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

Aug 13, 2018

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 proof, sigma protocol, witness indistinguishability, decentralized, collusion resistance

Table of Contents

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 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 with different identities, and tries to make a verifier accept anonymously by 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) [OT13] is an ABS scheme with decentralized key-issuing authorities. In an ABS scheme, a signer has credentials on her attributes. The signer is able to sign a message with a signing policy expressed as a boolean formula on attributes. There are assignment patterns to satisfy the boolean formula, and the attribute privacy of an ABS scheme should assure that the signatures signed by a user do not leak any information on the satisfying pattern which she used. We note that this property also requires the anonymity of the signer's identity. On the other hand, allowing decentralized multi-authorities is to have independent key-issuers each of which generates each private 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 ([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. 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 an DMA-ABS scheme by the Fiat-Shamir transform [FS86]. Actually, if a prover chooses a monotone boolean formula instead of an all-AND formula, and we apply the Fiat-Shamir transform to the Σ-protocol in our authentication scheme, then we obtain a DMA-ABS scheme.

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 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 $\mathit{transc}\langle A(a), B(b)\rangle$. 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 $A^{\mathbf{O}_i|_{i=1}^n}$. The probability of an event E is denoted by $\Pr[E]$. The conditional probability of an event E given events $F_1, \dots, F_n$ in this order is denoted by $\Pr[E|F_1, \dots, F_n]$. The distribution of a random variable X is denoted by $\mathit{dist}\big(X\big)$. The distribution of a random variable X whose probability is given by a joint probability of random variables $X, Y_1, \dots, Y_n$ is denoted by $\mathit{dist}\big(X|X, Y_1, \dots, Y_n\big)$. 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 $R \stackrel{\mathrm{def}}{=} \{(x, w) \in (\{0,1\}^*)^2 | \Phi(x, w) = \text{TRUE}\}$. We say that R is polynomially bounded if there exists a polynomial $\ell(\cdot)$ such that $|w| \le \ell(|x|)$ 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 $\mathtt{PP}$ 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: $L \stackrel{\mathrm{def}}{=} \{x \in \{0,1\}^*; \exists w \in \{0,1\}^*, (x, w) \in R\}$. 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 $\Pi = (\Pi.\mathtt{Setup}, \mathtt{P}, \mathtt{V})$, where $\Pi.\mathtt{Setup}$ is a set up algorithm for a set of public parameter values $\mathtt{PP}$, and $\mathtt{P}$ and $\mathtt{V}$ are a pair of interactive algorithms. $\mathtt{P}$, which is called a prover, is probabilistic and unbounded, and $\mathtt{V}$, which is called a verifier, is probabilistic polynomial-time (PPT). If $\mathtt{P}$ 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 $\Pi = (\Pi.\mathtt{Setup}, \mathtt{P}, \mathtt{V})$ [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 $\mathtt{P}$. 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 $\Sigma_{\mathrm{com}}(x, w) \rightarrow (\text{COM}, St)$. The second algorithm Σ_{cha} is executed by $\mathtt{V}$. 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 $\Sigma_{\mathrm{cha}}(x) \rightarrow \text{CHA}$. The third algorithm Σ_{res} is executed by $\mathtt{P}$. On input the state St and the challenge message CHA, it generates a response message RES.

It returns the message as $\varSigma_{\text{res}}(St, \textsc{cha}) \rightarrow \textsc{res}$. The fourth algorithm $\varSigma_{\text{vrf}}$ is executed by $\mathtt{V}$. On input the statement x and the messages $\textsc{com}$, $\textsc{cha}$ and $\textsc{res}$, it computes a boolean decision d. It returns the decision as $\varSigma_{\text{vrf}}(x, \textsc{com}, \textsc{cha}, \textsc{res}) \rightarrow d$. If $d = 1$, then we say that $\mathtt{P}$ is accepted by $\mathtt{V}$ on x. Otherwise, we say that $\mathtt{P}$ is rejected by $\mathtt{V}$ on x. The vector of all the messages $(\textsc{com}, \textsc{cha}, \textsc{res})$ is called a transcript of the interaction on x.

These four algorithms $(\varSigma_{\text{com}}, \varSigma_{\text{cha}}, \varSigma_{\text{res}}, \varSigma_{\text{vrf}})$ 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: $\Pr[\varSigma_{\text{vrf}}(x, \textsc{com}, \textsc{cha}, \textsc{res}) = 1 \mid \varSigma_{\text{com}}(x, w) \rightarrow (\textsc{com}, St), \varSigma_{\text{cha}}(x) \rightarrow \textsc{cha}, \varSigma_{\text{res}}(St, \textsc{cha}) \rightarrow \textsc{res}]$.

