Strongly Deniable Identification Schemes Immune to Prover's and Verifier's Ephemeral Leakage

Łukasz Krzywiecki(✉) and Marcin Słowik

Department of Computer Science, Faculty of Fundamental Problems of Technology, Wrocław University of Science and Technology, Wrocław, Poland
{lukasz.krzywiecki,marcin.slowik}@pwr.edu.pl

Abstract. In this paper, we consider *Identification Schemes* (IS) in the context of attacks against their deniability via Fiat-Shamir transformations. We address the following issue: How to design and implement a deniable IS, that is secure against ephemeral leakage on both a Prover's and a Verifier's side, and withstands attacks based on Fiat-Shamir transformation. We propose a new security model to address the leakage on the Verifier's side, extending the previous propositions [1]. During the *Query Stage*, we allow the malicious Verifier to set random values used on the Prover's side. Additionally, we allow malicious Prover to access ephemeral values of the Verifier during the *Impersonation Stage*. We introduce two generic constructions based on three-step IS. Finally, we provide an example scheme based on the extended construction from [1], which is provably deniable and secure in our new strong model.

Keywords: Identification scheme · Ephemeral secret setting · Ephemeral secret leakage · Deniability · Simulatability · Zero-knowledge proofs

1 Introduction

Identification schemes (IS) based on *Public Key Infrastructure* (PKI) allow a Prover, holding a secret key, to prove its possession via a zero-knowledge protocol executed with a Verifier holding a corresponding public key. There are two common requirements that IS should satisfy: (1) *security* - a malicious Prover should not be able to successfully complete the protocol without the corresponding secret key; (2) *privacy* - in some scenarios, the protocol should be deniable, meaning that its transcript must not be a strong proof of Prover's participation. Alternatively, there are cases in which the protocol should not be deniable and must provide a strong proof of Prover's participation. Typically, ISes require complex computations over large numbers, and are deployed on the users' electronic

Partially supported by funding from Polish National Science Centre (NCN) contract number DEC-2013/08/M/ST6/00928.

P. Farshim and E. Simion (Eds.): SecITC 2017, LNCS 10543, pp. 115–128, 2017.
https://doi.org/10.1007/978-3-319-69284-5_9

devices, which store sensitive secret keys. There are several common threats concerning this aspect, emerging from the fact that the end users see the devices as *black boxes*, and they have to trust that the scheme implementation processes are not tampered with. Very often, such devices are produced by vendors beyond of the end users' control, and as such are subject to malicious modification, which can bring about the following vulnerabilities:

- **Prover's Ephemeral Leakage**: Especially important for three round identification schemes, with three messages exchanged between a *Prover* and a *Verifier*:
  (1) the *Prover* sends a *commitment* to a random value to the *Verifier*;
  (2) the *Verifier* sends to the Prover another random value called a *challenge*;
  (3) a *response* message sent by the Prover is a result of a function of the challenge and the secret key masked by the committed ephemeral.
  At the Verifier's side, this *response* is checked by the means of the public key with the commitment and the challenge. If a malicious manufacturer implements a covert channel within a Prover's device, it can learn (or set) ephemeral values coined in the commitment phase, and unmask the secret key from the response. This way, the ephemeral leakage subsequently enables impersonation attacks using the Prover's identity. Note that Schnorr [2] and Okamoto [3] IS*es* are vulnerable to this attack. Recently, a remedy for that problem has been proposed in [1]. The solution is quite flexible and works for many similar three round constructions.
- **Verifier's Ephemeral Leakage**: Alternatively, if there is a back-door channel in a Verifier's device, it can be exploited by a malicious Prover to read ephemeral values coined by the Verifier before the challenge phase. There are IS*es* which rely on the secrecy of such values e.g. [4–6]. In all these schemes the Adversary knowing the Verifier's ephemeral value can impersonate the Prover without the secret key. It is worth to notice that typical three round identification schemes are immune, from their design, to attacks based on the Verifier's ephemeral leakage, since the only random value of the Verifier is the challenge revealed to the Prover in the second message. This statement, however, requires an assumption that the challenge value is coined strictly *after* the *commitment* phase, as otherwise impersonation would be trivial, due to simulatability property of the IS.
- **Losing Deniability**: Although typical three round IS*es* resist Verifier's ephemeral leakage attacks, they suffer from the deniability attacks mounted by the active malicious Verifier. Indeed, instead of coining the challenge at random, the Adversary can use a Fiat-Shamir transformation [7] and compute challenge as a hash value over the commitment, this way changing the scheme into an undeniable signature.

Problem Statement: In this paper we address the following issue: *How to design and implement a deniable* IS:

(1) *secure against ephemeral leakage on both Prover's and Verifier's side;*
(2) *withstanding attacks based on Fiat-Shamir transformation.*

1.1 Contribution of the Paper

The contribution of the paper is the following:

- We introduce a new strong security model for deniable identification schemes in which we allow *Adversaries*:
  - to set ephemerals on Provers' side in the Query Stage of the security experiment,
  - to read ephemerals used on Verifiers' side in the final (Impersonation) Stage of the security experiment.

  We define the IS to be secure if no Adversary, even given such a power and knowledge, is able to impersonate a Prover, without their secret key.
- We propose a general extension to three-rounds identification protocols, e.g. [1–3], hardening them against *Attacks on Deniability* by Fiat-Shamir transformation, secure in our stronger model.
- We show an example of our extension based on a modified Schnorr scheme, and prove its security in our model.

Our proposition is useful for systems based on three-round IS, where randomness leakage is possible. There is a growing demand for schemes secure in such scenarios, due to recent revelations regarding undermining cryptographic standards and implementations.

Remark: note that typical, 4-round *Malicious Verifier Zero Knowledge* schemes, that are based on commitments to challenge are not secure in the *Verifier Leakage* model. Coining challenge before Prover's commitment is sent may lead to straightforward impersonation: the challenge leakage allows for textbook simulation.

Previous Work. Identification schemes have been in use since the dawn of the modern, public-key cryptography [2,7–9]. Schnorr has introduced a DLP based construction [2], followed by [3] of Okamoto. Several ISes are specialized in terms of models or attack schemes, e.g. [10,11]. [12] introduced a notion of vulnerability to ephemeral leakage and proposed IS protocols invulnerable to such attacks. [13] shown IS secure against *Reset Attacks* based on stateless, deterministic signature schemes, CCA-secure asymmetric encryption schemes and pseudorandom functions with trapdoor commitments. *Subversion resilience* is a concept regarding security of various schemes in settings, where malicious manufacturer may replace original scheme with a modified one that behaves identically, but may leak additional information by hidden trapdoors in regular outputs [14–16].

