Designated-Verifier Linkable Ring Signatures

Pourandokht Behrouz[0000−0002−0653−1430], Panagiotis Grontas[0000−0001−7584−0643], Vangelis Konstantakatos[0000−0001−8089−5753], Aris Pagourtzis[0000−0002−6220−3722], and Marianna Spyrakou[0000−0001−5694−8405]

School of Electrical and Computer Engineering
National Technical University of Athens,
9, Iroon Polytechniou St, Postal Code 157 80, Athens, Greece
pbehrouz@mail.ntua.gr, pgrontas@corelab.ntua.gr,
vangelis1993@hotmail.com, pagour@cs.ntua.gr, mspyrakou@mail.ntua.gr

Abstract. We introduce *Designated-Verifier Linkable Ring Signatures (DVLRS)*, a novel cryptographic primitive which combines designated-verifier and linkable ring signatures. Our goal is to guarantee signer ambiguity and provide the capability to the designated verifier to add 'noise' using simulated signatures that are publicly verifiable, thus increasing overall privacy. More formally, we model unforgeability, anonymity, linkability and non-transferability for DVLRS and provide a secure construction in the Random Oracle model. Finally, we explore applications for our primitive, that revolve around the use case of an anonymous assessment system that also protects the subject of the evaluation, even if the private key is compromised.

Keywords: ring signatures, designated verifier, non-transferability, linkability, anonymity

1 Introduction

We present *Designated-Verifier Linkable Ring Signatures (DVLRS)*, a new type of privacy-oriented digital signature. Our primitive is a linkable ring signature [15], i.e. it protects the anonymity of the signers by 'hiding' their identity among a set of peers. Signed messages appear to be coming from the set as a whole, without the ability to exactly pinpoint the sender. Moreover, messages are publicly linkable, i.e. messages coming from the same sender can be identified and grouped together, without disclosing the sender's identity. At the same time our primitive is a designated-verifier signature [7], as it is simulatable by an entity designated during signing, while maintaining public verifiability. As a result, only this designated verifier can be convinced of which messages actually originate from signers in the ring. This option, however, is not available to the public, as all signatures are indistinguishable to them. Consequently, our scheme enhances the privacy of ring members, as no entity apart from the designated verifier can be convinced of the actual sender. At the same time, DVLRS provide more control to the designated verifier, as they can be used to inject 'noise' - fake messages with simulated signatures - thus altering the public view of the adversary.

More importantly, it provides protection to the designated verifier against an adversary who tries to extort or otherwise gain hold of their private key, as even if they succeed, they can gain no valuable information on which messages come from real ring members and which are 'noise'. This makes our scheme useful to a number of privacy-focused scenarios such as evaluation systems and surveys for sensitive data.

1.1 Related Work

Since our primitive combines the notions of *Designated-Verifier Signatures (DVS)* and *Linkable Ring Signatures (LRS)*, we review the evolution of these primitives by focusing on their semantics and their security properties.

DVS were proposed in [7] as a way to restrict the entities that can be convinced by a proof. The relevant property, *non-transferability*, states that the verifier cannot use the resulting signatures to convince other parties. Their construction utilizes an OR proof, stating in effect that the signer knows their secret signing key or the secret key of the verifier. Verification uses both public keys. As a result, the designated verifier can be sure that the signer created a signature they did not create themselves. However, the public, while being able to check if the signature is valid, cannot distinguish between a signer-generated and a *simulated* signature, i.e. one created with the secret key of the verifier. A variation, *strong* DVS, also proposed in [7], are not publicly verifiable as the secret key of the designated verifier is a required input of the verification algorithm. The simplest way to create strong DVS is to encrypt (part of) the signature with the public key of the designated verifier, but other constructions are possible [21]. The applications of DVS range from copyright management to receipt-free and coercion-resistance electronic voting [8].

Subsequent works refined the construction and security properties of DVS. In [22] *non-transferability* was formally defined in the context of *universal designated-verifier* signatures, where the designation functionality is not restricted to the signer. In [12], it was noted that in some previous schemes the signer or the designated verifier could delegate their signing rights, by giving away a function of their respective secret keys and not fully revealing them. As this capability could have negative effects in some applications, a new property *non-delegatability* was defined. It essentially states that a non-delegatable DVS is a proof of knowledge of the signer or designated verifier secret key. They also note that the original DVS scheme of [7] was non-delegatable. In [11] a generic definition of DVS and their related security notions is presented.

Ring signatures were originally proposed in [20] as a method to *anonymize* the signer of a message, by hiding their identity inside a group of possible signers-peers. The signature was verified by the ring public key, without anyone being able to pinpoint the exact signer. Unlike previous schemes, e.g. [3], rings can be formed spontaneously and there is no group manager that may revoke the anonymity of the members. [15] proposed a ring signature as an OR proof by using the classic technique of [5] and added the feature of linkability, where signatures coming from the same signer were linked together using *pseudoidentities* or *tags*. The pseudoidentities were 'alternate' public keys - group elements computed using the private key of the signer - embedded in the signature that enabled the signer to remain anonymous. Their construction could be used to prevent double-voting in anonymous electronic elections. Linkable ring signatures have also been used in anonymous cryptocurrencies like Monero [18]. While in [15] the linkability tag results in computational anonymity, other constructions provided for perfect anonymity [13] and improved security models [16]. *Non-slanderability* [13] ensures that no signer can generate a signature that is determined to be linked to another one not generated by the same signer. Another variation, ring signatures with *designated linkability* [14] restrict who can link signatures. We stress that these signatures are in essence designated linker as the designation is applicable only to linking. Our primitive is entirely different as it considers designation for the verifier, specifying who can be certain that a signature is real and therefore be convinced by it. One drawback of ring signatures, is that while the cryptographic construction might hide the signer, its identity could be revealed from the contents of the message. DVLRS bypasses this problem with the capacity for simulated messages created by the designated verifier.

The notions of designated verifier and ring signatures have been combined in [10], where any holder of a ring signature can designate it to a verifier, and [9] which provides a *strong* DV ring signature for whistle blowing. However, these works do not consider the property of linkability, which makes our scheme more versatile. A first attempt to add linkability to designated verifier ring signatures was made in [4] for use in receipt free e-voting. The resulting signatures, however, are only strongly designated, since part of the signature is encrypted with the public key of the verifier. In addition they are not publicly linkable as the pseudoidentities are encrypted as well. So both verification and linking require the secret key. Our approach, DVLRS, are both publicly verifiable and publicly linkable. Furthermore, in [4], they only achieve non-transferability against a computationally bounded adversary[1]. A big advantage of our work is that we accomplish perfect non-transferability, i.e. even an unbounded attacker cannot distinguish signatures from simulations.

1.2 Contribution

To the best of our knowledge, Designated-Verifier Linkable Ring Signatures are the first attempt to combine *plain* designated-verifier signatures and *publicly* linkable ring signatures. We provide a generic security model and formally define all the relevant security properties that we think should be satisfied: unforgeability, anonymity, linkability and non-transferability. The definition of linkability is extended to include non-slanderability. Our definition for non-transferability is also novel since it adapts the one in [11] for linkability. This is of particular interest, since one has to make sure that the linkability tag does not allow an

[1] There is no security model or security analysis provided in [4] for their signature scheme, however it is straightforward to see that a computationally unbounded attacker can distinguish simulations. We omit the proof due to space limitations.

attacker to distinguish simulations. Our security model is a novel contribution on its own, as it can be used to evaluate future DVLRS instantiations.

We also provide a concrete construction for DVLRS and proofs for all its claimed security properties in the random oracle model. The proposed scheme builds upon the work of [15] and adds a designated verifier capable of simulating signatures and linking them to arbitrary ring members. We achieve *perfect* non-transferability, by making these simulations information theoretically indistinguishable. By construction, in our scheme, unforgeability amounts to a proof of knowledge for the secret key of the signer or the designated verifier. Thus DVLRS cannot be delegated and our proof of unforgeability directly implies a proof of non-delegatability [12, 7]. Finally, we discuss applications of DVLRS by generalizing the case of an anonymous evaluation system that also protects the subject of the evaluation, even if the private key is compromised, by allowing the insertion of simulated signatures.

2 DVLRS Model

2.1 Notation and assumptions

The security parameter is denoted by λ. We let n denote the size of the universe $\mathcal{U}$ of possible public keys and $n_L = |L|$ for a subset (ring) $L \subseteq \mathcal{U}$. We denote equality with $=$ and assignment with $\leftarrow$. All our security definitions are in the form of games which take as input the security parameter and return 1 for $True$ and 0 for $False$. For conciseness, we return the condition and not its result. An algorithm that terminates unsuccessfully is denoted as returning $\perp$. A uniformly at random selection is denoted with $\leftarrow\!\!\$$. We assume the adversary $\mathcal{A}$ has state which is maintained throughout successive calls. In the games it is always omitted for brevity. We collectively refer to the cryptographic parameters of our scheme (groups, generators etc.) as params. They are an input to all our algorithms, but are also omitted. We denote a public key as pk and a secret key as sk. A pseudoidentity is denoted as pid. Typically it is computed as a function of the sk that is believed to be difficult to invert. Other parameters can also take part in its computation like the public keys of L like in [15], possibly combined with some event description from $\{0,1\}^*$ as in [13]. Its actual form depends on the application. We denote by $\mathcal{PID}$ the set of pid's. The designated verifier is denoted as D, while the index of the signer in the ring is π. The security of our scheme rests on standard cryptographic assumptions like the hardness of the discrete logarithm problem (DLP) and the decisional Diffie-Hellman assumption (DDH), which are omitted for brevity.

