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

Abstract. We propose 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. 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 define a syntax and security notions of the scheme. Then we give 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 common global identity string and a digital signature on it. We mention an instantiation of the generic scheme in the setting of bilinear groups.

Keywords: Interactive argument · Sigma protocol
Witness indistinguishability · Decentralized · Collusion resistance

1 Introduction

Global identities such as Passport Numbers (PNs) or Social Security Numbers (SSNs) in each country 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 to issue an attribute certificate at the registration stage. In the stage, the authority confirms our identities by the global identity string such as PN or SSN. Once the attribute certificate is issued, we become to be accepted at

D. Naccache et al. (Eds.): ICICS 2018, LNCS 11149, pp. 530–547, 2018.
https://doi.org/10.1007/978-3-030-01950-1_31

the authentication stage 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 the 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 the authentication, especially recently. The growth of the internet of things and related big data analysis have protecting privacy more critical to involved users. For the purpose, an authentication framework of identity strings and passwords should be evolved into a framework where anonymity is guaranteed at the authentication stage. For example, when a smart household machine generates a report about the situation of a house via the internet as a query for a useful suggestion (such as air conditioning or cooking recipes), the identity information is often unnecessary. A further example is a connected-to-the-internet vehicle which uses a combination of plural services like local traffic information system and the passenger's web-scheduler. The identity information should not be leaked even when the memberships are needed in the registration stages. In this example a user should be authenticated by the service providers at the same time in the authentication stages, anonymously. This is an authentication framework in which plural attributes of a single user are authenticated. However, there is a threat on anonymous authentication frameworks; *the collusion attack*. A malicious user collects private attribute keys from honest users who have different identities, and tries to make a verifier accept anonymously by using the merged attribute keys. Here the vary anonymity is a critical potential drawback from the view point of the collusion attacks.

1.1 Related Work and Our Contribution

A decentralized multi-authority attribute-based signature scheme (DMA-ABS) [11] is an ABS scheme with decentralized key-issuing authorities. In an ABS scheme, a signer has credentials (i.e. private secret keys) on her attributes, and a message has a signing policy expressed as a boolean formula on attributes. The signer is able to sign it if and only if her attributes satisfies the boolean formula. There are assignment patterns and the attribute privacy of an ABS scheme should assure that the signatures do not leak any information on the satisfying pattern. We note that this property also requires the anonymity of the signer's identity. A non-trivial task in constructing an ABS scheme is to assure *both* the collusion resistance and the attribute privacy. On the other hand, allowing decentralized multi-authorities is to have independent issuers each of which generates each private secret attribute-key to the user.

In this paper, we propose a new notion; 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 [9]. 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. We construct the kind of WIA by employing a commitment scheme as one of the building blocks.

As an application, we give a generic construction of a decentralized multi-authority anonymous authentication scheme, which can be converted into a DMA-ABS scheme by the Fiat-Shamir transform [8]. Actually, if a prover chooses a monotone boolean formula instead of an all-AND formula (as in this paper), and if we apply the Fiat-Shamir transform to the Σ-protocol in our authentication scheme, then we obtain a DMA-ABS scheme.

2 Preliminaries

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 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). The expression $a \in_? S$ returns a boolean 1 when $a \in A$, and otherwise 0. When an algorithm A with input a returns z, we denote it as $z \leftarrow A(a)$, or, $A(a) \rightarrow z$. When a probabilistic polynomial-time (PPT, for short) algorithm A with 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 input a and an algorithm B with input b interact with each other and return 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 $\mathbf{O}_1, \ldots, \mathbf{O}_n$ concurrently, i.e. in arbitrarily interleaved order of messages, we denote it by $A^{\mathbf{O}_i|_{i=1}^n}$. The probability of an event E is denoted by $\Pr[E]$. The conditional probability of an event E given events $F_1, \ldots, F_n$ in this order is denoted by $\Pr[E|F_1, \ldots, F_n]$. The distribution of a random variable X is denoted by $dist(X)$. The distribution of a random variable X whose probability is given by a joint probability of random variables $X, Y_1, \ldots, Y_n$ is denoted by $dist(X|X, Y_1, \ldots, Y_n)$. We say that a probability p is negligible in λ if it is upper-bounded by the inverse of any polynomial $\mathrm{poly}(\lambda)$ of positive coefficients (i.e. $p < 1/\mathrm{poly}(\lambda)$). We say that a probability p is overwhelming in λ if it is lower-bounded by $1 -$ (the inverse of any fixed polynomial $\mathrm{poly}(\lambda)$ of positive coefficients) (i.e. $p > 1 - 1/\mathrm{poly}(\lambda)$).

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 $R \stackrel{\text{def}}{=} \{(x, w) \in (\{0,1\}^*)^2 | \Phi(x, w) = \text{TRUE}\}$. We say that R is polynomially bounded if there exists a polynomial $\ell(\cdot)$ such that $|w| \leq \ell(|x|)$ for

any $(x, w) \in R$. 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. When a set of public parameter values PP are needed to define the predicate (for example, to set up algebraic operations), we denote it as $\Phi_{\mathtt{PP}}$. An NP language L for an NP relation R is defined as the set of all possible statements: $L \stackrel{\text{def}}{=} \{x \in \{0,1\}^*; \exists w \in \{0,1\}^*, (x, w) \in R\}$. We denote the set of witnesses of a statement x by $W(x)$: $W(x) \stackrel{\text{def}}{=} \{w \in \{0,1\}^* \mid (x, w) \in R\}$. We call the union W of all the sets $W(x)$ for $x \in L$ the *witness space* of L: $W \stackrel{\text{def}}{=} \bigcup_{x \in L} W(x)$. We denote an interactive proof system on an NP relation R [1,10] by $\Pi = (\Pi.\mathtt{Setup}, \mathtt{P}, \mathtt{V})$, where $\Pi.\mathtt{Setup}$ is a set up algorithm for a set of public parameter values PP, and P and V are a pair of interactive algorithms. P, which is called a prover, is probabilistic and unbounded, and V, which is 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 [4,5]**.** 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 $\Pi = (\Pi.\mathtt{Setup}, \mathtt{P}, \mathtt{V})$ [4,5]. 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 $\Sigma_{\mathrm{com}}(x, w) \rightarrow (\mathrm{COM}, St)$. The second algorithm Σ_{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 $\mathrm{CHA} \in_R \mathrm{CHASP}(1^\lambda)$ from the challenge space $\mathrm{CHASP}(1^\lambda) := \{0,1\}^{\omega(\lambda)}$, where $\omega(\cdot)$ is a super-log function [2]. It returns the message as $\Sigma_{\mathrm{cha}}(x) \rightarrow \mathrm{CHA}$. The third algorithm Σ_{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 $\Sigma_{\mathrm{res}}(St, \mathrm{CHA}) \rightarrow \mathrm{RES}$. The fourth algorithm Σ_{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 $\Sigma_{\mathrm{vrf}}(x, \mathrm{COM}, \mathrm{CHA}, \mathrm{RES}) \rightarrow d$. 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 $(\mathrm{COM}, \mathrm{CHA}, \mathrm{RES})$ is called a transcript of the interaction on x.

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

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

The fifth algorithm Σ_{ext} concerns with the following property.