The paper is organized in the following way. In Sect. 2 we review our strong security model, strongly based on models from [1]. In Sect. 3 we propose the extensions of generic three-rounds ISes following the *commit, challenge, response* schema, which protects against Fiat-Shamir transformation-based attacks on deniability. In Sect. 4 we modify the protocol from [1], and prove its security in our model.

2 System Model

Let us first recall the definition of IS from [1] loosely based on Okamoto's definition [3].

Definition 1 (Identification Scheme). *An identification scheme* IS *is a tuple of procedures* $(\mathsf{PG}, \mathsf{KG}_{\mathcal{P}}, \mathsf{KG}_{\mathcal{V}}, \mathcal{P}, \mathcal{V}, \pi)$*:*

$\mathsf{par} \leftarrow \mathsf{PG}(1^{\lambda})$**:** *takes the parameter* λ*, and outputs public parameters.*
$(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KG}_{\mathcal{P}}(\mathsf{par})$**:** *outputs secret and public keys of the prover.*
$(\mathsf{se}, \mathsf{pe}) \leftarrow \mathsf{KG}_{\mathcal{V}}(\mathsf{par})$**:** ***(optional)*** *outputs secret and public keys of the verifier.*
$\mathcal{P}(\mathsf{sk}, \mathsf{pe})$**:** *denotes the Prover algorithm which interacts with the Verifier* $\mathcal{V}$*.*
$\mathcal{V}(\mathsf{pk}, \mathsf{se})$**:** *denotes the Verifier algorithm which interacts with the Prover* $\mathcal{P}$*.*
$\pi(\mathcal{P}, \mathcal{V})$**:** *denotes the protocol of interactions between* $\mathcal{P}$ *and* $\mathcal{V}$*.*

IS *has Initialization and Operation Stages. In Initialization Stage, parameters and keys for users are generated. In the latter, a user proves interactively its identity in front of the Verifier:* $\pi(\mathcal{P}(\mathsf{sk}, \mathsf{pe}), \mathcal{V}(\mathsf{pk}, \mathsf{se}))$*. We write* $\pi(\mathcal{P}, \mathcal{V}) \rightarrow 1$ *if* $\mathcal{P}$ *and* $\mathcal{V}$ *have mutually accepted each other in* π*. The scheme is* complete *iff*

$$\Pr[(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KG}_{\mathcal{P}}(), (\mathsf{se}, \mathsf{pe}) \leftarrow \mathsf{KG}_{\mathcal{V}}(), \pi(\mathcal{P}(\mathsf{sk}, \mathsf{pe}), \mathcal{V}(\mathsf{pk}, \mathsf{se})) \rightarrow 1] = 1.$$

The optional, verifier key pair $(\mathsf{se}, \mathsf{pe})$ exists in several IS schemes. If the IS does not rely on it, or even explicitly denies its existence, we may assume $\mathsf{KG}_{\mathcal{V}}$ always returns $(\perp, \perp)$ on any input.

2.1 Impersonation Resilience

The fundamental security requirement for IS is that no malicious Prover algorithm $\mathcal{A}$, without the secret key sk corresponding to the public key pk used by the Verifier, should be accepted in protocol π. In other words, we require that probability $\Pr[\pi(\mathcal{A}(\mathsf{pk}, \mathsf{pe}), \mathcal{V}(\mathsf{pk}, \mathsf{se})) \rightarrow 1] \leq \epsilon_{\lambda}$ where ϵ_{λ} is a negligible function. We formally define our security model in Sect. 2.3.

2.2 Adversary Model

The process in which an Adversary gains knowledge about the attacked protocol is modeled by a *Query Stage* of the security experiment. This means that the Adversary runs a polynomial number ℓ of the protocol executions between the Prover and the Verifier: $\pi(\mathcal{P}(\mathsf{sk}, \mathsf{pe}), \mathcal{V}(\mathsf{pk}, \mathsf{se}))$. We consider the *Active Adversary* which actively participates in the stage, usually as a Verifier $\widetilde{\mathcal{V}}$, i.e. it actively chooses messages sent to the Prover. Based on [1], we assume the Adversary additionally adaptively sets the ephemeral values for the Prover in each protocol run in the *Query Stage*. Finally, extending the model from [1], we consider the Adversary that can read ephemeral values of the Verifier in the *Impersonation Stage*, immediately after those values are produced.

2.3 Security Experiments

Let $\bar{x}_i$ be adaptive ephemerals from a malicious Verifier $\widetilde{\mathcal{V}}$ injected to the Prover $\mathcal{P}^{\bar{x}_i}$ in the ith execution of the *Query Stage*. Let the view $v_i = \{T_1, \ldots, T_i\} \cup \{\bar{x}_1, \ldots, \bar{x}_i\}$ be the total knowledge $\mathcal{A}$ can gain after i runs of π, where T_i is the transcript of the protocol messages in the ith execution. The IS is CPLVE-secure if such a cumulated knowledge after ℓ executions does not help the Adversary to be accepted by the Verifier except with a negligible probability.

Definition 2 (Chosen Prover-Leaked Verifier Ephemeral – (CPLVE)). *Let* $\mathsf{IS} = (\mathsf{PG}, \mathsf{KG}_{\mathcal{P}}, \mathsf{KG}_{\mathcal{V}}, \mathcal{P}, \mathcal{V}, \pi)$. *We define security experiment* $\mathtt{Exp}_{\mathsf{IS}}^{\mathsf{CPLVE},\lambda,\ell}$*:*

`Init Stage`: $\mathsf{par} \leftarrow \mathsf{PG}(1^\lambda)$, $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KG}_{\mathcal{P}}(\mathsf{par})$, $(\mathsf{se}, \mathsf{pe}) \leftarrow \mathsf{KG}_{\mathcal{V}}(\mathsf{par})$. $\mathcal{A}$:$(\widetilde{\mathcal{P}}(\mathsf{pk}, \mathsf{pe}), \widetilde{\mathcal{V}}(\mathsf{pk}, \mathsf{pe}))$.