The fifth algorithm $\varSigma_{\text{ext}}$ concerns with the following property.
Special Soundness There is a PPT algorithm $\varSigma_{\text{ext}}$ called a *knowledge extractor*, which, on input a statement x and two accepting transcripts with a common commitment message and different challenge messages, $(\textsc{com}, \textsc{cha}, \textsc{res})$ and $(\textsc{com}, \textsc{cha}', \textsc{res}')$, $\textsc{cha} \neq \textsc{cha}'$, computes a witness $\hat{w}$ satisfying $(x, \hat{w}) \in R$ with an overwhelming probability in $|x|$:

$$\hat{w} \leftarrow \varSigma_{\text{ext}}(x, \textsc{com}, \textsc{cha}, \textsc{res}, \textsc{cha}', \textsc{res}'). \tag{1}$$

The sixth algorithm $\varSigma_{\text{sim}}$ concerns with the following property.
Honest-Verifier Zero-Knowledge There is a PPT algorithm called a *simulator* $\varSigma_{\text{sim}}$, which, on input a statement x, computes an accepting transcript on x:

$$(\widetilde{\textsc{com}}, \widetilde{\textsc{cha}}, \widetilde{\textsc{res}}) \leftarrow \varSigma_{\text{sim}}(x), \tag{2}$$

where the distribution of the simulated transcripts $\mathit{dist}\big(\widetilde{\textsc{com}}, \widetilde{\textsc{cha}}, \widetilde{\textsc{res}}\big)$ is identical to the distribution of the real accepting transcripts $\mathit{dist}\big(\textsc{com}, \textsc{cha}, \textsc{res}\big)$.
Note 1: Our Use Case In a $\varSigma$-protocol the challenge message $\textsc{cha}$ is a public coin. This property enables us in this paper to use the following variant of the simulator $\varSigma_{\text{sim}}(x)$: On input a simulated challenge message $\widetilde{\textsc{cha}}$ that is chosen uniformly at random, the variant generates a commitment $\widetilde{\textsc{com}}$ and a response $\widetilde{\textsc{res}}$:

$$\widetilde{\textsc{cha}} \in_R \textsc{cha}\textsc{Sp}(1^\lambda), \quad (\widetilde{\textsc{com}}, \widetilde{\textsc{res}}) \leftarrow \varSigma_{\text{sim}}(x, \widetilde{\textsc{cha}}). \tag{3}$$

Witness-Indistinguishability [FS90,Gol01] Let R be an NP relation. Suppose that an interactive argument system $\varPi = (\varPi.\mathtt{Setup}, \mathtt{P}, \mathtt{V})$ with a $\varSigma$-protocol $\varSigma$ 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 $\mathit{dist}\big(x, z, \mathit{transc}\langle \mathtt{P}(x, w_x), \mathtt{V}^*(x, z)\rangle\big)$ and $\mathit{dist}\big(x, z, \mathit{transc}\langle \mathtt{P}(x, w'_x), \mathtt{V}^*(x, z)\rangle\big)$ are identical.

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

A commit-and-prove scheme $\mathtt{CmtPrv}$ consists of five PPT algorithms: $\mathtt{CmtPrv} = (\mathtt{CmtPrv.Setup}, \mathtt{Cmt} = (\mathtt{Cmt.Com}, \mathtt{Cmt.Vrf}), \varPi = (\mathtt{P}, \mathtt{V}))$.
$\mathtt{CmtPrv.Setup}(1^\lambda) \rightarrow \mathtt{PP}$. On input the security parameter 1^λ, it generates a set of public parameter values $\mathtt{PP}$. It returns $\mathtt{PP}$.
$\mathtt{Cmt.Com}(\mathtt{PP}, m) \rightarrow (c, \kappa)$. 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).
$\mathtt{Cmt.Vrf}(\mathtt{PP}, c, m, \kappa) \rightarrow d$. 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 $\varPhi_{\mathtt{PP}}$ a predicate that returns the boolean decision: $\varPhi_{\mathtt{PP}}(c, (m, \kappa)) \overset{\text{def}}{=} (\mathtt{Cmt.Vrf}(\mathtt{PP}, c, m, \kappa))$. In the scheme there is an interactive argument system $\varPi = (\mathtt{P}, \mathtt{V})$ for the following relation R:

$$R := \{(c, (m, \kappa)) \in \{0,1\}^* \times (\{0,1\}^*)^2 \mid \varPhi_{\mathtt{PP}}(c, (m, \kappa)) = \textsc{true}\}. \tag{4}$$

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

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

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

$$\mathbf{Exp}^{\text{bind}}_{\texttt{Cmt},\mathbf{A}}(1^\lambda): \tag{5}$$

$$\texttt{PP} \leftarrow \texttt{CmtPrv.Setup}(1^\lambda), (c, m, \kappa, m', \kappa') \leftarrow \mathbf{A}(\texttt{PP}) \tag{6}$$

$$\text{If } \texttt{Cmt.Vrf}(\texttt{PP}, c, m, \kappa) = \texttt{Cmt.Vrf}(\texttt{PP}, c, m', \kappa') = 1 \wedge m \neq m', \text{then Return } \textsc{Win} \text{ else Return } \textsc{Lose} \tag{7}$$

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

Note 2: Our Use Case The commitment generation algorithm $\texttt{Cmt.Com}$ 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, $\texttt{Cmt.Com}(\texttt{PP}, m; r) \to (c, r)$.

2.3 Digital Signature Scheme [FS86]

A digital signature scheme $\texttt{Sig}$ consists of four PPT algorithms: $\texttt{Sig} = (\texttt{Sig.Setup}, \texttt{Sig.KG}, \texttt{Sig.Sign}, \texttt{Sig.Vrf})$.

$\texttt{Sig.Setup}(1^\lambda) \to \texttt{PP}$. On input the security parameter 1^λ, it generates a set of public parameter values $\texttt{PP}$. It returns $\texttt{PP}$.

