{Game}_{2-3}].$$

Since the probability that each signing query does not abort at least $(1 - 1/((\ell+1)(q_s+q_r)+1))^{(\ell+1)}$.

Lemma 4.17. The following equation holds.

$$\mathsf{Adv}_{\mathsf{A}_2}[\mathbf{Game}_{2-5}] = \mathsf{Adv}_{\mathsf{A}_2}[\mathbf{Game}_{2-4}].$$

Since outputs of Sign in $\mathbf{Game}_{2-5}$ and $\mathbf{Game}_{2-4}$ are same.

Lemma 4.18. The following inequality holds.

$$\mathsf{Adv}_{\mathsf{A}_2}[\mathbf{Game}_{2-6}] \geq (1 - 1/((\ell+1)(q_s+q_r)+1))^{(\ell+1)q_r} \times \mathsf{Adv}_{\mathsf{A}_2}[\mathbf{Game}_{2-5}].$$

Since the probability that each redaction query does not abort at least $(1 - 1/((\ell+1)(q_s+q_r)+1))^{(\ell+1)}$.

Lemma 4.19. The following inequality holds.

$$\begin{aligned}
&\mathsf{Adv}_{\mathsf{A}_2}[\mathbf{Game}_{2-7}] \\
&\geq \frac{\left(\frac{1}{2(\ell+1)(q_s+q_r)+2}\right)^2}{\left(1 - \frac{1}{(\ell+1)(q_s+q_r)+1}\right)^2 + \left(\frac{1}{2(\ell+1)(q_s+q_r)+2}\right)^2} \times \mathsf{Adv}_{\mathsf{A}_2}[\mathbf{Game}_{2-6}] \\
&= (1/(4(\ell+1)^2(q_s+q_r)^2+1)) \times \mathsf{Adv}_{\mathsf{A}_2}[\mathbf{Game}_{2-6}].
\end{aligned}$$

Since an output $(M^*, \mathsf{ADM}^*, \mathsf{DID}^*, \sigma^*)$ satisfies $(c^{(0)}, c^{(1)}) = (0,0)$ or $(2,1)$.

To summarize from Lemma 4.13 to Lemma 4.19, the following holds.
(In the following equation, e represents the Napier's constant.)

$$\begin{aligned}\mathsf{Adv}_{\mathsf{A}_2}[\mathbf{Game}_{2-7}] &\geq (1-1/((\ell+1)(q_s+q_r)+1))^{(\ell+1)(q_s+q_r)} \\ &\quad \times 1/(4(\ell+1)^2(q_s+q_r)^2+1) \times \mathsf{Adv}_{\mathsf{A}_2}[\mathbf{Game}_{2-0}] \\ &\geq (1/e) \times (1/(4(\ell+1)^2(q_s+q_r)^2+1)) \times \mathsf{Adv}_{\mathsf{A}_2}[\mathbf{Game}_{2-0}]\end{aligned}$$

Now we construct the algorithm B_2 which breaking the computational co-Diffie-Hellman assumption using the algorithm A_2. The operation of B_2 for the input co-Diffie-Hellman problem instance (g_2, g_2^{α}, h^*) is changed to h in $\mathbf{Game}_{2-7}$ to h^* and u to g_2^{α}.
Suppose B_2 do not abort receiving a forgery $(M^*, \mathsf{ADM}^*, \mathsf{DID}^*, \sigma^*)$ from A_2. B_3 parses σ^* as $(\sigma^*_{\mathsf{ADM}^*}, \Sigma^*_{\mathsf{agg}})$, sets $v^{(j)} \leftarrow (\mathsf{DID}^*||m^*_j)$ $(1 \leq j \leq \#M^*)$, and computes $w^{(1)} \leftarrow h \cdot \phi(u)^{b^{(1)}} \cdot \phi(g_2)^{rb^{(1)}}$, $w^{(j)} \leftarrow \phi(u)^{b^{(j)}} \cdot \phi(g_2)^{rb^{(j)}}$ $(2 \leq j \leq \#M^*)$. Then B_3 computes $\sigma^*_{m^*_1} \leftarrow \Sigma^*_{\mathsf{agg}} / \prod_{j=2}^{\#M*} \sigma_{m_j}$. Since $(M^*, \mathsf{ADM}^*, \mathsf{DID}^*, \sigma^*)$ is valid signature and $\mathsf{vk}_{\mathsf{Agg}} = g_2^{\alpha+r}$, $e(\sigma^*_{m^*_1}, g_2) = e\left((w^{(1)})^{\alpha+r}, g_2\right)$ holds. It implies that $\sigma^*_{m^*_1} = (w^{(1)})^{\alpha+r} = (h^* \cdot \phi(g_2)^{b^{(1)}})^{\alpha+r}$. Therefore, B_3 computes $(h^*)^{\alpha} = \sigma^*_{m^*_1} \cdot (\phi(u)^{b^{(1)}} \cdot (h^*)^r \cdot \phi(g_2)^{rb^{(1)}})^{-1}$ and outputs the solution $(h^*)^{\alpha}$ of the computational co-Diffie-Hellman problem instance (g_2, g_2^{α}, h^*).

Let $\epsilon_{\mathsf{co\text{-}cdh2}}$ is the probability that B_2 break the computational co-Diffie-Hellman assumption. We can bound the probability $\epsilon_{\mathsf{co\text{-}cdh2}} \geq \mathsf{Adv}_{\mathsf{A}_2}[\mathbf{Game}_{2-7}]$ and $\epsilon_{\mathsf{co\text{-}cdh2}} \geq (1/e) \times (1/(4(\ell+1)^2(q_s+q_r)^2+1)) \times \epsilon_{\mathsf{uf2}}$ holds. (e represents the Napier's constant.) If ϵ_{uf2} is non-negligiable in λ, B_2 breaks the computational co-Diffie-Hellman assumption with non-negligiable in $\epsilon_{\mathsf{co\text{-}cdh2}}$.

Case 3:

We consider an adversary A_3 that can generate a valid forgery with ϵ_{uf3} against our proposal redactable signature scheme. Let $\mathbf{Game}_{3-0}$ be the original unforgeability game in a redactable signature scheme and $\mathbf{Game}_{3-6}$ be directly related to solve the computational co-Diffie-Hellman problem. Define $\mathsf{Adv}_{\mathsf{A}_3}[\mathbf{Game}_{3-X}]$ as the advantage of an adversary A_3 in $\mathbf{Game}_{3-X}$.

