

On SDVS Sender Privacy In The Multi-Party Setting

Jeroen van Wier

Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg

Abstract. Strong designated verifier signature schemes rely on sender-privacy to hide the identity of the creator of a signature to all but the intended recipient. This property can be invaluable in, for example, the context of deniability, where the identity of a party should not be deducible from the communication sent during a protocol execution. In this work, we explore the technical definition of sender-privacy and extend it from a 2-party setting to an n -party setting. Afterwards, we show in which cases this extension provides a stronger security and in which cases it does not.

1 Introduction

Digital signatures have many useful applications in our everyday lives, from message authentication to software updates. In many cases, they provide a publicly verifiable way of proving the authenticity of a message. However, sometimes it is desired to prove authenticity only to the intended receiver, or designated verifier, of a message. Designated verifier signature (DVS) schemes were constructed for this reason, to allow for the signing of a message in such a way that the receiver would be fully convinced of its authenticity, but to third-party observers, the validity of the signature could be denied. Strong designated verifier signature (SDVS) schemes are the refinement of this idea, with the additional restraint that no-one but the creator and the designated verifier should be able to deduce from a signature who was the creator. While this concept has been studied extensively and is interpreted intuitively in the same way by many, the technical definitions for the property separating DVS schemes from SDVS schemes, known as sender-privacy, vary. In this work we analyze and generalize the definitions in current literature and aim to provide a universally applicable way to define this property, particularly focusing on the n -party setting. Furthermore, we prove that our general form of sender-privacy can be achieved by combining weaker forms of sender-privacy with non-transferability or unforgeability.

1.1 Related work

Chaum and van Antwerpen first introduced undeniable signatures in [CV89], which required interaction between the signer and verifier. In 1996 this requirement was removed by Chaum [Cha96] and by Jakobsson et al. [JSI96] separately,

who introduced designated verifier signatures. These formal definitions were later refined by Saeednia et al. [SKM03]. Rivest et al. introduced ring signatures in [RST01], which can be interpreted as DVS when a ring size of 2 is used, although not SDVS.

An important step was made when Laguillaumie and Vergnaud formalised *sender-privacy*, the property separating DVS from SDVS, in [LV04]. The notion of SDVS was further refined to Identity-Based SDVS by Susilo et al. [SZM04], where all private keys are issued using a master secret key (i.e. central authority). For this setting, sender-privacy was later formalized in a game-based manner by Huang et al. [Hua+06].

2 Preliminaries

We denote with $\kappa \in \mathbb{N}$ the security parameter of a scheme and implicitly assume that any algorithm that is part of a scheme is given input 1^κ , i.e. the string of κ 1's, in addition to its specified inputs. We implicitly assume that all adversaries are probabilistic polynomial-time Turing machines (PPT), although the results also hold if all adversaries are probabilistic polynomial-time quantum Turing machines (QPT). We write $[n]$ for the set $\{0, \dots, n\}$. We call a function $\varepsilon(n)$ negligible (denoted $\varepsilon \leq \text{negl}(n)$) if for every polynomial p there exists $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$ it holds that $\varepsilon(n) < \frac{1}{p(n)}$. We reserve \perp as an error symbol.

Definition 2.1. A designated verifier signature scheme (DVS scheme) is a tuple (Setup, KeyGen, Sign, Verify, Simulate) of PPT algorithms such that:

- **Setup:** Produces the public parameters of a scheme, **params**. It is implicitly assumed that these parameters are passed to the following algorithms.
- **KeyGen:** Produces a keypair (pk, sk) .
- **Sign $_{S \rightarrow V}(m) := \text{Sign}(\text{sk}_S, \text{pk}_S, \text{pk}_V, m)$:** Upon input of a sender's keypair, a verifier's public key, and a message m , produces a signature σ if all keys are valid and \perp otherwise.
- **Verify $_{S \rightarrow V}(m, \sigma) := \text{Verify}(\text{sk}_V, \text{pk}_V, \text{pk}_S, m, \sigma)$:** Upon input of a verifier's keypair, a sender's public key, a message m , and a signature σ , outputs the validity of σ (a boolean value) if all keys are valid and \perp otherwise.
- **Simulate $_{S \rightarrow V}(m) := \text{Simulate}(\text{sk}_V, \text{pk}_V, \text{pk}_S, m)$:** Upon input of a verifier's keypair, a sender's public key, and a message m , produces a simulated signature σ' .

2.1 Current definitions

The original definitions for strong verifier designation are a combination of what we currently distinguish as *non-transferability* and *sender privacy*. The following definitions are the initial attempts at defining strong verifier designation, and in their respective papers, they are accompanied by definitions for (non-strong) verifier designation, which are very much in line with the intuition behind non-transferability.

Definition 2.2 ([JSI96]). Let $(\mathcal{P}_A, \mathcal{P}_B)$ be a protocol for Alice to prove the truth of the statement Ω to Bob. We say that Bob is a JSI strong designated verifier if, for any protocol $(\mathcal{P}_A, \mathcal{P}_B, \mathcal{P}_C, \mathcal{P}_D)$ involving Alice, Bob, Cindy, and Dave, by which Dave proves the truth of some statement θ to Cindy, there is another protocol $(\mathcal{P}_C, \mathcal{P}'_D)$ such that Dave can perform the calculations of \mathcal{P}'_D , and Cindy cannot distinguish transcripts of $(\mathcal{P}_A, \mathcal{P}_B, \mathcal{P}_C, \mathcal{P}_D)$ from those of $(\mathcal{P}_C, \mathcal{P}'_D)$.

In the above definition, the intuition is that Alice proves a statement to Bob, e.g. the authenticity of a given message. Dave observes this interaction and tries to prove this observation to Cindy. However, strong designation in this sense prevents him from doing so, as any proof he could present to Cindy is indistinguishable (to Cindy) from a simulated proof.

Definition 2.3 ([SKM03]). Let $\mathcal{P}(A, B)$ be a protocol for Alice to prove the truth of the statement Ω to Bob. We say that $\mathcal{P}(A, B)$ is a SKM strong designated verifier proof if anyone can produce identically distributed transcripts that are indistinguishable from those of $\mathcal{P}(A, B)$ for everybody, except Bob.

In later work, we see the definition for strong verifier designation split. Non-transferability captures the notion that the verifier can produce signatures from anyone designated to himself, thus ensuring that no signature provides proof of signer-verifier interaction for third parties. Sender privacy adds to this that, from a signature, one cannot deduce the sender, thus allowing no third-party observer to use a signature to plausibly deduce that interaction between two parties happened.