`Query Stage`: *For* $i = 1$ *to* ℓ *run* $\pi(\mathcal{P}^{\bar{x}_i}(\mathsf{sk}, \mathsf{pe}), \widetilde{\mathcal{V}}(\mathsf{pk}, \mathsf{pe}, \bar{x}_i, v_{i-1}))$, *where* $\bar{x}_i \in \{\bar{x}_1, \ldots, \bar{x}_\ell\}$ *are the adaptive ephemerals from* $\widetilde{\mathcal{V}}$ *injected to the Prover* $\mathcal{P}^{\bar{x}_i}$ *in the* i*th execution, and* v_{i-1} *is the total view of* $\mathcal{A}$ *until the* i*th execution.*

`Impersonation Stage`: $\mathcal{A}$ *executes the protocol* $\pi(\widetilde{\mathcal{P}}(\mathsf{pk}, \mathsf{pe}, v_\ell, \bar{\boldsymbol{e}}), \mathcal{V}(\mathsf{pk}, \mathsf{se}))$, *where* $\bar{\boldsymbol{e}}$ *are the ephemerals of the Verifier leaked to the malicious Prover* $\widetilde{\mathcal{P}}$.

The advantage of $\mathcal{A}$ *in the experiment* $\mathtt{Exp}_{\mathsf{IS}}^{\mathsf{CPLVE},\lambda,\ell}$ *is the probability of acceptance in the last stage:*

$$\mathbf{Adv}(\mathcal{A}, \mathtt{Exp}_{\mathsf{IS}}^{\mathsf{CPLVE},\lambda,\ell}) = \Pr[\pi(\widetilde{\mathcal{P}}(\mathsf{pk}, \mathsf{pe}, v_\ell, \bar{\boldsymbol{e}}), \mathcal{V}(\mathsf{pk}, \mathsf{se})) \rightarrow 1].$$

We say that the IS *is* (λ, ℓ)-CPLVE*–secure if* $\mathbf{Adv}(\mathcal{A}, \mathtt{Exp}_{\mathsf{IS}}^{\mathsf{CPLVE},\lambda,\ell}) \leq \epsilon_\lambda$ *and* ϵ_λ *is negligible in* λ.

We utilize the definition of deniability from [17], which itself generalizes the idea from [18]. Let π be a protocol in IS. We assume an adversary $\mathcal{M}$ which inputs an arbitrary number of public keys $\mathbf{pk} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_\ell)$, randomly coined with an appropriate key generating algorithm. The adversary initiates an arbitrary number of protocols with the honest parties, some in a role of the prover, others in a role of the verifier. The view of $\mathcal{M}$ consists of its internal randomness, and the transcript of the entire interaction, in all the protocols in which $\mathcal{M}$ participated. We denote this view as $\mathsf{View}_{\mathcal{M}}(\mathbf{pk}, a)$.

Definition 3. *We say that* π *is a strongly deniable protocol of* IS *with respect to the class* A *of auxiliary inputs if for any adversary* $\mathcal{M}$*, for any input of public keys* $\mathbf{pk} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_\ell)$ *and any auxiliary input* $a \in A$*, there exists a simulator* $SIM_{\mathcal{M}}$ *that, running on the same inputs as* $\mathcal{M}$*, produces a simulated view which is indistinguishable from the real view of* $\mathcal{M}$*. That is, consider the following two probability distributions, where* $\mathbf{pk} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_\ell)$ *is the set of public keys of the honest parties:*

$$\mathcal{R}eal(\lambda, a) = [(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow \mathsf{KG}(1^\lambda); (a, \mathbf{pk}, \mathsf{View}_{\mathcal{M}}(\mathbf{pk}, a)]$$
$$\mathcal{S}im(\lambda, a) = [(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow \mathsf{KG}(1^\lambda); (a, \mathbf{pk}, SIM_{\mathcal{M}}(\mathbf{pk}, a)]$$

then for all probabilistic poly-time machines Dist *and all* $a \in A$*, there exists a function* ϵ_λ *negligible in* λ *s.t.:*

$$|\Pr_{x \in \mathcal{R}eal(\lambda,a)}[\mathsf{Dist}(x) = 1]| - |\Pr_{x \in \mathcal{S}im(\lambda,a)}[\mathsf{Dist}(x) = 1]| \leq \epsilon_\lambda.$$

The idea behind this definition is that no adversary can follow a strategy that is not simulatable, i.e. there exist a distinguisher differentiating between the real adversary and a simulator. In other words, all adversarial strategies are simulatable.

2.4 Deniability Attack in Active Mode

Let $T = (X, c, S)$ denote the transcript of a 3-round IS. In Fig. 1 we recall how active Verifier can use the Fiat-Shamir transformation to generate undeniable transcript of the protocol, effectively transforming the 3-round interactive IS into non-interactive signature scheme. The value r is a randomizing factor. In real signature schemes, the value r is replaced by message m. The hash input $i = (X, r)$ is an undeniable proof that the party $\mathcal{P}$ has participated in the protocol.

Let $\mathsf{IS} = (\mathsf{PG}, \mathsf{KG}_\mathcal{P}, \mathcal{P}, \mathcal{V}, \pi)$ be a secure 3-round identification scheme, where (X, c, S) denotes commitment, challenge, and response messages. Let $(\mathsf{sk}, \mathsf{pk})$ denote a pair of secret/public keys. Let $\mathcal{H}$ denote a secure collision resistant one way hash function into the challenge space. The *deniability breaking attack* on protocol $\pi(\mathcal{P}(\mathsf{sk}), \mathcal{V}(\mathsf{pk}))$:

1. $\mathcal{P}$: prepare X, $\quad \mathcal{P} \overset{X}{\rightsquigarrow} \mathcal{V}$.
2. $\mathcal{V} : r \leftarrow_R \{0,1\}^n$, $c = \mathcal{H}(X, r)$, $\quad \mathcal{P} \overset{c}{\leftsquigarrow} \mathcal{V}$.
3. $\mathcal{P}$: prepare S, $\quad \mathcal{P} \overset{S}{\rightsquigarrow} \mathcal{V}$.
4. $\mathcal{V}$: verify in a regular way.

Fig. 1. The attack on deniability of typical 3-round IS.

