Designated Verifier Signature with Claimability

Kyosuke Yamashita
Osaka University
Suita, Osaka, Japan
yamashita@ist.osaka-u.ac.jp

Keisuke Hara
National Institute of Advanced
Industrial Science and Technology
Koto-ku, Tokyo, Japan
Yokohama National University
Yokohama, Kanagawa, Japan

Yohei Watanabe
The University of
Electro-Communications
Chofu, Tokyo, Japan
Japan Datacom Co., Ltd.
Minato-ku, Tokyo, Japan

Naoto Yanai
Osaka University
Suita, Osaka, Japan
Japan Datacom Co., Ltd.
Minato-ku, Tokyo, Japan

Junji Shikata
Yokohama National University
Yokohama, Kanagawa, Japan

ABSTRACT

This paper considers the problem of balancing traceability and anonymity in designated verifier signatures (DVS), which are a kind of group-oriented signatures. That is, we propose claimable designated verifier signatures (CDVS), where a signer is able to claim that he/she indeed created a signature later. Ordinal DVS does not provide any traceability, which could indicate too strong anonymity. Thus, adding claimability, which can be seen as a sort of traceability, moderates anonymity. We demonstrate two generic constructions of CDVS from (i) ring signatures, (non-ring) signatures, pseudorandom function, and commitment scheme, and (ii) claimable ring signatures (by Park and Sealfon, CRYPTO'19).

CCS CONCEPTS

• **Security and privacy** → **Digital signatures**; *Access control.*

KEYWORDS

designated verifier signature, ring signature, anonymity, traceability, claimability.

ACM Reference Format:
Kyosuke Yamashita, Keisuke Hara, Yohei Watanabe, Naoto Yanai, and Junji Shikata. 2023. Designated Verifier Signature with Claimability. In *Proceedings of the10th ACM Asia Public-Key Cryptography Workshop (APKC '23), July 10–14, 2023, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3591866.3593071

1 INTRODUCTION

Group-oriented signature schemes, such as ring signatures [27] and group signatures [9], are equipped with a functionality that a signer can create a signature on behalf of a group of users, but verifiers cannot identify the signer. Specifically, in both schemes, the signer forms a group of other users, i.e., potential signers, and signs messages. The verifier is only convinced that the messages were signed by the group. By virtue of such an anonymity notion, these signature schemes can be used in applications such as e-commerce systems or e-voting potentially.

One of the major differences between group and ring signatures is that, when necessary, the former has the functionality to trace who the signer is. The functionality called *traceability* stems from the existence of a trusted third party; it has a secret key to trace the signer. On the other hand, there are only users but no authority to trace the signer in ring signatures. Therefore, the signers cannot claim ownership of their signed messages even if they want to claim it later. To make ring signatures *claimable*, Park and Sealfon [25] proposed *claimable ring signatures*, which enable the signer to generate a proof for a signature that the signer indeed generated it. As such, cryptographic protocols that provide a sort of anonymity sometimes lead to an ownership problem, and it is important in practice to consider traceability or claimability in group-oriented signatures.

In this paper, we focus on designated verifier signatures (DVS) [8, 17], which is a kind of group-oriented signature. In DVS, a signer can designate a verifier and create a signature so that they can only verify the signature. Off-the-record (OTR)[1] [6], a sort of anonymity notion for DVS, guarantees that a designated verifier has the ability to produce the signature (designating the verifier) *from any signer*; therefore, no third party is convinced of who generates the signature. DVS also has an ownership problem due to OTR. Consider that the signer wants to sign a non-disclosure agreement (NDA). OTR forces the verifier not to disclose the NDA. However, if the signer wants to make it public (e.g., due to waiver of NDA), there is no means of convincing any third party. Thus, it is preferable for the signer to be able to claim the ownership of signatures.

1.1 Our Contribution

In this paper, we introduce *claimable* DVS (CDVS), where claimability is a property that a signer is able to claim that he indeed created the signature. We give a formal definition of claimability, and demonstrate two generic constructions of CDVS. One is from standard ring signature, (non-ring, standard) signature, pseudorandom function, and commitment schemes. We note that this construction is based on the existing claimable ring signature scheme [25]. The

[1] OTR property is also known as so hiding property.

APKC '23, July 10–14, 2023, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-0183-2/23/07.
https://doi.org/10.1145/3591866.3593071

other is from claimable ring signatures. Although claimable ring signatures can be obtained from the same primitives as our first construction, we claim that the second construction is important as well. That is, if claimable ring signature is constructed from other primitives in the future, it immediately implies a new CDVS construction.

1.2 Related Work

DVS is proposed by Chaum [8], and by Jakobsson et al. [17] independently in 1996. Desmedt opens the question if it is possible to construct a multi-designated verifier signature (MDVS) scheme at CRYPTO'03 ramp session, and Laguillaumie and Vergnaud [20] answer this question positively. They construct MDVS based on ring signature scheme [5, 27]. (We note that Rivest et al. [27] mention that MDVS can be constructed from ring signature, before [20].) Follow-up works [2, 10, 21, 31] propose variants of (M)DVS from ring signature scheme.

We argue that such a construction is widely employed because (M)DVS is highly compatible with ring signatures. That is, anonymity in ring signature is similar to OTR. Anonymity in ring signature requires that a signer among a particular set of parties (called a "ring") signs on a message, but verifiers cannot distinguish who created the signature. In other words, every member in the ring is able to create a signature with respect to the ring. Recall that OTR requires that designated verifiers can simulate a signature. Thus, we can obtain (M)DVS from ring signature by regarding a ring as a set of a signer and designated verifiers.

Park and Sealfon [25] demonstrate that claimable ring signature can be obtained from any ring signature. It is known that ring signature can be constructed from both generic and specific assumptions as follows; the existence of trapdoor permutation in the random oracle model [27], public key encryption, signature, and ZAP [4], the discrete logarithm assumption [1, 15], and the RSA assumption [12]. Therefore, CDVS can be obtained from these assumptions for free.

The problem of balancing anonymity and traceability in group-oriented signatures is discussed in [22, 26]. That is, traceability in group signature obviously decreases anonymity. In particular, if a tracer is corrupted, then anonymity is completely compromised. On the other hand, anonymity in ring signature benefits malicious signers. As demonstrated in [26], a lot of methods have been proposed to solve these problems; regarding group signature, user dependent opening [18], decentralized tracing [24], message-dependent opening [29], distributed tracing [14], and accountable tracing [19], and regarding ring signature, accountable ring signature [32], linkable ring signature [23], traceable ring signature [13], and claimable ring signature [25].

Recently, designated verifier linkable ring signature has been proposed [3]. Linkability is a property that, given two signatures, we can decide if they are created by the same signer, without disclosing the signer. As mentioned in [26], linkability is a variant of traceability. Therefore, while linkability is incomparable to claimability, we argue that they tackle the problem of balancing traceability and anonymity of DVS.

There are variants of signature schemes that relate to DVS, such as designated confirmer signature scheme [7], strong DVS [28], or universal DVS [16, 30]. Thus, it might be possible to combine claimability with these variants.

Paper Organization. Section 2 introduces basic notation. In Section 3, we define CDVS. Section 4 and Section 5 present the first and the second generic constructions of CDVS, respectively. Finally, Section 6 concludes the paper.

2 PRELIMINARIES

Throughout this paper, we let poly() be a polynomial function, and negl() be a negligible function. For any $n \in \mathbb{N}$, let $[n] = \{1, 2, \cdots, n\}$. Let $\lambda \in \mathbb{N}$ be a security parameter.

A probabilistic polynomial time is abbreviated as PPT. When an algorithm Π has a subroutine X, we denote it by $\Pi.X$. We assume that every algorithm is given a security parameter 1^λ as an input.

If a probabilistic algorithm A takes an input x and a randomness r, we denote it by $A(x; r)$. For simplicity, we sometimes omit r from its interface to denote $A(x)$.

Security of primitives is defined by an experiment (or a game) between a challenger and an adversary. The adversary might be able to ask the challenger to call an oracle to obtain some value. We implicitly assume that when the challenger calls an oracle, the challenger chooses randomness that is given to the oracle, if the oracle runs a probabilistic algorithm inside.

2.1 Primitives

Definition 1 (Pseudorandom Function). *A pseudorandom function is a pair of polynomial time algorithms* $(\mathsf{KG}, \mathsf{Eval})$ *that work as follows:*

- $\mathsf{KG}(1^\lambda) \to k$*: Given a security parameter* 1^λ*, it outputs a key* k*.*
- $\mathsf{Eval}(k, x) = r$*: Given a key* k*, and a string* $x \in \{0,1\}^*$*, it outputs a string* $r \in \{0,1\}^\lambda$*.*

A PRF should satisfy the following condition. For any sufficiently large security parameter 1^λ*, any* $k \leftarrow \mathsf{KG}(1^\lambda)$*, any truly random function* F *whose range is the same as* $\mathsf{Eval}(k, \cdot)$*, and any PPT algorithm* D*, it holds that*

$$|\Pr[1 \leftarrow D^{\mathsf{Eval}(k,\cdot)}(1^\lambda)] - \Pr[1 \leftarrow D^{F(\cdot)}(1^\lambda)]| \leq 1/2 + \mathsf{negl}(\lambda).$$

Definition 2 (Commitment). *A commitment scheme consists of two polynomial time algorithms* $(\mathsf{Com}, \mathsf{Open})$ *that works as follows:*

- $\mathsf{Com}(\mathsf{m}; r) \to c$*: Given a message* m*, and a randomness* r*, it outputs a commitment* c*.*
- $\mathsf{Open}(c) = r$*: Given* c*, it outputs a randomness* r*.*

For convenience, we define the input of PRF to be an arbitrary polynomial length string.

A commitment scheme should satisfy the following conditions.

Definition 3 (Binding). *A commitment scheme* $(\mathsf{Com}, \mathsf{Open})$ *is binding if for any sufficiently large security parameter* λ*, and any PPT adversary* $\mathcal{A}$*, it holds that*

$$\Pr\left[(c, \mathsf{m}, r, \mathsf{m}', r') \leftarrow \mathcal{A}(1^\lambda) \ : \ \begin{array}{l} \mathsf{m} \neq \mathsf{m}' \wedge \\ \mathsf{Com}(\mathsf{m}; r) = c = \mathsf{Com}(\mathsf{m}'; r') \end{array} \right] \leq \mathsf{negl}(\lambda).$$

Definition 4 (Hiding). *A commitment scheme* (Com, Open) *is hiding if for any sufficiently large security parameter λ, and any stateful PPT adversary $\mathcal{A}$, it holds that*

$$\Pr\left[\begin{array}{c}(\mathsf{m}_0,\mathsf{m}_1)\leftarrow\mathcal{A}(1^\lambda)\\ b\leftarrow\{0,1\};r\leftarrow\{0,1\}^{\mathsf{poly}(\lambda)};\ :\ b'=b\\ c\leftarrow\mathsf{Com}(\mathsf{m}_b;r);b'\leftarrow\mathcal{A}(c)\end{array}\right]\leq 1/2+\mathsf{negl}(\lambda).$$

Definition 5 (Signature). *A signature scheme consists of three polynomial time algorithms* (KG, Sig, Verify) *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.
- $\mathsf{Verify}(\mathsf{pk},\mathsf{m},\sigma)=1/0$: *Given a public key* pk*, a message* m*, and a signature* σ*, it outputs* 1 *(meaning valid) or* 0 *(meaning invalid).*

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

Definition 6 (EUF-CMA). *A signature scheme* $\Pi=(\mathsf{KG},\mathsf{Sig},\mathsf{Verify})$ *is existentially unforgeable under an adaptive chosen-message attack (EUF-CMA) 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* ExpEUFSig *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 $\mathsf{Verify}(\mathsf{pk},\mathsf{m}^*,\sigma^*)=1\wedge\mathsf{m}^*\notin L$, otherwise 0

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

2.2 Ring Signatures

We introduce standard ring signature and claimable ring signature.

2.2.1 Ring Signature.

Definition 7 (Ring Signature). *A ring signature scheme consists of four polynomial time algorithms* (Set, KG, RSig, Verify) *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.
- $\mathsf{RSig}(\mathsf{pp},\mathsf{sk},\{\mathsf{pk}_i\}_{i\in[n]},\mathsf{m})\to\sigma$*: 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* $\perp$.
- $\mathsf{Verify}(\mathsf{pp},\{\mathsf{pk}_i\}_{i\in[n]},\mathsf{m},\sigma)=1/0$*: 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 valid) or* 0 *(meaning invalid).*

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

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

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

Definition 8 (EUF-CMA). *A ring signature scheme* $\Pi_{RS}=(\mathsf{Set},\mathsf{KG},\mathsf{RSig},\mathsf{Verify})$ *is existentially unforgeable under an adaptive chosen-message attack (EUF-CMA) if for any sufficiently large security parameter λ and any PPT adversary $\mathcal{A}$ who is allowed to make at most q queries to an oracle,* $\Pr[\mathsf{ExpEUFRS}^{O}_{\Pi_{RS},\mathcal{A}}(1^\lambda)=1]\leq\mathsf{negl}(\lambda)$ *where the experiment* $\mathsf{ExpEUFRS}^{O}_{\Pi_{RS},\mathcal{A}}(1^\lambda)$ *is defined as follows:*

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

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

where $n=\mathsf{poly}(\lambda)$ *s.t.* $n\leq q$ *and* O *is some additional oracle (if necessary), 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* $L_{\mathsf{PK}}:=L_{\mathsf{PK}}\cup\{(\mathsf{pk},\mathsf{sk})\}$.
O_{SK}*: Given* pk*, if* $(\mathsf{pk},\mathsf{sk})\in L_{\mathsf{PK}}$*, then it returns* sk*, and updates* $L_{\mathsf{SK}}:=L_{\mathsf{SK}}\cup\{(\mathsf{pk},\mathsf{sk})\}$*. 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* $\sigma\leftarrow\mathsf{RSig}(\mathsf{pp},\mathsf{sk},\{\mathsf{pk}\}\cup\{\mathsf{pk}_i\}_{i\in[n']},\mathsf{m})$ *and updates* $L_{\mathsf{Sign}}:=L_{\mathsf{Sign}}\cup\{(\mathsf{pk},\{\mathsf{pk}_i\}_{i\in[n']},\mathsf{m},\sigma)\}$.

For EUF-CMA of ring signature, we set $O:=\phi$.

We define anonymity with respect to adversarially chosen keys as follows.

Definition 9 (Anonymity). *A ring signature scheme* $\Pi_{RS}=(\mathsf{Set},\mathsf{KG},\mathsf{RSig},\mathsf{Verify})$ *satisfies anonymity if for any sufficiently large security parameter λ, and any PPT adversary $\mathcal{A}$ who is allowed to make at most q queries to oracles,* $|\Pr[\mathsf{ExpAno}^{O}_{\Pi_{RS},\mathcal{A}}(1^\lambda)=1]-1/2|\leq\mathsf{negl}(\lambda)$*, where* $\mathsf{ExpAno}^{O}_{\Pi_{RS},\mathcal{A}}(1^\lambda)$ *is defined as follows:*

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

$L_{\mathsf{PK}}:=\emptyset;L_{\mathsf{SK}}:=\emptyset;L_{\mathsf{Sign}}:=\emptyset;$
$(\mathsf{m}^*,\mathsf{pk}_0,\mathsf{pk}_1,\{\mathsf{pk}_i^*\}_{i\in[n]})\leftarrow\mathcal{A}^{\mathsf{O}_{\mathsf{PK}},\mathsf{O}_{\mathsf{SK}},\mathsf{O}_{\mathsf{RSig}},O}(\mathsf{pp});$
$b\leftarrow\{0,1\};\sigma_b\leftarrow\mathsf{RSig}(\mathsf{pp},\mathsf{sk}_b,\{\mathsf{pk}_0,\mathsf{pk}_1\}\cup\{\mathsf{pk}_i^*\}_{i\in[n]},\mathsf{m}^*);$
$b'\leftarrow\mathcal{A}^{\mathsf{O}_{\mathsf{PK}},\mathsf{O}_{\mathsf{SK}},\mathsf{O}_{\mathsf{RSig}},O}(\sigma_b)$:
output 1 if $(b'=b)$
$\wedge((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1)\in L_{\mathsf{PK}})\wedge((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1)\notin L_{\mathsf{SK}})$
otherwise 0.

where $n=\mathsf{poly}(\lambda)$ *s.t.* $n\leq q$ *and* O *is some additional oracle (if necessary), and the oracles* $\mathsf{O}_{\mathsf{PK}},\mathsf{O}_{\mathsf{SK}}$*, and* $\mathsf{O}_{\mathsf{RSig}}$ *are defines as in Definition 8. For anonymity of ring signature, we set* $O:=\phi$.

2.2.2 Claimable Ring Signature. We recall claimable ring signature proposed by Park and Sealfon [25]. Compared to ordinal ring signature, claimable ring signature has two additional algorithms Claim and ClmVrf. A signer runs Claim when he wants to claim the ownership of a signature. (Thus, Claim takes a secret key of a signer as an input.) We note that ClmVrf can be run by any party for checking the validity of a claim.

Definition 10 (Claimable Ring Signature). *Claimable ring signature is ring signature with two additional algorithms* Claim *and* ClmVrf *that work as follows:*

- $\mathsf{Claim}(\mathsf{pp}, \mathsf{sk}, \{\mathsf{pk}_i\}_{i\in[n]}, \sigma) \to \pi$*: Given a public parameter* pp*, a secret key* sk*, a set of public keys* $\{\mathsf{pk}_i\}_{i\in[n]}$ *where* $n = \mathsf{poly}(\lambda)$*, and a signature* σ*, it outputs a claim* π*. If there is no* $i \in [n]$ *s.t.* $(\mathsf{pk}_i, \mathsf{sk}) \leftarrow \mathsf{Set}(\mathsf{pp})$*, then it returns* $\perp$*.*
- $\mathsf{ClmVrf}(\mathsf{pp}, \mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n]}, \sigma, \pi) = 1/0$ *: Given a public parameter* pp*, a public key* pk*, a set of public keys* $\{\mathsf{pk}_i\}_{i\in[n]}$ *where* $n = \mathsf{poly}(\lambda)$*, a signature* σ*, and a claim* π*, it outputs* 1 *(meaning valid) or* 0 *(meaning invalid).*

Correctness of CDVS is defined as in Definition 7.

