Anonymous Deniable Identification in Ephemeral Setup & Leakage Scenarios

Łukasz Krzywiecki, Mirosław Kutyłowski, Jakub Pezda, Marcin Słowik

Department of Computer Science
Faculty of Fundamental Problems of Technology
Wrocław University of Science and Technology
lukasz.krzywiecki@pwr.wroc.pl

Abstract. In this paper we concern anonymous identification, where the verifier can check that the user belongs to a given group of users (just like in case of ring signatures), however a transcript of a session executed between a user and a verifier is deniable. That is, neither the verifier nor the prover can convice a third party that a given user has been involved in a session but also he cannot prove that any user has been interacting with the verifier. Thereby one can achieve high standards for protecting personal data according to the General Data Protection Regulation – the fact that an interaction took place might be a sensitive data from information security perspective.
We show a simple realization of this idea based on Schnorr identification scheme arranged like for ring signatures. We show that with minor modifications one can create a version immune to leakage of ephemeral keys.
We extend the above scenario to the case of k out of n, where the prover must use at least k private keys corresponding to the set of n public keys. With the most probable setting of $k = 2$ or 3, we are talking about the practical case of multifactor authentication that might be necessary for applications with higher security level.
Keywords: identification scheme, ephemeral secret setting, ephemeral secret leakage, deniability, simulatability

1 Introduction

The primary purpose of identification and authentication procedure performed before granting access to certain resources is to check that the applicant – called a *prover* – belongs to the group of users entitled to access these resources. In many cases, essentially there is no need to reveal the real identity of the user and to provide a proof for a future inspection. Nevertheless, in a traditional approach

- the verifier first requests the prover to reveal their identity,
- then the prover has to provide a proof of posession of a secret related to them, e.g. the prover may need to sign a challenge presented by the verifier.

In this way not only the potentially sensitive identity information can be injected into the system, but also a non-volatile cryptographic data of high quality could be created.

Privacy concerns The problem is that this data has to be protected against possible misuse. This obligation follows not only from the general rules of security engineering of "data minimalization", but also has been incorporated to the European legal system via the General Data Protection Regulation – as long as physical persons are concerned. Note also that the consequences of GDPR may also concern IoT as in many cases data about physical persons are indirectly leaked by IoT devices.

Note somewhat obsolete solutions based on a password shared between the prover and the verifier have the advantage of deniability – a transcript of an authentication session cannot convince anybody that an interaction really took place. By replacing this mechanism by a simple challenge-response protocol – where the prover presents a signature of the challenge – this deniability property is lost. A standard approach to reduce the problems of personal data protection is to replace the real identity information by pseudonyms. Then one cannot directly link the data created during an interaction with a physical person. An advanced version of this approach are anonymous credentials, where the system may not remember the access rights of a pseudonymous user; these rights follow from a proof of posession of attributes presented by the prover. For most anonymous credentials schemes, a holder of anonymous credentials presents an implicit pseudonymous identity to the system when presenting their attributes.

For the sake of authentication *pseudonymous signatures* related to pseudonymous identity can be created. There are efficient solutions, where a user holds a single signing key that can be used for all their pseudonyms without violating unlinkability of different pseudonyms. A notable example is Pseudonymous Signature designed by the German federal authority BSI for use in personal identification documents. Another approach is to rely upon group or ring signatures. In this case there is a strong cryptographic evidence for membership in a group. The difference between these two approaches is that in case of group signatures there is a deanonymization procedure, while for ring signatures anonymity within a ring is unconditional.

Deniable and Anonymous Identification In the signature based approaches, an interaction leaves a cryptographic trace that can be used as a proof of interaction against third parties. From the privacy preserving perspective, in the interactive identification process, we require quite opposite feature: the transcript of that interaction should not be used, later on, as a proof that the interaction really occurs. This can be achieved by the *deniability* property of the protocol, which is simulatable without the secret keys by the prover, or even by anybody in the ultimate case. Anonymous identification can be viewed as the extension to regular identification, where the actual prover is hidden within a group of potential

provers. It interactively convince the verifier, holding the set of the public keys, that it possesses one corresponding secret key.

Untrusted Devices Model The scheme computations are performed on prover's and verifier's electronic devices. We consider the scenarios, where users do not control the production process of those devices, and especially are not sure about the fairness of randomness the devices use. Malicious producers can leave back doors for randomness leakage, or evan allow the adversary to set it via a covert side channel. Particularly we consider the *Chosen Prover - Leaked Verifier Ephemeral* (CPLVE) model from [1] allowing the Adversary to adaptively set the ephemeral values for the Prover in each protocol run in the *Query Stage*, and learn ephemeral values of the Verifier in the *Impersonation Stage*, just right after those values are coined at random. Note that the identification protocols that start from the challange, based on the random ephemeral coined at the verifiers device, are not secure in this model. The anonymous ring authentication of Naor [2], and deniable identification schemes of Stinson and Wu [3], and Di Raimondo and Gennaro [4], are no exceptions, being vulnerable in the CPLVE model.

Problem Statement and Motivation Our purpose is to create an identification scheme, which addresses all the issues mentioned above, i.e. which utmostly protects user privacy, and could be securely deployed on electronic devices:

- the scheme is anonymous, i.e. the identity of the prover is protected by, information-theoretic means, within a predefined group of potential identifiers,
- the scheme is deniable, i.e. anyone can create a fake protocol transcripts – and thereby its value as a proof of interaction is useless,
- the authentication proof is strong for the verifier,
- the scheme is secure in CPLVE model, i.e. it could be securely implementable on untrusted devices, where the leakage of randomness is a potential threat.

Contribution The contribution of the paper is the following:

- we propose a Schnorr-like interactive, k-of-n anonymous and deniable identification scheme; we prove its anonymity and deniability;
- we propose a simplified, more efficient version of the general k-of-n for case $k = 1$; we prove its anonymity and deniability;
- we prove the security of the proposed schemes in the CPLVE model from [1].

A case of $k > 1$ can be used when a strong multifactor authentication is required. For instance for $k = 2$, the user has to use two different keys – one located on an identity card and one located on his laptop. Finally, we show that countermeasures devoted to protecting against consequences of ephemeral key leakage can be applied.

Our proposals are **deniable** in honest verifier setting, that is we provide efficient simulators for all transcripts, which produce outputs indistinguishable from honest protocol executions; **privacy-preserving** due to their anonymous nature and secure in case of **ephemeral leakage or setting**, that is in cases when the randomness source is controlled by an adversary, for instance in case of malicious hardware vendor.

Related Work Identification schemes have been proposed since the earliest days of public-key cryptography, e.g. [5–7]. In [8] Schnorr introduced DLP based construction, followed by [9] by Okamoto. There are also specialized identity based IS e.g. [10] provably secure in the standard model, or [11] secure against concurrent man-in-the-middle attack without random oracles by using a variant of BB signature scheme.

The notion of anonymous identification is strongly correlated with anonymous credentials – in fact it may be seen as simply presenting a single credential of a form "I belong to the group". Credential systems, however, usually require third parties and groups management. On the other hand, ring signatures allow any party to create such groups ad-hoc and proving possession of one of multiple, chosen secrets. Anonymous credentials were first introduced by Chaum in [12, 13]. More efficient schemes have been presented by Camenisch and Lysyanskaya in [14–16] and more recently by Pointcheval and Sanders in [17]. Ring signatures have been proposed by Rivest, Shamir and Tauman in [18]. A variation, called *threshold* ring signatures have been proposed by Bresson et al. in [19], which requires k-of-n secret keys to form a signature. The concept of deniable ring authentication, (deniable anonymous identification), has been first introduced by Naor in [2] and later continued by Susilo and Mu [20, 21]. Deniability of regular identification schemes was analysed by Stinson and Wu [3], and Di Raimondo and Gennaro [4].

The security of identification schemes under reset attacks on ephemeral values was raised by Canetti et al. in [22] in the context of zero-knowledge proofs. Countermeasures based on stateless digital signatures have been proposed by Bellare et al. in [23], on the other hand Krzywiecki [24] proposed a countermeasure based on bilinear pairings. In [1] Krzywiecki and Słowik explore deniable identification schemes secure against ephemeral leakage on both Prover's

and Verifier's side, also securing against attacks on deniability via Fiat-Shamir transformation.

