Short 2-Move Undeniable Signatures

Jean Monnerat* and Serge Vaudenay

EPFL, Switzerland
http://lasecwww.epfl.ch/

Abstract. Attempting to reach a minimal number of moves in cryptographic protocols is a quite classical issue. Besides the theoretical interests, minimizing the number of moves can clearly facilitate practical implementations in environments with communication constraints. In this paper, we offer a solution to this problem in the context of undeniable signatures with interactive verification protocols by proposing a way to achieve these protocols in 2 moves. To this goal, we review a scheme we proposed at Asiacrypt 2004 whose property is the full scalability of the signature length against security. We slightly modify (to make it non-transferable) a 2-move version of this scheme which was mentioned in the original article without a proof of security. In the random oracle model, we prove the security of the modified version against an active adversary and precisely assess the security in terms of the signature length. To the best of our knowledge, this scheme is the first 2-move undeniable signature scheme with a security proof.

Key words: Undeniable signatures, 2-move protocols.

1 Introduction

The concept of undeniable signature was introduced by Chaum and van Antwerpen [6] in 1989. The difference between this kind of signature and a classical one is that the verification of a signature cannot be achieved without the cooperation of the signer (originally, for privacy motivations). Namely, by interacting with a verifier in a so-called confirmation (resp. denial) protocol the signer is able to prove the validity (resp. invalidity) of a given message-signature pair. This property opposes to the universal verifiability of classical digital signatures and allows the signer to have a control on the spread of his signatures. Further applications of undeniable signatures such as licensing software or auctions were proposed in the literature. Since then, lots of contributions and new schemes have been published, among them are [3,5,8,9,13,14,17,18,19].

At Eurocrypt 2005, Kurosawa et al. [13] proposed a variant of the scheme of Chaum [5] with 3-move confirmation and denial protocols in the random oracle model. Although this scheme does not achieve non-transferability, it is the first

* Supported by a grant of the Swiss National Science Foundation, 200020-109133.

one presenting 3-move protocols with a security proof. Until this scheme proposal, all provably secure interactive undeniable signature schemes were composed of zero-knowledge confirmation and denial protocols which required at least 4 moves. Non-interactive variants of undeniable signatures can be obtained as shown in [12,15] using a so-called designated verifier technique by using classical techniques for non-transferability. In this setting, the signature is only intended to one designated recipient. To ensure that this one cannot convince another party of the validity of the signature, it is required that the recipient could have been able (with his secret key) to produce the signature. When this can be done perfectly, we say that the scheme satisfies perfect non-transferability. In this case, such (designated verifier) signatures cannot satisfy the non-repudiation property.

The main contribution of this article is to show how to achieve a scheme with interactive protocols having a minimal number of rounds. To this end, we revisit a 2-move variant of the MOVA undeniable signature we mentioned in [17] (without any security proof). In order to achieve perfect non-transferability, we modify the protocols of the MOVA scheme by adding a trapdoor one-way permutation with a secret key associated to the verifier. This differs from the commonly used techniques of trapdoor commitments which does not seem appropriate for a 2-move protocol. In the random oracle model, we provide some formal security proofs on the different required properties related to the confirmation and denial protocols such as the soundness, zero-knowledge and non-transferability. We redo the invisibility and unforgeability analysis in settings where the attacker has access to signing, confirmation and denial oracles. This provides precise security bounds and explain how to select MOVA parameters.

In the next section, we recall the definition of an undeniable signature. Section 3 is devoted to the security model of an undeniable signature. Then, we present the 4-move and modified 2-move versions of the MOVA scheme [17] in Section 4. We prove security properties of the modified 2-move version in the subsequent section. Finally, Section 6 concludes this paper.

2 Undeniable Signature

We consider two players who are the signer ($\mathbf{S}$) and the verifier ($\mathbf{V}$). Let $k \in \mathbf{N}$ be a security parameter, $\mathcal{M}$ be the message space and Σ be the signature space. An undeniable signature scheme is composed of the four following algorithms.

Setup The setup is composed of two probabilistic polynomial time algorithms $\mathsf{Setup}^{\mathbf{S}}$ and $\mathsf{Setup}^{\mathbf{V}}$ producing the signer's key pair $(\mathcal{K}_{\mathrm{p}}^{\mathbf{S}}, \mathcal{K}_{\mathrm{s}}^{\mathbf{S}}) \leftarrow \mathsf{Setup}^{\mathbf{S}}(1^k)$ and the verifier's key pair $(\mathcal{K}_{\mathrm{p}}^{\mathbf{V}}, \mathcal{K}_{\mathrm{s}}^{\mathbf{V}}) \leftarrow \mathsf{Setup}^{\mathbf{V}}(1^k)$.

Sign Let $m \in \mathcal{M}$ be a message to sign. On the input of the signer's secret key $\mathcal{K}_{\mathrm{s}}^{\mathbf{S}}$, the (probabilistic) polynomial time algorithm Sign generates a signature $\sigma \leftarrow \mathsf{Sign}(m, \mathcal{K}_{\mathrm{s}}^{\mathbf{S}})$ of m (which lies in Σ). We say that (m, σ) is *valid* if there exists a random tape such that $\mathsf{Sign}(m, \mathcal{K}_{\mathrm{s}}^{\mathbf{S}})$ outputs σ. Otherwise, we say that (m, σ) is *invalid*.

Confirm Let $(m, \sigma) \in \mathcal{M} \times \Sigma$ be a supposedly valid message-signature pair. $\mathsf{Confirm}$ is an interactive protocol between $\mathbf{S}$ and $\mathbf{V}$ i.e., a pair of interactive probabilistic polynomial time algorithms $\mathsf{Confirm}_{\mathbf{S}}$ and $\mathsf{Confirm}_{\mathbf{V}}$ such that m, σ, $\mathcal{K}_{\mathrm{p}}^{\mathbf{S}}$, $\mathcal{K}_{\mathrm{p}}^{\mathbf{V}}$ is input of both, $\mathcal{K}_{\mathrm{s}}^{\mathbf{S}}$ is the auxiliary input of $\mathsf{Confirm}_{\mathbf{S}}$, $\mathcal{K}_{\mathrm{s}}^{\mathbf{V}}$ is the auxiliary input of $\mathsf{Confirm}_{\mathbf{V}}$. At the end of the protocol, $\mathsf{Confirm}_{\mathbf{V}}$ outputs a boolean value which tells whether σ is accepted as valid signature of m.

Deny Let $(m, \sigma') \in \mathcal{M} \times \Sigma$ be an alleged invalid message-signature pair. Deny is an interactive protocol between $\mathbf{S}$ and $\mathbf{V}$ i.e., a pair of interactive probabilistic polynomial time algorithms $\mathsf{Deny}_{\mathbf{S}}$ and $\mathsf{Deny}_{\mathbf{V}}$ such that m, σ', $\mathcal{K}_{\mathrm{p}}^{\mathbf{S}}$, $\mathcal{K}_{\mathrm{p}}^{\mathbf{V}}$, is input of both, $\mathcal{K}_{\mathrm{s}}^{\mathbf{S}}$ is the auxiliary input of $\mathsf{Deny}_{\mathbf{S}}$, $\mathcal{K}_{\mathrm{s}}^{\mathbf{V}}$ is the auxiliary input of $\mathsf{Deny}_{\mathbf{V}}$. At the end of the protocol, $\mathsf{Deny}_{\mathbf{V}}$ outputs a boolean value which tells whether σ' is accepted as invalid signature.

An execution of the confirmation (resp. denial) protocol will be denoted by $\mathsf{Confirm}_{\mathbf{S},\mathbf{V}}(\star)$ (resp. $\mathsf{Deny}_{\mathbf{S},\mathbf{V}}(\star)$), where $\star$ is the common input of the players.

3 Security Model

This section is devoted to the different security notions which are required for an undeniable signature to be secure. We consider four basic security notions related to the confirmation and denial protocols which are the *completeness*, the *soundness*, *zero-knowledge*, and the *non-transferability*. The last one ensures that a malicious verifier is not able to convince any third party of the validity of the statement (e.g., a given message signature is valid) proven in the protocol. The non-transferability notion may be important in some applications where the validity of the proof itself is valuable (like for licensing software).

Security notions about the undeniable signature are considered as well. We require *non-repudiation* by resisting adaptive existential forgery attacks. Furthermore, since the motivation of undeniable signature was to avoid the universal verifiability (like for classical signatures), it is important that a scheme satisfies the *invisibility* property. We will consider an active attacker who has access to some oracles and who will have to distinguish a valid message-signature pair from a randomly picked one.

We recall the definition of the statistical distance between two distributions.

Definition 1. *The statistical distance Δ between two random variables X_1 and X_2 with range $\mathcal{X}$ is* $\Delta(X_1, X_1) := \frac{1}{2} \sum_{x \in \mathcal{X}} |\Pr[X_1 = x] - \Pr[X_2 = x]|$.