Definition 11 (EUF-CMA). *A claimable ring signature scheme* Π_{CRSIG} *is existentially unforgeable under an adaptive chosen-message attack (EUF-CMA) if for any sufficiently large security parameter* λ*, and a PPT adversary* $\mathcal{A}$ *who is allowed to make at most* q *queries to oracles, it holds that* $\Pr[\mathsf{ExpEUFRS}^{\mathsf{O}_{\mathsf{CRSClm}}}_{\Pi_{CRSIG},\mathcal{A}}(1^\lambda) = 1] \leq \mathsf{negl}(\lambda)$ *where the experiment* $\mathsf{ExpEUFRS}^{\mathsf{O}_{\mathsf{CRSClm}}}_{\Pi_{CRSIG},\mathcal{A}}(1^\lambda)$ *is defined as in Definition 8, apart from it additionally sets* $L_{\mathsf{Clm}} := \phi$ *at the beginning and* $\mathsf{O}_{\mathsf{CRSClm}}$ *is defined as follows:*

- $\mathsf{O}_{\mathsf{CRSClm}}$*: 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 signature* σ*, it does the followings:*
  - *If* $(\mathsf{pk}, \mathsf{sk}) \notin L_{\mathsf{PK}}$*, then returns* $\perp$*.*
  - *If* $(\mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n']}, \sigma, \pi) \in L_{\mathsf{Clm}}$*, then returns* π*.*
  - *Returns* $\pi \leftarrow \mathsf{Claim}(\mathsf{pp}, \mathsf{sk}, \{\mathsf{pk}\} \cup \{\mathsf{pk}_i\}_{i\in[n']}, \sigma)$ *where* $n' \leq q$ *and updates* $L_{\mathsf{Clm}} := L_{\mathsf{Clm}} \cup \{\mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n']}, \sigma, \pi\}$*.*

Definition 12 (Anonymity). *A claimable ring signature scheme* $\Pi_{CRSIG} = (\mathsf{Set}, \mathsf{KG}, \mathsf{RSig}, \mathsf{Verify}, \mathsf{Claim}, \mathsf{ClmVrf})$ *satisfies anonymity if for any sufficiently large security parameter* λ*, and any PPT adversary* $\mathcal{A}$ *that is allowed to make at most* q *queries to oracles, it holds that* $|\Pr[\mathsf{ExpAno}^{\mathsf{O}_{\mathsf{CRSClm}}}_{\Pi_{CRSIG},\mathcal{A}}(1^\lambda) = 1] - 1/2| \leq \mathsf{negl}(\lambda)$*, where* $\mathsf{O}_{\mathsf{CRSClm}}$ *is defined as in Definition 11, and* $\mathsf{ExpAno}^{\mathsf{O}_{\mathsf{CRSClm}}}_{\Pi_{CRSIG},\mathcal{A}}(1^\lambda)$ *is defined as in Definition 9 and Definition 11 with the modification so that it sets* $L_{\mathsf{Clm}} := \phi$ *at the beginning of the experiment.*

Now, we recall the definition of claimability. Intuitively, claimability is the ability of signers to claim the ownership of a signature. At the same time, we should deal with security issues that arise due to this additional property. That is, a malicious party might claim the ownership of a signature that is not created by him, or frames an honest party for creating a signature that is not created by the party. We require that such events occur only with negligible probability as security properties.

Definition 13 (Claimability). *A claimable ring signature scheme* $\Pi_{CRSIG} = (\mathsf{Set}, \mathsf{KG}, \mathsf{RSig}, \mathsf{Verify}, \mathsf{Claim}, \mathsf{ClmVrf})$ *satisfies claimability if the following three conditions hold:*

(Honest signer can claim.) *For any security parameter* λ*, any* $n = \mathsf{poly}(\lambda)$*, any* m*, any* $\mathsf{pp} \leftarrow \Pi_{CRSIG}.\mathsf{Set}(1^\lambda)$*, any* $(\mathsf{pk}, \mathsf{sk}), (\mathsf{pk}_1, \mathsf{sk}_1), \cdots, (\mathsf{pk}_n, \mathsf{sk}_n) \leftarrow \Pi_{CRSIG}.\mathsf{KG}(\mathsf{pp})$*, any* $\sigma \leftarrow \Pi_{CRSIG}.\mathsf{RSig}(\mathsf{pp}, \mathsf{sk}, \{\mathsf{pk}_i\}_{i\in[n]}, \mathsf{m})$*, and any* $\pi \leftarrow \Pi_{CRSIG}.\mathsf{Claim}(\mathsf{pp}, \mathsf{sk}, \{\mathsf{pk}\} \cup \{\mathsf{pk}_i\}_{i\in[n]}, \sigma, \mathsf{m})$*, it holds that*

$$\Pi_{CRSIG}.\mathsf{ClmVrf}(\mathsf{pp}, \mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n]}, \sigma, \pi) = 1.$$

(Non-signer cannot claim.) *For any sufficiently large security parameter* λ*, and any stateful PPT adversary* $\mathcal{A}$ *that is allowed to make at most* $q = \mathsf{poly}(()\lambda)$ *queries to oracles, it holds that* $\Pr[\mathsf{ExpFlsClmRS}_{\Pi_{CRSIG},\mathcal{A}}(1^\lambda) = 1] \leq \mathsf{negl}(\lambda)$ *where the experiment* $\mathsf{ExpFlsClmRS}_{\Pi_{CRSIG},\mathcal{A}}(1^\lambda)$ *is defined as follows:*

$\mathsf{ExpFlsClmRS}_{\Pi_{CRSIG},\mathcal{A}}(1^\lambda)$

$L_{\mathsf{PK}} := \emptyset; L_{\mathsf{SK}} := \emptyset; L_{\mathsf{Sign}} := \emptyset; L_{\mathsf{Clm}} := \emptyset;$
$\mathsf{pp} \leftarrow \Pi_{CRSIG}.\mathsf{Set}(1^\lambda); (\mathsf{pk}, \mathsf{sk}) \leftarrow \Pi_{CRSIG}.\mathsf{KG}(\mathsf{pp});$
$(\mathsf{m}^*, \{\mathsf{pk}^*_i\}_{i\in[n]}) \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{PK}}, \mathsf{O}_{\mathsf{SK}}, \mathsf{O}^{\mathsf{pk},\mathsf{sk}}_{\mathsf{RSig}}, \mathsf{O}^{\mathsf{pk},\mathsf{sk}}_{\mathsf{CRSClm}}}(\mathsf{pp}, \mathsf{pk});$
$\sigma \leftarrow \Pi_{CRSIG}.\mathsf{RSig}(\mathsf{pp}, \mathsf{sk}, \{\mathsf{pk}\} \cup \{\mathsf{pk}^*_i\}_{i\in[n]}, \mathsf{m}^*);$
$(i' \in [n], \pi^*) \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{PK}}, \mathsf{O}_{\mathsf{SK}}, \mathsf{O}^{\mathsf{pk},\mathsf{sk}}_{\mathsf{RSig}}, \mathsf{O}^{\mathsf{pk},\mathsf{sk}}_{\mathsf{CRSClm}}}(\sigma);$
$b = \Pi_{CRSIG}.\mathsf{ClmVrf}(\mathsf{pp}, \mathsf{pk}^*_{i'}, \{\mathsf{pk}\} \cup \{\mathsf{pk}^*_i\}_{i\in[n]}, \sigma, \pi^*);$
$b' = \Pi_{CRSIG}.\mathsf{Verify}(\mathsf{pp}, \{\mathsf{pk}\} \cup \{\mathsf{pk}^*_i\}_{i\in[n]}, \mathsf{m}^*, \sigma)$:
output 1 if $b = 1 \wedge b' = 1 \wedge \mathsf{pk} \neq \mathsf{pk}^*_{i'}$
otherwise 0

where $n = \mathsf{poly}(\lambda)$ *s.t.* $n \leq q$ *and* O_{PK} *and* O_{SK} *are defined as in Definition 8, and* $\mathsf{O}^{\mathsf{pk},\mathsf{sk}}_{\mathsf{RSig}}$ *and* $\mathsf{O}^{\mathsf{pk},\mathsf{sk}}_{\mathsf{CRSClm}}$ *work as follows:*

$\mathsf{O}^{\mathsf{pk},\mathsf{sk}}_{\mathsf{RSig}}$ *: It works as* $\mathsf{O}_{\mathsf{RSig}}$ *when given* $(\mathsf{pk}', \{\mathsf{pk}_i\}_{i\in[n]}, \mathsf{m})$*. In addition, given* $(\{\mathsf{pk}_i\}_{i\in[n]}, \mathsf{m})$*, it returns* σ *if* $(\mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n]}, \mathsf{m}, \sigma) \in L_{\mathsf{Sign}}$*. Otherwise, it returns* $\sigma \leftarrow \Pi_{CRSIG}.\mathsf{RSig}(\mathsf{pp}, \mathsf{sk}, \{\mathsf{pk}\} \cup \{\mathsf{pk}_i\}_{i\in[n]}, \mathsf{m})$*, and updates* $L_{\mathsf{Sign}} := L_{\mathsf{Sign}} \cup \{(\mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n]}, \mathsf{m}, \sigma)\}$*.*

$\mathsf{O}^{\mathsf{pk},\mathsf{sk}}_{\mathsf{CRSClm}}$ *: It works as* $\mathsf{O}_{\mathsf{CRSClm}}$ *when given* $(\mathsf{pk}', \{\mathsf{pk}_i\}_{i\in[n]}, \sigma)$*. In addition, given* $(\{\mathsf{pk}_i\}_{i\in[n]}, \sigma)$*, it returns* π *if* $(\mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n]}, \sigma, \pi) \in L_{\mathsf{Clm}}$*. Otherwise, it returns* $\pi \leftarrow \Pi_{CRSIG}.\mathsf{Claim}(\mathsf{pp}, \mathsf{sk}, \{\mathsf{pk}\} \cup \{\mathsf{pk}_i\}_{i\in[n]}, \sigma)$*, and updates* $L_{\mathsf{Clm}} := L_{\mathsf{Clm}} \cup \{(\mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n]}, \sigma, \pi)\}$*.*

(Malicious signer cannot frame an honest party)*: For any sufficiently large security parameter* λ*, and any PPT adversary* $\mathcal{A}$*, it holds that* $\Pr[\mathsf{ExpFrmRS}_{\Pi_{CRSIG},\mathcal{A}}(1^\lambda) = 1] \leq \mathsf{negl}(\lambda)$ *where* $\mathsf{ExpFrmRS}_{\Pi_{CRSIG},\mathcal{A}}(1^\lambda)$ *is defined as follows:*

$\mathsf{ExpFrmRS}_{\Pi_{CRSIG},\mathcal{A}}(1^\lambda)$

$L_{\mathsf{PK}} := \emptyset; L_{\mathsf{SK}} := \emptyset; L_{\mathsf{Sign}} := \emptyset; L_{\mathsf{Clm}} := \emptyset;$
$\mathsf{pp} \leftarrow \mathsf{Set}(1^\lambda); (\mathsf{pk}, \mathsf{sk}) \leftarrow \Pi_{CRSIG}.\mathsf{KG}(\mathsf{pp});$
$(\mathsf{m}^*, \{\mathsf{pk}^*_i\}_{i\in[n]}, \sigma^*, \pi^*) \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{PK}}, \mathsf{O}_{\mathsf{SK}}, \mathsf{O}^{\mathsf{pk},\mathsf{sk}}_{\mathsf{RSig}}, \mathsf{O}^{\mathsf{pk},\mathsf{sk}}_{\mathsf{CRSClm}}}(\mathsf{pp}, \mathsf{pk});$
$b = \Pi_{CRSIG}.\mathsf{ClmVrf}(\mathsf{pp}, \mathsf{pk}, \{\mathsf{pk}^*_i\}_{i\in[n]}, \sigma^*);$
$b' = \Pi_{CRSIG}.\mathsf{Verify}(\mathsf{pp}, \{\mathsf{pk}\} \cup \{\mathsf{pk}^*_i\}_{i\in[n]}, \mathsf{m}^*, \sigma^*)$:
output 1 if $b = 1 \wedge b' = 1 \wedge (\cdot, \cdot, \sigma^*, \cdot) \notin L_{\mathsf{Clm}}$
otherwise 0

where $n = \mathsf{poly}(\lambda)$ *s.t.* $n \leq q$*.*

We call the second condition claim unforgeability, and the third condition non-frameability.

3 CLAIMABLE DESIGNATED VERIFIER SIGNATURES

We formalize claimable designated verifier signature (CDVS). Namely, our definition of claimability is very similar to that of in Section 2.2.

3.1 Syntax

Definition 14 (CDVS). *A claimable designated verifier signature (CDVS) scheme consists of the following eight polynomial time algorithms* (Set, SKG, VKG, DVSign, Vrf, Sim, Claim, ClmVrf)*:*

- $\mathsf{Set}(1^\lambda) \to (\mathsf{pp}, \mathsf{msk})$*: Given a security parameter* 1^λ*, it outputs a public parameter* pp *and a master secret key* msk.
- $\mathsf{SKG}(\mathsf{pp}, \mathsf{msk}) \to (\mathsf{spk}, \mathsf{ssk})$*: Given a public parameter* pp*, and a master secret key* msk*, it outputs a signer's public key* spk *and secret key* ssk.
- $\mathsf{VKG}(\mathsf{pp}, \mathsf{msk}) \to (\mathsf{vpk}, \mathsf{vsk})$*: Given a public parameter* pp*, and a master secret key* msk*, it outputs a verifier's public key* vpk *and secret key* vsk.
- $\mathsf{DVSign}(\mathsf{pp}, \mathsf{spk}, \mathsf{ssk}, \mathsf{vpk}, \mathsf{m}) \to \sigma$*: Given a public parameter* pp*, a signer's public key* spk *and secret key* ssk*, a verifier's public key* vpk*, and a message* m*, it outputs a signature* σ.
- $\mathsf{Vrf}(\mathsf{pp}, \mathsf{vpk}, \mathsf{vsk}, \mathsf{spk}, \mathsf{m}, \sigma) \to 1/0$*: Given a public parameter* pp*, a verifier's public and secret keys* vpk *and* vsk*, a signer's public key* spk*, a message* m*, and a signature* σ*, it outputs* 1 *(meaning valid) or* 0 *(meaning invalid).*
- $\mathsf{Sim}(\mathsf{pp}, \mathsf{vpk}, \mathsf{vsk}, \mathsf{spk}, \mathsf{m}) \to \sigma$*: Given a public parameter* pp*, a verifier's public key* vpk *and secret key* vsk*, a signer's public key* spk*, and a message* m*, it outputs a simulated signature* σ.
- $\mathsf{Claim}(\mathsf{pp}, \mathsf{spk}, \mathsf{ssk}, \mathsf{vpk}, \sigma) \to \pi$*: Given a public parameter* pp*, a signer's public key* spk *and secret key* ssk*, a verifier's public key* vpk*, and a signature* σ*, it outputs a claim* π.
- $\mathsf{ClmVrf}(\mathsf{pp}, \mathsf{spk}, \mathsf{vpk}, \sigma, \pi) = 1/0$*: Given a public parameter* pp*, a signer's public key* spk*, a verifier's public key* vpk*, a signature* σ*, and a claim* π*, it outputs* 1 *(meaning valid) or* 0 *(meaning invalid).*

A CDVS scheme (Set, SKG, VKG, DVSign, Vrf, Sim, Claim, ClmVrf) *satisfies correctness if for any security parameter* λ*, any* $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Set}(1^\lambda)$*, any* $(\mathsf{spk}, \mathsf{ssk}) \leftarrow \mathsf{SKG}(\mathsf{pp}, \mathsf{msk})$*, any* $(\mathsf{vpk}, \mathsf{vsk}) \leftarrow \mathsf{VKG}(\mathsf{pp}, \mathsf{msk})$*, any message* m*, and any* $\sigma \leftarrow \mathsf{DVSign}(\mathsf{pp}, \mathsf{spk}, \mathsf{ssk}, \mathsf{vpk}, \mathsf{m})$*, it holds that* $\mathsf{Vrf}(\mathsf{pp}, \mathsf{vpk}, \mathsf{vsk}, \mathsf{spk}, \mathsf{m}, \sigma) = 1$.

We note that in [11], key generation algorithms take an identifier as part of an input, just to make the ownership of keys explicit. As we consider single designated verifier setting in this paper, we do not require an identifier as an input. The key generation algorithms are separated for generality. However, it is possible that both signer's and verifier's keys are generated by the same algorithm.

3.2 Requirements

Definition 15 (EUF-CMA). *A CDVS scheme* Π_{CDVS} = (Set, SKG, VKG, DVSign, Vrf, Sim, Claim, ClmVrf) *is existentially unforgeable under an adaptive chosen-message attack (EUF-CMA) if for any sufficiently large security parameter* λ*, and any PPT adversary* $\mathcal{A}$*, it holds that* $\Pr[\mathsf{ExpEUFDVS}_{\Pi_{CDVS},\mathcal{A}}(1^\lambda) = 1] \leq \mathsf{negl}(\lambda)$ *where the experiment* $\mathsf{ExpEUFDVS}_{\Pi_{CDVS},\mathcal{A}}(1^\lambda)$ *is defined as follows:*

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

$L_{\mathsf{VPK}} := \emptyset; L_{\mathsf{SPK}} := \emptyset; L_{\mathsf{VSK}} := \emptyset; L_{\mathsf{SSK}} := \emptyset; L_{\mathsf{Sign}} := \emptyset; L_{\mathsf{Clm}} := \emptyset;$
$(\mathsf{pp}, \mathsf{msk}) \leftarrow \Pi_{CDVS}.\mathsf{Set}(1^\lambda);$
$(\mathsf{spk}^*, \mathsf{vpk}^*, \mathsf{m}^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{SPK}},\mathsf{O}_{\mathsf{SSK}},\mathsf{O}_{\mathsf{VPK}},\mathsf{O}_{\mathsf{VSK}},\mathsf{O}_{\mathsf{DVSig}},\mathsf{O}_{\mathsf{Vrf}},\mathsf{O}_{\mathsf{Clm}}}(\mathsf{pp})$:
output 1 if $((\mathsf{spk}^*, \cdot) \in L_{\mathsf{SPK}}) \wedge ((\mathsf{vpk}^*, \mathsf{vsk}^*) \in L_{\mathsf{VPK}})$
$\wedge((\mathsf{spk}^*, \cdot) \notin L_{\mathsf{SSK}}) \wedge ((\mathsf{vpk}^*, \mathsf{vsk}^*) \notin L_{\mathsf{VSK}})$
$\wedge((\mathsf{vpk}^*, \mathsf{spk}^*, \mathsf{m}^*, \sigma^*) \notin L_{\mathsf{Sign}})$
$\wedge(\Pi_{CDVS}.\mathsf{Vrf}(\mathsf{pp}, \mathsf{vpk}^*, \mathsf{vsk}^*, \mathsf{spk}^*, m^*, \sigma^*) = 1)$
otherwise 0

where $\mathsf{O}_{\mathsf{SPK}}, \mathsf{O}_{\mathsf{SSK}}, \mathsf{O}_{\mathsf{VPK}}, \mathsf{O}_{\mathsf{VSK}}, \mathsf{O}_{\mathsf{DVSig}}, \mathsf{O}_{\mathsf{Vrf}}$ *and* $\mathsf{O}_{\mathsf{Clm}}$ *work as follows:*