$\texttt{Sig.KG}(\texttt{PP}) \to (\mathrm{PK}, \mathrm{SK})$. On input the set of public parameter values $\texttt{PP}$, this PPT algorithm generates a signing key SK and the corresponding public key PK. It returns $(\mathrm{PK}, \mathrm{SK})$.

$\texttt{Sig.Sign}(\texttt{PP}, \mathrm{PK}, \mathrm{SK}, m) \to \sigma$. On input the set of public parameter values $\texttt{PP}$, 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.

$\texttt{Sig.Vrf}(\texttt{PP}, \mathrm{PK}, m, \sigma) \to d$. 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 $\texttt{Sig}$: For any security parameter 1^λ and any message $m \in \mathcal{M}sg(1^\lambda)$, $\Pr[d = 1 \mid \texttt{PP} \leftarrow \texttt{Sig.Setup}(1^\lambda), (\mathrm{PK}, \mathrm{SK}) \leftarrow \texttt{Sig.KG}(\texttt{PP}), \sigma \leftarrow \texttt{Sig.Sign}(\texttt{PP}, \mathrm{PK}, \mathrm{SK}, m), d \leftarrow \texttt{Sig.Vrf}(\texttt{PP}, \mathrm{PK}, m, \sigma)] = 1$.

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

$$\mathbf{Exp}^{\text{euf-cma}}_{\texttt{Sig},\mathbf{F}}(1^\lambda): \tag{8}$$

$$\texttt{PP} \leftarrow \texttt{Sig.Setup}(1^\lambda), (\mathrm{PK}, \mathrm{SK}) \leftarrow \texttt{Sig.KG}(\texttt{PP}),\ (m^*, \sigma^*) \leftarrow \mathbf{F}^{\mathbf{SignO}(\texttt{PP}, \mathrm{PK}, \mathrm{SK}, \cdot)}(\texttt{PP}, \mathrm{PK}) \tag{9}$$

$$\text{If } m^* \notin \{m_j\}_{1 \le j \le q_{\text{s}}} \text{ and } \texttt{Sig.Vrf}(\mathrm{PK}, m^*, \sigma^*) = 1, \text{then Return } \textsc{Win} \text{ else Return } \textsc{Lose} \tag{10}$$

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

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

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:

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

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 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^a_1. That is;

$$\begin{array}{ccc} W^a & = W_0 \times & W^a_1, \\ \rotatebox{90}{$\in$} & & \rotatebox{90}{$\in$} \\ w^a & = \;(w_0, & w^a_1). \end{array} \tag{13}$$

Commit-and-Prove Scheme with $\boldsymbol{\Sigma}$-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:

$$\Phi_{0,\mathtt{PP}}(c_0, (w_0, r_0)) \overset{\text{def}}{=} (\mathtt{Cmt.Com(PP}_0, w_0; r_0) =_? (c_0, r_0)), \tag{14}$$

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

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

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

3.2 On the Existence of a $\boldsymbol{\Sigma}$-protocol for Simultaneous Satisfiability

We introduce for each index $a \in A$ the following composed relation determined by the two predicates $\Phi^a_{\mathtt{PP}}$ and $\Phi_{0,\mathtt{PP}}$. That is, the relation R^a_0 is for *simultaneous satisfiability* of the two predicates $\Phi^a_{\mathtt{PP}}$ and $\Phi_{0,\mathtt{PP}}$ on the base witness point w_0:

$$R^a_0 := \left\{(x^a_0 = (x^a, c_0), w^a_0 = (w_0, w^a_1, r_0)) \mid \begin{cases} \Phi^a_{\mathtt{PP}}(x^a, (w_0, w^a_1)) = \text{TRUE and} \\ \Phi_{0,\mathtt{PP}}(c_0, (w_0, r_0)) = \text{TRUE} \end{cases} \right\}, \; a \in A. \tag{17}$$

We require here that the Σ-protocols Σ^a and Σ_0 can be merged into a single Σ-protocol Σ^a_0 of an interactive argument system $\Pi^a_0 = (\Pi.\mathtt{Setup}, \mathtt{CmtPrv.Setup}, \mathtt{P}^a_0, \mathtt{V}^a_0)$ for the above relation R^a_0:

$$\Sigma^a_0 = (\Sigma^a_{0,\text{com}}, \Sigma^a_{0,\text{cha}}, \Sigma^a_{0,\text{res}}, \Sigma^a_{0,\text{vrf}}, \Sigma^a_{0,\text{ext}}, \Sigma^a_{0,\text{sim}}). \tag{18}$$

- $\Sigma^a_{0,\text{com}}(x^a_0, w^a_0) \to (\text{COM}^a, \text{COM}_{a,0}, St^a_0)$. This PPT algorithm is executed by $\mathtt{P}^a_0$. On input a statement $x^a_0 = (x^a, c_0)$ and a witness $w^a_0 = (w_0, w^a_1, r_0)$, it runs the algorithms $\Sigma^a_{\text{com}}(x^a, (w_0, w^a_1))$ and $\Sigma_{0,\text{com}}(c_0, (w_0, r_0))$

to obtain the commitment messages and the inner states, $(\textsc{com}^a, St^a)$ and $(\textsc{com}_{a,0}, St_{a,0})$, respectively, with a constraint that the knowledge extractor $\Sigma^a_{0,\mathrm{ext}}$ 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 $St^a_0 := (St^a, St_{a,0})$. It returns $(\textsc{com}^a, \textsc{com}_{a,0}, St^a_0)$. $\mathtt{P}^a_0$ sends $(\textsc{com}^a, \textsc{com}_{a,0})$ to $\mathtt{V}^a_0$ as a commitment message, and keeps the state St^a_0.