Special Soundness. There is a PPT algorithm Σ_{ext} called a *knowledge extractor*, which, on input a statement x and two accepting transcripts with a common commitment message and different challenge messages, $(\mathrm{COM}, \mathrm{CHA}, \mathrm{RES})$ 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 $|x|$: $\hat{w} \leftarrow \Sigma_{\text{ext}}(x, \text{COM}, \text{CHA}, \text{RES}, \text{CHA}', \text{RES}')$.

The sixth algorithm Σ_{sim} concerns with the following property.

Honest-Verifier Zero-Knowledge. There is a PPT algorithm called a *simulator* Σ_{sim}, which, on input a statement x, computes an accepting transcript on x: $(\tilde{\text{COM}}, \tilde{\text{CHA}}, \tilde{\text{RES}}) \leftarrow \Sigma_{\text{sim}}(x)$, where the distribution of the simulated transcripts $dist\big(\tilde{\text{COM}}, \tilde{\text{CHA}}, \tilde{\text{RES}}\big)$ is identical to the distribution of the real accepting transcripts $dist\big(\text{COM}, \text{CHA}, \text{RES}\big)$.

Note 1: Our Use Case. 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 $\Sigma_{\text{sim}}(x)$: 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 $\tilde{\text{RES}}$: $\tilde{\text{CHA}} \in_R \text{CHASP}(1^\lambda)$, $(\tilde{\text{COM}}, \tilde{\text{RES}}) \leftarrow \Sigma_{\text{sim}}(x, \tilde{\text{CHA}})$.

Witness-Indistinguishability [7,9]**.** Let R be an NP relation. Suppose that an interactive argument system $\Pi = (\Pi.\mathtt{Setup}, \mathtt{P}, \mathtt{V})$ 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 $\mathtt{V}^*$, 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 $dist\big(x, z, transc\langle \mathtt{P}(x, w_x), \mathtt{V}^*(x, z)\rangle\big)$ and $dist\big(x, z, transc\langle \mathtt{P}(x, w'_x), \mathtt{V}^*(x, z)\rangle\big)$ are identical.

2.2 Commit-and-Prove Scheme [3,6]

A commit-and-prove scheme $\mathtt{CmtPrv}$ consists of five PPT algorithms: $\mathtt{CmtPrv} = (\mathtt{CmtPrv.Setup}, \mathtt{Cmt} = (\mathtt{Cmt.Com}, \mathtt{Cmt.Vrf}), \Pi = (\mathtt{P}, \mathtt{V}))$.
$\mathtt{CmtPrv.Setup}(1^\lambda) \to \mathtt{PP}$. On input the security parameter 1^λ, it generates a set of public parameter values $\mathtt{PP}$. It returns $\mathtt{PP}$.
$\mathtt{Cmt.Com}(\mathtt{PP}, m) \to (c, \kappa)$. On input the set of public parameter values $\mathtt{PP}$, 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).
$\mathtt{Cmt.Vrf}(\mathtt{PP}, c, m, \kappa) \to d$. On input the set of public parameter values $\mathtt{PP}$, a commitment c, a message m and an opening key κ, this deterministic algorithm generates a boolean decision d. It returns d.

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

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

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

In this paper we focus on the following properties for the commitment part $\mathtt{Cmt}$.

Perfectly Hiding. For any security parameter 1^λ, any set of public parameter values $\mathtt{PP}$ and any two messages $m, m' \in \mathcal{M}sg(1^\lambda)$, the two distributions $dist\big(c \mid (c,\kappa) \leftarrow \mathtt{Cmt.Com}(\mathtt{PP}, m)\big)$ and $dist\big(c \mid (c,\kappa) \leftarrow \mathtt{Cmt.Com}(\mathtt{PP}, m')\big)$ are identical.

Computationally Binding. The attack of breaking binding property of $\mathtt{Cmt}$ by an algorithm $\mathbf{A}$ is defined by the following experiment.

$$\begin{aligned}\mathbf{Exp}^{\mathrm{bind}}_{\mathtt{Cmt},\mathbf{A}}(1^\lambda):\ & \mathtt{PP} \leftarrow \mathtt{CmtPrv.Setup}(1^\lambda), (c, m, \kappa, m', \kappa') \leftarrow \mathbf{A}(\mathtt{PP}) \\ & \text{If } \mathtt{Cmt.Vrf}(\mathtt{PP}, c, m, \kappa) = \mathtt{Cmt.Vrf}(\mathtt{PP}, c, m', \kappa') = 1 \wedge m \neq m', \\ & \quad \text{then Return Win else Return Lose}\end{aligned}$$

The advantage of $\mathbf{A}$ over $\mathtt{Cmt}$ is defined as $\mathbf{Adv}^{\mathrm{bind}}_{\mathtt{Cmt},\mathbf{A}}(\lambda) := \Pr[\mathbf{Exp}^{\mathrm{bind}}_{\mathtt{Cmt},\mathbf{A}}(1^\lambda)$ returns Win]. The commitment scheme $\mathtt{Cmt}$ is said to be *computationally binding* if for any set of public parameter values $\mathtt{PP}$ and any PPT algorithm $\mathbf{A}$, the advantage $\mathbf{Adv}^{\mathrm{bind}}_{\mathtt{Cmt},\mathbf{A}}(\lambda)$ is negligible in λ.

Note 2: Our Use Case. The commitment generation algorithm $\mathtt{Cmt.Com}$ uses random tapes [9]. 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, $\mathtt{Cmt.Com}(\mathtt{PP}, m; r) \rightarrow (c, r)$.

2.3 Digital Signature Scheme [8]

A digital signature scheme $\mathtt{Sig}$ consists of four PPT algorithms: $\mathtt{Sig} = (\mathtt{Sig.Setup}, \mathtt{Sig.KG}, \mathtt{Sig.Sign}, \mathtt{Sig.Vrf})$.
$\mathtt{Sig.Setup}(1^\lambda) \rightarrow \mathtt{PP}$. On input the security parameter 1^λ, it generates a set of public parameter values $\mathtt{PP}$. It returns $\mathtt{PP}$.
$\mathtt{Sig.KG}(\mathtt{PP}) \rightarrow (\mathrm{PK}, \mathrm{SK})$. On input the set of public parameter values $\mathtt{PP}$, this PPT algorithm generates a signing key SK and the corresponding public key PK. It returns $(\mathrm{PK}, \mathrm{SK})$.
$\mathtt{Sig.Sign}(\mathtt{PP}, \mathrm{PK}, \mathrm{SK}, m) \rightarrow \sigma$. On input the set of public parameter values $\mathtt{PP}$, 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.
$\mathtt{Sig.Vrf}(\mathtt{PP}, \mathrm{PK}, m, \sigma) \rightarrow d$. 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 $\mathtt{Sig}$: For any security parameter 1^λ and any message $m \in \mathcal{M}sg(1^\lambda)$, $\Pr[d = 1 \mid \mathtt{PP} \leftarrow \mathtt{Sig.Setup}(1^\lambda), (\mathrm{PK}, \mathrm{SK}) \leftarrow \mathtt{Sig.KG}(\mathtt{PP}), \sigma \leftarrow \mathtt{Sig.Sign}(\mathtt{PP}, \mathrm{PK}, \mathrm{SK}, m), d \leftarrow \mathtt{Sig.Vrf}(\mathtt{PP}, \mathrm{PK}, m, \sigma)] = 1$.

