Aggregate Signature with Detecting Functionality from Group Testing

Shingo Sato[1], Junji Shikata[2], and Tsutomu Matsumoto[2]

[1] National Institute of Information and Communications Technology (NICT), Japan
shingo-sato@nict.go.jp
[2] Graduate School of Environment and Information Sciences, and Institute of Advanced Sciences, Yokohama National University, Yokohama, Japan
shikata-junji-rb@ynu.ac.jp
tsutomu@ynu.ac.jp

Abstract. In this paper, we comprehensively study aggregate signatures with detecting functionality, that have functionality of both keyless aggregation of multiple signatures and identifying an invalid message from the aggregate signature, in order to reduce a total amount of signature-size for lots of messages. Our contribution is (i) to formalize strong security notions for both non-interactive and interactive protocols by taking into account related work such as fault-tolerant aggregate signatures and (non-)interactive aggregate MACs with detecting functionality (i.e., symmetric case); and (ii) to construct aggregate signatures with the functionality from group testing-protocols in a generic and comprehensive way. As instantiations, pairing-based constructions are provided.

Keywords: Aggregate signature · Digital signature · Group testing.

1 Introduction

Background and Related Work. Digital signature is a fundamental and important primitive in modern cryptography, and it has a wide range of applications that require integrity of data. In the era of IoT (Internet of Things), it is important to ensure integrity of data gathered from many and various IoT devices, however, it is often the case where a total amount of size of signatures for checking validity of big data is too large. From this viewpoint, we study techniques for the purpose of reducing a total amount of signature-size for many and various data, in particular, techniques of compressing (or aggregating) multiple signatures on data.

For the purpose mentioned above, Boneh et al. [4] proposed aggregate signatures and proposed a pairing-based scheme in the random oracle model (ROM). Assuming the weaker security model (i.e., certified-key model) in which signers have to prove knowledge of the secret key at key-registration, Rückert and Schröder gave an aggregate signature scheme using multilinear maps in the standard model [38]. Gentry and Ramzan proposed an identity-based aggregate

signature scheme in the ROM [16]. Hohenberger, Sahai, and Waters presented (identity-based) aggregate signature schemes using multilinear maps in the standard model [23]. Hartung et al. [21] proposed fault-tolerant aggregate signatures, that has functionality of both compressing multiple signatures and identifying an invalid message from the aggregate signature. There are other variants of aggregate signatures as follows: Sequential aggregate signatures in the ROM [31, 30, 3, 35, 6, 12, 28, 15] or in the standard model [30], and synchronized aggregate signatures in the ROM [16, 1, 24] or in the standard model [1], and fault-tolerant sequential aggregate signatures in the ROM [20] which is extended from [21] for sequential messages.

On the other hand, Katz and Lindell [26] proposed the aggregate message authentication code (AMAC) that can compress MAC-tags on multiple messages into a short aggregate tag. Hirose and Shikata [22] proposed AMAC that has the functionality of both compressing multiple MAC-tags into a short aggregate tag and identifying an invalid message from the aggregate tag. In [22], the model considers keyless aggregation like in [26], and the scheme is constructed from non-adaptive group testing in addition to the underlying MAC scheme in a generic way. Related work by applying non-adaptive group testing in symmetric-key cryptography includes [33, 34, 39, 36]. In addition, Sato and Shikata [40, 41] consider an interactive version of [22], and they constructed the interactive protocol from adaptive group testing in addition to the underlying MAC.

In this paper, we comprehensively study aggregate signatures with detecting functionality, that have functionality of both keyless aggregation of multiple signatures and identifying an invalid message from the aggregate signature, in order to reduce a total amount of signature-size for lots of messages. Our purpose is to formalize strong security notions considering related work mentioned above, especially for [21, 22, 40, 41], and to construct aggregate signatures with the functionality from group testing-protocols [10] in a generic and comprehensive way.

Contribution. The purpose of this paper is to formalize the model and security notions for both non-interactive and interactive protocols of aggregate signatures with detecting functionality, that have functionality of both keyless aggregation of multiple signatures and identifying an invalid message from the aggregate signature. In addition, we provide construction methodology for both protocols from group-testing protocols in a generic and comprehensive way. Specifically, the contribution of this paper is as follows.

1. We propose a formal model and security formalization of (non-interactive) aggregate signatures with detecting functionality (D-ASIG) that have functionality of both keyless aggregation of multiple signatures and identifying an invalid message from the aggregate signature. This kind of functionality was already proposed in fault-tolerant aggregate signatures in [21], however, the functionality of fault-tolerant aggregate signatures guarantees that valid messages must be regarded as valid from the aggregate signature even if some fault occurs, just like the property of error-correcting codes. On the other hand, D-ASIG in this paper guarantees that, from the aggregate signature,

(i) valid messages must be regarded as valid; and (ii) invalid messages must be regarded as invalid, even in presence of malicious adversary. This notion is formalized as identifiability in this paper; it consists of identifiability-completeness and identifiability-soundness; and identifiability-soundness can be weakened as identifiability-weak-soundness. To formalized identifiability, we extend the similar notion in the symmetric-key setting in [22] to our notion in the asymmetric setting for aggregate signatures.

2. We provide two kinds of generic constructions for D-ASIG: one is constructed from non-adaptive group-testing (NAGT) and aggregate signatures (ASIG), and we show that the resulting D-ASIG meets unforgeability, identifiability-completeness, and identifiability-weak-soundness; and the other is constructed from NAGT, ASIG and succinct non-interactive argument of knowledge (SNARK) (e.g., see [18]), and we show that the resulting D-ASIG meets unforgeability, identifiability-completeness, and identifiability-soundness. We also give instantiations of D-ASIG by using pairing.
3. We propose a formal model and security formalization of *interactive* aggregate signatures with detecting functionality (D-IASIG), that is, interactive protocols of aggregate signatures with detecting functionality. As in D-ASIG, we formalize identifiability even for D-IASIG in addition to unforgeability: we formalize identifiability-completeness, identifiability-soundness, and identifiability-weak-soundness along with the model of D-IASIG. For doing it, we extend the model of interactive protocols for aggregate MACs in the symmetric setting [40, 41] to the model of aggregate signatures in the asymmetric setting.
4. We provide two kinds of generic constructions for D-IASIG: one is constructed from adaptive group-testing (AGT) and ASIG, and we show that the resulting D-IASIG meets unforgeability, identifiability-completeness, and identifiability-weak-soundness; and the other is constructed from AGT, ASIG and SNARK, and we show that the resulting D-IASIG meets unforgeability, identifiability-completeness, and identifiability-soundness. We also give instantiations of D-IASIG by using pairing as in the case of D-ASIG.

Organization. The rest of this paper is organized as follows: In Section 2, we survey aggregate signature schemes in [4]. In Section 3, we briefly survey group-testing protocols, namely non-adaptive group-testing and adaptive group-testing. In Section 4, we propose a formal model and security formalization of aggregate signatures with detecting functionality (D-ASIG), and provide generic constructions of it. In addition, we give instantiations of D-ASIG based on pairing. In Section 5, we propose a formal model and security formalization of interactive aggregate signatures with detecting functionality (D-IASIG), and provide generic constructions of it. We also give instantiations of D-IASIG based on pairing. Finally, we conclude this paper in Section 6.

Notation. In this paper, we use the following: For a positive integer n, let $[n] := \{1, \ldots, n\}$. For n values $x_1, \ldots, x_n$ and a subset $I \subseteq [n]$ of indexes, let $(x_i)_{i \in I}$ be a sequence of elements of which indexes are in I, and let $\{x_i\}_{i \in I}$ be a set of elements of which indexes are in I. For a vector $\boldsymbol{x}$ with dimension n, let

x_i be the i-th entry ($i \in [n]$). For a $m \times n$ matrix $\boldsymbol{X}$, let $x_{i,j}$ be the entry at the i-th row and the j-th column ($i \in [m]$, $j \in [n]$). For a function $f : \mathbb{N} \to \mathbb{R}$, if $f(\lambda) = o(\lambda^{-c})$ for arbitrary positive c, then f is negligible in λ, and we write $\mathsf{negl}(\lambda)$. A probability is overwhelming if it is $1 - \mathsf{negl}(\lambda)$.

2 Aggregate Signature

In this section, we survey the model and construction of aggregate signature schemes in [4].

2.1 Syntax and Security Definition

An aggregate signature scheme ASIG consists of five polynomial-time algorithms $(\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy}, \mathsf{Agg}, \mathsf{AVrfy})$: For a security parameter λ, let $\mathcal{M} = \mathcal{M}(\lambda)$ be a message space.

Key Generation $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$:
A randomized algorithm KGen takes as input a security parameter 1^λ, and it outputs a public key pk and a secret key sk.

Signing $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{m})$:
A randomized or deterministic algorithm Sign takes as input a secret key sk and a message $\mathsf{m} \in \mathcal{M}$, and it outputs a signature $\sigma \in \mathcal{S}$.

Verification $1/0 \leftarrow \mathsf{Vrfy}(\mathsf{pk}, \mathsf{m}, \sigma)$:
A deterministic algorithm Vrfy takes as input a public key, a message $\mathsf{m} \in \mathcal{M}$, and a signature σ, and it outputs 1 or 0.

Aggregation $\widehat{\sigma} \leftarrow \mathsf{Agg}((\mathsf{pk}_1, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell, \sigma_\ell))$:
A randomized or deterministic algorithm Agg takes as input a tuple $(\mathsf{pk}_1, \mathsf{m}_1, \sigma_1)$, $\ldots$, $(\mathsf{pk}_\ell, \mathsf{m}_\ell, \sigma_\ell)$ of triplets of a public key, a message and a signature, and it outputs an aggregate signature $\widehat{\sigma}$.

Aggregate Verification $1/0 \leftarrow \mathsf{AVrfy}((\mathsf{pk}_1, \mathsf{m}_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell), \widehat{\sigma})$:
A deterministic algorithm AVrfy takes as input a tuple $(\mathsf{pk}_1, \mathsf{m}_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell)$ of pairs of a public key and a message, and an aggregate signature σ, and it outputs 1 or 0.

It is required that an aggregate signature scheme meets correctness as follows:

Definition 1 (Correctness). *An aggregate signature scheme* $\mathsf{ASIG} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy}, \mathsf{Agg}, \mathsf{AVrfy})$ *meets correctness if the following holds:*

- *For every* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$ *and every* $\mathsf{m} \in \mathcal{M}$, *it holds that* $\mathsf{Vrfy}(\mathsf{pk}, \mathsf{m}, \sigma) = 1$ *with at least probability* $1 - \mathsf{negl}(\lambda)$, *where* $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{m})$.
- *For any* $\ell = \mathsf{poly}(\lambda)$, *every* $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KGen}(1^\lambda)$, *and every* $\mathsf{m}_i \in \mathcal{M}$ *for* $i \in [\ell]$, *it holds that* $\mathsf{AVrfy}((\mathsf{pk}_1, \mathsf{m}_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell), \widehat{\sigma}) = 1$ *with at least probability* $1 - \mathsf{negl}(\lambda)$, *where* $\widehat{\sigma} \leftarrow \mathsf{Agg}((\mathsf{pk}_1, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell, \sigma_\ell))$ *and* $\sigma_i \leftarrow \mathsf{Sign}(\mathsf{sk}_i, \mathsf{m}_i)$ *for all* $i \in [\ell]$.

The security of aggregate signatures is defined as follows:

Definition 2 (EUF-CMA security). *An aggregate signature scheme* $\mathsf{ASIG} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy}, \mathsf{Agg}, \mathsf{AVrfy})$ *satisfies* EUF-CMA *security if for any PPT adversary* A *against* ASIG*, the advantage* $\mathsf{Adv}^{\text{euf-cma}}_{\mathsf{ASIG},\mathsf{A}}(\lambda) := \Pr[\mathsf{A}\ \textit{wins}]$ *is negligible in* λ*.* [A *wins*] *is the event that* A *wins in the following game:*

Setup. *A challenger generates a key-pair* $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KGen}(1^\lambda)$ *and sets* $\mathcal{Q} \leftarrow \emptyset$*. It gives* pk_1 *to* A*.*

Queries. *Given a signing-query* $\mathsf{m} \in \mathcal{M}$*, signing oracle* SIGN *returns* $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}_1, \mathsf{m})$ *and sets* $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{\mathsf{m}\}$*.*

Output. A *outputs a forgery* $((\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell), \widehat{\sigma})$*, where* $\gamma : [\ell] \to [\ell]$ *is a permutation.* A *wins if* $\mathsf{AVrfy}((\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell), \widehat{\sigma}) = 1$ *and* $\mathsf{m}_z \notin \mathcal{Q}$ *hold, where* $z \in [\ell]$ *is a message-index such that* $\gamma(z) = 1$*.*

2.2 Constructions

In this section, we describe the aggregate signature scheme [4]. Since this scheme is based on the co-Gap Diffie-Hellman (co-GDH) problem, we define several related problems in addition to co-GDH, and then we describe the scheme.

Bilinear Groups for Co-Diffie-Hellman. We define bilinear groups for co-Diffie-Hellman in order to describe the aggregate signature scheme in [4]. The following notation is used:

- G_1, G_2, and G_T are multiplicative cyclic groups of prime order p.
- g_1 and g_2 are generators of G_1 and G_2, respectively.
- ψ is a computable isomorphism from G_2 to G_1 with $\psi(g_2) = g_1$.
- e is a computable bilinear map $e : G_1 \times G_2 \to G_T$, namely, the map fulfills the following properties:
  - Bilinear: For all $u \in G_1$, $v \in G_2$, and $a, b \in \mathbb{Z}$, $e(u^a, v^b) = e(u, v)^{ab}$.
  - Non-degenerate: $e(g_1, g_2) \neq 1$.

First, in order to consider the case of $G_1 \neq G_2$, we define co-computational Diffie-Hellman (co-CDH) and co-decision Diffie-Hellman (co-DDH) problems as follows.

Definition 3 (Co-Computational Diffie-Hellman). *Given* $g_2, g_2^a \in G_2$ *and* $h \in G_1$*, compute* $h^a \in G_1$

Definition 4 (Co-Decision Diffie-Hellman). *Given* $g_2, g_2^a \in G_2$ *and* $h, h^b \in G_1$*, determine whether* $a = b$ *or not.*

In [5], it is known that when $G_1 = G_2$ and $g_1 = g_2$ holds, there are reductions from co-CDH and co-DDH to the standard CDH and DDH problems, respectively.

Next, we define co-GDH group pairs as follows.

Definition 5 (Decision Group Pair). *The pair* (G_1, G_2) *of two groups is a decision group pair for co-Diffie-Hellman if the group action on* G_1*, the group action on* G_2*, and the map* ψ *from* G_2 *to* G_1 *can be computed in one time unit, and decision co-Diffie-Hellman on* (G_1, G_2) *can be solved in one time unit.*

Definition 6 (Co-GDH Group Pair). *Assume that two groups G_1, G_2 are selected by following a security parameter λ. The advantage of a PPT algorithm* A *solving the* co-CDH *problem in groups G_1, G_2 is defined as*

$$\mathsf{Adv}_{\mathsf{A}}^{\text{co-cdh}}(\lambda) := \Pr[\mathsf{A}(g_2, g_2^a, h) \to h^a \mid a \xleftarrow{\$} \mathbb{Z}_p, h \xleftarrow{\$} G_1].$$

The pair (G_1, G_2) of two groups are a co-GDH *group pair if the pair is a decision group pair for co-Diffie-Hellman, and $\mathsf{Adv}_{\mathsf{A}}^{\text{co-cdh}}(\lambda) \leq \mathsf{negl}(\lambda)$ holds for any PPT algorithm* A.

Finally, we define bilinear group pairs for co-Diffie-Hellman, which are used in the aggregate signature scheme of [4].

Definition 7 (Bilinear Group Pair). *The pair (G_1, G_2) of two groups is a bilinear group pair if the group action on either can be computed in one time unit, the map ψ from G_2 to G_1 can be computed in one time unit, a bilinear map e is computable in one time unit.*

Definition 8 (Bilinear Group Pair for co-Diffie-Hellman). *Assume that two groups G_1, G_2 are selected by following a security parameter λ. A pair (G_1, G_2) of two groups is a bilinear group pair for co-Diffie-Hellman if it is a bilinear group pair and $\mathsf{Adv}_{\mathsf{A}}^{\text{co-cdh}}(\lambda) \leq \mathsf{negl}(\lambda)$ holds for any PPT algorithm* A.

Notice that if the pair (G_1, G_2) is a bilinear group pair for co-Diffie-Hellman, this pair is also co-GDH group pair.

Aggregate Signature in [4]. The aggregate signature scheme $\mathsf{ASIG}_{\mathsf{BGLS}} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy}, \mathsf{Agg}, \mathsf{AVrfy})$ is as follows: For a security parameter λ, set the following parameters: let $\{0,1\}^*$ be a message space. Let G_1, G_2 be the base groups, g_1 and g_2 be the generators of G_1 and G_2, respectively. Let ϕ be the computable isomorphism from G_2 to G_1, and $e : G_1 \times G_2 \to G_T$ be the bilinear map with target group G_T. Let $H : \{0,1\}^* \to G_1$ be a random oracle.

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$: Choose $x \xleftarrow{\$} \mathbb{Z}_p$ and compute $v \leftarrow g_2^x \in G_2$. Output $\mathsf{pk} = v$ and $\mathsf{sk} = x$.
- $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{m})$: Output $\sigma \leftarrow h^x \in G_1$, where $h \leftarrow H(\mathsf{m}) \in G_1$.
- $1/0 \leftarrow \mathsf{Vrfy}(\mathsf{pk}, \mathsf{m}, \sigma)$: Output 1 if $e(\sigma, g_2) = e(H(\mathsf{m}), v)$ holds, where $\mathsf{pk} = v$. Output 0 otherwise.
- $\widehat{\sigma} \leftarrow \mathsf{Agg}((\mathsf{pk}_1, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell, \sigma_\ell))$: Output $\widehat{\sigma} \leftarrow \prod_{i=1}^{\ell} \sigma_i \in G_1$.
- $1/0 \leftarrow \mathsf{AVrfy}((\mathsf{pk}_1, \mathsf{m}_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell), \widehat{\sigma})$: Output 1 if $e(\widehat{\sigma}, g_2) = \prod_{i=1}^{\ell} e(H(\mathsf{m}_i), v_i)$ holds, where $\mathsf{pk}_i = v_i$ for all $i \in [\ell]$. Output 0 otherwise.

The following proposition regarding the above scheme was proven in [4].

Proposition 1 ([4], Theorem 1). *If (G_1, G_2) is a bilinear group pair for co-Diffie-Hellman, then* $\mathsf{ASIG}_{\mathsf{BGLS}}$ *satisfies* EUF-CMA *security.*

3 Group-Testing Protocol

The first paper about group testing is published by Dorfman [9]. Group testing (e.g., [10]) is a method to specify positive items called *defectives* among many whole items with a small number of tests than the trivial individual testing for each item. The applications of group testing include screening blood samples for detecting a disease, and detecting clones which have a particular DNA sequence.