2 Preliminaries, Notation and Security Model

Let $x_1, \ldots, x_n \leftarrow_R X$ (or equivalently $x_1, \ldots, x_n \in_\$ X$) mean that each x_i is sampled independently and uniformly at random from the set X. Let $\{a_i\}_I$ denote a set of all elements with indexes $i \in I$, for some set of indexes I.

Let $\mathcal{G}(1^\lambda)$ be a group generation algorithm that takes as an input 1^λ, and outputs a tuple $\mathbb{G} = (G_1, G_2, G_T, g_1, g_2, q)$, where q is a prime number, $G_1 = \langle g_1 \rangle$, $G_2 = \langle g_2 \rangle$, $|G_1| = |G_2| = q$, and G_T is another group of prime order q. Let $\mathcal{H} : \{0,1\}^* \to G_1$ be a hash function. By abuse of notation, we will use $g \in \mathbb{G}$ to denote any generator of any of the three groups.

***Bilinear Map*:** Let G_1, G_2, G_T be prime order q groups as above. We say that function $\hat{e} : G_1 \times G_2 \to G_T$ is a bilinear map when the following conditions hold:

1) *Bilinearity*: $\forall a, b \in \mathbb{Z}_q^*$: $\hat{e}(g_1^a, g_2^b) = \hat{e}(g_1, g_2)^{ab}$.

2) *Non-degeneracy*: $\hat{e}(g_1, g_2) \neq 1$.

3) *Computability*: $\hat{e}$ is efficiently computable.

In this paper we assume that our groups form a type-3 pairing, that is there are no efficiently computable homomorphisms between G_1 and G_2.

The *discrete logarithm* (DL) assumption: For any probabilistic polynomial time (PPT) algorithm $\mathcal{A}_{\mathsf{DL}}$ it holds that:
$\Pr[\mathcal{A}_{\mathsf{DL}}(g, g^x) = x \mid \mathbb{G} \leftarrow_R \mathcal{G}(1^\lambda), x \leftarrow_R \mathbb{Z}_q^*, g \in \mathbb{G}] \leq \epsilon_{\mathsf{DL}}(\lambda)$, where $\epsilon_{\mathsf{DL}}(\lambda)$ is negligible.

The *Computational co-Diffie-Hellman* (CcDH) assumption: [25] For any probabilistic polynomial time (PPT) algorithm $\mathcal{A}_{\mathsf{CcDH}}$ it holds that:
$\Pr[\mathcal{A}_{\mathsf{CcDH}}(\mathbb{G}, a, a^x, b) = b^x \mid \mathbb{G} \leftarrow_R \mathcal{G}(1^\lambda), x \leftarrow_R \mathbb{Z}_q^*, a \in G_1, b \in G_2] \leq \epsilon_{\mathsf{CcDH}}(\lambda)$, where $\epsilon_{\mathsf{CcDH}}(\lambda)$ is negligible.

In an anonymous identification scheme, a prover interacts with a verifier to prove possession of a secret or a set of secrets corresponding to a selected subset of public values. We define the following:

Definition 1 (k-of-n Identification Scheme). *A k-of-n identification scheme* AIS *is a system which consists of four algorithms* $(\mathsf{ParGen}, \mathsf{KeyGen}, \mathcal{P}, \mathcal{V})$ *and a protocol* π*:*

$\mathsf{params} \leftarrow \mathsf{ParGen}(1^\lambda)$**:** *inputs the security parameter* λ*, and outputs public parameters available to all users of the system (we omit them from the rest of the description).*

$(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}()$**:** *outputs a pair* $(\mathsf{sk}, \mathsf{pk})$*, with the secret key* sk *and the corresponding public key* pk*. We assume the procedure* $\mathsf{KeyGen}()$ *is run* n *times producing the set* $\{(\mathsf{sk}_i, \mathsf{pk}_i)\}_1^n$*.*

$\mathcal{P}(\{sk_j\}_J, \{\mathsf{pk}_i\}_1^n)$**:** *denotes the prover – an ITM which interacts with the verifier* $\mathcal{V}$ *in the protocol* π*, where* $J = \{j_1, \ldots, j_k\}$ *are indexes of secret keys used in* $\mathcal{P}$*. Subsequently let* $Z = \{i_1, \ldots, i_z\}$ *denote all the other indexes. In each run we have* $J \cup Z = \{1, \ldots, n\}$ *and* $J \cap Z = \emptyset$*.*

$\mathcal{V}(\{\mathsf{pk}_i\}_1^n)$**:** *denotes the verifier – an ITM which interacts with the prover* $\mathcal{V}$ *in the protocol* π*.*

$\pi(\mathcal{P}, \mathcal{V})$**:** *denotes the protocol between the prover and the verifier.*

We distinguish two stages of the scheme:

- *Initialization: In this stage parameters are generated:* $\mathsf{params} \leftarrow \mathsf{ParGen}(1^\lambda)$*, and keys are registered, e.g. the procedure* $\mathsf{KeyGen}()$ *is run* n *times resulting with the set* $\{(a_i, A_i)\}_1^n$*.*
- *Operation: In this stage* $\mathcal{P}$*, having a subset of* k *secret keys* $\{a_{j_1}, \ldots, a_{j_k}\}$ *denoted as* $\{a_j\}_J$*, demonstrates the knowledge of it to* $\mathcal{V}$ *by performing the protocol* $\pi(\mathcal{P}(\{a_j\}_J, \{A_i\}_1^n), \mathcal{V}(\{A_i\}_1^n))$*. Finally the verifier outputs* 1 *for "accept" or* 0 *for "reject". For simplicity we denote* $\pi(\mathcal{P}, \mathcal{V}) \rightarrow 1$ *if* $\mathcal{P}$ *was accepted by* $\mathcal{V}$ *in* π*.*

We require that the scheme is correct, i.e.:

$$Pr[\mathsf{params} \leftarrow \mathsf{ParGen}(1^\lambda), \{(a_i, A_i)\}_1^n \leftarrow \mathsf{KeyGen}() : \forall_{\{a_j\}_J \subset \{a_i\}_1^n} \pi(\mathcal{P}(\{a_j\}_J, \{A_i\}_1^n), \mathcal{V}(\{A_i\}_1^n)) \rightarrow 1] = 1.$$

Anonymity of the scheme The anonymity of the scheme is a property, which describes the uncertainty of the verifier about the secret keys used by the prover. Intuitively, if the set of the secret keys of the prover is of the cardinality k, and the set of the public keys is of the cardinality n, then the chances of pointing out that the paricular secret key was used by the prover should be k/n, even if secret keys are known to the verifier.

Definition 2 (Anonymity). *Let* $\mathsf{AIS} = (\mathsf{ParGen}, \mathsf{KeyGen}, \mathcal{P}, \mathcal{V}, \pi)$ *be a k-of-n identification scheme. We define anonymity experiment* $\mathsf{Exp}_{\mathsf{AIS}}^{\mathsf{Ano},\lambda,\ell}$*:*

`Init stage` : *Let* $\mathsf{params} \leftarrow \mathsf{ParGen}(1^\lambda)$*,* $\{(\mathsf{sk}_i, \mathsf{pk}_i)\}_1^n \leftarrow \mathsf{KeyGen}()$*. Let the adversary* $\mathcal{A}$*, be the malicious algorithm given the set of all keys* $\{(\mathsf{sk}_i, \mathsf{pk}_i)\}_1^n$*. Let* $J = \{j_1, \ldots, j_k\} \subset \{1, \ldots, n\}$*.*

`Query stage` : $\mathcal{A}$ *can run itself a polynomial number* ℓ *of executions of the protocol since it has all the keys. Let* $\mathsf{v}^{\mathcal{P},\mathcal{V},\ell}$ *is the view* $\mathcal{A}$ *gains after the* ℓ *runs of* π*.*