$\mathsf{O}_{\mathsf{SPK}}$*: It computes* $(\mathsf{spk}, \mathsf{ssk}) \leftarrow \Pi_{CDVS}.\mathsf{SKG}(\mathsf{pp}, \mathsf{msk})$*, returns* spk *and updates* $L_{\mathsf{SPK}} := L_{\mathsf{SPK}} \cup \{(\mathsf{spk}, \mathsf{ssk})\}$.
$\mathsf{O}_{\mathsf{SSK}}$*: Given* spk*, if* $(\mathsf{spk}, \mathsf{ssk}) \in L_{\mathsf{SPK}}$*, then it returns* ssk *and updates* $L_{\mathsf{SSK}} := L_{\mathsf{SSK}} \cup \{(\mathsf{spk}, \mathsf{ssk})\}$*. Otherwise, return* $\perp$*. Note that we regard the signer corresponding to* $(\mathsf{spk}, \mathsf{ssk}) \in L_{\mathsf{SSK}}$ *as a corrupted one.*
$\mathsf{O}_{\mathsf{VPK}}$*: It computes* $(\mathsf{vpk}, \mathsf{vsk}) \leftarrow \Pi_{CDVS}.\mathsf{VKG}(\mathsf{pp}, \mathsf{msk})$*, returns* vpk*, and updates* $L_{\mathsf{VPK}} := L_{\mathsf{VPK}} \cup \{(\mathsf{vpk}, \mathsf{vsk})\}$.
$\mathsf{O}_{\mathsf{VSK}}$*: Given* vpk*, if* $(\mathsf{vpk}, \mathsf{vsk}) \in L_{\mathsf{VPK}}$*, then it returns* vsk*, and updates* $L_{\mathsf{VSK}} := L_{\mathsf{VSK}} \cup \{(\mathsf{vpk}, \mathsf{vsk})\}$*. Otherwise, return* $\perp$*. Note that we regard the verifier corresponding to* $(\mathsf{vpk}, \mathsf{vsk}) \in L_{\mathsf{VSK}}$ *as a corrupted one.*
$\mathsf{O}_{\mathsf{DVSig}}$*: Given* vpk, spk*, and* m*, it does the followings:*
- *If* $(\mathsf{vpk}, \cdot) \notin L_{\mathsf{VPK}}$ *or* $(\mathsf{spk}, \mathsf{ssk}) \notin L_{\mathsf{SPK}}$*, then returns* $\perp$.
- *If* $(\mathsf{vpk}, \mathsf{spk}, \mathsf{m}, \sigma) \in L_{\mathsf{Sign}}$*, then returns* σ.
- *Returns* $\sigma \leftarrow \Pi_{CDVS}.\mathsf{DVSign}(\mathsf{pp}, \mathsf{spk}, \mathsf{ssk}, \mathsf{vpk}, \mathsf{m})$*, and update* $L_{\mathsf{Sign}} := L_{\mathsf{Sign}} \cup \{(\mathsf{vpk}, \mathsf{spk}, \mathsf{m}, \sigma)\}$.

$\mathsf{O}_{\mathsf{Vrf}}$*: Given* vpk, spk, m *and* σ*, it does the followings:*
- *If* $(\mathsf{vpk}, \cdot) \notin L_{\mathsf{VPK}}$ *or* $(\mathsf{spk}, \cdot) \notin L_{\mathsf{SPK}}$*, then returns* $\perp$.
- *Returns* $b = \Pi_{CDVS}.\mathsf{Vrf}(\mathsf{pp}, \mathsf{vpk}, \mathsf{vsk}, \mathsf{spk}, \mathsf{m}, \sigma)$.

$\mathsf{O}_{\mathsf{Clm}}$*: Given* vpk, spk*, and* σ*, it does the followings:*
- *If* $(\mathsf{vpk}, \cdot) \notin L_{\mathsf{VPK}}$, $(\mathsf{spk}, \mathsf{ssk}) \notin L_{\mathsf{SPK}}$*, or* $(\mathsf{vpk}, \mathsf{spk}, \cdot, \sigma) \notin L_{\mathsf{Sign}}$ *then returns* $\perp$.
- *If* $(\mathsf{vpk}, \mathsf{spk}, \sigma, \pi) \in L_{\mathsf{Clm}}$*, then returns* π.
- *Returns* $\pi \leftarrow \Pi_{CDVS}.\mathsf{Claim}(\mathsf{pp}, \mathsf{spk}, \mathsf{ssk}, \mathsf{vpk}, \sigma)$*, and updates* $L_{\mathsf{Clm}} := L_{\mathsf{Clm}} \cup \{(\mathsf{vpk}, \mathsf{spk}, \sigma, \pi)\}$.

OTR is a fundamental security requirement for DVS, which guarantees that a designated verifier can simulate a signature. Namely, a non-designated verifier might be able to verify a signature, but it is useless thanks to OTR property.

Definition 16 (OTR). *A CDVS scheme* Π_{CDVS} = (Set, SKG, VKG, DVSign, Vrf, Sim, Claim, ClmVrf) *is off-the-record (OTR) if for any security parameter* λ*, and a stateful PPT adversary* $\mathcal{A}$*, it holds that* $|\Pr[\mathsf{ExpOTR}_{\Pi_{CDVS},\mathcal{A}}(1^\lambda) = 1] - 1/2| \leq \mathsf{negl}(\lambda)$ *where the experiment*

$\mathsf{ExpOTR}_{\Pi_{CDVS},\mathcal{A}}(1^\lambda)$ *is defined as follows:*

$\mathsf{ExpOTR}_{\Pi_{CDVS},\mathcal{A}}(1^\lambda)$

$L_{\mathsf{VPK}} := \emptyset; L_{\mathsf{SPK}} := \emptyset; L_{\mathsf{VSK}} := \emptyset; L_{\mathsf{SSK}} := \emptyset; L_{\mathsf{Sign}} := \emptyset; L_{\mathsf{Clm}} := \emptyset;$
$(\mathsf{pp}, \mathsf{msk}) \leftarrow \Pi_{CDVS}.\mathsf{Set}(1^\lambda);$
$(\mathsf{vpk}^*, \mathsf{spk}^*, \mathsf{m}^*) \leftarrow \mathcal{A}^{\mathsf{O_{SPK}},\mathsf{O_{SSK}},\mathsf{O_{VPK}},\mathsf{O_{VSK}},\mathsf{O_{DVSig}},\mathsf{O_{Vrf}},\mathsf{O_{Clm}}}(\mathsf{pp});$
return $\perp$ if $(\mathsf{vpk}^*, \mathsf{vsk}^*) \notin L_{\mathsf{VPK}} \vee (\mathsf{spk}^*, \mathsf{ssk}^*) \notin L_{\mathsf{SPK}}$;
$\sigma_0 \leftarrow \Pi_{CDVS}.\mathsf{DVSign}(\mathsf{pp}, \mathsf{spk}^*, \mathsf{ssk}^*, \mathsf{vpk}^*, \mathsf{m}^*);$
$\sigma_1 \leftarrow \Pi_{CDVS}.\mathsf{Sim}(\mathsf{pp}, \mathsf{vpk}^*, \mathsf{vsk}^*, \mathsf{spk}^*, \mathsf{m}^*); b \leftarrow \{0,1\};$
$b' \leftarrow \mathcal{A}^{\mathsf{O_{SPK}},\mathsf{O_{SSK}},\mathsf{O_{VPK}},\mathsf{O_{VSK}},\mathsf{O_{DVSig}},\mathsf{O_{Vrf}},\mathsf{O_{Clm}}}(\sigma_b)$:
output 1 if
$(b' = b) \wedge ((\mathsf{vpk}^*, \mathsf{vsk}^*) \notin L_{\mathsf{VSK}}) \wedge ((\mathsf{spk}^*, \mathsf{ssk}^*) \notin L_{\mathsf{SSK}})$
$\wedge((\cdot,\cdot,\cdot,\sigma_b) \notin L_{\mathsf{Vrf}}) \wedge ((\cdot,\cdot,\sigma_b,\cdot) \notin L_{\mathsf{Clm}})$
otherwise 0

where oracles are defines as in Definition 15.

In what follows, we introduce claimability of CDVS, by following [25]. That is, similar to anonymity in claimable ring signature, we require the following conditions:

- A genuine signer can claim the ownership of a signature.
- Non-signer cannot claim the ownership of a signature.
- No one is able to frame other parties as a signer.

Definition 17 (Claimability). *A CDVS scheme* Π_{CDVS} = (Set, SKG, VKG, DVSign, Vrf, Sim, Claim, ClmVrf) *satisfies claimability if the following three conditions hold:*

(Honest signer can claim.) *For any security parameter* λ*, any* $n = \mathsf{poly}(\lambda)$*, any* m*, any* $(\mathsf{pp}, \mathsf{msk}) \leftarrow \Pi_{CDVS}.\mathsf{Set}(1^\lambda)$*, any* $(\mathsf{spk}, \mathsf{ssk}) \leftarrow \Pi_{CDVS}.\mathsf{SKG}(\mathsf{pp}, \mathsf{msk})$*, any* $(\mathsf{vpk}, \mathsf{vsk}) \leftarrow \Pi_{CDVS}.\mathsf{VKG}(\mathsf{pp}, \mathsf{msk})$*, any* $\sigma \leftarrow \Pi_{CDVS}.\mathsf{DVSign}(\mathsf{pp}, \mathsf{spk}, \mathsf{ssk}, \mathsf{vpk}, \mathsf{m})$*, and any* $\pi \leftarrow \Pi_{CDVS}.\mathsf{Claim}(\mathsf{pp}, \mathsf{spk}, \mathsf{ssk}, \mathsf{vpk}, \sigma, \mathsf{m})$,

$$\Pi_{CDVS}.\mathsf{ClmVrf}(\mathsf{pp}, \mathsf{spk}, \mathsf{vpk}, \sigma, \pi) = 1.$$

(Non-signer cannot claim.) *For any sufficiently large security parameter* λ*, and any stateful PPT adversary* $\mathcal{A}$*, it holds that* $\Pr[\mathsf{ExpFlsClmDVS}_{\Pi_{CDVS},\mathcal{A}}(1^\lambda) = 1] \leq \mathsf{negl}(\lambda)$ *where the experiment* $\mathsf{ExpFlsClmDVS}_{\Pi_{CDVS},\mathcal{A}}(1^\lambda)$ *is defined as follows:*

$\mathsf{ExpFlsClmDVS}_{\Pi_{CDVS},\mathcal{A}}(1^\lambda)$

$L_{\mathsf{VPK}} := \emptyset; L_{\mathsf{SPK}} := \emptyset; L_{\mathsf{VSK}} := \emptyset; L_{\mathsf{SSK}} := \emptyset; L_{\mathsf{Sign}} := \emptyset; L_{\mathsf{Clm}} := \emptyset;$
$(\mathsf{pp}, \mathsf{msk}) \leftarrow \Pi_{CDVS}.\mathsf{Set}(1^\lambda); (\mathsf{spk}, \mathsf{ssk}) \leftarrow \Pi_{CDVS}.\mathsf{SKG}(\mathsf{pp});$
$(\mathsf{m}^*, \mathsf{vpk}^*, \mathsf{vsk}^*) \leftarrow$
$\mathcal{A}^{\mathsf{O_{SPK}},\mathsf{O_{SSK}},\mathsf{O_{VPK}},\mathsf{O_{VSK}},\mathsf{O}^{\mathsf{spk,ssk}}_{\mathsf{DVSig}},\mathsf{O_{Vrf}},\mathsf{O}^{\mathsf{spk,ssk}}_{\mathsf{Clm}}}(\mathsf{pp}, \mathsf{spk});$
$\sigma \leftarrow \Pi_{CDVS}.\mathsf{DVSign}(\mathsf{pp}, \mathsf{spk}, \mathsf{ssk}, \mathsf{vpk}^*, \mathsf{m}^*);$
$\pi^* \leftarrow \mathcal{A}^{\mathsf{O_{SPK}},\mathsf{O_{SSK}},\mathsf{O_{VPK}},\mathsf{O_{VSK}},\mathsf{O}^{\mathsf{spk,ssk}}_{\mathsf{DVSig}},\mathsf{O_{Vrf}},\mathsf{O}^{\mathsf{spk,ssk}}_{\mathsf{Clm}}}(\sigma);$
$b = \Pi_{CDVS}.\mathsf{ClmVrf}(\mathsf{pp}, \mathsf{vpk}^*, \mathsf{spk}, \sigma, \pi^*);$
$b' = \Pi_{CDVS}.\mathsf{Verify}(\mathsf{pp}, \mathsf{vpk}^*, \mathsf{vsk}^*, \mathsf{spk}, \mathsf{m}^*, \sigma)$:
output 1 if $b = 1 \wedge b' = 1 \wedge \mathsf{vpk}^* \neq \mathsf{spk}$
otherwise 0

where $\mathsf{O_{SPK}}, \mathsf{O_{SSK}}, \mathsf{O_{VPK}}, \mathsf{O_{VSK}}$ *and* $\mathsf{O_{Vrf}}$ *are defined as in Definition 15, and others are defined as follows:*

$\mathsf{O}^{\mathsf{spk,ssk}}_{\mathsf{DVSig}}$: *It works as* $\mathsf{O_{DVSig}}$ *when given* $(\mathsf{vpk}', \mathsf{spk}', \mathsf{m})$. *In addition, given* $(\mathsf{vpk}', \mathsf{m})$*, it returns* σ *if* $(\mathsf{vpk}', \mathsf{spk}, \mathsf{m}, \sigma) \in L_{\mathsf{Sign}}$. *Otherwise, it returns* $\sigma \leftarrow \Pi_{CDVS}.\mathsf{DVSign}(\mathsf{pp}, \mathsf{spk}, \mathsf{ssk}, \mathsf{vpk}', \mathsf{m})$*, and updates* $L_{\mathsf{Sign}} := L_{\mathsf{Sign}} \cup \{(\mathsf{vpk}', \mathsf{spk}, \mathsf{m}, \sigma)\}$.

$\mathsf{O}^{\mathsf{spk,ssk}}_{\mathsf{Clm}}$: *It works as* $\mathsf{O_{Clm}}$ *when given* $(\mathsf{vpk}', \mathsf{spk}', \sigma)$. *In addition, given* (vpk', σ)*, it returns* π *if* $(\mathsf{vpk}', \mathsf{spk}, \sigma, \pi) \in L_{\mathsf{Clm}}$. *Otherwise, it returns* $\pi \leftarrow \Pi_{CDVS}.\mathsf{Claim}(\mathsf{pp}, \mathsf{spk}, \mathsf{ssk}, \mathsf{vpk}', \sigma)$ *and updates* $L_{\mathsf{Clm}} := L_{\mathsf{Clm}} \cup \{(\mathsf{vpk}', \mathsf{spk}, \sigma, \pi)\}$.

(Malicious signer cannot frame an honest party)*: For any sufficiently large security parameter* λ*, and any PPT adversary* $\mathcal{A}$*, it holds that* $\Pr[\mathsf{ExpFrm}_{\Pi_{CDVS},\mathcal{A}}(1^\lambda) = 1] \leq \mathsf{negl}(\lambda)$ *where* $\mathsf{ExpFrm}_{\Pi_{CDVS},\mathcal{A}}(1^\lambda)$ *is defined as follows:*

$\mathsf{ExpFrm}_{\Pi_{CDVS},\mathcal{A}}(1^\lambda)$

$L_{\mathsf{VPK}} := \emptyset; L_{\mathsf{SPK}} := \emptyset; L_{\mathsf{VSK}} := \emptyset; L_{\mathsf{SSK}} := \emptyset; L_{\mathsf{Sign}} := \emptyset; L_{\mathsf{Clm}} := \emptyset;$
$(\mathsf{pp}, \mathsf{msk}) \leftarrow \Pi_{CDVS}.\mathsf{Set}(1^\lambda); (\mathsf{spk}, \mathsf{ssk}) \leftarrow \Pi_{CDVS}.\mathsf{SKG}(\mathsf{pp}, \mathsf{msk});$
$(\mathsf{m}^*, \mathsf{vpk}^*, \mathsf{vsk}^*, \sigma^*, \pi^*) \leftarrow$
$\mathcal{A}^{\mathsf{O_{SPK}},\mathsf{O_{SSK}},\mathsf{O_{VPK}},\mathsf{O_{VSK}},\mathsf{O}^{\mathsf{spk,ssk}}_{\mathsf{DVSig}},\mathsf{O_{Vrf}},\mathsf{O}^{\mathsf{spk,ssk}}_{\mathsf{Clm}}}(\mathsf{pp}, \mathsf{spk});$
$b = \Pi_{CDVS}.\mathsf{ClmVrf}(\mathsf{pp}, \mathsf{spk}, \mathsf{vpk}^*, \sigma^*, \pi^*);$
$b' = \Pi_{CDVS}.\mathsf{Verify}(\mathsf{pp}, \mathsf{vpk}^*, \mathsf{vsk}^*, \mathsf{spk}, \mathsf{m}^*, \sigma^*)$:
output 1 if $b = 1 \wedge b' = 1 \wedge (\cdot,\cdot,\sigma^*,\cdot) \notin L_{\mathsf{Clm}}$
otherwise 0

We call the second condition as claim unforgeability, and the third condition as non-frameability.

4 THE FIRST CONSTRUCTION

We provide a generic construction of CDVS from ring signature, (non-ring) signature, PRF, and commitment scheme. We remark that our construction and its proof are based on those of in [25].

4.1 Construction

Let Π_{RS} = (Set, KG, RSig, Verify) be a ring signature scheme, Σ = (KG, Sig, Verify) a signature scheme, PRF = (KG, Eval) a PRF, and Π_{COM} = (Com, Open) a commitment scheme. For simplicity, we assume without loss of generality that the random coins in Π_{COM}.Com and Σ.Sig and the output of PRF.Eval are the same lengths. We demonstrate our construction Π_{CDVS} = (Set, SKG, VKG, DVSign, Vrf, Sim, Claim, ClmVrf) of a CDVS scheme from them.

We remark that our CDVS scheme does not exactly follow the syntax of CDVS given by Definition 14 as follows:

- It does not require a master secret key.
- The verification algorithm does not take a secret key of a designated verifier.

Note that the syntax given by Definition 14 is just a general form. Furthermore, even if the verification algorithm does not take a designated verifier's secret key, our scheme is still secure. That is, thanks to OTR property, it is still useless for non-designated verifiers to verify a signature.