- $\mathbf{Game}_{3-0}$: Original unforgeability game in a redactable signature scheme.

$$\mathsf{Adv}_{\mathsf{A}_3}[\mathbf{Game}_{3-0}] = \epsilon_{\mathsf{uf3}}$$

- $\mathbf{Game}_{3-1}$: $\mathbf{Game}_{3-1}$ is the same as $\mathbf{Game}_{2-1}$.

- $\mathbf{Game}_{3-2}$: $\mathbf{Game}_{3-2}$ is the same as $\mathbf{Game}_{2-2}$.

- **Game**$_{3-3}$: We change a setting of the random oracle $\mathcal{O}^H$. Fix $h \xleftarrow{\$} \mathbb{G}_2$ and let $\mathbb{T}$ be a table that maintains a list of tuples $\langle v, w, b, c\rangle$ as explain below. We refer to this list for the query to $\mathcal{O}^h$. The initial state of $\mathbb{T}$ is empty. For queries $v^{(i)}$ to $\mathcal{O}^H$:
  - If $\langle v^{(i)}, w^{(i)}, \cdot, \cdot\rangle$ (Here, '$\cdot$' represents an arbitrary value) already appears in $\mathbb{T}$, then return $w^{(i)}$.
  - Choose $s^{(i)} \xleftarrow{\$} \mathbb{Z}_q$.
  - Flip a biased coin $c^{(i)} \in \{0, 1\}$ such that $\Pr[c^{(i)} = 0] = 1 - 1/((\ell+1)(q_s + q_r) + \ell)$, $\Pr[c^{(i)} = 1] = 1/((\ell+1)(q_s + q_r) + \ell)$.
  - If $c^{(i)} = 0$, compute $w^{(i)} = \phi(g_2)^{b^{(i)}}$.
  - If $c^{(i)} = 1$, compute $w^{(i)} = h \cdot \phi(g_2)^{b^{(i)}}$.
  - Insert $\langle v^{(i)}, w^{(i)}, s^{(i)}, c^{(i)}\rangle$ in $\mathbb{T}$ and return $w^{(i)}$.
- **Game**$_{3-4}$:We modify the signing algorithm $\mathsf{RS.Sign}$ in Step 6 as follows:
  - Set $v^{(0)} \leftarrow (\mathsf{DID}||\mathrm{ord}(\mathsf{ADM}))$, $v^{(j)} \leftarrow (\mathsf{DID}||m_j)$ $(1 \leq j \leq \#M)$.
  - Query $v^{(j)}$ $(0 \leq j \leq \#M)$ to $\mathcal{O}^H$. We assume $\langle v^{(j)}, w^{(j)}, b^{(j)}, c^{(j)}\rangle$ to be the tuple in $\mathbb{T}$ for each $v^{(j)}$ $(1 \leq j \leq \#M)$.
  - If $c^{(j)} = 0$ $(0 \leq \forall j \leq \#M)$, go to Step 6 of $\mathsf{RS.Sign}$. Otherwise return $\perp$ and abort.
- **Game**$_{3-5}$: We modify the signing algorithm $\mathsf{RS.Sign}$ in Step 7, 8 as follows:
  - Compute $\sigma_{\mathsf{ADM}} \leftarrow \phi(u)^{b^{(0)}} \cdot \phi(g_2)^{rb^{(0)}}$.
  - For all $m_j \in M$, compute $\sigma_{m_j} \leftarrow \phi(u)^{b^{(j)}} \cdot \phi(g_2)^{rb^{(j)}}$.
  - Compute $\Sigma_{\mathsf{agg}} \leftarrow \sigma_{\mathsf{ADM}} \cdot \prod_{m_j \in M} \sigma_{m_j}$.

  (By above modification, a signature Σ_{agg} can be generated without a knowledge of the $\mathsf{sk}_{\mathsf{Agg}}$.)
- **Game**$_{3-6}$: We change a setting of $\mathcal{O}^{\mathsf{Redact}}$.
  - Parse vk as $(\mathsf{vk}_{\mathsf{Fix}}, \mathsf{vk}_{\mathsf{Agg}}, t, n)$ and σ as $(\sigma_{\mathsf{Fix}}, \Sigma_{\mathsf{agg}})$.
  - If $\mathsf{DID} \in \mathbb{L}^{u-1}$, then abort.
  - Set $\mathbb{L}^u \leftarrow \mathbb{L}^{u-1} \cup \{\mathsf{DID}\}$.
  - If $\mathsf{MOD} \not\subseteq M \vee \mathsf{MOD} \cap \mathsf{ADM} \neq \emptyset$, then abort.

- Set $v^{(0)} \leftarrow (\mathsf{DID}||\mathrm{ord}(\mathsf{ADM}))$, $v^{(j)} \leftarrow (\mathsf{DID}||m_j)$ $(1 \leq j \leq \#M)$.
- Query $v^{(j)}$ $(0 \leq j \leq \#\mathsf{MOD})$ to $\mathcal{O}^H$. We assume $\langle v^{(j)}, w^{(j)}, b^{(j)}, c^{(j)} \rangle$ to be the tuple in $\mathbb{T}$ for each $v^{(j)}$ $(1 \leq j \leq \#\mathsf{MOD})$.
- If $e(\sigma_{\mathsf{Fix}}, g_2) \neq e(w^{(0)}, \mathsf{vk}_{\mathsf{Fix}})$, then abort.
- If $e(\Sigma_{\mathsf{agg}}, g_2) \neq \prod_{0 \leq j \leq \#M} e(w^{(j)}, \mathsf{vk}_{\mathsf{Agg}})$, then abort.
- If $c^{(j)} = 0$ $(\forall m_j \in \mathsf{MOD})$, go to next step. Otherwise return $\perp$ and abort.
- For all $m_j \in \mathsf{MOD}$, compute $\sigma_{m_j} \leftarrow \phi(u)^{b^{(j)}} \cdot \phi(g_2)^{rb^{(j)}}$.
- Compute $\sigma_{\mathsf{MOD}} \leftarrow \prod_{m_j \in \mathsf{MOD}} \sigma_{m_j}$, $\Sigma'_{\mathsf{agg}} \leftarrow \Sigma_{\mathsf{agg}}/\sigma_{\mathsf{MOD}}$.
- Set $M' \leftarrow M \backslash \mathsf{MOD}$, $\sigma' \leftarrow (\sigma_{\mathsf{Fix}}, \Sigma'_{\mathsf{agg}})$.
- Return $(M', \mathsf{ADM}, \mathsf{DID}, \sigma')$.

