The preliminary version of this paper appeared in *Proceedings of the 2018 International Conference on Information and Communications Security - ICICS 2018* - under the same title. This is the full version.

Witness-Indistinguishable Arguments with Σ-Protocols for Bundled Witness Spaces and its Application to Global Identities

Hiroaki Anada[1] and Seiko Arita[2]

[1] Department of Information Security, University of Nagasaki
W408, 1-1-1, Manabino, Nagayo-cho, Nishisonogi-gun, Nagasaki, 851-2195 Japan
`anada@sun.ac.jp`

[2] Graduate School of Information Security, Institute of Information Security
509, 2-14-1, Tsuruya-cho, Kanagawa-ku, Yokohama, 221-0835 Japan
`arita@iisec.ac.jp`

June 20, 2020

Abstract. We propose a generic construction of a Σ-protocol of commit-and-prove type, which is an AND-composition of Σ-protocols on statements that include a common commitment. Our protocol enables a prover to convince a verifier that the prover knows a bundle of witnesses that have a common component which we call a base witness point. When the component Σ-protocols are of witness-indistinguishable argument systems, our Σ-protocol is also a witness-indistinguishable argument system as a whole. As an application, we propose a decentralized multi-authority anonymous authentication scheme. We first give a syntax and security definitions of the scheme. Then we give a generic construction of the scheme. There a witness is a bundle of witnesses each of which consists of a common global identity string and a digital signature on it. We mention an instantiation in the setting of bilinear groups.

Keywords: interactive proof, sigma protocol, witness indistinguishability, decentralized, anonymity, collusion resistance

Table of Contents

1 Introduction

Global identities such as Passport Numbers (PNs), Social Security Numbers (SSNs) and e-mail addresses as global identifiers are currently common for identification. They are used not only for governmental identification but also for commercial services; that is, when we want to use a commercial service, we often ask the service administration authority for issuing an attribute certificate at the registration phase. In the phase, the authority confirms our identities by verifying the global identity string such as PN or SSN. Once the attribute certificate is issued, we become to be accepted at the authentication phase of the service. Hence the global identity strings work for us to be issued our attribute certificates. It is notable that recently multi-factor authentication schemes are utilized to prevent misauthentication. In the scheme a user of a service is granted access only after presenting several separate pieces of evidence. Actually the multi-factor authentication of using both a laptop PC, which is connected to Internet by a service provider, and a smartphone, which is activated by a cellular carrier, is getting usual. Thus, there is a compound model that involves independent administration authorities for us to be authenticated and receive benefit of a service.

Privacy protection is a function to be pursued in authentication. The growth of companies in the areas of the IT infrastructures made protecting privacy more critical for involved users because we use search engines, digital devices, social networking services and e-shopping services everyday. Considering this change of circumstance, the authentication framework using identity strings and passwords should be evolved into a framework where anonymity is guaranteed at the authentication phase. For example, when a smart household machine sends a report about the situation of a house via Internet as a query for useful suggestion (such as air conditioning or cooking recipes), the identity information is often unnecessary. A further example is connected vehicles which are connected to Internet and which use a combination of plural services like a local traffic information system and the passenger's web-scheduler, where the identity information should not be leaked even when the memberships should be made in the registration phase. In this example a user should be authenticated by the service providers simultaneously in the authentication phase, anonymously. This is an authentication framework in which plural attributes of a single user are authenticated. However, there is a threat on such anonymous authentication frameworks; *collusion attack*. For example, two malicious users with different identities bring together their private attribute certificates, and try to make a verifier accept anonymously by using the merged attribute certificates. Here the vary anonymity is a critical potential drawback from the view point of the collusion attack.

1.1 Our Contribution and Related Work

In this paper, we propose a new notion of a proof system; a witness-indistinguishable argument system (WIA) with Σ-protocols for a *bundled witness space*. It is known that WIA is a natural building block to achieve anonymity in cryptographic primitives ([Gol01]). However, there is no previous work for the multi-prover setting executed by a *hidden single prover* who is able to convince a verifier that she is certainly a single prover, though she is anonymous. By employing a commitment scheme as one of the building blocks we construct the kind of WIA as a kind of commit-and-prove scheme [CLOS02,EG14].

As an application, we give a generic construction of a decentralized multi-authority anonymous authentication scheme, which can be converted into a decentralized multi-authority attribute-based signature scheme (DMA-ABS) [OT13]. In an ABS scheme, a signer has certificates, which are also keys, on her attributes. The signer is able to sign a message which is associated with a signing policy expressed as a boolean formula on attributes if and only if her attributes satisfy the boolean formula. There are assignment patterns to satisfy the boolean formula, and the attribute privacy of an ABS scheme should assure that the signatures do not leak any information on the assignment pattern which she used. It should be noted that the attribute privacy implies the anonymity of the signer's identity. On the other hand, decentralized multi-authorities mean that there are independent key-issuing authorities each of which generates each private secret key for her. Our WIA can actually be converted into a DMA-ABS scheme if a prover chooses a monotone boolean formula instead of an all-AND formula, and if we apply the Fiat-Shamir transform [FS86] to the Σ-protocol of our authentication scheme.

The difference between the previous DMA-ABS schemes and our DMA-ABS scheme is that, in our DMA-ABS scheme, when a signer wants the authorities to issue private secret keys for her, the authorities *simply generate digital signatures on her single global identity string*. This feature is useful when her global identity string is easy to be validated.

1.2 Organization of the Paper

In Section 2, we prepare for needed notions and notations. In Section 3, we describe building blocks and give a generic construction of our witness-indistinguishable argument system with a Σ-protocol for the bundled witness space. In Section 4, we first define a syntax and security notions of our decentralized multi-authority anonymous authentication scheme. Then, we give a generic construction of the scheme. In Section 5, we conclude our work. In Appendix A, we briefly show an instantiation of the scheme in the setting of bilinear groups.

2 Preliminaries

The set of natural numbers is denoted by $\mathbb{N}$. We put $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$. The residue class ring of integers modulo a prime number p is denoted by $\mathbb{Z}_p$. The security parameter is denoted by λ. The bit length of a string a is denoted by $|a|$. The number of elements of a set S is denoted by $|S|$. A uniform random sampling of an element a from a finite set S is denoted as $a \in_R S$. The expression $a =_? b$ returns a boolean 1 (TRUE) when $a = b$, and otherwise 0 (FALSE). When an algorithm A with an input a returns z, we denote it as $z \leftarrow A(a)$, or, $A(a) \rightarrow z$. St means the inner state of an algorithm. When a probabilistic algorithm A with an input a and a randomness r on a random tape returns z, we denote it as $z \leftarrow A(a; r)$ When an algorithm A with an input a and an algorithm B with an input b interact with each other and B returns z, we denote it as $z \leftarrow \langle A(a), B(b)\rangle$. The transcript of all the messages of the interaction is denoted by $transc\langle A(a), B(b)\rangle$. When an algorithm A accesses an oracle $\mathbf{O}$, we denote it by $A^{\mathbf{O}}$. When A accesses n oracles $\mathbf{O}_1, \dots, \mathbf{O}_n$ concurrently (i.e. in arbitrarily interleaved order of messages), we denote it by $A^{\mathbf{O}_i|_{i=1}^n}$. A probability of an event E is denoted by $\Pr[E]$. A conditional probability of an event E given events $F_1, \dots, F_n$ in this order is denoted by $\Pr[E|F_1, \dots, F_n]$. A probability distribution which dominates a random variable X is denoted by $dist\big(X\big)$. A probability distribution which dominates a random variable X whose probability is a joint probability of random variables $Y_1, \dots, Y_n, X$ is denoted by $dist\big(X|Y_1, \dots, Y_n, X\big)$. A probability P is said to be negligible in λ if for any given positive polynomial $\mathrm{poly}(\lambda)$ $P < 1/\mathrm{poly}(\lambda)$ for sufficiently large $\lambda \in \mathbb{N}$. Two probabilities P and Q are said to be computationally indistinguishable if $|P - Q|$ is negligible in λ, which is denoted as $P \approx_{\mathrm{c}} Q$. A probability P is said to be overwhelming if $|1 - P|$ is negligible in λ.

2.1 Interactive Argument System, Σ-protocol and Witness-Indistinguishability

Suppose that there exists a predicate Φ that defines the membership of a binary relation R; i.e., Φ maps $(x, w) \in (\{0,1\}^*)^2$ to TRUE or FALSE. The relation R is defined as $R \stackrel{\mathrm{def}}{=} \{(x, w) \in (\{0,1\}^*)^2 | \Phi(x, w) = \text{TRUE}\}$. We consider relations parametrized by the security parameter λ. That is, R in our sense is a family $(R_\lambda)_{\lambda \in \mathbb{N}}$ of relations $R_\lambda \subseteq (\{0,1\}^*)^2$. We say that R is polynomially bounded if there exist a constant c and a polynomial $\ell(\cdot)$ such that for any λ, $|x| \leq c \cdot \lambda$ and $|w| \leq \ell(|x|)$ for any $(x, w) \in R_\lambda$. We say that R is an NP relation if R is polynomially bounded and Φ is computable within polynomial-time in $|x|$ as an algorithm. For a pair $(x, w) \in R$ we call x a statement and w a witness of x. We call R the witness relation, and $\Phi(\cdot, \cdot)$ the predicate of the witness relation R. An NP language L for an NP relation R is defined as the set of all possible statements: $L \stackrel{\mathrm{def}}{=} \{x \in \{0,1\}^*; \exists w \in \{0,1\}^*, (x, w) \in R\}$. We denote the set of witnesses of a statement x by $W(x)$: $W(x) \stackrel{\mathrm{def}}{=} \{w \in \{0,1\}^* \mid (x, w) \in R\}$. We call the union W of all the sets $W(x)$ for $x \in L$ the *witness space* of L: $W \stackrel{\mathrm{def}}{=} \bigcup_{x \in L} W(x)$. We denote an interactive proof system on an NP relation R [Bab85,GMR85] by $\Pi = (\Pi.\mathsf{Setup}, \mathsf{P}, \mathsf{V})$, where $\Pi.\mathsf{Setup}$ is a set up algorithm for a set pp of public parameters, and P and V are a pair of interactive algorithms. P called a prover is probabilistic and unbounded, and V called a verifier is probabilistic polynomial-time (PPT). If P is also limited to PPT, then Π is called an interactive *argument* system.

Σ-protocol [Cra96,Dam10] Let R be an NP relation. A *Σ-protocol* Σ on the relation R is a 3-move public-coin protocol of an interactive argument system $\Pi = (\Pi.\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ [Cra96,Dam10]. We introduce six PPT algorithms for a Σ-protocol: $\Sigma = (\Sigma_{\mathrm{com}}, \Sigma_{\mathrm{cha}}, \Sigma_{\mathrm{res}}, \Sigma_{\mathrm{vrf}}, \Sigma_{\mathrm{ext}}, \Sigma_{\mathrm{sim}})$. The first algorithm Σ_{com} is executed by P. On input a pair of a statement and a witness $(x, w) \in R$, it generates a commitment message

COM and outputs its inner state St. It returns them as $\Sigma_{\rm com}(x, w) \to (\text{COM}, St)$. The second algorithm $\Sigma_{\rm cha}$ is executed by V. On input the statement x, it reads out the size of the security parameter as 1^λ and chooses a challenge message $\text{CHA} \in_R \text{CHASP}(1^\lambda)$ from the challenge space $\text{CHASP}(1^\lambda) := \{0,1\}^{\omega(\lambda)}$, where $\omega(\cdot)$ is a super-log function [BP02]. It returns the message as $\Sigma_{\rm cha}(x) \to \text{CHA}$. The third algorithm $\Sigma_{\rm res}$ is executed by P. On input the state St and the challenge message CHA, it generates a response message RES. It returns the message as $\Sigma_{\rm res}(St, \text{CHA}) \to \text{RES}$. The fourth algorithm $\Sigma_{\rm vrf}$ is executed by V. On input the statement x and the messages COM, CHA and RES, it computes a boolean decision d. It returns the decision as $\Sigma_{\rm vrf}(x, \text{COM}, \text{CHA}, \text{RES}) \to d$. If $d = 1$, then we say that P is accepted by V on x. Otherwise, we say that P is rejected by V on x. The vector of all the messages $(\text{COM}, \text{CHA}, \text{RES})$ is called a transcript of the interaction on x.

These four algorithms $(\Sigma_{\rm com}, \Sigma_{\rm cha}, \Sigma_{\rm res}, \Sigma_{\rm vrf})$ must satisfy the following property.

Completeness For any $(x, w) \in R$, a prover $\mathsf{P}(x, w)$ has a verifier $\mathsf{V}(x)$ accept with probability 1: $\Pr[\Sigma_{\rm vrf}(x, \text{COM}, \text{CHA}, \text{RES}) = 1 \mid \Sigma_{\rm com}(x, w) \to (\text{COM}, St), \Sigma_{\rm cha}(x) \to \text{CHA}, \Sigma_{\rm res}(St, \text{CHA}) \to \text{RES}]$.

The fifth algorithm $\Sigma_{\rm ext}$ concerns with the following property.

Special Soundness There is a PPT algorithm $\Sigma_{\rm ext}$ called a *knowledge extractor*, which, on input a statement x and two accepting transcripts $(\text{COM}, \text{CHA}, \text{RES})$ and $(\text{COM}, \text{CHA}', \text{RES}')$, $\text{CHA} \neq \text{CHA}'$, computes a witness $\hat{w}$ satisfying $(x, \hat{w}) \in R$ with an overwhelming probability in $|x|$:

$$\hat{w} \leftarrow \Sigma_{\rm ext}(x, \text{COM}, \text{CHA}, \text{RES}, \text{CHA}', \text{RES}'). \tag{1}$$

Note here that commitment messages are common and challenge messages are different.

The sixth algorithm $\Sigma_{\rm sim}$ concerns with the following property.

Honest-Verifier Zero-Knowledge There is a PPT algorithm called a *simulator* $\Sigma_{\rm sim}$, which, on input a statement x, computes an accepting transcript on x:

$$(\tilde{\text{COM}}, \tilde{\text{CHA}}, \tilde{\text{RES}}) \leftarrow \Sigma_{\rm sim}(x), \tag{2}$$

where the distribution of the simulated transcripts $\mathit{dist}\big(\tilde{\text{COM}}, \tilde{\text{CHA}}, \tilde{\text{RES}}\big)$ is identical to the distribution of the real accepting transcripts $\mathit{dist}\big(\text{COM}, \text{CHA}, \text{RES}\big)$.

Note 1: Simulator Input In a Σ-protocol the challenge message CHA is a *public* coin. This property enables us in this paper to use the following variant of the simulator $\Sigma_{\rm sim}(x)$: On input a simulated challenge message $\tilde{\text{CHA}}$ that is chosen uniformly at random, the variant generates a commitment $\tilde{\text{COM}}$ and a response $\tilde{\text{RES}}$:

$$\tilde{\text{CHA}} \in_R \text{CHASP}(1^\lambda), \quad (\tilde{\text{COM}}, \tilde{\text{RES}}) \leftarrow \Sigma_{\rm sim}(x, \tilde{\text{CHA}}). \tag{3}$$

Witness-Indistinguishability [FS90,Gol01] Let R be an NP relation. Suppose that an interactive argument system $\Pi = (\Pi.\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ with a Σ-protocol Σ on the relation R is given. In this paper we focus on the following property.

Perfect Witness Indistinguishability For any PPT algorithm V^*, any sequences of witnesses $\mathbf{w} = (w_x)_{x \in L}$ and $\mathbf{w}' = (w'_x)_{x \in L}$ s.t. $w_x, w'_x \in W(x)$, any string $x \in L$ and any string $z \in \{0,1\}^*$, the two distributions $\mathit{dist}\big(x, z, \mathit{transc}\langle \mathsf{P}(x, w_x), \mathsf{V}^*(x, z)\rangle\big)$ and $\mathit{dist}\big(x, z, \mathit{transc}\langle \mathsf{P}(x, w'_x), \mathsf{V}^*(x, z)\rangle\big)$ are identical.

2.2 Commit-and-Prove Scheme [CLOS02,EG14]

A commit-and-prove scheme CmtPrv consists of five PPT algorithms: $\mathsf{CmtPrv} = (\mathsf{CmtPrv.Setup}, \mathsf{Cmt} = (\mathsf{Cmt.Com}, \mathsf{Cmt.Vrf}), \Pi = (\mathsf{P}, \mathsf{V}))$.

$\mathsf{CmtPrv.Setup}(1^\lambda) \to pp$. On input the security parameter 1^λ, it generates a set of public parameter pp. It returns pp.

$\mathsf{Cmt.Com}(m) \to (c, \kappa)$. On input a message m in the message space $\mathcal{M}sg(1^\lambda)$, this PPT algorithm generates a commitment c. It also generates an opening key κ. It returns (c, κ).

$\mathsf{Cmt.Vrf}(c, m, \kappa) \to d$. On input a commitment c, a message m and an opening key κ, this deterministic polynomial-time algorithm generates a boolean decision d. It returns d.

The correctness should hold for the commitment part Cmt of the scheme: For any security parameter 1^λ, any set of public parameter pp and any message $m \in \mathcal{M}sg(1^\lambda)$, $\Pr[d = 1 \mid (c, \kappa) \leftarrow \mathsf{Cmt.Com}(m), d \leftarrow \mathsf{Cmt.Vrf}(c, m, \kappa)] = 1$.