Completeness. *Given random key pairs* $(\mathcal{K}_{\mathrm{p}}^{\mathbf{S}}, \mathcal{K}_{\mathrm{s}}^{\mathbf{S}}) \leftarrow \mathsf{Setup}^{\mathbf{S}}(1^k)$, $(\mathcal{K}_{\mathrm{p}}^{\mathbf{V}}, \mathcal{K}_{\mathrm{s}}^{\mathbf{V}}) \leftarrow \mathsf{Setup}^{\mathbf{V}}(1^k)$, *for any valid (resp. invalid) message-signature pair* $(m, \sigma) \in \mathcal{M} \times \Sigma$, *the confirmation (resp. denial) protocol* $\mathsf{Confirm}_{\mathbf{S},\mathbf{V}}(m, \sigma, \mathcal{K}_{\mathrm{p}}^{\mathbf{S}}, \mathcal{K}_{\mathrm{p}}^{\mathbf{V}})$ *(resp.* $\mathsf{Deny}_{\mathbf{S},\mathbf{V}}(m, \sigma, \mathcal{K}_{\mathrm{p}}^{\mathbf{S}}, \mathcal{K}_{\mathrm{p}}^{\mathbf{V}})$*) outputs* 1 *with probability* 1 *when* $\mathbf{S}$ *and* $\mathbf{V}$ *correctly follow all steps of the protocol.*

Soundness. *Given random key pairs* $(\mathcal{K}_{\rm p}^{\mathbf{S}}, \mathcal{K}_{\rm s}^{\mathbf{S}}) \leftarrow \mathsf{Setup}^{\mathbf{S}}(1^k)$, $(\mathcal{K}_{\rm p}^{\mathbf{V}}, \mathcal{K}_{\rm s}^{\mathbf{V}}) \leftarrow \mathsf{Setup}^{\mathbf{V}}(1^k)$, *for any invalid (resp. valid) message-signature pair* $(m, \sigma) \in \mathcal{M} \times \Sigma$ *and any cheating signer* $\mathbf{S}^*$ *(modelled as a probabilistic polynomial time interactive algorithm with access to* $\mathcal{K}_{\rm s}^{\mathbf{S}}$*), the probability that the protocol* $\mathsf{Confirm}_{\mathbf{S}^*,\mathbf{V}}(m, \sigma, \mathcal{K}_{\rm p}^{\mathbf{S}}, \mathcal{K}_{\rm p}^{\mathbf{V}})$ *(resp.* $\mathsf{Deny}_{\mathbf{S}^*,\mathbf{V}}(m, \sigma, \mathcal{K}_{\rm p}^{\mathbf{S}}, \mathcal{K}_{\rm p}^{\mathbf{V}})$*) succeeds is negligible with respect to* k.

The success probability of $\mathbf{S}^*$ *is denoted by* $\mathsf{Succ}_{\mathbf{S}^*}^{\mathsf{sd\text{-}con}}$ *(resp.* $\mathsf{Succ}_{\mathbf{S}^*}^{\mathsf{sd\text{-}den}}$*).*

Straight-Line Zero-Knowledge *Let us consider some random key pairs generated as follows*

$$(\mathcal{K}_{\rm p}^{\mathbf{S}}, \mathcal{K}_{\rm s}^{\mathbf{S}}) \leftarrow \mathsf{Setup}^{\mathbf{S}}(1^k), \quad (\mathcal{K}_{\rm p}^{\mathbf{V}}, \mathcal{K}_{\rm s}^{\mathbf{V}}) \leftarrow \mathsf{Setup}^{\mathbf{V}}(1^k).$$

The confirmation (resp. denial) protocol is zero-knowledge if there exists a probabilistic polynomial time oracle machine $\mathcal{B}$ *called simulator such that for any probabilistic polynomial verifier* $\mathbf{V}^*$ *(with or without* $\mathcal{K}_{\rm s}^{\mathbf{V}}$*) and any valid (resp. invalid) pair* $(m, \sigma) \in \mathcal{M} \times \Sigma$, $\mathcal{B}^{\mathbf{V}^*}$ *outputs a transcript which is indistinguishable from the transcript of the protocol* $\mathsf{Confirm}_{\mathbf{S},\mathbf{V}^*}(m, \sigma, \mathcal{K}_{\rm p}^{\mathbf{S}}, \mathcal{K}_{\rm p}^{\mathbf{V}})$ *(resp.* $\mathsf{Deny}_{\mathbf{S},\mathbf{V}^*}(m, \sigma, \mathcal{K}_{\rm p}^{\mathbf{S}}, \mathcal{K}_{\rm p}^{\mathbf{V}})$*), where* $\mathbf{S}$ *is the honest signer. We assume that* $\mathcal{B}$ *and* $\mathbf{V}^*$ *share the same information (e.g.,* $\mathcal{K}_{\rm s}^{\mathbf{V}}$ *if any). Namely, when* $\mathbf{V}^*$ *has access to some random oracles,* $\mathcal{B}$ *can see the queries (and answers) as well. Moreover, we say that the protocol is straight-line zero-knowledge if* $\mathcal{B}$ *does not need to rewind* $\mathbf{V}^*$.

Non-Transferability. *Let us consider some random key pairs generated as follows*

$$(\mathcal{K}_{\rm p}^{\mathbf{S}}, \mathcal{K}_{\rm s}^{\mathbf{S}}) \leftarrow \mathsf{Setup}^{\mathbf{S}}(1^k), \quad (\mathcal{K}_{\rm p}^{\mathbf{V}}, \mathcal{K}_{\rm s}^{\mathbf{V}}) \leftarrow \mathsf{Setup}^{\mathbf{V}}(1^k).$$

The confirmation (resp. denial) protocol is said non-transferable if there exists a probabilistic polynomial time interactive machine $\mathcal{B}$ *with input* $\mathcal{K}_{\rm s}^{\mathbf{V}}$ *such that for any computationally unbounded verifier* $\tilde{\mathbf{V}}$*, any pair* $(m, \sigma) \in \mathcal{M} \times \Sigma$*, the transcript of* $\mathsf{Confirm}_{\mathcal{B},\tilde{\mathbf{V}}}(m, \sigma, \mathcal{K}_{\rm p}^{\mathbf{S}}, \mathcal{K}_{\rm p}^{\mathbf{V}})$ *(resp.* $\mathsf{Deny}_{\mathcal{B},\tilde{\mathbf{V}}}(m, \sigma, \mathcal{K}_{\rm p}^{\mathbf{S}}, \mathcal{K}_{\rm p}^{\mathbf{V}})$*) is indistinguishable from that of the protocol* $\mathsf{Confirm}_{\mathbf{S},\tilde{\mathbf{V}}}(m, \sigma, \mathcal{K}_{\rm p}^{\mathbf{S}}, \mathcal{K}_{\rm p}^{\mathbf{V}})$ *(resp.* $\mathsf{Deny}_{\mathbf{S},\tilde{\mathbf{V}}}(m, \sigma, \mathcal{K}_{\rm p}^{\mathbf{S}}, \mathcal{K}_{\rm p}^{\mathbf{V}})$*). When* $\tilde{\mathbf{V}}$ *has access to some random oracles,* $\mathcal{B}$ *does not see any queries (nor answers) made to them. However,* $\mathcal{B}$ *is assumed to be given a bit telling whether* (m, σ) *is valid or not.*

We consider here the two following notions of indistinguishability.

Perfect Zero-Knowledge (resp. Non-Transferability). Both transcript distributions are identical.

Statistical Zero-Knowledge (resp. Non-Transferability). The statistical distance between the two transcript distributions is negligible.

We note that the definition of non-transferability allows to avoid some attacks in which the verifier $\mathbf{V}^*$ identified with $\mathcal{K}_{\rm p}^{\mathbf{V}}$ forwards messages to the honest signer which were generated by a hidden verifier $\tilde{\mathbf{V}}$. Namely, our definition assures that $\mathbf{V}^*$ with knowledge of $\mathcal{K}_{\rm s}^{\mathbf{V}}$ could simulate the answer of $\mathbf{S}$ (without any help from $\mathbf{S}$) so that $\tilde{\mathbf{V}}$ does not have evidence of the proof validity.

Our definition of non-transferability is similar to that proposed by Camenisch and Michels [4] with the main difference that our version assumes that $\tilde{\mathbf{V}}$ is computationally unbounded. We can thus assume that $\tilde{\mathbf{V}}$ makes no queries to the signing and confirmation/denial oracles. Therefore, the non-transferability of the protocols presented below will also hold with respect to the Camenisch-Michels definition.

We note that the above definition of zero-knowledge is black-box which means that we require the existence of one "universal" simulator having an oracle access to the verifier which is able to produce an indistinguishable transcript for any verifier. More details about the black-box zero-knowledge notion are given in [10].

In the standard model, Barak et al. [1] proved that zero-knowledge proofs of an NP-complete language (possibly non-black-box) requires at least 3 moves. To overcome this limitation, the notion of zero-knowledge was extended in the random oracle model (for more details, see [2]) in which the queries to the random oracles are controlled by the simulator, i.e., it can simulate the output of the oracles provided that the output distribution is correct. Recently, Pass [21] proposed the notion of *deniable zero-knowledge* in the random oracle. The difference with classical zero-knowledge in the random oracle is that the simulator is no longer allowed to simulate the output of the random oracles, but is only able to observe the queries made to the random oracles as well as the corresponding answers. This actually means that the simulator's transcript really corresponds to the view of the verifier. In this model, Pass [21] showed that 2 moves are necessary to achieve zero-knowledge for NP and proposed a general 2-move protocol for NP which is not very convenient for practical purposes. In our results, proofs of zero-knowledge in the random oracle will be deniable as well.