`Challange stage` : *A challenger $\mathcal{C}$ draws random indexes $J = \{i_1, \ldots, i_k\}$, which are a subset of $\{1, \ldots, n\}$, and runs the protocol:* $\pi(\mathcal{P}(\{\mathsf{sk}_j\}_J, \{\mathsf{pk}_i\}_1^n), \mathcal{A}(\{(\mathsf{sk}_i, \mathsf{pk}_i)\}_1^n, \mathsf{v}^{\mathcal{P},\mathcal{V},\ell}))$. *After that adversary outputs its own index $\hat{d} \leftarrow \mathcal{A}$. The adversary wins if $\hat{d} \in J$.*

We define the advantage of $\mathcal{A}$ in the experiment $\mathtt{Exp}_{\mathsf{AIS}}^{\mathsf{Ano},\lambda,\ell}$ as:

$$\mathbf{Adv}(\mathcal{A}, \mathtt{Exp}_{\mathsf{AIS}}^{\mathsf{Ano},\lambda,\ell}) = |\Pr[\hat{d} \in J] - k/n|.$$

We say that the identification scheme AIS *is anonymous if $\mathbf{Adv}(\mathcal{A}, \mathtt{Exp}_{\mathsf{AIS}}^{\mathsf{Ano},\lambda,\ell})$ is negligible in λ.*

Deniability in Passive and Active Scenarios Deniability is a property which describes whether a knowledge behind protocol can be transferred. For instance, a classic Schnorr IS is deniable in passive scenario, because everyone can generate a valid transcript without any secret values. On the other hand, signature schemes are in general not deniable. Note that in many cases there is a strong link between signature schemes and identification schemes – an identification scheme can be converted into a signature scheme by Fiat-Shamir heuristics [5]: the challenge from an interactive proof may be replaced by output of a hash function on the parameters created before. When it comes to active adversaries, i.e. malicious verifiers, most 3-move authentication schemes lose the deniablility, as the verifier may use the Fiat-Shamir heuristic to turn the identification scheme into a signature scheme. The most natural solution to the problem is for the verifier to commit to their challenge(s) before the first prover's message. This, however becomes a problem in CPLVE adversary model, because the malicious prover may learn the challenge value before their first message and thus simulate the rest of the protocol.

Krzywiecki and Słowik explored these problems in [1] and provided generic solutions to the problem of strong deniability in the leakage scenarios. While we do not require strong deniability in active adversary scenario, analogous solutions are possible in case of our constructions.

Security Against Impersonation Intuitively the scheme is regarded as secure if it is impossible for any adversary prover algorithm $\mathcal{A}$, to be accepted by the verifier given only the public keys, but without the input of the appropriate secret keys. Following the *Chosen Prover - Leaked Verifier Ephemeral* model from [1] we allow the Adversary, in the *Query Stage* of the security experiment: 1) to run a polynomial number ℓ of the protocol executions; 2) to participate in that stage, as a Verifier $\tilde{\mathcal{V}}$, i.e. to adaptively choose messages sent to the Prover; 3) to adaptively set the ephemeral values for the Prover in each protocol run in the *Query Stage*. Moreover, as in the CPLVE model from [1], the adversary can

learn ephemeral values of the Verifier in the *Impersonation Stage*, just right after those values are coined at random.

Security Experiments Let $\bar{x}_i$ be adaptive ephemerals from a malicious Verifier $\tilde{\mathcal{V}}$ 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 AIS 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 3 (Chosen Prover-Leaked Verifier Ephemeral – (CPLVE)). *Let* $\mathsf{AIS} = (\mathsf{ParGen}, \mathsf{KeyGen}_{\mathcal{P}}, \mathsf{KeyGen}_{\mathcal{V}}, \mathcal{P}, \mathcal{V}, \pi)$. *We define security experiment* $\mathtt{Exp}^{\mathsf{CPLVE},\lambda,\ell}_{\mathsf{AIS}}$*:*

`Init Stage` : $\mathsf{params} \leftarrow \mathsf{ParGen}(1^\lambda)$, $\{(\mathsf{sk}, \mathsf{pk})\}_1^n \leftarrow \mathsf{KeyGen}_{\mathcal{P}}(\mathsf{params})$.
Let $J = \{j_1, \ldots, j_k\} \subset \{1, \ldots, n\}$. $\mathcal{A}$:$(\tilde{\mathcal{P}}(\{\mathsf{pk}\}_1^n), \tilde{\mathcal{V}}(\{\mathsf{pk}\}_1^n))$.

`Query Stage` : *For* $i = 1$ *to* ℓ *run* $\pi(\mathcal{P}^{\bar{x}_i}(\{\mathsf{sk}_j\}_J, \{\mathsf{pk}_i\}_1^n), \tilde{\mathcal{V}}(\{\mathsf{pk}_i\}_1^n, \bar{x}_i, v_{i-1}))$, *where* $\bar{x}_i$ *are the adaptive ephemerals from* $\tilde{\mathcal{V}}$ *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* $\pi(\tilde{\mathcal{P}}(\{\mathsf{pk}_i\}_1^n, v_\ell, \bar{\boldsymbol{e}}), \mathcal{V}(\{\mathsf{pk}_i\}_1^n)$, *where* $\bar{\boldsymbol{e}}$ *are the ephemerals of the Verifier leaked to the malicious Prover* $\tilde{\mathcal{P}}$.

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

$$\mathbf{Adv}(\mathcal{A}, \mathtt{Exp}^{\mathsf{CPLVE},\lambda,\ell}_{\mathsf{IS}}) = \Pr[\pi(\tilde{\mathcal{P}}(\{\mathsf{pk}_i\}_1^n, v_\ell, \bar{\boldsymbol{e}}), \mathcal{V}(\{\mathsf{pk}_i\}_1^n) \rightarrow 1].$$

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

3 Proposed Anonymous Identification Schemes Secure in CPLVE

We propose two AIS secure in CPLVE: a general k-of-n scheme and a more efficient 1-of-n scheme. To achieve these we apply the technique from [24], later used in [26, 1, 27] to immune against ephemeral setup values on provers devices. Namely, instead of sending in the last round, the value $s = x + ac$, for the ephemeral x, secret key a, and the challenge c, the prover sends $S = \hat{g}^s$, hiding the vulnerable data in the exponent, for the new generator g obtained from a one way-function. Therefore, even if ephemeral x is set or leaked maliciously, the $\hat{g}^a$ obtained by the adversary should not help in impersonation attack later on.

3.1 Efficient 1-of-n AIS Secure in CPLVE

The construction is depicted in Fig. 1. We use the fact that for a given challenge c, expressed as $c = \sum_{i=1}^{n} c_i$, the $n-1$ values c_i can be set randomly, but at least one c_j is determined by the rest, i.e. $c_j = c - \sum_{i \neq j} c_i$. Assume the prover has a secret key a_j. The anonymity is achieved in the following way: for all public keys for which the prover $\mathcal{P}$ does not possess the secret key, it simulates Schnorr IS transcript, computing c_i, s_i and then X_i. Subsequently it computes X_j for himself according Schnorr IS protocol. Then it aggregates all X_i as the product X send to the verifier. After obtaining the challenge c it computes the missing $c_j = c - \sum_{i \neq j} c_i$ - particular for the secret key a_j - and subsequently compute s_j in a regular Schnorr-like way. Then it aggregates all s_i as the sum s, hides it in the exponent $S = \hat{g}_2^s$ for $\hat{g}_2 = \mathcal{H}(X, c)$, and sends S to the verifier. $\mathcal{V}$ checks if $\hat{e}(g, S) = \hat{e}(X \prod_{i=1}^{n} A_i^{c_i}, \hat{g}_2)$ and $c = \sum_{i=1}^{n} c_i$.