We overview the signing algorithm and the claiming algorithm before the formal descriptions. The signing algorithm outputs a signature σ_{RS} generated by Π_{RS}.RSig and a commitment c generated by Π_{COM}.Com. Namely, c commits to a signature σ_Σ generated by using the signer's secret key. Thus, it is sufficient for claiming the ownership of σ to open σ_Σ and the randomness that is used to compute c, because verifiers can check the validity of the claim by verifying σ_Σ through the signer's public key pk_Σ. The formal description is as follows:

$\Pi_{\mathrm{CDVS}}.\mathsf{Set}(1^\lambda)$: Given a security parameter 1^λ, it outputs $\mathsf{pp} \leftarrow \Pi_{\mathrm{RS}}.\mathsf{Set}(1^\lambda)$, and $\mathsf{msk} := \phi$. (In what follows, we omit msk from interfaces for readability.)

$\Pi_{\mathrm{CDVS}}.\mathsf{SKG}(\mathsf{pp})$: Given a public parameter pp, it computes $k^{(s)}_{\mathsf{PRF}} \leftarrow \mathsf{PRF.KG}(1^\lambda)$, $(\mathsf{pk}^{(s)}_\Sigma, \mathsf{sk}^{(s)}_\Sigma) \leftarrow \Sigma.\mathsf{KG}(1^\lambda)$, and $(\mathsf{pk}^{(s)}_{\mathrm{RS}}, \mathsf{sk}^{(s)}_{\mathrm{RS}}) \leftarrow \Pi_{\mathrm{RS}}.\mathsf{KG}(\mathsf{pp})$, and outputs $\mathsf{spk} := (\mathsf{pk}^{(s)}_\Sigma, \mathsf{pk}^{(s)}_{\mathrm{RS}})$ and $\mathsf{ssk} := (\mathsf{pk}^{(s)}_\Sigma, \mathsf{pk}^{(s)}_{\mathrm{RS}}, k^{(s)}_{\mathsf{PRF}}, \mathsf{sk}^{(s)}_\Sigma, \mathsf{sk}^{(s)}_{\mathrm{RS}})$. Note that "$(s)$" stands for a signer's identity.

$\Pi_{\mathrm{CDVS}}.\mathsf{VKG}(\mathsf{pp})$: (The same as $\Pi_{\mathrm{CDVS}}.\mathsf{SKG}$.) Given a public parameter pp, it computes $k^{(v)}_{\mathsf{PRF}} \leftarrow \mathsf{PRF.KG}(1^\lambda)$, $(\mathsf{pk}^{(v)}_\Sigma, \mathsf{sk}^{(v)}_\Sigma) \leftarrow \Sigma.\mathsf{KG}(1^\lambda)$, and $(\mathsf{pk}^{(v)}_{\mathrm{RS}}, \mathsf{sk}^{(v)}_{\mathrm{RS}}) \leftarrow \Pi_{\mathrm{RS}}.\mathsf{KG}(\mathsf{pp})$, and outputs $\mathsf{vpk} := (\mathsf{pk}^{(v)}_\Sigma, \mathsf{pk}^{(v)}_{\mathrm{RS}})$ and $\mathsf{vsk} := (\mathsf{pk}^{(v)}_\Sigma, \mathsf{pk}^{(v)}_{\mathrm{RS}}, k^{(v)}_{\mathsf{PRF}}, \mathsf{sk}^{(v)}_\Sigma, \mathsf{sk}^{(v)}_{\mathrm{RS}})$. Note that "$(v)$" stands for a verifier's identity.

$\Pi_{\mathrm{CDVS}}.\mathsf{DVSign}(\mathsf{pp}, \mathsf{spk}, \mathsf{ssk}, \mathsf{vpk}, \mathsf{m})$: Given a public parameter pp, a signer's public key $\mathsf{spk} = (\mathsf{pk}^{(s)}_\Sigma, \mathsf{pk}^{(s)}_{\mathrm{RS}}, k^{(s)}_{\mathsf{PRF}}, \mathsf{sk}^{(s)}_\Sigma, \mathsf{sk}^{(s)}_{\mathrm{RS}})$, a verifier's public key $\mathsf{vpk} = (\mathsf{pk}^{(v)}_\Sigma, \mathsf{pk}^{(v)}_{\mathrm{RS}})$, and a message m, it first computes the followings:

(1) $\sigma_{\mathrm{RS}} \leftarrow \Pi_{\mathrm{RS}}(\mathsf{pp}, \mathsf{sk}^{(s)}_{\mathrm{RS}}, \{\mathsf{pk}^{(s)}_{\mathrm{RS}}\} \cup \{\mathsf{pk}^{(v)}_{\mathrm{RS}}\}, \mathsf{m})$
(2) $r_\Sigma = \mathsf{PRF.Eval}(k^{(s)}_{\mathsf{PRF}}, (\mathsf{spk}, \sigma_{\mathrm{RS}}, 0))$
(3) $\sigma_\Sigma = \Sigma.\mathsf{Sig}(\mathsf{sk}^{(s)}_\Sigma, (\mathsf{spk}, \sigma_{\mathrm{RS}}); r_\Sigma)$
(4) $r_{\mathsf{Com}} = \mathsf{PRF.Eval}(k^{(s)}_{\mathsf{PRF}}, (\mathsf{spk}, \sigma_{\mathrm{RS}}, 1))$
(5) $c = \Pi_{\mathrm{COM}}.\mathsf{Com}((\mathsf{spk}, \sigma_\Sigma); r_{\mathsf{Com}})$

If it holds that $\Pi_{\mathrm{CDVS}}.\mathsf{ClmVrf}(\mathsf{pp}, \mathsf{spk}, \mathsf{vpk}, \sigma, \pi) = 1$ where $\sigma = (\sigma_{\mathrm{RS}}, c)$ and $\pi = \Pi_{\mathrm{CDVS}}.\mathsf{Claim}(\mathsf{pp}, \mathsf{spk}, \mathsf{ssk}, \mathsf{vpk}, \sigma)$, then returns σ, otherwise $\perp$. Note that the condition on π is just needed for proving claimability, and thus it does not play any essential role in terms of signing functionality.

$\Pi_{\mathrm{CDVS}}.\mathsf{Verify}(\mathsf{pp}, \mathsf{vpk}, \mathsf{spk}, \mathsf{m}, \sigma)$: Given a public parameter pp, a verifier's pubic key $\mathsf{vpk} = (\mathsf{pk}^{(v)}_\Sigma, \mathsf{pk}^{(v)}_{\mathrm{RS}})$, a signer's public key $\mathsf{spk} = (\mathsf{pk}^{(s)}_\Sigma, \mathsf{pk}^{(s)}_{\mathrm{RS}})$ a message m, and a signature $\sigma = (\sigma_{\mathrm{RS}}, c)$, it outputs $b = \Pi_{\mathrm{RS}}.\mathsf{Verify}(\mathsf{pp}, \{\mathsf{pk}^{(s)}_{\mathrm{RS}}\} \cup \{\mathsf{pk}^{(v)}_{\mathrm{RS}}\}, \mathsf{m}, \sigma_{\mathrm{RS}})$.

$\Pi_{\mathrm{CDVS}}.\mathsf{Sim}(\mathsf{pp}, \mathsf{vpk}, \mathsf{vsk}, \mathsf{spk}, \mathsf{m})$: Given a public parameter pp, a verifier's pubic and secret keys $\mathsf{vpk} = (\mathsf{pk}^{(v)}_\Sigma, \mathsf{pk}^{(v)}_{\mathrm{RS}})$ and $\mathsf{vsk} = (\mathsf{pk}^{(v)}_\Sigma, \mathsf{pk}^{(v)}_{\mathrm{RS}}, k^{(v)}_{\mathsf{PRF}}, \mathsf{sk}^{(v)}_\Sigma, \mathsf{sk}^{(v)}_{\mathrm{RS}})$, a signer's public key $\mathsf{spk} = (\mathsf{pk}^{(s)}_\Sigma, \mathsf{pk}^{(s)}_{\mathrm{RS}})$, and a message m, it does the same as $\Pi_{\mathrm{CDVS}}.\mathsf{DVSign}$ by using the verifier's secret key:

(1) $\sigma_{\mathrm{RS}} \leftarrow \Pi_{\mathrm{RS}}(\mathsf{pp}, \mathsf{sk}^{(v)}_{\mathrm{RS}}, \{\mathsf{pk}^{(s)}_{\mathrm{RS}}\} \cup \{\mathsf{pk}^{(v)}_{\mathrm{RS}}\}, \mathsf{m})$
(2) $r_\Sigma = \mathsf{PRF.Eval}(k^{(v)}_{\mathsf{PRF}}, (\mathsf{spk}, \sigma_{\mathrm{RS}}, 0))$
(3) $\sigma_\Sigma = \Sigma.\mathsf{Sig}(\mathsf{sk}^{(v)}_\Sigma, (\mathsf{spk}, \sigma_{\mathrm{RS}}); r_\Sigma)$
(4) $r_{\mathsf{Com}} = \mathsf{PRF.Eval}(k^{(v)}_{\mathsf{PRF}}, (\mathsf{spk}, \sigma_{\mathrm{RS}}, 1))$
(5) $c = \Pi_{\mathrm{COM}}.\mathsf{Com}((\mathsf{spk}, \sigma_\Sigma); r_{\mathsf{Com}})$

If it holds that $\Pi_{\mathrm{CDVS}}.\mathsf{ClmVrf}(\mathsf{pp}, \mathsf{spk}, \mathsf{vpk}, \sigma, \pi) = 1$ where $\sigma = (\sigma_{\mathrm{RS}}, c)$ and $\pi = \Pi_{\mathrm{CDVS}}.\mathsf{Claim}(\mathsf{pp}, \mathsf{spk}, \mathsf{ssk}, \mathsf{vpk}, \sigma)$, then returns σ, otherwise $\perp$.

$\Pi_{\mathrm{CDVS}}.\mathsf{Claim}(\mathsf{pp}, \mathsf{spk}, \mathsf{ssk}, \mathsf{vpk}, \sigma)$: Given a public parameter pp, a signer's public and secret keys spk and $\mathsf{ssk} = (\mathsf{pk}^{(s)}_\Sigma, \mathsf{pk}^{(s)}_{\mathrm{RS}}, k^{(s)}_{\mathsf{PRF}}, \mathsf{sk}^{(s)}_\Sigma, \mathsf{sk}^{(s)}_{\mathrm{RS}})$, a verifier's public key vpk, and a signature $\sigma = (\sigma_{\mathrm{RS}}, c)$, it computes $r_\Sigma = \mathsf{PRF.Eval}(k^{(s)}_{\mathsf{PRF}}, (\mathsf{spk}, \sigma_{\mathrm{RS}}, 0))$, $r_{\mathsf{Com}} = \mathsf{PRF.Eval}(k^{(s)}_{\mathsf{PRF}}, (\mathsf{spk}, \sigma_{\mathrm{RS}}, 1))$, and $\sigma_\Sigma = \Sigma.\mathsf{Sig}(\mathsf{sk}^{(s)}_\Sigma, (\mathsf{spk}, \sigma_{\mathrm{RS}}); r_\Sigma)$. If $c \neq \Pi_{\mathrm{COM}}.\mathsf{Com}((\mathsf{spk}, \sigma_\Sigma); r_{\mathsf{Com}})$, then outputs $\perp$, otherwise returns $\pi = (r_{\mathsf{Com}}, \sigma_\Sigma)$.

$\Pi_{\mathrm{CDVS}}.\mathsf{ClmVrf}(\mathsf{pp}, \mathsf{spk}, \mathsf{vpk}, \sigma, \pi)$: Given a public parameter pp, a signer's public key $\mathsf{spk} = (\mathsf{pk}^{(s)}_\Sigma, \mathsf{pk}^{(s)}_{\mathrm{RS}})$, a verifier's public key $\mathsf{vpk} := (\mathsf{pk}^{(v)}_\Sigma, \mathsf{pk}^{(v)}_{\mathrm{RS}})$, a signature $\sigma = (\sigma_{\mathrm{RS}}, c)$, and a claim $\pi = (r_{\mathsf{Com}}, \sigma_\Sigma)$, it first computes $c' = \Pi_{\mathrm{COM}}.\mathsf{Com}((\mathsf{spk}, \sigma_\Sigma); r_{\mathsf{Com}})$. Then, it returns 1 if $c' = c$ and $\Sigma.\mathsf{Verify}(\mathsf{pk}^{(s)}_\Sigma, (\mathsf{spk}, \sigma_{\mathrm{RS}}), \sigma_\Sigma) = 1$, otherwise 0.

4.2 Security Proof

We prove correctness, EUF-CMA, OTR, and claimability of Π_{CDVS}. Correctness is immediate. Thus, we focus on the remaining properties. We remark that the proofs of EUF-CMA and claimability are similar to those of in [25], but OTR is totally new.

Lemma 1. *The CDVS scheme Π_{CDVS} satisfies EUF-CMA if the underlying ring signature scheme Π_{RS} satisfies EUF-CMA.*

Proof. Suppose for contradiction that there is a PPT adversary $\mathcal{A}$ that breaks EUF-CMA of Π_{CDVS} with non-negligible probability. We provide another PPT adversary $\mathcal{A}'$ that violates EUF-CMA of Π_{RS} with non-negligible probability by simulating $\mathcal{A}$ inside. We describe how $\mathcal{A}'$ works in $\mathsf{ExpEUFRS}_{\Pi_{\mathrm{RS}},\mathcal{A}'}(1^\lambda)$ by simulating the experiment $\mathsf{ExpEUFDVS}_{\Pi_{\mathrm{CDVS}},\mathcal{A}}(1^\lambda)$.

Setup phase. Given a public parameter pp, the adversary $\mathcal{A}'$ sets $L'_{\mathsf{VPK}}, L'_{\mathsf{SPK}}, L'_{\mathsf{VSK}}, L'_{\mathsf{SSK}}, L'_{\mathsf{Sign}}$, and L'_{Clm} as $\emptyset$. Then $\mathcal{A}'$ gives pp and $\mathsf{msk} := \emptyset$ to $\mathcal{A}$. Given a query from $\mathcal{A}$, the adversary $\mathcal{A}'$ simulates the answer as follows (without loss of generality, we assume that $\mathcal{A}$ does not make queries to $\mathsf{O}_{\mathsf{Vrf}}$, because verification can be done only by public keys):

$\mathsf{O}_{\mathsf{SPK}}$: Given pp, it computes $k^{(s)}_{\mathsf{PRF}} \leftarrow \mathsf{PRF.KG}(1^\lambda)$, and $(\mathsf{pk}^{(s)}_\Sigma, \mathsf{sk}^{(s)}_\Sigma) \leftarrow \Sigma.\mathsf{KG}(1^\lambda)$. It asks the challenger of $\mathsf{ExpEUFRS}_{\Pi_{\mathrm{RS}},\mathcal{A}'}$ to call O_{PK} on pp to receive $\mathsf{pk}^{(s)}_{\mathrm{RS}}$. It sets $\mathsf{spk} := (\mathsf{pk}^{(s)}_\Sigma, \mathsf{pk}^{(s)}_{\mathrm{RS}})$, and $\mathsf{ssk} := (\mathsf{pk}^{(s)}_\Sigma, \mathsf{pk}^{(s)}_{\mathrm{RS}}, k^{(s)}_{\mathsf{PRF}}, \mathsf{sk}^{(s)}_\Sigma, \phi)$, returns spk to $\mathcal{A}$, and updates $L_{\mathsf{SPK}} := L_{\mathsf{SPK}} \cup \{(\mathsf{spk}, \mathsf{ssk})\}$.

$\mathsf{O}_{\mathsf{SSK}}$: Given $\mathsf{spk} = (\mathsf{pk}^{(s)}_\Sigma, \mathsf{pk}^{(s)}_{\mathrm{RS}})$, it returns $\perp$ if $(\mathsf{spk}, \mathsf{ssk}) \notin L'_{\mathsf{SPK}}$. Otherwise, it asks the challenger of $\mathsf{ExpEUFRS}_{\Pi_{\mathrm{RS}},\mathcal{A}'}$ to call O_{SK} on $\mathsf{pk}^{(s)}_{\mathrm{RS}}$ to receive $\mathsf{sk}^{(s)}_{\mathrm{RS}}$. It updates $\mathsf{ssk} := (\mathsf{pk}^{(s)}_\Sigma, \mathsf{pk}^{(s)}_{\mathrm{RS}}, k^{(s)}_{\mathsf{PRF}}, \mathsf{sk}^{(s)}_\Sigma, \mathsf{sk}^{(s)}_{\mathrm{RS}})$ and $L_{\mathsf{SSK}} := L_{\mathsf{SSK}} \cup \{(\mathsf{spk}, \mathsf{ssk})\}$, and returns ssk.

$\mathsf{O}_{\mathsf{VPK}}$: The same as $\mathsf{O}_{\mathsf{SPK}}$. It returns $\mathsf{vpk} := (\mathsf{pk}^{(v)}_\Sigma, \mathsf{pk}^{(v)}_{\mathrm{RS}})$, and updates $L_{\mathsf{VPK}} := L_{\mathsf{VPK}} \cup \{(\mathsf{vpk}, \mathsf{vsk})\}$.

$\mathsf{O}_{\mathsf{VSK}}$: The same as $\mathsf{O}_{\mathsf{SSK}}$. It returns $\mathsf{vsk} := (\mathsf{pk}^{(v)}_\Sigma, \mathsf{pk}^{(v)}_{\mathrm{RS}}, k^{(v)}_{\mathsf{PRF}}, \mathsf{sk}^{(v)}_\Sigma, \mathsf{sk}^{(v)}_{\mathrm{RS}})$, and updates $L_{\mathsf{VSK}} := L_{\mathsf{VSK}} \cup \{(\mathsf{vpk}, \mathsf{vsk})\}$.