Existential Unforgeability. We consider the standard security notion of existential forgery under an adaptive chosen-message attack as defined by Goldwasser et al. [11] for classical digital signatures. This notion is similar to Kurosawa-Heng [13] and is adapted as follows.

An undeniable signature scheme is secure against an existential forgery under adaptive chosen-message attack if there exists no probabilistic polynomial time algorithm $\mathcal{F}$ which wins the following game with a non-negligible probability.

Game: *$\mathcal{F}$ receives a public key $\mathcal{K}_{\mathrm{p}}^{\mathbf{S}}$ from $(\mathcal{K}_{\mathrm{p}}^{\mathbf{S}}, \mathcal{K}_{\mathrm{s}}^{\mathbf{S}}) \leftarrow \mathsf{Setup}^{\mathbf{S}}(1^k)$ and a verifier's key pair $(\mathcal{K}_{\mathrm{p}}^{\mathbf{V}}, \mathcal{K}_{\mathrm{s}}^{\mathbf{V}}) \leftarrow \mathsf{Setup}^{\mathbf{V}}(1^k)$. Then, $\mathcal{F}$ can query some chosen messages to a signing oracle, some chosen pairs $(m, \sigma) \in \mathcal{M} \times \Sigma$ to a confirmation (and denial) protocol oracle and interact with it in a confirmation (denial) protocol where the oracle plays the role of the signer. All these queries must be polynomially bounded in k and can be sent adaptively. $\mathcal{F}$ wins the game if it outputs a valid pair $(m^*, \sigma^*) \in \mathcal{M} \times \Sigma$ such that m^* was not queried to the signing oracle.*

The success probability of $\mathcal{F}$ in this game is denoted by $\mathsf{Succ}_{\mathcal{F}}^{\mathsf{ef\text{-}cma}}$.

Invisibility. We use a similar definition as Kurosawa-Heng [13]. Consider first a probabilistic polynomial time algorithm $\mathcal{D}$ called *invisibility distinguisher*

and the two following games with respect to a bit b.
$\mathbf{Game}^{\mathsf{inv\text{-}cma}\text{-}b}$. *$\mathcal{D}$ receives $\mathcal{K}_{\mathrm{p}}^{\mathbf{S}}$ from $(\mathcal{K}_{\mathrm{p}}^{\mathbf{S}}, \mathcal{K}_{\mathrm{s}}^{\mathbf{S}}) \leftarrow \mathsf{Setup}^{\mathbf{S}}(1^k)$ and a verifier's key pair $(\mathcal{K}_{\mathrm{p}}^{\mathbf{V}}, \mathcal{K}_{\mathrm{s}}^{\mathbf{V}}) \leftarrow \mathsf{Setup}^{\mathbf{V}}(1^k)$, it can query some chosen messages to a signing oracle and some chosen message-signature pairs $(m, \sigma) \in \mathcal{M} \times \Sigma$ to some oracles running the confirmation and denial protocols. After a given time, $\mathcal{D}$ chooses one message $m^* \in \mathcal{M}$ which was not queried to the signing oracle and submits it to the challenger. If $b = 0$, he sets $\sigma^* = \mathsf{Sign}(m^*, \mathcal{K}_{\mathrm{s}}^{\mathbf{S}})$. Otherwise, σ^* is picked uniformly at random in Σ. $\mathcal{D}$ receives σ^*. After that, the distinguisher can query the signing, confirmation, and denial oracles again provided that m^* is not a query of the signing oracle and (m^*, σ^*) is not a query of the confirmation or denial protocols. Finally, $\mathcal{D}$ outputs a guess bit b'.*

We define the advantage of the distinguisher as follows

$$\mathsf{Adv}_{\mathcal{D}}^{\mathsf{inv\text{-}cma}} := \left| \Pr\left[b' = 1 \text{ in } \mathbf{Game}^{\mathsf{inv\text{-}cma\text{-}1}} \right] - \Pr\left[b' = 1 \text{ in } \mathbf{Game}^{\mathsf{inv\text{-}cma\text{-}0}} \right] \right|,$$

where probabilities are over the random tapes of the involved algorithms. An undeniable signature scheme is said to be invisible under a chosen-message attack *if there exists no probabilistic polynomial time algorithm $\mathcal{D}$ such that the advantage $\mathsf{Adv}_{\mathcal{D}}^{\mathsf{inv\text{-}cma}}$ is non-negligible.*

Note that this definition is similar to that of Galbraith et al. [8] except that the distinguisher is not allowed to query m^* to the signing oracle in our definition. The invisibility notion of Galbraith et al. cannot be satisfied when the signature is deterministic (which is the case for MOVA). This will be discussed in Remark 6.

4 MOVA Scheme

In this section, we present the scheme proposed in [17] as well as the underlying principles. This scheme generalizes the MOVA scheme [18] proposed earlier in 2004 in a very natural way and therefore will be called MOVA as well.

4.1 Preliminaries

We first recall some definitions, useful lemmas, and mathematical problems from [17] related to the interpolation of group homomorphisms.

Let G and H be two Abelian groups. Given $S := \{(x_1, y_1), \ldots, (x_s, y_s)\} \subseteq G \times H$, we say that the set of points S *interpolates in a group homomorphism* if there exists a group homomorphism $f : G \longrightarrow H$ such that $f(x_i) = y_i$ for $i = 1, \ldots, s$. We say that a set of points $B \subseteq G \times H$ *interpolates in a group homomorphism with another set of points* $A \subseteq G \times H$ if $A \cup B$ interpolates in a group homomorphism.

Lemma 2 ([17]). *Let G, H be two finite Abelian groups. We denote by d the order of H and by p the smallest prime factor of d.*

1. *Let $x_1, \ldots, x_s \in G$ which span a subgroup denoted by G'. The following properties are equivalent. In this case, we say that $x_1, \ldots, x_s$ H-generate G.*
   (a) *For all $y_1, \ldots, y_s \in H$, there exists at most one group homomorphism $f : G \longrightarrow H$ such that $f(x_i) = y_i$ for $i = 1, \ldots, s$.*
   (b) *$G' + dG = G$.*
2. *Let $x_1, \ldots, x_s \in G$ which H-generate G. The mapping $g : G \times \mathbf{Z}_d^s \to G$ which is defined by $g(r, a_1, \ldots, a_s) := dr + a_1x_1 + \cdots + a_sx_s$ is balanced.*
3. *Given a set of s points $S = \{(x_1, y_1), \ldots, (x_s, y_s)\}$, such that $x_1, \ldots, x_s$ H-generate G. We assume that there exists a function $f : G \longrightarrow H$ such that*

$$\rho := \Pr_{(r,a_1,\ldots,a_s) \in_U G \times \mathbf{Z}_d^s} [f(dr + a_1x_1 + \cdots + a_sx_s) = a_1y_1 + \cdots + a_sy_s] > \frac{1}{p}.$$

The set of points S interpolates in a group homomorphism.

Although, our treatment uses arbitrary G, H, d, p, the implementation analysis of [16] suggests that parameters $G = \mathbf{Z}_n^*$ (for n product of two primes), $d = p = 2$ lead to the most efficient protocols for the signer. The homomorphisms are the Legendre symbols in G.

n-S-GHI Problem (Group Homomorphism Interpolation Prob. [17])
Parameters: Two Abelian groups G and H, a set of s points $S \subseteq G \times H$, and $n \in \mathbf{N}$.
Instance Generation: n elements $x_1, \ldots, x_n \in_U G$ are picked uniformly at random.
Problem: Find $y_1, \ldots, y_n \in H$ such that $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ interpolates with S in a group homomorphism.
The success probability of an n-S-GHIP solver $\mathcal{A}$ will be denoted by $\mathsf{Succ}_{\mathcal{A}}^{n\text{-}S\text{-}\mathsf{GHIP}}$.

n-S-GHID Problem (n-S-GHI Decisional Problem)
Parameters: Two Abelian groups G and H, a set of s points $S \subseteq G \times H$ and $n \in \mathbf{N}$.
Instance Generation: The instance T is generated according to one of the two following ways and is denoted T_0 or T_1 respectively. T_0 is a set of points $\{(x_1, y_1), \ldots, (x_n, y_n)\} \in (G \times H)^n$ picked uniformly at random such that it interpolates with S in a group homomorphism. T_1 is picked uniformly at random in $(G \times H)^n$.
Problem: Decide whether the instance T is of type T_0 or T_1.
The advantage of an n-S-GHID distinguisher $\mathcal{D}$ is given by

$$\mathsf{Adv}_{\mathcal{D}}^{n\text{-}S\text{-}\mathsf{GHID}} := \left|\Pr[b = 0 \mid T \text{ is of type } T_0] - \Pr[b = 0 \mid T \text{ is of type } T_1]\right|,$$

where b denotes the output bit of $\mathcal{D}$.

The S-GHI (resp. S-GHID) problem defined in [17] corresponds to the 1-S-GHI (resp. 1-S-GHID) problem. We consider the n-S-GHI and n-S-GHID problems for sets S which interpolate in a unique group homomorphism. Hence, S defines a homomorphism. The n-S-GHI problem consists in computing it on n elements. The n-S-GHID problem consists in deciding whether a set of points T is in its graph.