Definition 2.4. A DVS scheme $\Pi = (\text{KeyGen}, \text{Sign}, \text{Verify}, \text{Simulate})$ is computationally non-transferable if for any adversary \mathcal{A} ,

$$\text{Adv}_{\Pi, \mathcal{A}}^{\text{NT}}(\kappa) = \Pr_{b \in \{0,1\}} [\text{G}_{\Pi, \mathcal{A}}^{\text{NT}}(\kappa, b) = b] - \frac{1}{2} \leq \text{negl}(\kappa),$$

where the game $\text{G}_{\Pi, \mathcal{A}}^{\text{NT}}$ is defined as follows:

Game 1: $\text{G}_{\Pi, \mathcal{A}}^{\text{NT}}(\kappa, b)$

```

1 params  $\leftarrow$  Setup
2  $(\text{pk}_S, \text{sk}_S) \leftarrow \text{KeyGen}$ 
3  $(\text{pk}_V, \text{sk}_V) \leftarrow \text{KeyGen}$ 
4  $(m^*, \text{state}) \leftarrow \mathcal{A}(1, \text{params}, \text{pk}_S, \text{sk}_S, \text{pk}_V, \text{sk}_V)$ 
5 if  $b = 0$  then
6    $\sigma^* = \text{Sign}(\text{sk}_S, \text{pk}_S, \text{pk}_V, m^*)$ 
7 else
8    $\sigma^* = \text{Simulate}(\text{sk}_V, \text{pk}_V, \text{pk}_S, m^*)$ 
9  $b' \leftarrow \mathcal{A}(2, \text{state}, \sigma^*)$ 
10 Output  $b'$ 
```

Definition 2.5. A DVS $\Pi = (\text{KeyGen}, \text{Sign}, \text{Verify}, \text{Simulate})$ is statistically non-transferable if for all S, V , and m , $\text{Sign}_{S \rightarrow V}(m)$ and $\text{Simulate}_{S \rightarrow V}(m)$ are statistically indistinguishable distributions.

For sender-privacy, many slightly different definitions are presented in literature. Many follow the form of Game 2, but with different oracles presented to the adversary. Note that this game is a generalized definition designed to be instantiated with a set of oracles \mathcal{O} to form the specific definitions found in literature. Besides the oracles, the game takes as parameters the security parameter κ , the number of parties n , and the challenge party index c . For each $i \in [n]$, party i is denoted P_i . P_n is designated as the verifier for the challenge. In much of the literature this game is played with 3 parties: S_0, S_1 , and V , who would here correspond with P_0, P_1 , and P_2 respectively in the $n = 2$ setting.

Game 2: $\mathsf{G}_{\Pi, \mathcal{A}, \mathcal{O}}^{\text{SendPriv}}(\kappa, n, c)$, the generalized game for sender-privacy.

- 1 $\text{params} \leftarrow \text{Setup}$
 - 2 $(\text{pk}_{P_0}, \text{sk}_{P_0}) \leftarrow \text{KeyGen}; \dots; (\text{pk}_{P_n}, \text{sk}_{P_n}) \leftarrow \text{KeyGen}$
 - 3 $(m^*, \text{state}) \leftarrow \mathcal{A}_{\text{sign}^{(1)}, \mathcal{O}_{\text{veri}}^{(1)}, \mathcal{O}_{\text{sim}}^{(1)}}(1, \text{params}, \text{pk}_{P_0}, \dots, \text{pk}_{P_n})$
 - 4 $\sigma^* = \text{Sign}_{P_c \rightarrow P_n}(m^*)$
 - 5 $c' \leftarrow \mathcal{A}_{\text{sign}^{(2)}, \mathcal{O}_{\text{veri}}^{(2)}, \mathcal{O}_{\text{sim}}^{(2)}}(2, \text{state}, \sigma^*)$
 - 6 Output c'
-

Definition 2.6 ([Hua+06]). A DVS $\Pi = (\text{KeyGen}, \text{Sign}, \text{Verify}, \text{Simulate})$ is a Hua-strong DVS if it is statistically non-transferable and for any PPT adversary \mathcal{A} ,

$$\text{Adv}_{\Pi, \mathcal{A}}^{\text{SendPriv}}(\kappa) = \Pr_{c \leftarrow \{0,1\}} [\mathsf{G}_{\Pi, \mathcal{A}}^{\text{SendPriv}}(\kappa, 2, c) = c] - \frac{1}{2} \leq \text{negl}(\kappa),$$

where $\mathsf{G}_{\Pi, \mathcal{A}}^{\text{SendPriv}}$ is played with the following oracles:

- $\mathcal{O}_{\text{sign}}^{(1)}$: Upon input (m_i, d_i) returns $\text{Sign}_{P_{d_i} \rightarrow P_2}(m_i)$ if $d_i \in \{0,1\}$ and \perp otherwise.
- $\mathcal{O}_{\text{sign}}^{(2)}$: Upon input (m_i, d_i) returns $\text{Sign}_{P_{d_i} \rightarrow P_2}(m_i)$ if $d_i \in \{0,1\}$ and $m_i \neq m^*$, and \perp otherwise.
- $\mathcal{O}_{\text{veri}}^{(1)}$: Upon input (σ_i, m_i, d_i) returns $\text{Verify}_{P_{d_i} \rightarrow P_2}(m_i)$ if $d_i \in \{0,1\}$ and \perp otherwise.
- $\mathcal{O}_{\text{veri}}^{(2)}$: Upon input (σ_i, m_i, d_i) returns $\text{Verify}_{P_{d_i} \rightarrow P_2}(m_i)$ if $d_i \in \{0,1\}$, $\sigma_i \neq \sigma^*$, and $m_i \neq m^*$, and \perp otherwise.
- $\mathcal{O}_{\text{sim}}^{(1)} = \mathcal{O}_{\text{sim}}^{(2)} = \emptyset$

In [Hua+06], Huang et al. define signer-privacy for identity-based-SDVS, a similar type of DVS where all keypairs are issued by a central authority. Here,

they allow signing queries from any party to any party, and the adversary is allowed to choose the two signer and the verifier parties. We explore this option for SDVS in Definition 4.2.

