On the Black-Box Separation Between Ring Signatures and Public Key Encryptions

Kyosuke Yamashita[1,2] and Keisuke Hara[2,3]

[1] Osaka University
[2] National Institute of Advanced Industrial Science and Technology
[3] Yokohama National University

Abstract. In this paper, we show that it is impossible to construct a public key encryption scheme (PKE) from a ring signature scheme in a black-box fashion in the standard model. Such an impossibility is highly non-trivial because, to the best of our knowledge, known generic constructions of ring signature scheme are based on public key cryptosystems or in the random oracle model. Technically, we introduce a new cryptographic primitive named indistinguishable multi-designated verifiers signature (IMDVS), and prove that (i) IMDVS is equivalent to PKE, and (ii) it is impossible to construct IMDVS from a ring signature scheme in a generic way. Our result suggests an essential gap between ring signature and group signature, as it is known that group signature implies PKE.

Keywords: black-box separation · ring signature · public key encryption · multi-designated verifiers signature.

1 Introduction

1.1 Black-Box Impossibility

Since the seminal work by Impagliazzo and Rudich [17], which shows that it is impossible to construct a key agreement scheme from a one-way function in a black-box fashion, it has been one of the most important tasks to investigate the relationship between cryptographic primitives. In general, such an impossibility is called *black-box impossiblity* (or *separation*).

Impagliazzo [16], demonstrates five possible worlds and their implications for computer science.[4] In particular, Cryptomania is the world where public key cryptography exists and Minicrypt is the world where a one-way function exists but public key cryptography does not. Loosely speaking, investigating the black-box (im)possibility of a cryptographic primitive is to uncover if it belongs to Cryptomania or Minicrypt.

Understanding the limitation of the power of a cryptographic primitive is useful, for instance, to cryptographic protocol design. That is, we prefer to use weaker primitives as building blocks for getting another cryptographic protocol.

[4] Another possible world is obfustopia [12], but we do not mention it in this paper.

Therefore, it is important to investigate if a cryptographic primitive belongs to Cryptomania or Minicrypt.

1.2 Our Problem

A ring signature scheme [27] is a signature scheme that equips with signer anonymity. In a ring signature scheme, potential signers constitute a group (or a ring), then a signer creates a signature on behalf of the group. Verifiers can confirm that the signature is indeed created by a member of the ring, but cannot detect who signed it. Thanks to this anonymity, it is expected to be used in many applications such as e-voting, e-cash, e-bidding, and e-lottery [26].

There are existing works that construct a ring signature scheme from cryptographic primitives so far, such as a public key encryption scheme, a signature scheme, and a two-message public-coin witness indistinguishability proof system (a.k.a. ZAP) [2, 3] and a trapdoor permutation in the random oracle model [28]. Furthermore, it is well known that we can obtain a ring signature scheme from an OR-proof system [7] by applying the Fiat-Shamir transformation [10] (thus in the random oracle model) [3].

To the best of our knowledge, all such generic constructions of a ring signature scheme are obtained from Cryptomania primitives or in the random oracle model. Particularly, no construction only from Minicrypt primitives has been proposed in the standard model. Therefore, the following question is still open.

Which does ring signature belong to Cryptomania or Minicrypt?

1.3 Our Contribution

In this paper, we provide strong evidence for the above question. Concretely, we prove that it is impossible to construct a PKE from a ring signature scheme in the standard model.[5]

Roughly, the separation is shown as follows. We first introduce a new cryptographic primitive called indistinguishable multi-designated verifiers signature (IMDVS). We then demonstrate that IMDVS is equivalent to PKE, but it is impossible to construct IMDVS from ring signature in a black-box fashion.

An IMDVS scheme is an extension of an MDVS scheme [8] in which only designated verifiers can verify a signature by using their secret verification keys. In addition, we require IMDVS to have *signature indistinguishability*, which guarantees that, even given a signature and two messages, a non-designated verifier cannot distinguish on which message the signature is created. We note that, while it was believed that (standard) MDVS can be obtained from ring signature in general [20, 21, 30, 33], recently it has been shown that such a construction is impossible in a black-box sense [32]. We emphasize that IMDVS is an artificial primitive just to prove our result. Therefore, it does not matter here how to instantiate it.

[5] Note that it is orthogonal to the construction from an OR-proof system as it is in the random oracle model.

1.4 Technical Overview

This section provides a technical overview of our result step by step. We first explain how IMDVS works. There are two key generation algorithms that output public key and secret key for signers and verifiers, respectively[6]. A signing algorithm takes a signer's secret key and a set of designated verifiers' public keys to sign on a message. A verification algorithm takes a set of designated verifiers' public keys, a signer's public keys, and one of the designated verifier's secret keys to verify a signature on a message. Besides unforgeability, we require signature indistinguishability, i.e. it is impossible for non-designated verifiers to decide if a pair of a message and a signature is valid or not.

(IMDVS $\rightarrow$ PKE). Firstly, we give an overview of our PKE scheme based on an IMDVS scheme. Our construction is inspired by the work of Okamoto [24], which demonstrates the equivalence between designated confirmer signature and PKE. We construct a 1-bit PKE scheme as follows. The key generation algorithm creates both a signer's and a verifier's keys of the underlying IMDVS scheme with respect to the same user. The public key is a pair of the signer's signing key and the verifier's public key and the secret key is a tuple of the signer's public key and the verifier's public and secret key. When encrypting a message $\mathsf{m} = 1$, the encryption algorithm creates a signature σ on the message m and outputs (σ, m) as a ciphertext. Otherwise, it outputs (σ, m') where $\mathsf{m}' \neq \mathsf{m}$. The decryption algorithm, given a ciphertext (σ, m), verifies σ on m by using its verifier's secret key. When the verification result is $b \in \{0, 1\}$, it interprets that the decryption result is b. The IND-CPA security of PKE follows from the signature indistinguishability of the underlying IMDVS scheme.

(PKE $\rightarrow$ IMDVS). Secondly, we give an overview of our IMDVS scheme based on a PKE scheme. We construct an IMDVS scheme from a PKE scheme and a (standard) signature scheme.[7] A signer's key pair is a key pair of the signature scheme and the verifier's key pair is a key pair of the PKE scheme. Suppose that there are n designated verifiers. When signing a message m, a signer creates a (standard) signature σ on the message m and generates a ciphertext c_i of σ by using each designated verifier's public key to output n ciphertexts $(c_1, \cdots, c_n)$ as the IMDVS signature. Given the IMDVS signature $(c_1, \cdots, c_n)$, a designated verifier decrypts a ciphertext whose index i corresponds to him and verifies the resulting signature by using the signer's public key of the underlying signature scheme.

(Ring Signature $\not\rightarrow$ IMDVS). Finally, we give an overview of the separation between ring signature and IMDVS. The proof is similar to that of [32], which uses the meta-reduction paradigm [13]. While we have to deal with subtleties in

[6] We require a setup algorithm that outputs a public parameter and a master secret key, but we do not mention it here.

[7] Note that it is known that PKE implies one-way function (that is, (standard) signature).

our proof, we here present a simplified version for easier understanding. When considering a construction of IMDVS from a ring signature scheme, a natural idea might be to regard a ring of the underlying ring signature scheme as a set of a signer and designated verifiers. However, we prove that it is impossible to reduce the unforgeability of such an IMDVS to that of the ring signature scheme.

Essentially, the impossibility stems from the difference in definitions of the unforgeability between IMDVS and ring signature. In a ring signature scheme, it is not allowed to corrupt a ring member, whereas in an IMDVS scheme, it is possible to corrupt a designated verifier. Let $\mathcal{A}$ be a polynomial time adversary who breaks unforgeability of the IMDVS scheme with non-negligible probability and R a polynomial time reduction algorithm who breaks unforgeability of the underlying ring signature scheme by accessing $\mathcal{A}$ in a black-box manner. Suppose that $\mathcal{A}$ makes a query that corrupts a designated verifier, necessitating R to corrupt a ring member. However, R cannot call its corruption oracle since this immediately violates the winning condition for R, and thus should answer the query from $\mathcal{A}$ by itself. If R is able to do this, it can break unforgeability of the underlying ring signature scheme without assuming the existence of $\mathcal{A}$ since it can create a signing key of a ring member by itself, which is a contradiction.

1.5 Implication of Our Result

We argue that our result provides two implications. One is for the nature of group-oriented signature schemes, and the other is for cryptographic protocol designs.

According to the survey by Perera et al. [26], it has been an important task to balance anonymity and traceability (i.e., an ability to identify a signer) in group-oriented signature schemes such as ring signature and group signature [6], where only group signature equips with traceability among them. Our result suggests an essential gap between them because it is known that group signature implies PKE [1, 9, 23]. Besides, achieving traceability in group-oriented signatures may require the same level of capability as public key cryptography, since it is known that accountable ring signature [31], which was proposed to fill the gap between ring signature and group signature, implies group signature (and thus PKE) [5].

As mentioned earlier, when we consider cryptographic designs of advanced primitives in a generic manner, it is preferable to employ building blocks as weak as possible. Since our result indicates that ring signature is strictly weaker than PKE, ring signature is more preferable alternative primitive as a building block rather than PKE.

1.6 Related Work

The line of black-box separation research has been successful, and there are many known results such as the impossibility of oblivious transfer from PKE [14], CCA-PKE from CPA-PKE (in a somewhat restricted model) [15], identity-based encryption scheme from trapdoor permutation [4]. Recently, it has been proven

that it is impossible to construct (standard) multi-designated verifiers signature from ring signature in a black-box fashion [32].

Several black-box (im)possibilities have been already known regarding signature schemes. It is widely known that standard signature scheme is equivalent to one-way function [29]. Considering the result of Impagliazzo and Rudich [17], standard signature scheme is separated from PKE. As mentioned earlier, it is known that group signature implies PKE [1, 9, 23]. Furthermore, blind signature cannot be obtained from one-way permutation [18].

Ring signature schemes that equip with various levels of tracing functionality have been proposed so far, such as accountable ring signature [31], linkable ring signature [22], traceable ring signature [11], deniable ring signature [19], and claimable/repudiable ring signature [25].

1.7 Paper Organization

The rest of our paper consists of the following. Section 2 introduces basic notation and definitions of some cryptographic primitives. In Section 3, we define IMDVS. Section 4 demonstrates the equivalence between IMDVS and PKE and in Section 5, we prove the separation between ring signature and IMDVS. Finally, Section 6 concludes the paper.

2 Preliminaries

2.1 Notation

Throughout this paper, we let $\lambda \in \mathbb{N}$ be a security parameter. We abbreviate a probabilistic polynomial time algorithm as a PPT algorithm. We denote a polynomial function and a negligible function by $\mathsf{poly}(\cdot)$ and $\mathsf{negl}(\cdot)$, respectively. For any $n \in \mathbb{N}$, let $[n] \coloneqq \{1, 2, \cdots, n\}$. A subroutine X of an algorithm Π is denoted by $\Pi.X$. A security property is defined by a game (or an experiment) between a challenger and an adversary. If the result of the game is 1, we say that the adversary wins the game.

2.2 Public Key Encryption

Definition 1 (Public Key Encryption). *A public key encryption (PKE) scheme with a message space $\mathcal{M}$ consists of three PPT algorithms* $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *that work as follows:*

- $\mathsf{KeyGen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$: *Given a security parameter 1^λ, it outputs a public key* pk *and a secret key* sk.
- $\mathsf{Enc}(\mathsf{pk}, \mathsf{m}) \to c$: *Given a public key* pk *and a message* m*, it outputs a ciphertext c.*
- $\mathsf{Dec}(\mathsf{sk}, c) \to \mathsf{m}$: *Given a secret key* sk *and a ciphertext c, it outputs a message* m.

A PKE scheme is correct if for any $\lambda \in \mathbb{N}$*, any* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$ *and any* $\mathsf{m} \in \mathcal{M}$*, it holds that* $\Pr[\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{m})) = \mathsf{m}] = 1 - \mathsf{negl}(\lambda)$.

Definition 2 (IND-CPA). *A PKE scheme* $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is indistinguishable under chosen plaintext attack (IND-CPA secure) if for any* $\lambda \in \mathbb{N}$*, any PPT stateful adversary* $\mathcal{A}$*, it holds that* $|\Pr[\mathsf{ExpINDCPA}_{\Pi,\mathcal{A}}(1^\lambda) = 1] - 1/2| \leq \mathsf{negl}(\lambda)$ *where the experiment* $\mathsf{ExpINDCPA}_{\Pi,\mathcal{A}}(1^\lambda)$ *is defined as follows:*

$\mathsf{ExpINDCPA}_{\Pi,\mathcal{A}}(\lambda)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda);$
$(\mathsf{m}_0, \mathsf{m}_1) \leftarrow \mathcal{A}(\mathsf{pk});$
$b \leftarrow \{0,1\}; c^* \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_b);$
$b' \leftarrow \mathcal{A}(c^*)$:
output 1 if $b' = b$, otherwise 0

2.3 Signature

Definition 3 (Signature). *A signature scheme with a message space* $\mathcal{M}$ *consists of three PPT algorithms* $(\mathsf{KG}, \mathsf{Sig}, \mathsf{Vrf})$ *that work as follows:*

- $\mathsf{KG}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$: *Given a security parameter* 1^λ*, it outputs a public key* pk *and a secret key* sk.
- $\mathsf{Sig}(\mathsf{sk}, \mathsf{m}) \to \sigma$: *Given a secret key* sk *and a message* m*, it outputs a signature* σ.
- $\mathsf{Vrf}(\mathsf{pk}, \mathsf{m}, \sigma) = 1/0$: *Given a public key* pk*, a message* m*, and a signature* σ*, it outputs* 1 *(meaning "valid") or* 0 *(meaning "invalid").*

A signature scheme $(\mathsf{KG}, \mathsf{Sig}, \mathsf{Vrf})$ *is correct if for any security parameter* λ*, any* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KG}(1^\lambda)$*, and any message* $\mathsf{m} \in \mathcal{M}$*, it holds that* $\mathsf{Vrf}(\mathsf{pk}, \mathsf{m}, \mathsf{Sig}(\mathsf{sk}, \mathsf{m})) = 1$.

Definition 4 (EUF-CMA). *A signature scheme* $\Pi = (\mathsf{KG}, \mathsf{Sig}, \mathsf{Vrf})$ *is existentially unforgeable under an adaptive chosen-message attack (EUF-CMA secure) if for any sufficiently large security parameter* λ *and any PPT adversary* $\mathcal{A}$*, it holds that* $\Pr[\mathsf{ExpEUFSig}_{\Pi,\mathcal{A}}(1^\lambda) = 1] \leq \mathsf{negl}(\lambda)$*, where* $\mathsf{ExpEUFSigExpEUFSig}_{\Pi,\mathcal{A}}(1^\lambda)$ *is defined as follows:*

$\mathsf{ExpEUFSig}_{\Pi,\mathcal{A}}(1^\lambda)$
$L := \emptyset; (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KG}(1^\lambda);$
$(\mathsf{m}^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{Sig}}}(\mathsf{pk})$:
output 1 if $\mathsf{Vrf}(\mathsf{pk}, \mathsf{m}^*, \sigma^*) = 1 \wedge (\mathsf{m}^*, \cdot) \notin L$,
otherwise 0

where $\mathsf{O}_{\mathsf{Sig}}$ *works as follows: Given a message* m*, it returns* σ *if* $(\mathsf{m}, \sigma) \in L$*. Otherwise, it returns* $\sigma \leftarrow \mathsf{Sig}(\mathsf{sk}, \mathsf{m})$ *and updates* $L := L \cup \{(\mathsf{m}, \sigma)\}$.