3 Extended Identification Schemes

3.1 General Idea – Commitment to an Unknown Value

The general idea behind the proposed extensions is that in order to achieve the strong deniability property in the *Verifier Ephemeral Leakage* scenario, the Verifier has to prove that the challenge has not been produced via the transformation of the Prover's commitment X. Therefore, at the beginning of the protocol, the Verifier itself randomly chooses a commitment to an unknown challenge, which can be opened by them only after they obtain the first message from the Prover. We propose two different methods for this purpose: (a) *Deterministic Encryption Method*; (b) *Proof of Computation Method*; which can be used separately or together.

3.2 Deterministic Encryption Method

This extension is based on the assumption that the scheme in subject can be used in conjunction with a deterministic asymmetric encryption, for which, w.l.o.g., we use the following definition.

Definition 4 (Asymmetric Encryption Scheme). *Let* $\mathsf{E} = (\mathsf{KG}_\mathsf{E}, \mathcal{E}, \mathcal{D})$ *denote a secure deterministic encryption scheme, s.t.* $(\mathsf{se}, \mathsf{pe}) \leftarrow \mathsf{KG}_\mathsf{E}()$*:*

(1) $\forall_{(m \in M)} : \mathcal{E}(\mathsf{pe}, m) \rightarrow c \in C, s.t. \mathcal{D}(\mathsf{se}, c) \rightarrow m,$
(2) $\forall_{(c \in C)} : \mathcal{D}(\mathsf{se}, c) \rightarrow m \in M, s.t. \mathcal{E}(\mathsf{pe}, m) \rightarrow c$

where $(\mathsf{se}, \mathsf{pe})$ *is a secret/public key pair;* M, C *are plaintext, and ciphertext spaces;* $(\mathsf{KG}_\mathsf{E}, \mathcal{E}, \mathcal{D})$ *are key generation, encryption and decryption algorithms.*

The only security property of E that is required in the proposed scheme is its *secrecy* or *one-wayness*, that is:

Definition 5 (Encryption One-Wayness). *An Asymmetric Encryption Scheme* E *has encryption one-wayness property, if for any PPT algorithm* $\mathcal{A}$*, for* $(\mathsf{se}, \mathsf{pe}) \leftarrow \mathsf{KG}_\mathsf{E}(1^\lambda)$ *and for a* $c \in C$ *selected uniformly at random:*

$$\Pr[\mathcal{A}(\mathsf{pe}, c) = \mathcal{D}(\mathsf{se}, c)] \leq \epsilon_\lambda$$

for a negligible function ϵ_λ*.*

Note that the equation is actually equivalent to $\Pr[\mathcal{E}(\mathsf{pe}, \mathcal{A}(\mathsf{pe}, c)) = c] \leq \epsilon_\lambda$ and to $\Pr[\mathcal{A}(\mathsf{pe}, \mathcal{E}(\mathsf{pe}, m)) = m] \leq \epsilon_\lambda$ for uniformly selected message $m \in M$.

An example of such a scheme is a textbook RSA Encryption [19]. With the E scheme, as of Definition 4, the extension is the following: at the beginning the Verifier chooses the ciphertext $\hat{c}$ randomly, which is immediately sent to the Prover. This is a commitment to a yet unknown challenge c, and corresponds to the Verifier's ephemeral value, known to the malicious Prover in the *Verifier Ephemeral Leakage* model. Then, the Verifier waits until it gets a commitment from the Prover and only then opens the commitment $m = \mathcal{D}(\mathsf{se}, \hat{c})$, chooses a random bit $b \leftarrow_R \{0, 1\}$ and sends m, b to the Prover. The bit b allows for randomization of c, but the information size of b is insufficient to indicate the Prover's identity, as both options are equally simulatable. Both parties compute the commitment with a secure one way hash function $c = \mathcal{H}(m, b)$. This reflects the situation in which both the Prover and the Verifier learn the commitment c only after X has been received by the Verifier. On the other hand, the Prover checks if the value m agrees with the commitment $\mathcal{E}(\mathsf{pe}, m) \stackrel{?}{=} \hat{c}$, and then it is convinced that the challenge m has not been produced by a Fiat-Shamir-like transformation over its own commitment X. If $\mathcal{E}(\mathsf{pe}, m) \neq \hat{c}$, the Prover stops the protocol.

The proposed extension is depicted in Fig. 2. Note that the IS has a slightly different interface as $\mathcal{P}$ and $\mathcal{V}$ take each others' public keys and their own secret keys on input (contradictory to the Definition 1 where only Prover's keys were considered). The single random bit b has a very small influence on the protocol

Let $\mathsf{IS} = (\mathsf{PG}, \mathsf{KG}_{\mathcal{P}}, \mathcal{P}, \mathcal{V}, \pi)$ be a secure 3-round identification scheme, where (X, c, S) denotes commitment, challenge, and response messages. Let $\mathcal{H}$ denote a secure collision resistant one way hash function into the challenge space. Let $(\mathsf{KG}_{\mathsf{E}}, \mathcal{E}, \mathcal{D})$ denote a secure encryption scheme. We add $\mathsf{KG}_{\mathcal{V}} = \mathsf{KG}_{\mathsf{E}}$ and modify the protocol $\pi(\mathcal{P}(\mathsf{sk}, \mathsf{pe}), \mathcal{V}(\mathsf{pk}, \mathsf{se}))$ in the following way:

0. $\mathcal{V}$: commits to unknown challenge $\hat{c} \in_R C$, $\mathcal{P} \xleftarrow{\hat{c}} \mathcal{V}$.
1. $\mathcal{P}$: prepares X, $\mathcal{P} \xrightarrow{X} \mathcal{V}$.
2. $\mathcal{V}$: waits for X, $m = \mathcal{D}(\mathsf{se}, \hat{c})$, $b \leftarrow_R \{0, 1\}$, $\mathcal{P} \xleftarrow{m,b} \mathcal{V}$.
3. $\mathcal{P}$: if $(\mathcal{E}(\mathsf{pe}, m) \stackrel{?}{=} \hat{c})$: $c = \mathcal{H}(m, b)$, prepares S, $\mathcal{P} \xrightarrow{S} \mathcal{V}$.
4. $\mathcal{V}$: $c = \mathcal{H}(m, b)$, verifies (X, c, S) in a regular way.

Fig. 2. Extension based on encryption scheme.

itself, but is crucial in proving the security of the underlying IS, when the proof uses *rewinding techniques*, in order to produce two distinct challenges for the same initial commitment.