3 Bringing sender-privacy to the multi-party setting

Sender privacy is meant to provide security in the setting where an eavesdropping adversary is trying to detect the identity of the sender of a signature. In the previously presented definitions, this is modeled by a coin flip between two senders, with a fixed verifier. This way of defining sender privacy is similar to key-privacy in public-key cryptography [Bel+01]. The key difference here is that public-key ciphertexts are only related to one keypair, the receiver's. However, designated verifier signatures are bound to two parties, the signer and the designated verifier. This creates the problem that the naive way of defining sender-privacy does not cover any attacks that require multiple parties. In key-privacy, any adversary requiring n parties for their attack can perform this attack in the two-party setting by simulating the other $n - 2$ parties themselves. However, in the case of SDVS schemes, this is not necessarily possible. The adversary could be unable to create signatures signed by one of the two challenge parties with their simulated parties as the verifier, as is depicted in Figure 1. In particular if one does not have statistical non-transferability, this might pose a problem. For this reason, we explicitly shape our definition for the multi-party setting. We explore settings where this is a non-issue in Section 5.

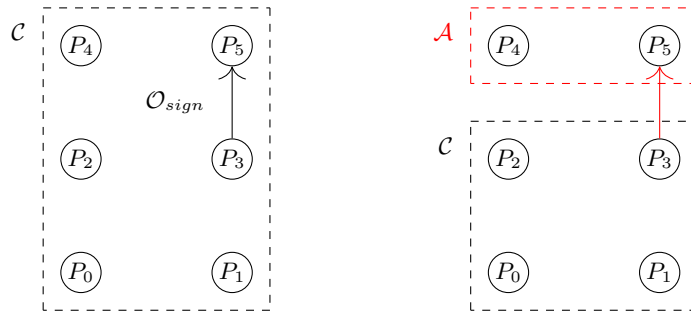


Fig. 1. Left: a 6-party setting where the adversary requests a signature using an oracle, Right: a 4-party setting where the adversary simulates another 2 parties but is now unable to obtain the same signature as on the left.

3.1 Oracles

Many different interpretations exist in the literature of what oracles the adversary should be given access to. The key choices here are whether (1) a simulation

oracle should be provided, (2) a verification oracle should be provided, and (3) whether the adversary should still have access to the oracles after the challenge has been issued. Whereas the precise attacker model might depend on the context and our framework allows us to capture this, we here choose to focus on the strongest level of security, by providing the adversary with as much as possible without trivially breaking the challenge.

Definition 3.1. For any n , let the standard n -sender SendPriv -oracles denote:

- $\mathcal{O}_{\text{sign}}^{(1)} = \mathcal{O}_{\text{sign}}^{(2)}$: Upon input (m_i, s, v) returns $\sigma_i := \text{Sign}_{P_s \rightarrow P_v}(m_i)$ if $s, v \in [n]$ and \perp otherwise.
- $\mathcal{O}_{\text{sim}}^{(1)} = \mathcal{O}_{\text{sim}}^{(2)}$: Upon input (m_i, s, v) returns $\sigma_i := \text{Simulate}_{P_s \rightarrow P_v}(m_i)$ if $s, v \in [n]$ and \perp otherwise.
- $\mathcal{O}_{\text{veri}}^{(1)}$: Upon input (m_i, σ_i, s, v) returns $\text{Verify}_{P_s \rightarrow P_v}(m_i, \sigma_i)$ if $s, v \in [n]$ and \perp otherwise.
- $\mathcal{O}_{\text{veri}}^{(2)}$: Upon input (m_i, σ_i, s, v) returns $\text{Verify}_{P_s \rightarrow P_v}(m_i, \sigma_i)$ if $s, v \in [n]$ and $\sigma_i \neq \sigma^*$, and \perp otherwise.

Note that the oracles make use of an implicit ordering of the parties. This makes no difference in any real-world application, but for constructing proofs we also define a set of oracles that allows this ordering to be hidden by a permutation.

Definition 3.2. For any set of oracles for $\mathcal{G}^{\text{SendPriv}}$ and any permutation π define the permuted oracles as follows, where $b \in \{0, 1\}$:

- $\mathcal{O}_{\text{sign}}^{(\pi, b)}$: On input (m_i, s, v) output $\mathcal{O}_{\text{sign}}^{(b)}(m_i, \pi(s), \pi(v))$
- $\mathcal{O}_{\text{sim}}^{(\pi, b)}$: On input (m_i, s, v) output $\mathcal{O}_{\text{sim}}^{(b)}(m_i, \pi(s), \pi(v))$
- $\mathcal{O}_{\text{veri}}^{(\pi, b)}$: On input (m_i, σ_i, s, v) output $\mathcal{O}_{\text{veri}}^{(b)}(m_i, \sigma_i, \pi(s), \pi(v))$

3.2 Definition

Taking all these things into consideration, we can now craft a definition of sender privacy. This definition is more in line with current research in ID-based-SDVS research such as [Hua+11].

Definition 3.3. A DVS scheme Π is n -party sender private with respect to \mathcal{O} if for any adversary \mathcal{A} ,

$$\text{Adv}_{\Pi, \mathcal{A}, \mathcal{O}}^{\text{SendPriv}}(\kappa, n) = \Pr_{c \leftarrow \{0, 1\}} [\mathcal{G}_{\Pi, \mathcal{A}}^{\text{SendPriv}}(\kappa, n, c) = c] - \frac{1}{2} \leq \text{negl}(\kappa).$$

A DVS scheme is n -party sender private if it is n -party sender private with respect to the standard n -sender SendPriv -oracles.

4 Alternative definitions

In this section, we look at possible alternative definitions that one could consider equally valid generalizations of the 2-party setting to the n -party setting. For example, in the 2-party setting, we pick the challenge uniformly at random between the two possible senders, thus one could consider picking uniformly at random from n senders in the n -party setting.

Definition 4.1. *A DVS scheme is n -party random-challenge sender private with respect to \mathcal{O} if for any adversary \mathcal{A} ,*

$$\text{Adv}_{\Pi, \mathcal{A}, \mathcal{O}}^{\text{nrSendPriv}}(\kappa, n) = \Pr_{c \leftarrow [n-1]} [\text{G}_{\Pi, \mathcal{A}, \mathcal{O}}^{\text{SendPriv}}(\kappa, n, c) = c] - \frac{1}{n} \leq \text{negl}(\kappa).$$

*A DVS scheme is n -party random-challenge sender private if it is n -party random-challenge sender private with respect to the standard n -sender **SendPriv**-oracles.*

Furthermore, one could strengthen the definition even more by allowing the adversary to choose which two senders the challenge is chosen from and which party is the verifier.