2.4 Multi-Designated Verifier Signature

In this section, we recall the definition of multi-designated verifier signature (MDVS). We follow the most standard definition of MDVS by [8] except that all designated verifiers are required to participate to simulate a signature.[8] They claim that the basic security requirements for MDVS are unforgeability, OTR, and consistency. Namely, consistency is a property that guarantees that verification results are the same among designated verifiers.

Let $\mathcal{I}$ denote a set of users' identities and we use $\mathcal{I}$ in the definition of an MDVS scheme. The formal definition is as follows. [9]

Definition 5 (MDVS). *A multi-designated verifier signature scheme (MDVS) scheme consists of the following six algorithms* $(\mathsf{Set}, \mathsf{SKG}, \mathsf{VKG}, \mathsf{Sig}, \mathsf{Vrf}, \mathsf{Sim})$*:*

- $\mathsf{Set}(1^\lambda) \to (\mathsf{pp}, \mathsf{msk})$: *Given a security parameter* 1^λ*, it outputs a public parameter* pp *and a master secret key* msk*.*
- $\mathsf{SKG}(\mathsf{pp}, \mathsf{msk}, \mathsf{id}_\mathsf{S}) \to (\mathsf{spk}_{\mathsf{id}_\mathsf{S}}, \mathsf{ssk}_{\mathsf{id}_\mathsf{S}})$: *Given a public parameter* pp*, a master secret key* msk*, and an identity* $\mathsf{id}_\mathsf{S} \in \mathcal{I}$*, it outputs the signer's public key* $\mathsf{spk}_{\mathsf{id}_\mathsf{S}}$ *and secret key* $\mathsf{ssk}_{\mathsf{id}_\mathsf{S}}$*.*
- $\mathsf{VKG}(\mathsf{pp}, \mathsf{msk}, \mathsf{id}_\mathsf{V}) \to (\mathsf{vpk}_{\mathsf{id}_\mathsf{V}}, \mathsf{vsk}_{\mathsf{id}_\mathsf{V}})$: *Given a public parameter* pp*, a master secret key* msk*, and an identity* $\mathsf{id}_\mathsf{V} \in \mathcal{I}$*, it outputs the verifier's public key* $\mathsf{vpk}_{\mathsf{id}_\mathsf{V}}$ *and secret key* $\mathsf{vsk}_{\mathsf{id}_\mathsf{V}}$*.*
- $\mathsf{Sig}(\mathsf{pp}, \mathsf{ssk}_{\mathsf{id}_\mathsf{S}}, \{\mathsf{vpk}_{\mathsf{id}_\mathsf{V}}\}_{\mathsf{id}_\mathsf{V} \in \mathcal{D}}, \mathsf{m}) \to \sigma$: *Given a public parameter* pp*, a signer's secret key* $\mathsf{ssk}_{\mathsf{id}_\mathsf{S}}$*, a set of verifiers' public keys* $\{\mathsf{vpk}_{\mathsf{id}_\mathsf{V}}\}_{\mathsf{id}_\mathsf{V} \in \mathcal{D}}$ *of designated verifiers* $\mathcal{D}$*, and a message* $\mathsf{m} \in \mathcal{M}$*, it outputs a signature* σ*.*
- $\mathsf{Vrf}(\mathsf{pp}, \{\mathsf{vpk}_{\mathsf{id}_\mathsf{V}}\}_{\mathsf{id}_\mathsf{V} \in \mathcal{D}}, \mathsf{vsk}_{\mathsf{id}'_\mathsf{V}}, \mathsf{spk}_{\mathsf{id}_\mathsf{S}}, \mathsf{m}, \sigma) \to 1/0$: *Given a public parameter* pp*, a set of public keys* $\{\mathsf{vpk}_{\mathsf{id}_\mathsf{V}}\}_{\mathsf{id}_\mathsf{V} \in \mathcal{D}}$ *of designated verifiers* $\mathcal{D}$*, a verifier's secret key* $\mathsf{vsk}_{\mathsf{id}'_\mathsf{V}}$*, a signer's public key* $\mathsf{spk}_{\mathsf{id}_\mathsf{S}}$*, a message* m*, and a signature* σ*, it outputs* 1 *(meaning accept) or* 0 *(meaning reject).*
- $\mathsf{Sim}(\mathsf{pp}, \{\mathsf{vpk}_{\mathsf{id}_\mathsf{V}}\}_{\mathsf{id}_\mathsf{V} \in \mathcal{D}}, \{\mathsf{vsk}_{\mathsf{id}_\mathsf{V}}\}_{\mathsf{id}_\mathsf{V} \in \mathcal{D}}, \mathsf{spk}_{\mathsf{id}_\mathsf{S}}, \mathsf{m}) \to \sigma$: *Given a public parameter* pp*, a set of public keys* $\{\mathsf{vpk}_{\mathsf{id}_\mathsf{V}}\}_{\mathsf{id}_\mathsf{V} \in \mathcal{D}}$ *of designated verifiers* $\mathcal{D}$*, a set of secret keys* $\{\mathsf{vsk}_{\mathsf{id}_\mathsf{V}}\}_{\mathsf{id} \in \mathcal{D}}$ *of designated verifiers* $\mathcal{D}$*, a signer's public key* $\mathsf{spk}_{\mathsf{id}_\mathsf{S}}$*, and a message* m*, it outputs a simulated signature* σ*.*

Definition 6 (Correctness). *An MDVS scheme* $\Pi = (\mathsf{Set}, \mathsf{SKG}, \mathsf{VKG}, \mathsf{Sig}, \mathsf{Vrf}, \mathsf{Sim})$ *satisfies correctness if for any security parameter* $\lambda \in \mathbb{N}$*, any* $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Set}(1^\lambda)$*, any set of verifiers' identities* $\mathcal{D} \subseteq \mathcal{I}$*, any verifier's identity* $\mathsf{id}'_\mathsf{V} \in \mathcal{D}$*, any signer's identity* $\mathsf{id}_\mathsf{S} \in \mathcal{I}$*, and any message* $\mathsf{m} \in \mathcal{M}$*, it holds that*

$$\mathsf{Vrf}(\mathsf{pp}, \{\mathsf{vpk}_{\mathsf{id}_\mathsf{V}}\}_{\mathsf{id}_\mathsf{V} \in \mathcal{D}}, \mathsf{vsk}_{\mathsf{id}'_\mathsf{V}}, \mathsf{spk}_{\mathsf{id}_\mathsf{S}}, \mathsf{m}, \mathsf{Sig}(\mathsf{pp}, \mathsf{ssk}_{\mathsf{id}_\mathsf{S}}, \{\mathsf{vpk}_{\mathsf{id}_\mathsf{V}}\}_{\mathsf{id}_\mathsf{V} \in \mathcal{D}}, \mathsf{m})) = 1,$$

[8] Note that this setting is limited compared to one by [8] in the sense that their definition considers simulation by any subset of designated verifiers. However, we stress that adopting a weaker definition makes our result better since our goal is to show a black-box impossibility from a ring signature scheme to an MDVS scheme.

[9] Note that, using $\mathcal{I}$, we give each algorithm an identifier only to make a user explicit. That is, we do not consider so-called "identity-based" primitives (e.g., identity-based signature).

where $(\mathsf{spk}_{\mathsf{id_S}}, \mathsf{ssk}_{\mathsf{id_S}}) \leftarrow \mathsf{SKG}(\mathsf{pp}, \mathsf{msk}, \mathsf{id_S})$ *and* $(\mathsf{vpk}_{\mathsf{id_V}}, \mathsf{vsk}_{\mathsf{id_V}}) \leftarrow \mathsf{VKG}(\mathsf{pp}, \mathsf{msk}, \mathsf{id_V})$ *for all* $\mathsf{id_V} \in \mathcal{D}$.

We require an MDVS scheme to satisfy unforgeability, consistency, and off-the-record (OTR) as security requirements, as discussed in [8].

Definition 7 (EUF-CMA). *An MDVS scheme* $\Pi = (\mathsf{Set}, \mathsf{SKG}, \mathsf{VKG}, \mathsf{Sig}, \mathsf{Vrf}, \mathsf{Sim})$ *is existentially unforgeable under an adaptive chosen-message attack (EUF-CMA) if for any security parameter* $\lambda \in \mathbb{N}$*, and any PPT adversary* $\mathcal{A}$*, it holds that* $\Pr[\mathsf{ExpEUFDVS}_{\Pi,\mathcal{A}}(1^\lambda) = 1] \leq \mathsf{negl}(\lambda)$ *where* $\mathsf{ExpEUFDVS}$ *is defined as follows:*

$\mathsf{ExpEUFDVS}_{\Pi,\mathcal{A}}(1^\lambda)$

$L_{\mathsf{VPK}} := \emptyset; L_{\mathsf{SPK}} := \emptyset; L_{\mathsf{VSK}} := \emptyset; L_{\mathsf{SSK}} := \emptyset; L_{\mathsf{Sign}} := \emptyset; L_{\mathsf{Vrf}} := \emptyset;$
$(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Set}(1^\lambda);$
$(\mathsf{id}^*_{\mathsf{S}}, \mathcal{D}^*, \mathsf{m}^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{O_{SPK}}, \mathsf{O_{SSK}}, \mathsf{O_{VPK}}, \mathsf{O_{VSK}}, \mathsf{O_{Sig}}, \mathsf{O_{Vrf}}}(\mathsf{pp})$:
Output 1 if $(\exists \mathsf{id}'_{\mathsf{V}} \in \mathcal{D}^* \setminus L_{\mathsf{VSK}}$ s.t. $\mathsf{Vrf}(\mathsf{pp}, \{\mathsf{vpk}_{\mathsf{id_V}}\}_{\mathsf{id_V} \in \mathcal{D}^*}, \mathsf{vsk}_{\mathsf{id}'_{\mathsf{V}}}, \mathsf{spk}_{\mathsf{id}^*_{\mathsf{S}}}, m^*, \sigma^*) = 1)$
$\wedge (\mathsf{id}^*_{\mathsf{S}} \in L_{\mathsf{SPK}}) \wedge ((\mathsf{id}^*_{\mathsf{S}}, \mathsf{spk}_{\mathsf{id}^*_{\mathsf{S}}}, \mathsf{spk}_{\mathsf{id}^*_{\mathsf{S}}}) \notin L_{\mathsf{SSK}})$
$\wedge (\forall \mathsf{id}'_{\mathsf{V}} \in \mathcal{D}^*, (\mathsf{id}'_{\mathsf{V}}, \mathsf{vpk}_{\mathsf{id}'_{\mathsf{V}}}, \mathsf{vsk}_{\mathsf{id}'_{\mathsf{V}}}) \in L_{\mathsf{VPK}}) \wedge ((\mathcal{D}^*, \mathsf{id}^*_{\mathsf{S}}, \mathsf{m}^*) \notin L_{\mathsf{Sign}})$
otherwise 0

where $\mathsf{O_{SPK}}, \mathsf{O_{SSK}}, \mathsf{O_{VPK}}, \mathsf{O_{VSK}}, \mathsf{O_{Sig}}$, *and* $\mathsf{O_{Vrf}}$ *work as follows:*

$\mathsf{O_{SPK}}$**:** *Given* $\mathsf{id_S} \in \mathcal{I}$*, if* $\mathsf{id_S}$ *has already been queried previously, then it searches for* $(\mathsf{id_S}, \mathsf{spk}_{\mathsf{id_S}}, \mathsf{ssk}_{\mathsf{id_S}})$ *from* L_{SPK} *and returns* $\mathsf{spk}_{\mathsf{id_S}}$*. Otherwise, it computes* $(\mathsf{spk}_{\mathsf{id_S}}, \mathsf{ssk}_{\mathsf{id_S}}) \leftarrow \mathsf{SKG}(\mathsf{pp}, \mathsf{msk}, \mathsf{id_S})$*, returns* $\mathsf{spk}_{\mathsf{id_S}}$*, and updates* $L_{\mathsf{SPK}} := L_{\mathsf{SPK}} \cup \{(\mathsf{id_S}, \mathsf{spk}_{\mathsf{id_S}}, \mathsf{ssk}_{\mathsf{id_S}})\}$.

$\mathsf{O_{SSK}}$**:** *Given* $\mathsf{id_S} \in \mathcal{I}$*, if* $(\mathsf{id_S}, \mathsf{spk}_{\mathsf{id_S}}, \mathsf{ssk}_{\mathsf{id_S}}) \in L_{\mathsf{SPK}}$*, then it returns* $\mathsf{ssk}_{\mathsf{id_S}}$*, and updates* $L_{\mathsf{SSK}} := L_{\mathsf{SSK}} \cup \{\mathsf{id_S}\}$*. Otherwise, it calls* $\mathsf{O_{SPK}}(\mathsf{id_S})$ *to generate* $(\mathsf{spk}_{\mathsf{id_S}}, \mathsf{ssk}_{\mathsf{id_S}})$ *along with updating* $L_{\mathsf{SPK}} := L_{\mathsf{SPK}} \cup \{(\mathsf{id_S}, \mathsf{spk}_{\mathsf{id_S}}, \mathsf{ssk}_{\mathsf{id_S}})\}$*, returns* $(\mathsf{spk}_{\mathsf{id_S}}, \mathsf{ssk}_{\mathsf{id_S}})$*, and updates* $L_{\mathsf{SSK}} := L_{\mathsf{SSK}} \cup \{\mathsf{id_S}\}$*. Note that we regard the signer corresponding to* $\mathsf{id_S} \in L_{\mathsf{SSK}}$ *as a corrupted signer.*

$\mathsf{O_{VPK}}$**:** *Given* $\mathsf{id_V} \in \mathcal{I}$*, if* $\mathsf{id_V}$ *has already been queried previously, then it searches for* $(\mathsf{id_V}, \mathsf{vpk}_{\mathsf{id_V}}, \mathsf{vsk}_{\mathsf{id_V}})$ *from* L_{VPK} *and returns* $\mathsf{vpk}_{\mathsf{id_V}}$*. Otherwise, it computes* $(\mathsf{vpk}_{\mathsf{id_V}}, \mathsf{vsk}_{\mathsf{id_V}}) \leftarrow \mathsf{VKG}(\mathsf{pp}, \mathsf{msk}, \mathsf{id_V})$*, returns* $\mathsf{vpk}_{\mathsf{id_V}}$*, and updates* $L_{\mathsf{VPK}} := L_{\mathsf{VPK}} \cup \{(\mathsf{id_V}, \mathsf{vpk}_{\mathsf{id_V}}, \mathsf{vsk}_{\mathsf{id_V}})\}$.