We denote by Φ a predicate that returns the boolean decision: $\Phi(c,(m,\kappa)) \stackrel{\text{def}}{=} (\mathsf{Cmt.Vrf}(c, m, \kappa))$. In the scheme there is an interactive argument system $\Pi = (\mathsf{P}, \mathsf{V})$ for the following relation R:

$$R := \{(c,(m,\kappa)) \in \{0,1\}^* \times (\{0,1\}^*)^2 \mid \Phi(c,(m,\kappa)) = \text{TRUE}\}. \tag{4}$$

In this paper we focus on the following properties for the commitment part Cmt.
Perfectly Hiding For any security parameter 1^λ, any set of public parameter pp and any two messages $m, m' \in \mathcal{M}sg(1^\lambda)$, the two distributions $dist\big(c \mid (c,\kappa) \leftarrow \mathsf{Cmt.Com}(m)\big)$ and $dist\big(c \mid (c,\kappa) \leftarrow \mathsf{Cmt.Com}(m')\big)$ are identical.
Computationally Binding The attack of breaking binding property of Cmt by an algorithm $\mathbf{A}$ is defined by the following experiment.

$$\mathsf{Exp}^{\text{bind}}_{\mathsf{Cmt},\mathbf{A}}(1^\lambda): \tag{5}$$
$$pp \leftarrow \mathsf{CmtPrv.Setup}(1^\lambda), (c, m, \kappa, m', \kappa') \leftarrow \mathbf{A}(pp) \tag{6}$$
$$\text{If } \mathsf{Cmt.Vrf}(c, m, \kappa) = \mathsf{Cmt.Vrf}(c, m', \kappa') = 1 \wedge m \neq m', \text{then Return } \textsc{Win} \text{ else Return } \textsc{Lose} \tag{7}$$

The advantage of $\mathbf{A}$ over Cmt is defined as $\mathbf{Adv}^{\text{bind}}_{\mathsf{Cmt},\mathbf{A}}(\lambda) := \Pr[\mathsf{Exp}^{\text{bind}}_{\mathsf{Cmt},\mathbf{A}}(1^\lambda) \text{ returns } \textsc{Win}]$. The commitment scheme Cmt is said to be *computationally binding* if for any set of public parameter pp and any PPT algorithm $\mathbf{A}$, the advantage $\mathbf{Adv}^{\text{bind}}_{\mathsf{Cmt},\mathbf{A}}(\lambda)$ is negligible in λ.
Note 2: Opening Key The commitment generation algorithm $\mathsf{Cmt.Com}$ uses random tapes [Gol01]. In this paper we are in the case that a randomness $r \in \{0,1\}^\lambda$ is used to generate a commitment c, and the opening key κ is the randomness: $\kappa := r$. That is, $\mathsf{Cmt.Com}(m; r) \rightarrow (c, r)$.

2.3 Digital Signature Scheme [FS86]

A digital signature scheme Sig consists of four PPT algorithms: $\mathsf{Sig} = (\mathsf{Sig.Setup}, \mathsf{Sig.KG}, \mathsf{Sig.Sign}, \mathsf{Sig.Vrf})$.
$\mathsf{Sig.Setup}(1^\lambda) \rightarrow pp$. On input the security parameter 1^λ, it generates a set of public parameter pp. It returns pp.
$\mathsf{Sig.KG}(1^\lambda) \rightarrow (\mathrm{PK}, \mathrm{SK})$. On input the security parameter 1^λ, this PPT algorithm generates a signing key SK and the corresponding public key PK. It returns $(\mathrm{PK}, \mathrm{SK})$.
$\mathsf{Sig.Sign}(\mathrm{PK}, \mathrm{SK}, m) \rightarrow \sigma$. On input the public key PK, the secret key SK and a message m in the message space $\mathcal{M}sg(1^\lambda)$, this PPT algorithm generates a signature σ. It returns σ.
$\mathsf{Sig.Vrf}(\mathrm{PK}, m, \sigma) \rightarrow d$. On input the public key PK, a message m and a signature σ, it returns a boolean d.

The correctness should hold for the scheme Sig: For any security parameter 1^λ and any message $m \in \mathcal{M}sg(1^\lambda)$, $\Pr[d = 1 \mid pp \leftarrow \mathsf{Sig.Setup}(1^\lambda), (\mathrm{PK}, \mathrm{SK}) \leftarrow \mathsf{Sig.KG}(1^\lambda), \sigma \leftarrow \mathsf{Sig.Sign}(\mathrm{PK}, \mathrm{SK}, m), d \leftarrow \mathsf{Sig.Vrf}(\mathrm{PK}, m, \sigma)] = 1$.

An adaptive chosen-message attack on the scheme Sig by a forger algorithm $\mathbf{F}$ is defined by the following experiment.

$$\mathsf{Exp}^{\text{euf-cma}}_{\mathsf{Sig},\mathbf{F}}(1^\lambda): \tag{8}$$
$$pp \leftarrow \mathsf{Sig.Setup}(1^\lambda), (\mathrm{PK}, \mathrm{SK}) \leftarrow \mathsf{Sig.KG}(1^\lambda),\ (m^*, \sigma^*) \leftarrow \mathbf{F}^{\mathbf{SignO}(\mathrm{PK},\mathrm{SK},\cdot)}(\mathrm{PK}) \tag{9}$$
$$\text{If } m^* \notin \{m_j\}_{1 \le j \le q_{\mathrm{s}}} \text{ and } \mathsf{Sig.Vrf}(\mathrm{PK}, m^*, \sigma^*) = 1, \text{then Return } \textsc{Win} \text{ else Return } \textsc{Lose} \tag{10}$$

In the experiment, $\mathbf{F}$ issues a signing query to its signing oracle $\mathbf{SignO}(\mathrm{PK}, \mathrm{SK}, \cdot)$ by sending a message m_j at most q_{s} times ($1 \le j \le q_{\mathrm{s}}$). As a reply, $\mathbf{F}$ receives a valid signature σ_j on m_j. After receiving replies, $\mathbf{F}$ returns a message and a signature (m^*, σ^*). A restriction is imposed on the algorithm $\mathbf{F}$: The set of queried messages $\{m_j\}_{1 \le j \le q_{\mathrm{s}}}$ should not contain the message m^*. The advantage of $\mathbf{F}$ over Sig is defined as $\mathbf{Adv}^{\text{euf-cma}}_{\mathsf{Sig},\mathbf{F}}(\lambda) := \Pr[\mathsf{Exp}^{\text{euf-cma}}_{\mathsf{Sig},\mathbf{F}}(1^\lambda) \text{ returns } \textsc{Win}]$. The digital signature scheme Sig is said to be *existentially unforgeable against adaptive chosen-message attacks* if for any given PPT algorithm $\mathbf{F}$, the advantage $\mathbf{Adv}^{\text{euf-cma}}_{\mathsf{Sig},\mathbf{F}}(\lambda)$ is negligible in λ.

3 Witness-Indistinguishable Argument with Σ-Protocol for Bundled Witness Space

In this section, we propose a generic construction of an interactive argument system that is a witness-indistinguishable argument system for a newly introduced *bundled witness space*. Our protocol of the interactive argument system is an AND-composition of Σ-protocols together with a commitment scheme, which is to prove the knowledge of witness pairs each of which consists of two components; one is a common component (such as a global identity string) and the other is an individual component (such as a digital signature issued by an individual authority on the global identity). We prove that our protocol is certainly a Σ-protocol. Finally, we prove that our interactive argument system with the protocol is perfectly witness-indistinguishable under the condition that the employed commitment scheme is perfectly hiding and the component Σ-protocols are perfectly witness-indistinguishable.

3.1 Building Blocks

Component Interactive Argument Systems with Σ-protocols For a polynomially bounded integer n, let A be the set of indices: $A := \{1, \ldots, n\}$. We start with an efficiently computable predicate Φ^a for each $a \in A$, which determines an NP witness relation R^a:

$$R^a = \{(x^a, w^a) \in \{0,1\}^* \times \{0,1\}^* \mid \Phi^a(x^a, w^a) = \text{TRUE}\}, a \in A. \tag{11}$$

We suppose for each $a \in A$ that there is an interactive argument system $\Pi^a = (\Pi.\mathsf{Setup}, \mathsf{P}^a, \mathsf{V}^a)$ which is executed in accordance with a Σ-protocol for the relation R^a:

$$\Sigma^a = (\Sigma^a_{\text{com}}, \Sigma^a_{\text{cha}}, \Sigma^a_{\text{res}}, \Sigma^a_{\text{vrf}}, \Sigma^a_{\text{ext}}, \Sigma^a_{\text{sim}}). \tag{12}$$

We suppose further that the witness space W^a decomposes into two components $W^a = W^a_0 \times W^a_1$ for each $a \in A$. In this paper, our interest is in the case that *all the 0-th spaces $W^a_0, a \in A$, are equal*, which we denote by W_0. We call the common set W_0 the *base witness space* of the witness spaces W^a for $a \in A$, and an element $w_0 \in W_0$ a *base witness point*. Then, we suppose that a witness $w^a \in W^a$ consists of w_0 and w^a_1, where the base witness point w_0 is *common* for all $a \in A$. That is, we will study the following type of *bundled witnesses* $(w^a)^{a \in A}$;

$$\begin{array}{ll} W^a = W_0 \times W^a_1, & \\ \quad\cup\!\!\!\!\!\!\!\! \quad\quad\quad \cup & a \in A. \\ w^a = (\exists w_0, \exists w^a_1), & \end{array} \tag{13}$$

Commit-and-Prove Scheme with Σ-protocol To construct an interactive argument system for the relations $(R^a)^{a \in A}$ with the base witness space W_0, we employ a commit-and-prove scheme with a Σ-protocol: $\mathsf{CmtPrv} = (\mathsf{CmtPrv.Setup}, \mathsf{Cmt} = (\mathsf{Cmt.Com}, \mathsf{Cmt.Vrf}), \Pi_0 = (\mathsf{P}_0, \mathsf{V}_0))$, where the predicate Φ_0 and the relation R_0 is defined as follows, and Π_0 is executed in accordance with a Σ-protocol Σ_0:

$$\Phi_0(c_0, (w_0, r_0)) \stackrel{\text{def}}{=} (\mathsf{Cmt.Com}(w_0; r_0) =_? (c_0, r_0)), \tag{14}$$

$$R_0 \stackrel{\text{def}}{=} \{(c_0, (w_0, r_0)) \in \{0,1\}^* \times (\{0,1\}^*)^2 \mid \Phi_0(c_0, (w_0, r_0)) = \text{TRUE}\}, \tag{15}$$

$$\Sigma_0 = (\Sigma_{0,\text{com}}, \Sigma_{0,\text{cha}}, \Sigma_{0,\text{res}}, \Sigma_{0,\text{vrf}}, \Sigma_{0,\text{ext}}, \Sigma_{0,\text{sim}}). \tag{16}$$

Note that a message m to be committed is a base witness point w_0.

3.2 On the Construction of a Σ-protocol for Simultaneous Satisfiability

We introduce for each $a \in A$ the following composed relation determined by the two predicates Φ^a and Φ_0. That is, the relation R^a_0 is for *simultaneous satisfiability* of the two predicates Φ^a and Φ_0 on the base witness point w_0:

$$R^a_0 := \left\{ (x^a_0 = (x^a, c_0), w^a_0 = (w_0, w^a_1, r_0)) \mid \begin{cases} \Phi^a(x^a, (w_0, w^a_1)) = \text{TRUE and} \\ \Phi_0(c_0, (w_0, r_0)) = \text{TRUE} \end{cases} \right\}, \ a \in A. \tag{17}$$

We *require* here that the Σ-protocols Σ^a and Σ_0 are turned into a *simultaneous* Σ-protocol Σ_0^a of an interactive argument system $\Pi_0^a = (\Pi.\mathsf{Setup}, \mathsf{CmtPrv.Setup}, \mathsf{P}_0^a, \mathsf{V}_0^a)$ for the above relation R_0^a:

$$\Sigma_0^a = (\Sigma_{0,\mathrm{com}}^a, \Sigma_{0,\mathrm{cha}}^a, \Sigma_{0,\mathrm{res}}^a, \Sigma_{0,\mathrm{vrf}}^a, \Sigma_{0,\mathrm{ext}}^a, \Sigma_{0,\mathrm{sim}}^a). \tag{18}$$

- $\Sigma_{0,\mathrm{com}}^a(x_0^a, w_0^a) \rightarrow (\mathrm{COM}^a, \mathrm{COM}_0^a, \overline{St_0^a})$. This PPT algorithm is executed by P_0^a. On input a statement $x_0^a = (x^a, c_0)$ and a witness $w_0^a = (w_0, w_1^a, r_0)$, it executes the algorithms $\Sigma_{\mathrm{com}}^a(x^a, (w_0, w_1^a))$ and $\Sigma_{0,\mathrm{com}}(c_0, (w_0, r_0))$. It obtains the commitment messages and the inner states, (COM^a, St^a) and $(\mathrm{COM}_0^a, St_0^a)$, respectively. There is a constraint that the knowledge extractor $\Sigma_{0,\mathrm{ext}}^a$ should return a witness which simultaneously satisfies the two predicates Φ^a and Φ_0 on the base witness point w_0. It sets the state as $\overline{St_0^a} := (St^a, St_0^a)$. It returns $(\mathrm{COM}^a, \mathrm{COM}_0^a, \overline{St_0^a})$. P_0^a sends $(\mathrm{COM}^a, \mathrm{COM}_0^a)$ to V_0^a as a commitment message, and keeps the state $\overline{St_0^a}$.
- $\Sigma_{0,\mathrm{cha}}^a(x_0^a) \rightarrow \mathrm{CHA}$. This PPT algorithm is executed by V_0^a. On input the statement x_0^a, it reads out the size of the security parameter as 1^λ and chooses a challenge message $\mathrm{CHA} \in_R \mathrm{CHASP}(1^\lambda)$. It returns CHA. V_0^a sends CHA to P_0^a as a challenge message.
- $\Sigma_{0,\mathrm{res}}^a(\overline{St_0^a}, \mathrm{CHA}) \rightarrow (\mathrm{RES}^a, \mathrm{RES}_0^a)$. This PPT algorithm is executed by P_0^a. On input the state $\overline{St_0^a}$ and the challenge message CHA, it executes the algorithms $\Sigma_{\mathrm{res}}^a(St^a, \mathrm{CHA})$ and $\Sigma_{0,\mathrm{res}}(St_0^a, \mathrm{CHA})$. It obtains the response messages RES^a and RES_0^a, respectively. There is the constraint that the knowledge extractor $\Sigma_{0,\mathrm{ext}}^a$ should return a witness which simultaneously satisfies Φ^a and Φ_0 on w_0. It returns $(\mathrm{RES}^a, \mathrm{RES}_0^a)$. P_0^a sends $(\mathrm{RES}^a, \mathrm{RES}_0^a)$ to V_0^a as a response message.
- $\Sigma_{0,\mathrm{vrf}}^a(x_0^a, (\mathrm{COM}^a, \mathrm{COM}_0^a), \mathrm{CHA}, (\mathrm{RES}^a, \mathrm{RES}_0^a)) \rightarrow d$. This deterministic polynomial-time algorithm is executed by V_0^a. On input the statement $x_0^a = (x^a, c_0)$ and all the messages $(\mathrm{COM}^a, \mathrm{COM}_0^a)$, CHA and $(\mathrm{RES}^a, \mathrm{RES}_0^a)$, it executes the algorithms $\Sigma_{\mathrm{vrf}}^a(x^a, \mathrm{COM}^a, \mathrm{CHA}, \mathrm{RES}^a)$ and $\Sigma_{0,\mathrm{vrf}}(c_0, \mathrm{COM}_0^a, \mathrm{CHA}, \mathrm{RES}_0^a)$. It obtains two boolean decisions d^a and d_0^a. If the both d^a and d_0^a are 1, then it returns $d := 1$, and otherwise $d := 0$. V_0^a returns d as the decision of the interactive protocol on x_0^a.
- $\Sigma_{0,\mathrm{ext}}^a(x_0^a, (\mathrm{COM}^a, \mathrm{COM}_0^a), \mathrm{CHA}, (\mathrm{RES}^a, \mathrm{RES}_0^a), \mathrm{CHA}', (\mathrm{RES}^{a\prime}, \mathrm{RES}_0^{a\prime})) \rightarrow (\hat{w}_0^a, \hat{w}_1^a, \hat{r}_0^a)$. This PPT algorithm is for knowledge extraction. On input the statement $x_0^a = (x^a, c_0)$ and two accepting transcripts $((\mathrm{COM}^a, \mathrm{COM}_0^a), \mathrm{CHA}, (\mathrm{RES}^a, \mathrm{RES}_0^a))$ and $((\mathrm{COM}^a, \mathrm{COM}_0^a), \mathrm{CHA}', (\mathrm{RES}^{a\prime}, \mathrm{RES}_0^{a\prime}))$, $\mathrm{CHA} \neq \mathrm{CHA}'$, it executes the algorithms $\Sigma_{\mathrm{ext}}^a(x^a, \mathrm{COM}^a, \mathrm{CHA}, \mathrm{RES}^a, \mathrm{CHA}', \mathrm{RES}^{a\prime})$ and $\Sigma_{0,\mathrm{ext}}(c_0, \mathrm{COM}_0^a, \mathrm{CHA}, \mathrm{RES}_0^a, \mathrm{CHA}', \mathrm{RES}_0^{a\prime})$. It obtains witnesses $(\hat{w}_0^a, \hat{w}_1^a)$ and $(\bar{w}_0^a, \hat{r}_0^a)$ which satisfy $(x^a, (\hat{w}_0^a, \hat{w}_1^a)) \in R^a$ and $(c_0, (\bar{w}_0^a, \hat{r}_0^a)) \in R_0$ with an overwhelming probability in $|x^a|$ and $|c_0|$, respectively. Note here that the commitment messages are common and the challenge messages are different. The simultaneous satisfiability on w_0 *must assure* the following equality:

$$\hat{w}_0^a = \bar{w}_0^a \text{ with probability one.} \tag{19}$$

It returns $(\hat{w}_0^a, \hat{w}_1^a, \hat{r}_0^a)$.
- $\Sigma_{0,\mathrm{sim}}^a(x_0^a, \tilde{\mathrm{CHA}}) \rightarrow ((\tilde{\mathrm{COM}}^a, \tilde{\mathrm{COM}}_0^a), (\tilde{\mathrm{RES}}^a, \tilde{\mathrm{RES}}_0^a))$. This PPT algorithm is for the simulation of an accepting transcript. On input a statement $x_0^a = (x^a, c_0)$ and a uniform random string $\tilde{\mathrm{CHA}} \in_R \mathrm{CHASP}(1^\lambda)$, it executes the algorithms $\Sigma_{\mathrm{sim}}^a(x^a, \tilde{\mathrm{CHA}})$ and $\Sigma_{0,\mathrm{sim}}(c_0, \tilde{\mathrm{CHA}})$. It obtains the remaining part of the transcripts $(\tilde{\mathrm{COM}}^a, \tilde{\mathrm{RES}}^a)$ and $(\tilde{\mathrm{COM}}_0^a, \tilde{\mathrm{RES}}_0^a)$, respectively. The simulated messages $((\tilde{\mathrm{COM}}^a, \tilde{\mathrm{COM}}_0^a), \tilde{\mathrm{CHA}}, (\tilde{\mathrm{RES}}^a, \tilde{\mathrm{RES}}_0^a))$ should form a distribution $dist\big((\tilde{\mathrm{COM}}^a, \tilde{\mathrm{COM}}_0^a), \tilde{\mathrm{CHA}}, (\tilde{\mathrm{RES}}^a, \tilde{\mathrm{RES}}_0^a) \mid \text{generated by } \mathrm{CHASP}(1^\lambda) \text{ and } \Sigma_{0,\mathrm{sim}}^a(x_0^a, \tilde{\mathrm{CHA}})\big)$ which is identical to the distribution $dist\big((\mathrm{COM}^a, \mathrm{COM}_0^a), \mathrm{CHA}, (\mathrm{RES}^a, \mathrm{RES}_0^a) \mid \text{real accepting transcript}\big)$.