The group testing techniques are classified into two types: the first type means the testing techniques by non-adaptive strategies, called non-adaptive group testing [42, 37, 11], and the second type means the techniques by adaptive strategies, called adaptive group testing (or called sequential group testing) [9, 29, 25, 11]. Suppose that there are totally ℓ items of which there are (at most) d defectives. In non-adaptive group testing, we need to know d beforehand and to select all the subsets of ℓ items to be tested without knowing the results of other tests. On the other hand, in adaptive group testing, we can do tests several times such that we can select a subset of items to be tested after observing the result of the previous test. In particular, a competitive group testing is an adaptive group testing which does not need to know d (i.e., the number of defectives) beforehand, and this type of testing is useful in a real application when it is not easy to estimate d beforehand.

3.1 Non-Adaptive Group-Testing

Non-adaptive group-testing is typically designed by providing a d-disjunct matrix, a d-cover-free family, or a d-separable matrix (e.g., see [10]). And, a non-adaptive group-testing protocol with u tests for ℓ items is represented by a $u \times \ell$ binary matrix, and the (i, j)-th element of the matrix is equal to 1 if and only if the i-th test is executed to the j-th item. Among such matrices for representing non-adaptive group-testing, a disjunct matrix (or cover-free family) is well studied in combinatorics and bioinformatics, and it is defined as follows.

Definition 9 (d-disjunct). *A matrix $\boldsymbol{G} = [\boldsymbol{g}_1, \ldots, \boldsymbol{g}_\ell] \in \{0,1\}^{u \times \ell}$ is d-disjunct if for any d columns $\boldsymbol{g}_{s_1}, \ldots, \boldsymbol{g}_{s_d}$ and every $\bar{\boldsymbol{g}} \in \{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_\ell\} \backslash \{\boldsymbol{g}_{s_1}, \ldots, \boldsymbol{g}_{s_d}\}$ $(s_1, \ldots, s_d \in [\ell])$, there exists $z \in [u]$ such that $u_z < \bar{g}_z$, where let $\boldsymbol{u} \leftarrow \bigvee_{i=1}^{d} \boldsymbol{g}_{s_i}$, and $\bigvee$ is the bitwise-OR operation.*

Equivalently, the d-disjunct matrix is stated as follows:

Definition 10. *A $u \times \ell$ binary matrix G is a d-disjunct matrix, if for arbitrary $d+1$ columns selected from the matrix, the resulting $u \times (d+1)$ matrix contains all the unit vectors with length $d+1$ in its rows.*

By using a d-disjunct matrix, a non-adaptive group-testing protocol can efficiently detect at most d positive items. We simply describe the process of group-testing protocol with a d-disjunct matrix $\boldsymbol{G} \in \{0,1\}^{u \times \ell}$ as follows: Let $S_i(\boldsymbol{G}) = \{j \mid j \in [\ell] \wedge g_{i,j} = 1\}$ for $i \in [u]$ and $\boldsymbol{G} \in \{0,1\}^{u \times \ell}$.

1. Let $J \leftarrow \{1, 2, \ldots, \ell\}$ be a set of indexes of all items.

2. For each $i \in [u]$, compress items with indexes in $S_i(\boldsymbol{G})$.
3. For each $i \in [u]$, set $J \leftarrow J \backslash S_i(\boldsymbol{G})$ if the test result of the i-th compressed item is negative.
4. Output J.

Then, J is the set of indexes of all positive items due to the d-disjunct property of $\boldsymbol{G}$.

In this paper, we mainly deal with non-adaptive group-testing based on disjuct matrices and its application to (non-interactive) aggregate signatures with detecting functionality (D-ASIG) in Section 4.2, however, other types of non-adaptive group-testing such as separable matrices can be applied to D-ASIG in a similar way. We note that the number of tests required in non-adaptive group testing using d-disjunct matrices is $O(d^2 \log \ell)$ (see [10]), and it is expected that the number of signatures can be reduced to $O(d^2 \log \ell)$ in D-ASIG instead of checking ℓ signatures.

3.2 Adaptive Group-Testing

The advantage of adaptive group-testing lies in specifying positive items (defectives) with a smaller number of tests than non-adaptive group-testing, while adaptive group-testing requires interactive communications between the entity who selects a subset to be tested and the entity who actually executes the tests. In this paper, we need the formal model of adaptive group testing (AGT), since we want to deal with AGT in a comprehensive way to construct interactive aggregate signatures with detecting functionality (D-IASIG) in a generic way. Hence, in this section, we describe the formal model of AGT provided in [41] and use it to construct D-IASIG in Section 5.2. As interesting AGT, we can consider the binary search algorithm [10], rake-and-winnow algorithm [11], Li's s-stage algorithm [29], and digging algorithm [10], since the number of tests required in them is $O(d \log \frac{\ell}{d})$ that is asymptotically optimal (see Lemma 4.5.2 of [10]). Therefore, it is expected that the number of signatures can be reduced to $O(d \log \frac{\ell}{d})$ that is much smaller than that of non-adaptive group-testing.

AGT can be regarded as an N-stage interactive protocol between two polynomial-time algorithms ($N \geq 2$). In order to define AGT protocols, we assume that

- a relation $R \subseteq \mathcal{X} \times \mathcal{Y}$ and an ID space $\mathcal{ID}$ are defined,
- an item with an index i is a pair of an ID $\mathsf{id}_i \in \mathcal{ID}$ and a value $\mathsf{x}_i \in \mathcal{X}$,
- we consider verification-items which determine whether items with the corresponding IDs are positive or negative. A verification-item with an index i is a pair of an ID $\mathsf{id}_i \in \mathcal{ID}$ and a verification-value $\mathsf{v}_i \in \mathcal{Y}$, and
- an item $(\mathsf{id}_i, \mathsf{x}_i)$ is negative if $(\mathsf{x}_i, \mathsf{v}_i) \notin R$, and is positive if $(\mathsf{x}_i, \mathsf{v}_i) \in R$.

For example, we consider the case of a pseudorandom function $F : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{T}$. Let $\mathcal{X} := \mathcal{M} \times \mathcal{T}$, $\mathcal{Y} := \mathcal{K}$, and $\mathcal{ID} := \{0,1\}^*$. We define $R = \{((\mathsf{m}, \mathsf{t}), \mathsf{k}) \in \mathcal{X} \times \mathcal{Y} \mid F(\mathsf{k}, \mathsf{m}) \neq \mathsf{t}\}$ as a relation. Then, an item with an index i is $(\mathsf{id}_i, \mathsf{m}_i, \mathsf{t}_i)$, and a verification-item with an index i is $(\mathsf{id}_i, \mathsf{k}_i)$, where $\mathsf{id}_i \in \mathcal{ID}$, $(\mathsf{m}_i, \mathsf{t}_i) \in \mathcal{X}$, and $\mathsf{k}_i \in \mathcal{Y}$. In addition, an item $(\mathsf{id}_i, \mathsf{m}_i, \mathsf{t}_i)$ is positive if $((\mathsf{m}_i, \mathsf{t}_i), \mathsf{k}_i) \in R$,

namely, $F(\mathsf{k}_i, \mathsf{m}_i) \neq \mathsf{t}_i$ holds. An item $(\mathsf{id}_i, \mathsf{m}_i, \mathsf{t}_i)$ is negative if $F(\mathsf{k}_i, \mathsf{m}_i) = \mathsf{t}_i$ holds.

An N-stage group-testing protocol for a relation $R \subseteq \mathcal{X} \times \mathcal{Y}$ is an interactive protocol between two entities: group-selector GS and inspector IS with three polynomial-time algorithms (GSel, Coms, Test). The following sets are defined:

- $\mathcal{ID} = \{0,1\}^\lambda$ is a ID space for a system parameter λ.
- $\mathcal{I} = \{(\mathsf{id}_1, \mathsf{x}_1), \ldots, (\mathsf{id}_\ell, \mathsf{x}_\ell)\}$ is a set of all tested items ($\mathsf{id}_i \in \mathcal{ID}$ and $\mathsf{x}_i \in \mathcal{X}$ for all $i \in [\ell]$).
- $\mathcal{V} = \{(\mathsf{id}_1, \mathsf{v}_1), \ldots, (\mathsf{id}_\ell, \mathsf{v}_\ell)\}$ is a set of all verification-items, ($\mathsf{id}_i \in \mathcal{ID}$ and $\mathsf{v}_i \in \mathcal{Y}$ for all $i \in [\ell]$).
- For a subset $\mathcal{G} \subseteq \mathcal{I}$, let $\mathcal{V}(\mathcal{G}) := \{(\mathsf{id}_i, \mathsf{v}_i) \mid i \in [\ell] \wedge (\mathsf{id}_i, \mathsf{x}_i) \in \mathcal{G}\}$.

Polynomial-time algorithms $(\mathsf{GSel}, \mathsf{Coms}, \mathsf{Test})$ are defined as follows:

Group Selection $(\{\mathcal{G}_1^{(s)}, \ldots, \mathcal{G}_{u^{(s)}}^{(s)}\}, \mathsf{st}^{(s)}) \leftarrow \mathsf{GSel}(J^{(s-1)}, \{\mathcal{G}_1^{(s-1)}, \ldots, \mathcal{G}_{u^{(s-1)}}^{(s-1)}\}, \mathsf{st}^{(s-1)})$: A deterministic algorithm GSel takes as input a set $J^{(s-1)} \subseteq \mathcal{I}$, subsets $\mathcal{G}_1^{(s-1)}, \ldots, \mathcal{G}_{u^{(s-1)}}^{(s-1)}$ of $J^{(s-1)}$, and the current internal state $\mathsf{st}^{(s-1)}$. It outputs new subsets $\mathcal{G}_1^{(s)}, \ldots, \mathcal{G}_{u^{(s)}}^{(s)}$ ($\mathcal{G}_i^{(s)} \subseteq \mathcal{I}$ for $i \in [u^{(s)}]$), and the next internal state $\mathsf{st}^{(s)}$.

Compression $\mathsf{com} \leftarrow \mathsf{Coms}(\mathcal{G})$: A deterministic or randomized algorithm Coms takes as input a subset $\mathcal{G} \subseteq \mathcal{I}$, and it outputs a compressed item-value com.

Testing $1/0 \leftarrow \mathsf{Test}(\mathcal{V}(\mathcal{G}), \mathsf{com})$: A deterministic algorithm Test takes as input a subset $\mathcal{V}(\mathcal{G}) \subseteq \mathcal{V}$, and it outputs 1 or 0.

We describe the N-stage adaptive group-testing protocol $\mathsf{AGT}(\mathcal{I}, \mathcal{V}) = \langle \mathsf{GS}(\mathcal{I}), \mathsf{IS}(\mathcal{V}) \rangle$ for R. Let $u^{(0)} = 1$, $\mathcal{G}_1^{(0)} = \mathcal{I}$, $J^{(0)} = \mathcal{I}$, and $\mathsf{st}^{(0)} = \emptyset$ at the setup phase. The s-th stage process of GS and IS is given as follows ($s \in [N]$):

$\mathsf{GS}(\mathcal{I})$: If $s = 1$, then start from Step 2.

Step 1: Receive $(b_1^{(s-1)}, \ldots, b_{u^{(s-1)}}^{(s-1)}) \in \{0,1\}^{u^{(s-1)}}$ from IS, and set $J^{(s-1)} \leftarrow J^{(s-2)} \backslash \{(\mathsf{id}, \mathsf{x}) \mid i \in [u^{(s-1)}] \wedge b_i^{(s-1)} = 1 \wedge (\mathsf{id}, \mathsf{x}) \in \mathcal{G}_i^{(s-1)}\}$.

- If $s > N$ or $J^{(s-1)} = \emptyset$, then halt
- Otherwise, move to Step 2.

Step 2: $(\{\mathcal{G}_1^{(s)}, \ldots, \mathcal{G}_{u^{(s)}}^{(s)}\}, \mathsf{st}^{(s)}) \leftarrow \mathsf{GSel}(J^{(s-1)}, \{\mathcal{G}_1^{(s-1)}, \ldots, \mathcal{G}_{u^{(s-1)}}^{(s-1)}\}, \mathsf{st}^{(s-1)})$.

Step 3: For each $i \in [u^{(s)}]$, $\mathsf{com}_i^{(s)} \leftarrow \mathsf{Coms}(\mathcal{G}_i^{(s)})$.

Step 4: Send $(\mathsf{com}_1^{(s)}, \ldots, \mathsf{com}_{u^{(s)}}^{(s)})$ to IS.

$\mathsf{IS}(\mathcal{V})$:

Step 1: Receive $(\mathsf{com}_1^{(s)}, \ldots, \mathsf{com}_{u^{(s)}}^{(s)})$ from GS.

Step 2: $(\{\mathcal{G}_1^{(s)}, \ldots, \mathcal{G}_{u^{(s)}}^{(s)}\}, \mathsf{st}^{(s)}) \leftarrow \mathsf{GSel}(J^{(s-1)}, \{\mathcal{G}_1^{(s-1)}, \ldots, \mathcal{G}_{u^{(s-1)}}^{(s-1)}\}, \mathsf{st}^{(s-1)})$.

Step 3: Let $J^{(s)} = J^{(s-1)}$, and for each $i \in [u^{(s)}]$, do the following:

(3-1): $b_i^{(s)} \leftarrow \mathsf{Test}(\mathcal{V}(\mathcal{G}_i^{(s)}), \mathsf{com}_i^{(s)})$.

(3-2): $J^{(s)} \leftarrow J^{(s)} \backslash \{(\mathsf{id}, \mathsf{x}) \mid b_i^{(s)} = 1 \wedge (\mathsf{id}, \mathsf{x}) \in \mathcal{G}_i^{(s)}\}$.

Step 4: Send $(b_1^{(s)}, \ldots, b_{u^{(s)}}^{(s)}) \in \{0,1\}^{u^{(s)}}$ to GS. If $s = N$ or $J^{(s)} = \emptyset$, then output $J = J^{(s)}$ and halt.

It is required that AGT protocols meet the following properties: correctness, GT-completeness, and GT-soundness.

Definition 11 (Correctness). *An adaptive group-testing protocol* $\mathsf{AGT} = (\mathsf{IS}, \mathsf{GS})$ *with* $(\mathsf{GSel}, \mathsf{Coms}, \mathsf{Test})$ *meets correctness if the following conditions hold:*

- *For any subset* $\mathcal{G} \subseteq \mathcal{I}$ *of items* $(\mathsf{id}_i, \mathsf{x}_i)$ *such that* $(\mathsf{id}_i, \mathsf{v}_i) \in \mathcal{V}$ *and* $(\mathsf{x}_i, \mathsf{v}_i) \notin R$*, it holds that* $\mathsf{Test}(\mathcal{V}(\mathcal{G}), \mathsf{com}) = 1$*, where* $\mathsf{com} \leftarrow \mathsf{Coms}(\mathcal{G})$.
- *Suppose that* $(\mathsf{x}_i, \mathsf{v}_i) \notin R$ *holds for every* $(\mathsf{id}_i, \mathsf{x}_i, \mathsf{v}_i)$ *with* $(\mathsf{id}_i, \mathsf{x}_i) \in \mathcal{I}$ *and* $(\mathsf{id}_i, \mathsf{v}_i) \in \mathcal{V}$*. It holds that* $\mathsf{AGT}(\mathcal{I}, \mathcal{V}) = \emptyset$ *if* GS *and* IS *correctly follow the protocol* AGT.

In order to formalize GT-completeness and GT-soundness, we define $D = \{(\mathsf{id}_i, \mathsf{x}_i) \mid i \in [\ell] \wedge (\mathsf{x}_i, \mathsf{v}_i) \in R\}$ as a set of positive items, and $\bar{D} = \{(\mathsf{id}_i, \mathsf{x}_i) \mid i \in [\ell] \wedge (\mathsf{x}_i, \mathsf{v}_i) \notin R\}$ as a set of negative items. Besides, let $\bar{J} = \mathcal{I} \backslash J$, where $J \leftarrow \mathsf{AGT}(\mathcal{I}, \mathcal{V})$.

Definition 12 (GT-completeness). *Suppose that* GS *and* IS *follow the protocol* AGT*. An adaptive group-testing protocol* AGT *with* $(\mathsf{GSel}, \mathsf{Coms}, \mathsf{Test})$ *meets GT-completeness if for* $J \leftarrow \mathsf{AGT}(\mathcal{I}, \mathcal{V})$*, we have* $\bar{D} \subseteq \bar{J}$ *with at least probability* $1 - \mathsf{negl}(\lambda)$.

Definition 13 (GT-soundness). *Suppose that* GS *and* IS *follow the protocol* AGT*. An adaptive group-testing protocol* AGT *with* $(\mathsf{GSel}, \mathsf{Coms}, \mathsf{Test})$ *meets GT-soundness if for* $J \leftarrow \mathsf{AGT}(\mathcal{I}, \mathcal{V})$*, we have* $D \subseteq J$ *with at least probability* $1 - \mathsf{negl}(\lambda)$.

4 Aggregate Signature with Detecting Functionality

4.1 Model and Security Definition

In this section, we introduce a model and security formalization of an aggregate signature scheme with detecting functionality (D-ASIG scheme for short) that has functionality of both keyless aggregation of multiple signatures and identifying an invalid message from the aggregate signature. This kind of functionality was already proposed in fault-tolerant aggregate signatures in [21], however, the functionality of fault-tolerant aggregate signatures guarantees that valid messages must be regarded as valid from the aggregate signature even if some fault occurs, just like the property of error-correcting codes. On the other hand, D-ASIG in this paper guarantees that, from the aggregate signature, (i) valid messages must be regarded as valid; and (ii) invalid messages must be regarded as invalid, even in presence of malicious adversary. This notion is formalized as identifiability which will be given later.

Formally, an aggregate signature scheme with detecting functionality (D-ASIG) consists of five polynomial-time algorithms (KGen, Sign, Vrfy, DAgg, DVrfy): For a security parameter λ, let $\mathcal{M} = \mathcal{M}(\lambda)$ be a message space.

Key Generation $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$: A randomized algorithm KGen takes as input a security parameter 1^λ, and it outputs a public key pk and a secret key sk.

Signing $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{m})$: A randomized or deterministic algorithm Sign takes as input a secret key sk and a message $\mathsf{m} \in \mathcal{M}$, and it outputs a signature σ.

Verification $1/0 \leftarrow \mathsf{Vrfy}(\mathsf{pk}, \mathsf{m}, \sigma)$: A deterministic algorithm Vrfy takes as input a public key pk, a message $\mathsf{m} \in \mathcal{M}$, and a signature σ, and it outputs 1 or 0.

Detectable Aggregation $(\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u) \leftarrow \mathsf{DAgg}(\boldsymbol{G}, ((\mathsf{pk}_1, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell, \sigma_\ell)))$: A randomized or deterministic algorithm DAgg takes as input a d-disjunct matrix $\boldsymbol{G} \in \{0,1\}^{u \times \ell}$, a tuple $((\mathsf{pk}_1, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell, \sigma_\ell))$ of triplets of a public key, a message, and a signature, and it outputs a tuple $(\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u)$ of aggregate signatures.

Detectable Verification $J \leftarrow \mathsf{DVrfy}(\boldsymbol{G}, ((\mathsf{pk}_1, \mathsf{m}_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell)), (\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u))$: A deterministic algorithm DVrfy takes as input a d-disjunct matrix $\boldsymbol{G} \in \{0,1\}^{u \times \ell}$, a tuple $((\mathsf{pk}_1, \mathsf{m}_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell))$ of pairs of a public key and a message, and a tuple $(\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u)$ of aggregate signatures, and it outputs a set J of pairs of a public key and a message.