4.2 Interactive Proofs

The original version of the MOVA scheme makes use of two 4-move interactive proofs, namely one for the confirmation protocol and one for the denial protocol. In the first proof, a prover proves that a set of points interpolates in a group homomorphism known by himself. In the second one, the prover knows a group homomorphism which interpolates in a set of points S and proves that a second set of points T does not interpolate in this group homomorphism. These two proofs, taken from [17], are given below. Again, G, H denote two Abelian groups and $d := |H|$ is the order of H with smallest prime factor p. The group homomorphism which is known by the prover is denoted by f. The security parameter of the following proofs is an integer denoted by ℓ.

GHIproof$_\ell(S)$
Parameters: G, H, d
Input: ℓ, $S = \{(g_1, e_1), \ldots, (g_s, e_s)\} \subseteq G \times H$

1: The verifier picks $r_i \in_U G$ and $a_{i,j} \in_U \mathbf{Z}_d$ uniformly at random for $i = 1, \ldots, \ell$ and $j = 1, \ldots, s$. He computes $u_i = dr_i + a_{i,1}g_1 + \cdots + a_{i,s}g_s$ and $w_i = a_{i,1}e_1 + \cdots + a_{i,s}e_s$ for $i = 1, \ldots, \ell$. He sends $u_1, \ldots, u_\ell$ to the prover.
2: The prover computes $v_i = f(u_i)$ for $i = 1, \ldots, \ell$. He sends to the verifier a commitment to $v_1, \ldots, v_\ell$.
3: The verifier sends all r_i's and $a_{i,j}$'s to the prover.
4: The prover checks that the u_i's computations are correct. He then opens his commitment.
5: The verifier checks that $v_i = w_i$ for $i = 1, \ldots, \ell$.

coGHIproof$_\ell(S, T)$
Parameters: G, H, d, p
Input: ℓ, $S = \{(g_1, e_1), \ldots, (g_s, e_s)\}$, $T = \{(x_1, z_1), \ldots, (x_t, z_t)\}$

1: The verifier picks $r_{i,k} \in_U G$, $a_{i,j,k} \in_U \mathbf{Z}_d$, and $\lambda_i \in_U \mathbf{Z}_p$ uniformly at random for $i = 1, \ldots, \ell$, $j = 1, \ldots, s$, $k = 1, \ldots, t$. He computes $u_{i,k} := dr_{i,k} + \sum_{j=1}^{s} a_{i,j,k}g_j + \lambda_i x_k$ and $w_{i,k} := \sum_{j=1}^{s} a_{i,j,k}e_j + \lambda_i z_k$. Set $u := (u_{1,1}, \ldots, u_{\ell,t})$ and $w := (w_{1,1}, \ldots, w_{\ell,t})$. He sends u and w to the prover.
2: The prover computes $v_{i,k} := f(u_{i,k})$ and $y_k := f(x_k)$ for $i = 1, \ldots, \ell$, $k = 1, \ldots, t$. Since $w_{i,k} - v_{i,k} = \lambda_i(z_k - y_k)$, he should be able[1]

[1] Note that this requires to select H in which one can extract discrete logarithms lying in the restricted set $\{0, 1, \ldots, p-1\}$. In practice, this may not be a problem since we prefer $p = 2$ as shown in [16].

to find every λ_i if the verifier is honest since $z_k \neq y_k$ for at least one k. Otherwise, he sets λ_i to a random value. He then sends a commitment to $\lambda = (\lambda_1, \ldots, \lambda_\ell)$ to the verifier.

3: The verifier sends all $r_{i,k}$'s and $a_{i,j,k}$'s to the prover.

4: The prover checks that u and w were correctly computed. He then opens the commitment to λ.

5: The verifier checks that the prover could find the right λ.

In the original article [17], a 2-move variant for these two protocols was suggested without a proof. The variant is achieved by removing the two messages sent in the middle of the protocol for achieving the zero-knowledge property through the commitment scheme. In order to maintain zero-knowledge, the verifier sends a kind of commitment on a seed which generates the challenges to the prover. This commitment can only be opened by the prover after this one solved the challenges. We notably modify the original 2-move protocols by adding a trapdoor one-way permutation with associated secret key $\mathcal{K}_{\mathrm{s}}^{\mathbf{V}}$. Namely, we consider the permutation $\mathsf{TPOW}_{\mathcal{K}_{\mathrm{p}}^{\mathbf{V}}}(\cdot)$ and its inverse $\mathsf{TPOW}_{\mathcal{K}_{\mathrm{s}}^{\mathbf{V}}}(\cdot)^{-1}$. We denote $\mathsf{Succ}_{\mathcal{A}}^{\mathsf{inv\text{-}tp}}$ the probability that an adversary $\mathcal{A}$ can compute $\mathsf{TPOW}_{\mathcal{K}_{\mathrm{s}}^{\mathbf{V}}}^{-1}(y)$ given a random y, without knowing $\mathcal{K}_{\mathrm{s}}^{\mathbf{V}}$. For the sake of simplicity, we use the same notation for both protocols. The 2-move variant of GHIproof is given here.

2-GHIproof$_\ell(S)$
Parameters: G, H, d
Input: ℓ, $S = \{(g_1, e_1), \ldots, (g_s, e_s)\} \subseteq G \times H$

1: The verifier picks seedC $\in_U \{0,1\}^{k_c}$ uniformly at random, and by applying a pseudorandom generator GenC on this seed, generates values $r_i \in G$ and $a_{i,j} \in \mathbf{Z}_d$ for $i = 1, \ldots, \ell$ and $j = 1, \ldots, s$. He computes $u_i = dr_i + a_{i,1}g_1 + \cdots + a_{i,s}g_s$, $w_i = a_{i,1}e_1 + \cdots + a_{i,s}e_s$ for $i = 1, \ldots, \ell$, and $\vartheta_c = \mathsf{TPOW}_{\mathcal{K}_{\mathrm{p}}^{\mathbf{V}}}(\text{seedC})$. Using a cryptographic hash function $H_c : \{0,1\}^* \to \{0,1\}^{k_c}$, the verifier computes $h_c := H_c(w_1, \ldots, w_\ell) \oplus \text{seedC}$. He sends $u_1, \ldots, u_\ell$, h_c and ϑ_c to the prover.

2: The prover computes the values $v_i = f(u_i)$ for $i = 1, \ldots, \ell$ and $\text{seedC}' = H_c(v_1, \ldots, v_\ell) \oplus h_c$. He checks that $\vartheta_c = \mathsf{TPOW}_{\mathcal{K}_{\mathrm{p}}^{\mathbf{V}}}(\text{seedC}')$ and that GenC(seedC$'$) generates values $a_{i,j}$'s and r_i's such that $u_i := dr_i + a_{i,1}g_1 + \cdots + a_{i,s}g_s$ for $i = 1, \ldots, \ell$. He sends seedC$'$ to the verifier.

3: The verifier checks that seedC$'$ = seedC.

The interactive proof **coGHIproof** can be transformed in a 2-move protocol in a similar way. Namely, the verifier picks seedD $\in \{0,1\}^{k_d}$, and uses a pseudorandom generator GenD to generate the $r_{i,k}$'s, $a_{i,j,k}$'s, and λ_i's, and $\vartheta_d = \mathsf{TPOW}_{\mathcal{K}_{\mathrm{p}}^{\mathbf{V}}}(\text{seedD})$. He then sends the corresponding u, w, $h_d := H_d(\lambda_1, \ldots, \lambda_\ell) \oplus$ seedD, and ϑ_d, where $H_d : \{0,1\}^* \to \{0,1\}^{k_d}$ is a cryptographic hash function. In step 2 of the protocol, the prover retrieves seedD$'$, and checks whether $\vartheta_d = \mathsf{TPOW}_{\mathcal{K}_{\mathrm{p}}^{\mathbf{V}}}(\text{seedD}')$ and GenD(seedD$'$) generates the right u, w. Then, he sends seedD$'$.

Note that the complexity of both protocols are comparable to their 4-move variants.

4.3 MOVA Description

Below, we briefly present the MOVA scheme. For a more detailed description, we refer to [17].

Setup. The signer chooses two Abelian groups Xgroup and Ygroup and a secret group homomorphism Hom : Xgroup $\rightarrow$ Ygroup. He picks seedK $\in \{0,1\}^{k_s}$ and using a pseudorandom generator GenK generates Lkey values $\mathrm{Xkey}_1, \ldots, \mathrm{Xkey}_{\mathrm{Lkey}} \in$ Xgroup. Then, he computes $\mathrm{Ykey}_i := \mathrm{Hom}(\mathrm{Xkey}_i)$ for $i = 1, \ldots, \mathrm{Lkey}$.

Public Key. $\mathcal{K}_{\mathrm{p}}^{\mathbf{S}} := (\mathrm{Xgroup}, \mathrm{Ygroup}, d, \mathrm{seedK}, (\mathrm{Ykey}_1, \ldots, \mathrm{Ykey}_{\mathrm{Lkey}}), \mathrm{para})$, where the set para $= (\mathrm{Lkey}, \mathrm{Lsig}, \mathrm{Icon}, \mathrm{Iden}, k_c, k_d, k_s)$ is composed of integer parameters.