$\mathsf{O}_{\mathsf{DVSig}}$: Given $\mathsf{spk} = (\mathsf{pk}^{(s)}_\Sigma, \mathsf{pk}^{(s)}_{\mathrm{RS}})$, $\mathsf{vpk} = (\mathsf{pk}^{(v)}_\Sigma, \mathsf{pk}^{(v)}_{\mathrm{RS}})$, and m, it returns $\perp$ if $(\mathsf{vpk}, \cdot) \notin L'_{\mathsf{VPK}}$ or $(\mathsf{spk}, \mathsf{ssk}) \notin L'_{\mathsf{SPK}}$. Otherwise, $\mathcal{A}'$ does the followings (note that $\mathcal{A}'$ knows $\mathsf{ssk} = (\mathsf{pk}^{(s)}_\Sigma, \mathsf{pk}^{(s)}_{\mathrm{RS}}, k^{(s)}_{\mathsf{PRF}}, \mathsf{sk}^{(s)}_\Sigma, \phi)$ as $(\mathsf{spk}, \mathsf{ssk}) \in L'_{\mathsf{SPK}}$):

(1) If $\{(\mathsf{vpk}, \mathsf{spk}, \mathsf{m}, \sigma)\} \in L_{\mathsf{Sign}}$, then returns σ.
(2) Asks the challenger of $\mathsf{ExpEUFRS}_{\Pi_{\mathrm{RS}},\mathcal{A}'}(1^\lambda)$ to call $\mathsf{O}_{\mathsf{RSig}}$ on spk, vpk and m to receive σ_{RS}
(3) $r_\Sigma = \mathsf{PRF.Eval}(k^{(s)}_{\mathsf{PRF}}, (\mathsf{spk}, \sigma_{\mathrm{RS}}, 0))$

(4) $\sigma_\Sigma = \Sigma.\mathsf{Sig}(\mathsf{sk}_\Sigma^{(s)}, (\mathsf{spk}, \sigma_{\mathrm{RS}}); r_\Sigma)$
(5) $r_{\mathsf{Com}} = \mathsf{PRF.Eval}(k_{\mathsf{PRF}}^{(s)}, (\mathsf{spk}, \sigma_{\mathrm{RS}}, 1))$
(6) $c = \Pi_{\mathrm{COM}}.\mathsf{Com}((\mathsf{spk}, \sigma_\Sigma); r_{\mathsf{Com}})$

If $\Pi_{\mathrm{CDVS}}.\mathsf{ClmVrf}(\mathsf{pp}, \mathsf{spk}, \mathsf{vpk}, \sigma, \pi) = 1$ where $\pi \leftarrow \Pi_{\mathrm{CDVS}}.\mathsf{Claim}(\mathsf{pp}, \mathsf{spk}, \mathsf{ssk}, \mathsf{vpk}, \sigma)$, then it returns $\sigma := (\sigma_{\mathrm{RS}}, c)$, and updates $L_{\mathsf{Sign}} := L_{\mathsf{Sign}} \cup \{(\mathsf{vpk}, \mathsf{spk}, \mathsf{m}, \sigma)\}$, otherwise $\perp$.

$\mathsf{O}_{\mathsf{Clm}}$: Given $\mathsf{vpk}, \mathsf{spk}$, and $\sigma = (\sigma_{\mathrm{RS}}, c)$, it returns $\perp$ if $(\mathsf{vpk}, \cdot) \notin L'_{\mathsf{VPK}}, (\mathsf{spk}, \mathsf{ssk}) \notin L'_{\mathsf{SPK}}$ for some $\mathsf{ssk} = (\mathsf{pk}_\Sigma^{(s)}, \mathsf{pk}_{\mathrm{RS}}^{(s)}, k_{\mathsf{PRF}}^{(s)}, \mathsf{sk}_\Sigma^{(s)}, \phi)$, or $(\mathsf{vpk}, \mathsf{spk}, \cdot, \sigma) \notin L'_{\mathsf{Sign}}$. Outputs π if $\{(\mathsf{vpk}, \mathsf{vsk}, \sigma, \pi)\} \in L_{\mathsf{Clm}}$. Otherwise, it computes $r_\Sigma = \mathsf{PRF.Eval}(k_{\mathsf{PRF}}^{(s)}, (\mathsf{spk}, \sigma_{\mathrm{RS}}, 0))$, $r_{\mathsf{Com}} = \mathsf{PRF.Eval}(k_{\mathsf{PRF}}^{(s)}, (\mathsf{spk}, \sigma_{\mathrm{RS}}, 1))$, and $\sigma_\Sigma = \Sigma.\mathsf{Sig}(\mathsf{sk}^{(s)}, (\mathsf{spk}, \sigma_{\mathrm{RS}}); r_\Sigma)$. If $c \neq \Pi_{\mathrm{COM}}.\mathsf{Com}((\mathsf{spk}, \sigma_\Sigma); r_{\mathsf{Com}})$, then it outputs $\perp$, otherwise outputs $\pi := (r_{\mathsf{Com}}, \sigma_\Sigma)$, and updates $L_{\mathsf{Clm}} := L_{\mathsf{Clm}} \cup \{(\mathsf{vpk}, \mathsf{vsk}, \sigma, \pi)\}$.

Challenge phase. When $\mathcal{A}$ outputs $(\mathsf{spk}^*, \mathsf{vpk}^*, \mathsf{m}^*, \sigma^*)$ where $\mathsf{vpk}^* = (\mathsf{pk}_\Sigma^{(v)}, \mathsf{pk}_{\mathrm{RS}}^{(v)})$ and $\mathsf{spk}^* = (\mathsf{pk}_\Sigma^{(s)}, \mathsf{pk}_{\mathrm{RS}}^{(s)})$, $\mathcal{A}'$ outputs $(\{\mathsf{pk}_{\mathrm{RS}}^{(s)}\} \cup \{\mathsf{pk}_{\mathrm{RS}}^{(v)}\}, \mathsf{m}^*, \sigma^*)$ as a challenge.

Analysis. Suppose that the output $(\mathsf{spk}^*, \mathsf{vpk}^*, \mathsf{m}^*, \sigma^*)$ by $\mathcal{A}$ results in $\mathsf{ExpEUFDVS}_{\Pi_{\mathrm{CDVS}}, \mathcal{A}}(1^\lambda) = 1$ (i.e., $\mathcal{A}$ violates EUF-CMA of Π_{CDVS}). Namely, $\Pi_{\mathrm{CDVS}}.\mathsf{Verify}(\mathsf{pp}, \mathsf{vpk}^*, \mathsf{spk}^*, \mathsf{m}^*, \sigma^*) = 1$ indicates $\Pi_{\mathrm{RS}}.\mathsf{Verify}(\mathsf{pp}, \{\mathsf{pk}_{\mathrm{RS}}^{(s)}\} \cup \{\mathsf{pk}_{\mathrm{RS}}^{(v)}\}, \mathsf{m}^*, \sigma^*) = 1$. Further, $(\mathsf{spk}^*, \cdot) \in L'_{\mathsf{SPK}}, (\mathsf{vpk}^*, \mathsf{vsk}^*) \in L'_{\mathsf{VPK}}, (\mathsf{spk}^*, \cdot) \notin L'_{\mathsf{SSK}}, (\mathsf{vpk}^*, \mathsf{vsk}^*) \notin L'_{\mathsf{VSK}}$, and $(\mathsf{vpk}^*, \mathsf{spk}^*, \mathsf{m}^*, \sigma^*) \notin L'_{\mathsf{Sign}}$ means $(\mathsf{pk}_{\mathrm{RS}}^{(s)}, \mathsf{sk}_{\mathrm{RS}}^{(s)}) \in L_{\mathsf{PK}}, (\mathsf{pk}_{\mathrm{RS}}^{(v)}, \mathsf{sk}_{\mathrm{RS}}^{(v)}) \in L_{\mathsf{PK}}, (\mathsf{pk}_{\mathrm{RS}}^{(v)}, \mathsf{sk}_{\mathrm{RS}}^{(v)}) \notin L_{\mathsf{SK}}, (\mathsf{pk}_{\mathrm{RS}}^{(s)}, \mathsf{sk}_{\mathrm{RS}}^{(s)}) \in L_{\mathsf{PK}}$, and $(\mathsf{pk}_{\mathrm{RS}}^{(s)}, \mathsf{pk}_{\mathrm{RS}}^{(v)}, \mathsf{m}^*, \sigma^*) \notin L_{\mathsf{Sign}}$, respectively. That is, when $\mathcal{A}$ breaks EUF-CMA of Π_{CDVS}, $\mathcal{A}'$ breaks EUF-CMA of Π_{RS} as well. Therefore, the existence of $\mathcal{A}$ that breaks EUF-CMA of Π_{CDVS} with non-negligible probability conflicts EUF-CMA of Π_{RS}. □

Lemma 2. *The CDVS scheme Π_{CDVS} satisfies OTR if the underlying ring signature scheme Π_{RS} satisfies anonymity.*

Proof. We assume for contradiction that there is a PPT adversary $\mathcal{A}$ that breaks OTR of Π_{CDVS} with non-negligible probability. We show that we can construct a PPT adversary $\mathcal{A}'$ that breaks anonymity of Π_{RS} with non-negligible probability by simulating $\mathcal{A}$ inside. We describe how $\mathcal{A}'$ works in $\mathsf{ExpAno}_{\Pi_{\mathrm{RS}}, \mathcal{A}'}(1^\lambda)$ by simulating the experiment $\mathsf{ExpOTR}_{\Pi_{\mathrm{CDVS}}, \mathcal{A}}(1^\lambda)$. (Namely, $\mathcal{A}'$ simulates $\mathsf{ExpOTR}_{\Pi_{\mathrm{CDVS}}, \mathcal{A}}(1^\lambda)$ as a challenger.)

Setup phase. Given a public parameter pp, the adversary $\mathcal{A}'$ sets $L'_{\mathsf{VPK}}, L'_{\mathsf{SPK}}, L'_{\mathsf{VSK}}, L'_{\mathsf{SSK}}, L'_{\mathsf{Sign}}$, and L'_{Clm} as $\emptyset$. Then $\mathcal{A}'$ gives pp and $\mathsf{msk} := \emptyset$ to $\mathcal{A}$. Queries from $\mathcal{A}$ are handled in the same way as in the proof of Lemma 1. When $\mathcal{A}$ outputs $\mathsf{spk}^* = (\mathsf{pk}_\Sigma^{(s)}, \mathsf{pk}_{\mathrm{RS}}^{(s)})$, $\mathsf{vpk}^* = (\mathsf{pk}_\Sigma^{(v)}, \mathsf{pk}_{\mathrm{RS}}^{(v)})$, and m^*, $\mathcal{A}'$ sets $\mathsf{pk}_0 := \mathsf{pk}_{\mathrm{RS}}^{(s)}$ and $\mathsf{pk}_1 := \mathsf{pk}_{\mathrm{RS}}^{(v)}$, and outputs $(\mathsf{m}^*, \mathsf{pk}_0, \mathsf{pk}_1)$ to receive a signature $\sigma_{\mathrm{RS},b}$ where $b \in \{0, 1\}$. (Note that $\mathcal{A}'$ does not output another set of public keys, and we regard $\{\mathsf{pk}_0\} \cup \{\mathsf{pk}_1\}$ as a ring.)

Guessing phase. Given a signature $\sigma_{\mathrm{RS},b}$ from the challenger of $\mathsf{ExpAno}_{\Pi_{\mathrm{RS}}, \mathcal{A}'}(1^\lambda)$, $\mathcal{A}'$ continues to simulate $\mathsf{ExpOTR}_{\Pi_{\mathrm{CDVS}}, \mathcal{A}}$. Namely, $\mathcal{A}'$ should return a signature $\sigma_{b^\dagger}$ that contains $\sigma_{\mathrm{RS},b}$ to $\mathcal{A}$ where $b^\dagger \in \{0, 1\}$ is a challenge bit for $\mathcal{A}$. However, $\mathcal{A}'$ does not know which secret key that corresponds to pk_0 or pk_1 is used to create $\sigma_{RS,b}$. Thus $\mathcal{A}'$ flips a coin to decide which secret key to use. This obviously halves the success probability of $\mathcal{A}'$, but it is still sufficient for our purpose. Formally, $\mathcal{A}'$ does the followings:

(1) Randomly chooses a randomness $b^\dagger \in \{0, 1\}$. If $b^\dagger = 0$, then sets $k_{\mathsf{PRF}}^\dagger := k_{\mathsf{PRF}}^{(s)}$ and $sk_\Sigma^\dagger := \mathsf{sk}_\Sigma^{(s)}$, otherwise sets $k_{\mathsf{PRF}}^\dagger := k_{\mathsf{PRF}}^{(v)}$ and $sk_\Sigma^\dagger := \mathsf{sk}_\Sigma^{(v)}$.
(2) $r_\Sigma = \mathsf{PRF.Eval}(k_{\mathsf{PRF}}^\dagger, (\mathsf{spk}^*, \sigma_{\mathrm{RS},b}, 0))$
(3) $\sigma_\Sigma = \Sigma.\mathsf{Sig}(\mathsf{sk}_\Sigma^\dagger, (\mathsf{spk}^*, \sigma_{\mathrm{RS}}); r_\Sigma)$
(4) $r_{\mathsf{Com}} = \mathsf{PRF.Eval}(k_{\mathsf{PRF}}^\dagger, (\mathsf{spk}^*, \sigma_{\mathrm{RS},b}, 1))$
(5) $c = \Pi_{\mathrm{COM}}.\mathsf{Com}((\mathsf{spk}^*, \sigma_\Sigma); r_{\mathsf{Com}})$

After these computations, $\mathcal{A}'$ gives $\sigma_{b^\dagger} = (\sigma_{\mathrm{RS}}, c)$ to $\mathcal{A}$. Again, queries from $\mathcal{A}$ are handled in the same way as in the proof of Lemma 1. When $\mathcal{A}$ outputs a guessing bit b', $\mathcal{A}'$ outputs b' as well.

Analysis. Suppose that the probability that $\mathcal{A}$ guesses correctly is $1/2 + \epsilon$ where ϵ is non-negligible. In other words, if $b^\dagger = b$, then it holds that $b' = b$ with probability $1/2 + \epsilon$. As this event happens with probability $1/2$, the probability that $\mathcal{A}'$ guesses correctly is at least $1/2 + \epsilon/2$, which violates the anonymity of Π_{RS}. □

Lemma 3. *The CDVS scheme Π_{CDVS} is claimable.*

Proof. The first condition of claimability is trivial from correctness of Σ and pseudorandomness of PRF. Thus, we focus on claim unforgeability and non-frameability. □

Claim 1. *If there is a PPT adversary $\mathcal{A}$ that violates the claim unforgeability of Π_{CDVS} with non-negligible probability, then we can construct a PPT adversary $\mathcal{A}'$ that violates binding property of Π_{COM} with non-negligible probability.*

Proof. Suppose that $\mathcal{A}$ outputs $(r_{\mathsf{Com}}, \sigma_\Sigma)$ that results in 1 in the following experiment with non-negligible probability.

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

$L_{\mathsf{VPK}} := \emptyset; L_{\mathsf{SPK}} := \emptyset; L_{\mathsf{VSK}} := \emptyset; L_{\mathsf{SSK}} := \emptyset; L_{\mathsf{Sign}} := \emptyset; L_{\mathsf{Clm}} := \emptyset;$
$(\mathsf{pp}, \mathsf{msk}) \leftarrow \Pi_{\mathrm{CDVS}}.\mathsf{Set}(1^\lambda);$
$(\mathsf{spk}, \mathsf{ssk}) = ((\mathsf{pk}_\Sigma^{(s)}, \mathsf{pk}_{\mathrm{RS}}^{(s)}), (\mathsf{pk}_\Sigma^{(s)}, \mathsf{pk}_{\mathrm{RS}}^{(s)}, k_{\mathsf{PRF}}^{(s)}, \mathsf{sk}_\Sigma^{(s)}, \mathsf{sk}_{\mathrm{RS}}^{(s)})) \leftarrow \Pi_{\mathrm{CDVS}}.\mathsf{SKG}(\mathsf{pp});$
$(\mathsf{m}^*, \mathsf{vpk}^*, \mathsf{vsk}^*)$
$= (\mathsf{m}^*, (\mathsf{pk}_\Sigma^{(v)}, \mathsf{pk}_{\mathrm{RS}}^{(v)}), (\mathsf{pk}_\Sigma^{(v)}, \mathsf{pk}_{\mathrm{RS}}^{(v)}, k_{\mathsf{PRF}}^{(v)}, \mathsf{sk}_\Sigma^{(v)}, \mathsf{sk}_{\mathrm{RS}}^{(v)}))$
$\leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{SPK}}, \mathsf{O}_{\mathsf{SSK}}, \mathsf{O}_{\mathsf{VPK}}, \mathsf{O}_{\mathsf{VSK}}, \mathsf{O}_{\mathrm{DVSig}}^{\mathsf{spk,ssk}}, \mathsf{O}_{\mathrm{Vrf}}, \mathsf{O}_{\mathrm{Clm}}^{\mathsf{spk,ssk}}}(\mathsf{pp}, \mathsf{spk});$
$\sigma = (\sigma_{\mathrm{RS}}, c) \leftarrow \Pi_{\mathrm{CDVS}}.\mathsf{DVSign}(\mathsf{pp}, \mathsf{spk}, \mathsf{ssk}, \mathsf{vpk}^*, \mathsf{m});$
$\pi^* = (r_{\mathsf{Com}}, \sigma_\Sigma) \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{SPK}}, \mathsf{O}_{\mathsf{SSK}}, \mathsf{O}_{\mathsf{VPK}}, \mathsf{O}_{\mathsf{VSK}}, \mathsf{O}_{\mathrm{DVSig}}^{\mathsf{spk,ssk}}, \mathsf{O}_{\mathrm{Vrf}}, \mathsf{O}_{\mathrm{Clm}}^{\mathsf{spk,ssk}}}(\sigma);$
$b = \Pi_{\mathrm{CDVS}}.\mathsf{ClmVrf}(\mathsf{pp}, \mathsf{vpk}^*, \mathsf{spk}, \sigma, \pi^*);$
$b' = \Pi_{\mathrm{CDVS}}.\mathsf{Verify}(\mathsf{pp}, \mathsf{vpk}^*, \mathsf{vsk}^*, \mathsf{spk}, \mathsf{m}^*, \sigma)$:
output 1 if $b = 1 \wedge b' = 1 \wedge \mathsf{vpk}^* \neq \mathsf{spk}$
otherwise 0

We provide a PPT adversary $\mathcal{A}'$ against binding property of Π_{COM} that simulates the above experiment as a challenger. In the simulation, $\mathcal{A}'$ firstly computes $(\mathsf{spk}, \mathsf{ssk})$ by itself, and thus it can answer any oracle queries made by $\mathcal{A}$ by using ssk.

When $\mathcal{A}'$ receives $\pi^* = (r_{\mathsf{Com}}, \sigma_\Sigma)$ from $\mathcal{A}$, it checks if $b = \Pi_{\mathrm{CDVS}}.\mathsf{ClmVrf}(\mathsf{pp}, \mathsf{spk}, \mathsf{vpk}^*, \sigma, \pi^*) = 1$, where $\sigma = (\sigma_{\mathrm{RS}}, c)$ is the

signature that $\mathcal{A}'$ creates. Observe that $b = 1$ indicates $\Pi_{\mathrm{COM}}.\mathsf{Com}((\mathsf{vpk}^*, \sigma_\Sigma); r_{\mathsf{Com}}) = c$ by the construction of $\Pi_{\mathrm{CDVS}}.\mathsf{ClmVrf}$. Further, suppose that $b' = \Pi_{\mathrm{CDVS}}.\mathsf{Verify}(\mathsf{pp}, \mathsf{vpk}^*, \mathsf{vsk}^*, \mathsf{spk}, \mathsf{m}^*, \sigma) = 1$. That is, as σ is a valid signature, it holds that $\Pi_{\mathrm{COM}}.\mathsf{Com}((\mathsf{spk}, \sigma'_\Sigma); r'_{\mathsf{Com}}) = c$ where σ'_Σ and r'_{Com} are computed during the computation of $\sigma \leftarrow \Pi_{\mathrm{CDVS}}.\mathsf{DVSign}(\mathsf{pp}, \mathsf{spk}, \mathsf{ssk}, \mathsf{vpk}^*, \mathsf{m}^*)$. However, as $\mathsf{spk} \neq \mathsf{vpk}^*$, this violates binding property of $\Pi_{\mathrm{COM}}.\mathsf{Com}$. As $\mathcal{A}$ breaks claim unforgeability with non-negligible probability, $\mathcal{A}'$ can also violate binding property of Π_{COM} with non-negligible probability by outputting $(\mathsf{vpk}^*, \pi^*, \mathsf{spk}, \pi')$ where $\pi' = (\sigma'_\Sigma, r'_{\mathsf{Com}})$. □

Claim 2. *If there is a PPT adversary that violates non-frameability of Π_{CDVS} with non-negligible probability, then we can construct another PPT adversary that violates EUF-CMA of Σ with non-negligible probability.*

Due to space limitation, we prove Claim 2 in Appendix A. To summarize the discussion, we have proved Lemma 3.