It is required that D-ASIG scheme meets correctness as follows:

Definition 14 (Correctness). *A D-ASIG scheme* D-ASig = (KGen, Sign, Vrfy, DAgg, DVrfy) *satisfies correctness if the following conditions hold:*

- *For every* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$ *and every* $\mathsf{m} \in \mathcal{M}$*, it holds that* $\mathsf{Vrfy}(\mathsf{pk}, \mathsf{m}, \sigma) = 1$ *with overwhelming probability, where* $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{m})$.
- *For every d-disjunct matrix* $\boldsymbol{G} \in \{0,1\}^{u \times \ell}$*, for every* $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KGen}(1^\lambda)$*, and every* $\mathsf{m}_i \in \mathcal{M}$ *for all* $i \in [\ell]$*, it holds that*

$$\mathsf{DVrfy}(\boldsymbol{G}, ((\mathsf{pk}_1, \mathsf{m}_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell)), (\widehat{\sigma}_1, \ldots, \widehat{\sigma}_\ell)) = \emptyset$$

with overwhelming probability, where $(\widehat{\sigma}_1, \ldots, \widehat{\sigma}_\ell) \leftarrow \mathsf{DAgg}(\boldsymbol{G}, ((\mathsf{pk}_1, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell, \sigma_\ell)))$ *and* $\sigma_i \leftarrow \mathsf{Sign}(\mathsf{pk}_i, \mathsf{m}_i)$ *for all* $i \in [\ell]$.

We define security notions of D-ASIG, EUF-CMA security and identifiablity. EUF-CMA security is formalized as in [21]:

Definition 15 (EUF-CMA security). *A D-ASIG scheme* D-ASig = (KGen, Sign, Vrfy, DAgg, DVrfy) *satisfies* EUF-CMA *security for any PPT adversary* A *against* D-ASig*, the advantage* $\mathsf{Adv}^{\text{euf-cma}}_{\text{D-ASig,A}}(\lambda) := \Pr[\mathsf{A}\ \textit{wins}]$ *is negligible in* λ. [A *wins*] *is the event that* A *wins in the following game:*

Setup. *A challenger generates a key-pair* $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KGen}(1^\lambda)$ *and sets* $\mathcal{Q} \leftarrow \emptyset$*. It gives* pk_1 *to* A.

Queries. A *is allowed to access the following oracle:*

- SIGN*: Given a signing-query* $\mathsf{m} \in \mathcal{Q}$*, signing oracle* SIGN *returns* $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}_1, \mathsf{m})$ *and sets* $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{\mathsf{m}\}$.

Output. A *outputs* $(\boldsymbol{G}, ((\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell)), (\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u))$*, where* $\gamma : [\ell] \to [\ell]$ *is a permutation. The challenger computes* $J \leftarrow \mathsf{DVrfy}(\boldsymbol{G}, ((\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell)), (\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u))$. A *wins if* $(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \notin J$ *and* $\mathsf{m}_z \notin \mathcal{Q}$ *hold, where* $z \in [\ell]$ *is a message-index such that* $\gamma(z) = 1$.

We newly introduce the notion of identifiablity in the context of aggregate signatures: identifiablity guarantees that, from the aggregate signature, (i) valid messages must be regarded as valid; and (ii) invalid messages must be regarded as invalid, even in presence of malicious adversary. This notion is formalized as follows.

Definition 16 (Identifiability). *Regarding the* identifiability *of D-ASIG scheme* $\mathsf{D\text{-}ASig} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy}, \mathsf{DAgg}, \mathsf{DVrfy})$*, we define* identifiability-completeness *and* identifiability-soundness*. We consider the following security game: Let* A *be a PPT adversary against* D-ASig*.*

Setup. *The challenger generates a key-pair* $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KGen}(1^\lambda)$ *and sets* $\mathcal{Q} \leftarrow \emptyset$*. It gives* pk_1 *to* A*.*

Queries. A *is allowed to access the following oracle:*

- SIGN*: Given a sign-query* $\mathsf{m} \in \mathcal{Q}$*, signing oracle* SIGN *returns* $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}_1, \mathsf{m})$ *and sets* $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{\mathsf{m}\}$.

Output. A *outputs* $(\boldsymbol{G}, (\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell, \sigma_\ell))$*, where* $\gamma : [\ell] \to [\ell]$ *is a permutation. The challenger computes* $(\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u) \leftarrow \mathsf{DAgg}(\boldsymbol{G}, (\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell, \sigma_\ell))$ *and* $J \leftarrow \mathsf{DVrfy}(\boldsymbol{G}, (\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell)), (\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u))$.

identifiability-completeness *and* identifiability-soundness *are defined as follows: For a set* $\{(\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell, \sigma_\ell)\}$*, let* $D = \{(\mathsf{pk}_{\gamma(i)}, \mathsf{m}_i) \mid i \in [\ell] \wedge \mathsf{Vrfy}(\mathsf{pk}_{\gamma(i)}, \mathsf{m}_i, \sigma_i) = 0\}$*, and* $\bar{D} = \{(\mathsf{pk}_{\gamma(i)}, \mathsf{m}_i) \mid i \in [\ell] \wedge \mathsf{Vrfy}(\mathsf{pk}_{\gamma(i)}, \mathsf{m}_i, \sigma_i) = 1\}$*. Let* $z \in [\ell]$ *be a message-index such that* $\gamma(z) = 1$*.*

- ***Completeness:*** D-ASig *satisfies* identifiability-completeness *if for any PPT adversary* A*, the following advantage is negligible in* λ*:*

$$\mathsf{Adv}^{\mathrm{complete}}_{\mathsf{D\text{-}ASig},\mathsf{A}}(\lambda) := \Pr\left[(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \in \bar{D} \cap J\right].$$

- ***Soundness:*** D-ASig *satisfies* identifiability-soundness *if for any PPT adversary* A*, the following advantage is negligible in* λ*:*

$$\mathsf{Adv}^{\mathrm{sound}}_{\mathsf{D\text{-}ASig},\mathsf{A}}(\lambda) := \Pr\left[(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \in D \backslash J\right].$$

In addition, identifiability-weak-soundness *is defined in the same way as the definition of* identifiability-soundness *except that the advantage of a PPT adversary* A *is defined as*

$$\mathsf{Adv}^{\text{w-sound}}_{\mathsf{D\text{-}ASig},\mathsf{A}}(\lambda) := \Pr\left[(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \in D' \backslash J\right],$$

where $D' = \{(\mathsf{pk}_{\gamma(i)}, \mathsf{m}_i) \mid i \in [\ell] \wedge \mathsf{Vrfy}(\mathsf{pk}_{\gamma(i)}, \mathsf{m}_i, \sigma_i) = 0 \wedge \mathsf{m}_i \notin \mathcal{Q}\}$.

The following proposition shows the relation between EUF-CMA security and identifiability-weak-soundness.

Proposition 2. *If a D-ASIG scheme* D-ASig *meets* EUF-CMA *security, then it also satisfies* identifiability-weak-soundness.

Proof. By using a PPT adversary A breaking identifiability-weak-soundness, we construct a PPT algorithm F^{euf} breaking EUF-CMA security, as follows: It takes as input the public key pk_1 of D-ASig and gives pk_1 to A. By using the given signing oracle SIGN^{euf}, F^{euf} simulates the oracle access of A in the straightforward way. When A outputs $(\boldsymbol{G}, (\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell, \sigma_\ell))$, F^{euf} computes $(\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u) \leftarrow \mathsf{DAgg}(\boldsymbol{G}, (\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell, \sigma_\ell))$, and outputs $(\boldsymbol{G}, ((\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell)), (\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u))$.

If the output of A fulfills $(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \in \{(\mathsf{pk}_{\gamma(i)}, \mathsf{m}_i) \mid i \in [\ell] \wedge \mathsf{Vrfy}(\mathsf{pk}_{\gamma(i)}, \mathsf{m}_i, \sigma_i) = 0 \wedge \mathsf{m}_i \notin \mathcal{Q}\} \backslash J$ for $z \in [\ell]$ such that $\gamma(z) = 1$, then the output of F^{euf} is a forgery of EUF-CMA security game since $(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \notin J$ and $\mathsf{m}_z \notin \mathcal{Q}$ hold. Thus, the output of F^{euf} is a valid forgery, and we obtain $\mathsf{Adv}^{\text{w-sound}}_{\text{D-ASig},\mathsf{A}}(\lambda) \leq \mathsf{Adv}^{\text{euf-cma}}_{\text{D-ASig},\mathsf{F}^{euf}}(\lambda)$. □

4.2 Generic Construction from Non-Adaptive Group-Testing

Construction I. We present a generic construction of D-ASIG starting from any aggregate signature scheme and any d-disjunct matrix, and prove that the resulting D-ASIG scheme satisfies EUF-CMA security, identifiability-completeness, and identifiability-weak-soundness. By applying a non-adaptive group-testing based on a d-disjunct matrix to DAgg and DVrfy algorithms, it is possible to detect invalid pairs of a public key and a message.

Our construction $\text{D-ASig}_1 = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy}, \mathsf{DAgg}, \mathsf{DVrfy})$ is as follows: Let $\mathsf{ASIG} = (\mathsf{KGen}^{asig}, \mathsf{Sign}^{asig}, \mathsf{Vrfy}^{asig}, \mathsf{Agg}^{asig}, \mathsf{AVrfy}^{asig})$ be an aggregate signature scheme. For $\boldsymbol{G} \in \{0,1\}^{u \times \ell}$ and $i \in [u]$, let $S_i(\boldsymbol{G}) = \{j \mid j \in [\ell] \wedge g_{i,j} = 1\}$.

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$: Output $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}^{asig}(1^\lambda)$.
- $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{m})$: Output $\sigma \leftarrow \mathsf{Sign}^{asig}(\mathsf{sk}, \mathsf{m})$.
- $1/0 \leftarrow \mathsf{Vrfy}(\mathsf{pk}, \mathsf{m}, \sigma)$: Output $1/0 \leftarrow \mathsf{Vrfy}^{asig}(\mathsf{pk}, \mathsf{m}, \sigma)$.
- $(\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u) \leftarrow \mathsf{DAgg}(\boldsymbol{G}, ((\mathsf{pk}_1, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell, \sigma_\ell)))$:
  1. For each $i \in [u]$, generate $\widehat{\sigma}_i \leftarrow \mathsf{Agg}^{asig}((\mathsf{pk}_k, \mathsf{m}_k, \sigma_k)_{k \in S_i(\boldsymbol{G})})$.
  2. Output $(\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u)$.
- $J \leftarrow \mathsf{DVrfy}(\boldsymbol{G}, ((\mathsf{pk}_1, \mathsf{m}_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell)), (\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u))$:
  1. $J \leftarrow \{(\mathsf{pk}_1, \mathsf{m}_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell)\}$.
  2. For each $i \in [u]$, if $\mathsf{AVrfy}^{asig}((\mathsf{pk}_k, \mathsf{m}_k)_{k \in S_i(\boldsymbol{G})}, \widehat{\sigma}_i) = 1$ holds, then set $J \leftarrow J \backslash \{(\mathsf{pk}_k, \mathsf{m}_k, \sigma_k)\}_{k \in S_i(\boldsymbol{G})}$.
  3. Output J.

Theorem 1 and 2 show the security of D-ASig_1.

Theorem 1. *If an aggregate signature scheme* ASIG *meets* EUF-CMA *security, then the resulting D-ASIG scheme* D-ASig_1 *satisfies* EUF-CMA *security.*

Proof. We prove the theorem by constructing a PPT algorithm F_I^{asig} breaking the EUF-CMA security of ASIG. F_I^{asig} is constructed as follows: In **Setup** phase, it takes as input the public key pk_1 of ASIG. It sets $\mathcal{Q} \leftarrow \emptyset$ and gives pk_1 to A. SIGN oracle is simulated as follows:

– SIGN: Given a sign-query $\mathsf{m} \in \mathcal{M}$, issue m to the given signing oracle SIGN^{asig} and get the response σ. Return σ and set $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{\mathsf{m}\}$.

When A outputs $((\boldsymbol{G}, ((\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell)), (\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u))$, then F_I^{asig} computes $J \leftarrow \mathsf{DVrfy}(\boldsymbol{G}, ((\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell)), (\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u))$. If there exits a message-index $z \in [\ell]$ such that $\gamma(z) = 1$, $(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z, \sigma_z) \notin J$, and $\mathsf{m}_z \notin \mathcal{Q}$, then it checks whether there exits $i \in [u]$ such that $z \in S_i(\boldsymbol{G})$ and $\mathsf{AVrfy}^{asig}((\mathsf{pk}_{\gamma(j)}, \mathsf{m}_j)_{j \in S_i(\boldsymbol{G})}, \widehat{\sigma}_i) = 1$. If there exists such an index $i \in [\ell]$, then F_I^{asig} outputs $(\mathsf{pk}_{\gamma(j)}, \mathsf{m}_j)_{j \in S_i(\boldsymbol{G})}$ and $\widehat{\sigma}_i$. Otherwise, it aborts.

We show that the output of F_I^{asig} is a valid forgery. If A outputs a valid forgery $((\boldsymbol{G}, ((\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell)), (\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u))$, then there exist the above indexes $z \in [\ell]$ and $i \in [u]$ such that $\mathsf{m}_z \notin \mathcal{Q}$ holds, and AVrfy accepts $(((\mathsf{pk}_{\gamma(j)}, \mathsf{m}_j))_{j \in S_i(\boldsymbol{G})}, \widehat{\sigma}_i)$ due to the winning condition of A (i.e., $\mathsf{m}_z \notin \mathcal{Q}$ and $(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \notin J$). Thus, the output of F_I^{asig} is a valid forgery in the EUF-CMA security game of ASIG, and we obtain $\mathsf{Adv}^{\text{euf-cma}}_{\text{D-ASig},\mathsf{A}}(\lambda) \leq \mathsf{Adv}^{\text{euf-cma}}_{\mathsf{ASIG},\mathsf{F}_I^{asig}}(\lambda)$. □

Theorem 2. *For* identifiability*, the resulting D-ASIG scheme* $\mathsf{D\text{-}ASig}_1$ *satisfies the following:*

– *If* $\boldsymbol{G}$ *is a d-disjunct matrix, and an aggregate signature scheme* ASIG *meets correctness, then* $\mathsf{D\text{-}ASig}_1$ *satisfies* identifiability-completeness.
– *If an aggregate signature scheme* ASIG *meets* EUF-CMA *security, then* $\mathsf{D\text{-}ASig}_1$ *satisfies* identifiability-weak-soundness.

Proof. We prove that $\mathsf{D\text{-}ASig}_1$ satisfies identifiability-completeness. For the output $(\boldsymbol{G}, (\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell, \sigma_\ell))$ of A, let

$$(\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u) \leftarrow \mathsf{DAgg}(\boldsymbol{G}, ((\mathsf{pk}_{\gamma(1)}, \sigma_1), \ldots, (\mathsf{pk}_{\gamma(1)}, \sigma_1)))$$

be a tuple of aggregate signatures. For any valid pair $(\mathsf{pk}_{\gamma(v)}, \mathsf{m}_v)$ such that $\mathsf{Vrfy}^{asig}(\mathsf{pk}_{\gamma(v)}, \mathsf{m}_v, \sigma_v) = 1$ $(v \in [\ell])$, there exists a valid pair $(((\mathsf{pk}_{\gamma(j)}, \mathsf{m}_j))_{j \in S_i(\boldsymbol{G})}, \widehat{\sigma}_i)$ such that $v \in S_i(\boldsymbol{G})$ and $\mathsf{Vrfy}^{asig}(((\mathsf{pk}_{\gamma(j)}, \mathsf{m}_j))_{j \in S_i(\boldsymbol{G})}, \widehat{\sigma}_i) = 1$, due to the d-disjunct property of $\boldsymbol{G}$ and the correctness of ASIG. Thus J does not include valid pairs $(\mathsf{pk}_{\gamma(v)}, \mathsf{m}_v)$ with overwhelming probability, and we have $\mathsf{Adv}^{\text{complete}}_{\mathsf{D\text{-}ASig}_1,\mathsf{A}}(\lambda) \leq \mathsf{negl}(\lambda)$.

By Proposition 2 and Theorem 1, $\mathsf{D\text{-}ASig}_1$ satisfies identifiability-soundness, and we obtain $\mathsf{Adv}^{\text{w-sound}}_{\mathsf{D\text{-}ASig}_1,\mathsf{A}}(\lambda) \leq \mathsf{Adv}^{\text{euf-cma}}_{\mathsf{ASIG},\mathsf{F}_I^{asig}}(\lambda)$.

From the discussion above, D-ASig satisfies both identifiability-completeness and identifiability-weak-soundness, and the proof is completed. □

Construction II. Construction I does not meet identifiability-soundness in general. Although Construction I is useful to detect substituted messages, some applications may need identifiability-soundness. Thus, we propose a generic construction $\mathsf{D\text{-}ASig}_2$ of D-ASIG schemes with identifiability-soundness property. The idea of this construction is as follows: We apply a SNARK system to DAgg algorithm. Namely, we consider a relation

$$R^{snk} = \{(((\mathsf{pk}_i, \mathsf{m}_i)_{i\in[\ell]}, \widehat{\sigma}^{asig}), (\sigma_i)_{i\in[\ell]}) \mid \widehat{\sigma}^{asig} = \mathsf{Agg}^{asig}((\mathsf{pk}_i, \sigma_i)_{i\in[\ell]}) \wedge (\forall i \in [\ell], \mathsf{Vrfy}^{asig}(\mathsf{pk}_i, \mathsf{m}_i, \sigma_i) = 1)\},$$

where $(\mathsf{pk}_i, \mathsf{m}_i)_{i\in[\ell]}, \widehat{\sigma}^{asig})$ is a statement and $(\sigma_i)_{i\in[\ell]}$ is a witness. Then, given multiple triplets $(\mathsf{pk}_k, \mathsf{m}_k, \sigma_k)_{k\in S_i(\boldsymbol{G})}$ for the i-th aggregation, DAgg generates a proof for a statement $((\mathsf{pk}_k, \mathsf{m}_k)_{k\in S_i(\boldsymbol{G})}, \widehat{\sigma}_i^{asig})$ by letting $(\sigma_i)_{i\in[\ell]}$ be a witness, where $\widehat{\sigma}_i^{asig}$ is an aggregate signature of an underlying aggregate signature scheme. If there are at least a triplets $(\mathsf{pk}_i, \mathsf{m}_i, \sigma_i)$ such that $\mathsf{Vrfy}(\mathsf{pk}_i, \mathsf{m}_i, \sigma_i) = 0$, then the verification of the SNARK system rejects, due to the knowledge-soundness property.

Furthermore, we can apply a concrete SNARK system under a non-standard assumption [13, 8, 2, 18, 14, 19, 32, 7, 27], since it is proven in [17] that for any SNARG/SNARK system for NP languages, there is no proof of soundness via any black-box reduction from a falsifiable assumption such as a one-way function, a trapdoor permutation, RSA, CDH, DDH, and LWE assumptions.