$\mathsf{O_{VSK}}$**:** *Given* $\mathsf{id_V} \in \mathcal{I}$*, if* $(\mathsf{id_V}, \mathsf{vpk}_{\mathsf{id_V}}, \mathsf{vsk}_{\mathsf{id_V}}) \in L_{\mathsf{VPK}}$*, then it returns* $\mathsf{vsk}_{\mathsf{id_V}}$*, and updates* $L_{\mathsf{VSK}} := L_{\mathsf{VSK}} \cup \{\mathsf{id_V}\}$*. Otherwise, it calls* $\mathsf{O_{VPK}}(\mathsf{id_V})$ *to generate* $(\mathsf{vpk}_{\mathsf{id_V}}, \mathsf{vsk}_{\mathsf{id_V}})$ *along with* $L_{\mathsf{VPK}} := L_{\mathsf{VPK}} \cup \{(\mathsf{id_V}, \mathsf{vpk}_{\mathsf{id_V}}, \mathsf{vsk}_{\mathsf{id_V}})\}$*, returns* $(\mathsf{vpk}_{\mathsf{id_V}}, \mathsf{vsk}_{\mathsf{id_V}})$*, and updates* $L_{\mathsf{VSK}} := L_{\mathsf{VSK}} \cup \{\mathsf{id_V}\}$*. Note that we regard the verifier corresponding to* $\mathsf{id_V} \in L_{\mathsf{VSK}}$ *as a corrupted verifier.*

$\mathsf{O_{Sig}}$**:** *Given* $\mathcal{D} \subseteq \mathcal{I}, \mathsf{id_S} \in \mathcal{I}$*, and* $\mathsf{m} \in \mathcal{M}$*, it does the followings:*

- *If* $(\mathsf{id_S}, \cdot, \cdot) \notin L_{\mathsf{SPK}}$*, then call* $\mathsf{O_{SPK}}$ *on* $\mathsf{id_S}$ *to generate* $(\mathsf{spk}_{\mathsf{id_S}}, \mathsf{ssk}_{\mathsf{id_S}})$.
- *For all* $\mathsf{id_V} \in \mathcal{D}$ *s.t.* $(\mathsf{id_V}, \cdot, \cdot) \notin L_{\mathsf{VPK}}$*, call* $\mathsf{O_{VPK}}$ *on* $\mathsf{id_V}$ *to generate* $(\mathsf{vpk}_{\mathsf{id_V}}, \mathsf{vsk}_{\mathsf{id_V}})$.
- *Return* $\sigma \leftarrow \mathsf{Sig}(\mathsf{pp}, \mathsf{ssk}_{\mathsf{id_S}}, \{\mathsf{vpk}_{\mathsf{id_V}}\}_{\mathsf{id_V} \in \mathcal{D}}, \mathsf{m})$*, and update* $L_{\mathsf{Sign}} := L_{\mathsf{Sign}} \cup \{(\mathcal{D}, \mathsf{id_S}, \mathsf{m})\}$.

$\mathsf{O}_{\mathsf{Vrf}}$**:** *Given* $\mathsf{id}'_{\mathsf{V}}, \mathsf{id}_{\mathsf{S}} \in \mathcal{I}, \mathsf{m} \in \mathcal{M}$, $\mathcal{D} \subseteq \mathcal{I}$ *where* $\mathsf{id}'_{\mathsf{V}} \in \mathcal{D}$, *and* σ, *it does the followings:*

- *If* $\mathsf{id}'_{\mathsf{V}} \notin \mathcal{D}$, *then return* 0.
- *If* $(\mathsf{id}_{\mathsf{S}}, \cdot, \cdot) \notin L_{\mathsf{SPK}}$, *then call* $\mathsf{O}_{\mathsf{SPK}}$ *on* id_{S} *to generate* $(\mathsf{spk}_{\mathsf{id}_{\mathsf{S}}}, \mathsf{ssk}_{\mathsf{id}_{\mathsf{S}}})$.
- *For all* $\mathsf{id}_{\mathsf{V}} \in \mathcal{D}$, *if* $(\mathsf{id}_{\mathsf{V}}, \cdot, \cdot) \notin L_{\mathsf{VPK}}$, *then call* $\mathsf{O}_{\mathsf{VPK}}$ *on* id_{V} *to generate* $(\mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}, \mathsf{vsk}_{\mathsf{id}_{\mathsf{V}}})$.
- *Return* $b = \mathsf{Vrf}(\mathsf{pp}, \{\mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}\}_{\mathsf{id}_{\mathsf{V}} \in \mathcal{D}}, \mathsf{vsk}_{\mathsf{id}'_{\mathsf{V}}}, \mathsf{spk}_{\mathsf{id}_{\mathsf{S}}}, \mathsf{m}, \sigma)$ *and update* $L_{\mathsf{Vrf}} := L_{\mathsf{Vrf}} \cup \{(\mathcal{D}, \mathsf{id}'_{\mathsf{V}}, \mathsf{id}_{\mathsf{S}}, \mathsf{m}, \sigma)\}$.

2.5 Ring Signature

In this section, we review the definition of ring signature. We follow the strongest definition by [3]. Namely, as security properties for ring signature, we require unforgeability w.r.t. insider corruptions and anonymity against full key exposure. We remark that this stronger definition makes our result better, as it means an MDVS scheme cannot be obtained from such a stronger ring signature scheme in a black-box manner.

Definition 8 (Ring Signature). *A ring signature scheme* $(\mathsf{Set}, \mathsf{KG}, \mathsf{Sig}, \mathsf{Vrf})$ *that work as follows:*

- $\mathsf{Set}(1^\lambda) \to \mathsf{pp}$: *Given a security parameter* 1^λ, *it outputs a public parameter* pp.
- $\mathsf{KG}(\mathsf{pp}) \to (\mathsf{pk}, \mathsf{sk})$: *Given a public parameter* pp, *it outputs a public key* pk *and a secret key* sk.
- $\mathsf{Sig}(\mathsf{pp}, \mathsf{sk}, \{\mathsf{pk}_i\}_{i\in[n]}, \mathsf{m}) \to \sigma$: *Given a public parameter* pp, *a secret key* sk, *a set of public keys (or a ring)* $\{\mathsf{pk}_i\}_{i\in[n]}$ *where* $n = \mathsf{poly}(\lambda)$, *and a message* m, *it outputs a signature* σ. *If there is no* $i \in [n]$ *s.t.* $(\mathsf{pk}_i, \mathsf{sk}) \leftarrow \mathsf{Set}(\mathsf{pp})$, *then it returns* $\perp$.
- $\mathsf{Vrf}(\mathsf{pp}, \{\mathsf{pk}_i\}_{i\in[n]}, \mathsf{m}, \sigma) = 1/0$: *Given a public parameter* pp, *a set of public keys* $\{\mathsf{pk}_i\}_{i\in[n]}$ *where* $n = \mathsf{poly}(\lambda)$, *a message* m, *and a signature* σ, *it outputs* 1 *(meaning accept) or* 0 *(meaning reject).*

A ring signature scheme $(\mathsf{Set}, \mathsf{KG}, \mathsf{Sig}, \mathsf{Vrf})$ *satisfies correctness if for any security parameter* λ, *any* $\mathsf{pp} \leftarrow \mathsf{Set}(1^\lambda)$, *and any message* $\mathsf{m} \in \mathcal{M}$, *it holds that*

$$\mathsf{Vrf}(\mathsf{pp}, \{\mathsf{pk}_i\}_{i\in[n]}, \mathsf{m}, \mathsf{Sig}(\mathsf{pp}, \mathsf{sk}, \{\mathsf{pk}_i\}_{i\in[n]}, \mathsf{m})) = 1,$$

where for any $i \in [n]$, pk_i *is generated by* KG, *and in particular, there exists* $i \in [n]$ *s.t.* $(\mathsf{pk}_i, \mathsf{sk}) \leftarrow \mathsf{KG}(\mathsf{pp})$.

Next, we define the unforgeability w.r.t. insider corruption as follows. Similar to MDVS, Anonymity is provided in Appendix **??**, as it does not appear in our discussion.

Definition 9 (Unforgeability w.r.t. Insider Corruptions). *A ring signature scheme* $\Pi_{\mathrm{RS}} = (\mathsf{Set}, \mathsf{KG}, \mathsf{Sig}, \mathsf{Vrf})$ *satisfies unforgeability w.r.t. insider corruptions if for any security parameter* λ *and any PPT adversary* $\mathcal{A}$ *who is allowed to make at most* $q = \mathsf{poly}(\lambda)$ *queries to oracles,* $\Pr[\mathsf{ExpEUFRS}_{\Pi_{\mathrm{RS}},\mathcal{A}}(1^\lambda) = 1] \leq \mathsf{negl}(\lambda)$ *where the experiment* $\mathsf{ExpEUFRS}_{\Pi_{\mathrm{RS}},\mathcal{A}}(1^\lambda)$ *is defined as follows:*

$\mathsf{ExpEUFRS}_{\Pi_{\mathrm{RS}},\mathcal{A}}(1^\lambda)$

$L_{\mathsf{PK}} \coloneqq \emptyset; L_{\mathsf{SK}} \coloneqq \emptyset; L_{\mathsf{Sign}} \coloneqq \emptyset; \mathsf{pp} \leftarrow \mathsf{Set}(1^\lambda);$
$(\{\mathsf{pk}_i^*\}_{i\in[n]}, \mathsf{m}^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{PK}},\mathsf{O}_{\mathsf{SK}},\mathsf{O}_{\mathsf{RSig}}}(\mathsf{pp})$:
Output 1 if $(\mathsf{Vrf}(\mathsf{pp}, \{\mathsf{pk}_i^*\}_{i\in[n]}, \mathsf{m}^*, \sigma^*) = 1) \wedge (\forall i \in [n], (\mathsf{pk}_i^*, \mathsf{sk}_i^*) \in L_{\mathsf{PK}})$
$\wedge(\forall i \in [n], (\mathsf{pk}_i^*, \mathsf{sk}_i^*) \notin L_{\mathsf{SK}}) \wedge (\forall j \in [n], (\mathsf{pk}_j^*, \{\mathsf{pk}_i^*\}_{i\in[n]\setminus\{j\}}, \mathsf{m}^*, \sigma^*) \notin L_{\mathsf{Sign}})$,
otherwise 0

where $n = \mathsf{poly}(\lambda)$ *s.t.* $n \leq q$*, and* $\mathsf{O}_{\mathsf{PK}}, \mathsf{O}_{\mathsf{SK}}$ *and* $\mathsf{O}_{\mathsf{RSig}}$ *work as follows:*

O_{PK}**:** *Given* pp*, it computes* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KG}(\mathsf{pp})$*, returns* pk*, and updates* $L_{\mathsf{PK}} \coloneqq L_{\mathsf{PK}} \cup \{(\mathsf{pk}, \mathsf{sk})\}$.

O_{SK}**:** *Given* pk*, if* $(\mathsf{pk}, \mathsf{sk}) \in L_{\mathsf{PK}}$*, then it returns* sk*, and updates* $L_{\mathsf{SK}} \coloneqq L_{\mathsf{SK}} \cup \{(\mathsf{pk}, \mathsf{sk})\}$*. Otherwise, it returns* $\perp$*. Note that we regard* L_{SK} *as a set of corrupted entities.*

$\mathsf{O}_{\mathsf{RSig}}$**:** *Given a signer's public key* pk*, a set of public keys* $\{\mathsf{pk}_i\}_{i\in[n']}$ *where* $n' = \mathsf{poly}(\lambda)$*, and a message* m*, it does the followings:*

- *If* $(\mathsf{pk}, \mathsf{sk}) \notin L_{\mathsf{PK}}$*, then returns* $\perp$.
- *If* $(\mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n']}, \mathsf{m}, \sigma) \in L_{\mathsf{Sign}}$*, then returns* σ.
- *Returns* $\sigma \leftarrow \mathsf{Sig}(\mathsf{pp}, \mathsf{sk}, \{\mathsf{pk}\} \cup \{\mathsf{pk}_i\}_{i\in[n']}, \mathsf{m})$ *and updates* $L_{\mathsf{Sign}} \coloneqq L_{\mathsf{Sign}} \cup \{(\mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n']}, \mathsf{m}, \sigma)\}$.

In the following, for simplicity, we say that a ring signature scheme satisfies EUF-CMA security if it satisfies the above definition.

3 Indistinguishable MDVS

Definition 10 (IMDVS). *An indistinguishable multi-designated verifier signature scheme (IMDVS) scheme consists of the following five algorithms* $(\mathsf{Set}, \mathsf{SKG}, \mathsf{VKG}, \mathsf{Sig}, \mathsf{Vrf})$*, where* Vrf *is deterministic and others are probabilistic:*

- $\mathsf{Set}(1^\lambda) \to (\mathsf{pp}, \mathsf{msk})$: *Given a security parameter* 1^λ*, it outputs a public parameter* pp *and a master secret key* msk.
- $\mathsf{SKG}(\mathsf{pp}, \mathsf{msk}, \mathsf{id}_\mathsf{S}) \to (\mathsf{spk}_{\mathsf{id}_\mathsf{S}}, \mathsf{ssk}_{\mathsf{id}_\mathsf{S}})$: *Given a public parameter* pp*, a master secret key* msk*, and an identity* $\mathsf{id}_\mathsf{S} \in \mathcal{I}$*, it outputs the signer's public key* $\mathsf{spk}_{\mathsf{id}_\mathsf{S}}$ *and secret key* $\mathsf{ssk}_{\mathsf{id}_\mathsf{S}}$.
- $\mathsf{VKG}(\mathsf{pp}, \mathsf{msk}, \mathsf{id}_\mathsf{V}) \to (\mathsf{vpk}_{\mathsf{id}_\mathsf{V}}, \mathsf{vsk}_{\mathsf{id}_\mathsf{V}})$: *Given a public parameter* pp*, a master secret key* msk*, and an identity* $\mathsf{id}_\mathsf{V} \in \mathcal{I}$*, it outputs the verifier's public key* $\mathsf{vpk}_{\mathsf{id}_\mathsf{V}}$ *and secret key* $\mathsf{vsk}_{\mathsf{id}_\mathsf{V}}$.
- $\mathsf{Sig}(\mathsf{pp}, \mathsf{ssk}_{\mathsf{id}_\mathsf{S}}, \{\mathsf{vpk}_{\mathsf{id}_\mathsf{V}}\}_{\mathsf{id}_\mathsf{V}\in\mathcal{D}}, \mathsf{m}) \to \sigma$: *Given a public parameter* pp*, a signer's secret key* $\mathsf{ssk}_{\mathsf{id}_\mathsf{S}}$*, a set of verifiers' public keys* $\{\mathsf{vpk}_{\mathsf{id}_\mathsf{V}}\}_{\mathsf{id}_\mathsf{V}\in\mathcal{D}}$ *of designated verifiers* $\mathcal{D}$*, and a message* $\mathsf{m} \in \mathcal{M}$*, it outputs a signature* σ.