Game 3: $\text{G}_{\Pi, \mathcal{A}, \mathcal{O}}^{\text{ChosenSendPriv}}(\kappa, n, c)$

- 1 $\text{params} \leftarrow \text{Setup}$
 - 2 $(\text{pk}_{P_0}, \text{sk}_{P_0}) \leftarrow \text{KeyGen}; \dots; (\text{pk}_{P_n}, \text{sk}_{P_n}) \leftarrow \text{KeyGen}$
 - 3 $(m^*, s_0, s_1, r, \text{state}) \leftarrow \mathcal{A}_{\text{sign}, \mathcal{O}_{\text{veri}}, \mathcal{O}_{\text{sim}}}^{(1)}(1, \text{params}, \text{pk}_{P_0}, \dots, \text{pk}_{P_n})$
 - 4 $\sigma^* = \text{Sign}_{P_{s_c} \rightarrow P_r}(m^*)$
 - 5 $c' \leftarrow \mathcal{A}_{\text{sign}, \mathcal{O}_{\text{veri}}, \mathcal{O}_{\text{sim}}}^{(2)}(2, \text{state}, \sigma^*)$
 - 6 Output c'
-

Definition 4.2. *A DVS scheme is n -party adversarial-challenge sender private with respect to \mathcal{O} if for any adversary \mathcal{A} ,*

$$\text{Adv}_{\Pi, \mathcal{A}, \mathcal{O}}^{\text{ChosenSendPriv}}(\kappa, n) = \Pr_{c \leftarrow \{0,1\}} [\text{G}_{\Pi, \mathcal{A}}^{\text{ChosenSendPriv}}(\kappa, n, c) = c] - \frac{1}{2} \leq \text{negl}(\kappa).$$

*A DVS scheme is n -party adversarial-challenge sender private if it is n -party adversarial-challenge sender private with respect to the standard n -sender **SendPriv**-oracles.*

4.1 Relations

As one might expect, the above-defined alternative definitions relate strongly to the main definition, Definition 3.3. In fact, in this section, we show that they are equivalent up to polynomial differences in the advantages.

For the universally random challenge, this can be done by simply only considering the cases where the challenge is P_0 or P_1 , which will be the case 2 out of n times, giving us a loss in the advantage of a factor $\frac{2}{n}$.

Theorem 4.3. *For any adversary \mathcal{A} , DVS scheme Π , and set of oracles \mathcal{O} ,*

$$\frac{2}{n} \cdot \text{Adv}_{\Pi, \mathcal{A}, \mathcal{O}}^{\text{SendPriv}}(\kappa, n) \leq \text{Adv}_{\Pi, \mathcal{A}, \mathcal{O}}^{nr\text{SendPriv}}(\kappa, n)$$

Proof.

$$\begin{aligned} & \text{Adv}_{\Pi, \mathcal{A}}^{nr\text{SendPriv}}(\kappa, n) \\ &= \Pr_{c \leftarrow [n-1]} [\text{G}_{\Pi, \mathcal{A}}^{\text{SendPriv}}(\kappa, n, c) = c] - \frac{1}{n} \\ &= \frac{2}{n} \Pr_{c \leftarrow [1]} [\text{G}_{\Pi, \mathcal{A}}^{\text{SendPriv}}(\kappa, n, c) = c] + \frac{n-2}{n} \Pr_{c \leftarrow [2, n-1]} [\text{G}_{\Pi, \mathcal{A}}^{\text{SendPriv}}(\kappa, n, c) = c] - \frac{1}{n} \\ &= \frac{2}{n} \left(\Pr_{c \leftarrow [1]} [\text{G}_{\Pi, \mathcal{A}}^{\text{SendPriv}}(\kappa, n, c) = c] - \frac{1}{2} \right) + \frac{n-2}{n} \Pr_{c \leftarrow [2, n-1]} [\text{G}_{\Pi, \mathcal{A}}^{\text{SendPriv}}(\kappa, n, c) = c] \\ &= \frac{2}{n} \cdot \text{Adv}_{\Pi, \mathcal{A}}^{\text{SendPriv}}(\kappa) + \frac{n-2}{n} \Pr_{c \leftarrow \{2, \dots, n-1\}} [\text{G}_{\Pi, \mathcal{A}}^{\text{SendPriv}}(\kappa, n, c) = c] \\ &\geq \frac{2}{n} \cdot \text{Adv}_{\Pi, \mathcal{A}, \mathcal{O}}^{\text{SendPriv}}(\kappa, n), \end{aligned}$$

where $[2, n-1] = \{2, \dots, n-1\}$. \square

Theorem 4.4. *For any adversary \mathcal{A} , set of oracles \mathcal{O} and DVS scheme Π , there exists an adversary \mathcal{B} such that*

$$\frac{1}{2} \text{Adv}_{\Pi, \mathcal{A}, \mathcal{O}}^{nr\text{SendPriv}}(\kappa, n) \leq \text{Adv}_{\Pi, \mathcal{B}, \mathcal{O}}^{\text{SendPriv}}(\kappa, n).$$

Proof. Here, we omit the subscripts Π and \mathcal{O} for Adv and G for simplicity. Let \mathcal{B} be defined as in Games 4 and 5. The permutation is used here to hide the index-

Game 4: $\mathcal{B}^{\mathcal{O}_{\text{sign}}^{(1)}, \mathcal{O}_{\text{veri}}^{(1)}, \mathcal{O}_{\text{sim}}^{(1)}}(1, \text{params}, \text{pk}_{P_0}, \dots, \text{pk}_{P_n})$

- 1 Pick a random permutation $\pi : [n] \mapsto [n]$ such that $\pi(n) = n$
 - 2 $(m^*, \text{state}) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{sign}}^{(\pi, 1)}, \mathcal{O}_{\text{veri}}^{(\pi, 1)}, \mathcal{O}_{\text{sim}}^{(\pi, 1)}}(1, \text{params}, \text{pk}_{P_{\pi(0)}}, \dots, \text{pk}_{P_{\pi(n)}})$
 - 3 Output $(m^*, (\pi, \text{state}))$
-

tion of the parties from the adversary. Note that applying a permutation π in this fashion is equivalent to generating the keypairs in the order $\pi^{-1}(0) \dots \pi^{-1}(n)$ and since these are i.i.d. samples the order of their generation does not affect the winning probability of \mathcal{A} . However, it guarantees that the winning probability

Game 5: $\mathcal{B}^{\mathcal{O}_{sign}^{(2)}, \mathcal{O}_{veri}^{(2)}, \mathcal{O}_{sim}^{(2)}}(2, \text{state}', \sigma^*)$

```

1 Parse  $\text{state}'$  as  $(\pi, \text{state})$ 
2  $c' \leftarrow \mathcal{A}^{\mathcal{O}_{sign}^{(2)}, \mathcal{O}_{veri}^{(2)}, \mathcal{O}_{sim}^{(2)}}(2, \text{state}, \sigma^*)$ 
3 if  $\pi(c') \in \{0, 1\}$  then
4   Output  $\pi(c')$ 
5 else
6   Output 0