This generic construction uses the following primitives:

- An aggregate signature scheme $\mathsf{ASIG} = (\mathsf{KGen}^{asig}, \mathsf{Sign}^{asig}, \mathsf{Vrfy}^{asig}, \mathsf{Agg}^{asig}, \mathsf{AVrfy}^{asig})$.
- A publicly verifiable SNARK system $\Pi_{\mathsf{SNARK}} = (\mathsf{Gen}, \mathsf{P}, \mathsf{V})$.

$\mathsf{D\text{-}ASig}_2 = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy}, \mathsf{DAgg}, \mathsf{DVrfy})$ is constructed as follows: For a security parameter λ, generate $(\mathsf{crs}, \mathsf{vrs}) \leftarrow \mathsf{Gen}(1^\lambda)$, and let $(\mathsf{crs}, \mathsf{vrs})$ be a public parameter of $\mathsf{D\text{-}ASig}_2$.

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$: Output $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}^{asig}(1^\lambda)$.
- $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{m})$: Output $\sigma \leftarrow \mathsf{Sign}^{asig}(\mathsf{sk}^{asig}, \mathsf{m})$.
- $1/0 \leftarrow \mathsf{Vrfy}(\mathsf{pk}, \mathsf{m}, \sigma)$: Output $1/0 \leftarrow \mathsf{Vrfy}^{asig}(\mathsf{pk}^{asig}, \mathsf{m}, \sigma)$.
- $(\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u) \leftarrow \mathsf{DAgg}(\boldsymbol{G}, ((\mathsf{pk}_1, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell, \sigma_\ell)))$:
  1. For each $i \in [u]$, do the following:
     (a) $\widehat{\sigma}_i^{asig} \leftarrow \mathsf{Agg}^{asig}((\mathsf{pk}_k, \mathsf{m}_k, \sigma_k)_{k\in S_i(\boldsymbol{G})})$.
     (b) $\widehat{\pi}_i \leftarrow \mathsf{P}(\mathsf{crs}, ((\mathsf{pk}_k, \mathsf{m}_k)_{k\in S_i(\boldsymbol{G})}, \widehat{\sigma}_i^{asig}), (\sigma_k)_{k\in S_i(\boldsymbol{G})})$. Notice that $\widehat{\pi}_i$ is a proof for a witness $(\sigma_k)_{k\in S_i(\boldsymbol{G})}$ such that
     $$\widehat{\sigma}_i^{asig} = \mathsf{Agg}^{asig}((\mathsf{pk}_k, \mathsf{m}_k, \sigma_k)_{k\in S_i(\boldsymbol{G})}) \wedge (\forall k \in S_i(\boldsymbol{G}), \mathsf{Vrfy}^{asig}(\mathsf{pk}_k, \mathsf{m}_k, \sigma_k) = 1).$$
     (c) Let $\widehat{\sigma}_i := (\widehat{\sigma}_i^{asig}, \widehat{\pi}_i)$.
  2. Output $(\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u)$.
- $J \leftarrow \mathsf{DVrfy}(\boldsymbol{G}, ((\mathsf{pk}_1, \mathsf{m}_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell)), (\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u))$:
  1. $J \leftarrow \{(\mathsf{pk}_1, \mathsf{m}_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell)\}$.

2. For each $i \in [u]$, set $J \leftarrow J \backslash \{(\mathsf{pk}_j, \mathsf{m}_j)\}_{j \in S_i(\boldsymbol{G})}$ if $\mathsf{AVrfy}^{asig}((\mathsf{pk}_k, \mathsf{m}_k)_{k \in S_i(\boldsymbol{G})}, \widehat{\sigma}_i) = 1$ and $\mathsf{V}(\mathsf{vrs}, (\mathsf{pk}_k, \mathsf{m}_k)_{k \in S_i(\boldsymbol{G})}, \widehat{\pi}_i) = 1$.
3. Output J.

The security of $\mathsf{D\text{-}ASig}_2$ is shown as follows.

Theorem 3. *If an aggregate signature scheme* ASIG *meets* EUF-CMA *security, then the resulting D-ASIG scheme* $\mathsf{D\text{-}ASig}_2$ *satisfies* EUF-CMA *security.*

Proof. By using a PPT adversary A breaking the EUF-CMA security of $\mathsf{D\text{-}ASig}_2$, we construct a PPT algorithm F_{II}^{asig} breaking the EUF-CMA security of ASIG, as follows: In **Setup** phase, it takes as input the public key pk_1 of ASIG. It generates $\mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda)$ and $\mathcal{Q} \leftarrow \emptyset$ and gives pk_1 to A. The SIGN oracle in EUF-CMA security game is simulated as follows:

– SIGN: Given a sign-query $\mathsf{m} \in \mathcal{M}$, issue m to the given signing oracle SIGN^{asig} and get the response σ. Return σ and set $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{\mathsf{m}\}$.

In **Output** phase, A outputs $(\boldsymbol{G}, ((\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell)), (\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u))$. Then, F_{II}^{asig} computes $J \leftarrow \mathsf{DVrfy}(\boldsymbol{G}, ((\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell)), (\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u))$, and finds $z \in [\ell]$ such that $\gamma(z) = 1$ and $\mathsf{m}_z \notin \mathcal{Q}$. If there exists $i \in [u]$ such that

– $z \in S_i(\boldsymbol{G})$,
– $\mathsf{AVrfy}^{asig}((\mathsf{pk}_{\gamma(k)}, \mathsf{m}_k)_{k \in S_i(\boldsymbol{G})}, \widehat{\sigma}_i^{asig*}) = 1$, and
– $\mathsf{V}(\mathsf{vrs}, ((\mathsf{pk}_{\gamma(k)}, \mathsf{m}_k)_{k \in S_i(\boldsymbol{G})}, \widehat{\sigma}_i^{asig*}), \widehat{\pi}_i) = 1$,

then it outputs the pair $((\mathsf{pk}_{\gamma(k)}, \mathsf{m}_k)_{k \in S_i(\boldsymbol{G})}, \widehat{\sigma}_i^{asig*})$ and halts.

We show that the output of F_{II}^{asig} is a valid forgery in the EUF-CMA security game of ASIG. The output of A fulfills the condition of forgeries of $\mathsf{D\text{-}ASig}_2$: $(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \notin J$ and $\mathsf{m}_z \notin \mathcal{Q}$. Then, due to $(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \notin J$, there exists $i \in [u]$ such that AVrfy^{asig} algorithm accepts $((\mathsf{pk}_{\gamma(j)}, \mathsf{m}_j)_{j \in S_i(\boldsymbol{G})}, \widehat{\sigma}_i^{asig})$. Besides, $\mathsf{m}_z \notin \mathcal{Q}$ holds. Therefore, the output of F_{II}^{asig} is a valid forgery of ASIG, and we have $\mathsf{Adv}^{\text{euf-cma}}_{\mathsf{D\text{-}ASig}_2, \mathsf{A}}(\lambda) \leq \mathsf{Adv}^{\text{euf-cma}}_{\mathsf{ASIG}, \mathsf{F}_{II}^{asig}}(\lambda)$. □

Theorem 4. *For identifiability, the resulting D-ASIG scheme* $\mathsf{D\text{-}ASig}_2$ *meets the following:*

– *If* $\boldsymbol{G} \in \{0,1\}^{u \times \ell}$ *is a d-disjunct matrix, and an aggregate signature scheme* ASIG *meets correctness, then* $\mathsf{D\text{-}ASig}_2$ *satisfies* identifiability-completeness.
– *If* $\boldsymbol{G} \in \{0,1\}^{u \times \ell}$ *is a d-disjunct matrix, and a SNARK system* Π_{SNARK} *meets knowledge-soundness, then* $\mathsf{D\text{-}ASig}_2$ *satisfies* identifiability-soundness.

Proof. Let A be a PPT adversary against $\mathsf{D\text{-}ASig}_2$. We prove that $\mathsf{D\text{-}ASig}_2$ satisfies identifiability-completeness. Let $(\boldsymbol{G}, (\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell, \sigma_\ell))$ be the output of a PPT adversary A. For any pair $(\mathsf{pk}_{\gamma(v)}, \mathsf{m}_v, \sigma_v)$ $(v \in [\ell])$ such that $\mathsf{Vrfy}^{asig}(\mathsf{pk}_{\gamma(v)}^{asig}, \mathsf{m}_v, \sigma_v) = 1$, there exists $i \in [u]$ such that $v \in S_i(\boldsymbol{G})$, $\mathsf{AVrfy}^{asig}((\mathsf{pk}_{\gamma(v)}, \mathsf{m}_v)_{v \in S_i(\boldsymbol{G})}, \widehat{\sigma}_i^{asig}) = 1$, and $\mathsf{V}(\mathsf{vrs}, ((\mathsf{pk}_{\gamma(k)}, \mathsf{m}_k)_{k \in S_i(\boldsymbol{G})}, \widehat{\sigma}_i), \widehat{\pi}_i) =$

1. Then, due to the correctness of ASIG and the completeness of Π_{SNARG}, such a pair $(\mathsf{pk}_{\gamma(v)}, \mathsf{m}_v)$ is not in J. Thus, we have $\mathsf{Adv}^{\mathrm{complete}}_{\text{D-ASig}_2,\mathsf{A}}(\lambda) \leq \mathsf{negl}(\lambda)$.

Next, we prove that D-ASig$_2$ satisfies identifiability-soundness. We construct a PPT algorithm F^{sd} breaking the knowledge-soundness of Π_{SNARK} as follows: In Setup phase, it takes as input the CRS crs and the verification key vrs of Π_{SNARK}. It generates $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KGen}(1^\lambda)$, sets $\mathcal{Q} \leftarrow \emptyset$, and gives pk_1 to A. Since F^{sd} has a secret key sk_1, it can simulate SIGN oracle in the straightforward way.

When A outputs $(\boldsymbol{G}, (\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell, \sigma_\ell))$, F^{sd} computes

$$(\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u) \leftarrow \mathsf{DAgg}(\boldsymbol{G}, ((\mathsf{pk}_{\gamma(1)}, \sigma_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \sigma_\ell))), \text{ and}$$
$$J \leftarrow \mathsf{DVrfy}(\boldsymbol{G}, ((\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell)), (\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u)).$$

Then, it finds $z \in [\ell]$ such that $\gamma(z) = 1$ and $\mathsf{Vrfy}(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z, \sigma_z) = 0$. If there exits $i \in [u]$ such that $z \in S_i(\boldsymbol{G})$, $\mathsf{AVrfy}^{asig}((\mathsf{pk}_{\gamma(k)}, \mathsf{m}_k)_{k \in S_i(\boldsymbol{G})}, \widehat{\sigma}_i^{asig}) = 1$, and $\mathsf{V}(\mathsf{vrs}, ((\mathsf{pk}_{\gamma(k)}, \mathsf{m}_k)_{k \in S_i(\boldsymbol{G})}, \widehat{\sigma}_i^{asig}), \widehat{\pi}_i) = 1$, then F^{sd} outputs the triplets $(x, \widehat{\pi}_i; w)$, where $x = ((\mathsf{pk}_{\gamma(k)}, \mathsf{m}_k)_{k \in S_i(\boldsymbol{G})}, \widehat{\sigma}_i^{asig})$ and $w = (\sigma_k)_{k \in S_i(\boldsymbol{G})}$.

F^{sd} simulates the environment of A completely. We show that F^{sd} breaks the knowledge-soundness of Π_{SNARK}. The output of A fulfills $(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \in D \backslash J$ for $z \in [\ell]$ such that $\gamma(z) = 1$. Then, due to $(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \notin J$, there exists $i \in [u]$ such that $z \in S_i(\boldsymbol{G})$, $\mathsf{AVrfy}^{asig}((\mathsf{pk}_{\gamma(k)}, \mathsf{m}_k)_{k \in S_i(\boldsymbol{G})}, \widehat{\sigma}_i^{asig}) = 1$, and $\mathsf{V}(\mathsf{vrs}, ((\mathsf{pk}_{\gamma(k)}, \mathsf{m}_k)_{k \in S_i(\boldsymbol{G})}, \widehat{\sigma}_i^{asig}), \widehat{\pi}_i) = 1$. In addition, the pair of the statement $x = ((\mathsf{pk}_{\gamma(k)}, \mathsf{m}_k)_{k \in S_i(\boldsymbol{G})}, \widehat{\sigma}_i^{asig*})$ and the witness $w = (\sigma_k)_{k \in S_i(\boldsymbol{G})}$ does not meet the relation

$$R^{snk} = \{(x, w) \mid \widehat{\sigma}_i^{asig} = \mathsf{Agg}^{asig}((\mathsf{pk}_{\gamma(k)}, \sigma_k)_{k \in S_i(\boldsymbol{G})}) \wedge (\forall k \in S_i(\boldsymbol{G}), \mathsf{Vrfy}^{asig}(\mathsf{pk}_{\gamma(k)}, \mathsf{m}_k, \sigma_k) = 1)\}$$

since $\mathsf{Vrfy}^{asig}(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z, \sigma_z) = 0$ holds. Hence, F^{sd} breaks the knowledge-soundness of Π_{SNARK}, and we have $\mathsf{Adv}^{\mathrm{sound}}_{\text{D-ASig}_2,\mathsf{A}}(\lambda) \leq \mathsf{Adv}^{\text{k-sound}}_{\Pi_{\mathsf{SNARK}},\mathsf{F}^{sd}}(\lambda)$.

From the above discussion, the proof is completed. □

4.3 Instantiations Based on Pairing

In this section, we give concrete constructions of D-ASIG schemes by applying concrete primitives to generic constructions presented in Section 4.2. For all generic constructions of D-ASIG schemes, we apply the pairing-based construction of aggregate signature schemes proposed in [4].

Instantiation of Construction I. We describe a concrete construction of D-ASig$_1$ by using an aggregate signature scheme of [4] and any non-adaptive group testing protocol with d-disjunct matrices.

An instantiation D-ASig$_{p1}$ = (KGen, Sign, Vrfy, DAgg, DVrfy) of D-ASig$_1$ is described as follows: For a security parameter λ, set the following public parameters

- G_1 and G_2 are multiplicative cyclic groups of prime order p.
- g_1 and g_2 are generators of G_1 and G_2, respectively.
- ψ is a computable isomorphism from G_2 to G_1.
- The bilinear map $e : G_1 \times G_2 \to G_T$ with a target group G_T.
- A message space is $\{0,1\}^*$.
- $H : \{0,1\}^* \to G_1$ is a random oracle.

The algorithms $(\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy}, \mathsf{DAgg}, \mathsf{DVrfy})$ are as follows:

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$: Choose $x \xleftarrow{\$} \mathbb{Z}_p$ and compute $v \leftarrow g_2^x \in G_2$. Output $\mathsf{pk} = v$ and $\mathsf{sk} = x$.
- $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{m})$: Output $\sigma \leftarrow h^x \in G_1$, where $h \leftarrow H(\mathsf{m}) \in G_1$.
- $1/0 \leftarrow \mathsf{Vrfy}(\mathsf{pk}, \mathsf{m}, \sigma)$: Output 1 if $e(\sigma, g_2) = e(H(\mathsf{m}), v)$ holds, where $\mathsf{pk} = v$. Output 0 otherwise.
- $(\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u) \leftarrow \mathsf{DAgg}(\boldsymbol{G}, (\mathsf{pk}_1, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell, \sigma_\ell))$: By using $\boldsymbol{G} \in \{0,1\}^{u \times \ell}$, generate a tuple of signatures as follows:
  1. For each $i \in [u]$, $\widehat{\sigma}_i \leftarrow \prod_{k \in S_i(\boldsymbol{G})} \sigma_k$.
  2. Output $(\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u)$.
- $J \leftarrow \mathsf{DVrfy}(\boldsymbol{G}, (\mathsf{pk}_1, \ldots, \mathsf{pk}_\ell), (\mathsf{m}_1, \ldots, \mathsf{m}_\ell), (\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u))$: Generate a set J of invalid message and signature pairs as follows:
  1. $J \leftarrow \{(\mathsf{pk}_1, \mathsf{m}_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell)\}$.
  2. For each $i \in [u]$, set $J \leftarrow J \backslash \{(\mathsf{pk}_k, \mathsf{m}_k)\}_{k \in S_i(\boldsymbol{G})}$ if $e(\widehat{\sigma}_i, g_2) = \prod_{k \in S_i(\boldsymbol{G})} e(H(\mathsf{m}_k), v_k)$, where $v_k = \mathsf{pk}_k$ for $k \in S_i(\boldsymbol{G})$.
  3. Output J.

By combining Proposition 1, Theorems 1 and 2, we obtain the following result about security of the above scheme.

Corollary 1. *If (G_1, G_2) is a bilinear group pair for co-Diffie-Hellman, then the resulting scheme* $\mathsf{D\text{-}ASig}_{p1}$ *satisfies* EUF-CMA *security.*

In addition, $\mathsf{D\text{-}ASig}_{p1}$ *satisfies the following identifiability:*

- *If $\boldsymbol{G}$ is a d-disjunct matrix, then* $\mathsf{D\text{-}ASig}_{p1}$ *satisfies* identifiability-completeness.
- *If (G_1, G_2) is a bilinear group pair for co-Diffie-Hellman,* $\mathsf{D\text{-}ASig}_{p1}$ *satisfies* identifiability-weak-soundness.

However, we note that $\mathsf{D\text{-}ASig}_{p1}$ does not meet identifiability-soundness, which can be seen as follows. An adversary has $((\mathsf{pk}_1, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell, \sigma_\ell))$ such that

$$\sigma_1 = H(\mathsf{m}_1)^{x_1}, \sigma_2 = H(\mathsf{m}_2)^{x_2}, \sigma_3 = H(\mathsf{m}_3)^{x_3}, \ldots, \sigma_\ell = H(\mathsf{m}_\ell)^{x_\ell},$$
$$(\text{where } \mathsf{sk}_1 = x_1, \mathsf{sk}_2 = x_2, \ldots, \mathsf{sk}_\ell = x_\ell),$$

by accessing SIGN oracle and generating key-pairs by itself. Then, it outputs $((\mathsf{pk}_1, \mathsf{m}_1, \sigma_1'), (\mathsf{pk}_2, \mathsf{m}_2, \sigma_2'), (\mathsf{pk}_3, \mathsf{m}_3, \sigma_3), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell, \sigma_\ell))$ such that

$$\sigma_1' = H(\mathsf{m}_1)^{x_1}\gamma, \sigma_2' = H(\mathsf{m}_2)^{x_2}\gamma^{-1}, \sigma_3 = H(\mathsf{m}_3)^{x_3}, \ldots, \sigma_\ell = H(\mathsf{m}_\ell)^{x_\ell},$$

where $\gamma \neq 1$, Vrfy of $\mathsf{ASIG}_{\mathsf{BGLS}}$ accepts this output, since

$$e(\sigma_1'\sigma_2' \prod_{i=3}^{\ell} \sigma_i, g_2) = e(\prod_{i=1}^{\ell} \sigma_i, g_2)$$

holds. Thus, although the triplets $(\mathsf{pk}_1, \mathsf{m}_1, \sigma_1'), (\mathsf{pk}_2, \mathsf{m}_2, \sigma_2')$ are invalid, $\widehat{\sigma}' := \sigma_1'\sigma_2' \prod_{i=3}^{\ell} \sigma_i$ is valid aggregate signature on $((\mathsf{pk}_1, \mathsf{m}_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell))$. Hence, the output satisfies the winning condition in identifiability-soundness game.