– $\mathsf{Vrf}(\mathsf{pp}, \{\mathsf{vpk}_{\mathsf{id_V}}\}_{\mathsf{id_V}\in\mathcal{D}}, \mathsf{vsk}_{\mathsf{id'_V}}, \mathsf{spk}_{\mathsf{id_S}}, \mathsf{m}, \sigma) \to 1/0$: *Given a public parameter* pp*, a set of public keys* $\{\mathsf{vpk}_{\mathsf{id_V}}\}_{\mathsf{id_V}\in\mathcal{D}}$ *of designated verifiers* $\mathcal{D}$*, a verifier's secret key* $\mathsf{vsk}_{\mathsf{id'_V}}$*, a signer's public key* $\mathsf{spk}_{\mathsf{id_S}}$*, a message* m*, and a signature* σ*, it outputs* 1 *(meaning accept) or* 0 *(meaning reject).*

Definition 11 (EUF-CMA). *An IMDVS scheme* $\Pi = (\mathsf{Set}, \mathsf{SKG}, \mathsf{VKG}, \mathsf{Sig}, \mathsf{Vrf})$ *is existentially unforgeable under an adaptive chosen-message attack (EUF-CMA) if for any security parameter* $\lambda \in \mathbb{N}$*, and any PPT adversary* $\mathcal{A}$*, it holds that* $\Pr[\mathsf{ExpEUFDVS}_{\Pi,\mathcal{A}}(1^\lambda) = 1] \leq \mathsf{negl}(\lambda)$ *where* $\mathsf{ExpEUFIMDVS}$ *is defined as follows:*

$\mathsf{ExpEUFIMDVS}_{\Pi,\mathcal{A}}(1^\lambda)$

$L_{\mathsf{VPK}} := \emptyset; L_{\mathsf{SPK}} := \emptyset; L_{\mathsf{VSK}} := \emptyset; L_{\mathsf{SSK}} := \emptyset; L_{\mathsf{Sign}} := \emptyset; L_{\mathsf{Vrf}} := \emptyset;$
$(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Set}(1^\lambda);$
$(\mathsf{id}^*_{\mathsf{S}}, \mathcal{D}^*, \mathsf{m}^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{O_{SPK}},\mathsf{O_{SSK}},\mathsf{O_{VPK}},\mathsf{O_{VSK}},\mathsf{O_{Sig}},\mathsf{O_{Vrf}}}(\mathsf{pp})$:
output 1 if $(\exists \mathsf{id'_V} \in \mathcal{D}^* \setminus L_{\mathsf{VSK}}$ s.t. $\mathsf{Vrf}(\mathsf{pp}, \{\mathsf{vpk}_{\mathsf{id_V}}\}_{\mathsf{id_V}\in\mathcal{D}^*}, \mathsf{vsk}_{\mathsf{id'_V}}, \mathsf{spk}_{\mathsf{id}^*_{\mathsf{S}}}, m^*, \sigma^*) = 1)$
$\wedge (\mathsf{id}^*_{\mathsf{S}} \in L_{\mathsf{SPK}}) \wedge ((\mathsf{id}^*_{\mathsf{S}}, \mathsf{spk}_{\mathsf{id}^*_{\mathsf{S}}}, \mathsf{spk}_{\mathsf{id}^*_{\mathsf{S}}}) \notin L_{\mathsf{SSK}})$
$\wedge (\forall \mathsf{id'_V} \in \mathcal{D}^*, (\mathsf{id'_V}, \mathsf{vpk}_{\mathsf{id'_V}}, \mathsf{vsk}_{\mathsf{id'_V}}) \in L_{\mathsf{VPK}}) \wedge ((\mathcal{D}^*, \mathsf{id}^*_{\mathsf{S}}, \mathsf{m}^*) \notin L_{\mathsf{Sign}})$
otherwise 0

where $\mathsf{O_{SPK}}, \mathsf{O_{SSK}}, \mathsf{O_{VPK}}, \mathsf{O_{VSK}}, \mathsf{O_{Sig}}$*, and* $\mathsf{O_{Vrf}}$ *are defined as in Definition 7.*

Definition 12 (Signature Indistinguishability). *An IMDVS* $\Pi = (\mathsf{Set}, \mathsf{SKG}, \mathsf{VKG}, \mathsf{Sig}, \mathsf{Vrf})$ *is signature indistinguishable if for any security parameter* $\lambda \in \mathbb{N}$ *and any PPT adversary* $\mathcal{A}$*, it holds that* $|\Pr[\mathsf{ExpSigIND}_{\Pi,\mathcal{A}}(\lambda) = 1] - 1/2| \leq \mathsf{negl}(\lambda)$ *where the experiment* $\mathsf{ExpSigIND}_{\Pi,\mathcal{A}}(\lambda)$ *is defined as follows:*

$\mathsf{ExpSigIND}_{\Pi,\mathcal{A}}(\lambda)$

$L_{\mathsf{VPK}} := \emptyset; L_{\mathsf{SPK}} := \emptyset; L_{\mathsf{VSK}} := \emptyset; L_{\mathsf{SSK}} := \emptyset; L_{\mathsf{Sign}} := \emptyset; L_{\mathsf{Vrf}} := \emptyset;$
$(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Set}(1^\lambda);$
$(\mathsf{m}_0, \mathsf{m}_1, \mathsf{id_S} \in \mathcal{I}, \mathcal{D} \subseteq \mathcal{I}) \leftarrow \mathcal{A}^{\mathsf{O_{SPK}},\mathsf{O_{SSK}},\mathsf{O_{VPK}},\mathsf{O_{VSK}},\mathsf{O_{Sig}}}(\mathsf{pp});$
$b \leftarrow \{0,1\}; \sigma \leftarrow \mathsf{Sig}(\mathsf{pp}, \mathsf{ssk}_{\mathsf{id_S}}, \{\mathsf{vpk}_i\}_{i\in\mathcal{D}}, \mathsf{m}_b);$
$b' \leftarrow \mathcal{A}^{\mathsf{O_{SPK}},\mathsf{O_{SSK}},\mathsf{O_{VPK}},\mathsf{O_{VSK}},\mathsf{O_{Sig}}}(\sigma);$
return $\perp$ if $(\exists \mathsf{id_V} \in \mathcal{D}$ s.t. $\mathsf{id_V} \in L_{\mathsf{VSK}}) \vee (\exists \mathsf{id_V} \in \mathcal{D}$ s.t. $(\mathsf{id_V}, \cdot, \cdot) \notin L_{\mathsf{VPK}})$:
Output 1 if $b' = b$, otherwise 0

where $\mathsf{O_{SPK}}, \mathsf{O_{SSK}}, \mathsf{O_{VPK}}, \mathsf{O_{VSK}}, \mathsf{O_{Sig}}$*, and* $\mathsf{O_{Vrf}}$ *work as follows:*

$\mathsf{O_{SPK}}$**:** *Given* $\mathsf{id_S} \in \mathcal{I}$*, if* $\mathsf{id_S}$ *has already been queried previously, then it searches for* $(\mathsf{id_S}, \mathsf{spk}_{\mathsf{id_S}}, \mathsf{ssk}_{\mathsf{id_S}})$ *from* L_{SPK} *and returns* $\mathsf{spk}_{\mathsf{id_S}}$*. Otherwise, it computes* $(\mathsf{spk}_{\mathsf{id_S}}, \mathsf{ssk}_{\mathsf{id_S}}) \leftarrow \mathsf{SKG}(\mathsf{pp}, \mathsf{msk}, \mathsf{id_S})$*, returns* $\mathsf{spk}_{\mathsf{id_S}}$*, and updates* $L_{\mathsf{SPK}} := L_{\mathsf{SPK}} \cup \{(\mathsf{id_S}, \mathsf{spk}_{\mathsf{id_S}}, \mathsf{ssk}_{\mathsf{id_S}})\}$.

$\mathsf{O_{SSK}}$**:** *Given* $\mathsf{id_S} \in \mathcal{I}$*, if* $(\mathsf{id_S}, \mathsf{spk}_{\mathsf{id_S}}, \mathsf{ssk}_{\mathsf{id_S}}) \in L_{\mathsf{SPK}}$*, then it returns* $\mathsf{ssk}_{\mathsf{id_S}}$*, and updates* $L_{\mathsf{SSK}} := L_{\mathsf{SSK}} \cup \{\mathsf{id_S}\}$*. Otherwise, it calls* $\mathsf{O_{SPK}}(\mathsf{id_S})$ *to generate* $(\mathsf{spk}_{\mathsf{id_S}}, \mathsf{ssk}_{\mathsf{id_S}})$ *along with updating* $L_{\mathsf{SPK}} := L_{\mathsf{SPK}} \cup \{(\mathsf{id_S}, \mathsf{spk}_{\mathsf{id_S}}, \mathsf{ssk}_{\mathsf{id_S}})\}$*, returns* $(\mathsf{spk}_{\mathsf{id_S}}, \mathsf{ssk}_{\mathsf{id_S}})$*, and updates* $L_{\mathsf{SSK}} := L_{\mathsf{SSK}} \cup \{\mathsf{id_S}\}$*. Note that we regard the signer corresponding to* $\mathsf{id_S} \in L_{\mathsf{SSK}}$ *as a corrupted signer.*

$\mathsf{O}_{\mathsf{VPK}}$: *Given* $\mathsf{id}_{\mathsf{V}} \in \mathcal{I}$*, if* id_{V} *has already been queried previously, then it searches for* $(\mathsf{id}_{\mathsf{V}}, \mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}, \mathsf{vsk}_{\mathsf{id}_{\mathsf{V}}})$ *from* L_{VPK} *and returns* $\mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}$*. Otherwise, it computes* $(\mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}, \mathsf{vsk}_{\mathsf{id}_{\mathsf{V}}}) \leftarrow \mathsf{VKG}(\mathsf{pp}, \mathsf{msk}, \mathsf{id}_{\mathsf{V}})$*, returns* $\mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}$*, and updates* $L_{\mathsf{VPK}} := L_{\mathsf{VPK}} \cup \{(\mathsf{id}_{\mathsf{V}}, \mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}, \mathsf{vsk}_{\mathsf{id}_{\mathsf{V}}})\}$.

$\mathsf{O}_{\mathsf{VSK}}$: *Given* $\mathsf{id}_{\mathsf{V}} \in \mathcal{I}$*, if* $(\mathsf{id}_{\mathsf{V}}, \mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}, \mathsf{vsk}_{\mathsf{id}_{\mathsf{V}}}) \in L_{\mathsf{VPK}}$*, then it returns* $\mathsf{vsk}_{\mathsf{id}_{\mathsf{V}}}$*, and updates* $L_{\mathsf{VSK}} := L_{\mathsf{VSK}} \cup \{\mathsf{id}_{\mathsf{V}}\}$*. Otherwise, it calls* $\mathsf{O}_{\mathsf{VPK}}(\mathsf{id}_{\mathsf{V}})$ *to generate* $(\mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}, \mathsf{vsk}_{\mathsf{id}_{\mathsf{V}}})$ *along with* $L_{\mathsf{VPK}} := L_{\mathsf{VPK}} \cup \{(\mathsf{id}_{\mathsf{V}}, \mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}, \mathsf{vsk}_{\mathsf{id}_{\mathsf{V}}})\}$*, returns* $(\mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}, \mathsf{vsk}_{\mathsf{id}_{\mathsf{V}}})$*, and updates* $L_{\mathsf{VSK}} := L_{\mathsf{VSK}} \cup \{\mathsf{id}_{\mathsf{V}}\}$*. Note that we regard the verifier corresponding to* $\mathsf{id}_{\mathsf{V}} \in L_{\mathsf{VSK}}$ *as a corrupted verifier.*

$\mathsf{O}_{\mathsf{Sig}}$: *Given* $\mathcal{D} \subseteq \mathcal{I}, \mathsf{id}_{\mathsf{S}} \in \mathcal{I}$*, and* $\mathsf{m} \in \mathcal{M}$*, it does the followings:*

- *If* $(\mathsf{id}_{\mathsf{S}}, \cdot, \cdot) \notin L_{\mathsf{SPK}}$*, then call* $\mathsf{O}_{\mathsf{SPK}}$ *on* id_{S} *to generate* $(\mathsf{spk}_{\mathsf{id}_{\mathsf{S}}}, \mathsf{ssk}_{\mathsf{id}_{\mathsf{S}}})$.
- *For all* $\mathsf{id}_{\mathsf{V}} \in \mathcal{D}$ *s.t.* $(\mathsf{id}_{\mathsf{V}}, \cdot, \cdot) \notin L_{\mathsf{VPK}}$*, call* $\mathsf{O}_{\mathsf{VPK}}$ *on* id_{V} *to generate* $(\mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}, \mathsf{vsk}_{\mathsf{id}_{\mathsf{V}}})$.
- *Return* $\sigma \leftarrow \mathsf{Sig}(\mathsf{pp}, \mathsf{ssk}_{\mathsf{id}_{\mathsf{S}}}, \{\mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}\}_{\mathsf{id}_{\mathsf{V}} \in \mathcal{D}}, \mathsf{m})$*, and update* $L_{\mathsf{Sign}} := L_{\mathsf{Sign}} \cup \{(\mathcal{D}, \mathsf{id}_{\mathsf{S}}, \mathsf{m})\}$.

On the Absence of a Verification Oracle. In the experiment $\mathsf{ExpSigIND}$, the adversary is not given a verification oracle. In fact, we could not construct IMDVS from PKE if we give the oracle to the adversary. We will elaborate on this point after we demonstrate the construction of IMDVS from PKE. However, we here emphasize that IMDVS is an artificial primitive, and thus such a restricted definition is not important if we can prove the separation.

4 Equivalence Between PKE and IMDVS

In this section, we prove that PKE and IMDVS are equivalent to each other. To do so, we demonstrate two generic constructions; the construction of PKE from IMDVS, and vice versa. Formally, we prove the following theorem in this section.

Theorem 1. *PKE and IMDVS are equivalent to each other.*

Proof. Theorem 2 and Theorem 3 conclude Theorem 1. □

4.1 A Generic Construction of PKE from IMDVS

Let $\Pi_I = (\mathsf{Set}, \mathsf{SKG}, \mathsf{VKG}, \mathsf{Sig}, \mathsf{Vrf})$ be an IMDVS scheme. We demonstrate a generic construction of IND-CPA secure PKE $\Pi_P = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ with a message space $\mathcal{M} = \{0, 1\}$ from $\Pi_I = (\mathsf{Set}, \mathsf{SKG}, \mathsf{VKG}, \mathsf{Sig}, \mathsf{Vrf})$. The construction is similar to the construction of PKE from a designated confirmer signature scheme by Okamoto [24]. Before the formal description, we provide an intuition for the construction.

In the encryption algorithm Enc, it chooses a message m uniformly at random and signs on m by Sig to obtain σ. In case the message to be encrypted is 1, it outputs m and σ as a ciphertext, otherwise, it chooses another message m' uniformly at random and outputs m' and σ. Given (m, σ), the decryption

algorithm outputs the verification result on (m, σ). That is, if (m, σ) is a valid pair, then it passes the verification so that the decryption result is 1, otherwise 0.

Here we should guarantee that, with overwhelming probability, Vrf rejects a pair (m', σ) where m' is a uniformly chosen message and σ is a signature on another uniformly chosen message m. It is almost trivial that this is the case. Concretely, if Vrf accepts such a pair with non-negligible probability, the adversary in $\mathsf{ExpEUFIMDVS}$ easily wins the game; it signs on a message and outputs the signature along with another uniformly chosen message.

The formal construction is as follows:

$\mathsf{KeyGen}(1^\lambda)$**:** Given 1^λ, it first runs $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Set}(1^\lambda)$. Then, it chooses $\mathsf{id} \in \mathcal{I}$ uniformly, and computes $(\mathsf{spk}_{\mathsf{id}}, \mathsf{ssk}_{\mathsf{id}}) \leftarrow \mathsf{SKG}(\mathsf{pp}, \mathsf{msk}, \mathsf{id})$ and $(\mathsf{vpk}_{\mathsf{id}}, \mathsf{vsk}_{\mathsf{id}}) \leftarrow \mathsf{VKG}(\mathsf{pp}, \mathsf{msk}, \mathsf{id})$. It outputs $\mathsf{pk} := (\mathsf{pp}, \mathsf{ssk}_{\mathsf{id}}, \mathsf{vpk}_{\mathsf{id}})$ and $\mathsf{sk} := (\mathsf{pp}, \mathsf{spk}_{\mathsf{id}}, \mathsf{vsk}_{\mathsf{id}}, \mathsf{vpk}_{\mathsf{id}})$.

$\mathsf{Enc}(\mathsf{pk}, b)$**:** Given $\mathsf{pk} = (\mathsf{pp}, \mathsf{ssk}_{\mathsf{id}}, \mathsf{vpk}_{\mathsf{id}})$ and a message $b \in \{0, 1\}$, it chooses $\mathsf{m} \leftarrow \mathcal{M}$, and computes $\sigma \leftarrow \mathsf{Sig}(\mathsf{pp}, \mathsf{ssk}_{\mathsf{id}}, \mathsf{vpk}_{\mathsf{id}}, \mathsf{m})$. When $b = 1$, it outputs $c := (\mathsf{m}, \sigma)$. Otherwise, it chooses $\mathsf{m}' \leftarrow \mathcal{M}$, and outputs $c := (\mathsf{m}', \sigma)$.