- $\Sigma^a_{0,\mathrm{cha}}(x^a_0) \to \textsc{cha}$. This PPT algorithm is executed by $\mathtt{V}^a_0$. On input the statement x^a_0, it reads out the size of the security parameter as 1^λ and chooses a challenge message $\textsc{cha} \in_R \textsc{chaSp}(1^\lambda)$. It returns $\textsc{cha}$. $\mathtt{V}^a_0$ sends $\textsc{cha}$ to $\mathtt{P}^a_0$ as a challenge message.
- $\Sigma^a_{0,\mathrm{res}}(St^a_0, \textsc{cha}) \to (\textsc{res}^a, \textsc{res}_{a,0})$. This PPT algorithm is executed by $\mathtt{P}^a_0$. On input the state St^a_0 and the challenge message $\textsc{cha}$, it runs the algorithms $\Sigma^a_{\mathrm{res}}(St^a, \textsc{cha})$ and $\Sigma_{0,\mathrm{res}}(St_{a,0}, \textsc{cha})$ to obtain the response messages $\textsc{res}^a$ and $\textsc{res}_{a,0}$, 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 $(\textsc{res}^a, \textsc{res}_{a,0})$. $\mathtt{P}^a_0$ sends $(\textsc{res}^a, \textsc{res}_{a,0})$ to $\mathtt{V}^a_0$ as a response message.
- $\Sigma^a_{0,\mathrm{vrf}}(x^a_0, (\textsc{com}^a, \textsc{com}_{a,0}), \textsc{cha}, (\textsc{res}^a, \textsc{res}_{a,0})) \to d$. 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 $(\textsc{com}^a, \textsc{com}_{a,0})$, $\textsc{cha}$ and $(\textsc{res}^a, \textsc{res}_{a,0})$, it runs the algorithms $\Sigma^a_{\mathrm{vrf}}(x^a, \textsc{com}^a, \textsc{cha}, \textsc{res}^a)$ and $\Sigma_{0,\mathrm{vrf}}(c_0, \textsc{com}_{a,0}, \textsc{cha}, \textsc{res}_{a,0})$ 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.
- $\Sigma^a_{0,\mathrm{ext}}(x^a_0, (\textsc{com}^a, \textsc{com}_{a,0}), \textsc{cha}, (\textsc{res}^a, \textsc{res}_{a,0}), \textsc{cha}', (\textsc{res}^{a\prime}, \textsc{res}_{a,0}{}')) \to (\hat{w}^a_0, \hat{w}^a_1, \hat{r}_{a,0})$. 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, $((\textsc{com}^a, \textsc{com}_{a,0}), \textsc{cha}, (\textsc{res}^a, \textsc{res}_{a,0}))$ and $((\textsc{com}^a, \textsc{com}_{a,0}), \textsc{cha}', (\textsc{res}^{a\prime}, \textsc{res}_{a,0}{}'))$, $\textsc{cha} \neq \textsc{cha}'$, it runs the algorithms $\Sigma^a_{\mathrm{ext}}(x^a, \textsc{com}^a, \textsc{cha}, \textsc{res}^a, \textsc{cha}', \textsc{res}^{a\prime})$ and $\Sigma_{0,\mathrm{ext}}(c_0, \textsc{com}_{a,0}, \textsc{cha}, \textsc{res}_{a,0}, \textsc{cha}', \textsc{res}_{a,0}{}')$ 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:

$$\hat{w}^a_0 = \hat{w}_{a,0} \text{ with probability one.} \tag{19}$$

It returns $(\hat{w}^a_0, \hat{w}^a_1, \hat{r}^a_0)$.

- $\Sigma^a_{0,\mathrm{sim}}(x^a_0, \widetilde{\textsc{cha}}) \to ((\widetilde{\textsc{com}}^a, \widetilde{\textsc{com}}_{a,0}), (\widetilde{\textsc{res}}^a, \widetilde{\textsc{res}}_{a,0}))$. 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 $\widetilde{\textsc{cha}} \in_R \textsc{chaSp}(1^\lambda)$, it runs the algorithms $\Sigma^a_{\mathrm{sim}}(x^a, \widetilde{\textsc{cha}})$ and $\Sigma_{0,\mathrm{sim}}(c_0, \widetilde{\textsc{cha}})$ to obtain the remaining part of the transcripts $(\widetilde{\textsc{com}}^a, \widetilde{\textsc{res}}^a)$ and $(\widetilde{\textsc{com}}_{a,0}, \widetilde{\textsc{res}}_{a,0})$, respectively. The simulated messages $((\widetilde{\textsc{com}}^a, \widetilde{\textsc{com}}_{a,0}), \widetilde{\textsc{cha}}, (\widetilde{\textsc{res}}^a, \widetilde{\textsc{res}}_{a,0}))$ should form a distribution $\mathit{dist}\big((\widetilde{\textsc{com}}^a, \widetilde{\textsc{com}}_{a,0}), \widetilde{\textsc{cha}}, (\widetilde{\textsc{res}}^a, \widetilde{\textsc{res}}_{a,0}) \mid \text{generated by } \textsc{chaSp}(1^\lambda) \text{ and } \Sigma^a_{0,\mathrm{sim}}(x^a_0, \widetilde{\textsc{cha}})\big)$ which is identical to the distribution $\mathit{dist}\big((\textsc{com}^a, \textsc{com}_{a,0}), \textsc{cha}, (\textsc{res}^a, \textsc{res}_{a,0}) \mid \text{real accepting transcript}\big)$.