2.2 DVLRS definition and basic properties

We begin by defining DVLRS and their basic security properties.

Definition 1. *A Designated-Verifier Linkable Ring Signature Π is a tuple of* PPT *algorithms (*$\mathsf{Setup}, \mathsf{KGen}, \mathsf{Sign}, \mathsf{Extract}, \mathsf{Sim}, \mathsf{Vrfy}, \mathsf{Link}$*) with the following syntax:*

- $\mathsf{params} \leftarrow \mathsf{Setup}(\lambda)$ *generates the parameters of DVLRS. These include cryptographic groups, the message space $\mathcal{MSG}$, and the set of possible pseudoidentities $\mathcal{PID}$.*
- $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}()$ *is the key generation algorithm which allows keys to be created in an ad-hoc manner. This algorithm is used by all players including the designated verifier.*
- $\sigma \leftarrow \mathsf{Sign}(L, \mathtt{m}, \mathsf{pk}_D, \mathsf{sk}_\pi)$ *is used to sign a message* $\mathtt{m}$ *by some* $\pi \in [n_L]$*.*
- $\mathsf{pid} \leftarrow \mathsf{Extract}(\sigma)$ *is an algorithm that can obtain the pseudoidentity* pid *from a signature.*
- $\sigma \leftarrow \mathsf{Sim}(L, \mathtt{m}, \mathsf{pk}_D, \mathsf{sk}_D, \mathsf{pid})$ *is the signature simulation algorithm that allows the designated verifier D to produce indistinguishable signatures for pseudoidentity* pid*.*
- $\{0, 1\} \leftarrow \mathsf{Vrfy}(\sigma, L, \mathtt{m}, \mathsf{pk}_D)$ *is the verification algorithm which outputs* 1 *if the signature is valid and* 0 *otherwise.*
- $\{0, 1\} \leftarrow \mathsf{Link}(\sigma, L, \sigma', L)$ *is the linking algorithm which outputs* 1 *if* σ *and* σ' *originate from the same signer or if they are simulated to look like they originate from the same signer.*

In Definition 1, the pseudoidentity pid must be given as input to the simulator to allow linking. This means that to link a simulated signature to a ring member, the designated verifier must first see a single signature from them. This might seem as a drawback of our definition, but in a practical application it is of no importance as its protocol could force all participants to post a single signed registration message for each pseudoidentity they assume. Such a message would not carry sensitive content. Then the designated verifier could use the $\mathsf{Extract}$ functionality to create a registry of pseudoidentities to simulate signatures. Furthermore, the designated verifier can create simulations taking random $\mathsf{pid} \leftarrow\$ \mathcal{PID}$. These won't be linked to the signatures of a real signer and can be generated before the verifier sees any signatures.

The completeness of our scheme is obtained from the following correctness properties that guarantee that honestly generated signatures are usable.

Verification Correctness states that honestly user-generated or simulated signatures are valid. More formally: If $\sigma \leftarrow \mathsf{Sign}(L, \mathtt{m}, \mathsf{pk}_D, \mathsf{sk})$ for $\mathsf{sk} \in L$ or $\sigma \leftarrow \mathsf{Sim}(L, \mathtt{m}, \mathsf{pk}_D, \mathsf{sk}_D, \mathsf{pid})$ for $(\mathsf{pk}_D, \mathsf{sk}_D) \leftarrow \mathsf{KGen}()$, then $\mathsf{Vrfy}(\sigma, L, \mathtt{m}, \mathsf{pk}_D) = 1$ with overwhelming probability. Otherwise $\mathsf{Vrfy}(\sigma, L, \mathtt{m}, \mathsf{pk}_D) = 0$

Linking Correctness states the conditions for linking. Two signatures over the same ring L, should always be linked if they are honestly generated by the same signer, if one is an honestly generated signature and a verifier created a simulation with the particular pseudoidentity or if they are simulations using the same pseudoidentity. Note that the inputs of the linking algorithm have to be valid signatures. If they are not, the output of this algorithm is irrelevant. Formally: $\mathsf{Link}(\sigma, L, \sigma', L) = 1$ if and only if one of the following holds:

i $\sigma \leftarrow \mathsf{Sign}(L, \mathtt{m}, \mathsf{pk}_D, \mathsf{sk}_\pi)$ and $\sigma' \leftarrow \mathsf{Sign}(L, \mathtt{m}', \mathsf{pk}'_D, \mathsf{sk}_\pi)$
ii $\sigma \leftarrow \mathsf{Sign}(L, \mathtt{m}, \mathsf{pk}_D, \mathsf{sk}_\pi)$ and $\sigma' \leftarrow \mathsf{Sim}(L, \mathtt{m}', \mathsf{pk}'_D, \mathsf{sk}'_D, \mathsf{Extract}(\sigma))$
iii $\sigma \leftarrow \mathsf{Sim}(L, \mathtt{m}, \mathsf{pk}_D, \mathsf{sk}_D, \mathsf{pid})$ and $\sigma' \leftarrow \mathsf{Sim}(L, \mathtt{m}', \mathsf{pk}'_D, \mathsf{sk}'_D, \mathsf{pid})$

Note here that we have limited linking to signatures formed over the same ring L. This is simply a choice made for ease of exposition. Modifying the definitions for linking over event tags [13] or even with no restrictions [1] is straightforward.

2.3 Adversarial capabilities

We will consider a strong adaptive adversary that has the ability to add more users to the system, take control of users of its choice, collect all signatures ever exchanged and request signatures and simulations at will on behalf of any of the users of any ring. To model these capabilities of $\mathcal{A}$ we utilize the following oracles[2], similar to [17, 13, 12]:

- $\mathsf{pk} \leftarrow \mathcal{JO}()$. The *Joining Oracle*, upon request adds a public key to the list of public keys $\mathcal{U}$, and returns it.
- $\mathsf{sk} \leftarrow \mathcal{CO}(\mathsf{pk})$. The *Corruption Oracle*, on input a public key pk that is an output of $\mathcal{JO}$ returns the secret key sk such that $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}()$.
- $\sigma \leftarrow \mathcal{SO}(L, \mathtt{m}, \mathsf{pk}_D, \mathsf{pk}_\pi)$. The *Signing Oracle*, on input a list of public keys L a message $\mathtt{m}$, a public key pk_D and a public key $\mathsf{pk}_\pi \in L$, outputs a signature σ such that $\sigma \leftarrow \mathsf{Sign}(L, \mathtt{m}, \mathsf{pk}_D, \mathsf{sk}_\pi)$ and $(\mathsf{pk}_\pi, \mathsf{sk}_\pi) \leftarrow \mathsf{KGen}()$.
- $\sigma \leftarrow \mathcal{MO}(L, \mathtt{m}, \mathsf{pk}_D, \mathsf{pid})$. The *Simulation Oracle*, on input a list of public keys L a message $\mathtt{m}$, a public key pk_D, a pseudoidentity pid, outputs a signature σ such that $\sigma \leftarrow \mathsf{Sim}(L, \mathtt{m}, \mathsf{pk}_D, \mathsf{sk}_D, \mathsf{pid})$ and $(\mathsf{pk}_D, \mathsf{sk}_D) \leftarrow \mathsf{KGen}()$.

These oracles capture the adaptive nature of $\mathcal{A}$. For example, as part of a potential attack, he can after receiving signatures of his choice from $\mathcal{SO}$, request that more users are added to the system from $\mathcal{JO}$, then request even more signatures potentially even from the newly added users, and so forth.

We must point out that while the adversary can collect all messages and signatures, it does not monitor communication addresses, timing information and related metadata, as such information would trivially enable them to distinguish between simulated and real signatures. In essence, we can assume that all signed messages are publicly available as standalone items. Additionally, we expect the designated verifier to adopt an obfuscation strategy when posting fake signatures.

2.4 Unforgeability

Unforgeability intuitively implies the inability of a party that is not a member of a ring to produce a valid signature for that ring, without designating themselves as the Designated-Verifier. To formally define unforgeability for a DVLRS scheme Π, we consider the experiment $\mathsf{Exp}^{\mathrm{unf}}_{\mathcal{A},\Pi,n}$ in Game 1.1.

The adversary queries all the oracles $(\mathcal{RO}, \mathcal{JO}, \mathcal{CO}, \mathcal{SO}, \mathcal{MO})$ according to any adaptive strategy. The corruption oracle $\mathcal{CO}$ models the ability of $\mathcal{A}$ to control any number of members of $\mathcal{U}$. With D_t we denote the set of indices of the keys that have been corrupted. $\mathcal{A}$ chooses the list of public keys L, a

[2] For convenience, we use the same symbol to denote both an oracle and its set of outputs.