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

$$\begin{aligned}\mathbf{Exp}^{\text{euf-cma}}_{\mathtt{Sig},\mathbf{F}}(1^\lambda):\ & \mathtt{PP} \leftarrow \mathtt{Sig.Setup}(1^\lambda), (\mathrm{PK}, \mathrm{SK}) \leftarrow \mathtt{Sig.KG}(\mathtt{PP}) \\ & (m^*, \sigma^*) \leftarrow \mathbf{F}^{\mathbf{SignO}(\mathtt{PP}, \mathrm{PK}, \mathrm{SK}, \cdot)}(\mathtt{PP}, \mathrm{PK}) \\ & \text{If } m^* \notin \{m_j\}_{1 \leq j \leq q_s} \text{ and } \mathtt{Sig.Vrf}(\mathrm{PK}, m^*, \sigma^*) = 1, \\ & \quad \text{then Return Win else Return Lose}\end{aligned}$$

In the experiment, $\mathbf{F}$ issues a signing query to its signing oracle $\mathbf{SignO}(\mathtt{PP}, \mathrm{PK}, \mathrm{SK}, \cdot)$ by sending a message m_j at most q_s times ($1 \leq j \leq q_\mathrm{s}$). 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 $\{m_j\}_{1 \leq j \leq q_\mathrm{s}}$ should not contain the message m^*. The advantage of $\mathbf{F}$ over $\mathtt{Sig}$ is defined as $\mathbf{Adv}^{\text{euf-cma}}_{\mathtt{Sig},\mathbf{F}}(\lambda) := \Pr[\mathbf{Exp}^{\text{euf-cma}}_{\mathtt{Sig},\mathbf{F}}(1^\lambda) \text{ returns } \textsc{Win}]$. The digital signature scheme $\mathtt{Sig}$ is said to be *existentially unforgeable against adaptive chosen-message attacks* if for any given PPT algorithm $\mathbf{F}$, the advantage $\mathbf{Adv}^{\text{euf-cma}}_{\mathtt{Sig},\mathbf{F}}(\lambda)$ is negligible in λ.

3 Witness-Indistinguishable Arguments with Σ-Protocols 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 $\Phi^a_{\mathtt{PP}}$ for each index $a \in A$, which determines an NP witness relation R^a:

$$R^a = \{(x^a, w^a) \in \{0,1\}^* \times \{0,1\}^* \mid \Phi^a_{\mathtt{PP}}(x^a, w^a) = \textsc{true}\}, a \in A. \tag{1}$$

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

$$\Sigma^a = (\Sigma^a_{\mathrm{com}}, \Sigma^a_{\mathrm{cha}}, \Sigma^a_{\mathrm{res}}, \Sigma^a_{\mathrm{vrf}}, \Sigma^a_{\mathrm{ext}}, \Sigma^a_{\mathrm{sim}}). \tag{2}$$

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 0th components* $W^a_0, a \in A$, *are equal, which we denote by* W_0. We call the equal set W_0 the *base witness space* of the witness spaces $W^a, a \in A$, and

an element $w_0 \in W_0$ a *base witness point*. Then a witness $w^a \in W^a$ consists of w_0 and w_1^a. That is, $W^a = W_0 \times W_1^a \ni (w_0, w_1^a) = w^a$.

Commit-and-Prove Scheme with Σ-Protocol. We employ a commit-and-prove scheme with a Σ-protocol: $\mathtt{CmtPrv} = (\mathtt{CmtPrv.Setup}, \mathtt{Cmt} = (\mathtt{Cmt.Com}, \mathtt{Cmt.Vrf}), \Pi_0 = (\mathtt{P}_0, \mathtt{V}_0))$, where the predicate $\Phi_{0,\mathtt{PP}}$ and the relation R_0 is defined as follows, and Π_0 is executed in accordance with a Σ-protocol Σ_0:

$$\Phi_{0,\mathtt{PP}}(c_0, (w_0, r_0)) \stackrel{\text{def}}{=} (\mathtt{Cmt.Com}(\mathtt{PP}_0, w_0; r_0) =_? (c_0, r_0)),$$

$$R_0 \stackrel{\text{def}}{=} \{(c_0, (w_0, r_0)) \in \{0,1\}^* \times (\{0,1\}^*)^2 \mid \Phi_{0,\mathtt{PP}}(c_0, (w_0, r_0)) = \text{TRUE}\}, \quad (3)$$

$$\Sigma_0 = (\Sigma_{0,\text{com}}, \Sigma_{0,\text{cha}}, \Sigma_{0,\text{res}}, \Sigma_{0,\text{vrf}}, \Sigma_{0,\text{ext}}, \Sigma_{0,\text{sim}}). \quad (4)$$

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

3.2 On the Existence of a Σ-Protocol for Simultaneous Satisfiability

We introduce for each index $a \in A$ the following composed relation determined by the two predicates $\Phi_{\mathtt{PP}}^a$ and $\Phi_{0,\mathtt{PP}}$. That is, the relation R_0^a is for *simultaneous satisfiability* of $\Phi_{\mathtt{PP}}^a$ and $\Phi_{0,\mathtt{PP}}$ on the base witness point w_0: For each $a \in A$,

$$R_0^a := \left\{ (x_0^a = (x^a, c_0), w_0^a = (w_0, w_1^a, r_0)) \middle| \begin{cases} \Phi_{\mathtt{PP}}^a(x^a, (w_0, w_1^a)) = \text{TRUE} \\ \Phi_{0,\mathtt{PP}}(c_0, (w_0, r_0)) = \text{TRUE} \end{cases} \right\}. \quad (5)$$

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

$$\Sigma_0^a = (\Sigma_{0,\text{com}}^a, \Sigma_{0,\text{cha}}^a, \Sigma_{0,\text{res}}^a, \Sigma_{0,\text{vrf}}^a, \Sigma_{0,\text{ext}}^a, \Sigma_{0,\text{sim}}^a). \quad (6)$$

- $\Sigma_{0,\text{com}}^a(x_0^a, w_0^a) \to (\text{COM}^a, \text{COM}_{a,0}, St_0^a)$. This PPT algorithm is executed by $\mathtt{P}_0^a$. 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 runs the algorithms $\Sigma_{\text{com}}^a(x^a, (w_0, w_1^a))$ and $\Sigma_{0,\text{com}}(c_0, (w_0, r_0))$ to obtain the commitment messages and the inner states, (COM^a, St^a) and $(\text{COM}_{a,0}, St_{a,0})$, respectively, with a constraint that the knowledge extractor $\Sigma_{0,\text{ext}}^a$ 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 $St_0^a := (St^a, St_{a,0})$. It returns $(\text{COM}^a, \text{COM}_{a,0}, St_0^a)$. $\mathtt{P}_0^a$ sends $(\text{COM}^a, \text{COM}_{a,0})$ to $\mathtt{V}_0^a$ as a commitment message, and keeps the state St_0^a.
- $\Sigma_{0,\text{cha}}^a(x_0^a) \to \text{CHA}$. This PPT algorithm is executed by $\mathtt{V}_0^a$. On input the statement x_0^a, it reads out the size of the security parameter as 1^λ and chooses a challenge message $\text{CHA} \in_R \text{CHASP}(1^\lambda)$. It returns CHA. $\mathtt{V}_0^a$ sends CHA to $\mathtt{P}_0^a$ as a challenge message.
- $\Sigma_{0,\text{res}}^a(St_0^a, \text{CHA}) \to (\text{RES}^a, \text{RES}_{a,0})$. This PPT algorithm is executed by $\mathtt{P}_0^a$. On input the state St_0^a and the challenge message CHA, it runs the algorithms $\Sigma_{\text{res}}^a(St^a, \text{CHA})$ and $\Sigma_{0,\text{res}}(St_{a,0}, \text{CHA})$ to obtain the response messages RES^a and $\text{RES}_{a,0}$, respectively, with the constraint that the knowledge extractor

$\Sigma^a_{0,\mathrm{ext}}$ should return a witness which simultaneously satisfies Φ^a and Φ_0 on w_0. It returns $(\mathrm{RES}^a, \mathrm{RES}_{a,0})$. $\mathtt{P}^a_0$ sends $(\mathrm{RES}^a, \mathrm{RES}_{a,0})$ to $\mathtt{V}^a_0$ as a response message.