$\mathsf{params} \leftarrow \mathsf{ParGen}(1^\lambda)$**:** Let $\mathbb{G} \leftarrow \mathcal{G}(1^\lambda)$, s.t. CcDH assumption holds. Let $\mathcal{H} : \{0,1\}^* \rightarrow G_2$ be a hash function. Let $\hat{e} : G_1 \times G_2 \rightarrow G_T$ be a bilnear map.
$\mathsf{KeyGen}()$**:**
For a key pair i do $\mathsf{sk}_i = a_i \leftarrow_\$ \mathbb{Z}_q^*$, $\mathsf{pk}_i = A_i = g_1^{a_i}$. Output (a_i, A_i).
$\pi(\mathcal{P}(\{a_j\}_J, \{A\}_1^n), \mathcal{V}(\{A\}_1^n)$**:**

1. $\mathcal{P}$: for $i \in 1, \ldots, n$ s.t. $i \neq j$ compute: $c_i, s_i \leftarrow_\$ \mathbb{Z}_q^*$, $X_i = g_1^{s_i} / A_i^{c_i}$
2. $\mathcal{P}$: chooses $x_j \leftarrow_\$ \mathbb{Z}_q^*$, $X_j = g_1^{x_j}$, $X = \prod_{i=1}^{n} X_i$ and sends X to the verifier $\mathcal{V}$.
3. $\mathcal{V}$: chooses $c \leftarrow_\$ \mathbb{Z}_q^*$, and sends c to the prover $\mathcal{P}$.
4. $\mathcal{P}$: computes $c_j = c - \sum_{i=1, i \neq j}^{n} c_i$, then $s_j = x_j + a_j c_j, \hat{g}_2 = \mathcal{H}(X|c)$, $s = \sum_{i=1}^{n} s_i, S = \hat{g}_2^s$, and sends $S, c_1, \ldots, c_n$ to the verifier $\mathcal{V}$.
5. $\mathcal{V}$: computes $\hat{g}_2 = \mathcal{H}(X|c)$ and accepts the verification iff $\hat{e}(g, S) = \hat{e}\left(X \prod_{i=1}^{n} A_i^{c_i}, \hat{g}_2\right)$ and $c = \sum_{i=1}^{n} c_i$.

Fig. 1. 1-of-n AIS secure in CPLVE.

Theorem 1. *The scheme proposed in Fig. 1 is correct, that is:*

$$Pr[\mathsf{params} \leftarrow \mathsf{ParGen}(1^\lambda), \{(a_i, A_i)\}_1^n \leftarrow \mathsf{KeyGen}() : \forall_{j \in \{1,\ldots,n\}} \pi(\mathcal{P}(a_j, \{A_i\}_1^n), \mathcal{V}(\{A_i\}_1^n)) \rightarrow 1] = 1.$$

Proof. For the valid protocol $(X, c, S, \{c_i\}_1^n)$ generated according to scheme the following equalities hold:

$$\begin{aligned} X \prod_{i=1}^{n} A_i^{c_i} &= \prod_{i=1}^{n} X_i \cdot \prod_{i=1}^{n} A_i^{c_i} = \prod_{i=1, i \neq j}^{n} (g^{s_i} / A_i^{c_i}) \cdot g^{x_j} \cdot \prod_{i=1}^{n} A_i^{c_i} \\ &= \prod_{i=1, i \neq j}^{n} g^{s_i} \cdot g^{x_j} g^{a_j c_j} = \prod_{i=1, i \neq j}^{n} g^{s_i} \cdot g^{s_j} = g^s \\ \hat{e}(g, S) &= \hat{e}(g, \hat{g}_2)^s = \hat{e}(g^s, \hat{g}_2) = \hat{e}(X \prod_{i=1}^{n} A_i^{c_i}, \hat{g}_2) \\ c &= c_j + \sum_{i=1, i \neq j}^{n} c_i = \sum_{i=1}^{n} c_i. \end{aligned}$$

□

Theorem 2. *The scheme proposed in Fig. 1 is deniable.*

Proof. To show the deniability property we show that the transcript of the scheme is simulatable by the following simulator algorithm:

Simulator $\mathcal{S}^{\pi}_{\mathsf{AIS}}()$:
For each $i \in \{1,\ldots,n\}$ do: $(c_i, s_i) \leftarrow_{\$} \mathbb{Z}_q^*$, $X_i = g^{s_i}/{A_i}^{c_i}$
$X = \prod_{i=1}^{n} X_i$, $c = \sum_{i=1}^{n} c_i$, $s = \sum_{i=1}^{n} s_i$, $\hat{g}_2 = \mathcal{H}(X|c)$, $S = \hat{g}_2^s$
return $(X, c, S, \{c_i\}_1^n)$

The tuples of simulated transcript and the real one are identically distributed.

□

Simulation in the CPLVE Model Let's denote ephemerals $\hat{x}_1, \{\hat{c}\}_1^{n-1}, \{\hat{s}\}_0^{n-1}$ as $\hat{E}$, a prover without secret key as $\mathcal{P}^{\hat{E}}(\{\mathsf{pk}_i\}_1^n)$, and an active adversary who injects ephemerals $\hat{E}$ to the prover as $\tilde{\mathcal{V}}^{\mathcal{O}_{\mathcal{H}}}(\{\mathsf{pk}_i\}_1^n, \hat{E})$. In random oracle model (ROM) we can simulate the protocol $\pi(\mathcal{P}^{\hat{E}}(\{\mathsf{pk}_i\}_1^n), \tilde{\mathcal{V}}^{\mathcal{O}_{\mathcal{H}}}(\{\mathsf{pk}_i\}_1^n, \hat{E})) \rightarrow 1$ between the prover and the verifier.

Theorem 3. *The modified scheme (Fig. 1) is simulatable in ROM for the* CPLVE *security model (Def. 3).*

Proof. We define simulator $\mathcal{S}^{\mathsf{CPLVE},\pi}_{\mathsf{AIS}}()$ in the following way:
1) **Hash queries** $\mathcal{O}_{\mathcal{H}}$**:** The simulator will use ROM table for hash queries $\mathcal{O}_{\mathcal{H}}$. The table is build of three columns: first for input, second for output and the last one for masking value. On each query $\mathcal{O}_{\mathcal{H}}(I_i)$ the oracle checks if given input is already in the table. If it is found the oracle returns the corresponding output. Otherwise we choose $r_i \leftarrow_R \mathbb{Z}_q^*$, compute $H_i = g_2^{r_i}$, save tuple (I_i, H_i, r_i) in the table and return H_i.
2) **Commitment:** For each injected tuple (c_i, s_i) we use it to calculate $X_i = g_1^{\hat{s}_i}/A_i^{\hat{c}_i}$. The value $X = \prod_{i=1}^{n-1} X_i \cdot g_1^{\hat{x}_n}$ is sent to the verifier $\tilde{\mathcal{V}}^{\mathcal{O}_{\mathcal{H}}}(\{\mathsf{pk}_i\}_1^n, \hat{E})$.
3) **Proof:** After receiving c from the verifier, we compute $c_n = c - \sum_{i=1}^{n-1}$ and call $\mathcal{O}_{\mathcal{H}}$ with input X, c. We denote response g_2^r from the oracle as $\hat{g}_2$. The proof is computed in the following way:

$$S = \prod_{i=1}^{n} S_i = \prod_{i=1}^{n} \hat{g}_2^{x_i + a_i c_i} = \prod_{i=1}^{n} g_2^{r(x_i + a_i c_i)} = \prod_{i=1}^{n} X_i^r A_i^{r c_i} = X^r \prod_{i=1}^{n} A_i^{r c_i}$$

The verification holds: $\hat{e}(S, g_2) = \hat{e}(\hat{g}_2, X \prod A_i^{c_i})$ for $\hat{g}_2 = g_2^r$. The simulated transcript and the **real** ones are identically distributed. □

Security Analysis We use the same idea as in the case of the original scheme. We show that algorithm $\mathcal{A}$ for which $\mathbf{Adv}(\mathcal{A}, \mathtt{Exp}_{\mathsf{AIS}}^{\mathsf{CPLVE},\lambda,\ell})$ is non-negligible can be used to break underlying CcDH problem with non-negligible probability. First we build an environment for execution by injecting instance of CcDH to our scheme. Then we let the adversary to gain some knowledge by using the simulator described in section 3.1. In the impersonation stage we use rewinding technique to obtain two tuples $(X, c, S, \{c_i\}_1^n)$ and $(X, c', S', \{c'_i\}_1^n)$ which will allow us to break CcDH problem with non-negligible probability.