5 THE SECOND CONSTRUCTION

Let $\Pi_{\mathrm{CRSIG}} = (\mathsf{Set}, \mathsf{KG}, \mathsf{RSig}, \mathsf{Verify}, \mathsf{Claim}, \mathsf{ClmVrf})$ be a claimable ring signature scheme. We demonstrate that we can construct a CDVS scheme $\Pi_{\mathrm{CDVS}} = (\mathsf{Set}, \mathsf{SKG}, \mathsf{VKG}, \mathsf{DVSign}, \mathsf{Vrf}, \mathsf{Sim}, \mathsf{Claim}, \mathsf{ClmVrf})$ based on Π_{CRSIG} in a generic manner. The second construction also does not require a master secret key, and the verification algorithm does not take a secret key of a designated verifier.

5.1 Construction

The construction Π_{CDVS} is as follows:

$\Pi_{\mathrm{CDVS}}.\mathsf{Set}(1^\lambda)$: Given a security parameter 1^λ, it outputs $\mathsf{pp} \leftarrow \Pi_{\mathrm{CRSIG}}.\mathsf{Set}(1^\lambda)$, and $\mathsf{msk} := \phi$.
$\Pi_{\mathrm{CDVS}}.\mathsf{SKG}(\mathsf{pp})$: Given a public parameter pp, it outputs $(\mathsf{spk}, \mathsf{ssk}) := (\mathsf{pk}, \mathsf{sk}) \leftarrow \Pi_{\mathrm{CRSIG}}.(\mathsf{pp})$.
$\Pi_{\mathrm{CDVS}}.\mathsf{VKG}(\mathsf{pp})$: Given a public parameter pp, it outputs $(\mathsf{vpk}, \mathsf{vsk}) := (\mathsf{pk}, \mathsf{sk}) \leftarrow \Pi_{\mathrm{CRSIG}}.(\mathsf{pp})$.
$\Pi_{\mathrm{CDVS}}.\mathsf{DVSign}(\mathsf{pp}, \mathsf{spk}, \mathsf{ssk}, \mathsf{vpk}, \mathsf{m})$: Given a public parameter pp, a signer's public key spk and secret key ssk, a verifier's public key vpk, and a message m, it outputs $\sigma \leftarrow \Pi_{\mathrm{CRSIG}}.\mathsf{RSig}(\mathsf{pp}, \mathsf{ssk}, \{\mathsf{spk}\} \cup \{\mathsf{vpk}\}, \mathsf{m})$.
$\Pi_{\mathrm{CDVS}}.\mathsf{Vrf}(\mathsf{pp}, \mathsf{vpk}, \mathsf{spk}, \mathsf{m}, \sigma)$: Given a public parameter pp, a verifier's public key vpk, a signer's public key spk, a message m, and a signature σ, it outputs $b = \Pi_{\mathrm{CRSIG}}.\mathsf{Vrf}(\mathsf{pp}, \{\mathsf{spk}\} \cup \{\mathsf{vpk}\}, \mathsf{m}, \sigma)$.
$\Pi_{\mathrm{CDVS}}.\mathsf{Sim}(\mathsf{pp}, \mathsf{vpk}, \mathsf{vsk}, \mathsf{spk}, \mathsf{m})$: Given a public parameter pp, a verifier's public key vpk and secret key vsk, a signer's public key spk, and a message m, it outputs $\sigma \leftarrow \Pi_{\mathrm{CRSIG}}.\mathsf{RSig}(\mathsf{pp}, \mathsf{vsk}, \mathsf{vpk}, \mathsf{spk}, \mathsf{m})$.
$\Pi_{\mathrm{CDVS}}.\mathsf{Claim}(\mathsf{pp}, \mathsf{spk}, \mathsf{ssk}, \mathsf{vpk}, \sigma)$: Given a public parameter pp, a signer's public key spk and secret key ssk, a verifier's public key vpk, and a signature σ, it outputs $\pi \leftarrow \Pi_{\mathrm{CRSIG}}.\mathsf{Claim}(\mathsf{pp}, \mathsf{ssk}, \{\mathsf{spk}\} \cup \{\mathsf{vpk}\}, \sigma)$.
$\Pi_{\mathrm{CDVS}}.\mathsf{ClmVrf}(\mathsf{pp}, \mathsf{spk}, \mathsf{vpk}, \sigma, \pi)$: Given a public parameter pp, a signer's public key spk, a verifier's public key vpk, a signature σ, and a claim π, it outputs $b = \Pi_{\mathrm{CRSIG}}.\mathsf{ClmVrf}(\mathsf{pp}, \mathsf{spk}, \{\mathsf{spk}\} \cup \{\mathsf{vpk}\}, \sigma, \pi)$.

5.2 Security Proof

We prove that Π_{CDVS} is a CDVS scheme. Correctness is immediate, and thus we show EUF-CMA, OTR, and claimability, respectively.

Lemma 4. *If Π_{CRSIG} satisfies EUF-CMA, then Π_{CDVS} also satisfies EUF-CMA.*

Proof. Suppose that there exists a PPT adversary $\mathcal{A}_{\mathrm{CDVS}}$ that breaks EUF-CMA of Π_{CDVS} with non-negligible probability. Then, we show that we can construct a PPT adversary $\mathcal{A}_{\mathrm{RS}}$ that breaks EUF-CMA of Π_{CRSIG} with non-negligible probability by simulating $\mathsf{ExpEUFDVS}_{\Pi_{\mathrm{CDVS}}, \mathcal{A}_{\mathrm{CDVS}}}(1^\lambda)$. In what follows, we demonstrate how $\mathcal{A}_{\mathrm{RS}}$ works in $\mathsf{ExpEUFRS}^{\mathsf{O}_{\mathrm{CRSClm}}}_{\Pi_{\mathrm{CRSIG}}, \mathcal{A}_{\mathrm{RS}}}(1^\lambda)$.

Setup phase. Given pp, the adversary $\mathcal{A}_{\mathrm{RS}}$ sets $L'_{\mathsf{VPK}}, L'_{\mathsf{SPK}}, L'_{\mathsf{VSK}}, L'_{\mathsf{SSK}}, L'_{\mathsf{Sign}}$, and L'_{Clm} as $\emptyset$. Then, $\mathcal{A}_{\mathrm{RS}}$ simulates $\mathcal{A}_{\mathrm{CDVS}}$ by giving pp. Namely, each oracle query is dealt with as follows (without loss of generality, we assume that $\mathcal{A}_{\mathrm{CDVS}}$ does not call $\mathsf{O}_{\mathsf{Vrf}}$, because $\Pi_{\mathrm{CDVS}}.\mathsf{Vrf}$ does not require a secret key of a designated verifier):

$\mathsf{O}_{\mathsf{SPK}}$: Given pp from $\mathcal{A}_{\mathrm{CDVS}}$, the adversary $\mathcal{A}_{\mathrm{RS}}$ asks the challenger of $\mathsf{ExpEUFRS}^{\mathsf{O}_{\mathrm{CRSClm}}}_{\Pi_{\mathrm{CRSIG}}, \mathcal{A}_{\mathrm{RS}}}(1^\lambda)$ to call O_{PK}, returns the answer spk to $\mathcal{A}_{\mathrm{CDVS}}$, and updates $L'_{\mathsf{SPK}} := L'_{\mathsf{SPK}} \cup \{(\mathsf{spk}, \cdot)\}$.
$\mathsf{O}_{\mathsf{SSK}}$: Given spk from $\mathcal{A}_{\mathrm{CDVS}}$, the adversary $\mathcal{A}_{\mathrm{RS}}$ returns $\perp$ if $(\mathsf{spk}, \cdot) \notin L_{\mathsf{SPK}}$. Otherwise, it asks the challenger of the experiment $\mathsf{ExpEUFRS}^{\mathsf{O}_{\mathrm{CRSClm}}}_{\Pi_{\mathrm{CRSIG}}, \mathcal{A}_{\mathrm{RS}}}(1^\lambda)$ to call O_{SK}, returns the answer ssk to $\mathcal{A}_{\mathrm{CDVS}}$, and updates $L'_{\mathsf{SSK}} := L'_{\mathsf{SSK}} \cup \{(\mathsf{spk}, \mathsf{ssk})\}$.
$\mathsf{O}_{\mathsf{VPK}}$: The same as $\mathsf{O}_{\mathsf{SPK}}$. The adversary $\mathcal{A}_{\mathrm{RS}}$ returns vpk to $\mathcal{A}_{\mathrm{CDVS}}$ and updates $L'_{\mathsf{VPK}} := L'_{\mathsf{VPK}} \cup \{(\mathsf{vpk}, \cdot)\}$.
$\mathsf{O}_{\mathsf{VSK}}$: The same as $\mathsf{O}_{\mathsf{SSK}}$. The adversary $\mathcal{A}_{\mathrm{RS}}$ returns vsk to $\mathcal{A}_{\mathrm{CDVS}}$ and updates $L'_{\mathsf{VSK}} := L'_{\mathsf{VSK}} \cup \{(\mathsf{vpk}, \mathsf{vsk})\}$.
$\mathsf{O}_{\mathsf{DVSig}}$: Given vpk, spk, and m, output σ if $\{(\mathsf{vpk}, \mathsf{spk}, \mathsf{m}, \sigma)\} \in L_{\mathsf{Sign}}$. Otherwise, the adversary $\mathcal{A}_{\mathrm{RS}}$ asks the challenger of $\mathsf{ExpEUFRS}^{\mathsf{O}_{\mathrm{CRSClm}}}_{\Pi_{\mathrm{CRSIG}}, \mathcal{A}_{\mathrm{RS}}}$ to call $\mathsf{O}_{\mathsf{RSig}}$, returns the answer σ to $\mathcal{A}_{\mathrm{CDVS}}$, and updates $L'_{\mathsf{Sign}} := L'_{\mathsf{Sign}} \cup \{(\mathsf{vpk}, \mathsf{spk}, \mathsf{m}, \sigma)\}$.
$\mathsf{O}_{\mathsf{Clm}}$: Given vpk, spk, and σ, output π if $\{(\mathsf{vpk}, \mathsf{vsk}, \sigma, \pi)\} \in L_{\mathsf{Clm}}$. Otherwise, $\mathcal{A}_{\mathrm{RS}}$ asks the challenger of $\mathsf{ExpEUFRS}^{\mathsf{O}_{\mathrm{CRSClm}}}_{\Pi_{\mathrm{CRSIG}}, \mathcal{A}_{\mathrm{RS}}}$ to call $\mathsf{O}_{\mathrm{CRSClm}}$, returns the answer π to $\mathcal{A}_{\mathrm{CDVS}}$, and updates $L'_{\mathsf{Clm}} := L'_{\mathsf{Clm}} \cup \{(\mathsf{vpk}, \mathsf{spk}, \sigma, \pi)\}$.

Challenge phase. When $\mathcal{A}_{\mathrm{CDVS}}$ outputs $(\mathsf{spk}^*, \mathsf{vpk}^*, \mathsf{m}^*, \sigma^*)$, the adversary $\mathcal{A}_{\mathrm{RS}}$ outputs $(\{\mathsf{spk}^*\} \cup \{\mathsf{vpk}^*\}, \mathsf{m}^*, \sigma^*)$ to the challenger.

Analysis. Given $(\{\mathsf{spk}^*\} \cup \{\mathsf{vpk}^*\}, \mathsf{m}^*, \sigma^*)$, if all the following conditions are satisfied, the challenger outputs 1 as the result of $\mathsf{ExpEUFRS}^{\mathsf{O}_{\mathrm{CRSClm}}}_{\Pi_{\mathrm{CRSIG}}, \mathcal{A}_{\mathrm{RS}}}(1^\lambda)$:

(1) Both spk^* and vpk^* are created by O_{PK}.
(2) Both spk^* and vpk^* are not queried to O_{SK}.
(3) Both $(\mathsf{spk}^*, \mathsf{vpk}^*, \mathsf{m}^*)$ and $(\mathsf{vpk}^*, \mathsf{spk}^*, \mathsf{m}^*)$ are not queried to $\mathsf{O}_{\mathsf{RSig}}$.
(4) $\Pi_{\mathrm{CRSIG}}.\mathsf{Vrf}(\mathsf{pp}, \{\mathsf{spk}^*\} \cup \{\mathsf{vpk}^*\}, \mathsf{m}^*, \sigma^*) = 1$.

Suppose that the output $(\mathsf{spk}^*, \mathsf{vpk}^*, \mathsf{m}^*, \sigma^*)$ by $\mathcal{A}_{\mathrm{CDVS}}$ results in $\mathsf{ExpEUFDVS}_{\Pi_{\mathrm{CDVS}}, \mathcal{A}_{\mathrm{CDVS}}}(1^\lambda) = 1$. Then, conditions (1), (2), and (3) are satisfied. Observe that $\Pi_{\mathrm{CDVS}}.\mathsf{Vrf}$ is exactly the same as $\Pi_{\mathrm{CRSIG}}.\mathsf{Vrf}$. Thus, $\Pi_{\mathrm{CDVS}}.\mathsf{Vrf}(\mathsf{pp}, \mathsf{spk}^*, \mathsf{vpk}^*, \mathsf{m}^*, \sigma^*) = 1$ implies $\Pi_{\mathrm{CRSIG}}.\mathsf{Vrf}(\mathsf{pp}, \{\mathsf{spk}^*\} \cup \{\mathsf{vpk}^*\}, \mathsf{m}^*, \sigma^*) = 1$, which means condition (4). As we are assuming that $\mathcal{A}_{\mathrm{CDVS}}$ breaks EUF-CMA of

Π_{CDVS} with non-negligible probability, $\mathcal{A}_{\mathsf{RS}}$ also breaks EUF-CMA of Π_{CRSIG} with non-negligible probability, which is a contradiction. □

Lemma 5. *If Π_{CRSIG} satisfies anonymity, then Π_{CDVS} satisfies OTR.*

Proof. Suppose that there exists a PPT adversary $\mathcal{A}_{\mathsf{CDVS}}$ that breaks OTR of Π_{CDVS} with non-negligible probability. Then, we show that we can construct a PPT adversary $\mathcal{A}_{\mathsf{RS}}$ that breaks anonymity of Π_{CRSIG} by simulating $\mathcal{A}_{\mathsf{CDVS}}$ in $\mathsf{ExpAno}^{\mathsf{O}_{\mathsf{Clm}}}_{\Pi_{\mathsf{CRSIG}},\mathcal{A}_{\mathsf{RS}}}(1^\lambda)$. We demonstrate how $\mathcal{A}_{\mathsf{RS}}$ works in $\mathsf{ExpAno}^{\mathsf{O}_{\mathsf{Clm}}}_{\Pi_{\mathsf{CRSIG}},\mathcal{A}_{\mathsf{RS}}}(1^\lambda)$.

Setup phase. Given pp, the adversary $\mathcal{A}_{\mathsf{RS}}$ sets $L'_{\mathsf{VPK}}, L'_{\mathsf{SPK}}, L'_{\mathsf{VSK}}$, $L'_{\mathsf{SSK}}, L'_{\mathsf{Sign}}$, and L'_{Clm} as $\emptyset$. Then, $\mathcal{A}_{\mathsf{RS}}$ simulates $\mathcal{A}_{\mathsf{CDVS}}$ by giving pp. Each oracle query is dealt with as the proof of Lemma 4.

Challenge phase. When $\mathcal{A}_{\mathsf{CDVS}}$ outputs $(\mathsf{m}^*, \mathsf{spk}^*, \mathsf{vpk}^*)$, $\mathcal{A}_{\mathsf{RS}}$ returns $(\mathsf{m}^*, \mathsf{pk}^*_0, \mathsf{pk}^*_1, \{\mathsf{pk}^*\}) := (\mathsf{m}^*, \mathsf{spk}^*, \mathsf{vpk}^*, \phi)$ to the challenger. Note that $\mathcal{A}_{\mathsf{RS}}$ does not return a set of public keys. Hence, the challenger will use a ring $\{\mathsf{pk}^*_0\} \cup \{\mathsf{pk}^*_1\}$.

Guessing phase. Given σ_b where $b \in \{0, 1\}$, the adversary $\mathcal{A}_{\mathsf{RS}}$ gives σ_b to $\mathcal{A}_{\mathsf{CDVS}}$. When $\mathcal{A}_{\mathsf{CDVS}}$ outputs a guessing bit b', $\mathcal{A}_{\mathsf{RS}}$ returns b' to the challenger. Note that oracle queries from $\mathcal{A}_{\mathsf{CDVS}}$ are treated as in the setup phase.

Analysis. Given b', the challenger outputs 1 as the result of $\mathsf{ExpAno}^{\mathsf{O}_{\mathsf{Clm}}}_{\Pi_{\mathsf{CRSIG}},\mathcal{A}_{\mathsf{RS}}}(1^\lambda)$ if the following conditions are satisfied:

(1) Both $\mathsf{pk}^*_0 (= \mathsf{spk}^*)$ and $\mathsf{pk}^*_1 (= \mathsf{vpk}^*)$ are created by O_{PK}.
(2) Both $\mathsf{pk}^*_0 (= \mathsf{spk}^*)$ and $\mathsf{pk}^*_1 (= \mathsf{vpk}^*)$ are not queried to O_{SK}.
(3) The signature σ_b is not queried to $\mathsf{O}_{\mathsf{Clm}}$.
(4) $b' = b$.