- $\Sigma^a_{0,\mathrm{vrf}}(x^a_0, (\mathrm{COM}^a, \mathrm{COM}_{a,0}), \mathrm{CHA}, (\mathrm{RES}^a, \mathrm{RES}_{a,0})) \to d$. This deterministic algorithm is executed by $\mathtt{V}^a_0$. On input the statement $x^a_0 = (x^a, c_0)$ and all the messages $(\mathrm{COM}^a, \mathrm{COM}_{a,0})$, CHA and $(\mathrm{RES}^a, \mathrm{RES}_{a,0})$, it runs the algorithms $\Sigma^a_{\mathrm{vrf}}(x^a, \mathrm{COM}^a, \mathrm{CHA}, \mathrm{RES}^a)$ and $\Sigma_{0,\mathrm{vrf}}(c_0, \mathrm{COM}_{a,0}, \mathrm{CHA}, \mathrm{RES}_{a,0})$ to obtain two boolean decisions d^a and $d_{a,0}$. If the both d^a and $d_{a,0}$ are 1, then it returns $d := 1$, and otherwise $d := 0$. $\mathtt{V}^a_0$ returns d as the decision of the interactive protocol on x^a_0.
- $\Sigma^a_{0,\mathrm{ext}}(x^a_0, (\mathrm{COM}^a, \mathrm{COM}_{a,0}), \mathrm{CHA}, (\mathrm{RES}^a, \mathrm{RES}_{a,0}), \mathrm{CHA}', (\mathrm{RES}^{a\prime}, \mathrm{RES}_{a,0}')) \to (\hat{w}^a_0, \hat{w}^a_1, \hat{r}_{a,0})$. This PPT algorithm is for knowledge extraction. On input the statement $x^a_0 = (x^a, c_0)$ and two accepting transcripts with a common commitment message and different challenge messages, $((\mathrm{COM}^a, \mathrm{COM}_{a,0}), \mathrm{CHA}, (\mathrm{RES}^a, \mathrm{RES}_{a,0}))$ and $((\mathrm{COM}^a, \mathrm{COM}_{a,0}), \mathrm{CHA}', (\mathrm{RES}^{a\prime}, \mathrm{RES}_{a,0}'))$, $\mathrm{CHA} \neq \mathrm{CHA}'$, it runs the algorithms $\Sigma^a_{\mathrm{ext}}(x^a, \mathrm{COM}^a, \mathrm{CHA}, \mathrm{RES}^a, \mathrm{CHA}', \mathrm{RES}^{a\prime})$ and $\Sigma_{0,\mathrm{ext}}(c_0, \mathrm{COM}_{a,0}, \mathrm{CHA}, \mathrm{RES}_{a,0}, \mathrm{CHA}', \mathrm{RES}_{a,0}')$ to obtain witnesses $(\hat{w}^a_0, \hat{w}^a_1)$ and $(\hat{w}_{a,0}, \hat{r}_{a,0})$ satisfying $(x^a, (\hat{w}^a_0, \hat{w}^a_1)) \in R^a$ and $(c_0, (\hat{w}_{a,0}, \hat{r}_{a,0})) \in R_0$ with an overwhelming probability in $|x^a|$ and $|c_0|$, respectively. Here the simultaneous satisfiability on w_0 should assure the following equality:

$$\hat{w}^a_0 = \hat{w}_{a,0} \text{ with probability one.} \tag{7}$$

  It returns $(\hat{w}^a_0, \hat{w}^a_1, \hat{r}^a_0)$.
- $\Sigma^a_{0,\mathrm{sim}}(x^a_0, \tilde{\mathrm{CHA}}) \to ((\tilde{\mathrm{COM}}^a, \tilde{\mathrm{COM}}_{a,0}), (\tilde{\mathrm{RES}}^a, \tilde{\mathrm{RES}}_{a,0}))$. This PPT algorithm is for the simulation of an accepting transcript. On input a statement $x^a_0 = (x^a, c_0)$ and a uniform random string $\tilde{\mathrm{CHA}} \in_R \mathrm{CHASP}(1^\lambda)$, it runs the algorithms $\Sigma^a_{\mathrm{sim}}(x^a, \tilde{\mathrm{CHA}})$ and $\Sigma_{0,\mathrm{sim}}(c_0, \tilde{\mathrm{CHA}})$ to obtain the remaining part of the transcripts $(\tilde{\mathrm{COM}}^a, \tilde{\mathrm{RES}}^a)$ and $(\tilde{\mathrm{COM}}_{a,0}, \tilde{\mathrm{RES}}_{a,0})$, respectively. The simulated messages $((\tilde{\mathrm{COM}}^a, \tilde{\mathrm{COM}}_{a,0}), \tilde{\mathrm{CHA}}, (\tilde{\mathrm{RES}}^a, \tilde{\mathrm{RES}}_{a,0}))$ should form $\mathit{dist}\big((\tilde{\mathrm{COM}}^a, \tilde{\mathrm{COM}}_{a,0}), \tilde{\mathrm{CHA}}, (\tilde{\mathrm{RES}}^a, \tilde{\mathrm{RES}}_{a,0}) \mid \text{gen. by } \mathrm{CHASP}(1^\lambda), \Sigma^a_{0,\mathrm{sim}}(x^a_0, \tilde{\mathrm{CHA}})\big)$ which is identical to $\mathit{dist}((\mathrm{COM}^a, \mathrm{COM}_{a,0}), \mathrm{CHA}, (\mathrm{RES}^a, \mathrm{RES}_{a,0}) \mid \text{real accepting})$.

Remark. To construct the algorithm $\Sigma^a_{0,\mathrm{com}}$ of commitment message and the algorithm $\Sigma^a_{0,\mathrm{res}}$ of response message is a non-trivial task. That is, we have to construct $\Sigma^a_{0,\mathrm{com}}$ and $\Sigma^a_{0,\mathrm{res}}$ so that the knowledge extractor $\Sigma^a_{0,\mathrm{ext}}$ 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 $\mathrm{COM}_{a,0}$, but we do not describe the inner treatment of the random tapes in $\Sigma^a_{0,\mathrm{com}}$ and $\Sigma^a_{0,\mathrm{res}}$ for generality. Hence our approach is to show the construction when we instantiate the Σ-protocol Σ^a_0.