(Redactions can be done without the knowledge of the $\mathsf{sk}_{\mathsf{Agg}}$.)

- **Game**$_{3-7}$: We receiving the output forgery $(M^*, \mathsf{ADM}^*, \mathsf{DID}^*, \sigma^*)$ from the adversary A_3,
  - Set $v^{(0)} \leftarrow (\mathsf{DID}^*||\mathrm{ord}(\mathsf{ADM}^*))$, $v^{(j)} \leftarrow (\mathsf{DID}||m_j^*)$ $(1 \leq j \leq \#M^*)$.
  - Query $v^{(j)}$ $(0 \leq j \leq \#M^*)$ to $\mathcal{O}^H$. We assume $\langle v^{(j)}, w^{(j)}, s^{(j)}, c^{(j)} \rangle$ to be the tuple in $\mathbb{T}$ for each $v^{(j)}$ $(0 \leq j \leq \#M^*)$.
  - If $c^{(1)} = 1$ and $c^{(j)} = 0$ $(2 \leq j \leq \#M^*)$, then accept. Otherwise reject and abort.

Lemma 4.20. If H is the random oracle model, the following equation holds.

$$\mathsf{Adv}_{\mathsf{A}_3}[\mathbf{Game}_{3-3}] = \mathsf{Adv}_{\mathsf{A}_1}[\mathbf{Game}_{3-2}]$$

Since the distribution of outputs of $\mathcal{O}^H$ in **Game**$_{3-3}$ and **Game**$_{3-2}$ is identical.

Lemma 4.21. The following inequality holds.

$$\mathsf{Adv}_{\mathsf{A}_3}[\mathbf{Game}_{3-4}] \geq (1 - 1/((\ell+1)(q_s+q_r)+\ell))^{(\ell+1)q_s} \times \mathsf{Adv}_{\mathsf{A}_3}[\mathbf{Game}_{3-3}].$$

Since the probability that each signing query does not abort at least $(1 - 1/((\ell+1)(q_s+q_r)+(\ell+1)))^{(\ell+1)}$.

Lemma 4.22. The following equation holds.

$$\mathsf{Adv}_{\mathsf{A}_3}[\mathbf{Game}_{3-5}] = \mathsf{Adv}_{\mathsf{A}_3}[\mathbf{Game}_{3-4}].$$

Since outputs of Sign in $\mathbf{Game}_{3-5}$ and $\mathbf{Game}_{3-4}$ are same.

Lemma 4.23. The following inequality holds.

$$\mathsf{Adv}_{\mathsf{A}_3}[\mathbf{Game}_{3-6}] \geq (1 - 1/((\ell+1)(q_s+q_r)+\ell))^{(\ell+1)q_r} \times \mathsf{Adv}_{\mathsf{A}_3}[\mathbf{Game}_{3-5}].$$

Since the probability that each redaction query does not abort at least $(1 - 1/((\ell+1)(q_s+q_r)+\ell))^{(\ell+1)}$.

Lemma 4.24. The following inequality holds.

$$\begin{aligned}\mathsf{Adv}_{\mathsf{A}_3}[\mathbf{Game}_{3-7}] \geq & (1 - 1/((\ell+1)(q_s+q_r)+\ell))^{\ell-1} \\ & \times (1/((\ell+1)(q_s+q_r)+\ell)) \times \mathsf{Adv}_{\mathsf{A}_3}[\mathbf{Game}_{3-6}].\end{aligned}$$

Since an output $(M^*, \mathsf{ADM}^*, \mathsf{DID}^*, \sigma^*)$ satisfies $c^{(0)} = 0$. The probability that $(M^*, \mathsf{ADM}^*, \mathsf{DID}^*, \sigma^*)$ satisfies $c^{(1)} = 1$ and $c^{(i)} = 0$ $(2 \leq i \leq \#M^*)$ at least $(1 - 1/((\ell+1)(q_s+q_r)+\ell))^{(\ell-1)} \times (1 - 1/((\ell+1)(q_s+q_r)+\ell))$.

To summarize from Lemma 4.13, Lemma 4.14, and from Lemma 4.20 to Lemma 4.24, the following holds. (In the following equation, e represents the Napier's constant.)

$$\begin{aligned}\mathsf{Adv}_{\mathsf{A}_3}[\mathbf{Game}_{3-7}] &\geq (1 - 1/((\ell+1)(q_s+q_r)+\ell))^{(\ell+1)(q_s+q_r)+\ell-1} \\ &\quad \times (1/((\ell+1)(q_s+q_r)+\ell)) \times \mathsf{Adv}_{\mathsf{A}_3}[\mathbf{Game}_{3-0}] \\ &\geq (1/e) \times (1/((\ell+1)(q_s+q_r)+\ell)) \times \mathsf{Adv}_{\mathsf{A}_3}[\mathbf{Game}_{3-0}]\end{aligned}$$

Now we construct the algorithm B_3 which breaking the computational co-Diffie-Hellman assumption using the algorithm A_3. The operation of B_3 for the input co-Diffie-Hellman problem instance (g_2, g_2^{α}, h^*) is changed to h in $\mathbf{Game}_{3-7}$ to h^* and u to g_2^{α}.
Suppose B_3 do not abort receiving a forgery $(M^*, \mathsf{ADM}^*, \mathsf{DID}^*, \sigma^*)$ from A_3. B_3 parses σ^* as $(\sigma^*_{\mathsf{ADM}^*}, \Sigma^*_{\mathsf{agg}})$, sets $v^{(j)} \leftarrow (\mathsf{DID}^* || m_j^*)$ $(1 \leq j \leq \#M^*)$, and computes $w^{(1)} \leftarrow h \cdot \phi(u)^{b^{(1)}} \cdot \phi(g_2)^{rb^{(1)}}$, $w^{(j)} \leftarrow \phi(u)^{b^{(j)}} \cdot \phi(g_2)^{rb^{(j)}}$ $(2 \leq j \leq \#M^*)$. Then B_3 computes $\sigma^*_{m_1^*} \leftarrow \Sigma^*_{\mathsf{agg}} / \prod_{j=2}^{\#M*} \sigma_{m_j}$. Since $(M^*, \mathsf{ADM}^*, \mathsf{DID}^*, \sigma^*)$ is valid signature and $\mathsf{vk}_{\mathsf{Agg}} = g_2^{\alpha+r}$, $e(\sigma^*_{m_1^*}, g_2) = e\left((w^{(1)})^{\alpha+r}, g_2\right)$ holds. It implies that $\sigma^*_{m_1^*} = (w^{(1)})^{\alpha+r} = (h^* \cdot \phi(g_2)^{b^{(1)}})^{\alpha+r}$. Threfore, B_3 computes $(h^*)^{\alpha} = \sigma^*_{m_1^*} \cdot (\phi(u)^{b^{(1)}} \cdot (h^*)^r \cdot \phi(g_2)^{rb^{(1)}})^{-1}$ and outputs the solution $(h^*)^{\alpha}$ of the computational co-Diffie-Hellman problem instance (g_2, g_2^{α}, h^*).