Secret Key. $\mathcal{K}_{\mathrm{s}}^{\mathbf{S}} := \mathrm{Hom}$.

The main goal of the setup is to ensure that the points $(\mathrm{Xkey}_i, \mathrm{Ykey}_i)$'s uniquely characterize Hom to avoid that several secret keys correspond to the same public key. This is necessary to guarantee the non-repudiation of the signature scheme. For this, one can either put many enough points or produce an interactive or non-interactive zero-knowledge proof of unique interpolation. These additional setup variants are described in [17]. In fact, the different setup variants ensure that $\mathrm{Xkey}_1, \ldots, \mathrm{Xkey}_{\mathrm{Lkey}}$ Ygroup-generate Xgroup. In this case, we say that the public key is *valid*.

Signature Generation. Let $m \in \{0,1\}^*$ be a message. Applying a pseudorandom generator GenS on the message m, the signer generates Lsig values $\mathrm{Xsig}_1, \ldots, \mathrm{Xsig}_{\mathrm{Lsig}} \in$ Xgroup. He then computes $\mathrm{Ysig}_i := \mathrm{Hom}(\mathrm{Xsig}_i)$ for $i = 1, \ldots, \mathrm{Lsig}$. The signature σ is $(\mathrm{Ysig}_1, \ldots, \mathrm{Ysig}_{\mathrm{Lsig}})$.

Confirmation Protocol. Given a message-signature pair (m, σ) as input and an integer Icon a security parameter, the signer (prover) and the verifier retrieve the values Xkey_i's, Xsig_j's from the message and the public key. The signer checks the validity of the signature. If this one is valid, the signer and the verifier run $\mathbf{GHIproof}_{\mathrm{Icon}}(S)$ on the set

$$S = \{(\mathrm{Xkey}_i, \mathrm{Ykey}_i) | i = 1, \ldots, \mathrm{Lkey}\} \cup \{(\mathrm{Xsig}_j, \mathrm{Ysig}_j) | j = 1, \ldots, \mathrm{Lsig}\}.$$

Otherwise, the signer aborts.

Denial Protocol. Given an alleged invalid message-signature pair (m, σ) as input and an integer Iden a security parameter, we denote the signature $\sigma = (\mathrm{Zsig}_1, \ldots, \mathrm{Zsig}_{\mathrm{Lsig}})$. The signer and the verifier retrieve the Xkey_i's and Xsig_j's. The signer checks the invalidity of (m, σ). If this one is really invalid, they run the protocol $\mathbf{coGHIproof}_{\mathrm{Iden}}(S, T)$ on the sets

$$S = \{(\mathrm{Xkey}_i, \mathrm{Ykey}_i) | i = 1, \ldots, \mathrm{Lkey}\} \quad T = \{(\mathrm{Xsig}_j, \mathrm{Zsig}_j) | j = 1, \ldots, \mathrm{Lsig}\}.$$

The 2-move version of MOVA is exactly as above except that **GHIproof** and **coGHIproof** are replaced by **2-GHIproof** and **2-coGHIproof** respectively.

5 Security of the 2-Move MOVA Scheme

Here, we prove that the 2-move modified version of the MOVA scheme satisfies the security properties mentioned in Section 3. The proofs of resistance against forgery attacks and invisibility were inspired from [13].

Theorem 3. *Let* $S = \{(\mathrm{Xkey}_1, \mathrm{Ykey}_1), \ldots, (\mathrm{Xkey}_{\mathrm{Lkey}}, \mathrm{Ykey}_{\mathrm{Lkey}})\}$ *and* e *denote the natural logarithm base. Assuming that* GenC, GenS, GenD, H_d, *and* H_c *are random oracles, that signer's public key is valid, and that* TPOW *is a trapdoor one-way permutation, the* MOVA *scheme with 2-move confirmation and denial protocols satisfies the following security properties.*

1. *The confirmation (resp. denial) protocol is complete.*
2. *Let* p *be the smallest prime factor of* d*. The confirmation (resp. denial) protocol is sound: for any invalid (valid) message-signature pair, any cheating signer* $\mathbf{S}^*$ *limited to* q_{H_c} *(resp.* q_{H_d}*) queries to* H_c *(resp.* H_d*), is such that the probability* $\mathsf{Succ}^{\mathsf{sd\text{-}con}}_{\mathbf{S}^*} < \mathsf{Succ}^{\mathsf{inv\text{-}tp}} + q_{H_c} p^{-\mathrm{Icon}}$ *(resp.* $\mathsf{Succ}^{\mathsf{sd\text{-}den}}_{\mathbf{S}^*} < \mathsf{Succ}^{\mathsf{inv\text{-}tp}} + q_{H_d} p^{-\mathrm{Iden}}$*), where* $\mathsf{Succ}^{\mathsf{inv\text{-}tp}}$ *is the maximum of* $\mathsf{Succ}^{\mathsf{inv\text{-}tp}}_{\mathcal{A}}$ *among all adversaries* $\mathcal{A}$ *which have similar complexity as* $\mathbf{S}^*$*.*
3. *The confirmation (resp. denial) protocol is perfect non-transferable.*
4. *The confirmation (resp. denial) protocol is statistical black-box straight-line zero-knowledge.*
5. *Assume that for any solver* $\mathcal{B}$ *with a given complexity, we have*

$$\mathsf{Succ}^{\text{Lsig-}S\text{-}\mathsf{GHIP}}_{\mathcal{B}} \leq \varepsilon.$$

   Then, any forger $\mathcal{F}$ *with similar complexity using* q_S *signing queries and* q_V *queries to the confirmation/denial oracle wins the existential forgery game under an adaptive chosen-message attack with a probability*

$$\mathsf{Succ}^{\mathsf{ef\text{-}cma}}_{\mathcal{F}} \leq e(1+q_S)(1+q_V)\varepsilon.$$

6. *Assume that for any algorithm* $\mathcal{B}$ *with a given complexity, we have*

$$\mathsf{Adv}^{\text{Lsig-}S\text{-}\mathsf{GHID}}_{\mathcal{B}} \leq \varepsilon \quad \textit{and} \quad \mathsf{Succ}^{\text{Lsig-}S\text{-}\mathsf{GHIP}}_{\mathcal{B}} \leq \varepsilon'.$$

   Then, any distinguisher $\mathcal{D}$ *with similar complexity using* q_S *signing queries and* q_V *queries to the confirmation/denial oracle wins the invisibility game under a chosen-message attack with advantage*

$$\mathsf{Adv}^{\mathsf{inv\text{-}cma}}_{\mathcal{D}} \leq e(1+q_S)(\varepsilon + 2e(1+q_V)\varepsilon').$$

Remark 4. The soundness and zero-knowledge of the confirmation and denial protocols as well as the invisibility and the resistance to existential forgery attacks hold in the random oracle model.

Remark 5. Similarly to [14], the efficiency of the security reduction for the existential forgery can be improved (factor $(1+q_V)^{-1}$ is removed) by replacing GHI problem by its *gap* variant [20]. This problem consists in solving the GHI problem using an access to an oracle which solves the GHID problem. This one helps to simulate the confirmation and denial oracles.

Proof. Below we prove Theorem 3. Completeness is omitted since it is obvious.

Soundness of Confirmation. Let $\mathbf{S}^*$ be a cheating prover who wants to confirm the validity of an invalid signature $\sigma = (\mathrm{Zsig}_1, \ldots, \mathrm{Zsig}_{\mathrm{Lsig}})$. Note that $\mathbf{S}^*$ is fed with the signer secret key $\mathcal{K}_s^{\mathbf{S}}$. Without loss of generality, we can assume that $\mathbf{S}^*$ always responds correctly to the verifier whenever he queries seedC to GenC. Indeed, he can check that seedC is the preimage of ϑ_c by TPOW and answer seedC to the challenge if correct. (With an honest verifier, there is no need to check whether the challenge is valid.) Hence, the verifier always accepts when the prover queries seedC to GenC. Similarly, we can assume that $\mathbf{S}^*$ always responds correctly to the verifier whenever he queries the right w to H_c because he can deduce seedC from h_c afterwards. Note that when $\mathbf{S}^*$ interacts with an honest verifier, the verifier only accepts if $\mathbf{S}^*$ outputs seedC.

We transform $\mathbf{S}^*$ into an algorithm to invert the trapdoor permutation as follows.

1. We receive a random challenge ϑ_c, whose preimage by TPOW is denoted seedC.
2. We generate the key material for the MOVA signature and generate some random values r_i's and $a_{i,j}$'s. We deduce some u_i's and w_i's and pick a random h_c. Then (u, h_c, ϑ_c) is a challenge for the prover. We simulate GenC as follows: for any query except seedC (we can check whether a value is seedC by checking that its image by TPOW is ϑ_c) we simulate a random oracle as usual i.e., we maintain a list of elements queried to GenC with corresponding answers and simulate according to this list. If the query is new, we simply pick the answer at random and add the pair in the list. For the query seedC we stop the overall simulation and yield seedC: the inversion of ϑ_c succeeded. We simulate H_c as follows: for any query except $w = (w_1, \ldots, w_{\mathrm{Icon}})$ we simulate a random oracle (like for GenC). For the query w we stop: the inversion of ϑ_c failed.
3. We run $\mathbf{S}^*$ according to our simulation rules. If $\mathbf{S}^*$ outputs some value, we check whether it is seedC. If it is, we output it, otherwise we fail.