Remark To construct the algorithm $\Sigma_{0,\mathrm{com}}^a$ of commitment message and the algorithm $\Sigma_{0,\mathrm{res}}^a$ of response message is a non-trivial task. That is, we have to construct $\Sigma_{0,\mathrm{com}}^a$ and $\Sigma_{0,\mathrm{res}}^a$ so that the knowledge extractor $\Sigma_{0,\mathrm{ext}}^a$ returns a witness which *simultaneously* satisfies Φ^a and Φ_0 on a base witness point w_0. The idea of the construction is to use a common random tape to generate commitment messages COM^a and COM_0^a, but we do not describe the inner treatment of the random tapes in $\Sigma_{0,\mathrm{com}}^a$ and $\Sigma_{0,\mathrm{res}}^a$ for generality. Hence our approach is to show the construction when we instantiate the Σ-protocol Σ_0^a. In Section A we actually demonstrate the construction of Σ_0^a in an algebraic setting.

3.3 Bundled Witness Space

We now introduce an NP witness relation for our *bundled witness spaces*. We first fix the base witness point w_0 in the base witness space W_0 and consider a subset $R^a_{w_0}$ for each NP witness relation $R^a, a \in A$:

$$R^a_{w_0} := \{(x^a, w^a) \in R^a \mid w^a = (w_0, w^a_1) \text{ for some } w^a_1\} \subset R^a,\ a \in A. \tag{20}$$

Then we run the base witness point w_0 to claim the following property.

Claim 1 *For a polynomially bounded integer n, let A be the set of indices $\{1, \ldots, n\}$. Then we have:*

$$\bigcup_{w_0 \in W_0} \Big(\prod_{a \in A} R^a_{w_0} \Big) \subset \prod_{a \in A} \Big(\bigcup_{w_0 \in W_0} R^a_{w_0} \Big) = \prod_{a \in A} R^a. \tag{21}$$

Proof. The equality of the right-hand side is because $\bigcup_{w_0 \in W_0} R^a_{w_0} = R^a$. An element of the left hand side is of the form $(x^1, (w_0, w^1_1)), \ldots, (x^n, (w_0, w^n_1))$ where $w_0 \in W_0$ and $(x^a, (w_0, w^a_0)) \in R^a$ for each $a \in A$. This is an element of $\prod_{a \in A} R^a$, and hence the inclusion follows. □

Deleting the redundancy, we obtain the following one-to-one correspondence as sets ('$\simeq$'):

$$R^{a \in A}_{\text{bnd}} \stackrel{\text{def}}{=} \{\big((x^a)^{a \in A}, w_0, (w^a_1)^{a \in A}\big) \in \{0,1\}^* \times (\{0,1\}^*)^2 \mid (x^a, (w_0, w^a_1)) \in R^a,\ a \in A\} \tag{22}$$

$$\simeq \bigcup_{w_0 \in W_0} \Big(\prod_{a \in A} R^a_{w_0} \Big). \tag{23}$$

Claim 2 *For a polynomially bounded integer n, let A be the set of indices $\{1, \ldots, n\}$. Then the relation $R^{a \in A}_{bnd}$ is an NP relation.*

Proof. We first note that the number of indices $|A|$ is polynomially bounded. To bound the bit lengths of witnesses by a fixed polynomial, let $\text{poly}^a(\cdot)$ denote for each $a \in A$ the polynomial which bounds the bit lengths of witnesses: $|w^a| < \text{poly}^a(|x^a|)$ for $(x^a, w^a) \in R^a$. Let a polynomial $\text{poly}(\cdot)$ be the sum: $\text{poly}(\cdot) := \sum_{a \in A} \text{poly}^a(\cdot)$. Then $\text{poly}(\cdot)$ bounds the bit length of the witness as

$$|w_0, (w^a_1)^{a \in A}| \le |(w_0, w^a_1)^{a \in A}| = |(w^a)^{a \in A}| \le \sum_{a \in A} \text{poly}^a(|x^a|) \le \sum_{a \in A} \text{poly}^a(|(x^a)^{a \in A}|) = \text{poly}(|(x^a)^{a \in A}|). \tag{24}$$

As for efficiency of deciding the membership of the relation $R^{a \in A}_{\text{bnd}}$, we just remember that the number of indices $|A|$ is polynomially bounded. □

Definition 1 (Relation for Bundled Witness Spaces) *For a polynomially bounded integer n, an NP witness relation for the bundled witness spaces is defined as $R^{a \in A}_{bnd}$.*

Definition 2 (Bundled Witness Spaces) *For a polynomially bounded integer n, let A be the set of indices $\{1, \ldots, n\}$. Let $R^a, a \in A$ be NP witness relations where each witness space decomposes $W^a = W_0 \times W^a_1, a \in A$. Then the bundled witness spaces is defined as follows.*

$$W^{a \in A}_{bnd} \stackrel{def}{=} W_0 \times (W^a_1)^{a \in A}. \tag{25}$$

3.4 Generic Construction of a Σ-protocol for the Bundled Witness Space

By using the above Σ-protocols $(\Sigma^a_0)^{a \in A}$ and a commitment generation algorithm $\mathsf{Cmt.Com}$, we construct an interactive argument system $\Pi^{a \in A}_{\text{bnd}} = (\mathsf{P}, \mathsf{V})$ for the witness relation $R^{a \in A}_{\text{bnd}}$ with a protocol $\Sigma^{a \in A}_{\text{bnd}}$. $\Sigma^{a \in A}_{\text{bnd}}$ is actually a Σ-protocol, which consists of the six PPT algorithms described below (see also Fig.1):

$$\Sigma^{a \in A}_{\text{bnd}} = (\Sigma^{a \in A}_{\text{bnd,com}}, \Sigma^{a \in A}_{\text{bnd,cha}}, \Sigma^{a \in A}_{\text{bnd,res}}, \Sigma^{a \in A}_{\text{bnd,vrf}}, \Sigma^{a \in A}_{\text{bnd,ext}}, \Sigma^{a \in A}_{\text{bnd,sim}}). \tag{26}$$

• $\Sigma^{a\in A}_{\mathrm{bnd,com}}((x^a)^{a\in A}, (w_0, (w^a_1)^{a\in A})) \to (c_0, (\mathrm{COM}^a, \mathrm{COM}^a_0)^{a\in A}, St)$. This PPT algorithm is executed by P. On input a statement that is a vector $(x^a)^{a\in A}$ and a witness that is a vector $(w_0, (w^a_1)^{a\in A})$, it computes a commitment c_0 to the base witness point w_0 with a randomness $r_0 \in_R \{0,1\}^\lambda$ by running the commitment generation algorithm of Cmt: $(c_0, r_0) \leftarrow \mathsf{Cmt.Com}(w_0; r_0)$. It sets the extended statement as $x^a_0 := (x^a, c_0)$ and the extended witness as $w^a_0 := (w_0, w^a_1, r_0)$ for each $a \in A$. It executes for each $a \in A$ the algorithm $\Sigma^a_{0,\mathrm{com}}(x^a_0, w^a_0)$. It obtains $(\mathrm{COM}^a, \mathrm{COM}^a_0, \overline{St^a_0})$. It sets the state as $St := (\overline{St^a_0})^{a\in A}$. It returns $(c_0, (\mathrm{COM}^a, \mathrm{COM}^a_0)^{a\in A}, St)$. P sends $(c_0, (\mathrm{COM}^a, \mathrm{COM}^a_0)^{a\in A})$ to V as a commitment message, and keeps the state St.
• $\Sigma^{a\in A}_{\mathrm{bnd,cha}}((x^a)^{a\in A}) \to \mathrm{CHA}$. This PPT algorithm is executed by V. On input the statement $(x^a)^{a\in A}$, it reads out the size of the security parameter as 1^λ and chooses a challenge message $\mathrm{CHA} \in_R \mathrm{CHASP}(1^\lambda)$. It returns CHA. V^a_0 sends CHA to P^a_0 as a challenge message.
• $\Sigma^{a\in A}_{\mathrm{bnd,res}}(St, \mathrm{CHA}) \to (\mathrm{RES}^a, \mathrm{RES}^a_0)^{a\in A}$. This PPT algorithm is executed by P. On input the state St and the challenge message CHA, it executes for each $a \in A$ the algorithm $\Sigma^a_{0,\mathrm{res}}(\overline{St^a_0}, \mathrm{CHA})$. It obtains $(\mathrm{RES}^a, \mathrm{RES}^a_0)$. It returns $(\mathrm{RES}^a, \mathrm{RES}^a_0)$. P sends $(\mathrm{RES}^a, \mathrm{RES}^a_0)^{a\in A}$ to V as a response message.
• $\Sigma^{a\in A}_{\mathrm{bnd,vrf}}((x^a)^{a\in A}) \to d$. This deterministic polynomial-time algorithm is executed by V. On input the statement $(x^a)^{a\in A}$ and all the messages $(c_0, (\mathrm{COM}^a, \mathrm{COM}^a_0)^{a\in A})$, CHA and $(\mathrm{RES}^a, \mathrm{RES}^a_0)^{a\in A}$, it first sets the extended statement as $x^a_0 := (x^a, c_0)$ for each $a \in A$. Then it executes for each $a \in A$ the algorithm $\Sigma^a_{0,\mathrm{vrf}}(x^a_0, \mathrm{COM}^a, \mathrm{COM}^a_0, \mathrm{CHA}, \mathrm{RES}^a, \mathrm{RES}^a_0)$. It obtains boolean decisions. If all the decisions are 1, then V returns 1, and otherwise, 0.

These four algorithms $(\Sigma^{a\in A}_{\mathrm{bnd,com}}, \Sigma^{a\in A}_{\mathrm{bnd,cha}}, \Sigma^{a\in A}_{\mathrm{bnd,res}}, \Sigma^{a\in A}_{\mathrm{bnd,vrf}})$ must satisfy the following property.

Proposition 1 (Completeness) *If Cmt is correct, and if Σ^a_0 is complete for each $a \in A$, then our $\Sigma^{a\in A}_{bnd}$ is complete.*

Proof. The completeness of our $\Pi^{a\in A}_{\mathrm{bnd}}$ comes from the correctness of Cmt and the completeness of Π^a_0 for each $a \in A$. □

• $\Sigma^{a\in A}_{\mathrm{bnd,ext}}((x^a)^{a\in A}, (c_0, (\mathrm{COM}^a, \mathrm{COM}^a_0)^{a\in A}), \mathrm{CHA}, (\mathrm{RES}^a, \mathrm{RES}^a_0)^{a\in A}, \mathrm{CHA}', ((\mathrm{RES}^a)', (\mathrm{RES}^a_0)')^{a\in A}) \to (\hat{w}_0, (\hat{w}^a_1)^{a\in A})$. This PPT algorithm is for knowledge extraction. On input the statement $(x^a)^{a\in A}$ and two accepting transcripts $((c_0, (\mathrm{COM}^a, \mathrm{COM}^a_0)^{a\in A}), \mathrm{CHA}, (\mathrm{RES}^a, \mathrm{RES}^a_0)^{a\in A}))$ and $((c_0, (\mathrm{COM}^a, \mathrm{COM}^a_0)^{a\in A}), \mathrm{CHA}', (\mathrm{RES}^{a\prime}, \mathrm{RES}^{a\prime}_0)^{a\in A}))$, $\mathrm{CHA} \neq \mathrm{CHA}'$, it first sets the extended statement as $x^a_0 := (x^a, c_0)$ for each $a \in A$. Note here that commitment messages are common and challenge messages are different. Then it executes for each $a \in A$ the algorithm $\Sigma^a_{0,\mathrm{ext}}(x^a_0, (\mathrm{COM}^a, \mathrm{COM}^a_0), \mathrm{CHA}, (\mathrm{RES}^a, \mathrm{RES}^a_0), \mathrm{CHA}', (\mathrm{RES}^{a\prime}, \mathrm{RES}^{a\prime}_0))$. It obtains $(\hat{w}^a_0, \hat{w}^a_1, \hat{r}^a_0)$. We emphasize that the *property (19) is needed here.* If this event does not occur (i.e. $\Sigma^a_{0,\mathrm{ext}}$ fails to extract a witness for at least one a), then it returns $\perp$. Otherwise, if $\hat{w}^a_0 = \hat{w}^{a'}_0$ for any $a, a' \in A$, then it sets the common value $\hat{w}_0 := \hat{w}^a_0$ and returns $(\hat{w}_0, (\hat{w}^a_1)^{a\in A})$. Otherwise it returns $\perp^*$. The binding property of the commitment scheme Cmt assures that the former case holds with an overwhelming probability, as claimed in the following proposition.

Proposition 2 (Special Soundness) *If Cmt is correct and computationally binding, and if Σ^a_0 has the special soundness for each $a \in A$, then our $\Sigma^{a\in A}_{bnd}$ has the special soundness.*

Proof. By employing $(\Sigma^{a\in A}_{\mathrm{bnd,com}}, \Sigma^{a\in A}_{\mathrm{bnd,cha}}, \Sigma^{a\in A}_{\mathrm{bnd,res}}, \Sigma^{a\in A}_{\mathrm{bnd,vrf}}, \Sigma^{a\in A}_{\mathrm{bnd,ext}})$ as subroutines, we construct a PPT algorithm $\mathbf{A}$ that breaks the binding property of Cmt in accordance with the experiment $\mathsf{Exp}^{\mathrm{bind}}_{\mathsf{Cmt},\mathbf{A}}(1^\lambda)$. $\mathbf{A}$ is given as input the set of public parameter pp_{CmtPrv}. $\mathbf{A}$ first reads out the security parameter 1^λ from pp_{CmtPrv}, and executes the setup algorithms $\Pi.\mathsf{Setup}(1^\lambda)$. It obtains the set of public parameter pp_Π. $\mathbf{A}$ merges the sets of public parameter as $pp := (pp_\Pi, pp_{\mathsf{CmtPrv}})$. Then $\mathbf{A}$ executes $\Pi^{a\in A}_{\mathrm{bnd}} = (\mathsf{P}, \mathsf{V})$. If the decision d of V is 1, then $\mathbf{A}$ rewinds P back to the timing at which P had sent the challenge message CHA of the protocol $\Sigma^{a\in A}_{\mathrm{bnd}}$. If the decision d of V is again 1, $\mathbf{A}$ executes the knowledge extractor $\Sigma^{a\in A}_{\mathrm{bnd,ext}}$ on input $((x^a)^{a\in A}, (c_0, (\mathrm{COM}^a, \mathrm{COM}^a_0)^{a\in A})), \mathrm{CHA}, (\mathrm{RES}^a, \mathrm{RES}^a_0)^{a\in A}, \mathrm{CHA}', ((\mathrm{RES}^a)', (\mathrm{RES}^a_0)')^{a\in A})$. If $\Sigma^{a\in A}_{\mathrm{bnd,ext}}$ outputs $\perp^*$, then there must be a pair $a, a' \in A^*, a \neq a'$ such that $(\hat{w}^a_0, \hat{w}^a_1, \hat{r}^a_0)$ and $(\hat{w}^{a'}_0, \hat{w}^{a'}_1, \hat{r}_{a',0})$ pass the verification $\mathsf{Cmt.Vrf}$ and $\hat{w}^a_0 \neq \hat{w}^{a'}_0$. The vector $(c_0, \hat{w}^a_0, \hat{r}^a_0, \hat{w}^{a'}_0, \hat{r}_{a',0})$ breaks the binding property to yields

WIN in $\mathsf{Exp}^{\mathrm{bind}}_{\mathsf{Cmt},\mathbf{A}}(1^\lambda)$. This completes the description of $\mathbf{A}$, and the following equality holds.

$$\mathbf{Adv}^{\mathrm{bind}}_{\mathsf{Cmt},\mathbf{A}}(\lambda) = \Pr[\Sigma^{a\in A}_{\mathrm{bnd,ext}} \text{ returns } \bot^*] \tag{27}$$

$$= 1 - (\Pr[\Sigma^{a\in A}_{\mathrm{bnd,ext}} \text{ returns}(\hat{w}_0, (\hat{w}^a_1)^{a\in A})] + \Pr[\Sigma^{a\in A}_{\mathrm{bnd,ext}} \text{ returns } \bot]). \tag{28}$$

Therefore,

$$\Pr[\Sigma^{a\in A}_{\mathrm{bnd,ext}} \text{ returns}(\hat{w}_0, (\hat{w}^a_1)^{a\in A})] = 1 - (\mathbf{Adv}^{\mathrm{bind}}_{\mathsf{Cmt},\mathbf{A}}(\lambda) + \Pr[\Sigma^{a\in A}_{\mathrm{bnd,ext}} \text{ returns } \bot]) \tag{29}$$

$$= 1 - (\mathbf{Adv}^{\mathrm{bind}}_{\mathsf{Cmt},\mathbf{A}}(\lambda) + (1 - \prod_{a\in A} \Pr[\Sigma^a_{0,\mathrm{ext}} \text{ returns a witness}])). \tag{30}$$

The right-hand side is an overwhelming probability because $\Pr[\Sigma^a_{0,\mathrm{ext}}$ returns a witness] is an overwhelming probability for each $a \in A$ and $|A|$ is bounded by a polynomial in $|x|$. □

Note 3: Our Use Case For simplicity of the later discussion, we hereafter assume that, for all $a \in A$, $\Pr[\Sigma^a_{0,\mathrm{ext}}$ returns a witness$] = 1$. That is, we assume that $\Pr[\Sigma^a_{0,\mathrm{ext}}$ returns $\bot] = 0$ for each $a \in A$.