Let $\epsilon_{\mathsf{co\text{-}cdh3}}$ is the probability that B_3 break the computational co-Diffie-Hellman assumption. We can bound the probability $\epsilon_{\mathsf{co\text{-}cdh3}} \geq \mathsf{Adv}_{\mathsf{A}_3}[\mathbf{Game}_{3-7}]$ and $\epsilon_{\mathsf{co\text{-}cdh3}} \geq (1/e) \times (1/((\ell+1)(q_s+q_r)+\ell)) \times \epsilon_{\mathsf{uf3}}$ holds. (e represents the Napier's constant.) Hence, if ϵ_{uf3} is non-negligiable in λ, B_3 breaks the computational co-Diffie-Hellman assumption with non-negligiable in $\epsilon_{\mathsf{co\text{-}cdh3}}$.

□

4.7 Security Proof for Transparency

Theorem 4.25. Our proposed t-out-of-n redactable signature scheme in the one-time redaction model (t, n)-RS Π_1 satisfies the transparency.

proof. We proceed by a sequence of games. Define $\mathsf{Adv}_{\mathsf{A}}[\mathbf{Game}_i]$ as the advantage of an adversary A in $\mathbf{Game}_i$.

- $\mathbf{Game}_0$: Original transparency game in a redactable signature scheme.
- $\mathbf{Game}_1$: We change the redaction algorithm $\mathsf{RS.Redact}$ in $\mathcal{O}^{\mathsf{Sign/Redact}}$.
  - Skip the step 2 of $\mathsf{RS.RedInf}$.

Let q_r be the total number of queries from an adversary A to $\mathcal{O}^{\mathsf{Redact}}$. Then, $|\mathsf{Adv}_{\mathsf{A}}[\mathbf{Game}_1] - \mathsf{Adv}_{\mathsf{A}}[\mathbf{Game}_0]| \leq q_r \times q_r/2^d$ holds. We consider distribution of output $\mathcal{O}^{\mathsf{Sign/Redact}}$ in case of $b = 0$ and $b = 1$ of $\mathbf{Game}_1$. Given an input $(M, \mathsf{ADM}, \mathsf{MOD})$ to $\mathcal{O}^{\mathsf{Sign/Redact}}$, $\mathcal{O}^{\mathsf{Sign/Redact}}$ compute

- $(M, \mathsf{ADM}, \mathsf{DID}_0, \sigma) \leftarrow \mathsf{Sign}(pp, \mathsf{sk}, M, \mathsf{ADM})$.
- $\mathsf{RI}_i \leftarrow \mathsf{RS.RedInf}(pp, \mathsf{vk}, \mathsf{rk}[i], M, \mathsf{ADM}, \mathsf{DID}_0, \sigma, \mathsf{MOD})$ for $1 \leq i \leq n$.
- $(M', \mathsf{ADM}', \mathsf{DID}_0, \sigma_0) \leftarrow \mathsf{RS.ThrRed}(pp, \mathsf{vk}, M, \mathsf{ADM}, \mathsf{DID}_0, \sigma, \{\mathsf{RI}_i\}_{i=1}^n)$.
- $(M', \mathsf{ADM}', \mathsf{DID}_1, \sigma_1) \leftarrow \mathsf{Sign}(pp, \mathsf{sk}, M', \mathsf{ADM}')$.

Distributions of DID_0 and DID_1 in $\mathbf{Game}_1$ are identical and $\mathcal{O}^{\mathsf{Sign/Redact}}$ skips the step 2 of $\mathsf{RS.RedInf}$. Therefore, distributions of $\{(M', \mathsf{ADM}', \mathsf{DID}_0, \sigma_0)\}$ and $\{(M', \mathsf{ADM}', \mathsf{DID}_1, \sigma_1)\}$ outputted by $\mathcal{O}^{\mathsf{Sign/Redact}}$ are identical. It means that $\mathsf{Adv}_{\mathsf{A}}[\mathbf{Game}_1] = 1/2$. Let ϵ_{Tran} is the advantage of an adversary A in original $\mathbf{Game}_0$. We can bound the probability $\epsilon_{\mathsf{Tran}} \leq q_r \times q_r/2^d + 1/2$. Therefore, our proposed t-out-of-n redactable signature scheme in the one-time redaction model (t, n)-RS Π_1 satisfies transparency. □

By Theorem 4.6 and Theorem 4.25, our proposed scheme satisfies the privacy.

Chapter 5

Conclusion

In this thesis, first, we give a new security proof for the synchronized aggregate signature scheme by Lee et al. [38] under the $\mathsf{OT\text{-}EUF\text{-}CMA}$ security for the $\mathsf{DS}_{\mathsf{MCL}}$ scheme in the ROM. Since the $\mathsf{OT\text{-}EUF\text{-}CMA}$ security for the $\mathsf{DS}_{\mathsf{MCL}}$ scheme is proven under the 1-MSDH-2 assumption, our result shows that the aggregate signature by Lee et al. [38] can be proven under the non-interactive and static assumption in the ROM.