3.3 Bundled Witness Space

We now introduce an NP witness relation for our *bundled witness space*. 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 $R^a, a \in A$:

$$R^a_{w_0} := \{(x^a, w^a) \in R^a \mid w^a = (w_0, w^a_1) \text{ for some } w^a_1\} \subset R^a,\ a \in A. \tag{8}$$

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:*

$$\bigcup_{w_0 \in W_0} \Big(\prod_{a \in A} R^a_{w_0} \Big) \subset \prod_{a \in A} \Big(\bigcup_{w_0 \in W_0} R^a_{w_0} \Big) = \prod_{a \in A} R^a. \tag{9}$$

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 $a \in A$. 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:

$$R^{a \in A}_{\mathrm{bnd}} \stackrel{\mathrm{def}}{=} \{((x^a)^{a \in A}, w_0, (w^a_1)^{a \in A}) | (x^a, (w_0, w^a_1)) \in R^a, a \in A\} \simeq \bigcup_{w_0 \in W_0} \Big(\prod_{a \in A} R^a_{w_0} \Big).$$

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. Omitted. (will appear in the full version).

Definition 1 (Relation for Bundled Witness Space). *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 Space). *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 space is defined as follows.*

$$W^{a \in A}_{bnd} \stackrel{def}{=} W_0 \times (W^a_1)^{a \in A}. \tag{10}$$

3.4 Generic Construction of Σ-Protocol for Bundled Witness Space

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

$$\Sigma^{a \in A}_{\mathrm{bnd}} = (\Sigma^{a \in A}_{\mathrm{bnd,com}}, \Sigma^{a \in A}_{\mathrm{bnd,cha}}, \Sigma^{a \in A}_{\mathrm{bnd,res}}, \Sigma^{a \in A}_{\mathrm{bnd,vrf}}, \Sigma^{a \in A}_{\mathrm{bnd,ext}}, \Sigma^{a \in A}_{\mathrm{bnd,sim}}). \tag{11}$$

- $\Sigma^{a\in A}_{\mathrm{bnd,com}}((x^a)^{a\in A},(w_0,(w_1^a)^{a\in A})) \to (c_0,(\mathrm{COM}^a,\mathrm{COM}_{a,0})^{a\in A},St)$. 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 $(w_0,(w_1^a)^{a\in A})$, 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: $(c_0,r_0) \leftarrow \mathtt{Cmt.Com}(w_0;r_0)$. It sets the extended statement as $x_0^a := (x^a,c_0)$ and the extended witness as $w_0^a := (w_0,w_1^a,r_0)$ for each $a \in A$. it runs the algorithms $\Sigma^a_{0,\mathrm{com}}(x_0^a,w_0^a)$ to obtain $(\mathrm{COM}^a,\mathrm{COM}_{a,0},St_0^a)$ for each $a \in A$. It sets the state as $St := (St_0^a)^{a\in A}$. 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. $\mathtt{V}_0^a$ sends CHA to $\mathtt{P}_0^a$ 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 runs the algorithms $\Sigma^a_{0,\mathrm{res}}(St_0^a,\mathrm{CHA})$ to obtain $(\mathrm{RES}^a,\mathrm{RES}_{a,0})$ for each $a \in A$. 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 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 $x_0^a := (x^a,c_0)$ for each $a \in A$. Then it runs the algorithms $\Sigma^a_{0,\mathrm{vrf}}(x_0^a,\mathrm{COM}^a,\mathrm{COM}_{a,0},\mathrm{CHA},\mathrm{RES}^a,\mathrm{RES}_{a,0})$ to obtain boolean decisions, for each $a \in A$. 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_0^a is complete for $a \in A$, 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_0^a for each $a \in A$. □

- $\Sigma^{a\in A}_{\mathrm{bnd,ext}}((x^a)^{a\in A},(c_0,(\mathrm{COM}^a,\mathrm{COM}_{a,0})^{a\in A}),\mathrm{CHA},(\mathrm{RES}^a,\mathrm{RES}_{a,0})^{a\in A},\mathrm{CHA}',((\mathrm{RES}^a)',(\mathrm{RES}_{a,0})')^{a\in A}) \to (\hat{w}_0,(\hat{w}_1^a)^{a\in A})$. This PPT algorithm is for knowledge extraction. On input the statement $(x^a)^{a\in A}$ and two accepting transcripts with a common commitment message and different challenge messages, $((c_0,(\mathrm{COM}^a,\mathrm{COM}_{a,0})^{a\in A}),\mathrm{CHA},(\mathrm{RES}^a,\mathrm{RES}_{a,0})^{a\in A}))$ and $((c_0,(\mathrm{COM}^a,\mathrm{COM}_{a,0})^{a\in A}),\mathrm{CHA}',(\mathrm{RES}^{a\prime},\mathrm{RES}_{a,0}{}')^{a\in A}))$, $\mathrm{CHA} \neq \mathrm{CHA}'$, it first sets the extended statement as $x_0^a := (x^a,c_0)$ for each $a \in A$. Then it runs the algorithms $\Sigma^a_{0,\mathrm{ext}}(x_0^a,(\mathrm{COM}^a,\mathrm{COM}_{a,0}),\mathrm{CHA},(\mathrm{RES}^a,\mathrm{RES}_{a,0}),\mathrm{CHA}',(\mathrm{RES}^{a\prime},\mathrm{RES}_{a,0}{}'))$ to obtain $(\hat{w}_0^a,\hat{w}_1^a,\hat{r}_0^a)$ for each $a \in A$. If this event does not occur (i.e. at least at one a $\Sigma^a_{0,\mathrm{ext}}$ fails to extract a witness), then it returns $\perp$. Otherwise, if $\hat{w}_0^a = \hat{w}_0^{a'}$ for any $a,a' \in A$, then it sets the common value $\hat{w}_0 := \hat{w}_0^a$ and returns $(\hat{w}_0,(\hat{w}_1^a)^{a\in A})$. 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_0^a has the special soundness for $a \in A$, then our $\Sigma_{bnd}^{a\in A}$ has the special soundness.*

Proof. Omitted. (will appear in the full version).

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

- $\Sigma_{\mathrm{bnd,sim}}^{a\in A}((x^a)^{a\in A}, \tilde{\mathrm{CHA}}) \rightarrow ((\tilde{c}_0, (\tilde{\mathrm{COM}}^a, \tilde{\mathrm{COM}}_0^a)^{a\in A}), (\tilde{\mathrm{RES}}^a, \tilde{\mathrm{RES}}_0^a)^{a\in A})$. 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 runs the commitment generation algorithm with a randomness $\tilde{r}_0$, $\mathtt{Cmt.Com}(\tilde{w}_0; \tilde{r}_0) \rightarrow (\tilde{c}_0, \tilde{r}_0)$, to obtain a commitment $\tilde{c}_0$. Then it sets the extended statement as $x_0^a := (x^a, \tilde{c}_0)$ for each $a \in A$. Then, it runs the algorithms $\Sigma_{0,\mathrm{sim}}^a(x_0^a, \tilde{\mathrm{CHA}})$ to obtain $((\tilde{\mathrm{COM}}^a, \tilde{\mathrm{COM}}_{a,0}), (\tilde{\mathrm{RES}}^a, \tilde{\mathrm{RES}}_{a,0}))$ for each $a \in A$. 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_0^a is honest-verifier zero-knowledge for $a \in A$, then our $\Sigma_{bnd}^{a\in A}$ is honest-verifier zero-knowledge.*

Proof. Omitted. (will appear in the full version).

Theorem 1. *If Cmt is correct, computationally binding and perfectly hiding, and if Σ_0^a is a Σ-protocol for $a \in A$, then our protocol $\Sigma_{bnd}^{a\in A}$ is a Σ-protocol.*

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

Theorem 2. *If the component interactive proof system Π_0^a with Σ_0^a is perfectly witness-indistinguishable for each $a \in A$, and if Cmt is perfectly hiding, then our interactive argument system $\Pi_{bnd}^{a\in A}$ with $\Sigma_{bnd}^{a\in A}$ is perfectly witness-indistinguishable.*

Proof. Omitted. (will appear in the full version).