Lemma 1. *The extension proposed in Fig. 2 protects against deniability attacks on 3-round* IS *via Fiat-Shamir transformation - as of Fig. 1.*

Proof. The proof is by contradiction. Assume that a malicious Verifier successfully, with non-negligible probability, mounts the attack resulting with transcript $T = (\hat{c}, X, m, b, S)$ and the proof $i = (X, r)$, s.t: $m = \mathcal{D}(\mathsf{se}, \hat{c})$, $c = \mathcal{H}(m, b)$ and $c = \mathcal{H}'(X, r)$ for any hash function $\mathcal{H}'$, then we successfully find a collision for the hash function $\mathcal{H}$ with inputs $i = (m, b)$ and $i = (X, r)$ (if $\mathcal{H} = \mathcal{H}'$), or break preimage resistance of either $\mathcal{H}$ (with the image being $c = \mathcal{H}'(X, r)$) or $\mathcal{H}'$ (with the image being $c = \mathcal{H}(m, b)$). □

Lemma 2. *The extension proposed in Fig. 2 retains zero-knowledge properties of the underlying* IS*.*

Proof (Sketch).

Completeness. Straightforward verification shows that if the original IS was complete, the modified scheme is complete as well. The addition of $\hat{c}$ and the way c is computed does not influence the protocol if only $\mathcal{H}$ is a secure hash function indistinguishable from a Random Oracle into the challenge space.

Soundness. The method of proving soundness of the modified scheme is closely related to the method used to prove the soundness of IS. In principle, $\mathcal{P}$ cannot derive any knowledge from the commitment scheme except with a negligible probability. If $\mathcal{P}$ could derive any information about the challenge message before the *commitment* phase, they would be able to break the *encryption one-wayness* of E (cf. Definition 5).

Zero-knowledge. The protocol is simulatable if only IS is simulatable. Let us choose $m \in M$ and $b \in \{0,1\}$ at random. Compute $c = \mathcal{H}(m,b)$ and simulate transcript (X,c,S) of IS for the given challenge c. Compute commitment $\hat{c} = \mathcal{E}(\mathsf{pe},m)$. Return $(\hat{c},X,(m,b),S)$ as the simulated transcript. □

3.3 Proof of Computation Method

This extension is based on the assumption that the Verifier's computing device $\mathsf{D}_{\mathcal{V}}$ is faster than the Prover's computing device $\mathsf{D}_{\mathcal{P}}$. Let $\mathsf{RT}_{\mathsf{D}}(A)$ denote a running time of the device D executing an algorithm A. Let (P,X) denote a computational problem in domain X, and ς denote its solution. Let $\mathsf{Ver}(\mathsf{P},X,\varsigma)$ denote a fast verification algorithm which returns 1 if ς is a solution for (P,X) or returns 0 otherwise. Let $\mathcal{S}(\mathsf{P},X)$ denote the algorithm solving (P,X). We assume that $\mathcal{S}(\mathsf{P},X)$ is "quite" complex, that is, on any device D it holds that: $\mathsf{RT}_{\mathsf{D}}(\varsigma = \mathcal{S}(\mathsf{P},X)) \gg \mathsf{RT}_{\mathsf{D}}(\mathsf{Ver}(\mathsf{P},X,\varsigma))$. To capture that the Verifier's computing device $\mathsf{D}_{\mathcal{V}}$ is faster than the Prover's computing device $\mathsf{D}_{\mathcal{P}}$ we assume that: $\mathsf{RT}_{\mathsf{D}_{\mathcal{V}}}(\mathcal{S}(\mathsf{P},X)) < \mathsf{RT}_{\mathsf{D}_{\mathcal{P}}}(\mathcal{S}(\mathsf{P},X))$, for any $(\mathsf{P},X,\mathcal{S})$.

Let $\mathcal{G}(\mathsf{P},w)$ be a domain generation algorithm for problem P that takes a seed $w \in Seed$ as an input, and outputs a domain X for P. Let $\mathcal{H} : \{0,1\}^* \rightarrow Seed$ be a one way function used to compute a seed w for $\mathcal{G}(P,w)$. Assume the following process of generating a sequence of problems P, X_i and its solutions ς_i from the random seed $w \in_R Seed$.

$\mathsf{Gen}(\mathsf{P},w)$:
`Init Stage`: $n = 0$, $X_0 = \mathcal{G}(\mathsf{P},w)$, $\varsigma_0 = \mathcal{S}(\mathsf{P},X_0)$
`Iterate` since `Start` signal until `Stop` signal:
$n = n+1$, $w_n = \mathcal{H}(\varsigma_{n-1})$, $X_n = \mathcal{G}(\mathsf{P},w_n)$, $\varsigma_n = \mathcal{S}(\mathsf{P},X_n)$,
`Return`: $\langle \varsigma_i \rangle_i^n$

Assume the verification process:

$\mathsf{Check}(\mathsf{P},w,\langle \varsigma_i \rangle_i^n)$:
`Init Stage`: $n = 0$, $X_0 = \mathcal{G}(\mathsf{P},w)$, $v_0 = \mathsf{Ver}(\mathsf{P},X_0,\varsigma_0)$
`Iterate` for all $i \in \{1 \ldots n\}$: $w_i = \mathcal{H}(\varsigma_{i-1})$, $X_i = \mathcal{G}(\mathsf{P},w_i)$, $v_i = \mathsf{Ver}(\mathsf{P},X_i,\varsigma_i)$
`Return`: $\prod_{i=0}^{n} v_i$

The Proof of Computation System PCS is a tuple of the above defined algorithms: $(\mathcal{G},\mathsf{P},\mathcal{S},\mathsf{Ver},\mathsf{Gen},\mathsf{Check},\mathcal{H})$. The proposed extension is depicted in Fig. 3

Lemma 3 *The extension proposed in Fig. 3 protects against deniability attacks on 3-round* IS *via Fiat-Shamir transformation - as of Fig. 1.*

Proof The proof is by contradiction. Similarly as in the proof of Lemma 1, if a malicious Verifier successfully, with non-negligible probability, attacks the

Let $\mathsf{IS} = (\mathsf{PG}, \mathsf{KG}_{\mathcal{P}}, \mathcal{P}, \mathcal{V}, \pi)$ be a secure 3-round identification scheme, where (X, c, S) denote commitment, challenge, and response messages. Let $\mathcal{H}$ denote a secure one way hash function into the challenge space. Let $\mathsf{PCS} = (\mathcal{G}, \mathsf{P}, \mathcal{S}, \mathsf{Ver}, \mathsf{Gen}, \mathsf{Check})$ denote a proof of computation system. We modify the protocol $\pi(\mathcal{P}(\mathsf{sk}), \mathcal{V}(\mathsf{pk}))$ in the following way:

0. $\mathcal{V} : w \in_R Seed$, `Start` iterating $\mathsf{Gen}(\mathsf{P}, w)$ — $\mathcal{P} \xleftarrow{w} \mathcal{V}$.
1. $\mathcal{P}$: prepares X, — $\mathcal{P} \xrightarrow{X} \mathcal{V}$.
2. $\mathcal{V}$: waits for X, $\langle \varsigma_i \rangle_i^n$ = `Stop` iterating Gen, — $\mathcal{P} \xleftarrow{\langle \varsigma_i \rangle_i^n} \mathcal{V}$.
3. $\mathcal{P}$: if $(\mathsf{Check}(\mathsf{P}, w, \langle \varsigma_i \rangle_i^n) \stackrel{?}{=} 1)$: $c = \mathcal{H}(\langle \varsigma_i \rangle_i^n)$, prepares S, — $\mathcal{P} \xrightarrow{S} \mathcal{V}$.
4. $\mathcal{V}$: verifies (X, c, S) in a regular way.

Fig. 3. Extension based on Proof of Computation System.

scheme getting the transcript $T = (w, X, \langle \varsigma_i \rangle_i^n, S)$ and the Fiat-Shamir undeniability proof $i = (X, m)$, s.t: $\langle \varsigma_i \rangle_i^n = \mathsf{Gen}(\mathsf{P}, w)$, $c = \mathcal{H}(\langle \varsigma_i \rangle_i^n)$, and $c = \mathcal{H}(X, m)$, then we successfully find a collision for the hash function $\mathcal{H}$ with inputs $i = (\langle \varsigma_i \rangle_i^n)$ and $i = (X, r)$ (if $\mathcal{H} = \mathcal{H}'$), or break preimage resistance of either $\mathcal{H}$ (with the image being $c = \mathcal{H}'(X, r)$) or $\mathcal{H}'$ (with the image being $c = \mathcal{H}(\langle \varsigma_i \rangle_i^n)$). □

4 Specific Scheme Proposition

To show the applicability of our propositions we introduce the modification of the scheme from [1] augmented with our first extension, using textbook RSA encryption. The proposed scheme is depicted in Fig. 4.

4.1 Simulation in the *Passive Adversary* Mode

The modified Schnorr IS preserves the simulatability property of its original version. The protocol transcript can be efficiently simulated by the following algorithm (for any public keys $(\mathsf{pk}, \mathsf{pe})$ and challenge message (m, b)):

$\mathtt{Sim}\ \mathcal{S}_{\mathsf{IS}}^{\mathsf{PA}}((\mathsf{pk}, \mathsf{pe} = (e, N)), (m, b))$:
$\hat{c} = m^e$, $c = \mathcal{H}_q(m, b)$, $s \leftarrow_R \mathbb{Z}_q^*$,
$X := (g^s / \mathsf{pk}^c)$, $\hat{g} := \mathcal{H}_G(X, c)$, $S := \hat{g}^s$
`return:`
$T = (\hat{c}, X, (m, b), S)$

Observe that for this transcript the verification holds: $\hat{e}(S, g) = \hat{e}(\mathcal{H}_G(X, c), X\mathsf{pk}^c)$. The simulator can play the simulated transcript $T = (\hat{c}, X, m, S)$ in the correct order, thus mimicking the real interaction between the parties. The real transcript and the simulated tuple are identically distributed.

$\mathsf{par} \leftarrow \mathsf{PG}(1^\lambda)$**:** Let $G, G_T \leftarrow \mathcal{G}(1^\lambda)$ be cyclic groups of a prime order q, and $g \in G$ be a generator. Let $\mathcal{H}_q : \{0,1\}^* \to \mathbb{Z}_q^*$ and $\mathcal{H}_G : \{0,1\}^* \to G$ be hash functions. Let $\hat{e} : G \times G \to G_T$ be a bilinear map. We assume that GDH holds in G, and $\hat{e}$ is $\mathcal{O}_{\mathsf{DDH}}$ oracle. Set $\mathsf{par} = (q, G, g, G_T, \mathcal{H}_q, \mathcal{H}_G, \hat{e})$.

$\mathsf{KG}_{\mathcal{P}}(\mathsf{par})$**:** $\mathsf{sk} \leftarrow_R \mathbb{Z}_q^*$, $\mathsf{pk} = g^{\mathsf{sk}}$. Output $(\mathsf{sk}, \mathsf{pk})$.

$\mathsf{KG}_{\mathcal{V}}(\mathsf{par})$**:** Randomly select large primes a and b s.t. $N = ab$ satisfies λ bits of security; Select relatively small value $e \approx 2^{16}$ co-prime to the order of group $\mathbb{Z}_N^\times$ and compute its inverse d modulo the order. Output $(\mathsf{se} = d, \mathsf{pe} = (N, e))$.

$\pi(\mathcal{P}(\mathsf{sk}, \mathsf{pe}), \mathcal{V}(\mathsf{pk}, \mathsf{se}))$ is the following protocol:

0. $\mathcal{V}$: commits to an unknown challenge $\hat{c} \in_R \mathbb{Z}_N^\times$, $\mathcal{P} \overset{\hat{c}}{\leftsquigarrow} \mathcal{V}$.
1. $\mathcal{P}$: selects $x \in_R \mathbb{Z}_q^*$, $X = g^x$, $\mathcal{P} \overset{X}{\rightsquigarrow} \mathcal{V}$.
2. $\mathcal{V}$: awaits X, $m = \hat{c}^d$, $b \leftarrow_R \{0,1\}$, $\mathcal{P} \overset{m,b}{\leftsquigarrow} \mathcal{V}$.
3. $\mathcal{P}$: aborts if $m^e \neq \hat{c}$, otherwise:
   $c = \mathcal{H}_q(m, b)$, $\hat{g} = \mathcal{H}_G(X, c)$, $S = \hat{g}^{x + \mathsf{sk} \cdot c}$, $\mathcal{P} \overset{S}{\rightsquigarrow} \mathcal{V}$.
4. $\mathcal{V}$: $c = \mathcal{H}_q(m, b)$, $\hat{g} = \mathcal{H}_G(X, c)$,
   accepts iff $\hat{e}(S, g) = \hat{e}(\hat{g}, X\mathsf{pk}^c)$.