Observe that $\mathsf{ExpOTR}_{\Pi_{\mathsf{CDVS}},\mathcal{A}_{\mathsf{CDVS}}}(1^\lambda) = 1$, indicates $\mathsf{ExpAno}^{\mathsf{O}_{\mathsf{Clm}}}_{\Pi_{\mathsf{CRSIG}},\mathcal{A}_{\mathsf{RS}}}(1^\lambda) = 1$. Thus, the existence of the adversary $\mathcal{A}_{\mathsf{CDVS}}$ that results in $\mathsf{ExpOTR}_{\Pi_{\mathsf{CDVS}},\mathcal{A}_{\mathsf{CDVS}}}(1^\lambda) = 1$ with non-negligible probability better than $1/2$ contradicts the anonymity of Π_{CRSIG}. □

Lemma 6. *If Π_{CRSIG} satisfies claimability, then Π_{CDVS} also satisfies claimability.*

Proof. The property denoted by "Honest signer can claim" is immediate. Thus, we show the remaining two properties. □

Claim 3. *If Π_{CRSIG} satisfies claim unforgeability, then Π_{CDVS} also satisfies claim unforgeability.*

Proof. Now we prove Claim 3. Assume for contradiction that there exists a PPT adversary $\mathcal{A}_{\mathsf{CDVS}}$ that breaks claim unforgeability of Π_{CDVS}. Then we demonstrate a PPT adversary $\mathcal{A}_{\mathsf{RS}}$ that breaks claim unforgeability of Π_{CRSIG} by simulating the experiment $\mathsf{ExpFlsClmDVS}_{\Pi_{\mathsf{CDVS}},\mathcal{A}_{\mathsf{CDVS}}}(1^\lambda)$ in $\mathsf{ExpFlsClmRS}_{\Pi_{\mathsf{CRSIG}},\mathcal{A}_{\mathsf{RS}}}(1^\lambda)$.

Setup phase. Given pp and pk from the challenger, $\mathcal{A}_{\mathsf{RS}}$ sets $\mathsf{spk} := \mathsf{pk}$ and gives $(\mathsf{pp}, \mathsf{spk})$ to $\mathcal{A}_{\mathsf{CDVS}}$. Queries from $\mathcal{A}_{\mathsf{CDVS}}$ to $\mathsf{O}_{\mathsf{SPK}}$, $\mathsf{O}_{\mathsf{VPK}}$, $\mathsf{O}_{\mathsf{SSK}}$, and $\mathsf{O}_{\mathsf{VSK}}$ are handled as in the proof of Lemma 4 (without loss of generality, we assume that $\mathcal{A}_{\mathsf{CDVS}}$ does not make queries to $\mathsf{O}_{\mathsf{Vrf}}$). Queries to $\mathsf{O}^{\mathsf{spk},\mathsf{ssk}}_{\mathsf{DVSig}}$ and $\mathsf{O}^{\mathsf{spk},\mathsf{ssk}}_{\mathsf{Clm}}$ are also dealt with as in the proof of Lemma 4, but with the following additional queries: Given $(\mathsf{vpk}', \mathsf{m})$ (resp., (vpk', σ)), $\mathcal{A}_{\mathsf{RS}}$ asks the challenger to call $\mathsf{O}_{\mathsf{RSig}}$ (resp., $\mathsf{O}_{\mathsf{CRSClm}}$) and returns the answer to $\mathcal{A}_{\mathsf{CDVS}}$.

Challenge phase. When $\mathcal{A}_{\mathsf{CDVS}}$ outputs $(\mathsf{m}^*, \mathsf{vpk}^*)$, the adversary $\mathcal{A}_{\mathsf{RS}}$ sets $\mathsf{pk}^* := \mathsf{vpk}^*$ and sends $(\mathsf{m}^*, \mathsf{pk}^*)$ to the challenger. (Recall that Π_{CDVS} does not require a verifier's secret key for the verification algorithm. Thus, $\mathcal{A}_{\mathsf{CDVS}}$ does not output vsk^*.)

Claim forgery phase. On receiving a proof σ from the challenger, the adversary $\mathcal{A}_{\mathsf{RS}}$ gives σ to $\mathcal{A}_{\mathsf{CDVS}}$. When $\mathcal{A}_{\mathsf{CDVS}}$ outputs a forgery claim π^*, the adversary $\mathcal{A}_{\mathsf{RS}}$ sends it to the challenger. Note that oracle queries from $\mathcal{A}_{\mathsf{CDVS}}$ are handled as in the setup phase.

Analysis. The output of $\mathsf{ExpFlsClmRS}_{\Pi_{\mathsf{CRSIG}},\mathcal{A}_{\mathsf{RS}}}(1^\lambda)$ is 1 if $b = \Pi_{\mathsf{CRSIG}}.\mathsf{ClmVrf}(\mathsf{pp}, \mathsf{pk}^*, \{\mathsf{pk}\} \cup \{\mathsf{pk}^*\}, \sigma, \pi^*) = 1$, $b' = \Pi_{\mathsf{CRSIG}}.\mathsf{Vrf}(\mathsf{pp}, \{\mathsf{pk}\} \cup \{\mathsf{pk}^*\}, \mathsf{m}^*, \sigma) = 1$, and $\mathsf{pk}^* \neq \mathsf{pk}$. Considering the fact that $\Pi_{\mathsf{CDVS}}.\mathsf{Vrf}$ (resp., $\Pi_{\mathsf{CDVS}}.\mathsf{ClmVrf}$) and $\Pi_{\mathsf{CRSIG}}.\mathsf{Vrf}$ (resp., $\Pi_{\mathsf{CRSIG}}.\mathsf{ClmVrf}$) are the same, the condition that $\mathsf{ExpFlsClmDVS}_{\Pi_{\mathsf{CDVS}},\mathcal{A}_{\mathsf{CDVS}}}(1^\lambda) = 1$ is the same as that of $\mathsf{ExpFlsClmRS}_{\Pi_{\mathsf{CRSIG}},\mathcal{A}_{\mathsf{RS}}}(1^\lambda) = 1$. Thus, the existence of $\mathcal{A}_{\mathsf{CDVS}}$ contradicts the claim unforgeability of Π_{CRSIG}. □

Claim 4. *If Π_{CRSIG} is non-frameable, then Π_{CDVS} is also non-frameable.*

Proof. We assume for contradiction that there exists a PPT adversary $\mathcal{A}_{\mathsf{CDVS}}$ that breaks non-frameability of Π_{CDVS} with non-negligible probability. Then we demonstrate that we can construct a PPT adversary $\mathcal{A}_{\mathsf{RS}}$ that breaks non-frameability of Π_{CRSIG} with non-negligible probability by simulating $\mathsf{ExpFrm}_{\Pi_{\mathsf{CDVS}},\mathcal{A}_{\mathsf{CDVS}}}(1^\lambda)$ as a challenger. In what follows, we show how $\mathcal{A}_{\mathsf{RS}}$ works in the experiment $\mathsf{ExpFrmRS}_{\Pi_{\mathsf{CRSIG}},\mathcal{A}_{\mathsf{RS}}}(1^\lambda)$.

Setup phase. Given pp and pk from the challenger, $\mathcal{A}_{\mathsf{RS}}$ sets $\mathsf{spk} := \mathsf{pk}$ and gives $(\mathsf{pp}, \mathsf{spk})$ to $\mathcal{A}_{\mathsf{CDVS}}$. Queries from $\mathcal{A}_{\mathsf{CDVS}}$ to oracles are handled as in the proof of Claim 3.

Challenge phase. When $\mathcal{A}_{\mathsf{CDVS}}$ outputs $(\mathsf{m}^*, \mathsf{vpk}^*, \sigma^*, \pi^*)$, the adversary $\mathcal{A}_{\mathsf{RS}}$ sets $\mathsf{pk}^* := \mathsf{vpk}^*$ and returns $(\mathsf{m}^*, \mathsf{pk}^*, \sigma^*, \pi^*)$ to the challenger. Note that $\mathcal{A}_{\mathsf{CDVS}}$ does not output vsk^*, as the verification algorithm does not take the verifier's secret key.

Analysis. The output of $\mathsf{ExpFrmRS}_{\Pi_{\mathsf{CRSIG}},\mathcal{A}_{\mathsf{RS}}}(1^\lambda)$ is 1 if $b = \Pi_{\mathsf{CRSIG}}.\mathsf{ClmVrf}(\mathsf{pp}, \mathsf{pk}^*, \{\mathsf{pk}\} \cup \{\mathsf{pk}^*\}, \sigma^*, \pi^*) = 1$, $b' = \Pi_{\mathsf{CRSIG}}.\mathsf{Vrf}(\mathsf{pp}, \{\mathsf{pk}\} \cup \{\mathsf{pk}^*\}, \mathsf{m}^*, \sigma^*) = 1$, and σ^* is not queried to $\mathsf{O}_{\mathsf{CRSClm}}$. By the same discussion as in the proof of Claim 3, we conclude that the existence of $\mathcal{A}_{\mathsf{CDVS}}$ conflicts non-frameability of Π_{CRSIG}. □

We conclude that Lemma 6 holds, by Claim 3 and Claim 4.

6 CONCLUSION

In this work, we introduce claimable designated verifier signature (CDVS). Then, we propose two generic constructions of CDVS. The first construction is based on (standard) ring signature, (non-ring, standard) signature, pseudorandom function, and commitment. The second construction is based solely on claimable ring signature.

ACKNOWLEDGMENTS

This research was in part conducted under a contract of "Research and development on IoT malware removal / make it non-functional technologies for effective use of the radio spectrum" among "Research and Development for Expansion of Radio Wave Resources (JPJ000254)," which was supported by the Ministry of Internal Affairs and Communications, Japan. This research was also in part conducted under JST CREST (JPMJCR21M5, JPMJCR22M1), and under JST AIP Acceleration Research (JPMJCR22U5).

REFERENCES

[1] Masayuki Abe, Miyako Ohkubo, and Koutarou Suzuki. 2002. 1-out-of-n Signatures from a Variety of Keys. In *Advances in Cryptology — ASIACRYPT 2002*. 415–432.
[2] Man Ho Allen Au, Guomin Yang, Willy Susilo, and Yunmei Zhang. 2014. (Strong) multidesignated verifiers signatures secure against rogue key attack. *Concurrency and Computation: Practice and Experience* 26 (2014).
[3] Pourandokht Behrouz, Panagiotis Grontas, Vangelis Konstantakatos, Aris Pagourtzis, and Marianna Spyrakou. 2022. Designated-Verifier Linkable Ring Signatures. In *Information Security and Cryptology – ICISC 2021*. 51–70.
[4] Adam Bender, Jonathan Katz, and Ruggero Morselli. 2006. Ring Signatures: Stronger Definitions, and Constructions Without Random Oracles. In *Theory of Cryptography*. 60–79.
[5] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. 2003. Aggregate and Verifiably Encrypted Signatures from Bilinear Maps *(EUROCRYPT'03)*. Springer-Verlag, 416–432.
[6] Nikita Borisov, Ian Goldberg, and Eric Brewer. 2004. Off-the-Record Communication, or, Why Not to Use PGP. In *Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society (WPES '04)*. Association for Computing Machinery, 77–84.
[7] David Chaum. 1994. Designated confirmer signatures. In *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer, 86–91.
[8] David Chaum. 1996. Private signature and proof systems. US Patent 5,493,614.
[9] David Chaum and Eugène Van Heyst. 1991. Group Signatures *(EUROCRYPT'91)*. 257–265.
[10] Sherman Chow. 2008. Multi-Designated Verifiers Signatures Revisited. *International Journal of Network Security* 7 (2008).
[11] Ivan Damgård, Helene Haagh, Rebekah Mercer, Anca Nitulescu, Claudio Orlandi, and Sophia Yakoubov. 2020. Stronger Security and Constructions of Multidesignated Verifier Signatures. In *TCC*.
[12] Yevgeniy Dodis, Aggelos Kiayias, Antonio Nicolosi, and Victor Shoup. 2004. Anonymous Identification in Ad Hoc Groups. In *Advances in Cryptology - EUROCRYPT 2004*. 609–626.
[13] Eiichiro Fujisaki and Koutarou Suzuki. 2007. Traceable Ring Signature. In *Public Key Cryptography – PKC 2007*. 181–200.
[14] Essam Ghadafi. 2015. Efficient Distributed Tag-Based Encryption and Its Application to Group Signatures with Efficient Distributed Traceability. In *Progress in Cryptology - LATINCRYPT 2014*. 327–347.
[15] Javier Herranz and Germán Sáez. 2003. Forking Lemmas for Ring Signature Schemes. In *Progress in Cryptology - INDOCRYPT 2003*. 266–279.
[16] Xinyi Huang, Willy Susilo, Yi Mu, and Futai Zhang. 2006. Restricted Universal Designated Verifier Signature. In *Proceedings of the Third International Conference on Ubiquitous Intelligence and Computing*. 874–882.
[17] Markus Jakobsson, Kazue Sako, and Russell Impagliazzo. 1996. Designated Verifier Proofs and Their Applications *(EUROCRYPT'96)*. Springer-Verlag, 143–154.
[18] Aggelos Kiayias, Yiannis Tsiounis, and Moti Yung. 2004. Traceable Signatures. In *Advances in Cryptology - EUROCRYPT 2004*. 571–589.
[19] Markulf Kohlweiss and Ian Miers. 2015. Accountable Metadata-Hiding Escrow: A Group Signature Case Study. *Proceedings on Privacy Enhancing Technologies* 2015 (02 2015).
[20] Fabien Laguillaumie and Damien Vergnaud. 2004. Multi-designated Verifiers Signatures. In *International Conference on Information, Communications and Signal Processing*.
[21] Fabien Laguillaumie and Damien Vergnaud. 2007. Multi-Designated Verifiers Signatures: Anonymity without Encryption. *Inf. Process. Lett.* 102, 2–3 (apr 2007), 127–132.
[22] Benoît Libert, Khoa Nguyen, Thomas Peters, and Moti Yung. 2021. Bifurcated Signatures: Folding the Accountability vs. Anonymity Dilemma into a Single Private Signing Scheme. In *Advances in Cryptology – EUROCRYPT 2021*. 521–552.
[23] Joseph K. Liu, Victor K. Wei, and Duncan S. Wong. 2004. Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups. In *Information Security and Privacy*. 325–335.
[24] Mark Manulis, Ahmad-Reza Sadeghi, and Jörg Schwenk. 2006. Linkable Democratic Group Signatures. In *Proceedings of the Second International Conference on Information Security Practice and Experience*. 187–201.
[25] Sunoo Park and Adam Sealfon. 2019. It wasn't me! Repudiability and Unclaimability of Ring Signatures. In *Annual International Cryptology Conference*. Springer, 159–190.
[26] Maharage Nisansala Sevwandi Perera, Toru Nakamura, Masayuki Hashimoto, Hiroyuki Yokoyama, Chen-Mou Cheng, and Kouichi Sakurai. 2022. A Survey on Group Signatures and Ring Signatures: Traceability vs. Anonymity. *Cryptography* 6, 1 (2022).
[27] Ronald L. Rivest, Adi Shamir, and Yael Tauman. 2006. *How to Leak a Secret: Theory and Applications of Ring Signatures*. Springer-Verlag, 164–186.
[28] Shahrokh Saeednia, Steve Kremer, and Olivier Markowitch. 2004. An Efficient Strong Designated Verifier Signature Scheme. In *Information Security and Cryptology - ICISC 2003*. 40–54.
[29] Yusuke Sakai, Keita Emura, Goichiro Hanaoka, Yutaka Kawai, Takahiro Matsuda, and Kazumasa Omote. 2013. Group Signatures with Message-Dependent Opening. In *Pairing-Based Cryptography – Pairing 2012*. 270–294.
[30] Ron Steinfeld, Laurence Bull, Huaxiong Wang, and Josef Pieprzyk. 2003. Universal designated-verifier signatures. In *Advances in Cryptology - ASIACRYPT 2003*. 523–542.
[31] Damien Vergnaud. 2008. New Extensions of Pairing-based Signatures into Universal (Multi) Designated Verifier Signatures. *Int. J. Found. Comput. Sci.* 20 (2008), 109–133.
[32] Shouhuai Xu and Moti Yung. 2004. Accountable Ring Signatures: A Smart Card Approach. In *Smart Card Research and Advanced Applications VI*. 271–286.

A PROOF OF CLAIM 2

Claim 2 states that if there is a PPT adversary $\mathcal{A}$ that violates non-frameability of Π_{CDVS} with non-negligible probability, then we can construct a PPT adversary $\mathcal{A}''$ that violates EUF-CMA of Σ with non-negligible probability.

Proof. Toward showing the adversary $\mathcal{A}''$, we first consider an intermediate PPT adversary $\mathcal{A}'$ that outputs (m^*, σ_Σ^*) in the experiment $\mathsf{ExpEUFSig}_{\Sigma,\mathcal{A}'}$ s.t. $\Sigma.\mathsf{Verify}(\mathsf{pk}, \mathsf{m}^*, \sigma_\Sigma^*) = 1$, but it is not clear if it results in $\mathsf{ExpEUFSig}_{\Sigma,\mathcal{A}'}(1^\lambda) = 1$. Then, we demonstrate a PPT adversary $\mathcal{A}''$ whose distribution of output is close to that of $\mathcal{A}'$, but certainly results in $\mathsf{ExpEUFSig}_{\Sigma,\mathcal{A}''}(1^\lambda) = 1$. We demonstrate how $\mathcal{A}'$ works in $\mathsf{ExpEUFSig}_{\Sigma,\mathcal{A}'}$ by simulating $\mathsf{ExpFrm}_{\Pi_{\mathsf{CDVS}},\mathcal{A}}$ as a challenger.

Setup phase. Given a public key pk_Σ from the challenger, $\mathcal{A}'$ sets $L'_{\mathsf{VPK}}, L'_{\mathsf{SPK}}, L'_{\mathsf{VSK}}, L'_{\mathsf{SSK}}, L'_{\mathsf{Sign}}$, and L'_{Clm} as $\emptyset$. Then, $\mathcal{A}'$ computes $\mathsf{pp} \leftarrow \Pi_{\mathsf{CDVS}}.\mathsf{Set}(1^\lambda)$, and $(\mathsf{spk}^\dagger, \mathsf{ssk}^\dagger) = ((\mathsf{pk}_\Sigma^{(s)}, \mathsf{pk}_{\mathsf{RS}}^{(s)}), (\mathsf{pk}_\Sigma^{(s)}, \mathsf{pk}_{\mathsf{RS}}^{(s)}, k_{\mathsf{PRF}}^{(s)}, \mathsf{sk}_\Sigma^{(s)}, \mathsf{sk}_{\mathsf{RS}}^{(s)})) \leftarrow \Pi_{\mathsf{CDVS}}.\mathsf{SKG}(\mathsf{pp})$. It sets $\mathsf{spk}' := (\mathsf{pk}_\Sigma, \mathsf{pk}_{\mathsf{RS}}^{(s)})$, and $\mathsf{ssk}' := (\mathsf{pk}_\Sigma^{(s)}, \mathsf{pk}_{\mathsf{RS}}^{(s)}, k_{\mathsf{PRF}}^{(s)}, \perp, \mathsf{sk}_{\mathsf{RS}}^{(s)})$, and gives spk' to $\mathcal{A}$.