4 Decentralized Multi-authority Anonymous Authentication Scheme

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

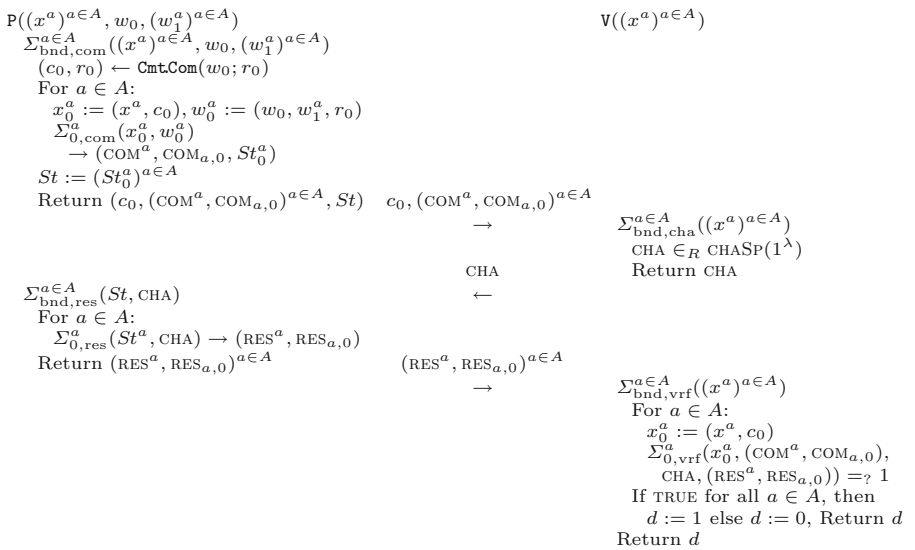

Fig. 1. The protocol $\Sigma^{a\in A}_{\mathrm{bnd}}$ of our proof system $\Pi^{a\in A}_{\mathrm{bnd}}$ for the NP witness relation $R^{a\in A}_{\mathrm{bnd}}$.

4.1 Syntax and Security Definitions

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

- $\mathtt{Setup}(1^\lambda) \rightarrow \mathtt{PP}$. This PPT algorithm is needed to generate a set of public parameter values PP. On input the security parameter 1^λ, it generates the set of values PP. It returns PP.
- $\mathtt{AuthKG}(\mathtt{PP}, a) \rightarrow (\mathrm{PK}^a, \mathrm{MSK}^a)$. This PPT algorithm is executed by a key-issuing authority indexed by a positive integer a. On input the set of public parameter values PP 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 $(\mathrm{PK}^a, \mathrm{MSK}^a)$.
- $\mathtt{PrivKG}(\mathtt{PP}, \mathrm{PK}^a, \mathrm{MSK}^a, \mathtt{gid}) \rightarrow \mathrm{sk}^a_{\mathtt{gid}}$. This PPT algorithm is executed by the a-th key-issuing authority. On input the set of public parameter values PP, the a-th public and master secret keys $(\mathrm{PK}^a, \mathrm{MSK}^a)$ and a string gid of a prover (a global identity string), it generates a private secret key $\mathrm{sk}^a_{\mathtt{gid}}$ of a prover. It returns $\mathrm{sk}^a_{\mathtt{gid}}$.
- $\langle \mathtt{P}(\mathtt{PP}, (\mathrm{PK}^a, \mathrm{sk}^a_{\mathtt{gid}})^{a\in A'}), \mathtt{V}(\mathtt{PP}, (\mathrm{PK}^a)^{a\in A'}) \rangle \rightarrow d$. 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 set of public parameter values PP and the public keys $(\mathrm{PK}^a)^{a\in A}$ to P and V and the corresponding private secret keys $(\mathrm{sk}^a_{\mathtt{gid}})^{a\in A}$ 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 Attack of Misauthentication. One of the possible attacks to cause misauthentication is the concurrent and collusion attack on our a-auth. For a formal treatment we define the following experiment on a-auth and an adversary algorithm $\mathbf{A}$.

$$\mathbf{Expr}^{\text{conc-coll}}_{\text{a-auth},\mathbf{A}}(1^\lambda): \quad q_A \leftarrow \mathbf{A}(1^\lambda), A := \{1,\ldots,q_A\}, \mathtt{PP} \leftarrow \mathtt{Setup}(1^\lambda)$$
$$\text{For } a \in A : (\mathrm{PK}^a, \mathrm{MSK}^a) \leftarrow \mathtt{AuthKG}(\mathtt{PP}, a)$$
$$q_I \leftarrow \mathbf{A}(\mathtt{PP}, (\mathrm{PK}^a)^{a\in A}), I := \{1,\ldots,q_I\}, \text{For } i \in I : \mathtt{gid}_i \in_R \{0,1\}^\lambda$$
$$\text{For } a \in A : \text{For } i \in I : \mathrm{sk}^a_{\mathtt{gid}_i} \leftarrow \mathtt{PrivKG}(\mathtt{PP}, \mathrm{PK}^a, \mathrm{MSK}^a, \mathtt{gid}_i)$$
$$(A^*, St^*) \leftarrow \mathbf{A}^{\mathrm{P}(\mathtt{PP},(\mathrm{PK}^a,\mathrm{sk}^a_{\mathtt{gid}_i})^{a\in A})|_{i\in I},\mathbf{PrivKO}(\mathtt{PP},\mathrm{PK}^{\cdot},\mathrm{MSK}^{\cdot},\cdot)}(\mathtt{PP}, (\mathrm{PK}^a)^{a\in A})$$
$$\langle \mathbf{A}(St^*), \mathtt{V}(\mathtt{PP}, (\mathrm{PK}^a)^{a\in A^*})\rangle \rightarrow d, \text{If } d = 1 \text{ then Return } \textsc{Win} \text{ else Return } \textsc{Lose}$$

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 set of public parameter values PP and the issued public keys $(\mathrm{PK}^a)^{a\in A}$, $\mathbf{A}$ outputs the number q_I of provers with which $\mathbf{A}$ interacts concurrently (i.e. in arbitrarily interleaved order of messages). In addition, $\mathbf{A}$ collects at most q_{sk} private secret keys by issuing queries to the private secret key oracle $\mathbf{PrivKO}(\mathtt{PP}, \mathrm{PK}^{\cdot}, \mathrm{MSK}^{\cdot}, \cdot)$ with an authority index $a \in A$ and a global identity string $\mathtt{gid}_j \in \{0,1\}^\lambda$ for $j = q_I + 1, \ldots, q_I + q_{\mathrm{sk}}$. We denote by A_j the set of authority indices for which the queries with the global identity string $\mathtt{gid}_j$ were issued. That is, $A_j := \{a \in A \mid \mathbf{A} \text{ receives } \mathrm{sk}^a_{\mathtt{gid}_j}\}, j = q_I + 1, \ldots, q_I + q_{\mathrm{sk}}$. We here require that the numbers q_A, q_I and q_{sk} are bounded by a polynomial in λ. At the last of this "learning phase", $\mathbf{A}$ outputs a target set of authority indices A^* and its inner state St^*. Next, in the "attacking phase", on input the inner state St^*, the adversary $\mathbf{A}$ interacts with the verifier $\mathtt{V}(\mathtt{PP}, (\mathrm{PK}^a)^{a\in A^*})$. If the decision d of V is 1, then the experiment returns Win and otherwise, returns Lose. A restriction is imposed on the adversary $\mathbf{A}$: The target set of authority indices A^* should not be a subset of any single set A_j: $A^* \not\subseteq A_j$, $j = q_I + 1, \ldots, q_I + q_{\mathrm{sk}}$. This restriction is because, otherwise, $\mathbf{A}$ is given private secret keys for A^* on a single $\mathtt{gid}_{i^*}$ for some i^*, $q_I < i^* \leq q_I + q_{\mathrm{sk}}$, and then $\mathbf{A}$ can trivially be accepted in the attacking phase.