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 $\textsc{com}^a$ and $\textsc{com}_{a,0}$, 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. In Section A we actually demonstrate the construction of Σ^a_0 in an algebraic setting.

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 $R^a, a \in A$:

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

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 $(x^1, (w_0, w_1^1)), \ldots, (x^n, (w_0, w_1^n))$ where $w_0 \in W_0$ and $(x^a, (w_0, w_0^a)) \in R^a$ 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 as sets ('$\simeq$'):

$$R^{a\in A}_{\mathrm{bnd}} \overset{\mathrm{def}}{=} \{\big((x^a)^{a\in A}, w_0, (w_1^a)^{a\in A}\big) \in \{0,1\}^* \times (\{0,1\}^*)^2 \mid (x^a, (w_0, w_1^a)) \in R^a,\ a \in A\} \tag{22}$$

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

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 $\mathrm{poly}^a(\cdot)$ denote for $a \in A$ the polynomial which bounds the bit lengths of witnesses: $|w^a| < \mathrm{poly}^a(|x^a|)$ for $(x^a, w^a) \in R^a$. Let a polynomial $\mathrm{poly}(\cdot)$ be the sum: $\mathrm{poly}(\cdot) := \sum_{a\in A} \mathrm{poly}^a(\cdot)$. Then $\mathrm{poly}(\cdot)$ bounds the bit length of the witness as

$$|w_0, (w_1^a)^{a\in A}| \le |(w_0, w_1^a)^{a\in A}| = |(w^a)^{a\in A}| \le \sum_{a\in A} \mathrm{poly}^a(|x^a|) \le \sum_{a\in A} \mathrm{poly}^a(|(x^a)^{a\in A}|) = \mathrm{poly}(|(x^a)^{a\in A}|). \tag{24}$$

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

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 $W^a = W_0 \times W_1^a, a \in A$. Then the bundled witness space is defined as follows.*

$$W^{a\in A}_{bnd} \overset{def}{=} W_0 \times (W_1^a)^{a\in A}. \tag{25}$$

3.4 Generic Construction of $\varSigma$-protocol for Bundled Witness Space

By using the above $\varSigma$-protocols $(\varSigma_0^a)^{a\in A}$ and a commitment generation algorithm Cmt.Com, we construct an interactive argument system $\varPi^{a\in A}_{\mathrm{bnd}} = (\mathtt{P}, \mathtt{V})$ for the witness relation $R^{a\in A}_{\mathrm{bnd}}$ with a protocol $\varSigma^{a\in A}_{\mathrm{bnd}}$. $\varSigma^{a\in A}_{\mathrm{bnd}}$ is actually a $\varSigma$-protocol, which consists of the six PPT algorithms described below (see also Fig.1):

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

- $\varSigma^{a\in A}_{\mathrm{bnd,com}}((x^a)^{a\in A}, (w_0, (w_1^a)^{a\in A})) \to (c_0, (\textsc{com}^a, \textsc{com}_{a,0})^{a\in A}, St)$. 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 $\varSigma^a_{0,\mathrm{com}}(x_0^a, w_0^a)$ to obtain $(\textsc{com}^a, \textsc{com}_{a,0}, St_0^a)$ for each $a \in A$. It sets the state as $St := (St_0^a)^{a\in A}$. It returns $(c_0, (\textsc{com}^a, \textsc{com}_{a,0})^{a\in A}, St)$. P sends $(c_0, (\textsc{com}^a, \textsc{com}_{a,0})^{a\in A})$ to V as a commitment message, and keeps the state St.
- $\varSigma^{a\in A}_{\mathrm{bnd,cha}}((x^a)^{a\in A}) \to \textsc{cha}$. 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 $\textsc{cha} \in_R \textsc{chaSp}(1^\lambda)$. It returns CHA. $\mathtt{V}_0^a$ sends CHA to $\mathtt{P}_0^a$ as a challenge message.
- $\varSigma^{a\in A}_{\mathrm{bnd,res}}(St, \textsc{cha}) \to (\textsc{res}^a, \textsc{res}_{a,0})^{a\in A}$. This PPT algorithm is executed by P. On input the state St and the challenge message CHA, it runs the algorithms $\varSigma^a_{0,\mathrm{res}}(St_0^a, \textsc{cha})$ to obtain $(\textsc{res}^a, \textsc{res}_{a,0})$ for each $a \in A$. It returns $(\textsc{res}^a, \textsc{res}_{a,0})$. P sends $(\textsc{res}^a, \textsc{res}_{a,0})^{a\in A}$ 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 $\mathtt{V}$. 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 runs the algorithms $\Sigma^a_{0,\mathrm{vrf}}(x^a_0, \mathrm{COM}^a, \mathrm{COM}_{a,0}, \mathrm{CHA}, \mathrm{RES}^a, \mathrm{RES}_{a,0})$ to obtain boolean decisions, for each $a \in A$. If all the decisions are 1, then $\mathtt{V}$ 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 $\mathtt{Cmt}$ is correct, and if Σ^a_0 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 $\mathtt{Cmt}$ 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 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 $x^a_0 := (x^a, c_0)$ for each $a \in A$. Then it runs the algorithms $\Sigma^a_{0,\mathrm{ext}}(x^a_0, (\mathrm{COM}^a, \mathrm{COM}_{a,0}), \mathrm{CHA}, (\mathrm{RES}^a, \mathrm{RES}_{a,0}), \mathrm{CHA}', (\mathrm{RES}^{a\prime}, \mathrm{RES}_{a,0}{}'))$ to obtain $(\hat{w}^a_0, \hat{w}^a_1, \hat{r}^a_0)$ 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 $\hat{w}^a_0 = \hat{w}^{a'}_0$ 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 $\mathtt{Cmt}$ assures that the former case holds with an overwhelming probability, as claimed in the following proposition.