However, there still have problems for the security of the synchronized aggregate signature scheme by Lee et al. [38]. First, the 1-MSDH-2 assumption is not standard assumption, it is desirable that the security is proven under a standard assumption (e.g., CDH assumption). Second, we prove the $\mathsf{EUF\text{-}CMA}$ for the aggregate signature scheme by Lee et al. in the certify key model. This model limits the use scenarios for the synchronized aggregate signature scheme. Removing the certify key model is an important open problem for practicality.

Second, we introduce the new notion of t-out-of-n redactable signature. Then we construct the t-out-of-n redactable signature scheme based on the aggregate signature scheme by Boneh et al. [11] and Shamir's secret sharing schemes. Then, we prove that our construction satisfies unforgeability and transparency.

However, our proposed model supports only the one-time redaction model which allows redacting signed message only one time for each signature. Our construction Π_1 does not satisfy the unforgeability in a model that allows redacting signed message many times. For example, $M = \{m_1, m_2, m_3\}$ and $\mathsf{ADM} = \emptyset$, an adversary who does the following operation generates a valid forgery in a multiple redactions model.

1. Given vk from C.

2. Query (M, ADM) to $\mathcal{O}^{\mathsf{Sign}}$ and get $(M, \mathsf{ADM}, \mathsf{DID}, \sigma)$.

3. Let $\mathsf{MOD}^1 = \{m_1\}$ $\mathsf{MOD}^2 = \{m_2\}$. Query $(M, \mathsf{ADM}, \mathsf{DID}, \sigma, \mathsf{MOD}^1)$ to $\mathcal{O}^{\mathsf{Redact}}$ and get $(M', \mathsf{ADM}, \mathsf{DID}, \sigma')$ and query $(M', \mathsf{ADM}, \mathsf{DID}, \sigma', \mathsf{MOD}^2)$ to $\mathcal{O}^{\mathsf{Redact}}$ and get $(M'', \mathsf{ADM}, \mathsf{DID}, \sigma'')$.

4. Parse σ as $(\sigma_{\mathsf{Fix}}, \Sigma_{\mathsf{agg}})$, σ' as $(\sigma'_{\mathsf{Fix}}, \Sigma'_{\mathsf{agg}})$, and σ'' as $(\sigma''_{\mathsf{Fix}}, \Sigma''_{\mathsf{agg}})$.

5. Compute $\sigma_{m_1} \leftarrow \Sigma_{\mathsf{agg}} \cdot (\Sigma'_{\mathsf{agg}})^{-1}$, $\Sigma^*_{\mathsf{agg}} \leftarrow \sigma_{m_1} \cdot \Sigma''_{\mathsf{agg}}$.

6. Set $M^* \leftarrow \{m_1, m_3\}$, $\sigma^* \leftarrow (\sigma_{\mathsf{Fix}}, \Sigma^*_{\mathsf{agg}})$ and output $(M^*, \mathsf{DID}, \mathsf{ADM}, \sigma^*)$

Giving a construction of (t, n)-RS in the multiple redactions model is an interesting open problem.

Bibliography

[1] J. H. Ahn, M. Green, and S. Hohenberger. Synchronized aggregate signatures: new definitions, constructions and applications. In *Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010, Chicago, Illinois, USA, October 4-8, 2010*, pages 473–484, 2010. URL: https://doi.org/10.1145/1866307.1866360, doi:10.1145/1866307.1866360.

[2] G. Ateniese, D. H. Chou, B. de Medeiros, and G. Tsudik. Sanitizable signatures. In S. D. C. di Vimercati, P. F. Syverson, and D. Gollmann, editors, *Computer Security - ESORICS 2005, 10th European Symposium on Research in Computer Security, Milan, Italy, September 12-14, 2005, Proceedings*, volume 3679 of *Lecture Notes in Computer Science*, pages 159–177. Springer, 2005. URL: https://doi.org/10.1007/11555827_10, doi:10.1007/11555827_10.

[3] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *CCS '93, Proceedings of the 1st ACM Conference on Computer and Communications Security, Fairfax, Virginia, USA, November 3-5, 1993.*, pages 62–73, 1993. URL: http://doi.acm.org/10.1145/168588.168596, doi:10.1145/168588.168596.

[4] A. Bender, J. Katz, and R. Morselli. Ring signatures: Stronger definitions, and constructions without random oracles. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, pages 60–79, 2006. URL: https://doi.org/10.1007/11681878_4, doi:10.1007/11681878_4.

[5] D. Bernhard, G. Fuchsbauer, E. Ghadafi, N. P. Smart, and B. Warinschi. Anonymous attestation with user-controlled linkability. *Int. J. Inf. Sec.*, 12(3):219–249, 2013. URL: https://doi.org/10.1007/s10207-013-0191-z, doi:10.1007/s10207-013-0191-z.

[6] P. Bichsel, J. Camenisch, G. Neven, N. P. Smart, and B. Warinschi. Get shorty via group signatures without encryption. In *Security and Cryptography for Networks, 7th International Conference, SCN 2010, Amalfi, Italy, September 13-15, 2010. Proceedings*, pages 381–398, 2010. URL: `https://doi.org/10.1007/978-3-642-15317-4_24`, `doi:10.1007/978-3-642-15317-4\_24`.

[7] A. Bilzhause, H. C. Pöhls, and K. Samelin. Position paper: The past, present, and future of sanitizable and redactable signatures. In *Proceedings of the 12th International Conference on Availability, Reliability and Security, Reggio Calabria, Italy, August 29 - September 01, 2017*, pages 87:1–87:9. ACM, 2017. URL: `https://doi.org/10.1145/3098954.3104058`, `doi:10.1145/3098954.3104058`.

[8] A. Boldyreva, C. Gentry, A. O'Neill, and D. H. Yum. Ordered multisignatures and identity-based sequential aggregate signatures, with applications to secure routing. In *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, October 28-31, 2007*, pages 276–285, 2007. URL: `https://doi.org/10.1145/1315245.1315280`, `doi:10.1145/1315245.1315280`.

[9] D. Boneh and X. Boyen. Short signatures without random oracles. In *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, pages 56–73, 2004. URL: `https://doi.org/10.1007/978-3-540-24676-3_4`, `doi:10.1007/978-3-540-24676-3\_4`.

[10] D. Boneh, X. Boyen, and E. Goh. Hierarchical identity based encryption with constant size ciphertext. In *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, pages 440–456, 2005. URL: `https://doi.org/10.1007/11426639_26`, `doi:10.1007/11426639\_26`.