Theorem 4. *Let* AIS *denote the modified identification scheme (as of Fig. 3).* AIS *is secure (in the sense of Def. 3), i.e. the advantage* $\mathbf{Adv}(\mathcal{A}, \mathtt{Exp}_{\mathsf{AIS}}^{\mathsf{CPLVE},\lambda,\ell})$ *is negligible in* λ*, for any PPT algorithm* $\mathcal{A}$*.*

Proof (Sketch). The proof is by contradiction. Suppose there is an algorithm $\mathcal{A}$ for which the adventage $\mathbf{Adv}(\mathcal{A}, \mathtt{Exp}_{\mathsf{AIS}}^{\mathsf{CPLVE},\lambda,\ell}))$ is non-negligible. We use it as a subprocedure of an efficient algorithm $\mathcal{A}_{\mathsf{CcDH}}$ that breaks the CcDH assumption, computing h^α for the given instance of CcDH problem (g, g^α, h) with non-negligible probability in the following way:

`Init Stage` : Let params be $\mathbb{G}$ from the CcDH problem and let (g, g^α, h) be a CcDH instance in $\mathbb{G}$. We set

$$j \leftarrow_\$ \{1, \ldots, n\}, \quad a_1, \ldots, a_{j-1}, a_{j+1}, \ldots, a_n \leftarrow_\$ \mathbb{Z}_q^*,$$
$$A_1 = g^{a_1}, \ldots, A_{j-1} = g^{a_{j-1}}, A_j = g^\alpha, A_{j+1} = g^{a_{j+1}}, \ldots, A_n = g^{a_n}$$

The adversary $\mathcal{A}$ is given the set of all public keys $\{A_i\}_1^n$. We initialize a ROM table for hash queries $\mathcal{O}_\mathcal{H}$. In the Query Stage we will use simulator $\mathcal{S}_{\mathsf{AIS}}^{\mathsf{CPLVE},\pi}()$ as in the proof of Theorem 3. In the Impersonation Stage $\mathcal{O}_\mathcal{H}$ will output the value h^r, where r is random mask.

`Query Stage` : We simulate ℓ executions of $\pi(\mathcal{P}^{\hat{E}}(\{\mathsf{pk}_i\}_1^n), \tilde{\mathcal{V}}^{\mathcal{O}_\mathcal{H}}(\{\mathsf{pk}_i\}_1^n, \hat{E}))$ without the secret key by using the simulator $\mathcal{S}_{\mathsf{AIS}}^{\mathsf{CPLVE},\pi}()$. Let $\mathsf{v}^{\mathcal{P},\tilde{\mathcal{V}},\bar{\boldsymbol{x}}(\ell)}$ be the view of the adversary, collected in this stage.

`Impersonation Stage` : We run $\pi(\tilde{\mathcal{P}}^{\mathcal{O}_\mathcal{H}}(\{\mathsf{pk}_i\}_1^n, \mathsf{v}^{\mathcal{P},\tilde{\mathcal{V}},\bar{\boldsymbol{x}}(\ell)}), \mathcal{V}(\{\mathsf{pk}_i\}_1^n))$ serving the role of honest verifier. We use the rewinding technique: we fix the commitment X and let $\tilde{\mathcal{P}}$ interact twice with the verifier, choosing each time different challenge, namely c and c', such that neither (X, c) and (X, c') were the input to $\mathcal{O}_\mathcal{H}$ in the Query Stage and setting $\mathcal{O}_\mathcal{H}(X, c) = h^r$, $\mathcal{O}_\mathcal{H}(X, c') = h^{r'}$, for $r, r' \leftarrow_\$ \mathbb{Z}_q^*$. These interactions result with following tuples: $(X, c, S, \{c_i\}_1^n, \hat{g}_2, r)$ and $(X, c', S', \{c'_i\}_1^n, \hat{g}'_2, r')$. If we assume that we accept the adversary both times (that is, the adversary succeeds with

non-negligible probability), we may express:

$$S = \prod_{i=1}^{n} \hat{g}_2^{x_i + a_i c_i} = \hat{g}_2^{\sum_{i=1}^{n} x_i + a_i c_i} = h^{r \sum_{i=1}^{n} x_i + a_i c_i}$$
$$S' = \prod_{i=1}^{n} \hat{g}_2^{\prime x_i + a_i c'_i} = \hat{g}_2^{\prime \sum_{i=1}^{n} x_i + a_i c'_i} = h^{r' \sum_{i=1}^{n} x_i + a_i c'_i}$$

We have that: $S^{(r^{-1})} = h^{\sum_{i=1}^{n} x_i + a_i c_i}$, and $S'^{(r'^{-1})} = h^{\sum_{i=1}^{n} x_i + a_i c'_i}$. So we have: $S^{(r^{-1})}/S'^{(r'^{-1})} = h^{\sum_{i=1}^{n} a_i c_i}/h^{\sum_{i=1}^{n} a_i c'_i} = h^{\sum_{i=1}^{n} a_i (c_i - c'_i)}$. Note that $c - c' \neq 0$, thus $\sum_i (c_i - c'_i) \neq 0$. Therefore for at least one index i the difference $(c_i - c'_i) \neq 0$. Because of our random choice of j we have $1/n$ chance that j would be that index. If so, with non-negligible probability, we have: $S^{(r^{-1})}/S'^{(r'^{-1})} = h^{(\alpha(c_j - c'_j) + \sum_{i \neq j} a_i (c_i - c'_i))}$. Thus we have: $h^{\alpha} = (S^{(r^{-1})}/S'^{(r'^{-1})} h^{\sum_{i \neq j} a_i (c_i - c'_i)})^{(c_j - c'_j)^{-1}}$. □

Theorem 5. *The scheme proposed in Fig. 1 is unconditionally anonymous.*

Proof. Let $(X, c, S, \{c_i\}_1^n)$ be the transcript of $\pi(\mathcal{P}(\_, \{A_i\}_1^n), \mathcal{V}(\{A_i\}_1^n))$. That is $X = \prod_{i=1}^{n} X_i$, and $S = (\mathcal{H}(X, c))^s$ for $s = \sum_{i=1}^{n} s_i$. That transcript, for the challenge c, can be produced by a user j by setting the exact c_i, s_i for $i \neq j$ and the exact value x_j s.t:

$$X = g^{x_j} \prod_{i \neq j} (g^{s_i}/A_i^{c_i}), \quad s = x_j + a_j (c - \sum_{i \neq j} c_i) + \sum_{i \neq j} s_i,$$
$$\{c_i\}_1^n = \{c_1, \ldots, c_{j-1}, (c - \sum_{i \neq j} c_i), c_{j+1}, \ldots, c_n\}.$$

The probability that the prover with the index j computes the exact c_i, s_i for $1 \leq i \leq n$, $i \neq j$, which altogether with the challenge c, determine c_j and s_j is $(1/(q^2))^{n-1}$. Then the probability that the user j chooses exactly the value $x_j \in \mathbb{Z}_q^*$ is $1/q$. Summing up, the probability that the user j generates the transcript is $(1/q) \cdot (1/(q^2))^{n-1}$. This probability does not depend on j so it is the same for all users of the group. □

3.2 General k-of-n Anonymous AIS Secure in CPLVE

The construction is depicted in Fig. 1. Let $k + z = n$. If we have a polynomial $L(x)$ of degree $z - 1$, and a set of shares $P = \{(x_i, y_i)\}_1^n$, s.t. $y_i = L(x_i)$ then the following conditions are true: 1) each subset of P of cardinality z can be used to interpolate the polynomial L; 2) a least k shares of the form $(x_j, L(x_j))$ were added to P after L was constructed. The anonymity is achieved in the following way: Assume the prover has secret keys $\{a_j\}_J$ for indexes from J. For all public keys for which the prover $\mathcal{P}$ does not possess the secret keys, it simulates the regular Schnorr IS transcripts, computing c_i, s_i and then X_i. Subsequently it computes X_j for each $j \in J$ according to the regular Schnorr IS protocol with the key a_j. Then it sends all X_i and X_j to the verifier. The verifier returns a challenge of random shares $P_C = \{(x_i, y_i)\}_1^k$. $\mathcal{P}$ prepares

shares $P_Z = \{(\mathcal{H}(X_i), c_i)\}_Z$ using simulated values from previous step. $\mathcal{P}$ interpolates a polynomial $L(x)$ for points $P_C \cup P_Z$, and computes for each $j \in J$: the missing $c_j = L(\mathcal{H}(X_j))$ and s_j in a regular Schnorr-like way. Then it sends $\{c_i, \mathcal{H}_{g_2}(X, c)^{s_i}\}_1^n$ to $\mathcal{V}$. The verifier, for each $i \in I$, verifies that s_i, hidden in the exponent, fulfills the Schnorr equation, and that the polynomial $L_{\bar{P}}$ interpolated for points $\bar{P} = \{(\mathcal{H}(X_i), c_i)\}_1^n$ also includes points from the challenge P_C.