Game 1.1: Unforgeability experiment $\mathsf{Exp}^{\mathrm{unf}}_{\mathcal{A},\Pi,n}$

```
Input  : $\lambda$
Output: $\{0,1\}$
$\mathsf{params} \leftarrow \Pi.\mathsf{Setup}(1^\lambda)$
$\mathcal{U} \leftarrow \big\{(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \Pi.\mathsf{KGen}()\big\}_{i=1}^{n}$
$(\sigma, L = \{\mathsf{pk}_i\}_{i=1}^{n_L}, \mathtt{m}, \mathsf{pk}_D, D_t) \leftarrow \mathcal{A}^{\mathcal{RO},\mathcal{JO},\mathcal{CO},\mathcal{SO},\mathcal{MO}}(\mathcal{U})$
return $\mathsf{Vrfy}(\sigma, L, \mathtt{m}, \mathsf{pk}_D) = 1$ AND $\forall i \in D_t\ \mathsf{pk}_i \notin L$ AND $D \notin D_t$ AND
$\sigma \notin \mathcal{SO}$ AND $\sigma \notin \mathcal{MO}$
```

designated verifier D with corresponding public key pk_D, a message $\mathtt{m}$ and creates a forged signature σ. The adversary succeeds if the signature verifies, (i.e. $\mathsf{Vrfy}(\sigma, L, \mathtt{m}, \mathsf{pk}_D) = 1$) and if none of the keys contained in L, nor pk_D, have been queried to $\mathcal{CO}$ and if the signature is not the query output of $\mathcal{SO}$ or $\mathcal{MO}$.

Note that this corresponds to the strong security notion of *Unforgeability w.r.t insider corruption* of [2], adapted for the existence of a Designated-Verifier.

Definition 2. *Unforgeability*
A DVLRS scheme Π is unforgeable if for any PPT *adversary $\mathcal{A}$:*

$$\mathsf{Adv}^{\mathrm{unf}}_{\mathcal{A}}(\lambda) = \Pr\Big[\mathsf{Exp}^{\mathrm{unf}}_{\mathcal{A},\Pi,n}(\lambda) = 1\Big] \leq \mathsf{negl}(\lambda)$$

2.5 Anonymity

Anonymity, also referred to as *signer ambiguity* in [15], intuitively implies the inability of any party, including the designated verifier, to identify the private key used to create a signature. Formally, we consider the experiment $\mathsf{Exp}^{\mathrm{anon}}_{\mathcal{A},\Pi,n,t}$.

Game 1.2: Anonymity experiment $\mathsf{Exp}^{\mathrm{anon}}_{\mathcal{A},\Pi,n,t}$

```
Input  : $\lambda$
Output: $\{0,1\}$
$\mathsf{params} \leftarrow \Pi.\mathsf{Setup}(1^\lambda)$
$\mathcal{U} \leftarrow \big\{(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \Pi.\mathsf{KGen}()\big\}_{i=1}^{n}$
$\mathsf{pk}_D \leftarrow \mathcal{A}(\mathbf{choose}, \mathcal{U})$
$(n_L, L = \{\mathsf{pk}_i\}_{i=1}^{n_L}, \mathtt{m}, D_t) \leftarrow \mathcal{A}^{\mathcal{RO},\mathcal{JO},\mathcal{CO},\mathcal{SO},\mathcal{MO}}(\mathcal{U})$
$\pi \leftarrow\$ [n_L]$
$\sigma \leftarrow \Pi.\mathsf{Sign}(L, \mathtt{m}, \mathsf{pk}_D, \mathsf{sk}_\pi)$
$\xi \leftarrow \mathcal{A}^{\mathcal{RO},\mathcal{SO},\mathcal{CO},\mathcal{MO}}(\mathbf{guess}, L, \mathtt{m}, \sigma, D_t)$
if $\mathcal{SO}$ has not been invoked for $(\mathsf{pk}_\pi, L)$ AND $\pi \notin D_t$ AND $\xi \neq \perp$ then
    return $\xi = \pi$ AND $0 \leq t < n_L - 1$
else
    return $\perp$
end
```

The adversary selects a designated verifier and samples public keys in order to be able to request signatures for messages of their choice. $\mathcal{A}$ can also utilise existing signatures or simulations by respectively querying the oracles $\mathcal{RO}, \mathcal{SO}, \mathcal{MO}$. In order to perform the attack, it selects a ring L of n_L public keys and a message $\mathtt{m}$ to its benefit. $\mathcal{A}$ has also the power to control up to t members of the ring L, modelled by calls to the oracle $\mathcal{CO}$. The set of indices of corrupted members is again denoted by D_t and is dynamically updated each time $\mathcal{CO}$ is used. The challenger randomly selects a ring member (indexed by π) and creates a signature on its behalf. $\mathcal{A}$ must guess which member of the ring has signed the signature. Clearly, if $\mathcal{A}$ controls all members of L except π it can trivially win. As a result, we require that there are at least two members that are not controlled by $\mathcal{A}$ and that the oracle $\mathcal{CO}$ has not been queried for π.

Also, recall that our definition of linking correctness, does not allow linking signatures on different rings. As a result, $\mathcal{A}$ cannot link σ with signatures originating from singleton subrings of L.

Definition 3. *Anonymity*
A DVLRS scheme Π is t-anonymous if for any PPT *adversary $\mathcal{A}$:*

$$\mathsf{Adv}^{\text{anon}}_{\mathcal{A}}(\lambda) = \Pr\left[\mathsf{Exp}^{\text{anon}}_{\mathcal{A},\Pi,n,t}(\lambda) = 1\right] - \frac{1}{n_L - t} \leq \mathsf{negl}(\lambda)$$

2.6 Linkability

Linkability intuitively means that if two signatures come from the same signer over the same ring L, they have to be linked. Our notion is even stronger, we require that a signer that controls $k-1$ private keys cannot produce k valid pairwise unlinkable signatures. To formally define linkability for a DVLRS scheme Π, we use the experiment $\mathsf{Exp}^{\text{link}}_{\mathcal{A},\Pi,n}$ in Game 1.3.

Game 1.3: Linkability experiment $\mathsf{Exp}^{\text{link}}_{\mathcal{A},\Pi,n}$

```
Input  : $\lambda$
Output: $\{0,1\}$
$\mathsf{params} \leftarrow \Pi.\mathsf{Setup}(1^\lambda)$
$\mathcal{U} \leftarrow \left\{(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \Pi.\mathsf{KGen}()\right\}^n_{i=1}$
$(\{\sigma_i\}^k_{i=1}, L = \{\mathsf{pk}_i\}^{n_L}_{i=1}, \{\mathtt{m}_i\}^k_{i=1}, \{\mathsf{pk}_{D_i}\}^k_{i=1}, D_t) \leftarrow \mathcal{A}^{\mathcal{RO},\mathcal{JO},\mathcal{CO},\mathcal{SO},\mathcal{MO}}(\mathcal{U})$

return $\mathsf{Vrfy}(\sigma_i, L, \mathtt{m}_i, \mathsf{pk}_{D_i}) = 1 \; \forall i \in [k]$ AND
$\mathsf{Link}(\sigma_i, L, \sigma_j, L) = 0 \; \forall i, j \in [k], \; i \neq j$ AND
$|\{pk_i : i \in D_t\} \cap L| < k$ AND $D_i \notin D_t \; \forall i \in [k]$ AND
$\sigma_i \notin \mathcal{SO} \; \forall i \in [k]$ AND
$\sigma_i \notin \mathcal{MO} \; \forall i \in [k]$
```

The adversary $\mathcal{A}$ queries all the oracles $(\mathcal{RO}, \mathcal{JO}, \mathcal{CO}, \mathcal{SO}, \mathcal{MO})$ according to any adaptive strategy. We denote by D_t the set of indices of ring members $\mathcal{A}$ has taken control of. This is modeled by calls to the corruption oracle $\mathcal{CO}$. $\mathcal{A}$ chooses the list of public keys L, k messages $\{\mathtt{m}_i\}_{i=1}^k$, k designated verifiers $\{\mathsf{pk}_{D_i}\}_{i=1}^k$ and creates k signatures $\{\sigma_i\}_{i=1}^k$. The adversary succeeds if all k signatures verify, (i.e. $\mathsf{Vrfy}(\sigma_i, L, \mathtt{m}_i, \mathsf{pk}_{D_i}) = 1$, $\forall i \in [k]$), if the signatures are pairwise unlinkable (i.e. $\mathsf{Link}(\sigma_i, L, \sigma_j, L) = 0$, $\forall i, j \in [k]$, $i \neq j$), if strictly less than k keys that are contained in L have been queried to $\mathcal{CO}$, if none of the signatures have a corrupted key as designated verifier and finally if the signatures $\{\sigma_i\}_{i=1}^k$ are not query outputs of $\mathcal{SO}$ or $\mathcal{MO}$. It should be noted, that designated verifiers are allowed to create signatures that are linked or unlinked with any given signature that is designated to them. This is by design, to ensure non-transferability.