```

of \mathcal{A} is the same for every c . Note that here we use \Pr_π to indicate the uniform probability over all $\pi : [n] \mapsto [n]$ such that $\pi(n) = n$.

$$\begin{aligned}
& \text{Adv}_{\mathcal{B}}^{\text{SendPriv}}(\kappa, n) \\
&= \Pr_{c \leftarrow [1]} [\mathcal{G}_{\mathcal{B}}^{\text{SendPriv}}(\kappa, n, c) = c] - \frac{1}{2} \\
&= \Pr_{c \leftarrow [1], \pi} [\mathcal{G}_{\mathcal{A}}^{\text{SendPriv}}(\kappa, n, \pi^{-1}(c)) = \pi^{-1}(c)] \\
&\quad + \frac{1}{2} \Pr_{\pi} [\mathcal{G}_{\mathcal{A}}^{\text{SendPriv}}(\kappa, n, \pi^{-1}(0)) \notin \{\pi^{-1}(0), \pi^{-1}(1)\}] - \frac{1}{2} \\
&= \Pr_{c \leftarrow [n-1]} [\mathcal{G}_{\mathcal{A}}^{\text{SendPriv}}(\kappa, n, c) = c] - \frac{1}{2} \Pr_{\pi} [\mathcal{G}_{\mathcal{A}}^{\text{SendPriv}}(\kappa, n, \pi^{-1}(0)) \in \{\pi^{-1}(0), \pi^{-1}(1)\}] \\
&= \frac{1}{2} \Pr_{c \leftarrow [n-1]} [\mathcal{G}_{\mathcal{A}}^{\text{SendPriv}}(\kappa, n, c) = c] - \frac{1}{2} \Pr_{\pi} [\mathcal{G}_{\mathcal{A}}^{\text{SendPriv}}(\kappa, n, \pi^{-1}(0)) = \pi^{-1}(1)] \\
&= \frac{1}{2} \left(\Pr_{c \leftarrow [n-1]} [\mathcal{G}_{\mathcal{A}}^{\text{SendPriv}}(\kappa, n, c) = c] - \frac{1}{n-1} \Pr_{c \leftarrow [n-1]} [\mathcal{G}_{\mathcal{A}}^{\text{SendPriv}}(\kappa, n, c) \neq c] \right) \\
&= \frac{1}{2} \left(\frac{n}{n-1} \Pr_{c \leftarrow [n-1]} [\mathcal{G}_{\mathcal{A}}^{\text{SendPriv}}(\kappa, n, c) = c] - \frac{1}{n-1} \right) \\
&= \frac{n}{2(n-1)} \text{Adv}_{\mathcal{A}}^{nr\text{SendPriv}}(\kappa, n) \geq \frac{1}{2} \text{Adv}_{\mathcal{A}}^{nr\text{SendPriv}}(\kappa, n)
\end{aligned}$$

□

Combining Theorem 4.3 and Theorem 4.4, we see that the advantages for fixed-challenge and random-challenge only differ by at most a linear factor. Thus these definitions are equivalent when considering negligible advantages.

Corollary 4.5. *For any $n \in \mathbb{N}$, an SDVS scheme is n -party random-challenge sender private if and only if it is n -party sender private.*

For adversarially-chosen challenges, we could try to simply consider only the cases where the adversary chooses P_0 and P_1 as the challenge senders and P_n as the challenge verifier. However, an adversary could be crafted to never choose this exact combination of parties. Thus, we hide the indexation of the parties under a random permutation. This is done only for the proof and has no impact on the

actual definition, as all parties' keypairs are i.i.d. samples. Since the adversary does not know this permutation, the chance of them picking these parties is in the order of n^{-3} and thus a loss of this order is incurred in the advantage.

Theorem 4.6. *For any adversary \mathcal{A} and set of oracles \mathcal{O} , there exists an adversary \mathcal{B} such that*

$$\frac{2}{n^3 - n} \cdot \text{Adv}_{\Pi, \mathcal{A}, \mathcal{O}}^{\text{ChosenSendPriv}}(\kappa, n) \leq \text{Adv}_{\Pi, \mathcal{B}, \mathcal{O}}^{\text{SendPriv}}(\kappa, n)$$

Proof. Fix \mathcal{A} . Let \mathcal{B} be defined as:

Game 6: $\mathcal{B}^{\mathcal{O}_{sign}^{(1)}, \mathcal{O}_{veri}^{(1)}, \mathcal{O}_{sim}^{(1)}}(1, \text{params}, \text{pk}_{P_0}, \dots, \text{pk}_{P_n})$

- 1 Pick a random permutation $\pi : [n] \mapsto [n]$
 - 2 $(m^*, s_0, s_1, r, \text{state}) \leftarrow \mathcal{A}^{\mathcal{O}_{sign}^{(\pi, 1)}, \mathcal{O}_{veri}^{(\pi, 1)}, \mathcal{O}_{sim}^{(\pi, 1)}}(1, \text{params}, \text{pk}_{P_{\pi(0)}}, \dots, \text{pk}_{P_{\pi(n)}})$
 - 3 **if** $\pi(s_0) = 0 \wedge \pi(s_1) = 1 \wedge \pi(r) = n$ **then**
 - 4 Output $(m^*, (0, \text{state}))$
 - 5 **else if** $\pi(s_0) = 1 \wedge \pi(s_1) = 0 \wedge \pi(r) = n$ **then**
 - 6 Output $(m^*, (1, \text{state}))$
 - 7 **else**
 - 8 Output $(m^*, (2, \text{state}))$
-

Game 7: $\mathcal{B}^{\mathcal{O}_{sign}^{(2)}, \mathcal{O}_{veri}^{(2)}, \mathcal{O}_{sim}^{(2)}}(2, \text{state}', \sigma^*)$

- 1 Parse state' as (b, state)
 - 2 $c' \leftarrow \mathcal{A}^{\mathcal{O}_{sign}^{(2)}, \mathcal{O}_{veri}^{(2)}, \mathcal{O}_{sim}^{(2)}}(2, \text{state}, \sigma^*)$
 - 3 **if** $b = 0$ **then**
 - 4 Output c'
 - 5 **else if** $b = 1$ **then**
 - 6 Output $1 - c'$
 - 7 **else**
 - 8 $c'' \leftarrow \{0, 1\}$
 - 9 Output c''
-