- $\Sigma^{a\in A}_{\mathrm{bnd,sim}}((x^a)^{a\in A}, \tilde{\mathrm{CHA}}) \to ((\tilde{c}_0, (\tilde{\mathrm{COM}}^a, \tilde{\mathrm{COM}}^a_0)^{a\in A}), (\tilde{\mathrm{RES}}^a, \tilde{\mathrm{RES}}^a_0)^{a\in A})$. This PPT algorithm is for the simulation of an accepting transcript. On input a statement $(x^a)^{a\in A}$ and a uniform random string $\tilde{\mathrm{CHA}} \in_R \mathrm{CHASP}(1^\lambda)$, it first chooses a base witness point $\tilde{w}_0 \in_R W_0$ uniformly at random, and executes the commitment generation algorithm with a randomness $\tilde{r}_0$, $\mathsf{Cmt.Com}(\tilde{w}_0; \tilde{r}_0) \to (\tilde{c}_0, \tilde{r}_0)$. It obtains a commitment $\tilde{c}_0$. Then it sets the extended statement as $x^a_0 := (x^a, \tilde{c}_0)$ for each $a \in A$. Then it executes for each $a \in A$ the algorithm $\Sigma^a_{0,\mathrm{sim}}(x^a_0, \tilde{\mathrm{CHA}})$. It obtains $((\tilde{\mathrm{COM}}^a, \tilde{\mathrm{COM}}^a_0), (\tilde{\mathrm{RES}}^a, \tilde{\mathrm{RES}}^a_0))$. It returns $((\tilde{c}_0, (\tilde{\mathrm{COM}}^a, \tilde{\mathrm{COM}}^a_0)^{a\in A}), (\tilde{\mathrm{RES}}^a, \tilde{\mathrm{RES}}^a_0)^{a\in A})$.

Proposition 3 (Honest-Verifyer Zero-Knowledge) *If* Cmt *is perfectly hiding, and if* Σ^a_0 *is honest-verifier zero-knowledge for each* $a \in A$*, then our* $\Sigma^{a\in A}_{bnd}$ *is honest-verifier zero-knowledge.*

Proof. The perfectly hiding property assures that the distribution of simulated commitment $\tilde{c}_0$ is the same as the real. Then on input $(x^a_0, \tilde{\mathrm{CHA}})$, the simulator $\Sigma^a_{0,\mathrm{sim}}$ works to return the remaining part of the simulated transcript, $((\tilde{\mathrm{COM}}^a, \tilde{\mathrm{COM}}^a_0), (\tilde{\mathrm{RES}}^a, \tilde{\mathrm{RES}}^a_0))$ for each $a \in A$. Then, the merged transcripts $((\tilde{c}_0, (\tilde{\mathrm{COM}}^a, \tilde{\mathrm{COM}}^a_0)^{a\in A}), (\tilde{\mathrm{RES}}^a, \tilde{\mathrm{RES}}^a_0)^{a\in A})$ is identically distributed to the real. □

Theorem 1 *If* Cmt *is correct, computationally binding and perfectly hiding, and if* Σ^a_0 *is a* Σ*-protocol for each* $a \in A$*, then our protocol* $\Sigma^{a\in A}_{bnd}$ *is a* Σ*-protocol.*

Proof. Propositions 1, 2 and 3 deduces that $\Sigma^{a\in A}_{\mathrm{bnd}}$ is a Σ-protocol. □

Theorem 2 *If the component interactive proof system* Π^a_0 *with* Σ^a_0 *is perfectly witness-indistinguishable for each* $a \in A$*, and if* Cmt *is perfectly hiding, then our interactive argument system* $\Pi^{a\in A}_{bnd}$ *with* $\Sigma^{a\in A}_{bnd}$ *is perfectly witness-indistinguishable.*

Proof. The transcripts form a distribution $dist^{a\in A} := dist\big((c_0, (\mathrm{COM}^a, \mathrm{COM}^a_0)^{a\in A}), \mathrm{CHA}, (\mathrm{RES}^a, \mathrm{RES}^a_0)^{a\in A}\big)$, where the challenge message CHA is chosen by any given PPT verifier V^* on input a set of statements $(x^a)^{a\in A}$, any given auxiliary input z and a commitment message $(c_0, (\mathrm{COM}^a, \mathrm{COM}^a_0)^{a\in A})$. If Cmt is perfectly hiding, then the distribution of the commitment c_0 is identical even if the committed element w_0 varies. For each $a \in A$, if Π^a_0 is perfectly witness-indistinguishable, then the distribution of the commitment message and the response message $dist\big((\mathrm{COM}^a, \mathrm{COM}^a_0), (\mathrm{RES}^a, \mathrm{RES}^a_0)\big)$ are identical even if the witness (w_0, w^a_1) varies and even if CHA chosen by $\mathsf{V}^*((x^a)^{a\in A}, z, (c_0, (\mathrm{COM}^a, \mathrm{COM}^a_0)^{a\in A}))$ deviates from the uniform random distribution. Therefore, for all $a \in A$, the distribution $dist^{a\in A}$ is identical even if the witness $(w_0, (w^a_1)_{a\in A})$ varies. □

Note. OR-composition and Boolean Formulas The OR-proof, and more generally the proof for monotone formulas, are also possible for our Σ-protocol $\Sigma^{a\in A}_{\mathrm{bnd}}$ (see [CDS94,AAS14]).

4 Decentralized Multi-Authority Anonymous Authentication Scheme

In this section, we give a syntax and security definitions of an interactive anonymous authentication scheme in a decentralized multi-authority setting on key generation.

$\mathsf{P}((x^a)^{a \in A}, w_0, (w_1^a)^{a \in A})$		$\mathsf{V}((x^a)^{a \in A})$
$\Sigma^{a \in A}_{\text{bnd,com}}((x^a)^{a \in A}, w_0, (w_1^a)^{a \in A})$		
$(c_0, r_0) \leftarrow \mathsf{Cmt.Com}(w_0; r_0)$		
For $a \in A$:		
$x_0^a := (x^a, c_0), w_0^a := (w_0, w_1^a, r_0)$		
$\Sigma^a_{0,\text{com}}(x_0^a, w_0^a) \to (\text{COM}^a, \text{COM}_0^a, \overline{St_0^a})$		
$St := (\overline{St_0^a})^{a \in A}$		
Return $(c_0, (\text{COM}^a, \text{COM}_0^a)^{a \in A}, St)$	$c_0, (\text{COM}^a, \text{COM}_0^a)^{a \in A}$ $\rightarrow$	$\Sigma^{a \in A}_{\text{bnd,cha}}((x^a)^{a \in A})$
		$\text{CHA} \in_R \text{CHASP}(1^\lambda)$
	CHA	Return CHA
$\Sigma^{a \in A}_{\text{bnd,res}}(St, \text{CHA})$	$\leftarrow$	
For $a \in A$:		
$\Sigma^a_{0,\text{res}}(St^a, \text{CHA}) \to (\text{RES}^a, \text{RES}_0^a)$		
Return $(\text{RES}^a, \text{RES}_0^a)^{a \in A}$	$(\text{RES}^a, \text{RES}_0^a)^{a \in A}$ $\rightarrow$	$\Sigma^{a \in A}_{\text{bnd,vrf}}((x^a)^{a \in A})$
		For $a \in A$:
		$x_0^a := (x^a, c_0)$
		$\Sigma^a_{0,\text{vrf}}(x_0^a, (\text{COM}^a, \text{COM}_0^a), \text{CHA}, (\text{RES}^a, \text{RES}_0^a)) =_? 1$
		If TRUE for all $a \in A$, then Return $d := 1$
		else Return $d := 0$
		Return d

Fig. 1. The protocol $\Sigma^{a \in A}_{\text{bnd}}$ of our proof system $\Pi^{a \in A}_{\text{bnd}}$ for the NP witness relation $R^{a \in A}_{\text{bnd}}$.

4.1 Syntax and Security Definitions

Our scheme a-auth consists of five PPT algorithms, (Setup, AuthKG, PrivKG, P, V).

- $\mathsf{Setup}(1^\lambda) \to pp$. This PPT algorithm is needed to generate a set of public parameter pp. On input the security parameter 1^λ, it generates the set pp. It returns pp.

- $\mathsf{AuthKG}(1^\lambda, a) \to (\text{PK}^a, \text{MSK}^a)$. This PPT algorithm is executed by a key-issuing authority indexed by a positive integer a. On input the security parameter 1^λ and the authority index a, it generates the a-th public key PK^a of the authority and the corresponding a-th master secret key MSK^a. It returns $(\text{PK}^a, \text{MSK}^a)$.

- $\mathsf{PrivKG}(\text{PK}^a, \text{MSK}^a, \mathtt{i}) \to \text{sk}^a_{\mathtt{i}}$. This PPT algorithm is executed by the a-th key-issuing authority. On input the a-th public and master secret keys $(\text{PK}^a, \text{MSK}^a)$ and a string $\mathtt{i}$ of a prover (a global identity string), it generates a private secret key $\text{sk}^a_{\mathtt{i}}$ of a prover. It returns $\text{sk}^a_{\mathtt{i}}$.

- $\langle \mathsf{P}((\text{PK}^a, \text{sk}^a_{\mathtt{i}})^{a \in A'}), \mathsf{V}((\text{PK}^a)^{a \in A'}) \rangle \to d$. These two interactive PPT algorithms are a prover who is to be authenticated, and a verifier who confirms that the prover certainly knows the secret keys for indices $a \in A'$, respectively, where A' denotes a subset of all indices at which the prover is issued her private secret keys by authorities. On input the public keys $(\text{PK}^a)^{a \in A}$ to P and V and the corresponding private secret keys $(\text{sk}^a_{\mathtt{i}})^{a \in A}$ to P, P and V interact with each other. After at most polynomially many (in λ) moves of messages between P and V, V returns $d := 1$ ("accept") or $d := 0$ ("reject").

We discuss two security notions for our authentication scheme a-auth; security against concurrent and collusion attacks that yield misauthentication, and anonymity for privacy of provers' global identities.

Security against Concurrent and Collusion Attack of Misauthentication One of the strongest attacks to cause misauthentication on our a-auth is the concurrent and collusion attack. For a formal treatment we define the following experiment on a-auth and an adversary algorithm $\mathbf{A}$.

$$\begin{aligned}
&\mathsf{Exp}^{\text{conc-coll}}_{\mathsf{a\text{-}auth},\mathbf{A}}(1^\lambda) && (31)\\
&pp \leftarrow \mathsf{Setup}(1^\lambda) && (32)\\
&(q_A, St) \leftarrow \mathbf{A}(pp), A := \{1,\dots,q_A\} && (33)\\
&\text{For } a \in A : (\mathrm{PK}^a, \mathrm{MSK}^a) \leftarrow \mathsf{AuthKG}(1^\lambda, a) && (34)\\
&(q_I, St) \leftarrow \mathbf{A}(St, (\mathrm{PK}^a)^{a\in A}), I := \{1,\dots,q_I\} && (35)\\
&\text{For } i \in I : \mathtt{i}_i \in_R \{0,1\}^\lambda && (36)\\
&\text{For } a \in A : \text{For } i \in I : \mathrm{sk}^a_{\mathtt{i}_i} \leftarrow \mathsf{PrivKG}(\mathrm{PK}^a, \mathrm{MSK}^a, \mathtt{i}_i) && (37)\\
&(\tilde{A}, St) \leftarrow \mathbf{A}(St),\ A^* := A\backslash\tilde{A} && (38)\\
&St \leftarrow \mathbf{A}^{\mathsf{P}((\mathrm{PK}^a,\mathrm{sk}^a_{\mathtt{i}_i})^{a\in A^*})|_{i\in I}, \mathbf{PrivKGO}(\mathrm{PK}^\cdot,\mathrm{MSK}^\cdot,\cdot)}(St, (\mathrm{MSK}^a)^{a\in\tilde{A}}) && (39)\\
&\langle \mathbf{A}(St), \mathsf{V}\rangle((\mathrm{PK}^a)^{a\in A^*}) \rightarrow d && (40)\\
&\text{If } d=1 \text{ then Return } \textsc{Win} \text{ else Return } \textsc{Lose} && (41)
\end{aligned}$$

Intuitively, the above experiment describes the attack as follows. $\mathbf{A}$ first outputs the number q_A of key-issuing authorities. Then $\mathbf{A}$ outputs the number q_I of concurrent provers. Then $\mathbf{A}$ outputs a set of indices of corrupted authorities $\tilde{A}$. The target set of authority indices is fixed as $A^* := A\backslash\tilde{A}$.

In the "learning phase" $\mathbf{A}$ is given as input the master secret keys $(\mathrm{MSK}^a)^{a\in\tilde{A}}$. Then $\mathbf{A}$ interacts concurrently with q_I provers on the target public keys $(\mathrm{PK}^a)^{a\in A^*}$ ("concurrent" means "in arbitrarily interleaved order of messages"). In addition, $\mathbf{A}$ collects private secret keys by issuing private secret key queries for $j = q_I+1,\dots,q_I+q_{\mathrm{sk}}$ to the oracle $\mathbf{PrivKGO}(\mathrm{PK}^\cdot, \mathrm{MSK}^\cdot, \cdot)$ with an authority index $a \in A^*$ and an identity string $\mathtt{i}_j \in \{0,1\}^\lambda$. We denote by A_j the set of authority indices for which the private secret key queries were issued with $\mathtt{i}_j$. That is,

$$A_j := \{a \in A \mid \mathbf{A} \text{ is given } \mathrm{sk}^a_{\mathtt{i}_j}\} \subset A^*. \tag{42}$$

Note that the maximum number of private secret key queries is $q_A q_{\mathrm{sk}}$. We require that the numbers q_A, q_I and q_{sk} are bounded by a polynomial in λ.

Next, in the "attacking phase", $\mathbf{A}$ is given as input the inner state St. $\mathbf{A}$ interacts with the verifier V on the target public keys $(\mathrm{PK}^a)^{a\in A^*}$. If the decision d of V is 1, then the experiment returns $\textsc{Win}$ and otherwise, returns $\textsc{Lose}$. Two restrictions are imposed on the adversary $\mathbf{A}$; the queried $\mathtt{i}_j$s are pairwise different, and any A_j is a proper subset of the target set A^*:

$$\begin{aligned}
&\mathtt{i}_{j_1} \neq \mathtt{i}_{j_2} \text{ for } j_1, j_2 \in \{q_I+1,\dots,q_I+q_{\mathrm{sk}}\}, j_1 \neq j_2, && (43)\\
&A_j \subsetneq A^*,\ j = q_I+1,\dots,q_I+q_{\mathrm{sk}}. && (44)
\end{aligned}$$

The advantage of an adversary $\mathbf{A}$ over our authentication scheme a-auth in the experiment is defined as:

$$\mathbf{Adv}^{\text{conc-coll}}_{\mathsf{a\text{-}auth},\mathbf{A}}(\lambda) \stackrel{\text{def}}{=} \Pr[\mathsf{Exp}^{\text{conc-coll}}_{\mathsf{a\text{-}auth},\mathbf{A}}(1^\lambda) = \textsc{Win}]. \tag{45}$$

An authentication scheme a-auth is called secure against concurrent and collusion attacks if, for any PPT algorithm $\mathbf{A}$, the advantage $\mathbf{Adv}^{\text{conc-coll}}_{\mathsf{a\text{-}auth},\mathbf{A}}(\lambda)$ is negligible in λ.