Let $I = \{i\}_1^n$, $J = \{j_1, \ldots, j_k\} \subset I$, $Z = \{i_1, \ldots, i_z\} \subset I$, $J \cup Z = I$, $J \cap Z = \emptyset$.

$\mathsf{params} \leftarrow \mathsf{ParGen}(1^\lambda)$**:** Let $\mathbb{G} \leftarrow \mathcal{G}(1^\lambda)$, s.t. CcDH assumption holds. Let $\mathcal{H}_{g_2} : \{0,1\}^* \rightarrow G_2$, and $\mathcal{H} : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$ be hash functions. Let $\hat{e} : G_1 \times G_2 \rightarrow G_T$ be a bilnear map. Set $\mathsf{params} = (G_1, G_2, G_T, g_1, g_2, q, \mathcal{H}_{g_2}, \mathcal{H}, \hat{e})$.

$\mathsf{KeyGen}()$**:**

For a key pair i do $\mathsf{sk}_i = a_i \leftarrow_\$ \mathbb{Z}_q^*$, $\mathsf{pk}_i = A_i = g_1^{a_i}$. Output (a_i, A_i).

$\pi(\mathcal{P}(\{a_j\}_J, \{A_i\}_1^n), \mathcal{V}(\{A_i\}_1^n))$**:**

1. $\mathcal{P}$: set $X_Z = \{X_i\}_Z$, s.t. $s_i, c_i \leftarrow_\$ \mathbb{Z}_q^*$, $X_i = g_1^{s_i}/A_i^{c_i}$ for each $i \in Z$.
2. $\mathcal{P}$: set $X_J = \{X_j\}_J$, s.t. for each $j \in J$ compute $x_j \leftarrow_\$ \mathbb{Z}_q^*$, $X_j = g_1^{x_j}$.
3. $\mathcal{P}$: sends $X = X_Z \cup X_J$ to the verifier $\mathcal{V}$.
4. $\mathcal{V}$: sets $P_C = \{(x_i, y_i)\}_1^k$, where each pair $x_i, y_i \leftarrow_\$ \mathbb{Z}_q^*$.
5. $\mathcal{V}$: computes $\hat{g}_2 = \mathcal{H}_{g_2}(X, P_C)$, sends P_C to the provers $\mathcal{P}$.
6. $\mathcal{P}$: compute the set $P_Z = \{(x_i, y_i)\}_Z$, s.t. $x_i = \mathcal{H}(X_i)$, $y_i = c_i$ for each $i \in Z$.
7. $\mathcal{P}$: sets $P = P_C \cup P_Z$, interpolates a polynomial $L_P(x)$ for points P.
8. $\mathcal{P}$: computes $\hat{g}_2 = \mathcal{H}_{g_2}(X, P_C)$
9. $\mathcal{P}$: for each $j \in J$, computes $c_j = L_P(\mathcal{H}(X_j))$, $s_j = x_j + a_j c_j$.
10. $\mathcal{P}$: for each $i \in I$ computes $S_i = \hat{g}_2^{s_i}$, sends $\{c_i, S_i\}_1^n$ to the verifier $\mathcal{V}$.
11. $\mathcal{V}$: sets $\bar{P} = \{(x_i, y_i)\}_1^n$, s.t. $x_i = \mathcal{H}(X_i)$, $y_i = c_i$ for each $i \in I$.
12. $\mathcal{V}$: interpolates a polynomial $L_{\bar{P}}(x)$ for points $\bar{P}$.
13. $\mathcal{V}$: accepts the verification iff $(\forall_{\{i \in I\}}\ \hat{e}(g_1, S_i) = \hat{e}(X_i A_i^{c_i}, \hat{g}_2)$ and $(\forall_{\{(x_i, y_i) \in P_C\}}\ L_{\bar{P}}(x_i) = y_i)$.

Fig. 2. The k-of-n anonymous Schnorr identification scheme.

Theorem 6. *The scheme proposed in Fig. 2 is correct, that is:*

$$Pr[\mathsf{params} \leftarrow \mathsf{ParGen}(1^\lambda), \{(a_i, A_i)\}_1^n \leftarrow \mathsf{KeyGen}() : \forall_{\{a_j\}_J \subset \{a_i\}_1^n}\ \pi(\mathcal{P}(\{a_j\}_J, \{A_i\}_1^n), \mathcal{V}(\{A_i\}_1^n)) \rightarrow 1] = 1.$$

Proof. For the valid protocol $(\{X_i\}_1^n, P_C, \{c_i, S_i\}_1^n)$ generated according to scheme the following equalities hold:

$$\begin{aligned}
\forall_{(i\in Z)}\ \ g_1^{s_i} &= g_1^{s_i - a_i c_i + a_i c_i} = g_1^{s_i}/A_i^{c_i} \cdot A_i^{c_i} = X_i A_i^{c_i}.\\
\forall_{(j\in J)}\ \ g_1^{s_j} &= g_1^{x_j + a_j c_j} = X_j A_j^{c_j}.\\
\forall_{\{i\in Z\cup J\}}\ \hat{e}\,(g_1, S_i) &= \hat{e}\,(g_1, \hat{g}_2)^{s_i} = \hat{e}\,(X_i A_i^{c_i}, \hat{g}_2)\,.
\end{aligned}$$

Moreover:
1) $P = P_C \cup P_Z$ and $L_P(x)$ is a polynomial interpolated for points P.
2) $P_J = \{(x_j, y_j)\}_J : \forall_{(j\in J)}\ (x_j = \mathcal{H}(X_j), y_j = L_P(x_j))\}$.
3) $\bar{P} = P_j \cup P_Z$ and $L_{\bar{P}}(x)$ is a polynomial interpolated for points $\bar{P}$.
Obviously $L_{\bar{P}}(x) = L_P(x)$ thus $(\forall_{\{(x_i,y_i)\in P_C\}}\ L_{\bar{P}}(x_i) = y_i)$. □

Theorem 7. *The scheme proposed in Fig. 2 is deniable.*

Proof. To show the deniability property we show that the transcript of the scheme is simulatable by the following simulator algorithm:

Simulator $\mathcal{S}_{\mathsf{AIS}}^{\pi,k,n}()$:
For each $i \in \{1,\ldots,n\}$ do: $(c_i, s_i) \leftarrow_{\$} \mathbb{Z}_q^*$, $X_i = g_1^{s_i}/{A_i}^{c_i}$
$\hat{g}_2 = \mathcal{H}_{g_2}(\{X_i\}_1^n)$
For each $i \in \{1,\ldots,n\}$ do: $S_i = \hat{g}_2^{s_i}$
$P = \{(x_i, y_i)\}_1^n$, s.t. $x_i = \mathcal{H}(X_i)$, $y_i = c_i$ for each $i \in \{1,\ldots,n\}$
Interpolate the polynomial $L_P(x)$ for points P
$P_C = \{(x_i, y_i)\}_1^k$, s.t. $x_i \leftarrow_{\$} \mathbb{Z}_q^*$, $y_i = L_P(x_i)$ for each $i \in \{1,\ldots,k\}$
`return` $(\{X_i\}_1^n, P_C, \{c_i, S_i\}_1^n)$