Instantiation of Construction II. We describe a concrete construction of $\mathsf{D\text{-}ASig}_2$ by applying an aggregate signature scheme [4] and a SNARK system [18].

An instantiation $\mathsf{D\text{-}ASig}_{p2} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy}, \mathsf{DAgg}, \mathsf{DVrfy})$ of $\mathsf{D\text{-}ASig}_2$ is described as follows: For a security parameter λ, set the following public parameters

- G_1 and G_2 are multiplicative cyclic groups of prime order p.
- g_1 and g_2 are generators of G_1 and G_2, respectively.
- ψ is a computable isomorphism from G_2 to G_1.
- The bilinear map $e : G_1 \times G_2 \to G_T$ with a target group G_T.
- A message space is $\{0,1\}^*$.
- $H : \{0,1\}^* \to G_1$ is a random oracle.
- The relation generator $\mathcal{R}$ of the underlying SNARK system generates the relation R^{snk} described in Construction II, with the form

$$R^{snk} = (p, G_1, G_2, G_T, e, g_1, g_2, l, \{u_i^{snk}(X), v_i^{snk}(X), w_i^{snk}(X)\}_{i\in\{0,1,\ldots,m\}}, t^{snk}(X)).$$

In addition, the CRS crs of the SNARK system is generated as follows: For the above parameters g_1, g_2, and $e : G_1 \times G_2 \to G_T$, we define $[a]_1$, $[b]_2$, and $[c]_T$ as g_1^a, g_2^b, and $e'(g_1, g_2)^c$, respectively, where $e' : G_1 \times G_2 \to G_T$ is a bilinear map with a target group G_T.

1. $\alpha, \beta, \gamma, \delta, x \xleftarrow{\$} \mathbb{Z}_q^*$.
2. $\mathsf{crs} = ([\mathsf{crs}_1]_1, [\mathsf{crs}_2]_2)$ is computed as follows:

$$\mathsf{crs}_1 = \left(\begin{array}{c} \alpha, \beta, \delta, \{x\}_{i\in\{0,1,\ldots,n-1\}}, \{\frac{\beta u_i^{snk}(x) + \alpha v_i^{snk}(x) + w_i^{snk}(x)}{\gamma}\}_{i\in\{0,1,\ldots,l\}}, \\ \{\frac{\beta u_i^{snk}(x) + \alpha v_i^{snk}(x) + w_i^{snk}(x)}{\delta}\}_{i\in\{l+1,\ldots,m\}}, \{\frac{x^i t^{snk}(x)}{\delta}\}_{i\in\{0,1,\ldots,n-2\}} \end{array} \right), \text{ and}$$

$$\mathsf{crs}_2 = \left(\beta, \gamma, \delta, \{x^i\}_{i\in\{0,1,\ldots,n-1\}}\right).$$

Notice that the verification key of the scheme [18] corresponds to the above crs since it is publicly verifiable.

The algorithms $(\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy}, \mathsf{DAgg}, \mathsf{DVrfy})$ are given as follows:

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$: Choose $x \xleftarrow{\$} \mathbb{Z}_p$ and compute $v \leftarrow g_2^x \in G_2$. Output $\mathsf{pk} = v$ and $\mathsf{sk} = x$.
- $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{m})$: Output $\sigma \leftarrow h^x \in G_1$, where $h \leftarrow H(\mathsf{m}) \in G_1$.

- $1/0 \leftarrow \mathsf{Vrfy}(\mathsf{pk}, \mathsf{m}, \sigma)$: Output 1 if $e(\sigma, g_2) = e(H(\mathsf{m}), v)$ holds, where $\mathsf{pk} = v$. Output 0 otherwise.
- $(\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u) \leftarrow \mathsf{DAgg}(\boldsymbol{G}, (\mathsf{pk}_1, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell, \sigma_\ell))$: By using $\boldsymbol{G} \in \{0,1\}^{u \times \ell}$, generate a tuple of signatures as follows:
  1. For each $i \in [u]$, do the following:
     - $\widehat{\sigma}_i^{asig} \leftarrow \prod_{k \in S_i(\boldsymbol{G})} \sigma_k$.
     - Generate $\widehat{\pi}_i$ as follows: Let $a_0^{(i)} = 1$, and generate the form $(a_1^{(i)}, \ldots, a_m^{(i)}) \in \mathbb{Z}_p^m$ of the statement $((\mathsf{pk}_k), \mathsf{m}_k)_{k \in S_i(\boldsymbol{G})}, \widehat{\sigma}_i^{asig})$ and the witness $(\sigma)_{i \in [\ell]}$. Choose $r_i, s_i \xleftarrow{\$} \mathbb{Z}_p$ and compute $\widehat{\pi}_i = ([A_i]_1, [C_i]_1, [B_i]_2)$, where

$$A_i = \alpha + \sum_{j=0}^{m} a_j^{(i)} u_j^{snk}(x) + r_i \delta, \qquad B_i = \beta + \sum_{j=0}^{m} a_j^{(i)} v_j^{snk}(x) + s_i \delta,$$

$$C_i = \frac{\sum_{j=l+1}^{m} a_j^{(i)} (\beta u_j^{snk}(x) + \alpha v_j^{snk}(x) + w_j^{snk}(x)) + h_i(x) t^{snk}(x)}{\delta} + A_i s_i + B_i r_i - r_i s_i \delta.$$

     - Let $\widehat{\sigma}_i = (\widehat{\sigma}_i^{asig}, \widehat{\sigma}_i)$
  2. Output $(\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u)$.
- $J \leftarrow \mathsf{DVrfy}(\boldsymbol{G}, (\mathsf{pk}_1, \ldots, \mathsf{pk}_\ell), (\mathsf{m}_1, \ldots, \mathsf{m}_\ell), (\widehat{\sigma}_1, \ldots, \widehat{\sigma}_u))$: Generate a set J of invalid message and signature pairs as follows:
  1. $J \leftarrow \{(\mathsf{pk}_1, \mathsf{m}_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell)\}$.
  2. For each $i \in [u]$, set $J \leftarrow J \backslash \{(\mathsf{pk}_k, \mathsf{m}_k)\}_{k \in S_i(\boldsymbol{G})}$ if the following holds for $\widehat{\sigma}_i = (\widehat{\sigma}_i^{asig}, \widehat{\pi} = ([A_i]_1, [C_i]_1, [B_i]_2))$:

$$e(\widehat{\sigma}_i^{asig}, g_2) = \prod_{k \in S_i(\boldsymbol{G})} e(H(\mathsf{m}_k), v_k), \text{ where } v_k = \mathsf{pk}_k \text{ for } k \in S_i(\boldsymbol{G}), \text{ and}$$

$$[A_i]_1 \cdot [B_i]_2 = [\alpha]_1 \cdot [\beta]_2 + \sum_{j=0}^{l} a_j^{(i)} \left[\frac{\beta u_j^{snk}(x) + \alpha v_j^{snk}(x) + w_j^{snk}(x)}{\gamma} \right]_1 \cdot [\gamma]_2 + [C_i]_1 \cdot [\delta]_2,$$

     where let $a_0^{(i)} = 1$, and let $(a_1^{(i)}, \ldots, a_l^{(i)}) \in \mathbb{Z}_p^l$ be the statement-form generated from $((\mathsf{pk}_k, \mathsf{m}_k)_{k \in S_i(\boldsymbol{G})}, \widehat{\sigma}_i^{asig})$.
  3. Output J.

By combining Propositions 1 and 4, and Theorems 3 and 4, we have the following result.

Corollary 2. *If (G_1, G_2) is a bilinear group pair for co-Diffie-Hellman, then the resulting scheme* $\mathsf{D\text{-}ASig}_{p2}$ *satisfies* EUF-CMA *security.*

In addition, $\mathsf{D\text{-}ASig}_{p2}$ *satisfies the following identifiability:*

- *If $\boldsymbol{G}$ is a d-disjunct matrix, then* $\mathsf{D\text{-}ASig}_{p2}$ *satisfies* identifiability-completeness.
- *If (G_1, G_2) is a bilinear group pair for co-Diffie-Hellman, then* $\mathsf{D\text{-}ASig}_{p2}$ *satisfies* identifiability-weak-soundness *under the generic bilinear group model.*

5 Interactive Aggregate Signature with Detecting Functionality

5.1 Model and Security Definition

In this section, we introduce a formal model and security definition of interactive aggregate signature scheme with detecting functionality (D-IASIG for short). In D-IASIG, each signer with a public key pk_i generates a signature σ_i on his/her local message m_i by using Sign, and sends $(\mathsf{pk}_i, \mathsf{m}_i, \sigma_i)$ to Aggregator DAgg. In order to specify all the invalid messages, Aggregator DAgg and Verifier DVrfy repeat the following process: DAgg selects a subset of pairs of public keys and messages, and generates a tuple of aggregate signatures from the multiple signatures of the selected pairs of public keys and messages; On receiving a set of pairs of public keys and messages and the tuple of aggregate signatures, DVrfy checks whether there is an invalid pair of a public key and a message by using aggregate signatures; Through a *feedback channel*[3] which is an authenticated channel with low bandwidth, DVrfy transmits the verification result to DAgg. Finally, DVrfy outputs a list that specifies invalid pairs of public keys and messages.

Formally, an interactive aggregate signature scheme with detecting functionality (D-IASIG for short) consists of an interactive protocol $\langle \mathsf{DAgg}, \mathsf{DVrfy} \rangle$ between two polynomial-time algorithms DAgg and DVrfy with six polynomial-time algorithm (KGen, Sign, Vrfy, Agg, AVrfy, GSel): For a security parameter λ, and let $\mathcal{M} = \mathcal{M}(\lambda)$ be a message space.

Key Generation $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$: A randomized algorithm KGen takes as input a security parameter 1^λ, and it outputs a public key pk and a secret key sk.

Signing $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{m})$: A randomized or deterministic algorithm Sign takes as input a secret key sk and a message $\mathsf{m} \in \mathcal{M}$, and it outputs a signature σ.

Verification $1/0 \leftarrow \mathsf{Vrfy}(\mathsf{pk}, \mathsf{m}, \sigma)$: A deterministic algorithm Vrfy takes as input a public key, a message $\mathsf{m} \in \mathcal{M}$, and a signature σ, and it outputs 1 or 0.

Aggregation $\widehat{\sigma} \leftarrow \mathsf{Agg}((\mathsf{pk}_1, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell, \sigma_\ell))$: A randomized or deterministic algorithm Agg takes as input a tuple $((\mathsf{pk}_1, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell, \sigma_\ell))$ of triplets of a public key, a message, and a signature, and it outputs an aggregate signature $\widehat{\sigma}$.

Aggregate Verification $1/0 \leftarrow \mathsf{AVrfy}(((\mathsf{pk}_1, \mathsf{m}_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell)), \widehat{\sigma})$: A deterministic algorithm AVrfy takes as input a tuple $((\mathsf{pk}_1, \mathsf{m}_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell))$ of pairs of a public key and a message, and an aggregate signature σ, and it outputs 1 or 0.

Group Selection $(\{\mathcal{G}_1^{(s)}, \ldots, \mathcal{G}_{u^{(s)}}^{(s)}\}, \mathsf{st}^{(s)}) \leftarrow \mathsf{GSel}(J^{(s-1)}, \{\mathcal{G}_1^{(s-1)}, \ldots, \mathcal{G}_{u^{(s-1)}}^{(s-1)}\}, \mathsf{st}^{(s-1)})$: Let $\mathcal{PMS} = \{(\mathsf{pk}_1, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell, \sigma_\ell)\}$. A deterministic algorithm

[3] For simplicity, we assume such a feedback channel physically, but we may use an independent digital signature scheme while maintaining keyless aggregation if a feedback channel is not authenticated.

GSel takes as input a subset $J^{(s-1)} \subseteq \mathcal{PMS}$, subsets $\mathcal{G}_1^{(s-1)}, \ldots, \mathcal{G}_{u^{(s-1)}}^{(s-1)}$ of $J^{(s-1)}$, and the current internal state $\mathsf{st}^{(s-1)}$. It outputs new subsets $\{\mathcal{G}_1^{(s)}, \ldots, \mathcal{G}_{u^{(s)}}^{(s)}\}$ and the next internal state $\mathsf{st}^{(s)}$.

Interactive-Detection $J \leftarrow \langle \mathsf{DAgg}(\mathcal{PMS}), \mathsf{DVrfy}(\mathcal{PM}) \rangle$:
Let $\mathcal{PMS} = \{(\mathsf{pk}_i, \mathsf{m}_i, \sigma_i)\}_{i \in [\ell]}$, and $\mathcal{PM} = \{(\mathsf{pk}_i, \mathsf{m}_i)\}_{i \in [\ell]}$. The N-stage interactive protocol $\langle \mathsf{DAgg}, \mathsf{DVrfy} \rangle$ between two polynomial-time DAgg and DVrfy takes as input $(\mathcal{PMS}, \mathcal{PM})$, and it outputs a set J of invalid pairs of a public key and a message.

Furthermore, we describe the protocol-framework of $\langle \mathsf{DAgg}(\mathcal{PMS}), \mathsf{DVrfy}(\mathcal{PM}) \rangle$.

For $\mathcal{PMS}$ and a subset $\mathcal{G} \subseteq \mathcal{PM}$, let $\mathcal{S}(\mathcal{G}) = \{(\mathsf{pk}, \mathsf{m}, \sigma) \in \mathcal{PMS} \mid (\mathsf{pk}, \mathsf{m}) \in \mathcal{G}\}$. Let $s = 1$, $u^{(0)} = 1$, $\mathcal{G}_1^{(0)} = \mathcal{PM}$, $J^{(0)} = \mathcal{PM}$, and $\mathsf{st}^{(0)} = \emptyset$ at the setup phase. The s-th stage process of DAgg and DVrfy is as follows ($s \in [N]$):

$\mathsf{DAgg}(\mathcal{PMS})$: If $s = 1$, then start from Step 2:

Step 1: Receive $(b_1^{u^{(s-1)}}, \ldots, b_{u^{(s-1)}}^{(s-1)}) \in \{0,1\}^{u^{(s-1)}}$ from DVrfy, and set $J^{(s-1)} \leftarrow J^{(s-2)} \backslash \{(\mathsf{pk}, \mathsf{m}) \mid i \in [u^{(s-1)}] \wedge b_i^{(s-1)} = 1 \wedge (\mathsf{id}, \mathsf{m}) \in \mathcal{G}_i^{(s-1)}\}$.
- If $s > N$ or $J^{(s-1)} = \emptyset$, then halt.
- Otherwise, move to Step 2.

Step 2: $(\{\mathcal{G}_1^{(s)}, \ldots, \mathcal{G}_{u^{(s)}}^{(s)}\}, \mathsf{st}^{(s)}) \leftarrow \mathsf{GSel}(J^{(s-1)}, \{\mathcal{G}_1^{(s-1)}, \ldots, \mathcal{G}_{u^{(s-1)}}^{(s-1)}\}, \mathsf{st}^{(s-1)})$.

Step 3: For each $i \in [u^{(s)}]$, $\widehat{\sigma}_i^{(s)} \leftarrow \mathsf{Agg}(\mathcal{S}(\mathcal{G}_i^{(s)}))$.

Step 4: Send $(\widehat{\sigma}_1^{(s)}, \ldots, \widehat{\sigma}_{u^{(s)}}^{(s)})$ to DVrfy.

$\mathsf{DVrfy}(\mathcal{PM})$:

Step 1: Receive $(\widehat{\sigma}_1^{(s)}, \ldots, \widehat{\sigma}_{u^{(s)}}^{(s)})$ from DAgg.

Step 2: $(\{\mathcal{G}_1^{(s)}, \ldots, \mathcal{G}_{u^{(s)}}^{(s)}\}, \mathsf{st}^{(s)}) \leftarrow \mathsf{GSel}(J^{(s-1)}, \{\mathcal{G}_1^{(s-1)}, \ldots, \mathcal{G}_{u^{(s-1)}}^{(s-1)}\}, \mathsf{st}^{(s-1)})$.

Step 3: Set $J^{(s)} \leftarrow J^{(s-1)}$ and for each $i \in [u^{(s)}]$, do the following:
- $b_i^{(s)} \leftarrow \mathsf{AVrfy}(\mathcal{G}_i^{(s)}, \widehat{\sigma}_i^{(s)})$.
- $J \leftarrow J \backslash \{(\mathsf{pk}, \mathsf{m}) \mid b_i^{(s)} = 1 \wedge (\mathsf{id}, \mathsf{m}) \in \mathcal{G}_i^{(s)}\}$.

Step 4: Send $(b_1^{u^{(s)}}, \ldots, b_{u^{(s)}}^{(s)}) \in \{0,1\}^{u^{(s)}}$ to DAgg. If $s = N$ or $J^{(s)} = \emptyset$, then output $J \leftarrow J^{(s)}$ and halt.

When both algorithms DAgg and DVrfy halt, $\langle \mathsf{DAgg}(\mathcal{PMS}), \mathsf{DVrfy}(\mathcal{PM}) \rangle$ outputs J.

The correctness of D-IASIG is defined as follows:

Definition 17 (Correctness). *A D-IASIG scheme* $\langle \mathsf{DAgg}, \mathsf{DVrfy} \rangle$ *with* (KGen, Sign, Vrfy, Agg, AVrfy, GSel) *satisfies correctness if the following conditions hold:*

- *For every* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$ *and every* $\mathsf{m} \in \mathcal{M}$, *it holds that* $\mathsf{Vrfy}(\mathsf{pk}, \mathsf{m}, \sigma) = 1$ *with overwhelming probability, where* $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{m})$.
- *For every* $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KGen}(1^\lambda)$ *and every* $\mathsf{m}_i \in \mathcal{M}$ *for all* $i \in [\ell]$, *it holds that* $\mathsf{AVrfy}((\mathsf{pk}_1, \mathsf{m}_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell), \widehat{\sigma}) = 1$ *with overwhelming probability, where* $\widehat{\sigma} \leftarrow \mathsf{Agg}((\mathsf{pk}_1, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell, \sigma_\ell))$, *and for all* $i \in [\ell]$, $\sigma_i \leftarrow \mathsf{Sign}(\mathsf{sk}_i, \mathsf{m}_i)$.

– *For every* $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KGen}(1^\lambda)$ *and every* $\mathsf{m}_i \in \mathcal{M}$ *for all* $i \in [\ell]$*, it holds that* $\langle \mathsf{DAgg}(\mathcal{PMS}), \mathsf{DVrfy}(\mathcal{PM}) \rangle = \emptyset$ *with overwhelming probability, where* $\sigma_i \leftarrow \mathsf{Sign}(\mathsf{sk}_i, \mathsf{m}_i)$ *for all* $i \in [\ell]$*.*

We define the security notions of D-IASIG: EUF-CMA security and identifiability.