Anonymity As is explained in Section 1, a critical feature to be attained is provers' anonymity on global identities when the provers are authenticated. For a formal treatment we define the following experiment on

$\mathsf{a\text{-}auth}$ and an adversary algorithm $\mathbf{A}$.

$$\begin{aligned}
&\mathsf{Exp}^{\text{ano}}_{\mathsf{a\text{-}auth},\mathbf{A}}(1^\lambda) &(46)\\
&\quad pp \leftarrow \mathsf{Setup}(1^\lambda) &(47)\\
&\quad (q_A, St) \leftarrow \mathbf{A}(pp), A := \{1, \dots, q_A\} &(48)\\
&\quad \text{For } a \in A : (\mathrm{PK}^a, \mathrm{MSK}^a) \leftarrow \mathsf{AuthKG}(1^\lambda, a) &(49)\\
&\quad \mathtt{i}_0, \mathtt{i}_1 \leftarrow \mathbf{A}(St, (\mathrm{PK}^a)^{a\in A}) &(50)\\
&\quad \text{For } a \in A : \text{For } i \in 0,1 : \mathrm{sk}^a_{\mathtt{i}_i} \leftarrow \mathsf{PrivKG}(\mathrm{PK}^a, \mathrm{MSK}^a, \mathtt{i}_i) &(51)\\
&\quad b \in_R \{0,1\}, b^* \leftarrow \mathbf{A}^{\mathsf{P}((\mathrm{PK}^a, \mathrm{sk}^a_{\mathtt{i}_b})^{a\in A})}(St, (\mathrm{sk}^a_{\mathtt{i}_0}, \mathrm{sk}^a_{\mathtt{i}_1})^{a\in A}) &(52)\\
&\quad \text{If } b = b^*, \text{ then Return } \textsc{Win}, \text{ else Return } \textsc{Lose} &(53)
\end{aligned}$$

Intuitively, the above experiment describes the attack as follows. The adversary algorithm $\mathbf{A}$, on input the security parameter 1^λ, first outputs the number q_A of key-issuing authorities. Then, on input the issued public keys $(\mathrm{PK}^a)^{a\in A}$, $\mathbf{A}$ designates two identity strings $\mathtt{i}_0$ and $\mathtt{i}_1$ (as is usual in the indistinguishability games). Next, $\mathbf{A}$ interacts with a prover P on input the private secret keys $(\mathrm{sk}^a_{\mathtt{i}_b})^{a\in A}$, where the index b is chosen uniformly at random. If the decision b^* of $\mathbf{A}$ is equal to b, then the experiment returns $\textsc{Win}$ and otherwise, returns $\textsc{Lose}$.

The advantage of an adversary $\mathbf{A}$ over our authentication scheme $\mathsf{a\text{-}auth}$ in the experiment is defined as: $\mathbf{Adv}^{\text{ano}}_{\mathsf{a\text{-}auth},\mathbf{A}}(\lambda) \stackrel{\text{def}}{=} \left|\Pr[\mathsf{Exp}^{\text{ano}}_{\mathsf{a\text{-}auth},\mathbf{A}}(1^\lambda) = \textsc{Win}] - (1/2)\right|$. An authentication scheme $\mathsf{a\text{-}auth}$ is called to have anonymity if, for any PPT algorithm $\mathbf{A}$, the advantage $\mathbf{Adv}^{\text{ano}}_{\mathsf{a\text{-}auth},\mathbf{A}}(\lambda)$ is negligible in λ.

4.2 Generic Construction

We give a generic construction of our authentication scheme $\mathsf{a\text{-}auth}$. The building blocks are the interactive proof system $\Pi^{a\in A}_{\text{bnd}}$ with our Σ-protocol $\Sigma^{a\in A}_{\text{bnd}}$ and a digital signature scheme Sig. We note that a commit-and-prove scheme CmtPrv is employed in $\Sigma^{a\in A}_{\text{bnd}}$.

• $\mathsf{Setup}(1^\lambda) \to pp$. On input the security parameter 1^λ, this PPT algorithm generates a set of public parameter by running the setup algorithms $\mathsf{Sig.Setup}(1^\lambda)$, $\Pi.\mathsf{Setup}(1^\lambda)$ and $\mathsf{CmtPrv.Setup}(1^\lambda)$. These algorithms are for the digital signature scheme Sig, the interactive argument systems $(\Pi^a_0)^{a\in A}$, and the commitment generation algorithm $\mathsf{Cmt.Com}$. They generate pp_{Sig}, pp_Π and pp_{Cmt}, respectively. It merges them as $pp := (pp_{\mathsf{Sig}}, pp_\Pi, pp_{\mathsf{Cmt}})$. It returns pp.

• $\mathsf{AuthKG}(1^\lambda, a) \to (\mathrm{PK}^a, \mathrm{MSK}^a)$. On input the security parameter 1^λ and an authority index a, this PPT algorithm executes the key generation algorithm $\mathsf{Sig.KG}(1^\lambda)$. It obtains a signing key SK and the corresponding public key PK. It sets the master secret key as $\mathrm{MSK}^a := \mathrm{SK}$ and the corresponding public key as $\mathrm{PK}^a := \mathrm{PK}$. It returns $(\mathrm{PK}^a, \mathrm{MSK}^a)$.

• $\mathsf{PrivKG}(\mathrm{PK}^a, \mathrm{MSK}^a, \mathtt{i}) \to \mathrm{sk}^a_{\mathtt{i}}$. On input a public key PK^a, the corresponding master secret key MSK^a and a string $\mathtt{i}$, this PPT algorithm executes the signing algorithm $\mathsf{Sig.Sign}(\mathrm{PK}^a, \mathrm{MSK}^a, \mathtt{i})$. It obtains a digital signature $\sigma^a_{\mathtt{i}}$ on the message $\mathtt{i}$. It puts a private secret key $\mathrm{sk}^a_{\mathtt{i}}$ as $\mathrm{sk}^a_{\mathtt{i}} := \sigma^a_{\mathtt{i}}$. It returns $\mathrm{sk}^a_{\mathtt{i}}$.

• $\mathsf{P}((\mathrm{PK}^a)^{a\in A}, (\mathrm{sk}^a_{\mathtt{i}})^{a\in A})$ and $\mathsf{V}((\mathrm{PK}^a)^{a\in A})$. On input the public keys $(\mathrm{PK}^a)^{a\in A}$ to the prover P and the verifier V, and the corresponding private secret keys $(\mathrm{sk}^a_{\mathtt{i}})^{a\in A}$ to P, PPT algorithms P and V first set the statements as $x^a := \mathrm{PK}^a$ for each $a \in A$ and P sets the witness as $w_0 := \mathtt{i}$ and $w^a_1 := \mathrm{sk}^a_{\mathtt{i}}$ for each $a \in A$. The witness spaces $W^a, a \in A$ are described as follows.

$$\begin{aligned}
W^a =& W_0 \times W^a_1, &(54)\\
&W_0 = \{\mathtt{i} \mid \text{string of length } \lambda\} = \{0,1\}^\lambda, &(55)\\
&W^a_1 = \{\sigma^a_{\mathtt{i}} \mid \sigma^a_{\mathtt{i}} \leftarrow \mathsf{Sig.Sign}(\mathrm{PK}^a, \mathrm{MSK}^a, \mathtt{i}) \text{ for some } \mathtt{i} \in W_0\}. &(56)
\end{aligned}$$

P and V execute the Σ protocol $\Sigma^{a\in A}_{\text{bnd}}$. V returns the returned boolean d of the verifier algorithm $\Sigma^{a\in A}_{\text{bnd,vrf}}$.

4.3 Properties

Theorem 3 *If the component proof system Π^a_0 is perfectly witness-indistinguishable for each $a \in A$, if the commitment scheme Cmt is perfectly hiding and computationally binding, and if the digital signature scheme*

$\mathsf{Setup}(1^\lambda)$	$\mathsf{AuthKG}(1^\lambda, a)$	$\mathsf{PrivKG}(\mathrm{PK}^a, \mathrm{MSK}^a, \mathtt{i})$
$pp_{\mathsf{Sig}} \leftarrow \mathsf{Sig.Setup}(1^\lambda)$	$(\mathrm{SK}, \mathrm{PK}) \leftarrow \mathsf{Sig.KG}(1^\lambda)$	$\sigma^a_{\mathtt{i}} \leftarrow \mathsf{Sig.Sign}(\mathrm{PK}^a, \mathrm{MSK}^a, \mathtt{i})$
$pp_{\Pi} \leftarrow \Pi.\mathsf{Setup}(1^\lambda)$	$\mathrm{PK}^a := \mathrm{PK}, \mathrm{MSK}^a := \mathrm{SK}$	$\mathrm{sk}^a_{\mathtt{i}} := \sigma^a_{\mathtt{i}}$
$pp_{\mathsf{CmtPrv}} \leftarrow \mathsf{CmtPrv.Setup}(1^\lambda)$	Return $(\mathrm{PK}^a, \mathrm{MSK}^a)$	Return $\mathrm{sk}^a_{\mathtt{i}}$
$pp := (pp_{\Pi}, pp_{\mathsf{CmtPrv}}, pp_{\mathsf{Sig}})$		
Return pp		

$\mathsf{P}((\mathrm{PK}^a)^{a\in A}, (\mathrm{sk}^a_{\mathtt{i}})^{a\in A})$		$\mathsf{V}((\mathrm{PK}^a)^{a\in A})$
For $a \in A$: $x^a := \mathrm{PK}^a, w^a_1 := \mathrm{sk}^a_{\mathtt{i}}$		For $a \in A$: $x^a := \mathrm{PK}^a$
$w_0 := \mathtt{i}$		
	(Execute $\Sigma^{a\in A}_{\mathrm{bnd}}$)	
		Return $(d \leftarrow \Sigma^{a\in A}_{\mathrm{bnd,vrf}})$

Fig. 2. Generic construction of our decentralized multi-authority anonymous authentication scheme a-auth.

Sig is existentially unforgeable against adaptive chosen-message attacks, then our a-auth is secure against concurrent and collusion attacks. More precisely, let q_A denote the maximum number of authorities. For any given PPT *algorithm* **A** *that executes a concurrent and collusion attack on our a-auth in accordance with the experiment $\mathsf{Exp}^{\textit{conc-coll}}_{\textsf{a-auth},\mathbf{A}}(1^\lambda)$, there exist a* PPT *algorithm* **F** *that generates an existential forgery on Sig in accordance with the experiment $\mathsf{Exp}^{\textit{euf-cma}}_{\mathsf{Sig},\mathbf{F}}(1^\lambda)$ and a* PPT *algorithm* **B** *that breaks the binding property of Cmt in accordance with the experiment $\mathsf{Exp}^{\textit{bind}}_{\mathsf{Cmt},\mathbf{B}}(1^\lambda)$ which satisfy the following inequality.*

$$\mathbf{Adv}^{\textit{conc-coll}}_{\textsf{a-auth},\mathbf{A}}(\lambda) \leq \frac{1}{|\textsc{ChaSp}(1^\lambda)|} + \sqrt{\frac{2^\lambda}{2^\lambda - 1} \cdot q_A \cdot \mathbf{Adv}^{\textit{euf-cma}}_{\mathsf{Sig},\mathbf{F}}(\lambda) + \mathbf{Adv}^{\textit{bind}}_{\mathsf{Cmt},\mathbf{B}}(\lambda)}. \tag{57}$$

Proof. Given any PPT algorithm **A** on $\mathsf{Exp}^{\text{conc-coll}}_{\textsf{a-auth},\mathbf{A}}(1^\lambda)$, we construct a PPT algorithm **F** that generates an existential forgery on Sig in accordance with the experiment $\mathsf{Exp}^{\text{euf-cma}}_{\mathsf{Sig},\mathbf{F}}(1^\lambda)$. **F** is given as input the set of public parameter pp_{Sig} and a public key $\mathrm{PK}_{\mathsf{Sig}}$. **F** first reads out the security parameter 1^λ from pp_{Sig}, and executes the setup algorithms $\Pi.\mathsf{Setup}(1^\lambda)$ and $\mathsf{CmtPrv.Setup}(1^\lambda)$. It obtains the sets of public parameter pp_{Π} and pp_{CmtPrv}, respectively. **F** merges the sets of public parameter as $pp := (pp_{\mathsf{Sig}}, pp_{\Pi}, pp_{\mathsf{CmtPrv}})$. Then **F** invokes the algorithm **A** with pp. **F** obtains the number q_A of key-issuing authorities from **A**. **F** chooses a *target index* $a^\dagger$ from the set $A := \{1, \ldots, q_A\}$ *uniformly at random.* **F** executes the authority key generation algorithm honestly for $a \in A$ *except* the target index $a^\dagger$. As for $a^\dagger$, **F** uses the input public key:

$$\text{For } a \in A, a \neq a^\dagger : (\mathrm{PK}^a, \mathrm{MSK}^a) \leftarrow \mathsf{AuthKG}(1^\lambda, a),$$
$$\text{For } a = a^\dagger : \mathrm{PK}^{a^\dagger} := \mathrm{PK}_{\mathsf{Sig}}.$$

F inputs St and the public keys $(\mathrm{PK}^a)^{a\in A}$ into **A**. Then **F** obtains the number q_I of concurrent provers from **A**. **F** sets I as $I := \{1, \ldots, q_I\}$. **F** inputs St into **A**. Then **F** obtains a set of corrupted authority indices $\tilde{A}$ from **A**. **F** puts $A^* := A \backslash \tilde{A}$. If $a^\dagger \in A^*$ (the case TgtIdx), then $a^\dagger$ is not in $\tilde{A}$ and **F** is able to input $(St, (\mathrm{MSK}^a)^{a\in\tilde{A}})$ into **A**. Otherwise **F** aborts.

Simulation of Concurrent Provers. When **A** invokes q_I provers $\mathsf{P}((\mathrm{PK}^a, \mathrm{sk}^a_{\mathtt{i}_i})^{a\in A^*})|_{i\in I}$, **F** chooses $\mathtt{i}^\dagger \in_R \{0,1\}^\lambda$. **F** executes the private secret key generation algorithm with input $\mathtt{i}^\dagger$ honestly for $a \in A^*$ where $a \neq a^\dagger$. As for $a = a^\dagger$, **F** issues a signing query to its oracle with $\mathtt{i}^\dagger$:

$$\text{For } a \in A^* \text{ s.t. } a \neq a^\dagger : \mathrm{sk}^a_{\mathtt{i}^\dagger} \leftarrow \mathsf{PrivKG}(\mathrm{PK}^a, \mathrm{MSK}^a, \mathtt{i}^\dagger),$$
$$\text{For } a = a^\dagger, \mathrm{sk}^{a^\dagger}_{\mathtt{i}^\dagger} \leftarrow \mathbf{SignO}(\mathrm{PK}, \mathrm{SK}, \mathtt{i}^\dagger).$$

In the simulation of concurrent provers $\mathsf{P}((\mathrm{PK}^a, \mathrm{sk}^a_{\mathtt{i}_i})^{a\in A^*})|_{i\in I}$, **F** uses the *single* set of private secret key $(\mathrm{sk}^a_{\mathtt{i}^\dagger})^{a\in A^*}$. This is a perfect simulation due to the perfect witness-indistinguishability of $\Pi.\mathsf{Setup}(1^\lambda)$.

Simulation of Private Secret Key Oracle. When **A** issues a private secret key query with $A_j \subsetneq A^*$ and $\mathtt{i}_j \in \{0,1\}^\lambda (j = q_I + 1, \ldots, q_I + q_{\mathrm{sk}})$, **F** executes the private secret key generation algorithm with $\mathtt{i}_j$ honestly for $a \in A^*$ such that $a \neq a^\dagger$. As for $a = a^\dagger$, **F** issues a signing query to its oracle with $\mathtt{i}_j$:

$$\text{For } a \in A^* \text{ s.t. } a \neq a^\dagger : \mathrm{sk}^a_{\mathtt{i}_j} \leftarrow \mathsf{PrivKG}(\mathrm{PK}^a, \mathrm{MSK}^a, \mathtt{i}_j),$$

$$\text{For } a = a^\dagger, \mathrm{sk}^{a^\dagger}_{\mathtt{i}_j} \leftarrow \mathbf{SignO}(\mathrm{PK}, \mathrm{SK}, \mathtt{i}_j).$$

F replies to **A** with the secret key $\mathrm{sk}^a_{\mathtt{i}_j}$. This is also a perfect simulation.
Generating Existential Forgery. In the "attacking phase", on input the inner state St, the adversary **A** interacts with the verifier.

That is, **F** executes a verifier V with an input $((\mathrm{PK}^a)^{a\in A^*})$. If the decision d of V is 1, then **F** *rewinds* (Bellare-Palacio [BP02]) **A** back to the timing at which **A** had sent the first message of the Σ-protocol $\Sigma^{a\in A}_{\mathrm{bnd}}$. If the decision d' of V is again 1, **F** executes the knowledge extractor $\Sigma^{a\in A}_{\mathrm{bnd,ext}}$ on input $((x^a)^{a\in A}, (c_0, (\mathrm{COM}^a, \mathrm{COM}^a_0)^{a\in A})), \mathrm{CHA}, (\mathrm{RES}^a, \mathrm{RES}^a_0)^{a\in A}, \mathrm{CHA}', ((\mathrm{RES}^a)', (\mathrm{RES}^a_0)')^{a\in A})$. If $\Sigma^{a\in A}_{\mathrm{bnd,ext}}$ outputs a witness $\hat{w} := (\hat{w}_0, (\hat{w}^a_1)^{a\in A})$, then **F** sets a message $\mathtt{i}^*$ as $\mathtt{i}^* := \hat{w}_0$. The restriction (43) and (44) of the experiment assures that $\exists j^* \in \{q_I + 1, \ldots, q_I + q_{\mathrm{sk}}\}$, $\exists \hat{a} \in (A^* \backslash A_{j^*})$. **F** chooses such an $\hat{a}$ uniformly at random and sets a signature σ^* as $\sigma^* := \hat{w}^{\hat{a}}_1$. **F** returns a forgery pair of a message and a signature $(\mathtt{i}^*, \sigma^*)$. This completes the description of **F**.