[11] D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*, pages 416–432, 2003. URL: `https://doi.org/10.1007/3-540-39200-9_26`, `doi:10.1007/3-540-39200-9\_26`.

[12] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. In C. Boyd, editor, *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings*, volume 2248 of *Lecture Notes in Computer Science*, pages 514–532. Springer, 2001. URL: https://doi.org/10.1007/3-540-45682-1_30, doi:10.1007/3-540-45682-1_30.

[13] C. Brzuska, H. Busch, Ö. Dagdelen, M. Fischlin, M. Franz, S. Katzenbeisser, M. Manulis, C. Onete, A. Peter, B. Poettering, and D. Schröder. Redactable signatures for tree-structured data: Definitions and constructions. In J. Zhou and M. Yung, editors, *Applied Cryptography and Network Security, 8th International Conference, ACNS 2010, Beijing, China, June 22-25, 2010. Proceedings*, volume 6123 of *Lecture Notes in Computer Science*, pages 87–104, 2010. URL: https://doi.org/10.1007/978-3-642-13708-2_6, doi:10.1007/978-3-642-13708-2_6.

[14] J. Camenisch, M. Dubovitskaya, K. Haralambiev, and M. Kohlweiss. Composable and modular anonymous credentials: Definitions and practical constructions. In T. Iwata and J. H. Cheon, editors, *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, volume 9453 of *Lecture Notes in Computer Science*, pages 262–288. Springer, 2015. URL: https://doi.org/10.1007/978-3-662-48800-3_11, doi:10.1007/978-3-662-48800-3_11.

[15] J. Camenisch, S. Hohenberger, and M. Ø. Pedersen. Batch verification of short signatures. In *Advances in Cryptology - EUROCRYPT 2007, 26th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Barcelona, Spain, May 20-24, 2007, Proceedings*, pages 246–263, 2007. URL: https://doi.org/10.1007/978-3-540-72540-4_14, doi:10.1007/978-3-540-72540-4_14.

[16] J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, pages 56–72, 2004. URL: https://doi.org/10.1007/978-3-540-28628-8_4, doi:10.1007/978-3-540-28628-8_4.

[17] S. Canard, D. Pointcheval, O. Sanders, and J. Traoré. Divisible e-cash made practical. In *Public-Key Cryptography - PKC 2015 - 18th IACR International Conference on Practice and Theory in Public-Key Cryptography, Gaithersburg, MD, USA, March 30 - April 1, 2015, Proceedings*, pages 77–100, 2015. URL: https://doi.org/10.1007/978-3-662-46447-2_4, doi:10.1007/978-3-662-46447-2_4.

[18] D. Chaum. Security without identification: Transaction systems to make big brother obsolete. *Commun. ACM*, 28(10):1030–1044, 1985. URL: https://doi.org/10.1145/4372.4373, doi:10.1145/4372.4373.

[19] H. Delfs and H. Knebl. *Introduction to Cryptography - Principles and Applications, Third Edition*. Information Security and Cryptography. Springer, 2015. URL: https://doi.org/10.1007/978-3-662-47974-2, doi:10.1007/978-3-662-47974-2.

[20] D. Demirel, D. Derler, C. Hanser, H. Pöhls, D. Slamanig, and G. Traverso. *PRISMACLOUD D4.4: Overview of Functional and Malleable Signature Schemes*. 2015.

[21] D. Derler, H. C. Pöhls, K. Samelin, and D. Slamanig. A general framework for redactable signatures and new constructions. In S. Kwon and A. Yun, editors, *Information Security and Cryptology - ICISC 2015 - 18th International Conference, Seoul, South Korea, November 25-27, 2015, Revised Selected Papers*, volume 9558 of *Lecture Notes in Computer Science*, pages 3–19. Springer, 2015. URL: https://doi.org/10.1007/978-3-319-30840-1_1, doi:10.1007/978-3-319-30840-1_1.

[22] S. D. Galbraith, K. G. Paterson, and N. P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008. URL: https://doi.org/10.1016/j.dam.2007.12.010, doi:10.1016/j.dam.2007.12.010.

[23] C. Gentry and Z. Ramzan. Identity-based aggregate signatures. In *Public Key Cryptography - PKC 2006, 9th International Conference on Theory and Practice of Public-Key Cryptography, New York, NY, USA, April 24-26, 2006, Proceedings*, pages 257–273, 2006. URL: https://doi.org/10.1007/11745853_17, doi:10.1007/11745853_17.

[24] S. Haber, Y. Hatano, Y. Honda, W. G. Horne, K. Miyazaki, T. Sander, S. Tezoku, and D. Yao. Efficient signature schemes supporting redaction, pseudonymization, and data deidentification. In M. Abe and V. D. Gligor, editors, *Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security, ASIACCS*

2008, Tokyo, Japan, March 18-20, 2008, pages 353–362. ACM, 2008. URL: https://doi.org/10.1145/1368310.1368362, doi:10.1145/1368310.1368362.

[25] G. Hartung, B. Kaidel, A. Koch, J. Koch, and A. Rupp. Fault-tolerant aggregate signatures. In *Public-Key Cryptography - PKC 2016 - 19th IACR International Conference on Practice and Theory in Public-Key Cryptography, Taipei, Taiwan, March 6-9, 2016, Proceedings, Part I*, pages 331–356, 2016. URL: https://doi.org/10.1007/978-3-662-49384-7_13, doi:10.1007/978-3-662-49384-7_13.

[26] S. Hohenberger, V. Koppula, and B. Waters. Universal signature aggregators. In *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II*, pages 3–34, 2015. URL: https://doi.org/10.1007/978-3-662-46803-6_1, doi:10.1007/978-3-662-46803-6_1.

[27] S. Hohenberger, A. Sahai, and B. Waters. Full domain hash from (leveled) multilinear maps and identity-based aggregate signatures. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pages 494–512, 2013. URL: https://doi.org/10.1007/978-3-642-40041-4_27, doi:10.1007/978-3-642-40041-4_27.

[28] S. Hohenberger and B. Waters. Short and stateless signatures from the RSA assumption. In *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, pages 654–670, 2009. URL: https://doi.org/10.1007/978-3-642-03356-8_38, doi:10.1007/978-3-642-03356-8_38.