Definition 18 (EUF-CMA security). *A D-IASIG scheme* $\mathsf{D\text{-}IASig} = \langle \mathsf{DAgg}, \mathsf{DVrfy} \rangle$ *with* (KGen, Sign, Vrfy, Agg, AVrfy, GSel) *satisfies* EUF-CMA *security if for any PPT adversary* A *against* D-IASig*, the advantage* $\mathsf{Adv}^{\text{euf-cma}}_{\mathsf{D\text{-}IASig},\mathsf{A}}(\lambda) := \Pr[\mathsf{A}\ \textit{wins}]$ *is negligible in* λ*.* [A *wins*] *is the event* A *wins in the following game:*

Setup. *A challenger generates* $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KGen}(1^\lambda)$*. It sets* $\mathcal{Q} \leftarrow \emptyset$ *and gives* pk_1 *to* A*.*

Queries. A *is allowed to access the following oracle:*

– SIGN*: Given a sign-query* $\mathsf{m} \in \mathcal{M}$*, signing oracle* SIGN *returns* $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}_1, \mathsf{m})$ *and sets* $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{\mathsf{m}\}$*.*

Output. A *outputs a forgery* $(\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1, \sigma_1)$*, …,* $(\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell, \sigma_\ell)$*. The challenger computes* $J \leftarrow \langle \mathsf{DAgg}(\mathcal{PMS}), \mathsf{DVrfy}(\mathcal{PM}) \rangle$*, where* $\mathcal{PMS} = ((\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell, \sigma_\ell))$ *and* $\mathcal{PM} = ((\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell))$*.*
A *wins if* $(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \notin J$ *and* $\mathsf{m}_z \notin \mathcal{Q}$*, where* $z \in [\ell]$ *is a message-index such that* $\gamma(z) = 1$*.*

Definition 19 (Identifiability). *Regarding the* identifiability *of D-IASIG* $\mathsf{D\text{-}IASig} = \langle \mathsf{DAgg}, \mathsf{DVrfy} \rangle$ *with* (KGen, Sign, Vrfy, Agg, AVrfy, GSel)*, we define* identifiability-completeness *and* identifiability-soundness*. We consider the following security game: Let* A *be a PPT adversary against* D-IASig*.*

Setup. *The challenger generates a key-pair* $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KGen}(1^\lambda)$ *and sets* $\mathcal{Q} \leftarrow \emptyset$*. It gives* pk_1 *to* A*.*

Queries. A *is allowed to access the following oracle:*

– SIGN*: Given a sign query* $\mathsf{m} \in \mathcal{M}$*, signing oracle* SIGN *returns* $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}_1, \mathsf{m})$ *and sets* $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{\mathsf{m}\}$*.*

Output. A *outputs* $((\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell, \sigma_\ell))$*. The challenger computes* $J \leftarrow \langle \mathsf{DAgg}(\mathcal{PMS}), \mathsf{DVrfy}(\mathcal{PM}) \rangle$*, where* $\mathcal{PMS} = ((\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell, \sigma_\ell))$ *and* $\mathcal{PM} = ((\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell))$*.*

identifiability-completeness *and* identifiability-soundness *are defined as follows: For* $\mathcal{PMS}$*, let* $D := \{(\mathsf{pk}_{\gamma(i)}, \mathsf{m}_i) \mid i \in [\ell] \wedge \mathsf{Vrfy}(\mathsf{pk}_{\gamma(i)}, \mathsf{m}_i, \sigma_i) = 0\}$*, and* $\bar{D} := \{(\mathsf{pk}_{\gamma(i)}, \mathsf{m}_i) \mid i \in [\ell] \wedge \mathsf{Vrfy}(\mathsf{pk}_{\gamma(i)}, \mathsf{m}_i, \sigma_i) = 1\}$*. Let* $z \in [\ell]$ *be a message-index such that* $\gamma(z) = 1$*.*

– ***Completeness****:* D-IASig *satisfies* identifiability-completeness *if for any PPT adversary* A*, the following advantage is negligible in* λ*:*

$$\mathsf{Adv}^{\text{complete}}_{\mathsf{D\text{-}IASig},\mathsf{A}}(\lambda) := \Pr\left[(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \in \bar{D} \cap J\right].$$

– ***Soundness****:* D-IASig *satisfies* identifiability-soundness *if for any PPT adversary* A*, the following advantage is negligible in* λ*:*

$$\mathsf{Adv}^{\text{sound}}_{\mathsf{D\text{-}IASig},\mathsf{A}}(\lambda) := \Pr\left[(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \in D \backslash J\right].$$

In addition, identifiability-weak-soundness *is defined in the same way as the definition of* identifiability-soundness *except that the advantage of a PPT adversary* A *is defined as*

$$\mathsf{Adv}^{\text{w-sound}}_{\text{D-IASig,A}}(\lambda) := \Pr\left[(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \in D' \backslash J\right],$$

where $D' = \{(\mathsf{pk}_{\gamma(i)}, \mathsf{m}_i) \mid i \in [\ell] \wedge \mathsf{Vrfy}(\mathsf{pk}_{\gamma(i)}, \mathsf{m}_i, \sigma_i) = 0 \wedge \mathsf{m}_i \notin \mathcal{Q}\}$.

The following proposition shows the relation between EUF-CMA security and identifiability-weak-soundness.

Proposition 3. *If a D-IASIG scheme* D-IASig *meets* EUF-CMA *security, it also satisfies* identifiability-weak-soundness.

Proof. By using a PPT adversary A breaking identifiability-weak-soundness, we construct F^{euf} breaking EUF-CMA security, as follows: It takes as input a public key pk_1 of D-IASig and gives pk_1 to A. By using the given signing oracle SIGN^{euf} in the EUF-CMA security game, it simulates the SIGN oracle in the identifiability-weak-soundness game, in the straightforward way. When A outputs $((\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell, \sigma_\ell))$, then F^{euf} also outputs $((\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell, \sigma_\ell))$.

If the output of A fulfills $(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \in \{(\mathsf{pk}_{\gamma(i)}, \mathsf{m}_i) \mid i \in [\ell] \wedge \mathsf{Vrfy}(\mathsf{pk}_{\gamma(i)}, \mathsf{m}_i, \sigma_i) = 0 \wedge \mathsf{m}_i \notin \mathcal{Q}\} \backslash J$ for $z \in [\ell]$ such that $\gamma(z) = 1$, then the output of F^{euf} is a valid forgery of D-IASig since $(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \notin J$ and $\mathsf{m}_z \notin \mathcal{Q}$ hold. Thus, the output of F^{euf} is a valid forgery, and we obtain $\mathsf{Adv}^{\text{w-sound}}_{\text{D-IASig,A}}(\lambda) \leq \mathsf{Adv}^{\text{euf-cma}}_{\text{D-IASig,F}^{euf}}(\lambda)$. □

5.2 Generic Construction from Adaptive Group-Testing

Construction I. We propose a generic construction of D-IASig starting from any aggregate signature scheme ASIG and any adaptive group-testing protocol AGT. The idea of this construction lies in specifying invalid pairs of a public key and a message by executing an adaptive group-testing protocol under the following assumption:

- we regard $\mathcal{PMS}$ of D-IASig as a set of all tested items in AGT, where for all $i \in [\ell]$, let $\mathsf{id}_i = (\mathsf{pk}_i, \mathsf{m}_i)$ and $\mathsf{x}_i = \sigma_i$,
- we regard $\mathcal{PM}$ of D-IASig as a set of all verification-items in AGT, where for all $i \in [\ell]$, let $\mathsf{id}_i = (\mathsf{pk}_i, \mathsf{m}_i)$ and $\mathsf{v}_i = (\mathsf{pk}_i, \mathsf{m}_i)$, and
- a relation of AGT is defined as $R = \{((\mathsf{pk}_i, \mathsf{m}_i, \sigma_i), (\mathsf{pk}_i, \mathsf{m}_i)) \mid i \in [\ell] \wedge \mathsf{Vrfy}(\mathsf{pk}_i, \mathsf{m}_i, \sigma_i) = 0\} \subseteq \mathcal{PMS} \times \mathcal{PM}$,

Let $\mathsf{ASIG} = (\mathsf{KGen}^{asig}, \mathsf{Sign}^{asig}, \mathsf{Vrfy}^{asig}, \mathsf{Agg}^{asig}, \mathsf{AVrfy}^{asig})$ be an aggregate signature scheme, and let $\mathsf{AGT} = \langle \mathsf{GS}^{agt}, \mathsf{IS}^{agt} \rangle$ be an adaptive group-testing protocol with $(\mathsf{GSel}^{agt}, \mathsf{Coms}^{agt}, \mathsf{Test}^{agt})$.

Our construction $\text{D-IASig}_1 = \langle \mathsf{DAgg}, \mathsf{DVrfy} \rangle$ with (KGen, Sign, Vrfy, Agg, AVrfy, GSel) is as follows:

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$: Output $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}^{asig}(1^\lambda)$.

- $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{m})$: Output $\sigma \leftarrow \mathsf{Sign}^{asig}(\mathsf{sk}, \mathsf{m})$.
- $1/0 \leftarrow \mathsf{Vrfy}(\mathsf{pk}, \mathsf{m}, \sigma)$: Output $1/0 \leftarrow \mathsf{Vrfy}^{asig}(\mathsf{pk}, \mathsf{m}, \sigma)$.
- $\widehat{\sigma} \leftarrow \mathsf{Agg}((\mathsf{pk}_1, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell, \sigma_\ell))$:
  Output $\widehat{\sigma} \leftarrow \mathsf{Agg}^{asig}((\mathsf{pk}_1, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell, \sigma_\ell))$.
- $1/0 \leftarrow \mathsf{AVrfy}((\mathsf{pk}_1, \mathsf{m}_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell), \widehat{\sigma})$:
  Output $1/0 \leftarrow \mathsf{AVrfy}^{asig}((\mathsf{pk}_1, \mathsf{m}_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell), \widehat{\sigma})$.
- $(\{\mathcal{G}_1^{(s)}, \ldots, \mathcal{G}_{u^{(s)}}^{(s)}\}, \mathsf{st}^{(s)}) \leftarrow \mathsf{GSel}(J^{(s-1)}, \{\mathcal{G}_1^{(s-1)}, \ldots, \mathcal{G}_{u^{(s-1)}}^{(s-1)}\}, \mathsf{st}^{(s-1)})$:
  Output $(\{\mathcal{G}_1^{(s)}, \ldots, \mathcal{G}_{u^{(s)}}^{(s)}\}, \mathsf{st}^{(s)}) \leftarrow \mathsf{GSel}^{agt}(J^{(s-1)}, \{\mathcal{G}_1^{(s-1)}, \ldots, \mathcal{G}_{u^{(s-1)}}^{(s-1)}\}, \mathsf{st}^{(s-1)})$.

Notice that the interactive protocol $\langle \mathsf{DAgg}(\mathcal{PMS}), \mathsf{DVrfy}(\mathcal{PM}) \rangle$ can be regarded as $\langle \mathsf{GS}^{agt}, \mathsf{IS}^{agt} \rangle$ which is replaced Coms^{agt} and Test^{agt} by Agg and AVrfy, respectively. We describe the interactive protocol $\mathsf{D\text{-}IASig}_1 = \langle \mathsf{DAgg}(\mathcal{PMS}), \mathsf{DVrfy}(\mathcal{PM}) \rangle$, as follows:

1. Let $u^{(0)} = 1$, $\mathcal{G}_1^{(0)} = \mathcal{PM}$, $J^{(0)} = \mathcal{PM}$, and $\mathsf{st}^{(0)} = \emptyset$.
2. DAgg and DVrfy are executed by following the protocol-framework $\langle \mathsf{DAgg}, \mathsf{DVrfy} \rangle$ in Section 5.1, with the above GSel.
3. Output the result obtained by running the above protocol (i.e., a set J of invalid pairs of a public key and a message).

Theorems 5 and 6 below show the security of D-IASig.

Theorem 5. *If an aggregate signature scheme* ASIG *meets* EUF-CMA *security, and an adaptive group-testing protocol* AGT *meets* GT-soundness*, then the resulting D-IASIG scheme* $\mathsf{D\text{-}IASig}_1$ *satisfies* EUF-CMA *security.*

Proof. Let A be a PPT adversary against $\mathsf{D\text{-}IASig}_1$. First, we consider an adversary which breaks the EUF-CMA security of $\mathsf{D\text{-}IASig}_1$ without making any forgeries of ASIG. If this adversary outputs $((\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell, \sigma_\ell))$ such that $(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \notin J$ and $\mathsf{m}_z \notin \mathcal{Q}$ ($z \in [\ell]$ and $\gamma(z) = 1$), then there exists $(s, i) \in [N] \times [u^{(s)}]$ such that $\mathsf{AVrfy}^{asig}(\mathcal{G}_i^{(s)}, \widehat{\sigma}_i^{(s)}) = 1$. Since the adversary does not generate any forgeries of ASIG, there exists such a pair (s, i) if $D \not\subseteq J$ holds. However, this does not occur with overwhelming probability, due to the GT-soundness of AGT. Thus, there does not exist such an adversary with at least probability $1 - \mathsf{negl}(\lambda)$.

Next, by using an adversary A making forgeries of ASIG, we construct a PPT algorithm F_I^{asig} breaking the EUF-CMA of ASIG, as follows: It takes as input a public key pk_1 of ASIG and gives pk_1 to A. By using the given signing oracle SIGN^{asig}, it can simulate SIGN oracles. When A outputs $(\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell, \sigma_\ell)$, F_I^{asig} runs $J \leftarrow \langle \mathsf{DAgg}(\mathcal{PMS}), \mathsf{DVrfy}(\mathcal{PM}) \rangle$. If there exists $z \in [\ell]$ such that $\gamma(z) = 1$, $(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \notin J$, and $\mathsf{m}_z \notin \mathcal{Q}$, then it checks whether there exists a pair $(s, i) \in [N] \times [u^{(s)}]$ such that $(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \in \mathcal{G}_i^{(s)}$ and $\mathsf{AVrfy}(\mathcal{G}_i^{(s)}, \widehat{\sigma}_i^{(s)}) = 1$ hold, where $\widehat{\sigma}_i^{(s)} \leftarrow \mathsf{Agg}(\mathcal{S}(\mathcal{G}_i^{(s)}))$. If there exists such a pair, it outputs $(\mathcal{G}_i^{(s)}, \widehat{\sigma}_i^{(s)})$.

We show that the output of F_I^{asig} is a valid forgery of ASIG. If the output of A fulfills $(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \notin J$ and $\mathsf{m}_z \notin \mathcal{Q}$ for $z \in [\ell]$ such that $\gamma(z) = 1$,

then there exists (s,i) such that $(\mathsf{pk}_{\gamma(z)},\mathsf{m}_z) \in \mathcal{G}_i^{(s)}$ and $\mathsf{AVrfy}\left(\mathcal{G}_i^{(s)},\widehat{\sigma}_i^{(s)}\right) = 1$, due to $(\mathsf{pk}_{\gamma(z)},\mathsf{m}_z) \notin J$. Hence, the output of F_I^{asig} satisfies the winning condition in the EUF-CMA security game of ASIG, and we have $\mathsf{Adv}^{\text{euf-cma}}_{\text{D-IASig}_1,\mathsf{A}}(\lambda) \leq \mathsf{Adv}^{\text{euf-cma}}_{\mathsf{ASIG},\mathsf{F}_I^{asig}}(\lambda) + \mathsf{negl}(\lambda)$. □

Theorem 6. *For* identifiability*, the resulting D-IASIG scheme* D-IASig_1 *meets the following:*

- *If an adaptive group-testing protocol* AGT *meets* GT-completeness*, then* D-IASig_1 *satisfies* identifiability-completeness.
- *If an aggregate signature scheme* ASIG *meets* EUF-CMA *security, and an adaptive group-testing protocol* AGT *meets* GT-soundness*, then* D-IASig_1 *satisfies* identifiability-weak-soundness.

Proof. Let A be a PPT adversary against D-IASig_1. We prove that D-IASig_1 satisfies identifiability-completeness. Let $((\mathsf{pk}_{\gamma(1)},\mathsf{m}_1,\sigma_1),\ldots,(\mathsf{pk}_{\gamma(\ell)},\mathsf{m}_\ell,\sigma_\ell))$ be the output of A. For any valid pair $(\mathsf{pk}_{\gamma(v)},\mathsf{m}_v)$ such that $\mathsf{Vrfy}^{asig}(\mathsf{pk}_{\gamma(v)},\mathsf{m}_v,\sigma_v) = 1$ $(v \in [\ell])$, there exists a valid pair $(\mathcal{G}_i^{(s)},\widehat{\sigma}_i^{(s)})$ $((s,i) \in [N] \times [u^{(s)}])$ such that $(\mathsf{pk}_{\gamma(v)},\mathsf{m}_v) \in \mathcal{G}_i^{(s)}$ and $\mathsf{Vrfy}^{asig}(\mathcal{G}_i^{(s)},\widehat{\sigma}_i) = 1$, due to the GT-completeness of AGT. If the underlying ASIG meets correctness, then J does not include valid pairs $(\mathsf{pk}_{\gamma(k)},\mathsf{m}_k)$ with overwhelming probability. Hence, we have $\mathsf{Adv}^{\text{complete}}_{\text{D-IASig}_1,\mathsf{A}}(\lambda) \leq \mathsf{negl}(\lambda)$.

In addition, we obtain $\mathsf{Adv}^{\text{w-sound}}_{\text{D-IASig}_1,\mathsf{A}}(\lambda) \leq \mathsf{Adv}^{\text{euf-cma}}_{\text{D-IASig}_1,\mathsf{F}^{asig}}(\lambda) + \mathsf{negl}(\lambda)$ by combining Proposition 3 and Theorem 5.

From the above discussion, D-IASig_1 satisfies both identifiability-completeness and identifiability-weak-soundness. □

Construction II. By using a SNARK system, we construct a D-IASIG scheme satisfying identifiability-soundness. The idea of this scheme is the same as that of a D-ASIG scheme.

Let $\mathsf{ASIG} = (\mathsf{KGen}^{asig},\mathsf{Sign}^{asig},\mathsf{Vrfy}^{asig},\mathsf{Agg}^{asig},\mathsf{AVrfy}^{asig})$ be an aggregate signature scheme, let $\mathsf{AGT} = \langle \mathsf{GS}^{agt},\mathsf{IS}^{agt}\rangle$ be an adaptive group-testing protocol with $(\mathsf{GSel}^{agt},\mathsf{Coms}^{agt},\mathsf{Test}^{agt})$, and let $\Pi_{\mathsf{SNARK}} = (\mathsf{Gen},\mathsf{P},\mathsf{V})$ be a publicly verifiable SNARK system.

Our construction $\text{D-IASig}_2 = \langle \mathsf{DAgg},\mathsf{DVrfy}\rangle$ with (KGen, Sign, Vrfy, Agg, AVrfy, GSel) is as follows: For a security parameter λ, generate $(\mathsf{crs},\mathsf{vrs}) \leftarrow \mathsf{Gen}(1^\lambda)$, and let $(\mathsf{crs},\mathsf{vrs})$ be a public parameter of D-IASig_2.

- $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$: Output $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}^{asig}(1^\lambda)$.
- $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk},\mathsf{m})$: Output $\sigma \leftarrow \mathsf{Sign}^{asig}(\mathsf{sk},\mathsf{m})$.
- $1/0 \leftarrow \mathsf{Vrfy}(\mathsf{pk},\mathsf{m},\sigma)$: Output $1/0 \leftarrow \mathsf{Vrfy}^{asig}(\mathsf{pk},\mathsf{m},\sigma)$.
- $\widehat{\sigma} \leftarrow \mathsf{Agg}((\mathsf{pk}_1,\mathsf{m}_1,\sigma_1),\ldots,(\mathsf{pk}_\ell,\mathsf{m}_\ell,\sigma_\ell))$:
  1. $\widehat{\sigma}^{asig} \leftarrow \mathsf{Agg}^{asig}((\mathsf{pk}_1,\mathsf{m}_1,\sigma_1),\ldots,(\mathsf{pk}_\ell,\mathsf{m}_\ell,\sigma_\ell))$.
  2. $\widehat{\pi} \leftarrow \mathsf{P}(\mathsf{crs},((\mathsf{pk}_1,\mathsf{m}_1),\ldots,(\mathsf{pk}_\ell,\mathsf{m}_\ell),\widehat{\sigma}^{asig}),(\sigma_1,\ldots,\sigma_\ell))$.
  3. Output $\widehat{\sigma} = (\widehat{\sigma}^{asig},\widehat{\pi})$.