Probability Evaluation The probability that the returned value $(\mathtt{i}^*, \sigma^*)$ is actually an existential forgery is evaluated as follows. We name the events in the above as:

$$\begin{aligned}
&\textsc{Acc} : d = 1,\\
&\textsc{Rst} : d = 1, d' = 1 \text{ and } c \neq c',\\
&\textsc{TgtIdx} : \hat{a} = a^\dagger,\\
&\textsc{Ext} : \Sigma^{a\in A}_{\mathrm{bnd,ext}} \text{ returns a witness } \hat{w} := (\hat{w}_0, (\hat{w}^a_1)^{a\in A^*}),\\
&\textsc{FgID} : \mathtt{i}^* \neq \mathtt{i}^\dagger,\\
&\textsc{Forge} : (\mathtt{i}^*, \sigma^*) \text{ is an existential forgery on } \mathsf{Sig}.
\end{aligned}$$

We have the following inequality by Reset Lemma [BP02].

$$\Pr[\textsc{Acc}] \leq \frac{1}{|\textsc{chaSp}(1^\lambda)|} + \sqrt{\Pr[\textsc{Rst}]}. \tag{58}$$

Besides, the above discussion as well as the definitions deduce the following equalities.

$$\mathbf{Adv}^{\text{conc-coll}}_{\mathsf{a\text{-}auth},\mathbf{A}}(\lambda) = \Pr[\textsc{Acc}], \tag{59}$$

$$\Pr[\textsc{TgtIdx}, \textsc{Rst}, \textsc{Ext}, \textsc{FgID}] = \Pr[\textsc{Forge}], \tag{60}$$

$$\Pr[\textsc{Forge}] = \mathbf{Adv}^{\text{euf-cma}}_{\mathsf{Sig},\mathbf{F}}(\lambda). \tag{61}$$

The left-hand side of the equality (60) is expanded as follows.

$$\begin{aligned}
&\Pr[\textsc{TgtIdx}, \textsc{Rst}, \textsc{Ext}, \textsc{FgID}]\\
=&\Pr[\textsc{TgtIdx}] \Pr[\textsc{Rst}, \textsc{Ext}, \textsc{FgID} \mid \textsc{TgtIdx}]\\
=&\Pr[\textsc{TgtIdx}] \Pr[\textsc{Rst}, \textsc{Ext}, \textsc{FgID}]\\
=&\Pr[\textsc{TgtIdx}] \Pr[\textsc{Rst}, \textsc{Ext}] \Pr[\textsc{FgID} \mid \textsc{Rst}, \textsc{Ext}].
\end{aligned} \tag{62}$$

Claim 3

$$\Pr[\textsc{TgtIdx}] = 1/|A| = 1/q_A. \tag{63}$$

Proof. $\hat{a}$ coincides with $a^\dagger$ with probability $1/|A|$. This is because $a^\dagger$ is chosen uniformly at random from A by **F** and no information that identifies $a^\dagger$ is leaked to **A**. □

Claim 4

$$\Pr[\textsc{FgID} \mid \textsc{Rst}, \textsc{Ext}] = \frac{2^\lambda - 1}{2^\lambda}. \tag{64}$$

Proof. $\mathtt{i}^*$ is not $\mathtt{i}^\dagger$ with probability $\frac{2^\lambda-1}{2^\lambda}$. This is because $\mathtt{i}^\dagger$ is chosen uniformly at random from $\{0,1\}^\lambda$ and no information that identifies the individual witnesses leak to $\mathbf{A}$ due to the perfect witness-indistinguishability of $\Pi.\mathsf{Setup}(1^\lambda)$. □

Claim 5 *If* TGTIDX *and* FGID *occurs, then $\mathtt{i}^*$ is not queried to* $\mathbf{F}$*'s oracle* **SignO***.*

Proof. This is because of the occurrence of the events TGTIDX and FGID and the restriction (43)(44). □

Lemma 1 *For any given* PPT *algorithm* $\mathbf{A}$ *that executes a concurrent and collusion attack on our* a-auth *in accordance with the experiment* $\mathsf{Exp}^{conc\text{-}coll}_{\mathsf{a\text{-}auth},\mathbf{A}}(1^\lambda)$*, there exists a* PPT *algorithm* $\mathbf{B}$ *that breaks the binding property of* Cmt *in accordance with the experiment* $\mathsf{Exp}^{bind}_{\mathsf{Cmt},\mathbf{B}}(1^\lambda)$ *satisfying the following equality.*

$$\Pr[\textsc{Rst}, \overline{\textsc{Ext}}] = \mathbf{Adv}^{bind}_{\mathsf{Cmt},\mathbf{B}}(\lambda). \tag{65}$$

Proof. Given any PPT algorithm $\mathbf{A}$ on $\mathsf{Exp}^{\text{conc-coll}}_{\mathsf{a\text{-}auth},\mathbf{A}}(1^\lambda)$, we construct a PPT algorithm $\mathbf{B}$ that breaks the binding property of Cmt in accordance with the experiment $\mathsf{Exp}^{\text{bind}}_{\mathsf{Cmt},\mathbf{B}}(1^\lambda)$. $\mathbf{B}$ is given as input the set of public parameter pp_{CmtPrv}. $\mathbf{B}$ first reads out the security parameter 1^λ from pp_{CmtPrv}, and executes the setup algorithms $\Pi.\mathsf{Setup}(1^\lambda)$ and $\mathsf{Sig.Setup}(1^\lambda)$. It obtains the sets of public parameter pp_Π and pp_{Sig}, respectively. $\mathbf{B}$ merges the sets of public parameter as $pp := (pp_{\mathsf{Sig}}, pp_\Pi, pp_{\mathsf{CmtPrv}})$. Then $\mathbf{B}$ invokes the algorithm $\mathbf{A}$ with 1^λ. It obtains the number q_A of key-issuing authorities. The simulation of concurrent provers and the simulation of the private secret key oracle are done in the same way. (Note that B does not need to choose a^*.) In the "attacking phase", $\mathbf{B}$ executes a verifier V with an input $((\mathrm{PK}^a)^{a\in A^*})$. If the decision d of V is 1, then $\mathbf{B}$ rewinds $\mathbf{A}$ back to the timing at which $\mathbf{A}$ had sent the challenge message of the Σ-protocol $\Sigma^{a\in A}_{\text{bnd}}$. If the decision d of V is again 1, $\mathbf{B}$ executes the knowledge extractor $\Sigma^{a\in A}_{\text{bnd,ext}}$ on input $((x^a)^{a\in A}, (c_0, (\mathrm{COM}^a, \mathrm{COM}^a_0)^{a\in A})), \mathrm{CHA}, (\mathrm{RES}^a, \mathrm{RES}^a_0)^{a\in A}, \mathrm{CHA}', ((\mathrm{RES}^a)', (\mathrm{RES}^a_0)')^{a\in A})$. If $\Sigma^{a\in A}_{\text{bnd,ext}}$ outputs $\bot^*$, then there must be a pair $a, a' \in A^*, a \neq a'$ such that $(\hat{w}^a_0, \hat{w}^a_1, \hat{r}^a_0)$ and $(\hat{w}^{a'}_0, \hat{w}^{a'}_1, \hat{r}_{a',0})$ pass the verification $\mathsf{Cmt.Vrf}$ and $\hat{w}^a_0 \neq \hat{w}^{a'}_0$. The vector $(c_0, \hat{w}^a_0, \hat{r}^a_0, \hat{w}^{a'}_0, \hat{r}_{a',0})$ breaks the binding property to yields WIN in $\mathsf{Exp}^{\text{bind}}_{\mathsf{Cmt},\mathbf{B}}(1^\lambda)$. This completes the description of $\mathbf{B}$, and $\mathbf{B}$ satisfies (65). □

Note that we have the equality:

$$\Pr[\textsc{Rst}] = \Pr[\textsc{Rst}, \textsc{Ext}] + \Pr[\textsc{Rst}, \overline{\textsc{Ext}}]. \tag{66}$$

Combining (60), (62), (63), (64), (65) and (66), we have:

$$\Pr[\textsc{Rst}] = \frac{2^\lambda}{2^\lambda - 1} \cdot q_A \cdot \Pr[\textsc{Forge}] + \mathbf{Adv}^{\text{bind}}_{\mathsf{Cmt},\mathbf{B}}(\lambda). \tag{67}$$

Combining (58), (59), (67) and (61), we have:

$$\mathbf{Adv}^{\text{conc-coll}}_{\mathsf{a\text{-}auth},\mathbf{A}}(\lambda) \leq \frac{1}{|\textsc{ChaSp}(1^\lambda)|} + \sqrt{\frac{2^\lambda}{2^\lambda - 1} \cdot q_A \cdot \mathbf{Adv}^{\text{euf-cma}}_{\mathsf{Sig},\mathbf{F}}(\lambda) + \mathbf{Adv}^{\text{bind}}_{\mathsf{Cmt},\mathbf{B}}(\lambda)}.$$

□

Theorem 4 *If the component proof system Π^a_0 is perfectly witness-indistinguishable for each $a \in A$, and if the commitment scheme* Cmt *is perfectly hiding, then our* a-auth *has anonymity. More precisely, for any given algorithm* $\mathbf{A}$ *that is not necessarily bounded and that executes the anonymity game on our* a-auth *in accordance with the experiment* $\mathsf{Exp}^{ano}_{\mathsf{a\text{-}auth},\mathbf{A}}(1^\lambda)$*, the following equality holds.*

$$\mathbf{Adv}^{ano}_{\mathsf{a\text{-}auth},\mathbf{A}}(\lambda) = 0. \tag{68}$$

Proof. The perfect witness-indistinguishability of Π^a_0 for each $a \in A$ and the perfectly hiding property of the commitment scheme Cmt assure that our proof system $\Pi^{a\in A}_{\text{bnd}}$ is perfectly witness-indistinguishable by Theorem 2. Then the two distribution $dist^{a\in A} := dist\big((c_0, (\mathrm{COM}^a, \mathrm{COM}^a_0)^{a\in A}), \mathrm{CHA}, (\mathrm{RES}^a, \mathrm{RES}^a_0)^{a\in A}\big)$ is identical even if the auxiliary input z is private secret keys $(\mathrm{sk}^a_{\mathtt{i}_0}, \mathrm{sk}^a_{\mathtt{i}_1})^{a\in A}$. Therefore, the advantage $\mathbf{Adv}^{\text{ano}}_{\mathsf{a\text{-}auth},\mathbf{A}}(\lambda)$ is zero. □

Note. Relation to Attribute-Based Identifications and Signatures Using a monotone formula instead of the AND-composition, a decentralized multi-authority attribute-based authentication scheme [AAHI13] is obtained over a small universe A. Moreover, the Fiat-Shamir transform [FS86] gives a decentralized multi-authority attribute-based signature scheme [OT13]. Note here that the security against the collusion attacks means the security against the passive attacks, and therefore the unforgeability is derived (see [AABN02]).

5 Conclusion

We proposed a generic construction of a Σ-protocol of commit-and-prove type, which is an AND-composition of Σ-protocols on the statements that include a common commitment. When the component Σ-protocols are of witness-indistinguishable argument systems, our Σ-protocol is also a witness-indistinguishable argument system as a whole. As an application, we gave a generic construction of a decentralized multi-authority anonymous authentication scheme. There a witness is a bundle of witnesses each of which decomposes into a fixed global identity string and a digital signature on it. We show an instantiation of the scheme in the setting of bilinear groups.

References

[AABN02] Michel Abdalla, Jee Hea An, Mihir Bellare, and Chanathip Namprempre. From identification to signatures via the fiat-shamir transform: Minimizing assumptions for security and forward-security. In *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, pages 418–433, 2002.

[AAHI13] Hiroaki Anada, Seiko Arita, Sari Handa, and Yosuke Iwabuchi. Attribute-based identification: Definitions and efficient constructions. In *Information Security and Privacy - 18th Australasian Conference, ACISP 2013, Brisbane, Australia, July 1-3, 2013. Proceedings*, pages 168–186, 2013.

[AAS14] Hiroaki Anada, Seiko Arita, and Kouichi Sakurai. Attribute-based signatures without pairings via the fiat-shamir paradigm. In *ASIAPKC'14, Proceedings of the 2nd ACM Wookshop on ASIA Public-Key Cryptography, June 3, 2014, Kyoto, Japan*, pages 49–58, 2014.

[Bab85] László Babai. Trading group theory for randomness. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA*, pages 421–429, 1985.

[BB04] Dan Boneh and Xavier Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, pages 223–238, 2004.

[BB08] Dan Boneh and Xavier Boyen. Short signatures without random oracles and the SDH assumption in bilinear groups. *J. Cryptology*, 21(2):149–177, 2008.

[BP02] Mihir Bellare and Adriana Palacio. GQ and Schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, pages 162–177, 2002.

[CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings*, pages 174–187, 1994.

[CLOS02] Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. Universally composable two-party and multi-party secure computation. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 494–503, 2002.

[Cra96] Ronald Cramer. *Modular Designs of Secure, yet Practical Cyptographic Protocols*. PhD thesis, University of Amsterdam, Amsterdam, the Netherlands, 1996.

[Dam10] Ivan Damgård. On σ-protocols. In Course Notes, http://cs.au.dk/ ivan/CPT.html, 2010.

[EG85] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Proceedings of CRYPTO 84 on Advances in Cryptology*, pages 10–18, New York, NY, USA, 1985. Springer-Verlag New York, Inc.

[EG14] Alex Escala and Jens Groth. Fine-tuning Groth-Sahai proofs. In *Public-Key Cryptography - PKC 2014 - 17th International Conference on Practice and Theory in Public-Key Cryptography, Buenos Aires, Argentina, March 26-28, 2014. Proceedings*, pages 630–649, 2014.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, pages 186–194, 1986.

[FS90] Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 416–426, 1990.

[GMR85] S Goldwasser, S Micali, and C Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, STOC '85, pages 291–304, New York, NY, USA, 1985. ACM.

[Gol01] Oded Goldreich. *The Foundations of Cryptography - Volume 1, Basic Techniques*. Cambridge University Press, 2001.

[GPS08] Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[Oka92] Tatsuaki Okamoto. 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*, pages 31–53, 1992.

[Oka06] Tatsuaki Okamoto. Efficient blind and partially blind signatures without random oracles. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, pages 80–99, 2006.

[OT13] Tatsuaki Okamoto and Katsuyuki Takashima. Decentralized attribute-based signatures. In *Public-Key Cryptography - PKC 2013 - 16th International Conference on Practice and Theory in Public-Key Cryptography, Nara, Japan, February 26 - March 1, 2013. Proceedings*, pages 125–142, 2013.

[Ped91] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*, pages 129–140, 1991.

[SNF11] Amang Sudarsono, Toru Nakanishi, and Nobuo Funabiki. Efficient proofs of attributes in pairing-based anonymous credential system. In *Privacy Enhancing Technologies - 11th International Symposium, PETS 2011, Waterloo, ON, Canada, July 27-29, 2011. Proceedings*, pages 246–263, 2011.

[TF12] Isamu Teranishi and Jun Furukawa. Anonymous credential with attributes certification after registration. *IEICE Transactions*, 95-A(1):125–137, 2012.

Appendices

A Instantiation

We discuss an instantiation of our generic authentication scheme $\mathsf{a\text{-}auth}$ that was given in Section 4. Basically, we can employ any three building blocks that satisfy the requirements stated in Section 4. Below we briefly mention an instantiation in the setting of bilinear groups. We put the symbol pp at the subscript of the algorithms to note that the set of public parameters pp is used in the transition functions of the underlying Turing machines that correspond to the algorithms.

The three building blocks are the pairing version of the Camenisch-Lysyanskaya digital signature scheme $\mathsf{Sig}^{\mathsf{CL}}$ (See Appendix C) [Oka06,SNF11,TF12], the pairing version of the Camenisch-Lysyanskaya perfectly witness-indistinguishable argument of knowledge system Π^{CL} (See Appendix D) [Oka06,SNF11,TF12], and the Pedersen-Okamoto commit-and-prove scheme $\mathsf{CmtPrv}^{\mathsf{PO}}$ (See Appendix E) which is a combination of the perfectly hiding commitment scheme $\mathsf{Cmt}^{\mathsf{Ped}}$ of Pedersen [Ped91] and the perfectly witness-indistinguishable argument of knowledge system Π^{Oka} by Okamoto [Oka92]. The five algorithms of our $\mathsf{a\text{-}auth}$ are instantiated as follows. (Also see Fig. 3.)

• $\mathsf{Setup}(1^\lambda) \to pp$. On input the security parameter 1^λ, this PPT algorithm executes the setup algorithm $\mathsf{Sig}^{\mathsf{CL}}.\mathsf{Setup}$. That is, it executes the group generation algorithm $\mathcal{BG}$ to generate bilinear groups of a prime order p of length $|p| = \lambda$: $\mathcal{BG}(1^\lambda) \to \Lambda := (p, e, \mathbb{G}, \tilde{\mathbb{G}}, \mathbb{G}_T, G, \tilde{G})$. Here $e : \mathbb{G} \times \tilde{\mathbb{G}} \to \mathbb{G}_T$ is a bilinear map and $G \in_R \mathbb{G}, \tilde{G} \in_R \tilde{\mathbb{G}}$ with $e(G, \tilde{G}) \neq 1_{\mathbb{G}_T}$ are the generators, respectively. Then it chooses a set of base elements for $\mathsf{Sig}^{\mathsf{CL}}$ and $\mathsf{CmtPrv}^{\mathsf{PO}}$ as $G_0, G_1, G_2, H \in_R \mathbb{G}$, $\tilde{G}_0 \in_R \tilde{\mathbb{G}}$. It returns the set of public parameter

$pp := (\Lambda, G_0, G_1, G_2, H, \tilde{G}_0)$. Note that, in the case of $\mathsf{Sig}^{\mathsf{CL}}$, the setup algorithm $\mathsf{Sig}^{\mathsf{CL}}.\mathsf{Setup}$ is also the setup algorithm $\Pi^{\mathsf{CL}}.\mathsf{Setup}$. As for $\mathsf{CmtPrv}^{\mathsf{PO}}.\mathsf{Setup}$, we use the group $\mathbb{G}$ for $\mathsf{CmtPrv}^{\mathsf{PO}}$.

• $\mathsf{AuthKG}_{pp}(1^\lambda, a) \to (\mathrm{PK}^a, \mathrm{MSK}^a)$. On input pp and an authority index a, this PPT algorithm chooses exponents $\alpha_a \in_R \mathbb{Z}_p$ and computes $\tilde{G}_{a,1} := \tilde{G}_0^{\alpha_a}$. It sets $\mathrm{PK}^a := \tilde{G}_{a,1}, \mathrm{MSK}^a := \alpha_a$. It returns $(\mathrm{PK}^a, \mathrm{MSK}^a)$.