Definition 4. *Linkability*
A DVLRS scheme $\mathit{\Pi}$ is linkable if for any PPT *adversary $\mathcal{A}$:*

$$\mathsf{Adv}_{\mathcal{A}}^{\text{link}}(\lambda) = \Pr\Big[\mathsf{Exp}_{\mathcal{A},\mathit{\Pi},n}^{\text{link}}(\lambda) = 1\Big] \leq \mathsf{negl}(\lambda)$$

In the literature for linkable ring signatures, a weaker definition of linkability is often used [1, 13]. This definition requires that a signer who controls a single private key, should not be able to produce two unlinkable signatures, but it allows, for example, a signer who knows two secret keys to produce three pairwise unlinkable signatures. This allows the adversary to circumvent linkability with a very realistic attack [16]; two colluding ring members who share their secret keys with each other can create signatures that are not linked to either of them. Therefore we opted for the stronger definition.

Finally, there is another notion closely related to linkability, called *non-slanderability* in [23]. Intuitively this ensures that given a signature generated by a member of a ring, even a collusion by all the rest, cannot produce a valid signature that is linked to it. However, this is a property that is implied by our stronger notion of linkability, together with unforgeability.

2.7 Non-Transferability

Non-Transferability means that given a valid signature that is linked to signatures that are the output of the Sign algorithm, an adversary cannot distinguish whether it is the output of the Sign or Sim algorithm. Intuitively this ensures that signatures are only useful to the designated verifier, since a third party can never know whether a signature is real or a simulation. To formally define Non-Transferability for a DVLRS scheme $\mathit{\Pi}$, we use $\mathsf{Exp}_{\mathcal{A},\mathit{\Pi},n}^{\text{trans}}$ in Game 1.4.

The adversary in this experiment can be computationally unbounded. Consequently, it is not given access to the oracles $\mathcal{CO}, \mathcal{SO}, \mathcal{MO}$ since it can just compute their outputs. $\mathcal{A}$ chooses the ring L, a message $\mathtt{m}$, a designated verifier pk_D and a target ring member $\mathsf{pk}_\pi \in L$. The system produces a signature σ_0 and a simulation σ_1 with the same pseudoidentity pid, and randomly chooses to give one of them to $\mathcal{A}$. Note that for σ_1 to be created, σ_0 has to to be generated first, so that the pid can be extracted. $\mathcal{A}$ then must guess if they received the signature or the simulation. Thus:

Game 1.4: Non-Transferability experiment $\mathsf{Exp}^{\text{trans}}_{\mathcal{A},\Pi,n}$

```
Input  : $\lambda$
Output: $\{0,1\}$
$\mathsf{params} \leftarrow \Pi.\mathsf{Setup}(1^\lambda)$
$\mathcal{U} \leftarrow \big\{(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \Pi.\mathsf{KGen}()\big\}_{i=1}^{n}$
$(L = \{\mathsf{pk}_i\}_{i=1}^{n_L}, \mathtt{m}, \mathsf{pk}_D, \mathsf{pk}_\pi) \leftarrow \mathcal{A}^{\mathcal{RO},\mathcal{JO}}(\textbf{choose}, \mathcal{U})$
$\sigma_0 \leftarrow \Pi.\mathsf{Sign}(L, \mathtt{m}, \mathsf{pk}_D, \mathsf{sk}_\pi)$
$\mathsf{pid}_0 \leftarrow \Pi.\mathsf{Extract}(\sigma_0)$
$\sigma_1 \leftarrow \Pi.\mathsf{Sim}(L, \mathtt{m}, \mathsf{pk}_D, \mathsf{sk}_D, \mathsf{pid}_0)$
$b \leftarrow\!\$\, \{0,1\}$
$b' \leftarrow \mathcal{A}^{\mathcal{RO},\mathcal{JO}}(\textbf{guess}, L, \mathtt{m}, \sigma_b)$
return $b = b'$
```

Definition 5. *Non-Transferability*

A DVLRS scheme Π is perfectly non-transferable if for any unbounded adversary $\mathcal{A}$:

$$\mathsf{Adv}^{\text{trans}}_{\mathcal{A}}(\lambda) = \Pr\big[\mathsf{Exp}^{\text{trans}}_{\mathcal{A},\Pi,n}(\lambda) = 1\big] - \frac{1}{2} = 0$$

It is worth noting that in $\mathsf{Exp}^{\text{trans}}_{\mathcal{A},\Pi,n}$, $\mathcal{A}$ has to distinguish between a signature and a simulation for the same pid. A more general security experiment would be for the system to randomly select a signer index $\pi \leftarrow\!\$\, [n_L]$ to generate $\sigma_0 \leftarrow \Pi.\mathsf{Sign}(L, \mathtt{m}, \mathsf{pk}_D, \mathsf{sk}_\pi)$ and randomly select a pid to generate $\sigma_1 \leftarrow \Pi.\mathsf{Sim}(L, \mathtt{m}, \mathsf{pk}_D, \mathsf{sk}_D, \mathsf{pid})$. $\mathcal{A}$ would again have to guess if they received the signature or the simulation. This stronger requirement however, is not needed to capture the intuitive notion of non-tranferability.

3 A DVLRS Construction

Our construction builds upon the signatures of [15], by adding a designated verifier capable of simulating signatures. Intuitively, the signature generation algorithm takes as input the public key of the designated verifier, apart from the public keys of the ring members. In effect, this means that a signature is a proof of knowledge of *either* a secret signing key *or* the secret designated verifier key. A 'real' signature is obtained from knowing the former, while a simulated signature from knowing the latter.

3.1 Setup

Our scheme operates in a group $\mathbb{G}$ of prime order q, where the DDH assumption holds. Messages are binary strings i.e. $\mathcal{MSG} = \{0,1\}^*$. The pseudoidentities are computed in $\mathbb{G}$, that is, $\mathcal{PID} = \mathbb{G}$. We assume that each signer has a credential consisting of a private part and its public counterpart. In particular, we consider n_L signers with private keys $\{\mathsf{sk}_i = x_i\}_{i=1}^{n_L} \in \mathbb{Z}_q$ and corresponding public keys

$\{\mathsf{pk}_i = y_i = g^{x_i}\}_{i=1}^{n_L} \in \mathbb{G}$. Messages are encoded as group elements $\mathtt{m} \in \mathbb{G}$. We assume two random oracles $\mathsf{H}_G, \mathsf{H}_q$ that map binary strings to $\mathbb{G}, \mathbb{Z}_q$ respectively.

3.2 Signature

The signer decides on a message $\mathtt{m}$ and signs it using DVLRS. Signature verification is public, but tied to a specific verifier identified by a key. The ring L consists of n_L public keys, namely $L = \{y_i\}_{i=1}^{n_L}$. The signer's index is π. We denote the designated verifier's private key by x_D and public key by $y_D = g^{x_D}$.

Signing In order to generate the signature for message $\mathtt{m}$, the signer invokes the $\mathsf{Sign}(L, \mathtt{m}, \mathsf{pk}_D, \mathsf{sk}_\pi)$ algorithm:

- The signer computes $h \leftarrow \mathsf{H}_{\mathbb{G}}(L)$ and $\hat{y} \leftarrow h^{x_\pi}$ as the pseudoidentity.
- The signer picks $u, w_\pi, r_\pi \leftarrow\$\, \mathbb{Z}_q$ uniformly at random and computes:

$$c_{\pi+1} \leftarrow \mathsf{H}_q(L, \hat{y},\, y_D,\, g^u,\, h^u,\, g^{w_\pi} y_D^{r_\pi},\, \mathtt{m})$$

- For $i \in \{\pi+1, ..., n_L, 1, ..., \pi-1\}$, the signer picks $s_i, w_i, r_i \leftarrow\$\, \mathbb{Z}_q$ and computes:

$$c_{i+1} \leftarrow \mathsf{H}_q(L,\, \hat{y},\, y_D,\, g^{s_i} y_i^{c_i+w_i},\, h^{s_i} \hat{y}^{c_i+w_i},\, g^{w_i} y_D^{r_i},\, \mathtt{m})$$

- Finally, the signer sets $s_\pi \leftarrow u - (c_\pi + w_\pi)x_\pi$.
- The signature is $\sigma = (c_1,\, \{s_i\}_{i=1}^{n_L},\, \{w_i\}_{i=1}^{n_L},\, \{r_i\}_{i=1}^{n_L},\, \hat{y})$

It is obvious from the form of the signature that $\mathsf{Extract}(\sigma) = \hat{y}$.