Fig. 4. The proposed modified IS.

4.2 Security Analysis

In our analysis we assume that there is an effective Adversary that breaks our scheme from Fig. 4. In the Query Stage, we interact with the Adversary, simulating the proofs without the secret key, but using the injected ephemerals. In the Impersonation Stage, there are two mutually exclusive possibilities: either the Adversary knows the challenge $c = \mathcal{H}_q(m, b)$ before sending X, or he does not. Therefore, in our reduction proof, we guess in which alternative the Adversary exists. If it knows the value $c = \mathcal{H}_q(m, b)$, we use it to break underlying security of RSA. If the Adversary attacks without the knowledge of the challenge $c = \mathcal{H}_q(m, b)$ we proceed as in the original proof from [1]. In the latter case, we follow the methodology from [2,3], using *rewinding technique*. Namely, we fix randomness $\hat{c}, X$, but change the bit b by setting it to 0 for the first run, and to 1 for the second run. This results with two tuples $(\hat{c}, X, m, 0, S_1)$, $(\hat{c}, X, m, 1, S_2)$ letting us solve the underlying hard problem – in this case CDH.

Theorem 4. *Let* IS *denote the modified identification scheme (as in Fig. 4).* IS *is secure (in the sense of Definition 2), i.e. the advantage* $\mathbf{Adv}(\mathcal{A}, \mathsf{Exp}^{\mathsf{CPLVE},\lambda,\ell})$ *is negligible in* λ*, for any PPT algorithm* $\mathcal{A}$*.*

We postpone the proof to the Appendix A.

5 Conclusion

In this paper, we have shown how to modify a wide class of three-move identification schemes secure against *Prover Ephemeral Injection* into identification

schemes secure against *Verifier Ephemeral Leakage* and *Deniability Attack*. We have shown an example based on a modified Schnorr IS from [1]. We have formalized a security model and proved the security of our constructions.

A Postponed Proof

Proof. We use ROM for hash queries. The proof is by contradiction. Suppose there is an Adversary $\mathcal{A} = (\widetilde{\mathcal{P}}, \widetilde{\mathcal{V}})$ for which $\mathbf{Adv}(\mathcal{A}, \mathtt{Exp}^{\mathsf{CPLVE},\lambda,\ell})$ is non-negligible. Thus, it can be used as a subprocedure: either to break security of RSA by taking eth root in $\mathbb{Z}_N^\times$ of a given challenge ciphertext $\widetilde{c}$, or to break GDH for the given instance g, g^α, g^β, by computing $g^{\alpha\beta}$, either with a non-negligible probability. Therefore we draw a bit d which determines our strategy. If $d = 0$, we assume a play against the Adversary in the first scenario, breaking the security of RSA; otherwise, we play against the Adversary in the second scenario, solving the CDH problem.

`Init Stage`: Let $\mathsf{par} \leftarrow \mathbb{G} = (q, g, G)$ and (g, g^α, g^β) be the CDH problem input instance. We set $\mathsf{pk} = g^\alpha$ and give it to $\mathcal{A}$. If $d = 0$, we assume pe and $\widetilde{c}$ to be the RSA input instance, thus we do not know the proper verifier's secret key; otherwise, we honestly generate verifier secret keys $(\mathsf{pe}, \mathsf{se})$. We initiate RO table with columns I, H, r.

`Query Stage`: We interactively simulate, with an active malicious Verifier $\widetilde{\mathcal{V}}$, the protocol $\pi(\mathcal{P}^{\bar{x}_i}(\mathsf{pk}), \widetilde{\mathcal{V}}^{\mathcal{O}_{\mathcal{H}_G}}(\mathsf{pk}, \bar{x}_i, \{v_{i-1}\}))$, without the secret key, using injected ephemerals $\bar{x}_i$, ℓ times.

Serving Hash queries $\mathcal{O}_{\mathcal{H}_G}(I_i)$: If input I_i is in the RO table, the oracle returns the corresponding output H_i. Otherwise, $r_i \leftarrow_R \mathbb{Z}_q^*$, $H_i = g^{r_i}$, add (I_i, H_i, r_i) to the RO table.

(1) **Commitment $\hat{c}$:** Receive the commitment $\hat{c}$ in the first message.
(2) **Commitment X:** Send $\widetilde{X} = g^{\bar{x}}$ to the Verifier $\widetilde{\mathcal{V}}$.
(3) **Proof S:** Upon obtaining m, b from the Verifier, check $m^e \stackrel{?}{=} \hat{c}$ and compute $\bar{c} = \mathcal{H}_q(m, b)$. Query $\mathcal{O}_{\mathcal{H}_G}(\widetilde{X}, \bar{c})$ for r. Set $\widetilde{S} = \widetilde{X}^r \mathsf{pk}^{r\bar{c}} = \hat{g}^{\bar{x}+\mathsf{sk}\bar{c}}$. Note that: $\hat{e}(\widetilde{S}, g) = \hat{e}(\hat{g}, \widetilde{X}\mathsf{pk}^{\bar{c}})$. The *simulated* transcript tuple $\widetilde{T} = (\hat{c}, \widetilde{X}, (m, b), \widetilde{S})$ and the potential *real* protocol execution transcript $T = (\hat{c}, X, (m, b), S)$ are of the same distribution.

`Impersonation Stage`: The strategy differs between the scenarios:

$d = 0$ We send the challenge ciphertext $\widetilde{c}$ as Verifier's commitment. If the Adversary computes the challenge $c = \mathcal{H}_q(m, b)$ before sending X, we use him to break the security of the underlying encryption scheme. Intercepting query $\mathcal{O}_{\mathcal{H}_q}(m, b)$, we obtain m breaking the *encryption one-wayness*, in this case, being the eth root of $\widetilde{c}$ in $\mathbb{Z}_N^\times$, as $m^e = \widetilde{c}$.