• $\mathsf{PrivKG}_{pp}(\mathrm{PK}^a, \mathrm{MSK}^a, \mathtt{i}) \to \mathrm{sk}^a_{\mathtt{i}}$. On input $\mathrm{PK}^a, \mathrm{MSK}^a$ and a string $\mathtt{i} \in \mathbb{Z}_p$, this PPT algorithm generates a CL signature on $\mathtt{i}$. That is, it chooses an exponent $\gamma_a, \delta_a \in_R \mathbb{Z}_p$ and computes $\Gamma_a := (G_0 G_1^{\mathtt{i}} G_2^{\gamma_a})^{1/(\delta_a+\alpha_a)}$. (We omit the index $\mathtt{i}$ for simplicity.) It sets $\mathrm{sk}^a_{\mathtt{i}} := (\Gamma_a, \gamma_a, \delta_a)$. The right-hand side is a CL signature on $\mathtt{i}$. It returns $\mathrm{sk}^a_{\mathtt{i}}$.

• $\langle \mathsf{P}_{pp}((\mathrm{PK}^a)^{a\in A}, (\mathrm{sk}^a_{\mathtt{i}})^{a\in A}), \mathsf{V}_{pp}((\mathrm{PK}^a)^{a\in A})\rangle \to 1/0$. P_{pp}, the prover, and V_{pp}, the verifier, take a common input $(\mathrm{PK}^a)^{a\in A}$. P_{pp} also takes as input her set of private keys $(\mathrm{sk}^a_{\mathtt{i}})^{a\in A}$ that are signatures on $\mathtt{i}$. These PPT interactive algorithms execute the following protocol of an argument system. Note here that the statement is $x := (x^a)^{a\in A} := (\mathrm{PK}^a)^{a\in A}$, and the witness is $w := (w_0, (w_1^a)^{a\in A}) := (\mathtt{i}, (\mathrm{sk}^a_{\mathtt{i}})^{a\in A})$. Thus, $w^a = (w_0, w_1^a)$. Hence, the predicate and relation for each $a \in A$ are the following ones.

$$\Phi^a_{pp}(x^a, w^a) \stackrel{\mathrm{def}}{=} \big(e(G_0 G_1^{\mathtt{i}} G_2^{\gamma_a}, \tilde{G}_0) =_? e(\Gamma_a, \tilde{G}_0^{\delta_a}\tilde{G}_{a,1})\big), \tag{69}$$

$$R^a \stackrel{\mathrm{def}}{=} \{(x^a, w^a) \in \tilde{\mathbb{G}} \times \big(\mathbb{Z}_p \times \mathbb{G} \times \mathbb{Z}_p^2\big) \mid \Phi^a_{pp}(x^a, w^a) = \mathrm{TRUE}\}. \tag{70}$$

• $\Sigma^{\mathsf{CL},a\in A}_{\mathsf{PO},\mathrm{com}}, \Sigma^{\mathsf{CL},a}_{\mathsf{PO},0,\mathrm{com}}$. To start the interactive argument, P_{pp} first chooses a randomness $u \in_R \mathbb{Z}_p$ and compute a Pedersen commitment to $\mathtt{i}$ as $C_0 := G^{\mathtt{i}} H^u$. P_{pp} puts a statement and the witness as $c_0 := C_0, (w_0, r_0) := (\mathtt{i}, u)$. Hence, the additional predicate and relation needed for our bundled witnesses are the following ones.

$$\Phi_{0,pp}(c_0, (w_0, r_0)) \stackrel{\mathrm{def}}{=} \big(C_0 =_? G^{\mathtt{i}} H^u\big), \tag{71}$$

$$R_0 \stackrel{\mathrm{def}}{=} \{(c_0, (w_0, r_0)) \in \mathbb{G} \times \mathbb{Z}_p^2 \mid \Phi_{0,pp}(c_0, (w_0, r_0)) = \mathrm{TRUE}\}. \tag{72}$$

Then the relation R_0^a for *simultaneous satisfiability* in the instantiation is the following one.

$$R_0^a := \left\{(x_0^a = (x^a, c_0), w_0^a = (w_0, w_1^a, r_0)) \mid \begin{cases}\Phi^a_{pp}(x^a, (w_0, w_1^a)) = \mathrm{TRUE} \text{ and} \\ \Phi_{0,pp}(c_0, (w_0, r_0)) = \mathrm{TRUE}\end{cases}\right\},\ a \in A. \tag{73}$$

Now, we show that the *simultaneous* Σ-protocol Σ_0^a is *actually constructed* as follows.

P_{pp} for each $a \in A$ chooses $v_a \in_R \mathbb{Z}_p$ and re-randomize the secret element Γ_a as $R_a := \Gamma_a G_2^{v_a}$, and puts $z_a := \gamma_a + v_a\delta_a$. Then P_{pp} chooses $r_{a,\mathtt{i}}, r_{a,z}, r_{a,v}, r_{a,\delta} \in_R \mathbb{Z}_p$ and computes $T_a := e(G_1, \tilde{G}_0)^{r_{a,\mathtt{i}}} \cdot e(G_2, \tilde{G}_0)^{r_{a,z}} e(G_2, \tilde{G}_1)^{r_{a,v}} e(R_a, \tilde{G}_0)^{-r_{a,\delta}}$. Besides, P_{pp} chooses an exponent $r_{a,u} \in_R \mathbb{Z}_p$ and computes $C_a := G^{r_{a,\mathtt{i}}} H^{r_{a,u}}$. P_{pp} sends the commitment message $(C_0, (R_a, T_a, C_a)^{a\in A})$ to V_{pp}. We emphasize that the randomness $r_{a,\mathtt{i}}$ is *commonly used for the both* Π^{CL} *and* Π^{Oka}.

• $\Sigma^{\mathsf{CL},a\in A}_{\mathsf{PO},\mathrm{cha}}, \Sigma^{\mathsf{CL},a}_{\mathsf{PO},0,\mathrm{cha}}$. V_{pp} computes a challenge message by choosing an exponent $c \in_R \mathbb{Z}_p$. V_{pp} sends c to P_{pp}.

• $\Sigma^{\mathsf{CL},a\in A}_{\mathsf{PO},\mathrm{res}}, \Sigma^{\mathsf{CL},a}_{\mathsf{PO},0,\mathrm{res}}$. P_{pp} computes the response message as $s_{a,\mathtt{i}} := r_{a,\mathtt{i}} + c\mathtt{i}, s_{a,z} := r_{a,z} + cz_a, s_{a,v} := r_{a,v} + cv_a, s_{a,\delta} := r_{a,\delta} + c\delta_a, s_{a,u} := r_{a,u} + cu$. P_{pp} sends $(s_{a,\mathtt{i}}, s_{a,z}, s_{a,v}, s_{a,\delta}, s_{a,u})^{a\in A}$ to V_{pp}.

• $\Sigma^{\mathsf{CL},a\in A}_{\mathsf{PO},\mathrm{vrf}}, \Sigma^{\mathsf{CL},a}_{\mathsf{PO},0,\mathrm{vrf}}$. V_{pp} checks whether the following equalities hold. If those hold, then return 1. Otherwise, 0.

$$\text{For } a \in A: \tag{74}$$

$$e(G_1, \tilde{G}_0)^{s_{a,\mathtt{i}}} e(G_2, \tilde{G}_0)^{s_{a,z}} e(G_2, \tilde{G}_{a,1})^{s_{a,v}} e(R_a, \tilde{G}_0)^{-s_{a,\delta}} =_? T_a(e(R_a, \tilde{G}_{a,1})/e(G_0, \tilde{G}_0))^c, \text{ and} \tag{75}$$

$$G^{s_{a,\mathtt{i}}} H^{s_{a,u}} =_? C_a C_0^c. \tag{76}$$

We again emphasize that the response $s_{a,\mathtt{i}}$ is *commonly used for the both* Π^{CL} *and* Π^{Oka}. This is why the property (19) is assured in the instantiation.

We omit the description of the knowledge extractor $\Sigma^{\mathsf{CL},a}_{\mathsf{PO},0,\mathrm{ext}}$ and the simulator $\Sigma^{\mathsf{CL},a}_{\mathsf{PO},0,\mathrm{sim}}$. (See Appendix D and E and the previous work cited there.)

As has been explained at (18), we have to show that the $\Sigma^{\mathsf{CL},a}_{\mathsf{PO},0}$ is actually a Σ-protocol and $\Pi^{\mathsf{CL},a}_{\mathsf{PO},0}$ is perfectly witness indistinguishable.

The following propositions hold.

$\mathsf{Setup}(1^\lambda)$	$\mathsf{AuthKG}_{pp}(a)$	$\mathsf{PrivKG}_{pp}(\mathrm{PK}^a, \mathrm{MSK}^a, \mathtt{i})$
$\Lambda := (p, e, \mathbb{G}, \tilde{\mathbb{G}}, \mathbb{G}_T, G, \tilde{G}) \leftarrow \mathcal{BG}(1^\lambda)$	$\alpha_a \in_R \mathbb{Z}_p, \tilde{G}_{a,1} := \tilde{G}_0^{\alpha_a}$	$\gamma_a, \delta_a \in_R \mathbb{Z}_p$
$G_0, G_1, G_2, H \in_R \mathbb{G}, \tilde{G}_0 \in_R \tilde{\mathbb{G}}$	$\mathrm{PK}^a := \tilde{G}_{a,1}, \mathrm{MSK}^a := \alpha_a$	$\Gamma_a := (G_0 G_1^{\mathtt{i}} G_2^{\gamma_a})^{1/(\delta_a + \alpha_a)}$
$pp := (\Lambda, G_0, G_1, G_2, H, \tilde{G}_0)$	Return $(\mathrm{PK}^a, \mathrm{MSK}^a)$	$\mathrm{sk}_{\mathtt{i}}^a := (\Gamma_a, \gamma_a, \delta_a)$
Return pp		Return $\mathrm{sk}_{\mathtt{i}}^a$

$\mathsf{P}_{pp}((\mathrm{PK}^a)^{a \in A}, (\mathrm{sk}_{\mathtt{i}}^a)^{a \in A})$		$\mathsf{V}_{pp}((\mathrm{PK}^a)^{a \in A})$
$u \in_R \mathbb{Z}_p, C_0 := G^{\mathtt{i}} H^u$		
For $a \in A$:		
$v_a \in_R \mathbb{Z}_p, R_a := \Gamma_a G_2^{v_a}, z_a := \gamma_a + v_a \delta_a$		
$r_{a,\mathtt{i}}, r_{a,z}, r_{a,v}, r_{a,\delta} \in_R \mathbb{Z}_p$		
$T_a := e(G_1, \tilde{G}_0)^{r_{a,\mathtt{i}}} e(G_2, \tilde{G}_0)^{r_{a,z}}$		
$\cdot e(G_2, \tilde{G}_{a,1})^{r_{a,v}} e(R_a, \tilde{G}_0)^{-r_{a,\delta}}$		
$r_{a,u} \in_R \mathbb{Z}_p, C_a := G^{r_{a,\mathtt{i}}} H^{r_{a,u}}$	$C_0, (R_a, T_a, C_a)^{a \in A}$ $\rightarrow$	
For $a \in A$:	c $\leftarrow$	$c \in_R \mathbb{Z}_p$
$s_{a,\mathtt{i}} := r_{a,\mathtt{i}} + c\mathtt{i}, s_{a,z} := r_{a,z} + cz_a$		
$s_{a,v} := r_{a,v} + cv_a, s_{a,\delta} := r_{a,\delta} + c\delta_a$		
$s_{a,u} := r_{a,u} + cu$	$(s_{a,\mathtt{i}}, s_{a,z}, s_{a,v}, s_{a,\delta}, s_{a,u})^{a \in A}$ $\rightarrow$	For $a \in A$:
		$e(G_1, \tilde{G}_0)^{s_{a,\mathtt{i}}} e(G_2, \tilde{G}_0)^{s_{a,z}}$
		$\cdot e(G_2, \tilde{G}_{a,1})^{s_{a,v}} e(R_a, \tilde{G}_0)^{-s_{a,\delta}}$
		$=_? T_a (e(R_a, \tilde{G}_{a,1}) / e(G_0, \tilde{G}_0))^c$
		and $G^{s_{a,\mathtt{i}}} H^{s_{a,u}} =_? C_a C_0^c$
		If all eqs. hold, Return 1
		else Return 0

Fig. 3. Instantiation of our decentralized multi-authority anonymous authentication scheme a-auth in the setting of bilinear groups.

Proposition 4 *$\Sigma_{\mathsf{PO},0}^{\mathsf{CL},a}$ is a Σ-protocol.*

Proof. Σ_0^a is a protocol with the common randomness $r_{a,\mathtt{i}}$ and the common response message $s_{a,\mathtt{i}}$ in the two Σ-protocols Σ^a and Σ_0. Therefore, in each parallel execution for each $a \in A$, the knowledge extractor Σ_{ext}^a and $\Sigma_{0,\mathrm{ext}}$ extract the same exponent $\hat{\mathtt{i}}^a$ with probability one. Moreover, all the extracted exponents $\hat{\mathtt{i}}^a, a \in A$ are equal to a single exponent $\hat{\mathtt{i}}$ with an overwhelming probability owing to the the binding property of the Pedersen commitment, which is under the discrete logarithm assumption on $\mathbb{G}$. □

Proposition 5 *$\Pi_{\mathsf{PO},0}^{\mathsf{CL},a}$ is perfectly witness indistinguishable.*

Proof. The distribution $dist\big((C_0, (R_a, T_a, C_a)^{a \in A}), c, (s_{a,\mathtt{i}}, s_{a,z}, s_{a,v}, s_{a,\delta}, s_{a,u})^{a \in A}\big)$ is identical even if the distribution of CHA deviates from the uniform random distribution. This is because $\Pi^{\mathsf{CL},a}, a \in A$, and Π^{Oka} are perfectly witness indistinguishable. □

Then we obtain the following theorems.

Theorem 5 *If $\mathsf{Cmt}^{\mathsf{Ped}}$ is perfectly hiding and computationally binding, and if $\mathsf{Sig}^{\mathsf{CL}}$ is existentially unforgeable against adaptive chosen-message attacks, then our a-auth is secure against concurrent and collusion attacks.*

Proof. Proposition 4, Proposition 5 and Theorem 3 assure the claim. □

Theorem 6 *Our a-auth has anonymity.*

Proof. $\mathsf{Sig}^{\mathsf{CL}}$ is perfectly witness-indistinguishable and $\mathsf{CmtPrv}^{\mathsf{PO}}$ is perfectly hiding. Theorem 4 assures the claim. □

B Algebraic Settings and Number-Theoretic Assumptions

Let $(p, \mathbb{G})$ denote a cyclic group of prime order p, where $|p| = \lambda$. Let G denote a generator chosen uniformly at random, $G \in_R \mathbb{G}\backslash\{1_{\mathbb{G}}\}$. Let $\mathcal{G}$ denote a PPT algorithm which, on input 1^λ, returns the set of parameters $\Lambda := (p, \mathbb{G}, G)$. That is, $\Lambda := (p, \mathbb{G}, G) \leftarrow \mathcal{G}(1^\lambda)$.

Let $(p, e, \mathbb{G}, \tilde{\mathbb{G}}, \mathbb{G}_T)$ denote bilinear groups of prime order p and of Type 3 [GPS08,BB08], where $|p| = \lambda$. Here we require that the bilinear map $e : \mathbb{G} \times \tilde{\mathbb{G}} \to \mathbb{G}_T$ is efficiently computable (i.e., polynomial-time in λ). Let G and $\tilde{G}$ denote generators chosen uniformly at random, $G \in_R \mathbb{G}\backslash\{1_{\mathbb{G}}\}, \tilde{G} \in_R \tilde{\mathbb{G}}\backslash\{1_{\tilde{\mathbb{G}}}\}$ with $e(G, \tilde{G}) \neq 1_{\mathbb{G}_T}$. Let $\mathcal{BG}$ denote a PPT algorithm which, on input 1^λ, returns the set of parameters $\Lambda := (p, e, \mathbb{G}, \tilde{\mathbb{G}}, \mathbb{G}_T, G, \tilde{G})$. That is, $\Lambda := (p, e, \mathbb{G}, \tilde{\mathbb{G}}, \mathbb{G}_T, G, \tilde{G}) \leftarrow \mathcal{BG}(1^\lambda)$. Bilinear groups are widely recognized in the form of the pairing on elliptic curves [GPS08].

B.1 Discrete Logarithm Assumption (DL) [EG85]

The DL assumption is stated as follows. For any PPT algorithm $\mathbf{S}$, the advantage of $\mathbf{S}$ over $\mathcal{G}$ defined by the following equality is negligible in λ:

$$\mathbf{Adv}^{\mathrm{dl}}_{\mathcal{G},\mathbf{S}}(\lambda) := \Pr[\gamma = \gamma^* \mid \Lambda \leftarrow \mathcal{G}(1^\lambda), \gamma \in_R \mathbb{Z}_p, \gamma^* \leftarrow \mathbf{S}(\Lambda, G, G^\gamma)]. \tag{77}$$

The probability is taken over the random tape of $\mathcal{G}$, the uniform random sampling of γ, and the random tape of $\mathbf{S}$.