The algorithm succeeds to invert the trapdoor permutation at the condition that either (event A) $\mathbf{S}^*$ succeeds without even querying seedC to GenC nor w to H_c, or (event B) that $\mathbf{S}^*$ queries seedC to GenC without querying w to H_c beforehand. Let C be the event that $\mathbf{S}^*$ queries w to H_c before querying seedC to GenC. Since the simulation is perfect, $\Pr[A \cup B] + \Pr[C]$ is the probability that $\mathbf{S}^*$ passes the protocol with an honest verifier. We have $\Pr[A \cup B] \leq \mathsf{Succ}^{\mathsf{inv\text{-}tp}}$. Below we show an upper bound for $\Pr[C]$. To this, we consider a simulator $\mathcal{B}$ which plays with $\mathbf{S}^*$ to win the following game:
Game: A challenger picks elements r_i's and $a_{i,j}$'s uniformly at random and compute $u_i = dr_i + \sum_{j=1}^{\mathrm{Lkey}} a_{i,j}\mathrm{Xkey}_j + \sum_{j=\mathrm{Lkey}+1}^{\mathrm{Lkey}+\mathrm{Lsig}} a_{i,j}\mathrm{Xsig}_{j-\mathrm{Lkey}}$ and $w_i := \sum_{j=1}^{\mathrm{Lkey}} a_{i,j}\mathrm{Ykey}_j + \sum_{j=\mathrm{Lkey}+1}^{\mathrm{Lkey}+\mathrm{Lsig}} a_{i,j}\mathrm{Zsig}_{j-\mathrm{Lkey}}$. The simulator $\mathcal{B}$ receives the u_i's and wins the game if he finds all the values w_i's.

$\mathcal{B}$ simply forwards the received challenges u_i's and picks h_c and ϑ_c uniformly at random in $\{0,1\}^{k_c}$. $\mathcal{B}$ simulates the oracle H_c as above, except that he guesses when the w_i's are queried. For this, he just picks an integer $\ell \in \{1, \ldots, q_{H_c}\}$

uniformly and stops the simulation at the ℓth query made to H_c. The simulator then answers the values w_i's. Note that $\mathbf{S}^*$ cannot query seedC to GenC when event C occurs. The simulation is perfect in the C case provided that ℓ is correctly guessed. Thus, we have $\Pr[D] \geq 1/q_{H_c} \cdot \Pr[C]$, where D denotes the event of winning the above game. By the assertion 3 of Lemma 2, $\Pr[D] \leq p^{-\mathrm{Icon}}$. Thus, $\Pr[C] \leq q_{H_c} p^{-\mathrm{Icon}}$. So, the confirmation cannot succeed with probability larger than $\mathsf{Succ}^{\mathsf{inv\text{-}tp}} + q_{H_c} p^{-\mathrm{Icon}}$.

Soundness of Denial. This proof works in a very similar way as for the confirmation. The only difference is that we replace GenC by GenD, H_c by H_d, Icon by Iden, k_c by k_d, seedC by seedD.

Non-Transferability of Confirmation. We describe a simulator $\mathcal{B}$ interacting with $\tilde{\mathbf{V}}$. First, $\mathcal{B}$ launches $\tilde{\mathbf{V}}$ and receives the first message (which should be $u = (u_1, \ldots, u_{\mathrm{Icon}})$, h_c, and ϑ_c). If (m, σ) is valid, the simulator computes $\mathrm{seedC}' = \mathsf{TPOW}^{-1}_{\mathcal{K}_{\mathrm{s}}^{\mathbf{V}}}(\vartheta_c)$ and using GenC generates coefficients a'_{ij} and r'_i and corresponding u'_i and w'_i for $i = 1, \ldots, \mathrm{Icon}$ and $j = 1, \ldots, \mathrm{Lkey} + \mathrm{Lsig}$. Then, $\mathcal{B}$ checks whether $u'_i = u_i$ for $i = 1, \ldots, \mathrm{Icon}$, $\mathrm{seedC}' = H_c(w'_1, \ldots, w'_{\mathrm{Icon}}) \oplus h_c$. If it is the case, $\mathcal{B}$ outputs the transcript $(h_c, w, \vartheta_c, \mathrm{seedC}')$. Otherwise, it outputs $(h_c, w, \vartheta_c, \mathsf{abort})$. If (m, σ) is invalid, the simulator outputs abort. Note that an honest signer would check exactly the same equalities (in a different way) and would answer exactly in the same way. Hence, the non-transferability is perfect.

Non-transferability of Denial. This proof is similar.

Straight-Line Zero-Knowledge of Confirmation. If $\mathbf{V}^*$ is given $\mathcal{K}_{\mathrm{s}}^{\mathbf{V}}$, the simulation can be done perfectly as for the non-transferability. Now, we consider that $\mathbf{V}^*$ (and the simulator $\mathcal{B}$) is not given $\mathcal{K}_{\mathrm{s}}^{\mathbf{V}}$. $\mathcal{B}$ runs the verifier $\mathbf{V}^*$ and looks at the queries made by $\mathbf{V}^*$ to the oracle GenC. $\mathcal{B}$ puts these q_{GenC} queries seedC_k for $1 \leq k \leq q_{\mathrm{GenC}}$ as well as the corresponding answers of GenC in memory. The simulator then receives the first message M of $\mathbf{V}^*$. If this one has not a correct format, the simulator outputs the transcript (M, abort). Otherwise, the simulator checks whether one answer among those queries seedC_k's made to GenC generates the challenges u_i's correctly and the image of this query by TPOW is equal to ϑ_c. If it is not the case, $\mathcal{B}$ outputs the transcript $(u_1, \ldots, u_{\mathrm{Icon}}, h_c, \vartheta_c, \mathsf{abort})$. Otherwise, the simulator is able to compute the right w_i's from this answer (the right r_i's and $a_{i,j}$'s) using the homomorphic property of Hom, namely $w_i = \mathrm{Hom}(u_i) = \sum_{j=1}^{\mathrm{Lkey}} a_{i,j} \mathrm{Ykey}_j + \sum_{j=\mathrm{Lkey}+1}^{\mathrm{Lkey}+\mathrm{Lsig}} a_{i,j} \mathrm{Ysig}_{j-\mathrm{Lkey}}$ for $i = 1, \ldots, \mathrm{Icon}$. From the w_i's, $\mathcal{B}$ computes $\mathrm{seedC}^* := h_c \oplus H_c(w_1, \ldots, w_{\mathrm{Icon}})$ and checks whether seedC^* generates the right r_i's and $a_{i,j}$'s. In the positive case, $\mathcal{B}$ outputs the transcript $(u_1, \ldots, u_{\mathrm{Icon}}, h_c, \vartheta_c, \mathrm{seedC}^*)$. In the negative case, it outputs the following transcript $(u_1, \ldots, u_{\mathrm{Icon}}, h_c, \vartheta_c, \mathsf{abort})$.

It remains to show that the two transcript distributions are statistically indistinguishable. When the first message has not a correct format, the two transcripts are clearly identical. Let consider the case where the verifier did not query any

seedC_k which produces the challenges u_i's and whose image by TPOW leads to ϑ_c. In this case, the honest prover will not abort the protocol only if he retrieves a seedC $= H(w_1, \ldots, w_{\text{Icon}}) \oplus h_c$ which generates the challenges u_i's and ϑ_c. This occurs only if the verifier $\mathbf{V}^*$ was able to guess that the output values of the query seedC to the oracle GenC generate the right r_i's and a_{ij}'s. Since GenC is a random oracle, no polynomial time verifier $\mathbf{V}^*$ can succeed to do that with a non-negligible probability. We still have to consider the case where the verifier queried a seedC_k which produces the challenges u_i's and ϑ_c. We see that the two transcripts are always identical, since the simulator clearly knows the answer of the honest prover by learning the right w_i's. Therefore, we can conclude that the two transcript distributions are statistically indistinguishable.

Straight-Line Zero-Knowledge of Denial. This proof is similar.