Forgery phase. Given a query from $\mathcal{A}$, the adversary $\mathcal{A}'$ simulates the answer as follows (without loss of generality, we assume that $\mathcal{A}_{\mathsf{CDVS}}$ does not call $\mathsf{O}_{\mathsf{Vrf}}$, because $\Pi_{\mathsf{CDVS}}.\mathsf{Vrf}$ does not require a secret key of a designated verifier):

$\mathsf{O}_{\mathsf{SPK}}$: Given a public parameter pp, it computes $k_{\mathsf{PRF}} \leftarrow \mathsf{PRF.KG}(1^\lambda)$, $(\mathsf{pk}_\Sigma, \mathsf{sk}_\Sigma) \leftarrow \Sigma.\mathsf{KG}(1^\lambda)$, and $(\mathsf{pk}_{\mathsf{RS}}, \mathsf{sk}_{\mathsf{RS}}) \leftarrow \Pi_{\mathsf{RS}}.\mathsf{KG}(\mathsf{pp})$. It sets $\mathsf{spk} := (\mathsf{pk}_\Sigma, \mathsf{pk}_{\mathsf{RS}})$, and $\mathsf{ssk} := (\mathsf{pk}_\Sigma, \mathsf{pk}_{\mathsf{RS}}, k_{\mathsf{PRF}}, \mathsf{sk}_\Sigma, \mathsf{sk}_{\mathsf{RS}})$, returns spk to $\mathcal{A}$, and updates $L_{\mathsf{SPK}} := L_{\mathsf{SPK}} \cup \{(\mathsf{spk}, \mathsf{ssk})\}$.
$\mathsf{O}_{\mathsf{SSK}}$: Given spk, it returns $\perp$ if $(\mathsf{spk}, \mathsf{ssk}) \notin L'_{\mathsf{SPK}}$ for some ssk. Otherwise, it returns ssk, and updates $L_{\mathsf{SSK}} := L_{\mathsf{SSK}} \cup \{(\mathsf{spk}, \mathsf{ssk})\}$.
$\mathsf{O}_{\mathsf{VPK}}$: The same as $\mathsf{O}_{\mathsf{SPK}}$ except that it returns $\mathsf{vpk} := (\mathsf{pk}_\Sigma, \mathsf{pk}_{\mathsf{RS}})$, and updates $L_{\mathsf{VPK}} := L_{\mathsf{VPK}} \cup \{(\mathsf{vpk}, \mathsf{vsk})\}$.
$\mathsf{O}_{\mathsf{VSK}}$: The same as $\mathsf{O}_{\mathsf{SSK}}$ except that it returns vsk, and updates $L_{\mathsf{VSK}} := L_{\mathsf{VSK}} \cup \{(\mathsf{vpk}, \mathsf{vsk})\}$.

$\mathsf{O}_{\mathsf{DVSig}}$: Given $\mathsf{spk} = (\mathsf{pk}_\Sigma, \mathsf{pk}_{\mathsf{RS}})$, $\mathsf{vpk} = (\mathsf{pk}'_\Sigma, \mathsf{pk}'_{\mathsf{RS}})$, and m, it returns $\perp$ if neither of the following conditions are satisfied: (i) $(\mathsf{vpk}, \cdot) \in L'_{\mathsf{VPK}} \wedge (\mathsf{spk}, \mathsf{ssk}) \in L'_{\mathsf{SPK}}$, nor (ii) $(\mathsf{vpk}, \cdot) \in L'_{\mathsf{VPK}} \wedge \mathsf{spk} = \mathsf{spk}'$. Otherwise, let $\mathcal{A}'$ does the followings (in what follows, we let $\mathsf{ssk} = (\mathsf{pk}_\Sigma, \mathsf{pk}_{\mathsf{RS}}, k_{\mathsf{PRF}}, \mathsf{sk}_\Sigma, \mathsf{sk}_{\mathsf{RS}})$):

(1) If $\{(\mathsf{vpk}, \mathsf{spk}, \mathsf{m}, \sigma)\} \in L_{\mathsf{Sign}}$, then returns σ.
(2) $\sigma_{\mathsf{RS}} \leftarrow \Pi_{\mathsf{RS}}.\mathsf{RSig}(\mathsf{pp}, \mathsf{sk}_{\mathsf{RS}}, \{\mathsf{pk}_{\mathsf{RS}}\} \cup \{\mathsf{pk}'_{\mathsf{RS}}\}, \mathsf{m})$
(3) If $\mathsf{spk} = \mathsf{spk}'$, then $\mathcal{A}'$ asks the challenger to call $\mathsf{O}_{\mathsf{Sig}}$ on $\mathsf{m}' = (\mathsf{spk}, \sigma_{\mathsf{RS}})$ to obtain a signature σ_Σ. Otherwise, $\mathcal{A}'$ computes $r_\Sigma = \mathsf{PRF}.\mathsf{Eval}(k_{\mathsf{PRF}}, (\mathsf{spk}, \sigma_{\mathsf{RS}}, 0))$ and $\sigma_\Sigma = \Sigma.\mathsf{Sig}(\mathsf{sk}_\Sigma, (\mathsf{spk}, \sigma_{\mathsf{RS}}); r_\Sigma)$.
(4) $r_{\mathsf{Com}} = \mathsf{PRF}.\mathsf{Eval}(k_{\mathsf{PRF}}, (\mathsf{spk}, \sigma_{\mathsf{RS}}, 1))$
(5) $c = \Pi_{\mathsf{COM}}.\mathsf{Com}((\mathsf{spk}, \sigma_\Sigma); r_{\mathsf{Com}})$

If $\Pi_{\mathsf{CDVS}}.\mathsf{ClmVrf}(\mathsf{pp}, \mathsf{spk}, \mathsf{vpk}, \sigma, \pi) = 1$ where $\pi = \Pi_{\mathsf{CDVS}}.\mathsf{Claim}(\mathsf{pp}, \mathsf{spk}, \mathsf{ssk}, \mathsf{vpk}, \sigma)$, then it returns $\sigma := (\sigma_{\mathsf{RS}}, c)$, and updates $L_{\mathsf{Sign}} := L_{\mathsf{Sign}} \cup \{(\mathsf{vpk}, \mathsf{spk}, \mathsf{m}, \sigma)\}$, otherwise $\perp$.

$\mathsf{O}_{\mathsf{Clm}}$: Given $\mathsf{vpk}, \mathsf{spk}$, and $\sigma = (\sigma_{\mathsf{RS}}, c)$, it returns $\perp$ if neither of the following conditions is satisfied: (i) $(\mathsf{vpk}, \cdot) \in L'_{\mathsf{VPK}} \wedge (\mathsf{spk}, \mathsf{ssk}) \in L'_{\mathsf{SPK}} \wedge (\mathsf{vpk}, \mathsf{spk}, \cdot, \sigma) \notin L'_{\mathsf{Sign}}$, nor (ii) $(\mathsf{vpk}, \cdot) \in L'_{\mathsf{VPK}} \wedge \mathsf{spk} = \mathsf{spk}' \wedge (\mathsf{vpk}, \mathsf{spk}, \cdot, \sigma) \notin L'_{\mathsf{Sign}}$. If $\{(\mathsf{vpk}, \mathsf{vsk}, \sigma, \pi)\} \in L_{\mathsf{Clm}}$, then returns π. Otherwise, it computes $r_\Sigma = \mathsf{PRF}.\mathsf{Eval}(k_{\mathsf{PRF}}, (\mathsf{spk}, \sigma_{\mathsf{RS}}, 0))$, and $r_{\mathsf{Com}} = \mathsf{PRF}.\mathsf{Eval}(k_{\mathsf{PRF}}, (\mathsf{spk}, \sigma_{\mathsf{RS}}, 1))$. If $\mathsf{spk} = \mathsf{spk}'$, then the adversary $\mathcal{A}'$ asks the challenger to call $\mathsf{O}_{\mathsf{Sig}}$ on $\mathsf{m}' = (\mathsf{spk}, \sigma_{\mathsf{RS}})$ to obtain σ_Σ. Otherwise, $\mathcal{A}'$ computes $\sigma_\Sigma = \Sigma.\mathsf{Sig}(\mathsf{sk}_\Sigma, (\mathsf{spk}, \sigma_{\mathsf{RS}}); r_\Sigma)$. If $c \neq \Pi_{\mathsf{COM}}.\mathsf{Com}((\mathsf{spk}, \sigma_\Sigma); r_{\mathsf{Com}})$, then it outputs $\perp$, otherwise outputs $\pi = (r_{\mathsf{Com}}, \sigma_\Sigma)$, and updates $L_{\mathsf{Clm}} := L_{\mathsf{Clm}} \cup \{(\mathsf{vpk}, \mathsf{vsk}, \sigma, \pi)\}$.

When $\mathcal{A}$ outputs $(\mathsf{m}^*, \mathsf{vpk}^*, \mathsf{vsk}^*, \sigma^*, \pi^*)$ where $\pi^* = (\sigma^*_\Sigma, r^*_{\mathsf{Com}})$, the adversary $\mathcal{A}'$ outputs $(\mathsf{m}^*, \sigma^*_\Sigma)$.

Analysis. We show that the signature output by $\mathcal{A}'$ should satisfy the requirements by the challenger, by the following hybrid argument.

Hybrid 0: The above experiment.

Hybrid 1: When the challenger in $\mathsf{ExpEUFSig}_{\Pi_{\mathsf{RS}}, \mathcal{A}'}$ calls $\mathsf{O}_{\mathsf{Sig}}$, it uses PRF for the randomness instead of using a true randomness. That is, when the challenger receives a query $(\mathsf{spk}, \sigma_{\mathsf{RS}})$, it computes $r = \mathsf{PRF}.\mathsf{Eval}(k_{\mathsf{PRF}}, (\mathsf{spk}, \sigma_{\mathsf{RS}}, 0))$ for randomness.

Hybrid 2: Instead of asking the challenger to call $\mathsf{O}_{\mathsf{Sig}}$ in the simulation of $\mathsf{O}_{\mathsf{DVSig}}$ and $\mathsf{O}_{\mathsf{Clm}}$, the adversary $\mathcal{A}'$ runs $\Sigma.\mathsf{Sig}$ by using the secret key $\mathsf{sk}^{(s)}_\Sigma$ generated by $\mathcal{A}'$ itself. That is, $\mathcal{A}'$ uses $(\mathsf{spk}^\dagger, \mathsf{ssk}^\dagger) = ((\mathsf{pk}^{(s)}_\Sigma, \mathsf{pk}^{(s)}_{\mathsf{RS}}), (\mathsf{pk}^{(s)}_\Sigma, \mathsf{pk}^{(s)}_{\mathsf{RS}}, k^{(s)}_{\mathsf{PRF}}, \mathsf{sk}^{(s)}_\Sigma, \mathsf{sk}^{(s)}_{\mathsf{RS}}))$ instead of $(\mathsf{spk}', \mathsf{ssk}')$. Namely, when given $\mathsf{spk} = \mathsf{spk}^\dagger$, $\mathcal{A}'$ does the followings:

$\mathsf{O}_{\mathsf{DVSig}}$: Given $\mathsf{spk}^\dagger$, $\mathsf{vpk} = (\mathsf{pk}'_\Sigma, \mathsf{pk}'_{\mathsf{RS}})$, and m, it returns $\perp$ if $(\mathsf{vpk}, \cdot) \in L'_{\mathsf{VPK}}$. Otherwise, $\mathcal{A}'$ does the followings:

(1) If $\{(\mathsf{vpk}, \mathsf{spk}^\dagger, \mathsf{m}, \sigma)\} \in L_{\mathsf{Sign}}$, then return σ.
(2) $\sigma_{\mathsf{RS}} \leftarrow \Pi_{\mathsf{RS}}.\mathsf{RSig}(\mathsf{pp}, \mathsf{sk}^{(s)}_{\mathsf{RS}}, \{\mathsf{pk}^{(s)}_{\mathsf{RS}}\} \cup \{\mathsf{pk}'_{\mathsf{RS}}\}, \mathsf{m})$.
(3) $r_\Sigma = \mathsf{PRF}.\mathsf{Eval}(k^{(s)}_{\mathsf{PRF}}, (\mathsf{spk}^\dagger, \sigma_{\mathsf{RS}}, 0))$ and $\sigma_\Sigma = \Sigma.\mathsf{Sig}(\mathsf{sk}^{(s)}_\Sigma, (\mathsf{spk}^\dagger, \sigma_{\mathsf{RS}}); r_\Sigma)$.
(4) $r_{\mathsf{Com}} = \mathsf{PRF}.\mathsf{Eval}(k^{(s)}_{\mathsf{PRF}}, (\mathsf{spk}^\dagger, \sigma_{\mathsf{RS}}, 1))$.
(5) $c = \Pi_{\mathsf{COM}}.\mathsf{Com}((\mathsf{spk}^{(s)}, \sigma_\Sigma); r_{\mathsf{Com}})$.

If $\Pi_{\mathsf{CDVS}}.\mathsf{ClmVrf}(\mathsf{pp}, \mathsf{spk}^\dagger, \mathsf{vpk}, \sigma, \pi) = 1$ where $\pi = \Pi_{\mathsf{CDVS}}.\mathsf{Claim}(\mathsf{pp}, \mathsf{spk}^\dagger, \mathsf{ssk}^\dagger, \mathsf{vpk}, \sigma)$, then it returns $\sigma := (\sigma_{\mathsf{RS}}, c)$, and updates $L_{\mathsf{Sign}} := L_{\mathsf{Sign}} \cup \{(\mathsf{vpk}, \mathsf{spk}^\dagger, \mathsf{m}, \sigma)\}$, otherwise $\perp$.

$\mathsf{O}_{\mathsf{Clm}}$: Given $\mathsf{vpk}, \mathsf{spk}^\dagger$, and $\sigma = (\sigma_{\mathsf{RS}}, c)$, it returns $\perp$ if $(\mathsf{vpk}, \cdot) \notin L'_{\mathsf{VPK}}$ or $(\mathsf{vpk}, \mathsf{spk}, \cdot, \sigma) \notin L'_{\mathsf{Sign}}$. If $\{(\mathsf{vpk}, \mathsf{vsk}^\dagger, \sigma, \pi)\} \in L_{\mathsf{Clm}}$, then returns π. Computes $r_\Sigma = \mathsf{PRF}.\mathsf{Eval}(k^{(s)}_{\mathsf{PRF}}, (\mathsf{spk}^\dagger, \sigma_{\mathsf{RS}}, 0))$, and $r_{\mathsf{Com}} = \mathsf{PRF}.\mathsf{Eval}(k^{(s)}_{\mathsf{PRF}}, (\mathsf{spk}, \sigma_{\mathsf{RS}}, 1))$. Then, $\mathcal{A}'$ computes $\sigma_\Sigma = \Sigma.\mathsf{Sig}(\mathsf{sk}^{(s)}_\Sigma, (\mathsf{spk}^\dagger, \sigma_{\mathsf{RS}}); r_\Sigma)$. If $c \neq \Pi_{\mathsf{COM}}.\mathsf{Com}((\mathsf{spk}^\dagger, \sigma_\Sigma); r_{\mathsf{Com}})$, then outputs $\perp$, otherwise outputs $\pi = (r_{\mathsf{Com}}, \sigma_\Sigma)$, and updates $L_{\mathsf{Clm}} := L_{\mathsf{Clm}} \cup \{(\mathsf{vpk}, \mathsf{vsk}^\dagger, \sigma, \pi)\}$.

Hybrid 0 and Hybrid 1 are computationally close due to pseudorandomness of PRF. We argue that Hybrid 1 and Hybrid 2 are identical, because $\mathcal{A}'$ completely simulates the oracle $\mathsf{O}_{\mathsf{Sig}}$.

It remains to prove that the pair of the message m^* and the signature σ^*_Σ that is contained in the claim $\pi^* = (r^*_{\mathsf{Com}}, \sigma^*_\Sigma)$ output by $\mathcal{A}$ in Hybrid 2 results in $\mathsf{ExpEUFSig}_{\Sigma, \mathcal{A}'}(1^\lambda) = 1$. Observe that Hybrid 2 is exactly the experiment $\mathsf{ExpFrm}_{\Pi_{\mathsf{CDVS}}, \mathcal{A}}(1^\lambda)$. By assumption, π^* passes the verification by $\Pi_{\mathsf{CDVS}}.\mathsf{ClmVrf}$ with non-negligible probability. If this event happens, it implies that $\Sigma.\mathsf{Verify}(\mathsf{pk}_\Sigma, \mathsf{m}^*, \sigma^*_\Sigma) = 1$. Thus, the adversary $\mathcal{A}'$ in Hybrid 0 also outputs such $(\mathsf{m}^*, \sigma^*_\Sigma)$ with non-negligible probability.

However, considering the condition for $\mathsf{ExpEUFSig}_{\Sigma, \mathcal{A}'}(1^\lambda) = 1$, we should show that m^* is not queried to $\mathsf{O}_{\mathsf{Sig}}$, which is not a direct implication by the conditions for $\mathsf{ExpFrm}_{\Pi_{\mathsf{CDVS}}, \mathcal{A}}(1^\lambda) = 1$. To deal with this problem, we introduce another PPT adversary $\mathcal{A}''$ that certainly results in $\mathsf{ExpEUFSig}_{\Sigma, \mathcal{A}''}(1^\lambda) = 1$.

Let q be the maximum number of queries that $\mathcal{A}'$ can make to $\mathsf{O}_{\mathsf{Sig}}$. The adversary $\mathcal{A}''$ is almost the same as $\mathcal{A}'$, but chooses $i \in [q]$ uniformly at random and does the following on the i-th query to $\mathsf{O}_{\mathsf{Sig}}$: Instead of calling $\mathsf{O}_{\mathsf{Sig}}$ on a message m^*, it chooses σ^*_Σ uniformly at random and computes a commitment c^* with respect to σ^*_Σ. Thanks to hiding property of Π_{COM}, the distribution of the output by $\mathcal{A}''$ is computationally close to that of $\mathcal{A}'$ unless c^* is not queried to $\mathsf{O}_{\mathsf{Clm}}$. Here, if $\mathcal{A}'$ queries to $\mathsf{O}_{\mathsf{Sig}}$ on m^* in the i-th query, m^* is not queried to $\mathsf{O}_{\mathsf{Clm}}$ due to the condition for $\mathsf{ExpFrm}_{\Pi_{\mathsf{CDVS}}, \mathcal{A}}(1^\lambda) = 1$. Therefore, if $\Pr[\mathsf{ExpEUFSig}_{\Sigma, \mathcal{A}'}(1^\lambda) = 1] = \epsilon$ where ϵ is non-negligible, it holds that $\Pr[\mathsf{ExpEUFSig}_{\Sigma, \mathcal{A}''}(1^\lambda) = 1] = \epsilon/q - \mathsf{negl}(\lambda)$, which is still non-negligible. □

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009