Proposition 2 (Special Soundness) *If $\mathtt{Cmt}$ is correct and computationally binding, and if Σ^a_0 has the special soundness for $a \in A$, 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 $\mathtt{Cmt}$ in accordance with the experiment $\mathbf{Exp}^{\mathrm{bind}}_{\mathtt{Cmt},\mathbf{A}}(1^\lambda)$. $\mathbf{A}$ is given as input the set of public parameter values $\mathtt{PP}_{\mathtt{CmtPrv}}$. $\mathbf{A}$ first reads out the security parameter 1^λ from $\mathtt{PP}_{\mathtt{CmtPrv}}$, and runs the setup algorithms $\Pi.\mathtt{Setup}(1^\lambda)$ to obtain the set of public parameter values $\mathtt{PP}_\Pi$. $\mathbf{A}$ merges the sets of public parameter values as $\mathtt{PP} := (\mathtt{PP}_\Pi, \mathtt{PP}_{\mathtt{CmtPrv}})$. Then $\mathbf{A}$ executes $\Pi^{a\in A}_{\mathrm{bnd}} = (\mathtt{P}, \mathtt{V})$. If the decision d of $\mathtt{V}$ is 1, then $\mathbf{A}$ rewinds $\mathtt{P}$ back to the timing at which $\mathtt{P}$ had sent the challenge message CHA of the protocol $\Sigma^{a\in A}_{\mathrm{bnd}}$. If the decision d of $\mathtt{V}$ is again 1, $\mathbf{A}$ runs 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 $\mathtt{Cmt.Vrf}$ 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 $\textsc{Win}$ in $\mathbf{Exp}^{\mathrm{bind}}_{\mathtt{Cmt},\mathbf{A}}(1^\lambda)$. This completes the description of $\mathbf{A}$, and the following equality holds.

$$\begin{aligned}\mathbf{Adv}^{\mathrm{bind}}_{\mathtt{Cmt},\mathbf{A}}(\lambda) &= \Pr[\Sigma^{a\in A}_{\mathrm{bnd,ext}} \text{ returns } \perp^*] &(27)\\ &= 1 - (\Pr[\Sigma^{a\in A}_{\mathrm{bnd,ext}} \text{ returns}(\hat{w}_0, (\hat{w}^a_1)^{a\in A})] + \Pr[\Sigma^{a\in A}_{\mathrm{bnd,ext}} \text{ returns } \perp]). &(28)\end{aligned}$$

Therefore,

$$\begin{aligned}\Pr[\Sigma^{a\in A}_{\mathrm{bnd,ext}} \text{ returns}(\hat{w}_0, (\hat{w}^a_1)^{a\in A})] &= 1 - (\mathbf{Adv}^{\mathrm{bind}}_{\mathtt{Cmt},\mathbf{A}}(\lambda) + \Pr[\Sigma^{a\in A}_{\mathrm{bnd,ext}} \text{ returns } \perp]) &(29)\\ &= 1 - (\mathbf{Adv}^{\mathrm{bind}}_{\mathtt{Cmt},\mathbf{A}}(\lambda) + (1 - \prod_{a\in A} \Pr[\Sigma^a_{0,\mathrm{ext}} \text{ returns a witness}])). &(30)\end{aligned}$$