The tuples of simulated transcript and the real one are identically distributed.
□

Theorem 8. *The scheme proposed in Fig. 2 is unconditionally anonymous.*

Proof. The reasoning is analogical to the one from proof of Theorem 5. Let $(\{X_i\}_1^n, P_C, \{c_i, s_i\}_1^n)$ be the transcript of $\pi(\mathcal{P}(\{a_j\}_J, \{A_i\}_1^n), \mathcal{V}(\{A_i\}_1^n))$, for some secret keys with indexes from J, s.t. $|J| = k$. For each $j \in J$ the prover algorithm $\mathcal{P}$ has to set the exact value x_j for the j. Besides, it has to set the exact values c_i, s_i for all indexes $i \in Z$ related to the public keys - not corresponding to known secret keys. The probability that $\mathcal{P}$ computes such exact values, which generates that particular transcript, does not depend on keys indexes. Thus per each secret key used, $\mathcal{P}$ has the same chance to compute the exact parameters, that define the transcript in question. Therefore, the probability, that a particular a_j was used, is k/n, and per each group of secret keys $\{a_j\}_J$ of cardinality k, the probability of forming that particular transcript is the same, and unrelated to values of indexes from J. □

Theorem 9. *The modified scheme (Fig. 2) is simulatable in ROM for the* CPLVE *security model (Def. 3).*

Proof. We define simulator $\mathcal{S}_{\mathsf{AIS}}^{\mathsf{CPLVE},\pi,k,n}()$ in the following way:
1) **Hash queries** $\mathcal{O}_{\mathcal{H}_{g_2}}$**:** The simulator will use ROM table for hash queries $\mathcal{O}_{\mathcal{H}_{g_2}}$. The table is build of three columns: first for input, second for output and the last one for masking value. On each query $\mathcal{O}_{\mathcal{H}_{g_2}}(I_i)$ the oracle checks if given input is already in the table. If it is found the oracle returns the corresponding output. Otherwise we choose $r_i \leftarrow_R \mathbb{Z}_q^*$, compute $H_i = g_2^{r_i}$, save tuple (I_i, H_i, r_i) in the table and return H_i.
2) **Commitment:** For each injected tuple (c_i, s_i) we use it to calculate $X_i = g_1^{\hat{s}_i}/A_i^{\hat{c}_i}$. For each injected tuple $\hat{x}_n$ we use it to calculate $X_i = g_1^{\hat{x}_n}$ The set $\{X_i\}_1^n$ is sent to the verifier $\tilde{\mathcal{V}}^{\mathcal{O}_{\mathcal{H}_{g_2}}}(\{\mathsf{pk}_i\}_1^n, \hat{E})$.
3) **Proof:** After receiving P_C from the verifier, we compute $L_P(x)$ in a regular way, and call $\mathcal{O}_{\mathcal{H}_{g_2}}$ with input $\{X_i\}_1^n, P_C$. We denote response g_2^r from the oracle as $\hat{g}_2$. Then for each $i = 1, \ldots, n$ compute $S_i = X_i^r A_i^{rc_i}$, and sends $\{c_i, S_i\}_1^n$ to the verifier $\mathcal{V}$. The verification on the verifier side holds: $\hat{e}(S_i, g_2)$=$\hat{e}(X_i A_i^{c_i}, \hat{g}_2)$ for $\hat{g}_2 = g_2^r$. Note that polynomials $L_P(x)$ and $L_{\bar{P}}(x)$ computed on both sides separately are the same. Thus the simulated transcript and the **real** one are identically distributed. □

Theorem 10. *Let* AIS *denote the k-of-n scheme (as of Fig. 2).* AIS *is secure (in the sense of Def. 3), i.e. the advantage* $\mathbf{Adv}(\mathcal{A}, \mathtt{Exp}_{\mathsf{AIS}}^{\mathsf{CPLVE},\lambda,\ell}))$ *is negligible in* λ*, for any PPT algorithm* $\mathcal{A}$*.*

Proof (Sketch). The proof is by contradiction. Suppose there is an algorithm $\mathcal{A}$ for which the advantage $\mathbf{Adv}(\mathcal{A}, \mathtt{Exp}_{\mathsf{AIS}}^{\mathsf{CPLVE},\lambda,\ell}))$ is non-negligible. We use it as a subprocedure of an efficient algorithm $\mathcal{A}_{\mathsf{CcDH}}$ that breaks the CcDH assumption, computing h^α for the given instance of CcDH problem (g, g^α, h) with non-negligible probability in the following way:

`Init Stage` : Let params be $\mathbb{G}$ from the CcDH problem and let (g, g^α, h) be a CcDH instance in $\mathbb{G}$. We set

$$d_1, \ldots, d_n \leftarrow \mathbb{Z}_q^*, Y = g^\alpha, A_1 = Y^{d_1} = g^{\alpha d_0}, \ldots, A_n = Y^{d_n} = g^{\alpha d_n}.$$

The adversary $\mathcal{A}$ is given the set of all public keys $\{A_i\}_1^n$. We initialize a ROM table for hash queries $\mathcal{O}_{\mathcal{H}}$. In the Query Stage we will use simulator $\mathcal{S}_{\mathsf{AIS}}^{\mathsf{CPLVE},\pi,k,n}()$, as in the proof of Theorem 3. In the Impersonation Stage $\mathcal{O}_{\mathcal{H}}$ will output the value h^r, where r is random mask.

`Query Stage` : We simulate l executions of $\pi(\mathcal{P}^{\hat{E}}(\{\mathsf{pk}_i\}_1^n), \tilde{\mathcal{V}}^{\mathcal{O}_{\mathcal{H}}}(\{\mathsf{pk}_i\}_1^n, \hat{E}))$ without the secret key by using the simulator $\mathcal{S}_{\mathsf{AIS}}^{\mathsf{CPLVE},\pi,k,n}()$. Let $\mathsf{v}^{\mathcal{P},\tilde{\mathcal{V}},\bar{\boldsymbol{x}}(\ell)}$ be the view of the adversary collected in this stage.

`Impersonation Stage` : We run $\pi(\tilde{\mathcal{P}}^{\mathcal{O}_{\mathcal{H}}}(\{\mathsf{pk}_i\}_1^n, \mathsf{v}^{\mathcal{P},\tilde{\mathcal{V}},\bar{\boldsymbol{x}}(\ell)}), \mathcal{V}(\{\mathsf{pk}_i\}_1^n))$ serving the role of honest verifier. We use the rewinding technique: we fix the

commitment $\{X_i\}_1^n$ and let $\tilde{\mathcal{P}}$ interact twice with the verifier, choosing each time different challenge, namely c and c', such that neither $(\{X_i\}_1^n, P_C)$ and $(\{X_i\}_1^n, P'_C)$ were the input to $\mathcal{O}_{\mathcal{H}_{g_2}}$ in the Query Stage and setting $\mathcal{O}_{\mathcal{H}_{g_2}}(\{X_i\}_1^n, P_C) = h^r$, $\mathcal{O}_{\mathcal{H}_{g_2}}(\{X_i\}_1^n, P'_C) = h^{r'}$, for $r, r' \leftarrow_{\$} \mathbb{Z}_q^*$. These result with: $(\{X_i\}_1^n, P_C, \{c_i, S_i\}_1^n, \hat{g}_2, r)$ and $(\{X_i\}_1^n, P'_C, \{c'_i, S'_i\}_1^n, \hat{g}'_2, r')$. If we assume that we accept the adversary both times (that is, the adversary succeeds with non-negligible probability), then we have for each i: $g^{s_i} = X_i A_i^{c_i}$ and $g^{s'_i} = X_i A_i^{c'_i}$. Because $P_c \neq P'_c$ there is at least one index i s.t. $S_i \neq S'_i$. So we have: $({S_i}^{(r^{-1})})/({S'_i}^{(r'^{-1})}) = h^{a_i(c_i - c'_i)} = h^{\alpha d_i(c_i - c'_i)}$. Finally, we have: $h^{\alpha} = (({S_i}^{(r^{-1})})/({S'_i}^{(r'^{-1})}))^{((d_i(c_i - c'_i))^{-1})}$. □

4 Switch-back to Regular Security Model

Observe that the proposed schemes, secure in the CPLVE model, can be easily converted to schemes secure in the regular model (without ephemeral leakages), i.e. secure in the model from Definition 3, where corresponding sets of injected and leaked ephemeras are empty: $\{\bar{x}_i\}_1^{\ell} = \emptyset$, and $\bar{\boldsymbol{e}} = \emptyset$.