[29] S. Hohenberger and B. Waters. Synchronized aggregate signatures from the RSA assumption. In J. B. Nielsen and V. Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, volume 10821 of *Lecture Notes in Computer Science*, pages 197–229. Springer, 2018. URL: https://doi.org/10.1007/978-3-319-78375-8_7, doi:10.1007/978-3-319-78375-8_7.

[30] T. Izu, M. Izumi, N. Kunihiro, and K. Ohta. Yet another sanitizable and deletable signatures. In *25th IEEE International Conference on Advanced Information Networking and Applications Workshops, WAINA 2011, Biopolis, Singapore, March 22-25, 2011*,

pages 574–579. IEEE Computer Society, 2011. URL: https://doi.org/10.1109/WAINA.2011.117, doi:10.1109/WAINA.2011.117.

[31] T. Izu, N. Kanaya, M. Takenaka, and T. Yoshioka. PIATS: A partially sanitizable signature scheme. In *Information and Communications Security, 7th International Conference, ICICS 2005, Beijing, China, December 10-13, 2005, Proceedings*, pages 72–83, 2005. URL: https://doi.org/10.1007/11602897_7, doi:10.1007/11602897_7.

[32] T. Izu, N. Kunihiro, K. Ohta, M. Sano, and M. Takenaka. Sanitizable and deletable signature. In K. Chung, K. Sohn, and M. Yung, editors, *Information Security Applications, 9th International Workshop, WISA 2008, Jeju Island, Korea, September 23-25, 2008, Revised Selected Papers*, volume 5379 of *Lecture Notes in Computer Science*, pages 130–144. Springer, 2008. URL: https://doi.org/10.1007/978-3-642-00306-6_10, doi:10.1007/978-3-642-00306-6_10.

[33] T. Izu, N. Kunihiro, K. Ohta, M. Sano, and M. Takenaka. Yet another sanitizable signature from bilinear maps. In *Proceedings of the The Forth International Conference on Availability, Reliability and Security, ARES 2009, March 16-19, 2009, Fukuoka, Japan*, pages 941–946. IEEE Computer Society, 2009. URL: https://doi.org/10.1109/ARES.2009.14, doi:10.1109/ARES.2009.14.

[34] T. Izu, N. Kunihiro, K. Ohta, M. Takenaka, and T. Yoshioka. A sanitizable signature scheme with aggregation. In E. Dawson and D. S. Wong, editors, *Information Security Practice and Experience, Third International Conference, ISPEC 2007, Hong Kong, China, May 7-9, 2007, Proceedings*, volume 4464 of *Lecture Notes in Computer Science*, pages 51–64. Springer, 2007. URL: https://doi.org/10.1007/978-3-540-72163-5_6, doi:10.1007/978-3-540-72163-5_6.

[35] R. Johnson, D. Molnar, D. X. Song, and D. A. Wagner. Homomorphic signature schemes. In B. Preneel, editor, *Topics in Cryptology - CT-RSA 2002, The Cryptographer's Track at the RSA Conference, 2002, San Jose, CA, USA, February 18-22, 2002, Proceedings*, volume 2271 of *Lecture Notes in Computer Science*, pages 244–262. Springer, 2002. URL: https://doi.org/10.1007/3-540-45760-7_17, doi:10.1007/3-540-45760-7_17.

[36] E. Kiltz, A. Mityagin, S. Panjwani, and B. Raghavan. Append-only signatures. In L. Caires, G. F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, editors, *Automata,*

Languages and Programming, 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005, Proceedings, volume 3580 of *Lecture Notes in Computer Science*, pages 434–445. Springer, 2005. URL: https://doi.org/10.1007/11523468_36, doi:10.1007/11523468_36.

[37] S. Krenn, H. C. Pöhls, K. Samelin, and D. Slamanig. Protean signature schemes. In J. Camenisch and P. Papadimitratos, editors, *Cryptology and Network Security - 17th International Conference, CANS 2018, Naples, Italy, September 30 - October 3, 2018, Proceedings*, volume 11124 of *Lecture Notes in Computer Science*, pages 256–276. Springer, 2018. URL: https://doi.org/10.1007/978-3-030-00434-7_13, doi:10.1007/978-3-030-00434-7_13.

[38] K. Lee, D. H. Lee, and M. Yung. Aggregating cl-signatures revisited: Extended functionality and better efficiency. In *Financial Cryptography and Data Security - 17th International Conference, FC 2013, Okinawa, Japan, April 1-5, 2013, Revised Selected Papers*, pages 171–188, 2013. URL: https://doi.org/10.1007/978-3-642-39884-1_14, doi:10.1007/978-3-642-39884-1_14.

[39] S. Lim, E. Lee, and C. Park. A short redactable signature scheme using pairing. *Security and Communication Networks*, 5(5):523–534, 2012. URL: https://doi.org/10.1002/sec.346, doi:10.1002/sec.346.

[40] S. Lu, R. Ostrovsky, A. Sahai, H. Shacham, and B. Waters. Sequential aggregate signatures and multisignatures without random oracles. In *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, pages 465–485, 2006. URL: https://doi.org/10.1007/11761679_28, doi:10.1007/11761679_28.

[41] A. Lysyanskaya, S. Micali, L. Reyzin, and H. Shacham. Sequential aggregate signatures from trapdoor permutations. In *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, pages 74–90, 2004. URL: https://doi.org/10.1007/978-3-540-24676-3_5, doi:10.1007/978-3-540-24676-3_5.

[42] A. Lysyanskaya, R. L. Rivest, A. Sahai, and S. Wolf. Pseudonym systems. In *Selected Areas in Cryptography, 6th Annual International Workshop, SAC'99, Kingston, Ontario,*

Canada, August 9-10, 1999, Proceedings, pages 184–199, 1999. URL: https://doi.org/10.1007/3-540-46513-8_14, doi:10.1007/3-540-46513-8_14.

[43] J. Ma, J. Liu, M. Wang, and W. Wu. An efficient and secure design of redactable signature scheme with redaction condition control. In M. H. A. Au, A. Castiglione, K. R. Choo, F. Palmieri, and K. Li, editors, *Green, Pervasive, and Cloud Computing - 12th International Conference, GPC 2017, Cetara, Italy, May 11-14, 2017, Proceedings*, volume 10232 of *Lecture Notes in Computer Science*, pages 38–52, 2017. URL: https://doi.org/10.1007/978-3-319-57186-7_4, doi:10.1007/978-3-319-57186-7_4.