$d = 1$ In ROM, we run $\pi(\widetilde{\mathcal{P}}^{\mathcal{O}_{\mathcal{H}_G}}(\mathsf{pk}, \mathsf{pe}, \{v_i\}), \mathcal{V}(\mathsf{pk}, \mathsf{se}))$ playing the role of the honest Verifier. We use the *rewinding technique*: we fix the random value x used for $X = g^x$ by $\widetilde{\mathcal{P}}$, and upon obtaining a correct proof message, we rewind the prover back to the challenge phase, choosing $b = 0$ in the first run and $b = 1$ in the second run. This gives us $c_1 = \mathcal{H}_q(m, 0)$ for the first run and $c_2 = \mathcal{H}_q(m, 1)$ for the second run. Finally, we get tuples $(\hat{c}, X, m, 0, c_1, S_1, \hat{g}_1, r_1)$ and $(\hat{c}, X, m, 1, c_2, S_2, \hat{g}_2, r_2)$. By inspecting RO tables, we obtain $\hat{g}_1 = \mathcal{O}_{\mathcal{H}_G}(X, c_1) \to g^{\beta r_1}$, $\hat{g}_2 = \mathcal{O}_{\mathcal{H}_G}(X, c_2) \to g^{\beta r_2}$. If we accept the Prover both times, i.e.: $\hat{e}(S_1, g) = \hat{e}(\hat{g}_1, X\mathsf{pk}^{c_1})$ and $\hat{e}(S_2, g) = \hat{e}(\hat{g}_2, X\mathsf{pk}^{c_2})$. Hence we conclude: $S_1 = g^{\beta r_1(x+\alpha c_1)}$ and $S_2 = g^{\beta r_2(x+\alpha c_2)}$. Thus ${S_1}^{1/r_1}/{S_2}^{1/r_2} = g^{\beta(\alpha c_1 - \alpha c_2)}$ and $g^{\alpha\beta} = ({S_1}^{1/r_1}/{S_2}^{1/r_2})^{1/(c_1-c_2)}$.

Now, let p denote the non-negligible probability of $\mathcal{A}$ breaking our scheme. Let p_0 be the probability that it knows $c = \mathcal{H}_q(m, b)$ before sending X. Let $p_1 = 1 - p_0$ be the probability that it doesn't know $c = \mathcal{H}_q(m, b)$ before sending X. Thus, we break RSA with probability $\frac{1}{2}pp_0$, or alternatively, we break CDH with probability $\frac{1}{2}p(1 - p_0)$. Hence, we break one of the problems with non negligible probability, which contradicts our assumptions for any probability value $p_0 \in [0, 1]$. □

References

1. Krzywiecki, Ł.: Schnorr-like identification scheme resistant to malicious subliminal setting of ephemeral secret. In: Bica, I., Reyhanitabar, R. (eds.) SECITC 2016. LNCS, vol. 10006, pp. 137–148. Springer, Cham (2016). doi:10.1007/978-3-319-47238-6_10
2. Schnorr, C.P.: Efficient signature generation by smart cards. J. Cryptol. **4**(3), 161–174 (1991). http://dx.doi.org/10.1007/BF00196725
3. Okamoto, T.: Provably secure and practical identification schemes and corresponding signature schemes. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 31–53. Springer, Heidelberg (1993). doi:10.1007/3-540-48071-4_3
4. Stinson, D.R., Wu, J.: An efficient and secure two-flow zero-knowledge identification protocol. J. Math. Cryptol. (JMC) **1**(3), 201–220 (2007)
5. Wu, J., Stinson, D.R.: An efficient identification protocol and the knowledge-of-exponent assumption. IACR Cryptology ePrint Archive 2007, 479 (2007)
6. Bender, J., Dagdelen, Ö., Fischlin, M., Kügler, D.: The PACE—AA protocol for machine readable travel documents, and its security. In: Keromytis, A.D. (ed.) FC 2012. LNCS, vol. 7397, pp. 344–358. Springer, Heidelberg (2012). doi:10.1007/978-3-642-32946-3_25
7. Fiat, A., Shamir, A.: How to prove yourself: practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987). doi:10.1007/3-540-47721-7_12
8. Feige, U., Fiat, A., Shamir, A.: Zero-knowledge proofs of identity. J. Cryptol. **1**(2), 77–94 (1988). http://dx.doi.org/10.1007/BF02351717
9. Guillou, L.C., Quisquater, J.-J.: A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. In: Barstow, D., et al. (eds.) EUROCRYPT 1988. LNCS, vol. 330, pp. 123–128. Springer, Heidelberg (1988). doi:10.1007/3-540-45961-8_11

10. Kurosawa, K., Heng, S.-H.: Identity-based identification without random oracles. In: Gervasi, O., Gavrilova, M.L., Kumar, V., Laganà, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) ICCSA 2005. LNCS, vol. 3481, pp. 603–613. Springer, Heidelberg (2005). doi:10.1007/11424826_64
11. Kurosawa, K., Heng, S.-H.: The power of identification schemes. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 364–377. Springer, Heidelberg (2006). doi:10.1007/11745853_24
12. Canetti, R., Goldreich, O., Goldwasser, S., Micali, S.: Resettable zero-knowledge (extended abstract). In: Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing (STOC 2000), pp. 235–244 (2000). http://doi.acm.org/10.1145/335305.335334
13. Bellare, M., Fischlin, M., Goldwasser, S., Micali, S.: Identification protocols secure against reset attacks. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 495–511. Springer, Heidelberg (2001). doi:10.1007/3-540-44987-6_30
14. Ateniese, G., Magri, B., Venturi, D.: Subversion-resilient signature schemes. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, 12–6 October 2015, pp. 364–375 (2015)
15. Russell, A., Tang, Q., Yung, M., Zhou, H.: Cliptography: clipping the power of kleptographic attacks. IACR Cryptology ePrint Archive 2015, 695 (2015). http://eprint.iacr.org/2015/695
16. Hanzlik, L., Kluczniak, K., Kutyłowski, M.: Controlled randomness – a defense against backdoors in cryptographic devices. In: Phan, R.C.-W., Yung, M. (eds.) Mycrypt 2016. LNCS, vol. 10311, pp. 215–232. Springer, Cham (2017). doi:10.1007/978-3-319-61273-7_11
17. Raimondo, M.D., Gennaro, R., Krawczyk, H.: Deniable authentication and key exchange. In: Juels, A., Wright, R.N., di Vimercati, S.D.C. (eds.) Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS 2006), Alexandria, 30 October–3 November 2006, pp. 400–409. ACM (2006). http://doi.acm.org/10.1145/1180405.1180454
18. Dwork, C., Naor, M., Sahai, A.: Concurrent zero-knowledge. In: Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing (STOC 1998), pp. 409–418 (1998). http://doi.acm.org/10.1145/276698.276853
19. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (1978). http://dx.doi.org/10.1145/359340.359342