The advantage of an adversary $\mathbf{A}$ over our authentication scheme a-auth in the experiment is defined as: $\mathbf{Adv}^{\text{conc-coll}}_{\text{a-auth},\mathbf{A}}(\lambda) \stackrel{\text{def}}{=} \Pr[\mathbf{Expr}^{\text{conc-coll}}_{\text{a-auth},\mathbf{A}}(1^\lambda) = \textsc{Win}]$. An authentication scheme a-auth is called secure against concurrent and collusion attacks of misauthentication if, for any given PPT algorithm $\mathbf{A}$, the advantage $\mathbf{Adv}^{\text{conc-coll}}_{\text{a-auth},\mathbf{A}}(\lambda)$ is negligible in λ.

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

$$
\begin{aligned}
&\mathbf{Expr}^{\mathrm{ano}}_{\mathtt{a\text{-}auth},\mathbf{A}}(1^\lambda): \quad q_A \leftarrow \mathbf{A}(1^\lambda), A := \{1,\ldots,q_A\}, \mathtt{PP} \leftarrow \mathtt{Setup}(1^\lambda)\\
&\text{For } a \in A : (\mathrm{PK}^a, \mathrm{MSK}^a) \leftarrow \mathtt{AuthKG}(\mathtt{PP}, a)\\
&\mathtt{gid}_0, \mathtt{gid}_1 \leftarrow \mathbf{A}(\mathtt{PP}, (\mathrm{PK}^a)^{a\in A})\\
&\text{For } a \in A : \text{For } i \in 0,1 : \mathrm{sk}^a_{\mathtt{gid}_i} \leftarrow \mathtt{PrivKG}(\mathtt{PP}, \mathrm{PK}^a, \mathrm{MSK}^a, \mathtt{gid}_i)\\
&b \in_R \{0,1\}, b^* \leftarrow \mathbf{A}^{\mathrm{P}(\mathtt{PP},(\mathrm{PK}^a,\mathrm{sk}^a_{\mathtt{gid}_b})^{a\in A})}(\mathtt{PP}, (\mathrm{PK}^a, \mathrm{sk}^a_{\mathtt{gid}_0}, \mathrm{sk}^a_{\mathtt{gid}_1})^{a\in A})\\
&\text{If } b = b^*, \text{ then Return } \textsc{Win}, \text{ else Return } \textsc{Lose}
\end{aligned}
$$

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 $\mathtt{gid}_0$ and $\mathtt{gid}_1$ (as is usual in the indistinguishability games). Next, $\mathbf{A}$ interacts with a prover P on input even the private secret keys $(\mathrm{sk}^a_{\mathtt{gid}_b})^{a\in A}$, where the index b is chosen uniformly at random. If the decision b^* of $\mathbf{A}$ is equal to b, then the experiment returns Win and otherwise, returns Lose.

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

4.2 Generic Construction

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

- $\mathtt{Setup}(1^\lambda) \to \mathtt{PP}$. On input the security parameter 1^λ, this PPT algorithm generates a set of public parameter values by running the setup algorithms $\mathtt{Sig.Setup}(1^\lambda)$, $\Pi.\mathtt{Setup}(1^\lambda)$ and $\mathtt{CmtPrv.Setup}(1^\lambda)$. These algorithms are for the digital signature scheme Sig, the interactive argument systems $(\Pi^a_0)^{a\in A}$, and the commitment generation algorithm Cmt.Com. They generate $\mathtt{PP}_{\mathtt{Sig}}$, $\mathtt{PP}_{\Pi}$ and $\mathtt{PP}_{\mathtt{Cmt}}$, respectively. It merges them as $\mathtt{PP} := (\mathtt{PP}_{\mathtt{Sig}}, \mathtt{PP}_{\Pi}, \mathtt{PP}_{\mathtt{Cmt}})$. It returns PP.
- $\mathtt{AuthKG}(\mathtt{PP}, a) \to (\mathrm{PK}^a, \mathrm{MSK}^a)$. On input the set of public parameter values PP and an authority index a, this PPT algorithm executes the key generation algorithm $\mathtt{Sig.KG}(\mathtt{PP}_{\mathtt{Sig}})$ to obtain 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 $(\mathrm{PK}^a, \mathrm{MSK}^a)$.
- $\mathtt{PrivKG}(\mathtt{PP}, \mathrm{PK}^a, \mathrm{MSK}^a, \mathtt{gid}) \to \mathrm{sk}^a_{\mathtt{gid}}$. On input the set of public parameter values PP, a public key PK^a, the corresponding master secret key MSK^a and a string gid, this PPT algorithm executes the signing algorithm

$\mathtt{Setup}(1^\lambda)$ $\mathtt{PP}_{\mathtt{Sig}} \leftarrow \mathtt{Sig.Setup}(1^\lambda)$ $\mathtt{PP}_{\Pi} \leftarrow \Pi.\mathtt{Setup}(1^\lambda)$ $\mathtt{PP}_{\mathtt{CmtPrv}} \leftarrow \mathtt{CmtPrv.Setup}(1^\lambda)$ $\mathtt{PP} := (\mathtt{PP}_{\Pi}, \mathtt{PP}_{\mathtt{CmtPrv}}, \mathtt{PP}_{\mathtt{Sig}})$ Return PP	$\mathtt{AuthKG}(\mathtt{PP}, a)$ $(\mathrm{SK}, \mathrm{PK}) \leftarrow \mathtt{Sig.KG}(\mathtt{PP}_{\mathtt{Sig}})$ $\mathrm{PK}^a := \mathrm{PK}, \mathrm{MSK}^a := \mathrm{SK}$ Return $(\mathrm{PK}^a, \mathrm{MSK}^a)$	$\mathtt{PrivKG}(\mathtt{PP}, \mathrm{PK}^a, \mathrm{MSK}^a, \mathtt{gid})$ $\sigma^a_{\mathtt{gid}} \leftarrow \mathtt{Sig.Sign}(\mathtt{PP}_{\mathtt{Sig}}, \mathrm{PK}^a, \mathrm{MSK}^a, \mathtt{gid})$ $\mathrm{sk}^a_{\mathtt{gid}} := \sigma^a_{\mathtt{gid}}$ Return $\mathrm{sk}^a_{\mathtt{gid}}$
$\mathtt{P}(\mathtt{PP}, (\mathrm{PK}^a)^{a\in A}, (\mathrm{sk}^a_{\mathtt{gid}})^{a\in A})$ For $a \in A$: $x^a := \mathrm{PK}^a, w_1^a := \mathrm{sk}^a_{\mathtt{gid}}$ $w_0 := \mathtt{gid}$	(Execute $\Sigma^{a\in A}_{\mathrm{bnd}}$)	$\mathtt{V}(\mathtt{PP}, (\mathrm{PK}^a)^{a\in A})$ For $a \in A$: $x^a := \mathrm{PK}^a$ Return $(d \leftarrow \Sigma^{a\in A}_{\mathrm{bnd,vrf}})$

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

$\mathtt{Sig.Sign}(\mathtt{PP}_{\mathtt{Sig}}, \mathrm{PK}^a, \mathrm{MSK}^a, \mathtt{gid})$ to obtain a digital signature $\sigma^a_{\mathtt{gid}}$ on the message gid. It puts a private secret key $\mathrm{sk}^a_{\mathtt{gid}}$ as $\mathrm{sk}^a_{\mathtt{gid}} := \sigma^a_{\mathtt{gid}}$. It returns $\mathrm{sk}^a_{\mathtt{gid}}$.