Unforgeability. Let $\mathcal{F}$ be a forger who succeeds to existentially forge a signature under an adaptive chosen-message attack with a non-negligible probability ε. We will construct an algorithm $\mathcal{B}$ which solves the Lsig-S-GHI problem with $S := \{(\text{Xkey}_1, \text{Ykey}_1), \ldots, (\text{Xkey}_{\text{Lkey}}, \text{Ykey}_{\text{Lkey}})\}$ using the forger $\mathcal{F}$ and $\mathcal{K}_{\text{s}}^{\mathbf{V}}$. At the beginning, $\mathcal{B}$ receives the challenges $x_1, \ldots, x_{\text{Lsig}} \in$ Xgroup of the Lsig-S-GHI problem. Then, $\mathcal{B}$ runs the forger and simulates the queries to the random oracle GenS, q_S queries to the signing oracle Sign and q_V queries to the denial/confirmation oracle Ver. We can assume that all messages sent to Sign resp. Ver were previously queried to GenS (since the oracle Sign resp. Ver has to make such queries anyway). $\mathcal{B}$ simulates the oracles GenS and Sign as follows:

GenS. For each message m queried to GenS, $\mathcal{B}$ maintains a list of each message and corresponding answer $(m, \text{Xsig}_1, \ldots, \text{Xsig}_{\text{Lsig}})$. If the message was already queried, $\mathcal{B}$ outputs the corresponding answer in the list. Otherwise, he picks $a_{i,j} \in_U \mathbf{Z}_d$ and $r_i \in_U$ Xgroup uniformly at random for $1 \leq i \leq$ Lsig, $1 \leq j \leq$ Lkey. With probability q, he answers $\text{Xsig}_i := dr_i + \sum_{j=1}^{\text{Lkey}} a_{i,j}\text{Xkey}_j$ for $i = 1, \ldots,$ Lsig. We call it type-1 answer. With probability $1-q$, the answer is $\text{Xsig}_i := dr_i + x_i + \sum_{j=1}^{\text{Lkey}} a_{i,j}\text{Xkey}_j$ for $i = 1, \ldots,$ Lsig. We call it type-2 answer. For each message, $\mathcal{B}$ keeps the coefficients $a_{i,j}$'s and r_i's and answer type in memory. Note that the simulation is perfect by the assertion 2 of Lemma 2, since the public key is valid.

Sign. For a message m, if the answer to the GenS query of m was of type-1, then $\mathcal{B}$ answers $\text{Ysig}_i := \sum_{j=1}^{\text{Lkey}} a_{i,j}\text{Ykey}_j$ for $i = 1, \ldots,$ Lsig. Otherwise, it aborts the simulation.

Let (m_i, σ_i) denote the ith query to Ver for $1 \leq i \leq q_V$ and $(m_{q_V+1}, \sigma_{q_V+1})$ denote the $\mathcal{F}$ output. In order to simulate the answers of the queries made to Ver, $\mathcal{B}$ guesses the smallest i such that (m_i, σ_i) is a valid forged pair (i.e., m was not queried to Sign). To this, $\mathcal{B}$ simply picks ℓ uniformly at random in $\{1, \ldots q_V + 1\}$. $\mathcal{B}$ deals with the ith query as follows:

$i < \ell$. To any query (m_i, σ_i), $\mathcal{B}$ checks whether m_i was submitted to Sign. If it is the case, $\mathcal{B}$ is able to decide whether (m_i, σ_i) is valid and simulates

the appropriate protocol. Otherwise, $\mathcal{B}$ guesses that (m_i, σ_i) is invalid and simulate the appropriate protocol. The simulation is done as the simulator in the proof of non-transferability of the confirmation (resp. denial) protocol.

$i = \ell$. Let $(m_\ell, \sigma_\ell) := (m_\ell, \mathrm{Ysig}_1, \ldots, \mathrm{Ysig}_{\mathrm{Lsig}})$. If the corresponding Xsig_i's were of type-1, $\mathcal{B}$ aborts. Otherwise, when ℓ was correctly guessed $\mathrm{Ysig}_i = y_i + \sum_{j=1}^{\mathrm{Lkey}} a_{i,j}\mathrm{Ykey}_j$ and $\mathcal{B}$ is able to deduce the y_i's of the Lsig-S-GHI problem.

It remains to compute the probability that $\mathcal{B}$ retrieves the y_i's and did not abort. This event occurs if $\mathcal{B}$ is able to simulate all Sign queries, guess the right ℓ and use the message m_ℓ to deduce the y_i's. Therefore, $\Pr[\mathcal{B}\text{ succeeds}|\mathcal{F}\text{ succeeds}] = q^{q_S}(1-q)/(q_V+1)$. As for the full-domain hash technique [7] and as in [13], the optimal $q_{\mathrm{opt}} = q_S/(q_S+1)$. Thus, the success probability is greater or equal to $(1/e(1+q_S)(1+q_V))\varepsilon$.

Invisibility. Let $\mathcal{D}$ be a distinguisher which breaks the invisibility of the MOVA scheme with an advantage ε. We construct an algorithm $\mathcal{B}$ which solves the Lsig-S-GHID problem by using $\mathcal{D}$ and $\mathcal{K}_{\mathrm{s}}^{\mathbf{V}}$. At the beginning, $\mathcal{B}$ is challenged with a tuple $\{(x_1, y_1), \ldots, (x_{\mathrm{Lsig}}, y_{\mathrm{Lsig}})\} \in (\mathrm{Xgroup} \times \mathrm{Ygroup})^{\mathrm{Lsig}}$ for which it has to decide whether $\mathrm{Hom}(x_i) = y_i$ for all $1 \leq i \leq \mathrm{Lsig}$ or if this tuple was picked at random. Like for the proof of the existential forgery, the simulator $\mathcal{B}$ runs $\mathcal{D}$ and simulates the queries to the random oracle GenS, q_S queries to the signing oracle Sign and the queries to the denial/confirmation oracle Ver. We can assume that each message queried to Sign or Ver was previously queried to the random oracle GenS. We assume that no query m to Ver was submitted to Sign beforehand. (Otherwise, we can just simulate them with $\mathcal{K}_{\mathrm{s}}^{\mathbf{V}}$.) Let Forge be the event in which $\mathcal{D}$ sends a valid message-signature pair to Ver. We first remove all instances for which the event Forge occurs. So, we can now assume that $\mathcal{D}$ never submits any valid pair (m, σ) to Ver such that m was not previously submitted to Sign. $\mathcal{B}$ simulates the oracles just like in the proof of unforgeability with $\ell = q_V + 1$ (we excluded valid forged pairs).

After a given time, the distinguisher $\mathcal{D}$ sends a message m^* to the challenger of the invisibility game which is simulated by $\mathcal{B}$. We can assume that m^* was queried to GenS (otherwise $\mathcal{B}$ simulates a new query). If the answer of m^* to GenS was of type-1, $\mathcal{B}$ aborts the simulation. Otherwise, it sends the challenge signature $(\mathrm{Ysig}_1^*, \ldots, \mathrm{Ysig}_{\mathrm{Lsig}}^*)$ where $\mathrm{Ysig}_i^* := y_i + \sum_{j=1}^{\mathrm{Lkey}} a_{i,j}\mathrm{Ykey}_j$ for $1 \leq i \leq \mathrm{Lsig}$. Then, $\mathcal{D}$ continues to query the oracles which are simulated by $\mathcal{B}$ as above.

Finally, $\mathcal{D}$ outputs a guess bit b'. The simulator $\mathcal{B}$ outputs the same bit b' as guess bit to the Lsig-S-GHID challenger or a random bit when $\mathcal{B}$ aborted.

Using the homomorphic property of Hom, we deduce that the set $\{(x_i, y_i)\}_{i=1}^{\mathrm{Lsig}}$ interpolates in a group homomorphism with the set of points S if and only if $(m^*, \mathrm{Ysig}_1^*, \ldots, \mathrm{Ysig}_{\mathrm{Lsig}}^*)$ is a valid message-signature pair. Hence, when the simulator does not abort and the event Forge does not occur, $\mathcal{B}$ perfectly simulates the invisibility games. It remains to compute the advantage of $\mathcal{B}$.

For a bit b, we denote A_b the probability event that $\mathcal{B}$ does not abort when the challenge to $\mathcal{B}$ was of the form T_b (thus, $\mathcal{B}$ simulates the game $\mathbf{Game}^{\mathsf{inv\text{-}cma}\text{-}b}$ to $\mathcal{D}$). Note that the probability $\Pr[A_1] = \Pr[A_0]$ can be bounded in an optimal

way as in the proof of existential forgery attacks, namely, by choosing q adequately we get $\Pr[A_1] \geq (1/e(1+q_S))$. We now define the events B_b and D_b which occur when $\mathcal{B}$ and $\mathcal{D}$ respectively outputs the bit 0 when the challenge was of the form T_b. Note that if A_b happens, both events B_b and D_b occurs simultaneously. Let us denote ε_0 resp. ε_1, the probability for $\mathcal{D}$ to output 0 in the game $\mathbf{Game}^{\mathsf{inv\text{-}cma\text{-}0}}$ resp. $\mathbf{Game}^{\mathsf{inv\text{-}cma\text{-}1}}$. We now estimate $\Pr[B_0|A_0]$ and $\Pr[B_1|A_1]$ with respect to ε_0 and ε_1. To this end, we notice that the event $B_0|A_0$ resp. $B_1|A_1$ occurs simultaneously with the event where $\mathcal{D}$ outputs 0 in the game $\mathbf{Game}^{\mathsf{inv\text{-}cma\text{-}0}}$ resp. $\mathbf{Game}^{\mathsf{inv\text{-}cma\text{-}1}}$, provided that the event Forge does not occur. Hence, applying the difference lemma of Shoup [22] leads to

$$|\Pr[B_b|A_b] - \varepsilon_b| \leq \Pr[\mathsf{Forge}]$$

for $b = 0, 1$. From this, we can deduce that $\Pr[B_0|A_0] \geq \varepsilon_0 - \Pr[\mathsf{Forge}]$ and $\Pr[B_1|A_1] \leq \varepsilon_1 + \Pr[\mathsf{Forge}]$. Without loss of generality, we can assume that $\Pr[B_0] \geq \Pr[B_1]$. The advantage of $\mathcal{B}$ is then equal to

$$\begin{aligned}\Pr[B_0] - \Pr[B_1] = \Pr[\neg A_0] \cdot (\Pr[B_0|\neg A_0] - \Pr[B_1|\neg A_1]) \\ + \Pr[A_0] \cdot (\Pr[B_0|A_0] - \Pr[B_1|A_1]).\end{aligned}$$

Since $\Pr[B_0|\neg A_0] = \Pr[B_1|\neg A_1] = 1/2$ and $\varepsilon_0 - \varepsilon_1 = \mathsf{Adv}_{\mathcal{D}}^{\mathsf{inv-cma}}$, we finally have

$$\mathsf{Adv}_{\mathcal{B}}^{\text{Lsig-}S\text{-}\mathsf{GHID}} \geq \frac{1}{(1+q_S)e}\left(\mathsf{Adv}_{\mathcal{D}}^{\mathsf{inv-cma}} - 2\Pr[\mathsf{Forge}]\right).$$

We can conclude by noting that Forge occurs with a probability bounded by $e(1+q_S)(1+q_V)\varepsilon'$ by assertion 5. □

Remark 6. MOVA scheme can be made probabilistic so that the invisibility notion defined in [8] is satisfied. To this, it suffices to append some randomness r to the message to sign and to add r in the signature. The drawback is that the signature enlarges.