[44] K. Miyazaki, G. Hanaoka, and H. Imai. Digitally signed document sanitizing scheme based on bilinear maps. In F. Lin, D. Lee, B. P. Lin, S. Shieh, and S. Jajodia, editors, *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2006, Taipei, Taiwan, March 21-24, 2006*, pages 343–354. ACM, 2006. URL: https://doi.org/10.1145/1128817.1128868, doi:10.1145/1128817.1128868.

[45] K. Miyazaki, G. Hanaoka, and H. Imai. Invisibly sanitizable digital signature scheme. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, 91-A(1):392–402, 2008. URL: https://doi.org/10.1093/ietfec/e91-a.1.392, doi:10.1093/ietfec/e91-a.1.392.

[46] K. Miyazaki, M. Iwamura, T. Matsumoto, R. Sasaki, H. Yoshiura, S. Tezuka, and H. Imai. Digitally signed document sanitizing scheme with disclosure condition control. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, 88-A(1):239–246, 2005. URL: http://search.ieice.org/bin/summary.php?id=e88-a_1_239&category=D&year=2005&lang=E&abst=.

[47] M. O. Ozmen, R. Behnia, and A. A. Yavuz. Fast authentication from aggregate signatures with improved security. In *Financial Cryptography and Data Security - 23rd International Conference, FC 2019, Frigate Bay, St. Kitts and Nevis, February 18-22, 2019, Revised Selected Papers*, pages 686–705, 2019. URL: https://doi.org/10.1007/978-3-030-32101-7_39, doi:10.1007/978-3-030-32101-7_39.

[48] H. C. Pöhls and K. Samelin. On updatable redactable signatures. In I. Boureanu, P. Owesarski, and S. Vaudenay, editors, *Applied Cryptography and Network Security - 12th International Conference, ACNS 2014, Lausanne, Switzerland, June 10-13, 2014. Proceedings*, volume 8479 of *Lecture Notes in Computer Science*, pages 457–

475. Springer, 2014. URL: https://doi.org/10.1007/978-3-319-07536-5_27, doi:10.1007/978-3-319-07536-5_27.

[49] H. C. Pöhls and K. Samelin. Accountable redactable signatures. In *10th International Conference on Availability, Reliability and Security, ARES 2015, Toulouse, France, August 24-27, 2015*, pages 60–69, 2015. URL: https://doi.org/10.1109/ARES.2015.10, doi:10.1109/ARES.2015.10.

[50] D. Pointcheval and O. Sanders. Reassessing security of randomizable signatures. In *Topics in Cryptology - CT-RSA 2018 - The Cryptographers' Track at the RSA Conference 2018, San Francisco, CA, USA, April 16-20, 2018, Proceedings*, pages 319–338, 2018. URL: https://doi.org/10.1007/978-3-319-76953-0_17, doi:10.1007/978-3-319-76953-0_17.

[51] K. Samelin, H. C. Pöhls, A. Bilzhause, J. Posegga, and H. de Meer. Redactable signatures for independent removal of structure and content. In M. D. Ryan, B. Smyth, and G. Wang, editors, *Information Security Practice and Experience - 8th International Conference, ISPEC 2012, Hangzhou, China, April 9-12, 2012. Proceedings*, volume 7232 of *Lecture Notes in Computer Science*, pages 17–33. Springer, 2012. URL: https://doi.org/10.1007/978-3-642-29101-2_2, doi:10.1007/978-3-642-29101-2_2.

[52] O. Sanders. Efficient redactable signature and application to anonymous credentials. In A. Kiayias, M. Kohlweiss, P. Wallden, and V. Zikas, editors, *Public-Key Cryptography - PKC 2020 - 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, May 4-7, 2020, Proceedings, Part II*, volume 12111 of *Lecture Notes in Computer Science*, pages 628–656. Springer, 2020. URL: https://doi.org/10.1007/978-3-030-45388-6_22, doi:10.1007/978-3-030-45388-6_22.

[53] O. Sanders. Improving revocation for group signature with redactable signature. In J. A. Garay, editor, *Public-Key Cryptography - PKC 2021 - 24th IACR International Conference on Practice and Theory of Public Key Cryptography, Virtual Event, May 10-13, 2021, Proceedings, Part I*, volume 12710 of *Lecture Notes in Computer Science*, pages 301–330. Springer, 2021. URL: https://doi.org/10.1007/978-3-030-75245-3_12, doi:10.1007/978-3-030-75245-3_12.

[54] A. Saxena, J. Misra, and A. Dhar. Increasing anonymity in bitcoin. In *Financial Cryptography and Data Security - FC 2014 Workshops, BITCOIN and WAHC*

2014, Christ Church, Barbados, March 7, 2014, Revised Selected Papers, pages 122–139, 2014. URL: https://doi.org/10.1007/978-3-662-44774-1_9, doi:10.1007/978-3-662-44774-1_9.

[55] D. Schröder. How to aggregate the CL signature scheme. In *Computer Security - ESORICS 2011 - 16th European Symposium on Research in Computer Security, Leuven, Belgium, September 12-14, 2011. Proceedings*, pages 298–314, 2011. URL: https://doi.org/10.1007/978-3-642-23822-2_17, doi:10.1007/978-3-642-23822-2_17.

[56] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979. URL: http://doi.acm.org/10.1145/359168.359176, doi:10.1145/359168.359176.

[57] R. Steinfeld, L. Bull, and Y. Zheng. Content extraction signatures. In K. Kim, editor, *Information Security and Cryptology - ICISC 2001, 4th International Conference Seoul, Korea, December 6-7, 2001, Proceedings*, volume 2288 of *Lecture Notes in Computer Science*, pages 285–304. Springer, 2001. URL: https://doi.org/10.1007/3-540-45861-1_22, doi:10.1007/3-540-45861-1_22.

[58] M. Tezuka and K. Tanaka. Redactable signature with compactness from set-commitment. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E104.A(9):1175–1187, 2021. doi:10.1587/transfun.2020DMP0013.

[59] Y. Zhao. Practical aggregate signature from general elliptic curves, and applications to blockchain. In *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security, AsiaCCS 2019, Auckland, New Zealand, July 09-12, 2019*, pages 529–538, 2019. URL: https://doi.org/10.1145/3321705.3329826, doi:10.1145/3321705.3329826.