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 $\perp] = 0$ 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

```
$\mathtt{P}((x^a)^{a\in A}, w_0, (w_1^a)^{a\in A})$                                  $\mathtt{V}((x^a)^{a\in A})$
  $\Sigma^{a\in A}_{\mathrm{bnd,com}}((x^a)^{a\in A}, w_0, (w_1^a)^{a\in A})$
    $(c_0, r_0) \leftarrow \mathtt{Cmt.Com}(w_0; r_0)$
    For $a \in A$:
      $x_0^a := (x^a, c_0), w_0^a := (w_0, w_1^a, r_0)$
      $\Sigma^a_{0,\mathrm{com}}(x_0^a, w_0^a) \rightarrow (\textsc{com}^a, \textsc{com}_{a,0}, St_0^a)$
    $St := (St_0^a)^{a\in A}$
    Return $(c_0, (\textsc{com}^a, \textsc{com}_{a,0})^{a\in A}, St)$
                                     $c_0, (\textsc{com}^a, \textsc{com}_{a,0})^{a\in A}$
                                     $\rightarrow$
                                                                  $\Sigma^{a\in A}_{\mathrm{bnd,cha}}((x^a)^{a\in A})$
                                                                    $\textsc{cha} \in_R \textsc{chaSp}(1^\lambda)$
                                     $\textsc{cha}$                 Return $\textsc{cha}$
                                     $\leftarrow$
  $\Sigma^{a\in A}_{\mathrm{bnd,res}}(St, \textsc{cha})$
    For $a \in A$:
      $\Sigma^a_{0,\mathrm{res}}(St^a, \textsc{cha}) \rightarrow (\textsc{res}^a, \textsc{res}_{a,0})$
    Return $(\textsc{res}^a, \textsc{res}_{a,0})^{a\in A}$
                                     $(\textsc{res}^a, \textsc{res}_{a,0})^{a\in A}$
                                     $\rightarrow$
                                                                  $\Sigma^{a\in A}_{\mathrm{bnd,vrf}}((x^a)^{a\in A})$
                                                                    For $a \in A$:
                                                                      $x_0^a := (x^a, c_0)$
                                                                      $\Sigma^a_{0,\mathrm{vrf}}(x_0^a, (\textsc{com}^a, \textsc{com}_{a,0}), \textsc{cha}, (\textsc{res}^a, \textsc{res}_{a,0}))$
                                                                        $=_? 1$
                                                                    If $\textsc{true}$ for all $a \in A$, then Return $d := 1$
                                                                    else Return $d := 0$
                                                                  Return $d$
```

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 $R^{a\in A}_{\mathrm{bnd}}$.

simulation of an accepting transcript. On input a statement $(x^a)^{a\in A}$ and a uniform random string $\widetilde{\textsc{cha}} \in_R \textsc{chaSp}(1^\lambda)$, 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 $\Sigma^a_{0,\mathrm{sim}}(x_0^a, \widetilde{\textsc{cha}})$ to obtain $((\widetilde{\textsc{com}}^a, \widetilde{\textsc{com}}_{a,0}), (\widetilde{\textsc{res}}^a, \widetilde{\textsc{res}}_{a,0}))$ for each $a \in A$. It returns $((\tilde{c}_0, (\widetilde{\textsc{com}}^a, \widetilde{\textsc{com}}_{a,0})^{a\in A}), (\widetilde{\textsc{res}}^a, \widetilde{\textsc{res}}_{a,0})^{a\in A})$.

Proposition 3 (Honest-Verifyer Zero-Knowledge) *If $\mathtt{Cmt}$ is perfectly hiding, and if Σ_0^a is honest-verifier zero-knowledge for $a \in A$, 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 $(x_0^a, \widetilde{\textsc{cha}})$, the simulator $\Sigma^a_{0,\mathrm{sim}}$ works to return the remaining part of the simulated transcript, $((\widetilde{\textsc{com}}^a, \widetilde{\textsc{com}}_{a,0}), (\widetilde{\textsc{res}}^a, \widetilde{\textsc{res}}_{a,0}))$, for each $a \in A$. Then, the merged transcripts $((\tilde{c}_0, (\widetilde{\textsc{com}}^a, \widetilde{\textsc{com}}_{a,0})^{a\in A}), (\widetilde{\textsc{res}}^a, \widetilde{\textsc{res}}_{a,0})^{a\in A})$ is identically distributed to the real. □

Theorem 1 *If $\mathtt{Cmt}$ is correct, computationally binding and perfectly hiding, and if Σ_0^a is a Σ-protocol for $a \in A$, then our protocol $\Sigma^{a\in A}_{bnd}$ 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_0^a with Σ_0^a is perfectly witness-indistinguishable for each $a \in A$, and if $\mathtt{Cmt}$ 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 $dist^{a\in A} := dist\big((c_0, (\textsc{com}^a, \textsc{com}_{a,0})^{a\in A}), \textsc{cha}, (\textsc{res}^a, \textsc{res}_{a,0})^{a\in A}\big)$, where the challenge message $\textsc{cha}$ is chosen by any given $\textsc{ppt}$ verifier $\mathtt{V}^*$ on input a set of statements $(x^a)^{a\in A}$, any given auxiliary input z and a commitment message $(c_0, (\textsc{com}^a, \textsc{com}_{a,0})^{a\in A})$. If $\mathtt{Cmt}$ 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_0^a is perfectly witness-indistinguishable, then the distribution of the commitment message and the response message $dist\big((\textsc{com}^a, \textsc{com}_{a,0}), (\textsc{res}^a, \textsc{res}_{a,0})\big)$ are identical even if the witness (w_0, w_1^a) varies and even if $\textsc{cha}$ chosen by $\mathtt{V}^*((x^a)^{a\in A}, z, (c_0, (\textsc{com}^a, \textsc{com}_{a,0})^{a\in A}))$ deviates from the uniform random distribution. Therefore, for all $a \in A$, the distribution $dist^{a\in A}$ is identical even if the witness $(w_0, (w_1^a)_{a\in A})$ varies. □

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

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.