– $1/0 \leftarrow \mathsf{AVrfy}((\mathsf{pk}_1, \mathsf{m}_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell), \widehat{\sigma})$:
Output 1 if
  - $\mathsf{AVrfy}^{asig}((\mathsf{pk}_1, \mathsf{m}_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell), \widehat{\sigma}^{asig}) = 1$, and
  - $\mathsf{V}(\mathsf{vrs}, ((\mathsf{pk}_1, \mathsf{m}_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell), \widehat{\sigma}^{asig}), \widehat{\pi}) = 1$

  hold. Output 0 otherwise.
– $(\{\mathcal{G}_1^{(s)}, \ldots, \mathcal{G}_{u^{(s)}}^{(s)}\}, \mathsf{st}^{(s)}) \leftarrow \mathsf{GSel}(J^{(s-1)}, \{\mathcal{G}_1^{(s-1)}, \ldots, \mathcal{G}_{u^{(s-1)}}^{(s-1)}\}, \mathsf{st}^{(s-1)})$:
Output $(\{\mathcal{G}_1^{(s)}, \ldots, \mathcal{G}_{u^{(s)}}^{(s)}\}, \mathsf{st}^{(s)}) \leftarrow \mathsf{GSel}^{agt}(J^{(s-1)}, \{\mathcal{G}_1^{(s-1)}, \ldots, \mathcal{G}_{u^{(s-1)}}^{(s-1)}\}, \mathsf{st}^{(s-1)})$

We describe the interactive protocol $\mathsf{D\text{-}IASig}_2 = \langle \mathsf{DAgg}(\mathcal{PMS}), \mathsf{DVrfy}(\mathcal{PM}) \rangle$, as follows:

1. Let $u^{(0)} = 1$, $\mathcal{G}_1^{(0)} = \mathcal{PM}$, $J^{(0)} = \mathcal{PM}$, and $\mathsf{st}^{(0)} = \emptyset$.
2. DAgg and DVrfy are executed by following the protocol-framework $\langle \mathsf{DAgg}, \mathsf{DVrfy} \rangle$ in Section 5.1, with the above GSel.
3. Output the result obtained by running the above protocol (i.e., a set J of invalid pairs of a public key and a message).

Theorems 7 and 8 show the security of $\mathsf{D\text{-}IASig}_2$.

Theorem 7. *If an aggregate signature scheme* ASIG *meets* EUF-CMA *security, an adaptive group-testing protocol* AGT *meets* GT-soundness*, then the resulting D-IASIG scheme* $\mathsf{D\text{-}IASig}_2$ *satisfies* EUF-CMA *security.*

Proof. Let A be a PPT adversary against $\mathsf{D\text{-}IASig}_2$. First, we consider an adversary which breaks the EUF-CMA security of $\mathsf{D\text{-}IASig}_2$ without making any forgeries of ASIG. In the same way as the proof of Theorem 5, the probability that A outputs such a valid forgery is at most $\mathsf{negl}(\lambda)$ due to the GT-soundness of AGT.

Next, by using an adversary A making forgeries of ASIG, we construct a PPT algorithm F_{II}^{asig} breaking the EUF-CMA security of ASIG, as follows: It takes as input a public key pk_1 of ASIG. It generates $(\mathsf{crs}, \mathsf{vrs}) \leftarrow \mathsf{Gen}(1^\lambda)$ and gives pk_1 to A. By using the given signing oracle SIGN^{asig} and public keys, it can simulate SIGN oracles. When A outputs $(\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell, \sigma_\ell)$, F_2^{asig} runs $J \leftarrow \langle \mathsf{DAgg}(\mathcal{PMS}), \mathsf{DVrfy}(\mathcal{PM}) \rangle$. If there exists $z \in [\ell]$ such that $\gamma(z) = 1$, $(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \notin J$, and $\mathsf{m}_z \notin \mathcal{Q}$, then it checks whether there exists pair $(s, i) \in [N] \times [u^{(s)}]$ such that $(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \in \mathcal{G}_i^{(s)}$, $\mathsf{AVrfy}^{asig}(\mathcal{G}_i^{(s)}, \widehat{\sigma}_i^{asig(s)}) = 1$ and $\mathsf{V}(\mathsf{vrs}, (\mathcal{G}_i^{(s)}, \widehat{\sigma}_i^{(s)}), \widehat{\pi}_i^{(s)}) = 1$ hold, where $\widehat{\sigma}_i^{(s)} = (\widehat{\sigma}_i^{asig(s)}, \widehat{\pi}_i^{(s)}) \leftarrow \mathsf{Agg}(\mathcal{S}(\mathcal{G}_i^{(s)}))$. If there exists such a pair, it outputs $(\mathcal{G}_i^{(s)}, \widehat{\sigma}_i^{(s)})$.

We show that the output of F_{II}^{asig} is a valid forgery of ASIG. If the output of A fulfills $(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \notin J$ and $\mathsf{m}_z \notin \mathcal{Q}$ for $z \in [\ell]$ such that $\gamma(z) = 1$, then there exists (s, i) such that $(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \in \mathcal{G}_i^{(s)}$ and $\mathsf{AVrfy}^{asig}\left(\mathcal{G}_i^{(s)}, \widehat{\sigma}_i^{(s)}\right) = 1$, due to $(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \notin J$. Hence, the output of F_2^{asig} is a valid forgery in the EUF-CMA security game of ASIG, and we have $\mathsf{Adv}^{\text{euf-cma}}_{\mathsf{D\text{-}IASig}_2, \mathsf{A}}(\lambda) \leq \mathsf{Adv}^{\text{euf-cma}}_{\mathsf{ASIG}, \mathsf{F}_{II}^{asig}}(\lambda) + \mathsf{negl}(\lambda)$. □

Theorem 8. *For* identifiability*, the resulting D-IASIG scheme* $\mathsf{D\text{-}IASig}_2$ *meets the following:*

- *If an adaptive group-testing protocol* AGT *meets* GT-completeness*, then* $\mathsf{D\text{-}IASig}_2$ *satisfies* identifiability-completeness.
- *If a SNARK system* Π_{SNARK} *meets knowledge-soundness, and an adaptive group-testing protocol* AGT *meets* GT-soundness*, then* D-IASig *satisfies* identifiability-soundness.

Proof. It is shown that $\mathsf{D\text{-}IASig}_2$ satisfies identifiability-completeness, in the same way as the proof of Theorem 6.

We prove that $\mathsf{D\text{-}IASig}_2$ satisfies identifiability-soundness. We construct a PPT algorithm F_2^{sd} breaking the knowledge-soundness of Π_{SNARK} as follows: It takes as input a CRS crs and the verification key vrs of Π_{SNARK}, and generates $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KGen}^{asig}(1^\lambda)$. It sets $\mathcal{Q} \leftarrow \emptyset$ and gives pk_1 to A. By using sk_1, F^{sd} simulates SIGN oracle.

When A outputs $(\boldsymbol{G}, (\mathsf{pk}_{\gamma(1)}, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_{\gamma(\ell)}, \mathsf{m}_\ell, \sigma_\ell))$, F^{sd} computes $J \leftarrow \langle \mathsf{DAgg}(\mathcal{PMS}), \mathsf{DVrfy}(\mathcal{PM}) \rangle$. For $z \in [\ell]$ such that $\gamma(z) = 1$, it finds $(s, i) \in [N] \times [u^{(s)}]$ such that $(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z, \sigma_z) \in \mathcal{G}_i^{(s)}$, $\mathsf{Vrfy}^{asig}(\mathcal{G}_i^{(s)}, \widehat{\sigma}_i^{asig(s)}) = 1$, and $\mathsf{V}(\mathsf{vrs}, (\mathcal{G}_i^{(s)}, \widehat{\sigma}_i^{asig(s)}), \widehat{\pi}_i^{(s)}) = 1$. Then, it outputs $(x, \widehat{\pi}_i^{(s)}; w)$, where $x = (\mathcal{G}_i^{(s)}, \widehat{\sigma}_i^{asig(s)})$ and $w = \mathcal{S}(\mathcal{G}_i^{(s)})$.

We show that F^{sd} breaks the knowledge-soundness of Π_{SNARK}. The output of A fulfills $(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \in D \backslash J$ for $z \in [\ell]$ such that $\gamma(z) = 1$. Then, due to $(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z) \notin J$, there exists $(s, i) \in [N] \times [u^{(s)}]$ such that $z \in \mathcal{G}_i^{(s)}$, $\mathsf{AVrfy}^{asig}(\mathcal{G}_i^{(s)}, \widehat{\sigma}_i^{asig(s)}) = 1$, and $\mathsf{V}(\mathsf{vrs}, (\mathcal{G}_i^{(s)}, \widehat{\sigma}_i^{asig(s)}), \widehat{\pi}_i^{(s)}) = 1$. In addition, the pair of the statement $x = (\mathcal{G}_i^{(s)}, \widehat{\sigma}_i^{asig(s)})$ and the witness $w = \mathcal{S}(\mathcal{G}_i^{(s)})$ does not meet the relation

$$R^{snk} = \{(x, w) \mid \widehat{\sigma}_i^{asig(s)} = \mathsf{Agg}^{asig}(\mathcal{S}(\mathcal{G}_i^{(s)})) \wedge (\forall (\mathsf{pk}_{\gamma(k)}, \mathsf{m}_k) \in \mathcal{G}_i^{(s)}, \mathsf{Vrfy}^{asig}(\mathsf{pk}_{\gamma(k)}, \mathsf{m}_k, \sigma_k) = 1)\}$$

since $\mathsf{Vrfy}^{asig}(\mathsf{pk}_{\gamma(z)}, \mathsf{m}_z, \sigma_z) = 0$ holds. Hence, F^{sd} breaks the knowledge-soundness of Π_{SNARK}, and we have $\mathsf{Adv}^{\mathrm{sound}}_{\mathsf{D\text{-}IASig}_2, \mathsf{A}}(\lambda) \leq \mathsf{Adv}^{\text{k-sound}}_{\Pi_{\mathsf{SNARK}}, \mathsf{F}^{sd}}(\lambda)$.

From the above discussion, the proof is completed. □

5.3 Instantiations Based on Pairing

In this section, we give concrete constructions of D-IASIG schemes by applying concrete primitives to generic constructions presented in Section 5.2. For all generic constructions of D-IASIG schemes, we apply the pairing-based construction of aggregate signature schemes proposed in [4].

Instantiation of Construction I. We describe a concrete $\mathsf{D\text{-}IASig}_1$ construction based on pairing, by applying an aggregate signature scheme of [4] and the binary-search as an adaptive group-testing protocol.

The construction $\mathsf{D\text{-}IASig}_{p1} = \langle \mathsf{DAgg}, \mathsf{DVrfy} \rangle$ with (KGen, Sign, Vrfy, AVrfy, GSel) is as follows: The public parameters of $\mathsf{D\text{-}IASig}_{p1}$ are the same as those of $\mathsf{D\text{-}ASig}_{p1}$ in Section 4.3. Notice that the binary-search adaptive group testing protocol can be expressed by GSel algorithm because the Coms and Test algorithms depend on applications. The algorithms (KGen, Sign, Vrfy, AVrfy, GSel) are given as follows:

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$: Choose $x \xleftarrow{\$} \mathbb{Z}_p$ and compute $v \leftarrow g_2^x \in G_2$. Output $\mathsf{pk} = v$ and $\mathsf{sk} = x$.
- $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{m})$: Output $\sigma \leftarrow h^x \in G_1$, where $h \leftarrow H(\mathsf{m}) \in G_1$.
- $1/0 \leftarrow \mathsf{Vrfy}(\mathsf{pk}, \mathsf{m}, \sigma)$: Output 1 if $e(\sigma, g_2) = e(H(\mathsf{m}), v)$ holds, where $\mathsf{pk} = v$. Output 0 otherwise.
- $\widehat{\sigma} \leftarrow \mathsf{Agg}((\mathsf{pk}_1, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell, \sigma_\ell))$: Output $\widehat{\sigma} \leftarrow \prod_{i=1}^{\ell} \sigma_i \in G_1$.
- $1/0 \leftarrow \mathsf{AVrfy}((\mathsf{pk}_1, \mathsf{m}_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell), \widehat{\sigma})$: Output 1 if $e(\widehat{\sigma}, g_2) = \prod_{i=1}^{\ell} e(H(\mathsf{m}_i), v_i)$ holds, where $\mathsf{pk}_i = v_i$ for all $i \in [\ell]$. Output 0 otherwise.
- $(\{\mathcal{G}_1^{(s)}, \ldots, \mathcal{G}_{u^{(s)}}^{(s)}\}, \mathsf{st}^{(s)}) \leftarrow \mathsf{GSel}(J^{(s-1)}, \{\mathcal{G}_1^{(s-1)}, \ldots, \mathcal{G}_{u^{(s-1)}}^{(s-1)}\}, \mathsf{st}^{(s-1)})$:
  1. Let $i \leftarrow 1$ and $\mathsf{st}^{(s)} \leftarrow \emptyset$.
  2. For each $j \in [u^{s-1}]$ and $\mathcal{G}_j^{(s-1)} = \{(\mathsf{pk}_{j,1}\mathsf{m}_{j,1}), \ldots, (\mathsf{pk}_{j,k}, \mathsf{m}_{j,k})\}$, if $\mathcal{G}_j^{(s-1)} \subseteq J^{(s-1)}$ holds, then let

$$\begin{aligned} \mathcal{G}_i^{(s)} &\leftarrow \{(\mathsf{pk}_{j,1}\mathsf{m}_{j,1}), \ldots, (\mathsf{pk}_{j,k/2}, \mathsf{m}_{j,k/2})\}, \\ \mathcal{G}_{i+1}^{(s)} &\leftarrow \{(\mathsf{pk}_{j,k/2+1}\mathsf{m}_{j,k/2+1}), \ldots, (\mathsf{pk}_{j,k}, \mathsf{m}_{j,k})\}, \text{ and} \\ i &\leftarrow i + 2. \end{aligned}$$

  3. Output $(\{\mathcal{G}_1^{(s)}, \ldots, \mathcal{G}_{u^{(s)}}^{(s)}\}, \mathsf{st}^{(s)})$

The interactive protocol $\langle \mathsf{DAgg}(\mathcal{PMS}), \mathsf{DVrfy}(\mathcal{PM}) \rangle$ is executed by following the protocol-framework in Section 5.1, with the above GSel.

By combining Proposition 1, Theorems 5 and 6, we can see that, if (G_1, G_2) is a bilinear pair for co-Diffie-Hellman, the resulting D-IASIG scheme $\mathsf{D\text{-}IASig}_{p1}$ satisfies $\mathsf{EUF\text{-}CMA}$ security; and that the D-IASIG scheme $\mathsf{D\text{-}IASig}_{p1}$ satisfies $\mathsf{identifiability\text{-}completeness}$ and $\mathsf{identifiability\text{-}weak\text{-}soundness}$.

Instantiation of Construction II. We describe a concrete pairing-based construction of $\mathsf{D\text{-}IASig}_2$, by applying the aggregate signature scheme of [4], the binary-search as an adaptive group-testing protocol, and the SNARK system of [18].

The construction $\mathsf{D\text{-}IASig}_{p2} = \langle \mathsf{DAgg}, \mathsf{DVrfy} \rangle$ with (KGen, Sign, Vrfy, AVrfy, GSel) is as follows: The public parameters of $\mathsf{D\text{-}IASig}_{p2}$ are the same as those of $\mathsf{D\text{-}ASig}_{p2}$ in Section 4.3. Notice that the binary-search adaptive group testing protocol can be expressed by GSel algorithm because the Coms and Test algorithms depend on applications. The algorithms (KGen, Sign, Vrfy, AVrfy, GSel) are given as follows:

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$: Choose $x \xleftarrow{\$} \mathbb{Z}_p$ and compute $v \leftarrow g_2^x \in G_2$. Output $\mathsf{pk} = v$ and $\mathsf{sk} = x$.
- $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{m})$: Output $\sigma \leftarrow h^x \in G_1$, where $h \leftarrow H(\mathsf{m}) \in G_1$.
- $1/0 \leftarrow \mathsf{Vrfy}(\mathsf{pk}, \mathsf{m}, \sigma)$: Output 1 if $e(\sigma, g_2) = e(H(\mathsf{m}), v)$ holds, where $\mathsf{pk} = v$. Output 0 otherwise.
- $\widehat{\sigma} \leftarrow \mathsf{Agg}((\mathsf{pk}_1, \mathsf{m}_1, \sigma_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell, \sigma_\ell))$:
  1. Compute $\widehat{\sigma}^{asig} \leftarrow \prod_{i=1}^{\ell} \sigma_i \in G_1$.
  2. Generate $\widehat{\pi} = ([A]_1, [C]_1, [B]_2)$ as follows: Let $a_0 = 1$, and generate the form $(a_1, \ldots, a_m) \in \mathbb{Z}_p^m$ of the statement $((\mathsf{pk}_k), \mathsf{m}_k)_{k \in S_i(\boldsymbol{G})}, \widehat{\sigma}_i^{asig})$ and the witness $(\sigma)_{i \in [\ell]}$. Choose $r, s \xleftarrow{\$} \mathbb{Z}_p$ and compute
$$A = \alpha + \sum_{i=0}^{m} a_i u_i^{snk}(x) + r\delta, \qquad B = \beta + \sum_{i=0}^{m} a_i v_i^{snk}(x) + s\delta,$$
$$C = \frac{\sum_{i=l+1}^{m} a_i(\beta u_i^{snk}(x) + \alpha v_i^{snk}(x) + w_i^{snk}(x)) + h(x)t^{snk}(x)}{\delta} + As + Br - rs\delta.$$
  3. Output $\widehat{\sigma} = (\widehat{\sigma}^{asig}, \widehat{\pi})$.
- $1/0 \leftarrow \mathsf{AVrfy}((\mathsf{pk}_1, \mathsf{m}_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell), \widehat{\sigma})$: Output 1 if for $\widehat{\sigma} = (\widehat{\sigma}^{asig}, \widehat{\pi} = ([A]_1, [C]_1, [B]_2))$ the following conditions holds:
$$e(\widehat{\sigma}^{asig}, g_2) = \prod_{i=1}^{\ell} e(H(\mathsf{m}_i), v_i), \text{ and}$$
$$[A]_1 \cdot [B]_2 = [\alpha]_1 \cdot [\beta]_2 + \sum_{i=0}^{l} a_i \left[\frac{\beta u_i^{snk}(x) + \alpha v_i^{snk}(x) + w_i^{snk}(x))}{\gamma}\right]_1 \cdot [\gamma]_2 + [C]_1 \cdot [\delta]_2,$$
  where let $a_0 = 1$, and let $(a_1, \ldots, a_l) \in \mathbb{Z}_p^l$ be the statement-form generated from $(((\mathsf{pk}_1, \mathsf{m}_1), \ldots, (\mathsf{pk}_\ell, \mathsf{m}_\ell)), \widehat{\sigma}^{asig})$. Output 0 otherwise.
- $(\{\mathcal{G}_1^{(s)}, \ldots, \mathcal{G}_{u^{(s)}}^{(s)}\}, \mathsf{st}^{(s)}) \leftarrow \mathsf{GSel}(J^{(s-1)}, \{\mathcal{G}_1^{(s-1)}, \ldots, \mathcal{G}_{u^{(s-1)}}^{(s-1)}\}, \mathsf{st}^{(s-1)})$:
  1. Let $i \leftarrow 1$ and $\mathsf{st}^{(s)} \leftarrow \emptyset$.
  2. For each $j \in [u^{s-1}]$ and $\mathcal{G}_j^{(s-1)} = \{(\mathsf{pk}_{j,1}\mathsf{m}_{j,1}), \ldots, (\mathsf{pk}_{j,k}, \mathsf{m}_{j,k})\}$, if $\mathcal{G}_j^{(s-1)} \subseteq J^{(s-1)}$ holds, then let
$$\mathcal{G}_i^{(s)} \leftarrow \{(\mathsf{pk}_{j,1}\mathsf{m}_{j,1}), \ldots, (\mathsf{pk}_{j,k/2}, \mathsf{m}_{j,k/2})\},$$
$$\mathcal{G}_{i+1}^{(s)} \leftarrow \{(\mathsf{pk}_{j,k/2+1}\mathsf{m}_{j,k/2+1}), \ldots, (\mathsf{pk}_{j,k}, \mathsf{m}_{j,k})\}, \text{ and}$$
$$i \leftarrow i + 2.$$
  3. Output $(\{\mathcal{G}_1^{(s)}, \ldots, \mathcal{G}_{u^{(s)}}^{(s)}\}, \mathsf{st}^{(s)})$.