Verification To verify the signature $\sigma = (c_1,\, \{s_i\}_{i=1}^{n_L},\, \{w_i\}_{i=1}^{n_L},\, \{r_i\}_{i=1}^{n_L},\, \hat{y})$ a public verifier invokes the $\mathsf{Vrfy}(\sigma, L, \mathtt{m}, y_D)$ algorithm which:

- Recomputes $h \leftarrow \mathsf{H}_{\mathbb{G}}(L)$.
- For all ring members indexed by $i \in [n_L]$ it computes:

$$z'_i \leftarrow g^{s_i} y_i^{c_i+w_i},\; z''_i \leftarrow h^{s_i} \hat{y}^{c_i+w_i},\; z'''_i \leftarrow g^{w_i} y_D^{r_i},$$
$$c_{i+1} \leftarrow \mathsf{H}_q(L,\, \hat{y},\, y_D,\, z'_i,\, z''_i,\, z'''_i,\, \mathtt{m})$$

- The signature verifies if and only if:

$$c_1 = \mathsf{H}_q(L,\, \hat{y},\, y_D,\, z'_n,\, z''_n,\, z'''_n,\, \mathtt{m})$$

Simulation In order to generate a simulated signature on message $\mathtt{m}$, the designated verifier invokes the $\mathsf{Sim}(L, \mathtt{m}, y_D, x_D, \hat{y})$ algorithm, for some pseudoidentity $\hat{y} \in \mathbb{G}$:

- Compute $h \leftarrow \mathsf{H}_{\mathbb{G}}(L)$.
- Pick $\alpha, \beta, s_1 \leftarrow\$\, \mathbb{Z}_q$ and compute:

$$c_2 \leftarrow \mathsf{H}_q(L,\, \hat{y},\, y_D,\, g^{s_1} y_1^{\beta},\, h^{s_1} \hat{y}^{\beta},\, g^{\alpha},\, \mathtt{m})$$

- For $i \in \{2, \ldots, n_L\}$ wrapping-up to 1, select $s_i, w_i, r_i \leftarrow\$\, \mathbb{Z}_q$ and compute:

$$c_{i+1} \leftarrow \mathsf{H}_q(L,\, \hat{y},\, y_D,\, g^{s_i} y_i^{c_i+w_i},\, h^{s_i} \hat{y}^{c_i+w_i},\, g^{w_i} y_D^{r_i},\, \mathtt{m})$$

- Set $w_1 \leftarrow \beta - c_1$ and $r_1 \leftarrow (\alpha - w_1) \cdot x_D^{-1}$.
- The simulated signature is $\sigma = (c_1,\, \{s_i\}_{i=1}^{n_L},\, \{w_i\}_{i=1}^{n_L},\, \{r_i\}_{i=1}^{n_L},\, \hat{y})$

Linking The linking algorithm $\mathsf{Link}(\sigma, L, \sigma', L)$ outputs 1 if and only if $\mathsf{Extract}(\sigma) = \mathsf{Extract}(\sigma')$.

4 Security Analysis of our construction

We now analyse the security of our construction for completeness, unforgeability (Definition 2), anonymity (Definition 3), linkability (Definition 4) and non-tranferability (Definition 5).

4.1 Completeness

Lemma 1. *An honestly generated DVLRS σ verifies correctly.*

Proof. It suffices to show that $z'_\pi = g^u$ and $z''_\pi = h^u$. Indeed:

$$\begin{aligned} z_\pi \; &= g^{s_\pi} y_i^{c_\pi + w_\pi} = g^{u - x_\pi (c_\pi + w_\pi)} y_\pi^{c_\pi + w_\pi} = g^u \\ z''_\pi &= h^{s_\pi} \hat{y}^{c_\pi + w_\pi} = h^{u - x_\pi (c_\pi + w_\pi)} \hat{y}^{c_\pi + w_\pi} = h^u \end{aligned}$$

□

Lemma 2. *A simulated DVLRS σ verifies correctly.*

Proof. It suffices to show that $z'_1 = g^{s_1} {y_1}^\beta$ and $z''_1 = h^{s_1} \hat{y}^\beta$ and $z'''_1 = g^a$. Indeed:

$$\begin{aligned} z'_1 &= g^{s_1} y_1^{c_1 + w_1} = g^{s_1} y_1^{c_1 + \beta - c_1} = g^{s_1} y_1^\beta \\ z''_1 &= h^{s_1} \hat{y}^{c_1 + w_1} = h^{s_1} \hat{y}^{c_1 + \beta - c_1} = h^{s_1} \hat{y}^\beta \\ z'''_1 &= g^{w_1} y_D^{r_1} = g^{w_1} g^{x_D (a - w_1) x_D^{-1}} = g^a \end{aligned}$$

□

Lemma 3. *Our DVLRS scheme has linking correctness.*

Proof. Assume two signatures σ, σ' created on the same ring L. If they are honestly generated from the same signer π then $\mathsf{Extract}(\sigma) = \mathsf{Extract}(\sigma') = \hat{y} = h^{x_\pi}$ which means that $\mathsf{Link}(\sigma, L, \sigma', L) = 1$. If σ is honestly generated from signer π and σ' is a simulation for $\hat{y} = h^{x_\pi}$ then by construction $\mathsf{Link}(\sigma, L, \sigma', L) = 1$. The same applies to the case of two honest simulations. □

Theorem 1. *Our DVLRS scheme has verification and linking correctness.*

Proof. A direct consequence of Lemma 1, Lemma 2, Lemma 3. □

4.2 Unforgeability

Theorem 2 (Unforgeability). *Our DVLRS scheme is unforgeable in the $\mathcal{RO}$ model if DLP is hard in $\mathbb{G}$.*

The proof for Theorem 2 employs techniques from [15, 13, 19, 6].

Proof. Assume a PPT adversary $\mathcal{A}$ which makes at most q_H queries to H_q and $\mathsf{H}_{\mathbb{G}}$ combined, and at most q_O queries to $\mathcal{JO}$, $\mathcal{CO}$, $\mathcal{SO}$ and $\mathcal{MO}$ combined and $\mathsf{Adv}^{\text{unf}}_{\mathcal{A}}(\lambda) > \mathsf{negl}(\lambda)$. We will create an algorithm $\mathcal{M}$, that given as an input a generator $g \in \mathbb{G}$ and n_0 DLP instances $\{y_i\}_{i=1}^{n_0}$,outputs the discrete logarithm of at least one of them, i.e. a x_j such that $g^{x_j} = y_j$ for some $j \in [n_0]$ by using $\mathcal{A}$ as a subroutine, and therefore providing us with the desired contradiction. $\mathcal{M}$ sets as params $\mathbb{G}, g, q$ and $\mathcal{U} = \{y_i\}_{i=1}^{n_0}$ as the initial set of public keys, and gives them to $\mathcal{A}$. Whenever $\mathcal{A}$ queries one of the oracles, $\mathcal{M}$ will answer as below:

- H_q: $\mathcal{M}$ outputs $r \leftarrow\$\, \mathbb{Z}_q$.
- $\mathsf{H}_{\mathbb{G}}$: $\mathcal{M}$ calculates $r \leftarrow\$\, \mathbb{Z}_q$ and outputs g^r.
- $\mathcal{JO}$: $\mathcal{M}$ calculates $r \leftarrow\$\, \mathbb{Z}_q$ and adds g^r to $\mathcal{U}$.
- $\mathcal{CO}$: $\mathcal{M}$ on a query for a $y_j \notin \{y_i\}_{i=1}^{n_0}$ outputs the corresponding secret key r_j such as $g^{r_j} = y_j$. Otherwise halts for queries of $y_j \in \{y_i\}_{i=1}^{n_0}$.
- $\mathcal{SO}$: $\mathcal{A}$ gives $\mathcal{M}$, $L \subset \mathcal{U}$,a message $\mathtt{m} \in \mathbb{G}$,$y_\pi \in L$ and $y_D \in \mathcal{U}$. Let $g^r = h = \mathsf{H}_{\mathbb{G}}(L)$. If $y_\pi \notin \{y_i\}_{i=1}^{n_0}$ $\mathcal{M}$ knows r_π such that $g^{r_\pi} = y_\pi$ and computes $\sigma \leftarrow \mathsf{Sign}(L, \mathtt{m}, y_D, r_\pi)$, while maintaining consistencies for H_q and $\mathsf{H}_{\mathbb{G}}$. It outputs σ. Otherwise it chooses randomly $\{c_i\}_{i=1}^{n_L}, \{w_i\}_{i=1}^{n_L}, \{r_i\}_{i=1}^{n_L}, \{s_i\}_{i=1}^{n_L} \leftarrow\$\, \mathbb{Z}_q$ and computes $\hat{y} \leftarrow y_\pi^r$. For each $i \in [n_L]$ back patch to:
$$c_{i+1} \leftarrow \mathsf{H}_q(L,\, \hat{y},\, y_D,\, g^{s_i} y_i^{c_i+w_i},\, h^{s_i}\hat{y}^{c_i+w_i},\, g^{w_i} y_D^{r_i},\, \mathtt{m})$$
and output $\sigma = (c_1,\, \{s_i\}_{i=1}^{n_L},\, \{w_i\}_{i=1}^{n_L},\, \{r_i\}_{i=1}^{n_L},\, \hat{y})$. Note that this looks just like a signature generated by ring member with public key y_π.
- $\mathcal{MO}$: $\mathcal{A}$ gives $\mathcal{M}$, $L \subseteq \mathcal{U}$,message $\mathtt{m} \in \mathbb{G}$,$y_D \in \mathcal{U}$ and $\hat{y} \in \mathbb{G}$. Let $g^r = h = \mathsf{H}_{\mathbb{G}}(L)$. If $y_D \notin \{y_i\}_{i=1}^{n_0}$ $\mathcal{M}$ knows r_D such that $g^{r_D} = y_D$ and computes $\sigma \leftarrow \mathsf{Sim}(L, \mathtt{m}, y_D, r_D, \hat{y})$, while keeping consistencies for H_q and $\mathsf{H}_{\mathbb{G}}$. It outputs σ. Otherwise it chooses randomly $\{c_i\}_{i=1}^{n_L}, \{w_i\}_{i=1}^{n_L}, \{r_i\}_{i=1}^{n_L}, \{s_i\}_{i=1}^{n_L} \leftarrow\$\, \mathbb{Z}_q$. For each $i \in [n_L]$ back patch to:
$$c_{i+1} \leftarrow \mathsf{H}_q(L,\, \hat{y},\, y_D,\, g^{s_i} y_i^{c_i+w_i},\, h^{s_i}\hat{y}^{c_i+w_i},\, g^{w_i} y_D^{r_i},\, \mathtt{m})$$
and output $\sigma = (c_1,\, \{s_i\}_{i=1}^{n_L},\, \{w_i\}_{i=1}^{n_L},\, \{r_i\}_{i=1}^{n_L},\, \hat{y})$. Note that this looks just like a simulation generated by designated verifier with public key y_D and pseudoidentity $\hat{y}$.