$\mathsf{Dec}(\mathsf{sk}, c)$**:** Given $\mathsf{sk} = (\mathsf{pp}, \mathsf{spk}_{\mathsf{id}}, \mathsf{vsk}_{\mathsf{id}}, \mathsf{vpk}_{\mathsf{id}})$ and $c = (\mathsf{m}, \sigma)$, it outputs $b = \mathsf{Vrf}(\mathsf{pp}, \mathsf{vpk}_{\mathsf{id}}, \mathsf{vsk}_{\mathsf{id}}, \mathsf{spk}_{\mathsf{id}}, \mathsf{m}, \sigma)$.

Correctness of Π_P immediately follows from the correctness and EUF-CMA security of Π_I. That is, when c is a ciphertext of $b = 1$, it holds that $\mathsf{Dec}(\mathsf{sk}, c) = 1$ with probability 1 due to the correctness of Π_I. On the other hand, when $b = 0$, $\mathsf{Dec}(\mathsf{sk}, c) = 0$ with overwhelming probability, since otherwise we can construct a PPT adversary that breaks EUF-CMA security of Π_I with overwhelming probability.

Theorem 2. *If Π_I is signature indistinguishable, then Π_P is IND-CPA secure.*

Proof. We assume for contradiction that there exists a PPT adversary $\mathcal{A}$ that breaks the IND-CPA security of Π_P with non-negligible advantage ϵ. Then, we construct a PPT adversary $\mathcal{B}$ that breaks the signature indistinguishability of Π_I as follows.

Setup Phase Given pp, $\mathcal{B}$ first chooses $\mathsf{id} \in \mathcal{I}$ uniformly. It makes queries to $\mathsf{O}_{\mathsf{SSK}}$ and $\mathsf{O}_{\mathsf{VPK}}$ on id to obtain $(\mathsf{spk}_{\mathsf{id}}, \mathsf{ssk}_{\mathsf{id}})$ and $\mathsf{vpk}_{\mathsf{id}}$, respectively. It sets $\mathsf{pk} := (\mathsf{pp}, \mathsf{ssk}_{\mathsf{id}}, \mathsf{vpk}_{\mathsf{id}})$ and simulates $\mathcal{A}$ on pk. Finally, it returns $\mathsf{m}_0 = 0, \mathsf{m}_1 = 1, \mathsf{id}$, and $\mathcal{D} = \{\mathsf{vpk}_{\mathsf{id}}\}$ to the challenger of $\mathsf{ExpSigIND}_{\Pi_I, \mathcal{B}}(\lambda)$.

Guessing Phase Given a signature σ on either m_0 or m_1, say m_b, $\mathcal{B}$ chooses $b^\dagger \in \{0, 1\}$ uniformly, and simulates $\mathcal{A}$ on $(\mathsf{m}_{b^\dagger}, \sigma)$. Note that the probability that $(\mathsf{m}_{b^\dagger}, \sigma)$ is a valid pair is only $1/2$, but it is sufficient for our analysis. When $\mathcal{A}$ outputs a bit b', it returns b' to the challenger of $\mathsf{ExpSigIND}_{\Pi_I, \mathcal{B}}(\lambda)$.

Analysis Observe that $\mathcal{B}$ does not violate the conditions to abort the experiment. Therefore, given b' the challenger outputs 1 if $b' = b$, otherwise 0.

Recall that $\mathcal{B}$ runs $\mathcal{A}$ on a valid pair with probability 1/2. Therefore, the advantage of $\mathcal{B}$ that guesses the challenge bit b correctly is $\epsilon/2$, which is still non-negligible. □

4.2 A Generic Construction of IMDVS from PKE

We demonstrate a generic construction Π_I of IMDVS from a PKE scheme and standard signature scheme. Note that it is known that a standard signature scheme can be constructed from a PKE scheme. Let $\Pi_P = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE and $\Pi_S = (\mathsf{KG}, \mathsf{Sig}, \mathsf{Vrf})$ a signature scheme. Without loss of generality, we assume that the plaintext space of Π_P is equal to the signature space of Π_S (i.e. a signature created by $\Pi_S.\mathsf{Sig}$ can be encrypted by $\Pi_P.\mathsf{Enc}$).

Before the formal description, we provide an intuition of the construction. The construction does not require a setup algorithm. The signer's (resp., the verifier's) key generation algorithm outputs a key pair of the standard signature scheme (resp., the PKE scheme).

In the signing algorithm, it signs on a message by using the signer's signing key. Then, for each designated verifier, the signature is encrypted by using the designated verifier's public key. Thus, if n verifiers are designated, then the signature is a tuple of n ciphertexts of the underlying PKE scheme.

In the verification algorithm, a designated verifier first decrypts a ciphertext that corresponds to him by using his verification key (decryption key of the underlying PKE scheme) to obtain a standard signature. Finally, it verifies the signature by using the signer's public key.

Roughly, EUF-CMA security of Π_I is guaranteed by EUF-CMA security of Π_S, and signature indistinguishability stems from IND-CPA security of Π_P.

$\mathsf{SKG}(\mathsf{id}_\mathsf{S})$**:** Given $\mathsf{id}_\mathsf{S} \in \mathcal{I}$, it computes $(\mathsf{pk}_{sig,\mathsf{id}_\mathsf{S}}, \mathsf{sk}_{sig,\mathsf{id}_\mathsf{S}}) \leftarrow \Pi_S.\mathsf{KG}(1^\lambda)$. It outputs $\mathsf{spk}_{\mathsf{id}_\mathsf{S}} := (\mathsf{id}_\mathsf{S}, \mathsf{pk}_{sig,\mathsf{id}_\mathsf{S}})$ and $\mathsf{ssk}_{\mathsf{id}_\mathsf{S}} := (\mathsf{id}_\mathsf{S}, \mathsf{sk}_{sig,\mathsf{id}_\mathsf{S}})$.

$\mathsf{VKG}(\mathsf{id}_\mathsf{V})$**:** Given $\mathsf{id}_\mathsf{V} \in \mathcal{I}$, it computes $(\mathsf{pk}_{pke,\mathsf{id}_\mathsf{V}}, \mathsf{sk}_{pke,\mathsf{id}_\mathsf{V}}) \leftarrow \Pi_P.\mathsf{KeyGen}(1^\lambda)$. It outputs $\mathsf{vpk}_{\mathsf{id}_\mathsf{V}} := (\mathsf{id}_\mathsf{V}, \mathsf{pk}_{pke,\mathsf{id}_\mathsf{V}})$ and $\mathsf{vsk}_{\mathsf{id}_\mathsf{V}} := (\mathsf{id}_\mathsf{V}, \mathsf{sk}_{pke,\mathsf{id}_\mathsf{V}})$.

$\mathsf{Sig}(\mathsf{ssk}_{\mathsf{id}_\mathsf{S}}, \{\mathsf{vpk}_{\mathsf{id}_\mathsf{V}}\}_{\mathsf{id}_\mathsf{V} \in \mathcal{D}}, \mathsf{m})$**:** Given $\mathsf{ssk}_{\mathsf{id}_\mathsf{S}} = (\mathsf{id}_\mathsf{S}, \mathsf{sk}_{sig,\mathsf{id}_\mathsf{S}})$, $\{\mathsf{vpk}_{\mathsf{id}_\mathsf{V}}\}_{\mathsf{id}_\mathsf{V} \in \mathcal{D}}$ where $\mathsf{vpk}_{\mathsf{id}_\mathsf{V}} = (\mathsf{id}_\mathsf{V}, \mathsf{pk}_{pke,\mathsf{id}_\mathsf{V}})$, and a message m, it does the followings: It creates $\sigma_{\mathsf{id}_\mathsf{S}} \leftarrow \Pi_S.\mathsf{Sig}(\mathsf{sk}_{sig,\mathsf{id}_\mathsf{S}}, \mathsf{m})$ and for each $\mathsf{id}_\mathsf{V} \in \mathcal{D}$, it computes $c_{\mathsf{id}_\mathsf{V}} \leftarrow \Pi_P.\mathsf{Enc}(\mathsf{pk}_{pke,\mathsf{id}_\mathsf{V}}, \sigma_{\mathsf{id}_\mathsf{S}})$. It outputs $\sigma := \{(\mathsf{id}_\mathsf{V}, c_{\mathsf{id}_\mathsf{V}})\}_{\mathsf{id}_\mathsf{V} \in \mathcal{D}}$.

$\mathsf{Vrf}(\{\mathsf{vpk}_{\mathsf{id}_\mathsf{V}}\}_{\mathsf{id}_\mathsf{V} \in \mathcal{D}}, \mathsf{vsk}_{\mathsf{id}'_\mathsf{V}}, \mathsf{spk}_{\mathsf{id}_\mathsf{S}}, \mathsf{m}, \sigma)$**:** Given $\{\mathsf{vpk}_{\mathsf{id}_\mathsf{V}}\}_{\mathsf{id}_\mathsf{V} \in \mathcal{D}}$ where $\mathsf{vpk}_{\mathsf{id}_\mathsf{V}} = (\mathsf{id}_\mathsf{V}, \mathsf{pk}_{pke,\mathsf{id}_\mathsf{V}})$, $\mathsf{vsk}_{\mathsf{id}'_\mathsf{V}} = (\mathsf{id}'_\mathsf{V}, \mathsf{sk}_{pke,\mathsf{id}'_\mathsf{V}})$, $\mathsf{spk}_{\mathsf{id}_\mathsf{S}}$ where $\mathsf{spk}_{\mathsf{id}_\mathsf{S}} = (\mathsf{id}_\mathsf{S}, \mathsf{pk}_{sig,\mathsf{id}_\mathsf{S}})$, m, and σ where $\sigma = \{(\mathsf{id}_\mathsf{V}, c_{\mathsf{id}_\mathsf{V}})\}_{\mathsf{id}_\mathsf{V} \in \mathcal{D}'}$, it outputs $\perp$ if $\mathcal{D} \neq \mathcal{D}'$ or $\mathsf{id}'_\mathsf{V} \notin \mathcal{D}$. Otherwise, it computes $\sigma_{\mathsf{id}_\mathsf{S}} \leftarrow \Pi_P.\mathsf{Dec}(\mathsf{sk}_{pke,\mathsf{id}'_\mathsf{V}}, c_{\mathsf{id}'_\mathsf{V}})$ and outputs $b = \Pi_S.\mathsf{Vrf}(\mathsf{pk}_{sig,\mathsf{id}_\mathsf{S}}, \mathsf{m}, \sigma_{\mathsf{id}_\mathsf{S}})$.

Theorem 3. *The construction Π_I is EUF-CMA secure if Π_S is EUF-CMA secure.*

Proof. We assume for contradiction that there exists a PPT adversary $\mathcal{A}$ that breaks EUF-CMA security of Π_I with non-negligible probability ϵ. Then, we demonstrate a PPT adversary $\mathcal{B}$ that breaks EUF-CMA security of Π_S with non-negligible probability, using $\mathcal{A}$. We assume, without loss of generality, that $\mathcal{A}$ makes at most $q = \mathsf{poly}(\lambda)$ queries to the oracle $\mathsf{O}_{\mathsf{SPK}}$.

Observe that a signer's key pair of Π_I is a key pair of Π_S, and $\Pi_I.\mathsf{Vrf}$ uses $\Pi_S.\mathsf{Vrf}$ inside. Thus, in case $\mathcal{A}$ succeeds in forging a signature, it means that $\mathcal{A}$ outputs a ciphertext of a standard signature, and the standard signature passes the verification by $\Pi_S.\mathsf{Vrf}$, without knowing the signing key of a signer. The algorithm $\mathcal{B}$ works as follows.

Setup Phase Given pk^*_{sig}, it initiates $L_{\mathsf{VPK}} := \emptyset, L_{\mathsf{SPK}} := \emptyset, L_{\mathsf{VSK}} := \emptyset, L_{\mathsf{SSK}} := \emptyset, L_{\mathsf{Sign}} := \emptyset$ and $L_{\mathsf{Vrf}} := \emptyset$.

Forgery Phase It runs the adversary $\mathcal{A}$ with simulating oracles as follows:

$\mathsf{O}_{\mathsf{SPK}}$: Given $\mathsf{id}_{\mathsf{S}} \in \mathcal{I}$, if id_{S} has already been queried previously, then it searches for $(\mathsf{id}_{\mathsf{S}}, \mathsf{spk}_{\mathsf{id}_{\mathsf{S}}}, \mathsf{ssk}_{\mathsf{id}_{\mathsf{S}}})$ from L_{SPK} and returns $\mathsf{spk}_{\mathsf{id}_{\mathsf{S}}}$. Otherwise, it does the following:
- (This is a once for all task.) With probability $1/q$, it sets $\mathsf{spk}_{\mathsf{id}_{\mathsf{S}}} := \mathsf{pk}^*_{sig}$, returns $\mathsf{spk}_{\mathsf{id}_{\mathsf{S}}}$ and updates $L_{\mathsf{SPK}} := L_{\mathsf{SPK}} \cup \{(\mathsf{id}_{\mathsf{S}}, \mathsf{vpk}_{\mathsf{id}_{\mathsf{S}}}, \bot)\}$.
- Otherwise, it computes $(\mathsf{spk}_{\mathsf{id}_{\mathsf{S}}}, \mathsf{ssk}_{\mathsf{id}_{\mathsf{S}}}) := (\mathsf{pk}_{sig,\mathsf{id}_{\mathsf{S}}}, \mathsf{sk}_{sig,\mathsf{id}_{\mathsf{S}}}) \leftarrow \Pi_S.\mathsf{KG}(1^\lambda)$, returns $\mathsf{spk}_{\mathsf{id}_{\mathsf{S}}}$ to $\mathcal{A}$, and updates $L_{\mathsf{SPK}} := L_{\mathsf{SPK}} \cup \{(\mathsf{id}_{\mathsf{S}}, \mathsf{spk}_{\mathsf{id}_{\mathsf{S}}}, \mathsf{ssk}_{\mathsf{id}_{\mathsf{S}}})\}$.

$\mathsf{O}_{\mathsf{SSK}}$: Given $\mathsf{id}_{\mathsf{S}} \in \mathcal{I}$, it does the following. If id_{S} is an identifier such that $\mathsf{spk}_{\mathsf{id}_{\mathsf{S}}} = \mathsf{pk}^*_{sig}$, then abort the experiment and output $\bot$. If $(\mathsf{id}_{\mathsf{S}}, \mathsf{spk}_{\mathsf{id}_{\mathsf{S}}}, \mathsf{ssk}_{\mathsf{id}_{\mathsf{S}}}) \in L_{\mathsf{SPK}}$, then it returns $\mathsf{ssk}_{\mathsf{id}_{\mathsf{S}}}$, and updates $L_{\mathsf{SSK}} := L_{\mathsf{SSK}} \cup \{\mathsf{id}_{\mathsf{S}}\}$. Otherwise, it does the following.
- (This does not happen if $\mathsf{spk}_{\mathsf{id}_{\mathsf{S}}} := \mathsf{pk}^*_{sig}$ has already occurred in the simulation of $\mathsf{O}_{\mathsf{SPK}}$.) With probability $1/q$, it aborts the experiment. We regard this event as a corruption query on pk^*_{sig} is made.
- Otherwise, it computes $(\mathsf{pk}_{sig,\mathsf{id}_{\mathsf{S}}}, \mathsf{sk}_{sig,\mathsf{id}_{\mathsf{S}}}) \leftarrow \Pi_S.\mathsf{KG}(1^\lambda)$, sets $L_{\mathsf{SPK}} := L_{\mathsf{SPK}} \cup \{(\mathsf{id}_{\mathsf{S}}, \mathsf{spk}_{\mathsf{id}_{\mathsf{S}}}, \mathsf{ssk}_{\mathsf{id}_{\mathsf{S}}})\}$, returns $(\mathsf{spk}_{\mathsf{id}_{\mathsf{S}}}, \mathsf{ssk}_{\mathsf{id}_{\mathsf{S}}})$ to $\mathcal{A}$, and updates $L_{\mathsf{SSK}} := L_{\mathsf{SSK}} \cup \{\mathsf{id}_{\mathsf{S}}\}$.