Consequences for the Signature Parameters. One of the main advantage of MOVA scheme as stated in [17] is the fully scalable signature size. It was argued that one could potentially consider signatures of size of 20 bits, but the corresponding security level was not precisely quantified. Namely, the efficiency of the security reduction in [17] is not detailed and the security model did not consider queries to the confirmation/denial oracle. Our security reduction provides a more precise result. Assuming that any solver with same computational resource as a given forger cannot solve Lsig-S-GHI problem with a probability significatively greater than $|\text{Ygroup}|^{-\text{Lsig}}$, the assertion 5 of Theorem 3 shows that we have $\mathsf{Succ}_{\mathcal{F}}^{\mathsf{ef\text{-}cma}} \leq |\text{Ygroup}|^{-\text{Lsig}} e(q_S+1)(q_V+1)$. Note that the assumption can be reached by scaling Xgroup adequately, namely without any modification of the signature size. This is the case when Hom is the Legendre symbol $(\cdot/p)$ defined on an RSA modulus $n = pq$. A signature size of Lsig ≥ 52 bits achieves a success probability for the existential forgeability of at most 2^{-20} with $q_S = 2^{10}$ and

$q_V = 2^{20}$. Similarly, assuming that $\mathsf{Adv}_{\mathcal{B}}^{\text{Lsig-}S\text{-GHID}} \approx 0$ for any $\mathcal{B}$ with similar complexity as the invisibility distinguisher $\mathcal{D}$, assertion 6 of Theorem 3 shows that $\mathsf{Adv}_{\mathcal{D}}^{\mathsf{inv\text{-}cma}} \approx 2e^2 q_S q_V 2^{-\text{Lsig}}$, which leads to $\mathsf{Adv}_{\mathcal{D}}^{\mathsf{inv\text{-}cma}} \approx 2^{-18}$. Results for the soundness can be obtained with $\mathsf{Succ}^{\mathsf{inv\text{-}tp}} \approx 0$. For the 2-move verification protocols, we can achieve a soundness probability of 2^{-20} with Icon = Iden = 60, $q_{H_c} = q_{H_d} = 2^{40}$.

6 Conclusion

We revisited a 2-move variant of the MOVA undeniable signature scheme which was proposed without any proof. By using a trapdoor one-way permutation adequately, we were able to make the verification protocols non-transferable. All the other required security properties are thoroughly analyzed in the random oracle model, thereby allowing to quantify the security of the different properties in terms of the signature parameters. So, as far as we know, this is the first time a provably secure undeniable signature scheme with 2-move confirmation and denial protocols is obtained. This result shows that minimal number of moves in an undeniable signature with interactive protocols can be reached in practice.

References

1. B. Barak, Y. Lindell, and S. P. Vadhan. Lower Bounds for Non-Black-Box Zero Knowledge. In *44th Annual IEEE Symposium on Foundations of Computer Science, FOCS '03*, pages 384–393. IEEE Computer Society, 2003.
2. M. Bellare and P. Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *1st ACM Conference on Computer and Communications Security*, pages 62–73. ACM Press, 1993.
3. J. Boyar, D. Chaum, I. Damgård, and T. P. Pedersen. Convertible Undeniable Signatures. In *Advances in Cryptology – CRYPTO '90*, volume 537 of *Lecture Notes in Computer Science*, pages 189–205. Springer-Verlag, 1991.
4. J. Camenisch and M. Michels. Confirmer Signature Schemes Secure against Adaptive Adversaries. In *Advances in Cryptology – EUROCRYPT '00*, volume 1807 of *Lecture Notes in Computer Science*, pages 243–258. Springer-Verlag, 2000.
5. D. Chaum. Zero-Knowledge Undeniable Signatures. In *Advances in Cryptology – EUROCRYPT '90*, volume 473 of *Lecture Notes in Computer Science*, pages 458–464. Springer-Verlag, 1990.
6. D. Chaum and H. van Antwerpen. Undeniable Signatures. In *Advances in Cryptology – CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pages 212–217. Springer-Verlag, 1990.
7. J.-S. Coron. On the Exact Security of Full Domain Hash. In *Advances in Cryptology – CRYPTO '00*, volume 1880 of *Lecture Notes in Computer Science*, pages 229–235. Springer-Verlag, 2000.
8. S. D. Galbraith and W. Mao. Invisibility and Anonymity of Undeniable and Confirmer Signatures. In *Topics in Cryptology – CT–RSA '03*, volume 2612 of *Lecture Notes in Computer Science*, pages 80–97. Springer-Verlag, 2003.
9. R. Gennaro, H. Krawczyk, and T. Rabin. RSA-Based Undeniable Signatures. In *Advances in Cryptology – CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 132–149. Springer-Verlag, 1997.

10. O. Goldreich. *Foundations of Cryptography, Volume I Basic Tools*. Cambridge University Press, 2001.
11. S. Goldwasser, S. Micali, and R. L. Rivest. A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988.
12. M. Jakobsson, K. Sako, and R. Impagliazzo. Designated Verifier Proofs and Their Applications. In *Advances in Cryptology – EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 143–154. Springer-Verlag, 1996.
13. K. Kurosawa and S.-H. Heng. 3-Move Undeniable Signature Scheme. In *Advances in Cryptology – EUROCRYPT '05*, volume 3494 of *Lecture Notes in Computer Science*, pages 181–197. Springer-Verlag, 2005.
14. F. Laguillaumie and D. Vergnaud. Short Undeniable Signatures Without Random Oracles: the Missing Link. In *Progress in Cryptology – INDOCRYPT '05*, volume 3797 of *Lecture Notes in Computer Science*, pages 283–296. Springer-Verlag, 2005.
15. H. Lipmaa, G. Wang, and F. Bao. Designated Verifier Signature Schemes: Attacks, New Security Notions and a New Construction. In *Automata, Languages and Programming: 32nd International Colloquium, ICALP '05*, volume 3580 of *Lecture Notes in Computer Science*, pages 459–471. Springer-Verlag, 2005.
16. J. Monnerat, Y. A. Oswald, and S. Vaudenay. Optimization of the MOVA Undeniable Signature Scheme. In *Progress in Cryptology – MYCRYPT '05*, volume 3715 of *Lecture Notes in Computer Science*, pages 196–209. Springer-Verlag, 2005.
17. J. Monnerat and S. Vaudenay. Generic Homomorphic Undeniable Signatures. In *Advances in Cryptology – ASIACRYPT '04*, volume 3329 of *Lecture Notes in Computer Science*, pages 354–371. Springer-Verlag, 2004.
18. J. Monnerat and S. Vaudenay. Undeniable Signatures Based on Characters: How to Sign with One Bit. In *Public Key Cryptography – PKC '04*, volume 2947 of *Lecture Notes in Computer Science*, pages 69–85. Springer-Verlag, 2004.
19. W. Ogata, K. Kurosawa, and S.-H. Heng. The Security of the FDH Variant of Chaum's Undeniable Signature Scheme. In *Public Key Cryptography – PKC '05*, volume 3386 of *Lecture Notes in Computer Science*, pages 328–345. Springer-Verlag, 2005. Extended version available on: Cryptology ePrint Archive, Report 2004/290, `http://eprint.iacr.org/`.
20. T. Okamoto and D. Pointcheval. The Gap-Problems: A New Class of Problems for the Security of Cryptographic Schemes. In *Public Key Cryptography – PKC '01*, volume 1992 of *Lecture Notes in Computer Science*, pages 104–118. Springer-Verlag, 2001.
21. R. Pass. On Deniability in the Common Reference String and Random Oracle Model. In *Advances in Cryptology – CRYPTO '03*, volume 2729 of *Lecture Notes in Computer Science*, pages 316–337. Springer-Verlag, 2003.
22. V. Shoup. Sequences of Games: A Tool for Taming Complexity in Security Proofs. Cryptology ePrint Archive, Report 2004/332, 2004. `http://eprint.iacr.org/`.