We setup both schemes in groups where DL problem is hard. Then we modify the 1-of-n scheme protocol π in the following way: In Step 4. from Fig. 1 the prover sends to the verifier the value s instead of S. Then in Step 5. the verifier checks if $g^s = X \prod_{i=1}^n A_i^{c_i}$ instead of $\hat{e}(g, S) = \hat{e}(X \prod_{i=1}^n A_i^{c_i}, \hat{g}_2)$. Similarly, we modify the k-of-n scheme protocol π in the following way: In Step 10. from Fig. 2 the prover sends to the verifier $\{c_i, s_i\}_1^n$ instead of $\{c_i, S_i\}_1^n$. While in Step 13. the verifier checks for each $i \in I$ if $g^{s_i} = X_i A_i^{c_i}$ instead of $\hat{e}(g, S) = \hat{e}(X \prod_{i=1}^n A_i^{c_i}, \hat{g}_2)$. In both cases, this effectively removes the need for pairing-friendly groups, as all operations are performed in G_1 and $\mathbb{Z}_q^*$. The resulting schemes are *deniable*, *secure* for impersonation, and *anonymous*.

5 Conclusion

We proposed 1-of-n and k-of-n interactive anonymous identification schemes, that support privacy of users *two-fold*: namely privacy regarded as ability to deny the paticipation in the protocol interaction, and privacy regarded as anonymity of identifiers hidden in the subset of potential provers. The schemes withstand the impersonation attacks in the strong CPLVE model, which allow the adversary to set provers ephemerals in the query stage, and learn verifier ephemerals during the impersonation stage of the attack. This justifies for implementation on devices, which manufacturing process is not under the sole control of the end-users, and when fair randomness cannot be guaranteed.

References

1. Krzywiecki, Ł., Słowik, M.: Strongly deniable identification schemes immune to prover's and verifier's ephemeral leakage. In: International Conference for Information Technology and Communications, Springer (2017) 115–128
2. Naor, M.: Deniable ring authentication. In: Annual International Cryptology Conference, Springer (2002) 481–498
3. Stinson, D.R., Wu, J.: An efficient and secure two-flow zero-knowledge identification protocol. IACR Cryptology ePrint Archive **2006** (2006) 337. Available from: http://eprint.iacr.org/2006/337
4. Raimondo, M.D., Gennaro, R.: New approaches for deniable authentication. J. Cryptology **22**(4) (2009) 572–615. Available from: https://doi.org/10.1007/s00145-009-9044-3
5. Fiat, A., Shamir, A.: How To Prove Yourself: Practical Solutions to Identification and Signature Problems. In: Advances in Cryptology — CRYPTO '86: Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg (1987) 186–194. Available from: http://dx.doi.org/10.1007/3-540-47721-7_12
6. Feige, U., Fiat, A., Shamir, A.: Zero-knowledge proofs of identity. Journal of cryptology **1**(2) (1988) 77–94
7. Guillou, L.C., Quisquater, J.J.: A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. In: Lecture Notes in Computer Science on Advances in Cryptology-EUROCRYPT'88, New York, NY, USA, Springer-Verlag New York, Inc. (1988) 123–128. Available from: http://dl.acm.org/citation.cfm?id=55554.55565
8. Schnorr, C.P.: Efficient signature generation by smart cards. J. Cryptology **4**(3) (1991) 161–174
9. Okamoto, T.: Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes. In: Advances in Cryptology — CRYPTO' 92: 12th Annual International Cryptology Conference Santa Barbara, California, USA August 16–20, 1992 Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg (1993) 31–53. Available from: http://dx.doi.org/10.1007/3-540-48071-4_3
10. Kurosawa, K., Heng, S.H.: Identity-Based Identification Without Random Oracles. In: Computational Science and Its Applications – ICCSA 2005: International Conference, Singapore, May 9-12, 2005, Proceedings, Part II. Springer Berlin Heidelberg, Berlin, Heidelberg (2005) 603–613. Available from: http://dx.doi.org/10.1007/11424826_64
11. Kurosawa, K., Heng, S.H.: The Power of Identification Schemes. In: Public Key Cryptography - PKC 2006: 9th International Conference on Theory and Practice in Public-Key Cryptography, New York, NY, USA, April 24-26, 2006. Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg (2006) 364–377. Available from: http://dx.doi.org/10.1007/11745853_24
12. Chaum, D.: Security without identification: Transaction systems to make big brother obsolete. Communications of the ACM **28**(10) (1985) 1030–1044
13. Chaum, D.: Showing credentials without identification transferring signatures between unconditionally unlinkable pseudonyms. In: International Conference on Cryptology, Springer (1990) 245–264
14. Camenisch, J., Lysyanskaya, A.: An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: International Conference on the Theory and Applications of Cryptographic Techniques, Springer (2001) 93–118
15. Camenisch, J., Lysyanskaya, A.: A signature scheme with efficient protocols. In: Security in communication networks. Springer (2002) 268–289

16. Camenisch, J., Lysyanskaya, A.: Signature schemes and anonymous credentials from bilinear maps. In: Annual International Cryptology Conference, Springer (2004) 56–72
17. Pointcheval, D., Sanders, O.: Short randomizable signatures. In: Cryptographers' Track at the RSA Conference, Springer (2016) 111–126
18. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In Boyd, C., ed.: Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings. Volume 2248 of LNCS., Springer (2001) 552–565
19. Bresson, E., Stern, J., Szydlo, M.: Threshold ring signatures and applications to ad-hoc groups. In: Annual International Cryptology Conference, Springer (2002) 465–480
20. Susilo, W., Mu, Y.: Non-interactive deniable ring authentication. In: International Conference on Information Security and Cryptology, Springer (2003) 386–401
21. Susilo, W., Mu, Y.: Deniable ring authentication revisited. In: International Conference on Applied Cryptography and Network Security, Springer (2004) 149–163
22. 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 '00, New York, NY, USA, ACM (2000) 235–244. Available from: http://doi.acm.org/10.1145/335305.335334
23. Bellare, M., Fischlin, M., Goldwasser, S., Micali, S.: Identification Protocols Secure against Reset Attacks. In: Advances in Cryptology — EUROCRYPT 2001: International Conference on the Theory and Application of Cryptographic Techniques Innsbruck, Austria, May 6–10, 2001 Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg (2001) 495–511. Available from: http://dx.doi.org/10.1007/3-540-44987-6_30
24. Krzywiecki, L.: Schnorr-like identification scheme resistant to malicious subliminal setting of ephemeral secret. In Bica, I., Reyhanitabar, R., eds.: Innovative Security Solutions for Information Technology and Communications - 9th International Conference, SECITC 2016, Bucharest, Romania, June 9-10, 2016, Revised Selected Papers. Volume 10006 of Lecture Notes in Computer Science. (2016) 137–148. Available from: https://doi.org/10.1007/978-3-319-47238-6_10
25. Saito, T., Uchiyama, S.: The co-diffie-hellman problem over elliptic curves. Reports of the Faculty of Science and Engineering **33**(1) (2004) 1–8
26. Krzywiecki, L., Kutylowski, M.: Security of okamoto identification scheme: a defense against ephemeral key leakage and setup. In Wang, C., Kantarcioglu, M., eds.: Proceedings of the Fifth ACM International Workshop on Security in Cloud Computing, SCC@AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2, 2017, ACM (2017) 43–50. Available from: https://doi.org/10.1145/3055259.3055267
27. Krzywiecki, L., Wlislocki, T.: Deniable key establishment resistance against ekci attacks. Security and Communication Networks **2017** (2017) 7810352:1–7810352:13. Available from: https://doi.org/10.1155/2017/7810352