The permutation is used here to hide the indexation of the parties from the adversary. Note that applying a permutation π in this fashion is equivalent to generating the keypairs in the order $\pi^{-1}(0) \dots \pi^{-1}(n)$ and since these are i.i.d. samples the order of their generation does not affect the winning probability of \mathcal{A} . When playing game $G_{\Pi, \mathcal{B}}^{\text{SendPriv}}$, we can now distinguish two cases:

1. $\{\pi(s_0), \pi(s_1)\} = \{0, 1\}$ and $\pi(r) = n$. Since π is random and unknown to \mathcal{A} , this happens with probability $\frac{2(n-2)!}{(n+1)!}$. In this case, \mathcal{A} has chosen P_0 and P_1 as the possible signers and P_n as the verifier, making $G_{\Pi, \mathcal{A}}^{\text{ChosenSendPriv}}$ and $G_{\Pi, \mathcal{B}}^{\text{SendPriv}}$ equivalent.
2. Otherwise, \mathcal{A} has chosen different signers or verifiers, in which case $G_{\Pi, \mathcal{B}}^{\text{SendPriv}}$ becomes equivalent to a random coin flip, with probability $\frac{1}{2}$ of guessing c .

Combining this, we get that

$$\Pr_{c \leftarrow \{0,1\}} [G_{\Pi, \mathcal{B}, \mathcal{O}}^{\text{SendPriv}}(\kappa, n, c) = c] = \frac{2(n-2)!}{(n+1)!} \Pr_{c \leftarrow \{0,1\}} [G_{\Pi, \mathcal{A}, \mathcal{O}}^{\text{ChosenSendPriv}}(\kappa, n, c) = c] + \left(1 - \frac{2(n-2)!}{(n+1)!}\right) \frac{1}{2}.$$

Thus,

$$\text{Adv}_{\Pi, \mathcal{B}, \mathcal{O}}^{\text{SendPriv}}(\kappa, n) = \frac{2(n-2)!}{(n+1)!} \cdot \text{Adv}_{\Pi, \mathcal{A}, \mathcal{O}}^{\text{ChosenSendPriv}}(\kappa, n).$$

□

Corollary 4.7. *For any $n \in \mathbb{N}$, an SDVS scheme is n -party adversarial-challenge sender private if and only if it is n -party sender private.*

Proof. Theorem 4.6 shows that if a scheme is sender private then it is also adversarial-challenge sender private since the advantage differs by a factor $\mathcal{O}(n^3)$. The other direction is trivial, as any adversary for sender-privacy can trivially be transformed into an adversary for adversarial-challenge sender privacy, always outputting $s_0 = 0, s_1 = 1, r = n$, which gives both adversaries the exact same winning probability. □

5 Alternative oracles

In this section we show that one can use other properties of SDVS schemes, e.g. non-transferability and unforgeability, to provide equally strong sender-privacy while giving the adversary weaker oracles. This allows us to more easily prove that existing schemes satisfy our definition. Note that in this section we only consider the cases where the security advantages are negligible. First, we will focus on the verification oracle, showing that they can be removed without impacting the quality of the security when the scheme is unforgeable. Then, we show that the number of parties can be limited to 3 ($n = 2$) when a scheme is both unforgeable and non-transferable.

Definition 5.1. *A DVS scheme $\Pi = (\text{KeyGen}, \text{Sign}, \text{Verify}, \text{Simulate})$ is n -party strongly-unforgeable with respect to \mathcal{O} if for any adversary \mathcal{A} ,*

$$\text{Adv}_{\Pi, \mathcal{A}, \mathcal{O}}^{\text{UF}}(\kappa, n) = \Pr [G_{\Pi, \mathcal{A}, \mathcal{O}}^{\text{UF}}(\kappa, n) = \top] \leq \text{negl}(\kappa),$$

Game 8: $G_{\Pi, \mathcal{A}, \mathcal{O}}^{\text{UF}}(\kappa, n)$

```

1 params  $\leftarrow$  Setup
2  $(\text{pk}_{P_0}, \text{sk}_{P_0}) \leftarrow \text{KeyGen}; \dots; (\text{pk}_{P_n}, \text{sk}_{P_n}) \leftarrow \text{KeyGen}$ 
3  $(m^*, \sigma^*, s, v) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{sign}}, \mathcal{O}_{\text{veri}}, \mathcal{O}_{\text{sim}}}(\text{params}, \text{pk}_{P_0}, \dots, \text{pk}_{P_n})$ 
4 if  $\text{Verify}_{P_s \rightarrow P_v}(m^*, \sigma^*) = 1$  and  $\forall i : \sigma^* \neq \sigma_i$  then
5    $\perp$  Output  $\top$ .
6 else
7    $\perp$  Output  $\perp$ .

```

where the game $G_{\Pi, \mathcal{A}, \mathcal{O}}^{\text{UF}}$ is defined in Game 8. A DVS scheme is n -party strongly-unforgeable if it is n -party strongly-unforgeable with respect to $\mathcal{O}_{\text{sign}}^{(1)}, \mathcal{O}_{\text{sim}}^{(1)}, \mathcal{O}_{\text{veri}}^{(1)}$ from the standard n -sender SendPriv -oracles.

Theorem 5.2. Let $n \in \mathbb{N}$ and $\mathcal{O} = \{\mathcal{O}_{\text{sign}}^{(1)}, \mathcal{O}_{\text{sign}}^{(2)}, \mathcal{O}_{\text{sim}}^{(1)}, \mathcal{O}_{\text{sim}}^{(2)}, \mathcal{O}_{\text{veri}}^{(1)}, \mathcal{O}_{\text{veri}}^{(2)}\}$ be the n -sender standard oracles. Any DVS scheme that is n -party sender private with respect to $\mathcal{O}' = \{\mathcal{O}_{\text{sign}}^{(1)}, \mathcal{O}_{\text{sign}}^{(2)}, \mathcal{O}_{\text{sim}}^{(1)}, \mathcal{O}_{\text{sim}}^{(2)}, \mathcal{O}_{\text{veri}}^{(1)} = \emptyset, \mathcal{O}_{\text{veri}}^{(2)} = \emptyset\}$ and strongly unforgeable is n -party sender private (with respect to \mathcal{O}).