B.2 Strong Diffie-Hellman Assumption (SDH) [BB04]

The SDH assumption is stated as follows. Let q be a natural number that is a function of λ bounded by a polynomial in λ. For any PPT algorithm $\mathbf{S}$ and for any q, the advantage of $\mathbf{S}$ over $\mathcal{BG}$ defined by the following equality is negligible in λ:

$$\mathbf{Adv}^{\mathrm{sdh}}_{\mathcal{BG},\mathbf{S}}(\lambda) := \Pr[\Gamma^{\gamma+e} = G \mid \Lambda \leftarrow \mathcal{BG}(1^\lambda), \gamma \in_R \mathbb{Z}_p, (\Gamma, e) \leftarrow \mathbf{S}(\Lambda, (\tilde{G}^\gamma, \tilde{G}^{\gamma^2}, \ldots, \tilde{G}^{\gamma^q}))]. \tag{78}$$

The probability is taken over the random tape of $\mathcal{G}$, the uniform random sampling of γ, and the random tape of $\mathbf{S}$.

C Camenisch-Lysyanskaya Signatures, Pairing Version [Oka06,SNF11,TF12]

The pairing version of the Camenisch-Lysyanskaya signature scheme $\mathsf{Sig}^{\mathsf{CL}}$, which was originally in the RSA setting, was proposed by Okamoto [Oka06]. We summarize the digital signature scheme here in the form which is found in Sudarsono-Nakanishi-Funabiki [SNF11] and Teranishi and Furukawa [TF12]. $\mathsf{Sig}^{\mathsf{CL}}$ consists of four PPT algorithms, $\mathsf{Sig}^{\mathsf{CL}} := (\mathsf{Sig}^{\mathsf{CL}}.\mathsf{Setup}, \mathsf{Sig}^{\mathsf{CL}}.\mathsf{KG}_{pp}, \mathsf{Sig}^{\mathsf{CL}}.\mathsf{Sign}_{pp}, \mathsf{Sig}^{\mathsf{CL}}.\mathsf{Vrf}_{pp})$.

• $\mathsf{Sig}^{\mathsf{CL}}.\mathsf{Setup}(1^\lambda) \to pp$. On input the security parameter 1^λ, this PPT algorithm generates a set of public parameter. That is, it executes a group generation algorithm $\mathcal{BG}$ to generate bilinear groups of a prime order p of length $|p| = \lambda$: $\Lambda := (p, e, \mathbb{G}, \tilde{\mathbb{G}}, \mathbb{G}_T, G, \tilde{G}) \leftarrow \mathcal{BG}(1^\lambda)$. Besides, it chooses a set of base elements of $G_0, G_1, G_2 \in_R \mathbb{G}, \tilde{G}_0 \in_R \tilde{\mathbb{G}}$. It returns $pp := (\lambda, G_0, G_1, G_2, \tilde{G}_0)$.

• $\mathsf{Sig}^{\mathsf{CL}}.\mathsf{KG}_{pp}(1^\lambda) \to (\mathrm{PK}, \mathrm{SK})$. On input 1^λ this PPT algorithm chooses an exponent $\alpha \in_R \mathbb{Z}_p$ and computes $\tilde{G}_1 := \tilde{G}_0^\alpha$. It sets a public key and the corresponding secret key as $\mathrm{PK} := \tilde{G}_1, \mathrm{SK} := \alpha$, respectively. It returns $(\mathrm{PK}, \mathrm{SK})$.

• $\mathsf{Sig}^{\mathsf{CL}}.\mathsf{Sign}_{pp}(\mathrm{PK}, \mathrm{SK}, m) \to \sigma$. On input PK, SK and a message $m \in \mathbb{Z}_p$, this PPT algorithm chooses two randomnesses $\gamma, \delta \in \mathbb{Z}_p$. It computes $\Gamma := (G_0 G_1^m G_2^\gamma)^{1/(\delta+\alpha)}$. It sets a signature $\sigma := (\Gamma, \gamma, \delta)$. It returns σ.

• $\mathsf{Sig}^{\mathsf{CL}}.\mathsf{Vrf}_{pp}(\mathrm{PK}, m, \sigma) \to 1/0$. On input PK, m and σ, this deterministic polynomial time algorithm returns a boolean decision 1 if the following holds, and otherwise 0: $e(G_0 G_1^m G_2^\gamma) =_? e(\Gamma, \tilde{G}_0^\delta \tilde{G}_1)$.

The pairing version of the Camenisch-Lysyanskaya signature scheme $\mathsf{Sig}^{\mathsf{CL}}$ is known to be existentially unforgeable against adaptive chosen-message attacks under the Strong Diffie-Hellman assumption on $\mathcal{BG}$ (see Appendix B.2) [Oka06,SNF11,TF12].

D Camenisch-Lysyanskaya WIAoK, Pairing Version [Oka06,SNF11,TF12]

The pairing version of the Camenisch-Lysyanskaya argument of knowledge system Π^{CL}, which was originally in the RSA setting, was first proposed by Okamoto [Oka06]. We summarize the argument system here in the form found in Sudarsono-Nakanishi-Funabiki [SNF11] and Teranishi and Furukawa [TF12]. $\Pi^{\mathsf{CL}} = (\Pi^{\mathsf{CL}}.\mathsf{Setup}, \mathsf{P}_{pp}, \mathsf{V}_{pp})$ is executed in accordance with a Σ-protocol $\Sigma^{\mathsf{CL}} = (\Sigma^{\mathsf{CL}}_{\mathrm{com}}, \Sigma^{\mathsf{CL}}_{\mathrm{cha}}, \Sigma^{\mathsf{CL}}_{\mathrm{res}}, \Sigma^{\mathsf{CL}}_{\mathrm{vrf}}, \Sigma^{\mathsf{CL}}_{\mathrm{ext}}, \Sigma^{\mathsf{CL}}_{\mathrm{sim}})$.

The setup algorithm $\Pi^{\mathsf{CL}}.\mathsf{Setup}$ is the same as $\mathsf{Sig}^{\mathsf{CL}}.\mathsf{Setup}$. The set of public parameter pp is common.

For $\alpha \in_R \mathbb{Z}_p$, the statement is $x := \tilde{G}_1 := \tilde{G}_0^{\alpha}$. For a given string $\mathtt{i} \in \mathbb{Z}_p$, choose two randomnesses $\gamma, \delta \in \mathbb{Z}_p$ and compute $\Gamma := (G_0 G_1^{\mathtt{i}} G_2^{\gamma})^{1/(\delta+\alpha)}$. The witness of the statement x is $w := (\mathtt{i}, \Gamma, \gamma, \delta)$. Note that $\sigma := (\Gamma, \gamma, \delta)$ *is a Camenisch-Lysyanskaya signature on the message* $\mathtt{i}$. The following key equation holds.

$$e(G_0 G_1^{\mathtt{i}} G_2^{\gamma}, \tilde{G}_0) = e(\Gamma, \tilde{G}_0^{\delta} \tilde{G}_1). \tag{79}$$

It is notable that the statement x does not include any information on the witness w, and the number of elements in $W(x)$ is p^3 because there are three independent variable in w; that is, $(\mathtt{i}, \gamma, \delta)$. In other words, this number is the number of the solutions of the *equation* (79) determined by pp and x.

The protocol between P_{pp} and V_{pp} is a Σ-protocol. It goes as follows.

- $\Sigma^{\mathsf{CL}}_{\mathrm{com}}(x, w) \to (\textsc{com}, St)$. This PPT algorithm is executed by P_{pp}. On input a statement x and a witness w, it chooses $v \in_R \mathbb{Z}_p$ and re-randomize the secret element Γ as $R := \Gamma G_2^{v}$. It puts $z := \gamma + v\delta$. It chooses $r_{\mathtt{i}}, r_z, r_v, r_\delta \in_R \mathbb{Z}_p$ and computes $T := e(G_1, \tilde{G}_0)^{r_{\mathtt{i}}} e(G_2, \tilde{G}_0)^{r_z} e(G_2, \tilde{G}_1)^{r_v} e(R, \tilde{G}_0)^{-r_\delta}$. It puts the commitment message as $\textsc{com} := (R, T)$. It returns $\textsc{com}$ and its inner state St. P_{pp} sends $\textsc{com}$ to V_{pp}. Note that the following equality holds after the re-randomization.

$$e(G_1, \tilde{G}_0)^{\mathtt{i}} e(G_2, \tilde{G}_0)^{z} e(G_2, \tilde{G}_1)^{v} e(R, \tilde{G}_0)^{-\delta} = e(R, \tilde{G}_1)/e(G_0, \tilde{G}_0). \tag{80}$$

- $\Sigma^{\mathsf{CL}}_{\mathrm{cha}}(x) \to \textsc{cha}$. This PPT algorithm is executed by V_{pp}. On input the statement x, it reads out the size of the security parameter as 1^λ and chooses a challenge message $c \in_R \textsc{chaSp}(1^\lambda)$. It puts the challenge message as $\textsc{cha} := c$. It returns $\textsc{cha}$. V_{pp} sends $\textsc{cha}$ to P_{pp}.
- $\Sigma^{\mathsf{CL}}_{\mathrm{res}}(St, \textsc{cha}) \to \textsc{res}$. This PPT algorithm is executed by P_{pp}. On input the state St^a and the challenge message $\textsc{cha}$, it computes $s_{\mathtt{i}} := r_{\mathtt{i}} + c\mathtt{i}, s_z := r_z + cz, s_v := r_v + cv, s_\delta := r_\delta + c\delta$. It sets the response message as $\textsc{res} := (s_{\mathtt{i}}, s_z, s_v, s_\delta)$. It returns $\textsc{res}$. P_{pp} sends $\textsc{res}$ to V_{pp}.
- $\Sigma^{\mathsf{CL}}_{\mathrm{vrf}}(x, \textsc{com}, \textsc{cha}, \textsc{res}) \to d$. This deterministic polynomial-time algorithm is executed by V_{pp}. On input the statement x and all the messages $(\textsc{com}, \textsc{cha}, \textsc{res})$, it checks whether the following equality holds. If it holds, then return 1 ("accept"), and otherwise, 0 ("reject").

$$e(G_1, \tilde{G}_0)^{s_{\mathtt{i}}} e(G_2, \tilde{G}_0)^{s_z} e(G_2, \tilde{G}_1)^{s_v} e(R, \tilde{G}_0)^{-s_\delta} =_? T(e(R, \tilde{G}_1)/e(G_0, \tilde{G}_0))^{c}. \tag{81}$$

For the remaining two, $\Sigma^{\mathsf{CL}}_{\mathrm{ext}}$ and $\Sigma^{\mathsf{CL}}_{\mathrm{sim}}$, see [SNF11,TF12]. The protocol Σ^{CL} is known to be a Σ-protocol.

Π^{CL} is *perfectly witness-indistinguishable* [FS90]. This is because the distribution of transcripts is independent of the witness $w \in W(x)$ even if the distribution of $\textsc{cha}$ deviates from the uniform random distribution.

E Pedersen-Okamoto Commitment-and-Prove Scheme [Ped91,Oka92]

The Pedersen commitment scheme [Ped91] $\mathsf{Cmt}^{\mathsf{Ped}}$ is a commitment scheme in the discrete logarithm setting. $\mathsf{Cmt}^{\mathsf{Ped}}$ consists of three PPT algorithms, $\mathsf{Cmt}^{\mathsf{Ped}} = (\mathsf{Cmt}^{\mathsf{Ped}}.\mathsf{Setup}, \mathsf{Cmt}^{\mathsf{Ped}}.\mathsf{Com}_{pp}, \mathsf{Cmt}^{\mathsf{Ped}}.\mathsf{Vrf}_{pp})$.

- $\mathsf{Cmt}^{\mathsf{Ped}}.\mathsf{Setup}(1^\lambda) \to pp$. On input the security parameter 1^λ, this PPT algorithm generates a set of public parameter. That is, it executes a group generation algorithm $\mathcal{G}$ to generate a cyclic group of a prime order p of length $|p| = \lambda$: $\Lambda := (p, \mathbb{G}, G) \leftarrow \mathcal{G}(1^\lambda)$. In addition, it chooses $\rho \in_R \mathbb{Z}_p$ and computes $H := G^{\rho}$. It returns $pp := (p, \mathbb{G}, G, H)$.
- $\mathsf{Cmt}^{\mathsf{Ped}}.\mathsf{Com}_{pp}(m) \to (C, \kappa)$. On input a message $m \in \mathbb{Z}_p$, this PPT algorithm generates a commitment $c \in \mathbb{G}$ and an opening key $\kappa \in \mathbb{Z}_p$. That is, it chooses $u \in_R \mathbb{Z}_p$ and computes the commitment $C = G^m H^u$ to m, and it sets κ as $\kappa := u$. It returns (C, κ).

• $\mathsf{Cmt}^{\mathsf{Ped}}.\mathsf{Vrf}_{pp}(C, m, \kappa) \to d$. On input C, m and κ, this deterministic polynomial-time algorithm generates a boolean decision d. That is, it checks whether $C = G^m H^\kappa$ holds or not. If it holds, then it returns $d := 1$, and otherwise, $d := 0$.

$\mathsf{Cmt}^{\mathsf{Ped}}$ is *perfectly hiding*. the distribution of the commitment C is independent of the committed message m. $\mathsf{Cmt}^{\mathsf{Ped}}$ is *computationally binding* under the discrete logarithm assumption on $\mathcal{G}$ (see Appendix B.1). If a commitment C is opened in two different ways $(m, \kappa) \neq (m', \kappa')$ with non-negligible probability in λ, then a PPT algorithm $\mathbf{S}$ is constructed and it solves instances of the discrete logarithm problem, $H = G^\rho$. with a non-negligible probability in λ.

The Okamoto interactive argument system $\Pi^{\mathsf{Oka}} = (\Pi^{\mathsf{Oka}}.\mathsf{Setup}, \mathsf{P}_{pp}, \mathsf{V}_{pp})$ [Oka92] is executed in accordance with a Σ-protocol $\Sigma^{\mathsf{Oka}} = (\Sigma^{\mathsf{Oka}}_{\mathrm{com}}, \Sigma^{\mathsf{Oka}}_{\mathrm{cha}}, \Sigma^{\mathsf{Oka}}_{\mathrm{res}}, \Sigma^{\mathsf{Oka}}_{\mathrm{vrf}}, \Sigma^{\mathsf{Oka}}_{\mathrm{ext}}, \Sigma^{\mathsf{Oka}}_{\mathrm{sim}})$.

The setup algorithm $\Pi^{\mathsf{Oka}}.\mathsf{Setup}$ is the same as $\mathsf{Cmt}^{\mathsf{Ped}}.\mathsf{Setup}$. The set of public parameter pp is common.

For $t, u \in_R \mathbb{Z}_p$, the statement is $x := X := G^t H^u$. The witness of x is $w = (t, u)$. It is notable that the number of elements in $W(x)$ is p because there are one independent variable in w; that is, one of t and u. In other words, this number is the number of the solutions of the *equation* $X = G^t H^u$ determined by pp and x.

The protocol between P_{pp} and V_{pp} is a Σ-protocol. It goes as follows.

• $\Sigma^{\mathsf{Oka}}_{\mathrm{com}}(x, w) \to (\textsc{com}, St)$. This PPT algorithm is executed by P_{pp}. On input a statement x and a witness w, it chooses $r_t, r_u \in_R \mathbb{Z}_p$ and computes $C := G^{r_t} H^{r_u}$. It puts the commitment message as $\textsc{com} := C$. It returns COM and its inner state St. P_{pp} sends COM to V_{pp}.

• $\Sigma^{\mathsf{Oka}}_{\mathrm{cha}}(x) \to \textsc{cha}$. This PPT algorithm is executed by V_{pp}. On input the statement x, it reads out the size of the security parameter as 1^λ and chooses a challenge message $c \in_R \textsc{cha}\mathrm{Sp}(1^\lambda)$. It puts the challenge message as $\textsc{cha} := c$. It returns CHA. V_{pp} sends CHA to P_{pp}.

• $\Sigma^{\mathsf{Oka}}_{\mathrm{res}}(St, \textsc{cha}) \to \textsc{res}$. This PPT algorithm is executed by P_{pp}. On input the state St^a and the challenge message CHA, it computes $s_t := r_t + ct, s_u := r_u + cu$. It sets the response message as $\textsc{res} := (s_t, s_u)$. P_{pp} sends RES to V_{pp}.

• $\Sigma^{\mathsf{Oka}}_{\mathrm{vrf}}(x, \textsc{com}, \textsc{cha}, \textsc{res}) \to d$. This deterministic polynomial-time algorithm is executed by V_{pp}. On input the statement x and all the messages $(\textsc{com}, \textsc{cha}, \textsc{res})$, it checks whether the following equality holds: $G^{s_t} H^{s_u} =_? C X^c$.

For the remaining two, $\Sigma^{\mathsf{Oka}}_{\mathrm{ext}}$ and $\Sigma^{\mathsf{Oka}}_{\mathrm{sim}}$, see [Oka92]. The protocol Σ^{Oka} is known to be a Σ-protocol.

Π^{Oka} is *perfectly witness-indistinguishable* [FS90]. This is because the distribution of transcripts is independent of the witness $w \in W(x)$ even if the distribution of CHA deviates from the uniform random distribution.

Combining the Pedersen commitment scheme $\mathsf{Cmt}^{\mathsf{Ped}}$ and the Okamoto interactive argument system Π^{Oka} with the Σ-protocol Σ^{Oka}, we obtain the Pedersen-Okamoto commit-and-prove scheme $\mathsf{CmtPrv}^{\mathsf{PO}} = (\mathsf{CmtPrv}^{\mathsf{PO}}.\mathsf{Setup}, \mathsf{Cmt}^{\mathsf{Ped}} = (\mathsf{Cmt}^{\mathsf{Ped}}.\mathsf{Com}_{pp}, \mathsf{Cmt}^{\mathsf{Ped}}.\mathsf{Vrf}_{pp}), \Pi^{\mathsf{Oka}} = (\mathsf{P}_{pp}, \mathsf{V}_{pp}))$, where the setup algorithm $\mathsf{CmtPrv}^{\mathsf{PO}}.\mathsf{Setup}$ is the same as $\mathsf{Cmt}^{\mathsf{Ped}}.\mathsf{Setup}$ and $\Pi^{\mathsf{Oka}}.\mathsf{Setup}$.