- $\mathtt{P}(\mathtt{PP}, (\mathrm{PK}^a)^{a\in A}, (\mathrm{sk}^a_{\mathtt{gid}})^{a\in A})$ and $\mathtt{V}(\mathtt{PP}, (\mathrm{PK}^a)^{a\in A})$. On input the set of public parameter values PP and the public keys $(\mathrm{PK}^a)^{a\in A}$ to the prover P and the verifier V, and the corresponding private secret keys $(\mathrm{sk}^a_{\mathtt{gid}})^{a\in A}$ to P, PPT algorithms P and V first set the statements as $x^a := \mathrm{PK}^a$ for $a \in A$ and P sets the witness as $w_0 := \mathtt{gid}$ and $w_1^a := \mathrm{sk}^a_{\mathtt{gid}}$ for $a \in A$. The witness spaces $W^a, a \in A$ are described as follows: $W^a = W_0 \times W_1^a, W_0 = \{\mathtt{gid} \mid \text{string of length } \lambda\} = \{0,1\}^\lambda, W_1^a = \{\sigma^a_{\mathtt{gid}} \mid \sigma^a_{\mathtt{gid}} \leftarrow \mathtt{Sig.Sign}(\mathtt{PP}_{\mathtt{Sig}}, \mathrm{PK}^a, \mathrm{MSK}^a, \mathtt{gid})$ for some $\mathtt{gid} \in W_0\}$. P and V execute the Σ protocol $\Sigma^{a\in A}_{\mathrm{bnd}}$. V returns the returned boolean d of the verifier algorithm $\Sigma^{a\in A}_{\mathrm{bnd,vrf}}$.

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 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* $\mathbf{A}$ *that executes a concurrent and collusion attack on our a-auth in accordance with the experiment* $\mathbf{Expr}^{conc\text{-}coll}_{a\text{-}auth,\mathbf{A}}(1^\lambda)$*, there exists a* PPT *algorithm* $\mathbf{F}$ *that generates an existential forgery on Sig in accordance with the experiment* $\mathbf{Exp}^{euf\text{-}cma}_{Sig,\mathbf{F}}(1^\lambda)$ *and there exists a* PPT *algorithm* $\mathbf{B}$ *that breaks the bandaging property of Cmt in accordance with the experiment* $\mathbf{Exp}^{bind}_{Cmt,\mathbf{B}}(1^\lambda)$ *satisfying the following inequality.*

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

Proof. Omitted. (will appear in the full version).

Theorem 4. *If the component proof system Π_0^a 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* PPT *algorithm* $\mathbf{A}$ *that executes the anonymity game on our* `a-auth` *in accordance with the experiment* $\mathbf{Expr}^{ano}_{\texttt{a-auth},\mathbf{A}}(1^\lambda)$*, the following equality holds.*

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

Proof. Omitted. (will appear in the full version).

5 On Instantiation and Implementation

In this section, we briefly discuss instantiation and implementation of our generic authentication scheme `a-auth` in Sect. 4.

Basically, we can employ any three building blocks that satisfy the requirements stated in Sect. 4. We here briefly mention an instantiation in the setting of bilinear groups. The three building blocks are the pairing version of the Camenisch-Lysyanskaya digital signature scheme $\texttt{Sig}^{\texttt{CL}}$ by Sudarsono-Nakanishi-Funabiki [14] and Teranishi-Furukawa [15], the pairing version of the Camenisch-Lysyanskaya perfectly witness-indistinguishable argument of knowledge system $\Pi^{\texttt{CL}}$ by [14,15], and the Pedersen-Okamoto commit-and-prove scheme $\texttt{CmtPrv}^{\texttt{PO}}$ [12,13].

As for implementation, we expect a similar result to the result found in [14] because the execution of the Pedersen-Okamoto commit-and-prove is fast. When the number of authorities involved in our authentication is 3, the expected times for proof-generation and verification are both under 0.5 seconds except the communication time. (See Sect. 5.2 of [14] "the total number of string attribute types".)

6 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 mentioned an instantiation of the scheme in the setting of bilinear groups. A post-quantum instantiation should be our future work.

References

1. Babai, L.: Trading group theory for randomness. In: Proceedings of the 17th Annual ACM Symposium on Theory of Computing, 6–8 May 1985, Providence, Rhode Island, USA, pp. 421–429 (1985). https://doi.org/10.1145/22145.22192
2. Bellare, M., Palacio, A.: GQ and schnorr identification schemes: proofs of security against impersonation under active and concurrent attacks. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 162–177. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45708-9_11
3. Canetti, R., Lindell, Y., Ostrovsky, R., Sahai, A.: Universally composable two-party and multi-party secure computation. In: Proceedings on 34th Annual ACM Symposium on Theory of Computing, 19–21 May 2002, Montréal, Québec, Canada, pp. 494–503 (2002). https://doi.org/10.1145/509907.509980
4. Cramer, R.: Modular designs of secure, yet practical cyptographic protocols. Ph.D. thesis, University of Amsterdam, Amsterdam, The Netherlands (1996)
5. Damgård, I.: On σ-protocols. Course Notes (2010). http://cs.au.dk/ivan/CPT.html
6. Escala, A., Groth, J.: Fine-tuning groth-sahai proofs. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 630–649. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54631-0_36
7. Feige, U., Shamir, A.: Witness indistinguishable and witness hiding protocols. In: Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, 13–17 May 1990, Baltimore, Maryland, USA, pp. 416–426 (1990). https://doi.org/10.1145/100216.100272
8. Fiat, A., Shamir, A.: How to prove yourself: practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987). https://doi.org/10.1007/3-540-47721-7_12
9. Goldreich, O.: The Foundations of Cryptography: Basic Techniques, vol. 1. Cambridge University Press, Cambridge (2001)
10. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems. In: Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing, STOC 1885, pp. 291–304. ACM, New York (1985). https://doi.org/10.1145/22145.22178
11. Okamoto, T., Takashima, K.: Decentralized attribute-based signatures. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 125–142. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36362-7_9
12. Okamoto, T.: Provably secure and practical identification schemes and corresponding signature schemes. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 31–53. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-48071-4_3
13. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 129–140. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-46766-1_9
14. Sudarsono, A., Nakanishi, T., Funabiki, N.: A pairing-based anonymous credential system with efficient attribute proofs. JIP **20**(3), 774–784 (2012). https://doi.org/10.2197/ipsjjip.20.774
15. Teranishi, I., Furukawa, J.: Anonymous credential with attributes certification after registration. IEICE Trans. **95-A**(1), 125–137 (2012). http://search.ieice.org/bin/summary.php?id=e95-a_1_125