$\mathsf{O}_{\mathsf{VPK}}$: Given $\mathsf{id}_{\mathsf{V}} \in \mathcal{I}$, if id_{V} has already been queried previously, then it searches for $(\mathsf{id}_{\mathsf{V}}, \mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}, \mathsf{vsk}_{\mathsf{id}_{\mathsf{V}}})$ from L_{VPK} and returns $\mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}$ to $\mathcal{A}$. Otherwise, it computes $(\mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}, \mathsf{vsk}_{\mathsf{id}_{\mathsf{V}}}) := (\mathsf{pk}_{pke,\mathsf{id}_{\mathsf{V}}}, \mathsf{sk}_{pke,\mathsf{id}_{\mathsf{V}}}) \leftarrow \Pi_P.\mathsf{KG}(1^\lambda)$, returns $\mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}$ to $\mathcal{A}$, and updates $L_{\mathsf{VPK}} := L_{\mathsf{VPK}} \cup \{(\mathsf{id}_{\mathsf{V}}, \mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}, \mathsf{vsk}_{\mathsf{id}_{\mathsf{V}}})\}$.

$\mathsf{O}_{\mathsf{VSK}}$: Given $\mathsf{id}_{\mathsf{V}} \in \mathcal{I}$, if $(\mathsf{id}_{\mathsf{V}}, \mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}, \mathsf{vsk}_{\mathsf{id}_{\mathsf{V}}}) \in L_{\mathsf{VPK}}$, then it returns $\mathsf{vsk}_{\mathsf{id}_{\mathsf{V}}}$, and updates $L_{\mathsf{VSK}} := L_{\mathsf{VSK}} \cup \{\mathsf{id}_{\mathsf{V}}\}$. Otherwise, it generates $(\mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}, \mathsf{vsk}_{\mathsf{id}_{\mathsf{V}}}) := (\mathsf{pk}_{pke,\mathsf{id}_{\mathsf{V}}}, \mathsf{sk}_{pke,\mathsf{id}_{\mathsf{V}}}) \leftarrow \Pi_P.\mathsf{KG}(1^\lambda)$ along with updating $L_{\mathsf{VPK}} := L_{\mathsf{VPK}} \cup \{(\mathsf{id}_{\mathsf{V}}, \mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}, \mathsf{vsk}_{\mathsf{id}_{\mathsf{V}}})\}$, returns $(\mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}, \mathsf{vsk}_{\mathsf{id}_{\mathsf{V}}})$, and updates $L_{\mathsf{VSK}} := L_{\mathsf{VSK}} \cup \{\mathsf{id}_{\mathsf{S}}\}$.

$\mathsf{O}_{\mathsf{Sig}}$: Given $\mathcal{D} \subseteq \mathcal{I}, \mathsf{id}_{\mathsf{S}} \in \mathcal{I}$, and $\mathsf{m} \in \mathcal{M}$, it does the followings:
- If $(\mathsf{id}_{\mathsf{S}}, \cdot, \cdot) \notin L_{\mathsf{SPK}}$, then it simulates $\mathsf{O}_{\mathsf{SPK}}$ on id_{S} to generate $(\mathsf{id}_{\mathsf{S}}, \mathsf{spk}_{\mathsf{id}_{\mathsf{S}}}, \mathsf{ssk}_{\mathsf{id}_{\mathsf{S}}})$.

- For all $\mathsf{id_V} \in \mathcal{D}$ s.t. $(\mathsf{id_V}, \cdot, \cdot) \notin L_{\mathsf{VPK}}$, it simulates $\mathsf{O_{VPK}}$ on $\mathsf{id_V}$ to generate $(\mathsf{id_V}, \mathsf{vpk}_{\mathsf{id_V}}, \mathsf{vsk}_{\mathsf{id_V}})$.
- If $\mathsf{id_S}$ is an identity such that $\mathsf{spk}_{\mathsf{id_S}} = \mathsf{pk}^*_{sig}$, then it calls $\mathsf{O_{Sig}}$ of $\mathsf{ExpEUFSig}$ on m to obtain a signature $\sigma_{\mathsf{id_S}}$, otherwise it creates $\sigma_{\mathsf{id_S}} \leftarrow \Pi_S.\mathsf{Sig}(\mathsf{ssk}_{\mathsf{id_S}}, \mathsf{m})$. For each $\mathsf{id_V} \in \mathcal{D}$, it computes $c_{\mathsf{id_V}} \leftarrow \Pi_P.\mathsf{Enc}(\mathsf{vpk}_{\mathsf{id_V}}, \sigma_{\mathsf{id_S}})$. It returns $\sigma = \{(\mathsf{id_V}, c_{\mathsf{id_V}})\}_{\mathsf{id_V} \in \mathcal{D}}$, and updates $L_{\mathsf{Sign}} \coloneqq L_{\mathsf{Sign}} \cup \{(\mathcal{D}, \mathsf{id_S}, \mathsf{m})\}$.

$\mathsf{O_{Vrf}}$**:** Given $\mathsf{id}'_{\mathsf{V}}, \mathsf{id_S} \in \mathcal{I}, \mathsf{m} \in \mathcal{M}$, $\mathcal{D} \subseteq \mathcal{I}$ where $\mathsf{id}'_{\mathsf{V}} \in \mathcal{D}$, and σ, it does the followings:

- If $\mathsf{id}'_{\mathsf{V}} \notin \mathcal{D}$, then return 0.
- If $(\mathsf{id_S}, \cdot, \cdot) \notin L_{\mathsf{SPK}}$, then it simulates $\mathsf{O_{SPK}}$ on $\mathsf{id_S}$ to generate $(\mathsf{id_S}, \mathsf{spk}_{\mathsf{id_S}}, \mathsf{ssk}_{\mathsf{id_S}})$.
- For all $\mathsf{id_V} \in \mathcal{D}$, if $(\mathsf{id_V}, \cdot, \cdot) \notin L_{\mathsf{VPK}}$, it simulates $\mathsf{O_{VPK}}$ on $\mathsf{id_V}$ to generate $(\mathsf{id_V}, \mathsf{vpk}_{\mathsf{id_V}}, \mathsf{vsk}_{\mathsf{id_V}})$.
- Parse $\sigma = \{(\mathsf{id_V}, c_{\mathsf{id_V}})\}_{\mathsf{id_V} \in \mathcal{D}}$. It computes $\sigma_{\mathsf{id}'_{\mathsf{V}}} = \Pi_P.\mathsf{Dec}(\mathsf{vsk}_{\mathsf{id}'_{\mathsf{V}}}, c_{\mathsf{id}'_{\mathsf{V}}})$, returns $b = \Pi_S.\mathsf{Vrf}(\mathsf{spk}_{\mathsf{id_S}}, \mathsf{m}, \sigma_{\mathsf{id}'_{\mathsf{V}}})$, and updates $L_{\mathsf{Vrf}} \coloneqq L_{\mathsf{Vrf}} \cup \{(\mathcal{D}, \mathsf{id}'_{\mathsf{V}}, \mathsf{id_S}, \mathsf{m}, \sigma)\}$.

When $\mathcal{A}$ outputs $(\mathsf{id}^*_{\mathsf{S}}, \mathcal{D}^*, \mathsf{m}^*, \sigma^*)$, $\mathcal{B}$ does the followings. Let $\sigma^* = \{(\mathsf{id_V}, c_{\mathsf{id_V}})\}_{\mathsf{id_V} \in \mathcal{D}^*}$. It aborts if (i) pk^*_{sig} is not embedded in a signer's public key, or (ii) $\mathsf{id}^*_{\mathsf{S}}$ is not the identifier whose public key is pk^*_{pke}. Otherwise, for all $\mathsf{id_V} \in \mathcal{D} \setminus L_{\mathsf{VSK}}$, it computes $\sigma_{\mathsf{id_V}} = \Pi_P.\mathsf{Dec}(\mathsf{vpk}_{\mathsf{id_V}}, c_{\mathsf{id_V}})$, and confirms if $\Pi_S.\mathsf{Vrf}(\mathsf{vpk}_{\mathsf{id_V}}, m^*, \sigma_{\mathsf{id_V}}) = 1$. If such $\mathsf{id_V}$ is found, then $\mathcal{B}$ returns $\sigma_{\mathsf{id_V}}$ to the challenger. Otherwise, it aborts. (Observe that when $\mathcal{A}$ wins, there must be such $\mathsf{id_V}$.)

Analysis We first observe that if $\mathcal{A}$ wins and $\mathcal{B}$ does not abort, then $\mathcal{B}$ wins as well. Thus, we evaluate the probability that $\mathcal{B}$ does not abort. The algorithm $\mathcal{B}$ aborts if one of the following events happens:

E_1: pk^*_{sig} is not embedded in a signer's public key.
E_2: An identity $\mathsf{id_S}$ s.t. $\mathsf{spk}_{\mathsf{id_S}} = \mathsf{pk}^*_{sig}$ is queried to $\mathsf{O_{SSK}}$.
E_3: $\mathsf{id}^*_{\mathsf{S}}$ is not the identifier whose public key is pk^*_{pke}.

We can evaluate the probability that each event does not occur as follows.

Regarding the event E_1, when $\mathcal{B}$ simulates $\mathsf{O_{SPK}}$ on $\mathsf{id_S}$, the probability that pk^*_{sig} is *not* embedded to the signer's public key is $1 - 1/q$. Since at most q queries are made during the experiment, the probability that E_1 happens is $(1 - 1/q)^q \approx 1/e$ for sufficiently large q. Thus, the probability that E_1 does not occur is $1 - 1/e$.

Regarding the event E_2, observe that there should be at least one non-corrupted signer's public key for the $\mathcal{A}$'s winning condition on $\mathsf{ExpEUFIMDVS}$. In other words, at least one identity should not be queried to $\mathsf{O_{SSK}}$. As an identity $\mathsf{id_S}$ s.t. $\mathsf{spk}_{\mathsf{id_S}} = \mathsf{pk}^*_{pke}$ is chosen uniformly at random, and at most q queries are made, the event E_2 does not occur with probability better than $1/q$.

Regarding the event E_3, similar to the above discussion, at most q signing keys are created during the experiment. Thus, the probability that the identifier

whose public key is pk^*_{pke} is chosen as $\mathsf{id}^*_{\mathsf{S}}$ (that is, the event E_3 does not occur) is at least $1/q$.

Recall that the winning probability of $\mathcal{A}$ is non-negligible ϵ, and $\mathcal{B}$ wins if $\mathcal{A}$ wins. To sum up the discussion, the advantage of $\mathcal{B}$ in the experiment $\mathsf{ExpINDCPA}$ is at least

$$\epsilon \cdot \left(1 - \frac{1}{e}\right) \cdot \frac{1}{q^2},$$

which is still non-negligible. □

Theorem 4. *The construction Π_I is signature indistinguishable if Π_P is IND-CPA secure.*

Proof. (The proof is similar to that of Theorem 3.) We demonstrate a PPT adversary $\mathcal{B}$ that breaks IND-CPA security of Π_P with non-negligible advantage with assuming the existence of a PPT adversary $\mathcal{A}$ that breaks signature indistinguishability of Π_I with non-negligible advantage ϵ for contradiction. Namely, $\mathcal{B}$ plays the experiment $\mathsf{ExpINDCPA}$ by running $\mathcal{A}$. The adversary $\mathcal{B}$ also simulates oracle answers that appear in the experiment $\mathsf{ExpSigIND}$ for $\mathcal{A}$. Without loss of generality, we assume that $\mathcal{A}$ obtains $q = \mathsf{poly}(\lambda)$ verifier's public keys during $\mathsf{ExpSigIND}$.

Setup Phase Given pk^*_{pke}, $\mathcal{B}$ sets $L_{\mathsf{VPK}} := \emptyset, L_{\mathsf{SPK}} := \emptyset, L_{\mathsf{VSK}} := \emptyset, L_{\mathsf{SSK}} := \emptyset, L_{\mathsf{Sign}} := \emptyset$, and $L_{\mathsf{Vrf}} := \emptyset$. It runs $\mathcal{A}$ with simulating oracles as follows (note that Π_I does not have a setup algorithm):

$\mathsf{O}_{\mathsf{SPK}}$**:** Given $\mathsf{id}_{\mathsf{S}} \in \mathcal{I}$, if id_{S} has already been queried previously, then it searches for $(\mathsf{id}_{\mathsf{S}}, \mathsf{spk}_{\mathsf{id}_{\mathsf{S}}}, \mathsf{ssk}_{\mathsf{id}_{\mathsf{S}}})$ from L_{SPK} and returns $\mathsf{spk}_{\mathsf{id}_{\mathsf{S}}}$. Otherwise, it computes $(\mathsf{spk}_{\mathsf{id}_{\mathsf{S}}}, \mathsf{ssk}_{\mathsf{id}_{\mathsf{S}}}) := (\mathsf{pk}_{sig,\mathsf{id}_{\mathsf{S}}}, \mathsf{sk}_{sig,\mathsf{id}_{\mathsf{S}}}) \leftarrow \Pi_S.\mathsf{KG}(1^\lambda)$, returns $\mathsf{spk}_{\mathsf{id}_{\mathsf{S}}}$, and updates $L_{\mathsf{SPK}} := L_{\mathsf{SPK}} \cup \{(\mathsf{id}_{\mathsf{S}}, \mathsf{spk}_{\mathsf{id}_{\mathsf{S}}}, \mathsf{ssk}_{\mathsf{id}_{\mathsf{S}}})\}$.

$\mathsf{O}_{\mathsf{SSK}}$**:** Given $\mathsf{id}_{\mathsf{S}} \in \mathcal{I}$, if $(\mathsf{id}_{\mathsf{S}}, \mathsf{spk}_{\mathsf{id}_{\mathsf{S}}}, \mathsf{ssk}_{\mathsf{id}_{\mathsf{S}}}) \in L_{\mathsf{SPK}}$, then it returns $\mathsf{ssk}_{\mathsf{id}_{\mathsf{S}}}$, and updates $L_{\mathsf{SSK}} := L_{\mathsf{SSK}} \cup \{\mathsf{id}_{\mathsf{S}}\}$. Otherwise, it generates $(\mathsf{spk}_{\mathsf{id}_{\mathsf{S}}}, \mathsf{ssk}_{\mathsf{id}_{\mathsf{S}}}) := (\mathsf{spk}_{\mathsf{id}_{\mathsf{S}}}, \mathsf{ssk}_{\mathsf{id}_{\mathsf{S}}}) \leftarrow \Pi_S.\mathsf{KG}(1^\lambda)$ along with updating $L_{\mathsf{SPK}} := L_{\mathsf{SPK}} \cup \{(\mathsf{id}_{\mathsf{S}}, \mathsf{spk}_{\mathsf{id}_{\mathsf{S}}}, \mathsf{ssk}_{\mathsf{id}_{\mathsf{S}}})\}$, returns $(\mathsf{spk}_{\mathsf{id}_{\mathsf{S}}}, \mathsf{ssk}_{\mathsf{id}_{\mathsf{S}}})$, and updates $L_{\mathsf{SSK}} := L_{\mathsf{SSK}} \cup \{\mathsf{id}_{\mathsf{S}}\}$.