We can assume that whenever $\mathcal{A}$ outputs a successful forgery, it has queried to the random oracles all of the n_L queries used in the Vrfy algorithm. It is trivial to show that if it had not, it would have only $\mathsf{negl}(\lambda)$ probability of success.
Also without loss of generality we can assume that successful forgeries will have $L \subseteq \{y_i\}_{i=1}^{n_0}$ and $y_D \in \{y_i\}_{i=1}^{n_0}$. Let $\{X_i\}_{i=i_1}^{i_{n_L}}$ denote the first time each of the

queries used in Vrfy appear in the transcript of $\mathcal{A}$. We call a successful forgery σ an (l, π)-forgery if $i_1 = l$ and

$$X_{i_{n_L}} = \mathsf{H}_q(L,\, \hat{y},\, y_D,\, g^{s_{\pi-1}} y_{\pi-1}^{c_{\pi-1}+w_{\pi-1}},\, h^{s_{\pi-1}} \hat{y}^{c_{\pi-1}+w_{\pi-1}},\, g^{w_{\pi-1}} y_D^{r_{\pi-1}},\, \mathtt{m})$$

Since $1 \leq l \leq q_H + n_L q_O$ and $1 \leq \pi \leq n_L$, there exist some l, π such that the probability that $\mathcal{A}$ produces a successful (l, π)-forgery is non negligible.
$\mathcal{M}$ will do a rewind simulation for each value of l and π. From the Rewind on Success Lemma[15] it will obtain with non negligible probability two successful (l, π)-forgeries σ, σ' with:

$$g^u = g^{s_\pi} y_\pi^{c_\pi + w_\pi} = g^{s_\pi + x_\pi(c_\pi + w_\pi)} \tag{1}$$

$$h^v = h^{s_\pi} \hat{y}^{c_\pi + w_\pi} = h^{s_\pi + x(c_\pi + w_\pi)} \tag{2}$$

$$g^\nu = g^{w_\pi} y_D^{r_\pi} = g^{w_\pi + x_D \cdot r_\pi} \tag{3}$$

$$g^u = g^{s'_\pi} y_\pi^{c'_\pi + w'_\pi} = g^{s'_\pi + x_\pi(c'_\pi + w'_\pi)} \tag{4}$$

$$h^v = h^{s'_\pi} \hat{y}^{c'_\pi + w'_\pi} = h^{s'_\pi + x(c'_\pi + w'_\pi)} \tag{5}$$

$$g^\nu = g^{w'_\pi} y_D^{r'_\pi} = g^{w'_\pi + x_D \cdot r'_\pi} \tag{6}$$

Since $c_\pi \neq c'_\pi$ it holds that $s_\pi \neq s'_\pi$ or $w_\pi \neq w'_\pi \wedge r_\pi \neq r'_\pi$.

- If $s_\pi \neq s'_\pi$: solving 1, 4 yields:

$$x_\pi = \frac{s'_\pi - s_\pi}{c_\pi - c'_\pi + w_\pi - w'_\pi} \mod q$$

- If $w_\pi \neq w'_\pi \ \wedge \ r_\pi \neq r'_\pi$: solving 3, 6 yields:

$$x_D = \frac{w'_\pi - w_\pi}{r_\pi - r'_\pi} \mod q$$

$\mathcal{M}$ has solved at least one hard DLP instance, a contradiction. □

4.3 Anonymity

Theorem 3 (Anonymity). *Our DVLRS scheme is anonymous (signer ambiguous) in the $\mathcal{RO}$ model if the DDH assumption holds in $\mathbb{G}$.*

To prove Theorem 3 we adapt the proofs of [15, 16] for our scheme. In particular, assume a PPT adversary $\mathcal{A}$ which succeeds in the experiment $\mathsf{Exp}_{\mathcal{A},\Pi,n,t}^{\mathrm{Anon}}$ with $\mathsf{Adv}_{\mathcal{A}}^{\mathrm{anon}}(\lambda)$ at least ϵ after at most n queries to $\mathcal{JO}$, $q_{\mathsf{H}_\mathbb{G}}$ queries to $\mathsf{H}_\mathbb{G}$ and running time T. Then there exists an algorithm $\mathcal{M}$ that breaks the DDH assumption for $\mathbb{G}$ in time at most $n q_{\mathsf{H}_\mathbb{G}} T$ with probability at least $\frac{1}{2} + \frac{\epsilon}{4}$.

Proof. We construct $\mathcal{M}$. Its input will be the group $\mathbb{G}$ (of order q) and a tuple of elements $A_\beta, B_\beta, C_\beta \in \mathbb{G}$. Assume that $A_\beta = g^a, B_\beta = g^b$ for $a, b \in \mathbb{Z}_q$ unknown to $\mathcal{M}$. Its output will be a bit β indicating if $C_\beta = g^{ab}$, ($\beta = 1$) or not ($\beta = 0$).

$\mathcal{M}$ begins by simulating $\mathcal{JO}$. For $y_\pi \in L$ to hold, it follows that $\mathcal{A}$ has queried $\mathcal{JO}$ for it. $\mathcal{M}$ selects a random $\mathcal{A}$ query and substitutes $y_\pi = A_\beta$. All other such queries are answered with a random $y \leftarrow\!\$\, \mathbb{G}$. In the same manner $\mathcal{M}$ randomly selects one of the queries to $\mathsf{H}_\mathbb{G}$ and returns $h = B_\beta$. All other queries to $\mathsf{H}_\mathbb{G}$ are simulated by returning $h = g^k, k \leftarrow\!\$\, \mathbb{Z}_q$. The answer to an $\mathcal{SO}(L', \mathtt{m}', y_D, y)$ query for some $L' = \{y_i\}_{i=1}^{n'_L}$ is simulated as:

- Select $\pi' \leftarrow\!\$\, n_{L'}$
- If $h \neq B_\beta$ then set $\hat{y} = y_{\pi'}^k$
- If $h = B_\beta$ and $y_\pi = A_\beta$ then set $\hat{y} = C_\beta$
- If $h = B_\beta$ and $y_\pi \neq A_\beta$ then set $\hat{y} = B_\beta^{x_{\pi'}}$
- Select $c_{\pi'}, \{s_i, w_i, r_i\}_{i=1}^{n_{L'}} \leftarrow\!\$\, \mathbb{Z}_q$ and for $i \in \{\pi', \cdots, n_{L'}, 1, \cdots, \pi' - 1\}$ compute $\left\{c_{i+1} \leftarrow \mathsf{H}_q(L',\, \hat{y},\, y_D,\, g^{s_i} y_i^{c_i + w_i},\, h^{s_i} \hat{y}^{c_i + w_i},\, g^{w_i} y_D^{r_i},\, \mathtt{m}')\right\}$
- Set $c_{\pi'} \leftarrow \mathsf{H}_q(L',\, \hat{y},\, y_D,\, g^{s_{\pi'-1}} y_{\pi'-1}^{c_{\pi'-1} + w_{\pi'-1}},\, h^{s_{\pi'-1}} \hat{y}^{c_{\pi'-1} + w_{\pi'-1}},\, g^{w_{\pi'-1}} y_D^{r_{\pi'-1}},\, \mathtt{m}')$.

When $\mathcal{M}$ receives $(L, \mathtt{m}, D_t)$ it checks if $y_\pi \in L$ and $\pi \notin D_t$, which occurs with probability $\frac{n_L - t}{n}$ and that there exists a query for L in $\mathsf{H}_\mathbb{G}$, which is true with probability $\frac{1}{q_\mathsf{H}}$. Otherwise it halts. If $\mathsf{H}_\mathbb{G}(L)$ has not been queried with L, then it sets $B_\beta = \mathsf{H}_\mathbb{G}(L)$. Then $\mathcal{M}$ generates the challenge signature σ. The $\mathcal{CO}$ calls are answered faithfully, except if x_π is requested. Then $\mathcal{M}$ returns $\perp$. If the **guess** stage is successfully completed, $\mathcal{A}$ returns $\xi \in [n_L]$. If $\xi = \pi$ then $\mathcal{M}$ returns 1. Otherwise $\mathcal{M}$ selects uniformly at random from $\{0, 1\}$.