The interactive protocol $\langle \mathsf{DAgg}(\mathcal{PMS}), \mathsf{DVrfy}(\mathcal{PM}) \rangle$ is executed by following the framework-protocol in Section 5.1.

By Proposition 4 and Theorems 7 and 8, we can see that, if (G_1, G_2) is a bilinear pair for co-Diffie-Hellman, the resulting D-IASIG scheme $\mathsf{D\text{-}IASig}_{p2}$ satisfies EUF-CMA security; and that the D-IASIG scheme $\mathsf{D\text{-}IASig}_{p2}$ satisfies identifiability-completeness and identifiability-soundness in the generic bilinear group model.

6 Conclusion

In this paper, we comprehensively studied aggregate signatures with detecting functionality, that had functionality of both keyless aggregation of multiple signatures and identifying an invalid message from the aggregate signature, in order to reduce a total amount of signature-size for lots of messages. Specifically, we formalized the model and security notions for both non-interactive and interactive protocols of aggregate signatures with detecting functionality, and we provided construction methodology for both protocols from group-testing protocols in a generic and comprehensive way. As instantiations, pairing-based constructions were provided.

As explained in [21], aggregate signatures have interesting applications including sensor networks, secure logging, and authenticating software. We would like to expect that aggregate signatures with detecting functionality would be useful primitives for such applications in the era of IoT.

Acknowledgements. This paper is based on results obtained from a project, JPNP16007, commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

References

1. Ahn, J.H., Green, M., Hohenberger, S.: Synchronized aggregate signatures: new definitions, constructions and applications. In: ACM Conference on Computer and Communications Security. pp. 473–484. ACM (2010)
2. Ben-Sasson, E., Chiesa, A., Tromer, E., Virza, M.: Succinct non-interactive zero knowledge for a von neumann architecture. In: USENIX Security Symposium. pp. 781–796. USENIX Association (2014)
3. Boldyreva, A., Gentry, C., O'Neill, A., Yum, D.H.: Ordered multisignatures and identity-based sequential aggregate signatures, with applications to secure routing. In: ACM Conference on Computer and Communications Security. pp. 276–285. ACM (2007)
4. Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and verifiably encrypted signatures from bilinear maps. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 2656, pp. 416–432. Springer (2003)
5. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the weil pairing. J. Cryptology **17**(4), 297–319 (2004)
6. Brogle, K., Goldberg, S., Reyzin, L.: Sequential aggregate signatures with lazy verification from trapdoor permutations - (extended abstract). In: ASIACRYPT. Lecture Notes in Computer Science, vol. 7658, pp. 644–662. Springer (2012)
7. Campanelli, M., Fiore, D., Querol, A.: Legosnark: Modular design and composition of succinct zero-knowledge proofs. In: ACM Conference on Computer and Communications Security. pp. 2075–2092. ACM (2019)
8. Danezis, G., Fournet, C., Groth, J., Kohlweiss, M.: Square span programs with applications to succinct NIZK arguments. In: ASIACRYPT (1). Lecture Notes in Computer Science, vol. 8873, pp. 532–550. Springer (2014)
9. Dorfman, R.: The detection of defective members of large populations. The Annals of Mathematical Statistics **Vol. 14, No. 4**, 436–440 (1943)

10. Du, D.Z., Hwang, F.K.: Combinatorial Group Testing and Its Applications (2nd Edition), Series on Applied Mathematics, vol. 12. World Scientific (2000)
11. Eppstein, D., Goodrich, M.T., Hirschberg, D.S.: Improved combinatorial group testing algorithms for real-world problem sizes. SIAM J. Comput. **36**(5), 1360–1375 (2007)
12. Fischlin, M., Lehmann, A., Schröder, D.: History-free sequential aggregate signatures. In: SCN. Lecture Notes in Computer Science, vol. 7485, pp. 113–130. Springer (2012)
13. Gennaro, R., Gentry, C., Parno, B., Raykova, M.: Quadratic span programs and succinct nizks without pcps. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 7881, pp. 626–645. Springer (2013)
14. Gennaro, R., Minelli, M., Nitulescu, A., Orrù, M.: Lattice-based zk-snarks from square span programs. In: ACM Conference on Computer and Communications Security. pp. 556–573. ACM (2018)
15. Gentry, C., O'Neill, A., Reyzin, L.: A unified framework for trapdoor-permutation-based sequential aggregate signatures. In: Public Key Cryptography (2). Lecture Notes in Computer Science, vol. 10770, pp. 34–57. Springer (2018)
16. Gentry, C., Ramzan, Z.: Identity-based aggregate signatures. In: Public Key Cryptography. Lecture Notes in Computer Science, vol. 3958, pp. 257–273. Springer (2006)
17. Gentry, C., Wichs, D.: Separating succinct non-interactive arguments from all falsifiable assumptions. In: STOC. pp. 99–108. ACM (2011)
18. Groth, J.: On the size of pairing-based non-interactive arguments. In: EUROCRYPT (2). Lecture Notes in Computer Science, vol. 9666, pp. 305–326. Springer (2016)
19. Groth, J., Kohlweiss, M., Maller, M., Meiklejohn, S., Miers, I.: Updatable and universal common reference strings with applications to zk-snarks. In: CRYPTO (3). Lecture Notes in Computer Science, vol. 10993, pp. 698–728. Springer (2018)
20. Hartung, G., Kaidel, B., Koch, A., Koch, J., Hartmann, D.: Practical and robust secure logging from fault-tolerant sequential aggregate signatures. In: ProvSec. Lecture Notes in Computer Science, vol. 10592, pp. 87–106. Springer (2017)
21. Hartung, G., Kaidel, B., Koch, A., Koch, J., Rupp, A.: Fault-tolerant aggregate signatures. In: Public Key Cryptography (1). Lecture Notes in Computer Science, vol. 9614, pp. 331–356. Springer (2016)
22. Hirose, S., Shikata, J.: Non-adaptive group-testing aggregate mac scheme. In: The 14th International Conference on Information Security Practice and Experience (ISPEC 2018). Lecture Notes in Computer Science, vol. 11125, pp. 357–372. Springer (2018)
23. Hohenberger, S., Sahai, A., Waters, B.: Full domain hash from (leveled) multilinear maps and identity-based aggregate signatures. In: CRYPTO (1). Lecture Notes in Computer Science, vol. 8042, pp. 494–512. Springer (2013)
24. Hohenberger, S., Waters, B.: Synchronized aggregate signatures from the RSA assumption. In: EUROCRYPT (2). Lecture Notes in Computer Science, vol. 10821, pp. 197–229. Springer (2018)
25. Hwang, F.K.: A method for detecting all defective members in a population by group testing. Journal of the American Statistical Association **Vol. 67, No. 339**, 605–608 (1972)
26. Katz, J., Lindell, A.Y.: Aggregate message authentication codes. In: CT-RSA. Lecture Notes in Computer Science, vol. 4964, pp. 155–169. Springer (2008)

27. Kosba, A.E., Papadopoulos, D., Papamanthou, C., Song, D.: MIRAGE: succinct arguments for randomized algorithms with applications to universal zk-snarks. In: USENIX Security Symposium. pp. 2129–2146. USENIX Association (2020)
28. Lee, K., Lee, D.H., Yung, M.: Sequential aggregate signatures made shorter. In: ACNS. Lecture Notes in Computer Science, vol. 7954, pp. 202–217. Springer (2013)
29. Li, C.H.: A sequential method for screening experimental variables. Journal of the American Statistical Association **Vol. 57, No. 298**, 455–477 (1962)
30. Lu, S., Ostrovsky, R., Sahai, A., Shacham, H., Waters, B.: Sequential aggregate signatures and multisignatures without random oracles. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 4004, pp. 465–485. Springer (2006)
31. Lysyanskaya, A., Micali, S., Reyzin, L., Shacham, H.: Sequential aggregate signatures from trapdoor permutations. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 3027, pp. 74–90. Springer (2004)
32. Maller, M., Bowe, S., Kohlweiss, M., Meiklejohn, S.: Sonic: Zero-knowledge snarks from linear-size universal and updatable structured reference strings. In: ACM Conference on Computer and Communications Security. pp. 2111–2128. ACM (2019)
33. Minematsu, K.: Efficient message authentication codes with combinatorial group testing. In: ESORICS (1). Lecture Notes in Computer Science, vol. 9326, pp. 185–202. Springer (2015)
34. Minematsu, K., Kamiya, N.: Symmetric-key corruption detection: When xor-macs meet combinatorial group testing. In: ESORICS 2019, Part I. Lecture Notes in Computer Science, vol. 11735, pp. 595–615. Springer (2019)
35. Neven, G.: Efficient sequential aggregate signed data. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 4965, pp. 52–69. Springer (2008)
36. Ogawa, Y., Sato, S., Shikata, J., Imai, H.: Aggregate message authentication codes with detecting functionality from biorthogonal codes. In: 2020 IEEE International Symposium on Information Theory (ISIT 2020). IEEE (2020)
37. Porat, E., Rothschild, A.: Explicit non-adaptive combinatorial group testing schemes. In: ICALP (1). Lecture Notes in Computer Science, vol. 5125, pp. 748–759. Springer (2008)
38. Rückert, M., Schröder, D.: Aggregate and verifiably encrypted signatures from multilinear maps without random oracles. In: ISA. Lecture Notes in Computer Science, vol. 5576, pp. 750–759. Springer (2009)
39. Sato, S., Hirose, S., Shikata, J.: Sequential aggregate macs with detecting functionality revisited. In: Network and System Security (NSS 2019). Lecture Notes in Computer Science, vol. 11928, pp. 387–407. Springer (2019)
40. Sato, S., Shikata, J.: Interactive aggregate message authentication scheme with detecting functionality. In: International Conference on Advanced Information Networking and Applications (AINA 2019). pp. 1316–1328. Springer (2019)
41. Sato, S., Shikata, J.: Interactive aggregate message authentication equipped with detecting functionality from adaptive group testing. In: Cryptology ePrint Archive. IACR (Oct 2020)
42. Thierry-Mieg, N.: A new pooling strategy for high-throughput screening: the shifted transversal design. BMC Bioinformatics **7**, 28 (2006)

Appendix A: Succinct Non-Interactive Argument of Knowledge

In this section, we give the definition of succinct non-interactive argument of knowledge (SNARK) and a concrete construction in [18].

A.1 Succinct Non-Interactive Argument of Knowledge

A succinct non-interactive argument of knowledge (SNARK) system for a relation $R \subseteq \{0,1\}^* \times \{0,1\}^*$ consists of three polynomial-time algorithms $(\mathsf{Gen}, \mathsf{P}, \mathsf{V})$:

- $(\mathsf{crs}, \mathsf{vrs}) \leftarrow \mathsf{Gen}(1^\lambda)$: Gen takes as input a security parameter 1^λ, and it outputs a common reference string (CRS) crs and a verification key vrs.
- $\pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w)$: P takes as input a CRS crs, a statement x, and a witness w, and it outputs a proof π.
- $1/0 \leftarrow \mathsf{V}(\mathsf{vrs}, x, \pi)$: V takes as input a verification key vrs, a statement x, and a proof π, and it outputs 1 or 0.

It is required that SNARK systems meet the following properties *completeness*, *knowledge-soundness*, and *succinctness*: Let $L(R) = \{x \mid \exists w \text{ s.t. } (x,w) \in R\}$ be the language defined by R.

Completeness For every $(x,w) \in R$, it holds that

$$\Pr[\mathsf{V}(\mathsf{vrs}, x, \pi) = 1 \mid (\mathsf{crs}, \mathsf{vrs}) \leftarrow \mathsf{Gen}(1^\lambda); \pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w)] \geq 1 - \mathsf{negl}(\lambda).$$

Knowledge-Soundness For any PPT cheating prover P^*, it holds that

$$\mathsf{Adv}^{\text{k-sound}}_{\mathsf{P}^*}(\lambda) := \Pr\left[\begin{array}{c|c} \mathsf{V}(\mathsf{vrs}, x, \pi) = 1 \wedge & (\mathsf{crs}, \mathsf{vrs}) \leftarrow \mathsf{Gen}(1^\lambda); \\ (x,w) \notin R & (x, \pi; w) \leftarrow (\mathsf{P}^* \parallel \mathsf{Ext}_{\mathsf{P}^*})^{\mathsf{V}^*}(\mathsf{crs}) \end{array}\right] \leq \mathsf{negl}(\lambda),$$

where the verifier oracle V^* takes as input a pair (x, π) of a statement and a proof, and outputs $1/0 \leftarrow \mathsf{V}(\mathsf{vrs}, x, \pi)$.

Succinctness There exists a universal polynomial poly such that the proof-size and the running time of Gen and V are at most $\mathsf{poly}(\lambda + |x|)$.

Definition 20 (Publicly Verifiable and Designated Verifier). *A SNARK system Π_{SNARK} is publicly verifiable if Π_{SNARK} meets knowledge-soundness against a cheating prover P^* which is given a verification key vrs. Π_{SNARK} is designated verifier if Π_{SNARK} meets knowledge-soundness against P^* which is not given vrs.*

A.2 SNARK System in [18]

The SNARK system for quadratic arithmetic programs $\Pi_{\mathsf{SNARK}_G} = (\mathsf{Gen}, \mathsf{P}, \mathsf{V})$ is as follows: Let G_1, G_2 be the base groups, g_1 and g_2 be the generators of G_1 and G_2, respectively. We write $[a]_1$, $[b]_2$, and $[c]_T$ for g_1^a, g_2^b, and $e(g_1,g_2)^c$, respectively, where $e : G_1 \times G_2 \to G_T$ is a bilinear map with target group G_T. A relation generator $\mathcal{R}$ returns relations of the form

$$R = (p, G_1, G_2, G_T, e, g_1, g_2, l, \{u_i(X), v_i(X), w_i(X)\}_{i \in \{0,1,\ldots,m\}}, t(X))$$

with $p = p(\lambda)$ for a security parameter λ and polynomials $u_i(X), v_i(X), w_i(X)$ with degree $n-1$. This relation defines a language of statements $(a_1, \ldots, a_t) \in \mathbb{Z}_p^l$ and witnesses $(a_{l+1}, \ldots, a_m) \in \mathbb{Z}_p^{m-l}$ such that

$$\sum_{i=0}^{m} a_i u_i(X) \cdot \sum_{i=0}^{m} a_i v_i(X) = \sum_{i=0}^{m} a_i w_i(X) + h(X)t(X),$$

for some quotient polynomial $h(X)$ with degree $n-2$.

– $(\mathsf{crs}, \mathsf{vrs}) \leftarrow \mathsf{Gen}(1^\lambda)$:
  1. $\alpha, \beta, \gamma, \delta, x \xleftarrow{\$} \mathbb{Z}_q^*$.
  2. $([\mathsf{crs}_1]_1, [\mathsf{crs}_2]_2)$ is computed as follows:

$$\mathsf{crs}_1 = \left(\begin{array}{c} \alpha, \beta, \delta, \{x\}_{i\in\{0,1,\ldots,n-1\}}, \{\frac{\beta u_i(x)+\alpha v_i(x)+w_i(x)}{\gamma}\}_{i\in\{0,1,\ldots,l\}}, \\ \{\frac{\beta u_i(x)+\alpha v_i(x)+w_i(x)}{\delta}\}_{i\in\{l+1,\ldots,m\}}, \{\frac{x^i t(x)}{\delta}\}_{i\in\{0,1,\ldots,n-2\}} \end{array} \right), \text{ and}$$
$$\mathsf{crs}_2 = \left(\beta, \gamma, \delta, \{x^i\}_{i\in\{0,1,\ldots,n-1\}}\right).$$

  3. Output $\mathsf{crs} = ([\mathsf{crs}_1]_1, [\mathsf{crs}_2]_2)$ and $\mathsf{vrs} = ([\mathsf{crs}_1]_1, [\mathsf{crs}_2]_2)$.

– $\pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w)$: For a statement $x = (a_1, \ldots, a_l) \in \mathbb{Z}_p^l$ and $w = (a_{l+1}, \ldots, a_m) \in \mathbb{Z}_p^{m-l}$, generate a proof π as follows:
  1. Choose $r, s \xleftarrow{\$} \mathbb{Z}_p$.
  2. Compute $([A]_1, [C]_1, [B]_2)$, where

$$A = \alpha + \sum_{i=0}^{m} a_i u_i(x) + r\delta, \qquad B = \beta + \sum_{i=0}^{m} a_i v_i(x) + s\delta,$$
$$C = \frac{\sum_{i=l+1}^{m} a_i(\beta u_i(x) + \alpha v_i(x) + w_i(x)) + h(x)t(x)}{\delta} + As + Br - rs\delta.$$

– $1/0 \leftarrow \mathsf{V}(\mathsf{vrs}, x, \pi)$: Output 1 if

$$[A]_1 \cdot [B]_2 = [\alpha]_1 \cdot [\beta]_2 + \sum_{i=0}^{l} a_i \left[\frac{\beta u_i(x) + \alpha v_i(x) + w_i(x)}{\gamma}\right]_1 \cdot [\gamma]_2 + [C]_1 \cdot [\delta]_2.$$

holds, and output 0 otherwise.

The following proposition proven in [18] shows the security of the above construction.

Proposition 4 ([18], Theorem 2). *The SNARK system Π_{SNARK_G} satisfies completeness. It also fulfills knowledge-soundness in the generic bilinear group model.*