$\mathsf{O}_{\mathsf{VPK}}$**:** Given $\mathsf{id}_{\mathsf{V}} \in \mathcal{I}$, if id_{V} has already been queried previously, then it searches for $(\mathsf{id}_{\mathsf{V}}, \mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}, \mathsf{vsk}_{\mathsf{id}_{\mathsf{V}}})$ from L_{VPK} and returns $\mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}$. Otherwise, it does the following:

- (This is a once for all task.) With probability $1/q$, it sets $\mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}} := \mathsf{pk}^*_{pke}$, returns $\mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}$, and updates $L_{\mathsf{VPK}} := L_{\mathsf{VPK}} \cup \{(\mathsf{id}_{\mathsf{V}}, \mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}, \bot)\}$.
- Otherwise, it computes $(\mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}, \mathsf{vsk}_{\mathsf{id}_{\mathsf{V}}}) := (\mathsf{pk}_{pke,\mathsf{id}_{\mathsf{V}}}, \mathsf{sk}_{pke,\mathsf{id}_{\mathsf{V}}}) \leftarrow \Pi_P.\mathsf{KeyGen}(1^\lambda)$, returns $\mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}$, and updates $L_{\mathsf{VPK}} := L_{\mathsf{VPK}} \cup \{(\mathsf{id}_{\mathsf{V}}, \mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}, \mathsf{vsk}_{\mathsf{id}_{\mathsf{V}}})\}$.

$\mathsf{O}_{\mathsf{VSK}}$**:** Given $\mathsf{id}_{\mathsf{V}} \in \mathcal{I}$, it does the following. If id_{V} is an identifier such that $\mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}} = \mathsf{pk}^*_{pke}$, then abort the experiment and output $\bot$. If $(\mathsf{id}_{\mathsf{V}}, \mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}, \mathsf{vsk}_{\mathsf{id}_{\mathsf{V}}}) \in L_{\mathsf{VPK}}$, then it returns $\mathsf{vsk}_{\mathsf{id}_{\mathsf{V}}}$, and updates $L_{\mathsf{VSK}} := L_{\mathsf{VSK}} \cup \{\mathsf{id}_{\mathsf{V}}\}$. Otherwise, it does the following.

- (This does not happen if $\mathsf{vpk}_{\mathsf{id}'_\mathsf{V}} \coloneqq \mathsf{pk}^*_{pke}$ has already occurred for some $\mathsf{id}'_\mathsf{V} \in \mathcal{I}$ in the simulation of $\mathsf{O}_{\mathsf{VPK}}$.) With probability $1/q$, it aborts the experiment. We regard this event as a corruption query on $\mathsf{vpk}_{\mathsf{id}_\mathsf{V}}$ is made.
- Otherwise, it computes $(\mathsf{vpk}_{\mathsf{id}_\mathsf{V}}, \mathsf{vsk}_{\mathsf{id}_\mathsf{V}}) \coloneqq (\mathsf{pk}_{pke,\mathsf{id}_\mathsf{V}}, \mathsf{sk}_{pke,\mathsf{id}_\mathsf{V}}) \leftarrow \Pi_P.\mathsf{KG}(1^\lambda)$, sets $L_{\mathsf{VPK}} \coloneqq L_{\mathsf{VPK}} \cup \{(\mathsf{id}_\mathsf{V}, \mathsf{vpk}_{\mathsf{id}_\mathsf{V}}, \mathsf{vsk}_{\mathsf{id}_\mathsf{V}})\}$, returns $(\mathsf{vpk}_{\mathsf{id}_\mathsf{V}}, \mathsf{vsk}_{\mathsf{id}_\mathsf{V}})$, and updates $L_{\mathsf{VSK}} \coloneqq L_{\mathsf{VSK}} \cup \{\mathsf{id}_\mathsf{V}\}$.

$\mathsf{O}_{\mathsf{Sig}}$**:** Given $\mathcal{D} \subseteq \mathcal{I}, \mathsf{id}_\mathsf{S} \in \mathcal{I}$, and $\mathsf{m} \in \mathcal{M}$, it does the followings:

- If $(\mathsf{id}_\mathsf{S}, \cdot, \cdot) \notin L_{\mathsf{SPK}}$, then it simulates $\mathsf{O}_{\mathsf{SPK}}$ on id_S to generate $(\mathsf{id}_\mathsf{S}, \mathsf{spk}_{\mathsf{id}_\mathsf{S}}, \mathsf{ssk}_{\mathsf{id}_\mathsf{S}})$.
- For all $\mathsf{id}_\mathsf{V} \in \mathcal{D}$ s.t. $(\mathsf{id}_\mathsf{V}, \cdot, \cdot) \notin L_{\mathsf{VPK}}$, it simulates $\mathsf{O}_{\mathsf{VPK}}$ on id_V to generate $(\mathsf{id}_\mathsf{V}, \mathsf{vpk}_{\mathsf{id}_\mathsf{V}}, \mathsf{vsk}_{\mathsf{id}_\mathsf{V}})$.
- Then, it creates $\sigma_{\mathsf{id}_\mathsf{S}} \leftarrow \Pi_S.\mathsf{Sig}(\mathsf{ssk}_{\mathsf{id}_\mathsf{S}}, \mathsf{m})$ and for each $\mathsf{id}_\mathsf{V} \in \mathcal{D}$, it computes $c_{\mathsf{id}_\mathsf{V}} \leftarrow \Pi_P.\mathsf{Enc}(\mathsf{vpk}_{\mathsf{id}_\mathsf{V}}, \sigma_{\mathsf{id}_\mathsf{S}})$. It returns $\sigma = \{(\mathsf{id}_\mathsf{V}, c_{\mathsf{id}_\mathsf{V}})\}_{\mathsf{id}_\mathsf{V} \in \mathcal{D}}$, and updates $L_{\mathsf{Sign}} \coloneqq L_{\mathsf{Sign}} \cup \{(\mathcal{D}, \mathsf{id}_\mathsf{S}, \mathsf{m})\}$.

At some point, $\mathcal{A}$ outputs $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{id}^*_\mathsf{S}, \mathcal{D})$. The algorithm $\mathcal{B}$ terminates and outputs $\perp$ if (i) pk^*_{pke} is not embedded in a verifier's public key, or (ii) $\mathcal{D}$ does not contain $\mathsf{vpk}_{\mathsf{id}^*_\mathsf{V}} = \mathsf{pk}^*_{pke}$. Otherwise, $\mathcal{B}$ computes $\mathsf{m}^*_0 \coloneqq \sigma_0 \leftarrow \Pi_S.\mathsf{Sig}(\mathsf{ssk}_{\mathsf{id}^*_\mathsf{S}}, \mathsf{m}_0)$ and $\mathsf{m}^*_1 \coloneqq \sigma_1 \leftarrow \Pi_S.\mathsf{Sig}(\mathsf{ssk}_{\mathsf{id}^*_\mathsf{S}}, \mathsf{m}_1)$, and returns m^*_0 and m^*_1 to the challenger of $\mathsf{ExpINDCPA}$.

Guessing Phase Given a ciphertext c^*, $\mathcal{B}$ does the following. It flips a random coin $b^\dagger \in \{0,1\}$ and computes $c_{\mathsf{id}_\mathsf{V}} \leftarrow \Pi_P.\mathsf{Enc}(\mathsf{pk}_{pke,\mathsf{id}_\mathsf{V}}, \mathsf{m}^*_{b^\dagger})$ for each $\mathsf{id}_\mathsf{V} \in \mathcal{D} \setminus \{\mathsf{id}^*_\mathsf{V}\}$. Then, it gives $\{(\mathsf{id}_\mathsf{V}, c_{\mathsf{id}_\mathsf{V}})\}_{\mathsf{id}_\mathsf{V} \in \mathcal{D}}$ to $\mathcal{A}$ where $c_{\mathsf{id}^*_\mathsf{V}} = c^*$. Note that the probability that $b^\dagger$ is equal to the bit chosen by the challenger of $\mathsf{ExpINDCPA}$ is exactly $1/2$, but it is sufficient for our purpose. When $\mathcal{A}$ outputs $b' \in \{0,1\}$, $\mathcal{B}$ returns b' to the challenger.

Analysis Observe that $\mathcal{B}$ wins if and only if $\mathcal{A}$ wins $\mathsf{ExpINDCPA}$ between $\mathcal{B}$. However, there are cases that $\mathcal{B}$ aborts as follows:

E_1: pk^*_{pke} is not embedded in a verifier's public key.
E_2: An identity id_V s.t. $\mathsf{vpk}_{\mathsf{id}_\mathsf{V}} = \mathsf{pk}^*_{pke}$ is queried to $\mathsf{O}_{\mathsf{VSK}}$.
E_3: $\mathcal{D}$ does not contain $\mathsf{vpk}_{\mathsf{id}_\mathsf{V}} = \mathsf{pk}^*_{pke}$.

We can evaluate the probability that each event does not occur as follows.

When $\mathcal{B}$ simulates $\mathsf{O}_{\mathsf{VPK}}$ on id_V, the probability that pk^*_{pke} is *not* embedded to the verifier's public key is $1-1/q$. Since at most q queries are made during the experiment, the probability that E_1 happens is $(1-1/q)^q \approx 1/e$ for sufficiently large q. Thus, the probability that E_1 does not occur is $1-1/e$.

Regarding the event E_2, observe that there should be at least one non-corrupted verifier's public key for $\mathcal{A}$'s winning condition on $\mathsf{ExpEUFIMDVS}$. In other words, at least one identity should not be queried to $\mathsf{O}_{\mathsf{VSK}}$. As an identity id_V s.t. $\mathsf{vpk}_{\mathsf{id}_\mathsf{V}} = \mathsf{pk}^*_{pke}$ is chosen uniformly at random, and at most q queries are made, the event E_2 does not occur with probability better than $1/q$.

The third event can be analyzed in a similar manner as E_2. That is, at least one verifier's public key is contained in $\mathcal{D}$. As an identity id_V s.t. $\mathsf{vpk}_{\mathsf{id}_\mathsf{V}} = \mathsf{pk}^*_{pke}$

is chosen randomly, and at most q verifier's public key is created during the experiment, the probability that $\mathsf{id}'_{\mathsf{V}} \in \mathcal{D}$ is at least $1/q$.

Recall that the advantage of $\mathcal{A}$ is non-negligible ϵ. Further, the probability that the signature $\{(\mathsf{id}_{\mathsf{V}}, c_{\mathsf{id}_{\mathsf{V}}})\}_{\mathsf{id}_{\mathsf{V}} \in \mathcal{D}}$ given to $\mathcal{A}$ is valid with probability 1/2. To sum up the discussion, the advantage of $\mathcal{B}$ in the experiment $\mathsf{ExpINDCPA}$ is at least

$$\frac{\epsilon}{2} \cdot \left(1 - \frac{1}{e}\right) \cdot \frac{1}{q^2},$$

which is still non-negligible. □

On Simulation of a Verification Oracle As mentioned in Section 3, an adversary is not given a verification oracle in $\mathsf{ExpSigIND}$. This is because we could not find how $\mathcal{B}$ simulates a verification oracle. Suppose that a verification oracle works as follows; given $\mathsf{id}'_{\mathsf{V}}, \mathsf{id}_{\mathsf{S}} \in \mathcal{I}, \mathsf{m} \in \mathcal{M}$, $\mathcal{D} \subseteq \mathcal{I}$ where $\mathsf{id}'_{\mathsf{V}} \in \mathcal{D}$, and σ, it outputs the verification result $b \in \{0, 1\}$. Let us consider the case that $\mathcal{D} = \{\mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}}\}$ where $\mathsf{vpk}_{\mathsf{id}_{\mathsf{V}}} = \mathsf{pk}^*_{pke}$. To verify σ, $\mathcal{B}$ should first decrypt it by using the secret key sk^*_{pke}, which corresponds to pk^*_{pke}. However, as we are considering IND-CPA security, no decryption oracle is given to $\mathcal{B}$. Therefore, $\mathcal{B}$ should decrypt σ without relying on a decryption oracle, which is almost impossible as long as Π_P is IND-CPA secure.

5 Separation Between Ring Signature and IMDVS

This section demonstrates that it is impossible to construct an IMDVS scheme from a ring signature scheme, which implies the separation of a PKE scheme from a ring signature scheme. Precisely, we prove that, given a ring signature scheme Π_{RS}, there is no black-box construction $\Pi_{\mathrm{IMDVS}}^{\Pi_{\mathrm{RS}}}$ whose EUF-CMA security is reduced to the EUF-CMA security of Π_{RS}.

The proof is similar to that of [32], which shows the impossibility of an MDVS scheme from a ring signature scheme. Roughly, the impossibility stems from the difference in the definitions of the EUF-CMA security between ring signature and IMDVS.

Theorem 5. *Let $\Pi_{\mathrm{RS}} = (\mathsf{Set}, \mathsf{KG}, \mathsf{Sig}, \mathsf{Vrf})$ be a ring signature scheme. There is no black-box construction $\Pi_{\mathrm{IMDVS}}^{\Pi_{\mathrm{RS}}} = (\mathsf{Set}, \mathsf{SKG}, \mathsf{VKG}, \mathsf{Sig}, \mathsf{Vrf})$ of an IMDVS scheme based on Π_{RS}, whose EUF-CMA security is reduced to EUF-CMA security of Π_{RS}.*

Proof. We assume for contradiction that there exists a PPT adversary $\mathcal{A}$ that breaks the EUF-CMA security of $\Pi_{\mathrm{IMDVS}}^{\Pi_{\mathrm{RS}}}$ with non-negligible probability ϵ. We demonstrate a PPT reduction algorithm R that accesses $\mathcal{A}$ in a black-box fashion to break the EUF-CMA security of Π_{RS} with non-negligible probability, or Π_{RS} is not EUF-CMA secure.

Although we do not know how $\Pi_{\mathrm{IMDVS}}^{\Pi_{\mathrm{RS}}}$ is constructed, we put the following natural assumptions on it: $\Pi_{\mathrm{IMDVS}}^{\Pi_{\mathrm{RS}}}.\mathsf{Set}$ uses $\Pi_{\mathrm{RS}}.\mathsf{Set}$, $\Pi_{\mathrm{IMDVS}}^{\Pi_{\mathrm{RS}}}.\mathsf{SKG}$ and $\Pi_{\mathrm{IMDVS}}^{\Pi_{\mathrm{RS}}}.\mathsf{VKG}$ use

$\Pi_{\mathrm{RS}}.\mathsf{KG}$ respectively, $\Pi_{\mathrm{IMDVS}}^{\Pi_{\mathrm{RS}}}.\mathsf{Sig}$ uses $\Pi_{\mathrm{RS}}.\mathsf{Sig}$ and $\Pi_{\mathrm{IMDVS}}^{\Pi_{\mathrm{RS}}}.\mathsf{Vrf}$ uses $\Pi_{\mathrm{RS}}.\mathsf{Vrf}$. Therefore, when $\mathcal{A}$ makes a query to an oracle, R asks the challenger of $\mathsf{ExpEUFRS}$ to call its oracle to answer the query by $\mathcal{A}$.