Proof. Fix $n \in \mathbb{N}$. Suppose DVS scheme Π is n -party sender private with respect to $\mathcal{O}' = \{\mathcal{O}_{\text{sign}}^{(1)}, \mathcal{O}_{\text{sign}}^{(2)}, \mathcal{O}_{\text{sim}}^{(1)}, \mathcal{O}_{\text{sim}}^{(2)}, \mathcal{O}_{\text{veri}}^{(1)} = \emptyset, \mathcal{O}_{\text{veri}}^{(2)} = \emptyset\}$ and strongly unforgeable, but not n -party sender private with respect to \mathcal{O} . Then there exists an adversary \mathcal{A} such that $\text{Adv}_{\Pi, \mathcal{A}, \mathcal{O}}^{\text{SendPriv}}(\kappa) \not\leq \text{negl}(\kappa)$. Let \mathcal{A}' be \mathcal{A} , except every query $\mathcal{O}_{\text{veri}}^{(b)}(m_i, \sigma_i, s, v)$ is replaced with \top if (m_i, σ_i) was the result of a signing or simulating oracle query and \perp otherwise. Since \mathcal{A}' no longer uses the verification oracles, we have $\text{Adv}_{\Pi, \mathcal{A}', \mathcal{O}}^{\text{SendPriv}} = \text{Adv}_{\Pi, \mathcal{A}', \mathcal{O}'}^{\text{SendPriv}} \leq \text{negl}(\kappa)$, i.e. \mathcal{A}' has the same advantage with respect to \mathcal{O} and \mathcal{O}' , as they only differ in the verification oracles.

Now consider the adversary \mathcal{B} , who intends to create a forged signature. \mathcal{B} runs \mathcal{A} , recording all signing and simulating queries. Whenever \mathcal{A} makes a verification query for a valid signature that was not the result of a signing or simulating query, \mathcal{B} outputs this signature and halts. Note that the only difference in the behavior of \mathcal{A} and \mathcal{A}' can occur when \mathcal{A} makes such a query. Since the difference between $\text{Adv}_{\Pi, \mathcal{A}', \mathcal{O}}^{\text{SendPriv}}$ and $\text{Adv}_{\Pi, \mathcal{A}, \mathcal{O}}^{\text{SendPriv}}$ is more than negligible, we have that such a query occurs with more than negligible probability, giving \mathcal{B} a more than negligible probability of constructing a forgery. This contradicts the fact that Π is strongly unforgeable. \square

Theorem 5.3. Any DVS scheme Π that is 2-party sender private, strongly unforgeable, and computationally non-transferable is n -party sender private for any $n \geq 2$.

Proof. Suppose a DVS scheme Π is 2-party sender private, strongly unforgeable, and computationally non-transferable. Assume towards a contradiction that Π

is not n -party sender private for some fixed $n > 2$. By Theorem 5.2, this means Π is also not n -party sender private with respect to

$$\mathcal{O}' = \{\mathcal{O}_{sign}^{(1)}, \mathcal{O}_{sign}^{(2)}, \mathcal{O}_{sim}^{(1)}, \mathcal{O}_{sim}^{(2)}, \mathcal{O}_{veri}^{(1)} = \emptyset, \mathcal{O}_{veri}^{(2)} = \emptyset\}.$$

Thus, there exists an adversary \mathcal{A} such that $\text{Adv}_{\Pi, \mathcal{A}, \mathcal{O}'}^{\text{SendPriv}}(\kappa, n) \not\leq \text{negl}(\kappa)$. Let $\mathcal{A}'(1, \text{params}, \text{pk}_{P_0}, \text{pk}_{P_1}, \text{pk}_{P_2})$ be as follows: First, sample $n-2$ keypairs $(\text{sk}'_{P_2}, \text{pk}'_{P_2}) \dots (\text{sk}'_{P_{n-1}}, \text{pk}'_{P_{n-1}})$ representing parties $P'_2 \dots P'_{n-1}$ and set $P'_0 = P_0$, $P'_1 = P_1$, $P'_n = P_2$. Then, run \mathcal{A} with the oracles \mathcal{O}'' defined as follows, with $b = 1, 2$:

- $\mathcal{O}_{veri}^{(b)} = \emptyset$.
- $\mathcal{O}_{sign}^{(b)}(m_i, s, v) :$
 - If $s, v \in \{0, 1, n\}$, return $\mathcal{O}_{sign}^{(b)}(m_i, \max(2, s), \max(2, v))$.
 - If $s \in \{2, \dots, n-1\}$ and $v \in [n]$, return $\text{Sign}_{P'_s \rightarrow P'_v}(m_i)$.
 - If $s \in \{0, 1, n\}$ and $v \in \{2, \dots, n-1\}$, return $\text{Simulate}_{P'_s \rightarrow P'_v}(m_i)$.
 - Else, return \perp .
- $\mathcal{O}_{sim}^{(b)}(m_i, s, v) :$
 - If $s, v \in \{0, 1, n\}$, return $\mathcal{O}_{sim}^{(b)}(m_i, \max(2, s), \max(2, v))$.
 - If $v \in \{2, \dots, n-1\}$ and $s \in [n]$, return $\text{Simulate}_{P'_s \rightarrow P'_v}(m_i)$.
 - If $v \in \{0, 1, n\}$ and $s \in \{2, \dots, n-1\}$, return $\text{Sign}_{P'_s \rightarrow P'_v}(m_i)$.
 - Else, return \perp .

Note that these oracles make use of the fact that one can simulate or sign a signature without, respectively, the sender's or verifier's secret key. Thus we circumvent the issue mentioned in Section 3. In the oracles, \max is used here to map n to 2, as n and 2 are the challenge verifiers in the n - and 2-party respectively.