In the case of a DDH tuple ($\beta = 1$), $\mathcal{A}$ succeeds in $\mathsf{Exp}_{\mathcal{A},\Pi,n,t}^{\text{Anon}}$ with probability at least $\frac{1}{n_L - t} + \epsilon$. This means:

$$\Pr[\mathcal{M}(A_\beta, B_\beta, C_\beta) = 1 | \beta = 1] \geq (\frac{1}{n_L - t} + \epsilon) + \frac{1}{2}(1 - \frac{1}{n_L - t} - \epsilon) \geq \frac{1}{2} + \frac{1}{2(n_L - t)} + \frac{\epsilon}{2}$$

In the case of a non-DDH tuple ($\beta = 0$), if $\xi = \pi$ then $\mathcal{M}$ cannot return 0. Otherwise, it selects its output uniformly at random:

$$\Pr[\mathcal{M}(A_\beta, B_\beta, C_\beta) = 0 | \beta = 0] = (\frac{1}{n_L - t}) \cdot 0 + (1 - \frac{1}{n_L - t}) \cdot \frac{1}{2} = \frac{1}{2} - \frac{1}{2(n_L - t)}$$

As a result: $\Pr[\mathcal{M}(A_\beta, B_\beta, C_\beta) = \beta] \geq \frac{1}{2} + \frac{\epsilon}{4}$, a contradiction.

Regarding the running time $\mathcal{M}$ halts with probability $\frac{n_L - t}{n} \cdot \frac{1}{q_H}$. Thus after at most $\frac{n q_H}{n_L - t}$ executions $\mathcal{A}$ will have succeeed once. If the running time of $\mathcal{A}$ is at most T on success, $\mathcal{M}$ requires at most $n q_h T$ steps. □

4.4 Linkability

Theorem 4 (Linkability). *Our DVLRS scheme is linkable in the $\mathcal{RO}$ model if DLP is hard in $\mathbb{G}$.*

Proof. We adapt the techniques of [15, 19, 6] for our stronger definition. Assume a PPT adversary $\mathcal{A}$ which makes at most q_H queries to H_q and $H_\mathbb{G}$ combined, and at most q_O queries to $\mathcal{JO}, \mathcal{CO}, \mathcal{SO}$ and $\mathcal{MO}$ combined, with $\mathsf{Adv}_{\mathcal{A}}^{\text{link}}(\lambda) > \mathsf{negl}(\lambda)$. We will create and algorithm $\mathcal{M}$ that given as input a group $\mathbb{G}$ of order q, and n_0 DLP instances $\{y_i\}_{i=1}^{n_0}$ outputs the discrete logarithm of at least one of them, i.e a x_j such that $g^{x_j} = y_j$ for some $j \in [n_0]$ by using $\mathcal{A}$, and therefore providing us with the desired contradiction on the assumption that DPL is hard. $\mathcal{M}$ sets as params $\mathbb{G}, g, q$ and $\mathcal{U} = \{y_i\}_{i=1}^{n_0}$ as the initial set of public keys, and gives them to $\mathcal{A}$. $\mathcal{M}$ simulates the oracle calls of $\mathcal{A}$ as in the proof of Theorem 2.

After a successful run, $\mathcal{A}$ will have output k signatures $\{\sigma_i\}_{i=1}^{k}$ that are pairwise unlinkable, for a ring $L \subseteq \mathcal{U}$ of its choice, for which it has corrupted less than k keys and has not corrupted any of the designated verifier keys $\{y_{D_i}\}_{i=1}^{k}$. W.l.o.g $\{y_{D_i}\}_{i=1}^{k} \subset \{y_i\}_{i=1}^{n_0}$ and at least one of the $y_i \in L$ for $i \in [n_0]$. The adversary must have, with negligible exception, queried to the random oracles all of the queries used in the Vrfy algorithm. So following the notation of the unforgeability proof in subsection 4.2, these will be (l_i, π_i)-forgeries for some values of $0 < l_i < q_H + n_L q_O$ and $1 < \pi_i < n_L$ for all $i \in [k]$.

We can distinguish 2 cases:

Case 1: $\mathcal{A}$ produces,with negligible exception, signatures tuples with less than k distinct π_i. Therefore there will be at least one pair of signatures that are (l_a, π)-forgery and (l_b, π)-forgery for the same value π and w.l.o.g $l_a < l_b$. $\mathcal{M}$ will do a rewind simulation to the l_a'th query, and by the Rewind on Success Lemma[15], will get with non negligible probability $\{\sigma_i'\}_{i=a}^{k}$ with σ_a' being an (l_a, π)-forgery.

As in the proof of Theorem 2, we can derive from equations 1,2,3,4,5,6 that:

- If $s_\pi \neq s_\pi'$: solving 1, 2, 4, 5 yields:

$$x_\pi = x = \frac{s_\pi' - s_\pi}{c_\pi - c_\pi' + w_\pi - w_\pi'} \mod q$$

- If $w_\pi \neq w_\pi' \wedge r_\pi \neq r_\pi'$: solving 3, 6 yields:

$$x_D = \frac{w_\pi' - w_\pi}{r_\pi - r_\pi'} \mod q$$

This means that either the pseudoidentity of σ_a is $\hat{y}_a = h^{x_\pi}$ or the discrete logarithm of y_{D_a} is solved.

Now $\mathcal{M}$ does a rewind of the first transcript of $\mathcal{A}$ to the l_b'th query, and similarly ether solves the discrete logarithm of y_{D_b} or the pseudoidentity of σ_b is $\hat{y}_b = h^{x_\pi}$. However that means that $\mathsf{Link}(\sigma_a, L, \sigma_b, L) = 1$ which contradicts our assumption that the signatures $\mathcal{A}$ outputs are pairwise unlinkable.

Case 2: $\mathcal{A}$ produces signatures with k distinct π_i. $\mathcal{M}$ will do k rewind simulations, to the l_i'th query for every $i \in [k]$ and similarly to the unforgeability proof in subsection 4.2 will each time solve the discrete logarithm of y_{D_i} or y_{π_i}. Solving even one of the y_{D_i} is enough, since we assumed that $\{y_{D_i}\}_{i=1}^{k} \subset \{y_i\}_{i=1}^{n_0}$. Otherwise the discrete logarithm of every $y \in L$ is found, and since we assumed that at least one of the $y_i \in L$ for $i \in [n_0]$, again $\mathcal{M}$ has won. □

4.5 Non-Transferability

Theorem 5 (Non-Transferability). *Our DVLRS scheme is perfectly non-transferable in the $\mathcal{RO}$ model.*

Proof. We argue that the distributions of Sign and Sim for the same message $\mathtt{m}$, ring L, designated verifier public key y_D and pseudoidentity $\hat{y}$ are the same.

We can look at each part of the signature σ_0 and simulation σ_1 separately: c_1 for both is the output of the random oracle H_q with at least one part of it's input chosen at random. Thus, c_1 is distributed uniformly at random in $\mathbb{Z}_q$.

All of the s_i are chosen at random for σ_1. For σ_0 all but s_π are also chosen at random. However $s_\pi \leftarrow u - (c_\pi + w_\pi)x_\pi$ with u being a random value, therefore s_π is also a uniformly random value in $\mathbb{Z}_q$. With a similar argument it can be shown that the values r_i and w_i are also distributed uniformly at random in $\mathbb{Z}_q$ for both the signature and the simulation.

The only remaining part of the signature is the pseudoidentity $\hat{y}$ and this will be the exact same group element for both σ_0 and σ_1. Note that the verifier can only produce such a simulation after first having seen a real signature from a ring member with that given pseudoidentity. This however gives no advantage to an adversary who tries to distinguish a signature from a simulation.

It is clear that $\mathcal{A}$ in $\mathsf{Exp}^{\text{trans}}_{\mathcal{A},\Pi,n}$ is given at random one of two valid σ_0, σ_1 which follow the same distribution. Thus $\mathcal{A}$ cannot do better than a random guess. □

Another property that we can prove for our scheme, is that an adversary that is given a simulation or a signature that have a random pseudoidentity, cannot distinguish them with non-negligible advantage. This additional property can be useful in some applications to give the designated verifier more freedom on creating 'noise'. Our scheme has this property, albeit only against a computationally bounded adversary. We omit the details.

5 Applications

DVLRS schemes have a number of useful applications which benefit from the novel combination of anonymity, linkability and non-transferability. Concretely, anonymous surveys or feedback systems satisfy the need for quality improvement by involving anonymous opinions of reviewers. Such systems are used, for instance, in educational institutions for instructor evaluation. However, the possibility of negative reviews, exacerbated by the anonymity of the reviewers, might hinder adoption. DVLRS allow for the evaluation to keep only its intrinsic value for the instructors by enabling them, as designated verifiers, to add 'noise' to the reviews using messages with simulated signatures. Thus the reviewees will be able to improve themselves and at the same time avoid repercussions for negative reviews. This feature will make adoption of such systems easier.