The reduction algorithm R works in $\mathsf{ExpEUFRS}$ as follows. Given a public parameter $\mathsf{pp}_{\mathsf{RS}}$, R initiates $L_{\mathsf{VPK}} \coloneqq \emptyset, L_{\mathsf{SPK}} \coloneqq \emptyset, L_{\mathsf{VSK}} \coloneqq \emptyset, L_{\mathsf{SSK}} \coloneqq \emptyset, L_{\mathsf{Sign}} \coloneqq \emptyset$, and $L_{\mathsf{Vrf}} \coloneqq \emptyset$. Then, it computes $(\mathsf{pp}_{\mathsf{IMDVS}}, \mathsf{msk}_{\mathsf{IMDVS}}) \leftarrow \Pi_{\mathrm{IMDVS}}^{\Pi_{\mathrm{RS}}}.\mathsf{Set}(1^\lambda)$ (based on $\mathsf{pp}_{\mathsf{RS}}$), and runs $\mathcal{A}$ on $\mathsf{pp}_{\mathsf{IMDVS}}$. When $\mathcal{A}$ outputs $(\mathsf{id}_{\mathsf{S}}^\dagger, \mathcal{D}^\dagger, \mathsf{m}^\dagger, \sigma^\dagger)$, R returns $(R^* \coloneqq \{\mathsf{pk}_i^*\}_{i\in[n]}, \mathsf{m}^*, \sigma^*)$ to the challenger where $n = \mathsf{poly}(\lambda)$.

The adversary $\mathsf{R}^{\mathcal{A}}$ succeeds to forge a ring signature if all the following conditions are satisfied.

- $\Pi_{\mathrm{RS}}.\mathsf{Vrf}(\mathsf{pp}_{\mathsf{RS}}, R^*, \mathsf{m}^*, \sigma^*) = 1$.
- Every pk_i^* is created by the oracle O_{PK}.
- Every pk_i^* is not queried to O_{SK} (i.e., no ring member is corrupted).
- The signature σ^* is not created by $\mathsf{O}_{\mathsf{RSig}}$ on $(\mathsf{pk}_j^*, R^*, \mathsf{m}^*)$ for some $\mathsf{pk}_j^* \in R^*$.

We focus on the third condition and provide an intuition behind our proof. If $\mathcal{A}$ makes a query that necessitates corrupting a public key in R^* (we call such a query as a *ring corruption query*), then R should answer it without relying on O_{SK}. However, if such a computation is possible, then R is able to break the EUF-CMA security of Π_{RS} by itself without assuming the existence of $\mathcal{A}$. We further consider the case where $\mathcal{A}$ never makes a ring corruption query. In this case, we show that Π_{RS} is not EUF-CMA secure. The formal argument is as follows.

Case 1: $\mathcal{A}$ makes a ring corruption query. As mentioned above, R is able to create a signing key of a ring member of the ring R^* without calling O_{SK}. We demonstrate a PPT algorithm R' that breaks the EUF-CMA security of Π_{RS} by simulating R as follows.

- Given a public parameter $\mathsf{pp}_{\mathsf{RS}}$, it creates $R^* = \{\mathsf{pk}_i^*\}_{i\in[n]}$ by calling O_{PK}, where $n = \mathsf{poly}(\lambda)$.
- For each $i \in [n]$, it tries to create a signing key sk_i^* by simulating R. If such a key is obtained, it moves to the next step. If the signing key is not obtained for all $i \in [n]$, then it terminates and outputs $\perp$.
- It chooses a message m^* at random and computes $\sigma^* \leftarrow \Pi_{\mathrm{RS}}.\mathsf{Sig}(\mathsf{pp}_{\mathsf{RS}}, \mathsf{sk}_i^*, R^*, \mathsf{m}^*)$.
- It returns $(R^*, \mathsf{m}^*, \sigma^*)$ to the challenger.

Observe that $(\mathsf{pk}_i^*, R^*, \mathsf{m}^*, \sigma^*)$ is not recorded in L_{Sign} as it is generated locally by R'. Thus, if R' outputs $(R^*, \mathsf{m}^*, \sigma^*)$, then it wins the game. Considering the probability that $\mathcal{A}$ breaks the EUF-CMA security of $\Pi_{\mathrm{IMDVS}}^{\Pi_{\mathrm{RS}}}$, R is able to find a signing key of a ring member of R^* with probability better than ϵ. Therefore, R' breaks the EUF-CMA security of $\Pi_{\mathrm{IMDVS}}^{\Pi_{\mathrm{RS}}}$ with probability better than ϵ, which is non-negligible.

Case 2: $\mathcal{A}$ never makes a ring corruption query. This case should be devided in two subcases: whether $\mathcal{A}$ asks a query that necessitates the query $(\mathsf{pk}_j^*, R^*, \mathsf{m}_{\mathsf{RS}}^*)$ to $\mathsf{O}_{\mathsf{RSig}}$ for some $j \in [n]$ or not.

If it is the case, then R should return a valid ring signature without calling $\mathsf{O}_{\mathsf{RSig}}$, and thus we can construct another PPT algorithm R' that breaks the EUF-CMA security of Π_{RS} with non-negligible probability by a similar discussion as Case 1.

Otherwise, Π_{RS} is no longer EUF-CMA secure since $\mathcal{A}$ neither necessitates to corrupt a ring member nor to create a ring signature with respect to R^*.

In the first subcase, we can construct a PPT algorithm R' as follows.

- Given a public parameter $\mathsf{pp}_{\mathsf{RS}}$, it creates $R^* = \{\mathsf{pk}_i^*\}_{i\in[n]}$ by calling O_{PK}, where $n = \mathsf{poly}(\lambda)$.
- It chooses a message m^* at random and (somehow) computes a signature σ^* s.t. $\Pi_{\mathrm{RS}}.\mathsf{Vrf}(\mathsf{pp}_{\mathsf{RS}}, R^*, \mathsf{m}^*, \sigma^*) = 1$. If it cannot create such a ring signature σ^*, then it terminates and outputs $\perp$.
- It returns $(R^*, \mathsf{m}^*, \sigma^*)$ to the challenger.

Observe that R' wins the game if it outputs $(R^*, \mathsf{m}^*, \sigma^*)$. Considering the winning probability of $\mathcal{A}$, the probability that R' succeeds in forging a ring signature is at least ϵ, which is non-negligible.

We finally consider the second subcase. Recall that we are assuming that the key generation algorithms of $\Pi_{\mathrm{IMDVS}}^{\Pi_{\mathrm{RS}}}$ need to call O_{PK}. Thus, every public key of a ring member of R^* is created by the oracle O_{PK}. Since we are considering Case 2 and its second subcase, no ring member of R^* is corrupted and a forged signature σ^* is not created by $\mathsf{O}_{\mathsf{Sig}}$. Therefore, even if we assume the existence of $\mathcal{A}$ that breaks the EUF-CMA security of $\Pi_{\mathrm{IMDVS}}^{\Pi_{\mathrm{RS}}}$, R is able to simulate $\mathcal{A}$ to break the EUF-CMA security of Π_{RS} with non-negligible probability. Thus, if Π_{RS} allows this subcase, it is no longer EUF-CMA secure. □

As mentioned earlier, it is known that group signature implies PKE [1, 9, 23]. Therefore, combining it with Theorem 5, we obtain the following corollary.

Corollary 1. *There is no black-box construction of group signature based on ring signature.*

6 Conclusion and Future Work

In this paper, we have proven that it is impossible to construct PKE from ring signature in a black-box manner and partly answer the question if ring signature belongs to Cryptomania or Minicrypt. Furthermore, this result indicates an essential difference between ring signature and group signature because group signature implies PKE [1].

While it is known that ring signature can be constructed by using Cryptomania primitives in the standard model or OR-proof systems in the random oracle model, to the best of our knowledge, it is still unclear if we can obtain ring signature only from Minicrypt primitives in the standard model. We leave this as an interesting open problem.

Acknowledgment

This research was in part supported by Grant-in-Aid for Scientific Research (A) (JP23H00468). This research also was in part supported by JST-CREST JPMJCR22M1 and JST-AIP JPMJCR22U5.

References

1. Abdalla, M., Warinschi, B.: On the minimal assumptions of group signature schemes. In: Lopez, J., Qing, S., Okamoto, E. (eds.) Information and Communications Security. pp. 1–13. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
2. Backes, M., Döttling, N., Hanzlik, L., Kluczniak, K., Schneider, J.: Ring signatures: Logarithmic-size, no setup—from standard assumptions. In: Ishai, Y., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2019. pp. 281–311. Springer International Publishing, Cham (2019)
3. Bender, A., Katz, J., Morselli, R.: Ring signatures: Stronger definitions, and constructions without random oracles. In: Halevi, S., Rabin, T. (eds.) Theory of Cryptography. pp. 60–79. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
4. Boneh, D., Papakonstantinou, P., Rackoff, C., Vahlis, Y., Waters, B.: On the impossibility of basing identity based encryption on trapdoor permutations. In: Proceedings of the 2008 49th Annual IEEE Symposium on Foundations of Computer Science. p. 283–292. FOCS '08, IEEE Computer Society, USA (2008)
5. Bootle, J., Cerulli, A., Chaidos, P., Ghadafi, E., Groth, J.: Foundations of fully dynamic group signatures. In: Manulis, M., Sadeghi, A., Schneider, S.A. (eds.) Applied Cryptography and Network Security - 14th International Conference, ACNS 2016, Guildford, UK, June 19-22, 2016. Proceedings. Lecture Notes in Computer Science, vol. 9696, pp. 117–136. Springer (2016). https://doi.org/10.1007/978-3-319-39555-5_7, https://doi.org/10.1007/978-3-319-39555-5_7
6. Chaum, D., van Heyst, E.: Group signatures. In: Davies, D.W. (ed.) Advances in Cryptology — EUROCRYPT '91. pp. 257–265. Springer Berlin Heidelberg, Berlin, Heidelberg (1991)
7. Cramer, R., Damgård, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: Desmedt, Y.G. (ed.) Advances in Cryptology — CRYPTO '94. pp. 174–187. Springer Berlin Heidelberg, Berlin, Heidelberg (1994)
8. Damgård, I., Haagh, H., Mercer, R., Nitulescu, A., Orlandi, C., Yakoubov, S.: Stronger security and constructions of multi-designated verifier signatures. In: Pass, R., Pietrzak, K. (eds.) Theory of Cryptography. pp. 229–260. Springer International Publishing, Cham (2020)
9. Emura, K., Hanaoka, G., Sakai, Y.: Group signature implies pke with non-interactive opening and threshold pke. In: Echizen, I., Kunihiro, N., Sasaki, R. (eds.) Advances in Information and Computer Security. pp. 181–198. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
10. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) Advances in Cryptology — CRYPTO' 86. pp. 186–194. Springer Berlin Heidelberg, Berlin, Heidelberg (1987)
11. Fujisaki, E., Suzuki, K.: Traceable ring signature. In: Public Key Cryptography – PKC 2007. pp. 181–200 (2007)

12. Garg, S., Pandey, O., Srinivasan, A., Zhandry, M.: Breaking the sub-exponential barrier in obfustopia. In: Coron, J.S., Nielsen, J.B. (eds.) Advances in Cryptology – EUROCRYPT 2017. pp. 156–181. Springer International Publishing, Cham (2017)
13. Gennaro, R., Trevisan, L.: Lower bounds on the efficiency of generic cryptographic constructions. In: Proceedings of the 41st Annual Symposium on Foundations of Computer Science. p. 305. FOCS '00, IEEE Computer Society, USA (2000)
14. Gertner, Y., Kannan, S., Malkin, T., Reingold, O., Viswanathan, M.: The relationship between public key encryption and oblivious transfer. In: Proceedings 41st Annual Symposium on Foundations of Computer Science. pp. 325–335 (2000)
15. Gertner, Y., Malkin, T., Myers, S.: Towards a separation of semantic and cca security for public key encryption. In: Vadhan, S.P. (ed.) Theory of Cryptography. pp. 434–455. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
16. Impagliazzo, R.: A personal view of average-case complexity. In: Proceedings of Structure in Complexity Theory. Tenth Annual IEEE Conference. pp. 134–147 (1995)
17. Impagliazzo, R., Rudich, S.: Limits on the provable consequences of one-way permutations. In: Goldwasser, S. (ed.) Advances in Cryptology — CRYPTO' 88. pp. 8–26. Springer New York, New York, NY (1990)
18. Katz, J., Schröder, D., Yerukhimovich, A.: Impossibility of blind signatures from one-way permutations. In: Ishai, Y. (ed.) Theory of Cryptography. pp. 615–629. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
19. Komano, Y., Ohta, K., Shimbo, A., Kawamura, S.: Toward the fair anonymous signatures: Deniable ring signatures. In: Pointcheval, D. (ed.) Topics in Cryptology – CT-RSA 2006. pp. 174–191. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
20. Laguillaumie, F., Vergnaud, D.: Multi-designated verifiers signatures. In: Lopez, J., Qing, S., Okamoto, E. (eds.) Information and Communications Security. pp. 495–507. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
21. Laguillaumie, F., Vergnaud, D.: Multi-designated verifiers signatures: anonymity without encryption. Information Processing Letters **102**(2), 127–132 (2007)
22. Liu, J.K., Wei, V.K., Wong, D.S.: Linkable spontaneous anonymous group signature for ad hoc groups. In: Wang, H., Pieprzyk, J., Varadharajan, V. (eds.) Information Security and Privacy. pp. 325–335. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
23. Ohtake, G., Fujii, A., Hanaoka, G., Ogawa, K.: On the theoretical gap between group signatures with and without unlinkability. In: Preneel, B. (ed.) Progress in Cryptology – AFRICACRYPT 2009. pp. 149–166. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
24. Okamoto, T.: Designated confirmer signatures and public-key encryption are equivalent. In: Desmedt, Y.G. (ed.) Advances in Cryptology — CRYPTO '94. pp. 61–74. Springer Berlin Heidelberg, Berlin, Heidelberg (1994)
25. Park, S., Sealfon, A.: It wasn't me! repudiability and unclaimability of ring signatures. In: Annual International Cryptology Conference. pp. 159–190. Springer (2019)
26. Perera, N., Nakamura, T., Hashimoto, M., Yokoyama, H., Cheng, C.M., Sakurai, K.: A survey on group signatures and ring signatures: Traceability vs. anonymity. Cryptography **6**, 3 (01 2022)
27. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: Boyd, C. (ed.) Advances in Cryptology — ASIACRYPT 2001. pp. 552–565. Springer Berlin Heidelberg, Berlin, Heidelberg (2001)
28. Rivest, R.L., Shamir, A., Tauman, Y.: How to Leak a Secret: Theory and Applications of Ring Signatures, p. 164–186. Springer-Verlag (2006)

29. Rompel, J.: One-way functions are necessary and sufficient for secure signatures. In: Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing. p. 387–394. STOC '90, Association for Computing Machinery, New York, NY, USA (1990)
30. Vergnaud, D.: New extensions of pairing-based signatures into universal designated verifier signatures. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) Automata, Languages and Programming. pp. 58–69. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
31. Xu, S., Yung, M.: Accountable ring signatures: A smart card approach. In: Quisquater, J.J., Paradinas, P., Deswarte, Y., El Kalam, A.A. (eds.) Smart Card Research and Advanced Applications VI. pp. 271–286. Springer US, Boston, MA (2004)
32. Yamashita, K., Hara, K.: On the black-box impossibility of multi-designated verifiers signature schemes from ring signature schemes. Cryptology ePrint Archive, Paper 2023/1249 (2023), https://eprint.iacr.org/2023/1249
33. Zhang, Y., Au, M.H., Yang, G., Susilo, W.: (strong) multi-designated verifiers signatures secure against rogue key attack. In: Xu, L., Bertino, E., Mu, Y. (eds.) Network and System Security. pp. 334–347. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)