Since Π is 2-party sender private, we have $\text{Adv}_{\Pi, \mathcal{A}', \mathcal{O}''}^{\text{SendPriv}}(\kappa, 2) \leq \text{negl}(\kappa)$. When we replace all oracle calls by their respective functionality, then $\text{G}_{\Pi, \mathcal{A}, \mathcal{O}'}^{\text{SendPriv}}(\kappa, n, c)$ and $\text{G}_{\Pi, \mathcal{A}', \mathcal{O}''}^{\text{SendPriv}}(\kappa, 2, c)$ differ, up to relabeling of the parties, only in one way : some Sign executions in $\text{G}_{\Pi, \mathcal{A}, \mathcal{O}'}^{\text{SendPriv}}(\kappa, n, c)$ have been replaced by Simulate in $\text{G}_{\Pi, \mathcal{A}', \mathcal{O}''}^{\text{SendPriv}}(\kappa, 2, c)$ and vice versa. Suppose $i \in \mathbb{N}$ such replacements have been made, then for $0 \leq j \leq i$ let $\text{G}_j(\kappa, c)$ be $\text{G}_{\Pi, \mathcal{A}, \mathcal{O}'}^{\text{SendPriv}}(\kappa, n, c)$ with only the first j such replacements made, i.e. $\text{G}_0(\kappa, c) = \text{G}_{\Pi, \mathcal{A}, \mathcal{O}'}^{\text{SendPriv}}(\kappa, n, c)$ and $\text{G}_i(\kappa, c) = \text{G}_{\Pi, \mathcal{A}', \mathcal{O}''}^{\text{SendPriv}}(\kappa, 2, c)$. Since, by construction, $\Pr[\text{G}_0(\kappa, c) = c] - \frac{1}{2} \not\leq \text{negl}(\kappa)$ and $\Pr[\text{G}_i(\kappa, c) = c] - \frac{1}{2} \leq \text{negl}(\kappa)$, we can fix a lowest k such that $\Pr[\text{G}_k(\kappa, c) = c] - \frac{1}{2} \not\leq \text{negl}(\kappa)$ and $\Pr[\text{G}_{k+1}(\kappa, c) = c] - \frac{1}{2} \leq \text{negl}(\kappa)$. G_k and G_{k+1} differ only in one replacement. Without loss of generality, assume one $\text{Sign}_{P'_s \rightarrow P'_v}(m)$ was replaced by $\text{Simulate}_{P'_s \rightarrow P'_v}(m)$.

Now define an adversary \mathcal{B} for G^{NT} as follows: $\mathcal{B}(1, \text{params}, \text{pk}_S, \text{sk}_S, \text{pk}_V, \text{sk}_V)$ picks a $c \in \{0, 1\}$ and runs $\text{G}_k(c, \kappa)$, replacing pk_s with pk_S , sk_s with sk_S , pk_v with pk_V , and sk_v with sk_V . This replacement is only a relabeling. The execution of G_k is stopped at the one difference with G_{k+1} , then outputs $(m, (\text{state}, c))$, where state is the current state of G_k and m the message in the replaced Sign .

$\mathcal{B}(2, (\text{state}, c), \sigma)$ then continues the execution of G_k with σ as the result of the replaced **Sign** until G_k outputs c' . \mathcal{B} then outputs 0 if $c = c'$ and 1 otherwise.

Note that in $G_{\Pi, \mathcal{B}}^{\text{NT}}(0, \kappa)$, i.e. the case where a **Sign** is used in the non-transferability game, \mathcal{B} plays $G_k(c, \kappa)$ and in $G_{\Pi, \mathcal{B}}^{\text{NT}}(1, \kappa)$, \mathcal{B} plays $G_{k+1}(c, \kappa)$. Thus we have that

$$\Pr_b [G_{\Pi, \mathcal{B}}^{\text{NT}}(b, \kappa) = b] = \frac{1}{2} \Pr_c [G_k(c, \kappa) = c] + \frac{1}{2} \Pr_c [G_{k+1}(c, \kappa) \neq c].$$

This directly implies that

$$\text{Adv}_{\Pi, \mathcal{B}}^{\text{NT}}(\kappa, n) = \frac{1}{2} \left(\Pr_c [G_k(c, \kappa) = c] - \Pr_c [G_{k+1}(c, \kappa) = c] \right) \not\leq \text{negl}(\kappa).$$

This contradicts our assumption that Π is computationally non-transferable, thus Π must be n -party sender private. \square

6 Conclusion

In this paper, we provided a way of defining sender privacy in the n -party setting that is novel for DVS schemes, a generalization of existing definitions and in line with definitions for other types of schemes in the multi-party setting, in particular ID-based SDVS schemes. We explored the effects of choosing the challenge differently and observed that this induces only polynomial differences in the advantage the adversary has. Furthermore, we showed how other properties of a SDVS scheme can be used to boost the sender privacy of a scheme from an alternative definition to our definition. In particular, we have proven that under the assumption of strong unforgeability and computational non-transferability a 2-party sender-private scheme is n -party sender private. The proven relations are important since the SDVS schemes are often meant to be employed in an n -party setting and we give sufficient conditions for this to be secure.

We would like to stress that the objective of this paper is to formulate sender privacy in such a way that it covers all theoretical types of attacks that should be intuitively covered by this property. Thus, the definition presented is not necessarily technically different from previous definitions, in fact, it will coincide in many cases. As such we do not provide separating examples of schemes that satisfy one definition but not another, as any such case would be extremely artificial. Instead, the definitions in this work and their equivalence should be used to simplify proofs where sender privacy property is used, both in classical and quantum use cases.

7 Acknowledgements

JvW is supported by the Luxembourg National Research Fund (FNR), under the joint CORE project Q-CoDe (CORE17/IS/11689058/Q-CoDe/Ryan).

References

- [Bel+01] Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. “Key-privacy in public-key encryption”. In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2001, pp. 566–582.
- [Cha96] David Chaum. *Private signature and proof systems*. US Patent 5,493,614. Feb. 1996.
- [CV89] David Chaum and Hans Van Antwerpen. “Undeniable signatures”. In: *Conference on the Theory and Application of Cryptology*. Springer. 1989, pp. 212–216.
- [Hua+06] Xinyi Huang, Willy Susilo, Yi Mu, and Futai Zhang. “Short (identity-based) strong designated verifier signature schemes”. In: *International Conference on Information Security Practice and Experience*. Springer. 2006, pp. 214–225.
- [Hua+11] Qiong Huang, Guomin Yang, Duncan S Wong, and Willy Susilo. “Identity-based strong designated verifier signature revisited”. In: *Journal of Systems and Software* 84.1 (2011), pp. 120–129.
- [JSI96] Markus Jakobsson, Kazue Sako, and Russell Impagliazzo. “Designated verifier proofs and their applications”. In: *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 1996, pp. 143–154.
- [LV04] Fabien Laguillaumie and Damien Vergnaud. “Designated verifier signatures: anonymity and efficient construction from any bilinear map”. In: *International Conference on Security in Communication Networks*. Springer. 2004, pp. 105–119.
- [RST01] Ronald L Rivest, Adi Shamir, and Yael Tauman. “How to leak a secret”. In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2001, pp. 552–565.
- [SKM03] Shahrokh Saeednia, Steve Kremer, and Olivier Markowitch. “An efficient strong designated verifier signature scheme”. In: *International conference on information security and cryptology*. Springer. 2003, pp. 40–54.
- [SZM04] Willy Susilo, Fangguo Zhang, and Yi Mu. “Identity-based strong designated verifier signature schemes”. In: *Australasian Conference on Information Security and Privacy*. Springer. 2004, pp. 313–324.