In particular, by using DVLRS, reviewers can form rings according to organizational characteristics (e.g. a course) and anonymously submit authentic feedback using the Sign algorithm. The reviewees, on the other hand, will be able to group feedback signed by the same reviewer, using the Link algorithm. The linkability property, will allow the formation of a consistent view of individual opinions and their evolution through the course of time. Thus, instructors, as the designated verifiers, will be able to adapt their techniques and monitor the results. At the same time, they will be able to simulate signatures using the Sim algorithm and link them to the pseudoidentity of a reviewer, or to a random one. By the non-transferability property of the designated verifier signature, only the reviewee is aware which of the feedback were signed by the reviewers, and the results of the feedback are not transferable to a third party - i.e. a higher authority. Even if the private key of the designated verifier or any of the signers is compromised, it would still be impossible to distinguish which messages were generated by the designated verifier and which by the ring members. So, as we discussed in subsection 1.1, removing public verifiability cannot provide any help in this scenario.

Another use case of the DVLRS scheme is for enhanced privacy for leaking secrets, an application that served as the original motivation for ring signatures in [20]. As an example, assume that a member of a corrupted organisation wants to leak information to the authorities without revealing his identity. Using a plain ring signature as in [20], could result in the corrupted leadership punishing all the ring members indiscriminately. With DVLRS the information can be safely leaked by setting the authorities as the designated verifier so that there can be no proof of any leak. Now the signature cannot be used to convince the public, but the law enforcement can still use the information to initiate an investigation. Additionally, linkability helps the informant give updates on his information.

Similar applications, that benefit from the non-transferability property of the designated verifier apart from anonymity and linkability, can be found in protecting databases that contain sensitive data (such as medical or financial records). In many such cases, anonymity in the identities of the participants is not enough, since such data is extremely valuable or the subjects can be identified from the content of the messages. For instance, in financial surveys the participating companies usually submit fiscal data, which can be collected and correlated to other public sources. DVLRS schemes defend against such attacks, as the simulated signatures cast doubt on the authenticity of such data to everyone except the designated verifier. For the same reason, they add another layer of protection for the survey participants, as the ability to simulate signatures indistinguishable to the original makes buyers unsure of whether they are paying for authentic data. Furthermore, using the linkability property all the data belonging to a single entity could be linked, speeding up data retrieval and change tracking in the survey data for any participant.

6 Conclusion and Future Work

In further work, we plan to implement DVLRS and integrate it with an anonymous evaluation system, as described in section 5. We also intend to explore different constructions of DVLRS that improve their security, functionality and efficiency. Concretely, our instantiation has signatures with size that scales linearly to that of the ring. We plan to make a construction with constant size signatures as in [23, 1], while retaining all the desired security properties. Another direction we aim to explore, is an alternative construction that achieves unconditional anonymity in a manner similar to [13].

Finally, we plan to consider the possibility of modifying the semantics of non-transferability so that it treats ring members in a privileged manner, by allowing them to distinguish which signatures actually come from other ring members and which are simulations. This will open up new applications for DVLRS.

References

[1] Man Ho Au, Sherman S. M. Chow, Willy Susilo, and Patrick P. Tsang. "Short Linkable Ring Signatures Revisited". In: *Third European PKI Workshop: EuroPKI 2006,* vol. 4043. LNCS. Springer, 2006, pp. 101–115. DOI: 10.1007/11774716_9.

[2] Adam Bender, Jonathan Katz, and Ruggero Morselli. "Ring Signatures: Stronger Definitions, and Constructions without Random Oracles". In: *J. Cryptol.* 22.1 (2009), pp. 114–138.

[3] David Chaum and Eugène van Heyst. "Group Signatures". In: *EUROCRYPT 91.* Vol. 547. LNCS. Springer, 1991, pp. 257–265. DOI: 10.1007/3-540-46416-6_22.

[4] Guomin Chen, Chunhui Wu, Wei Han, Xiaofeng Chen, Hyunrok Lee, and Kwangjo Kim. "A New Receipt-Free Voting Scheme Based on Linkable Ring Signature for Designated Verifiers". In: *2008 International Conference on Embedded Software and Systems Symposia.* 2008, pp. 18–23. DOI: 10.1109/ICESS.Symposia.2008.54.

[5] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. "Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols". In: *CRYPTO 94.* Vol. 839. LNCS. Springer, 1994, pp. 174–187.

[6] Javier Herranz and Germán Sáez. "Forking Lemmas for Ring Signature Schemes". In: *INDOCRYPT 03.* Vol. 2904. LNCS. Springer, 2003, pp. 266–279. DOI: 10.1007/978-3-540-24582-7_20.

[7] Markus Jakobsson, Kazue Sako, and Russell Impagliazzo. "Designated Verifier Proofs and Their Applications". In: *EUROCRYPT 96.* Vol. 1070. LNCS. Springer, 1996, pp. 143–154.

[8] Ari Juels, Dario Catalano, and Markus Jakobsson. "Coercion-resistant electronic elections". In: *WPES 2005.* ACM, 2005, pp. 61–70. DOI: 10.1145/1102199.1102213.

[9] Ji-Seon Lee and Jik Hyun Chang. "Strong Designated Verifier Ring Signature Scheme". In: *Innovations and Advanced Techniques in Computer and Information Sciences and Engineering.* Dordrecht: Springer Netherlands, 2007, pp. 543–547. ISBN: 978-1-4020-6268-1.

[10] Jin Li and Yanming Wang. "Universal Designated Verifier Ring Signature (Proof) Without Random Oracles". In: *Emerging Directions in Embedded*

and Ubiquitous Computing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 332–341. ISBN: 978-3-540-36851-9.

[11] Yong Li, Willy Susilo, Yi Mu, and Dingyi Pei. "Designated Verifier Signature: Definition, Framework and New Constructions". In: *Ubiquitous Intelligence and Computing*. Vol. 4611. LNCS. Springer, 2007, pp. 1191–1200. DOI: 10.1007/978-3-540-73549-6_116.

[12] Helger Lipmaa, Guilin Wang, and Feng Bao. "Designated Verifier Signature Schemes: Attacks, New Security Notions and a New Construction". In: *ICALP*. Vol. 3580. LNCS. Springer, 2005, pp. 459–471.

[13] Joseph K. Liu, Man Ho Au, Willy Susilo, and Jianying Zhou. "Linkable Ring Signature with Unconditional Anonymity". In: *IEEE Trans. Knowl. Data Eng.* 26.1 (2014), pp. 157–165.

[14] Joseph K. Liu, Willy Susilo, and Duncan S. Wong. "Ring Signature with Designated Linkability". In: *Advances in Information and Computer Security, IWSEC 06,* vol. 4266. LNCS. Springer, 2006, pp. 104–119. DOI: 10.1007/11908739_8.

[15] Joseph K. Liu, Victor K. Wei, and Duncan S. Wong. "Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups (Extended Abstract)". In: *ACISP*. Vol. 3108. LNCS. Springer, 2004, pp. 325–335.

[16] Joseph K. Liu and Duncan S. Wong. "Enhanced security models and a generic construction approach for linkable ring signature". In: *Int. J. Found. Comput. Sci* 17.06 (2006), pp. 1403–1422.

[17] Joseph K. Liu and Duncan S. Wong. "Solutions to Key Exposure Problem in Ring Signature". In: *Int. J. Netw. Secur.* 6.2 (2008), pp. 170–180.

[18] Shen Noether. *Ring Signature Confidential Transactions for Monero*. Cryptology ePrint Archive, Report 2015/1098.

[19] David Pointcheval and Jacques Stern. "Security Proofs for Signature Schemes". In: *EUROCRYPT 96*. Vol. 1070. LNCS. Springer, 1996, pp. 387–398.

[20] Ronald L. Rivest, Adi Shamir, and Yael Tauman. "How to Leak a Secret". In: *ASIACRYPT 01*. Vol. 2248. LNCS. Springer, 2001, pp. 552–565. DOI: 10.1007/3-540-45682-1_32.

[21] Shahrokh Saeednia, Steve Kremer, and Olivier Markowitch. "An Efficient Strong Designated Verifier Signature Scheme". In: Jan. 2003, pp. 40–54. ISBN: 978-3-540-21376-5. DOI: 10.1007/978-3-540-24691-6_4.

[22] Ron Steinfeld, Laurence Bull, Huaxiong Wang, and Josef Pieprzyk. "Universal Designated-Verifier Signatures". In: *ASIACRYPT 03,* vol. 2894. LNCS. Springer, 2003, pp. 523–542. DOI: 10.1007/978-3-540-40061-5_33.

[23] Patrick P. Tsang and Victor K. Wei. "Short Linkable Ring Signatures for E-Voting, E-Cash and Attestation". In: *Information Security Practice and Experience*. Vol. 3439. LNCS. Springer, 2005, pp. 48–60. DOI: 10.1007/978-3-540-31979-5_5.