4.1 Syntax and Security Definitions

Our scheme $\mathtt{a\text{-}auth}$ consists of five PPT algorithms, ($\mathtt{Setup}$, $\mathtt{AuthKG}$, $\mathtt{PrivKG}$, $\mathtt{P}$, $\mathtt{V}$).

- $\mathtt{Setup}(1^\lambda) \to \mathtt{PP}$. This PPT algorithm is needed to generate a set of public parameter values $\mathtt{PP}$. On input the security parameter 1^λ, it generates the set of values $\mathtt{PP}$. It returns $\mathtt{PP}$.
- $\mathtt{AuthKG}(\mathtt{PP}, a) \to (\mathrm{PK}^a, \mathrm{MSK}^a)$. This PPT algorithm is executed by a key-issuing authority indexed by a positive integer a. On input the set of public parameter values $\mathtt{PP}$ 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 $(\mathrm{PK}^a, \mathrm{MSK}^a)$.
- $\mathtt{PrivKG}(\mathtt{PP}, \mathrm{PK}^a, \mathrm{MSK}^a, \mathtt{gid}) \to \mathrm{sk}_{\mathtt{gid}}^a$. This PPT algorithm is executed by the a-th key-issuing authority. On input the set of public parameter values $\mathtt{PP}$, the a-th public and master secret keys $(\mathrm{PK}^a, \mathrm{MSK}^a)$ and a string $\mathtt{gid}$ of a prover (a global identity string), it generates a private secret key $\mathrm{sk}_{\mathtt{gid}}^a$ of a prover. It returns $\mathrm{sk}_{\mathtt{gid}}^a$.
- $\langle \mathtt{P}(\mathtt{PP}, (\mathrm{PK}^a, \mathrm{sk}_{\mathtt{gid}}^a)^{a\in A'}), \mathtt{V}(\mathtt{PP}, (\mathrm{PK}^a)^{a\in A'})\rangle \to d$. 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 set of public parameter values $\mathtt{PP}$ and the public keys $(\mathrm{PK}^a)^{a\in A}$ to $\mathtt{P}$ and $\mathtt{V}$ and the corresponding private secret keys $(\mathrm{sk}_{\mathtt{gid}}^a)^{a\in A}$ to $\mathtt{P}$, $\mathtt{P}$ and $\mathtt{V}$ interact with each other. After at most polynomially many (in λ) moves of messages between $\mathtt{P}$ and $\mathtt{V}$, $\mathtt{V}$ returns $d := 1$ ("accept") or $d := 0$ ("reject").

We discuss two security notions for our authentication scheme $\mathtt{a\text{-}auth}$; 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 possible attacks to cause misauthentication is the concurrent and collusion attack on our $\mathtt{a\text{-}auth}$. For a formal treatment we define the following experiment on $\mathtt{a\text{-}auth}$ and an adversary algorithm $\mathbf{A}$.

$$\mathbf{Expr}_{\mathtt{a\text{-}auth},\mathbf{A}}^{\text{conc-coll}}(1^\lambda): \tag{31}$$

$$q_A \leftarrow \mathbf{A}(1^\lambda), A := \{1, \ldots, q_A\}, \mathtt{PP} \leftarrow \mathtt{Setup}(1^\lambda), \text{For } a \in A : (\mathrm{PK}^a, \mathrm{MSK}^a) \leftarrow \mathtt{AuthKG}(\mathtt{PP}, a) \tag{32}$$

$$q_I \leftarrow \mathbf{A}(\mathtt{PP}, (\mathrm{PK}^a)^{a\in A}), I := \{1, \ldots, q_I\}, \text{For } i \in I : \mathtt{gid}_i \in_R \{0,1\}^\lambda \tag{33}$$

$$\text{For } a \in A : \text{For } i \in I : \mathrm{sk}_{\mathtt{gid}_i}^a \leftarrow \mathtt{PrivKG}(\mathtt{PP}, \mathrm{PK}^a, \mathrm{MSK}^a, \mathtt{gid}_i) \tag{34}$$

$$(A^*, St^*) \leftarrow \mathbf{A}^{\mathtt{P}(\mathtt{PP},(\mathrm{PK}^a,\mathrm{sk}_{\mathtt{gid}_i}^a)^{a\in A})|_{i\in I}, \mathbf{PrivKO}(\mathtt{PP},\mathrm{PK}^\cdot,\mathrm{MSK}^\cdot,\cdot)}(\mathtt{PP}, (\mathrm{PK}^a)^{a\in A}) \tag{35}$$

$$\langle \mathbf{A}(St^*), \mathtt{V}(\mathtt{PP}, (\mathrm{PK}^a)^{a\in A^*})\rangle \to d, \text{If } d = 1 \text{ then Return } \textsc{Win} \text{ else Return } \textsc{Lose} \tag{36}$$

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 $\mathtt{PP}$ 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,

$$A_j := \{a \in A \mid \mathbf{A} \text{ receives } \mathrm{sk}_{\mathtt{gid}_j}^a\}, j = q_I + 1, \ldots, q_I + q_{\mathrm{sk}}. \tag{37}$$

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 $\mathtt{V}$ is 1, then the experiment returns $\textsc{Win}$ and otherwise, returns $\textsc{Lose}$.

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:

$$A^* \nsubseteq A_j,\ j = q_I + 1, \dots, q_I + q_{\mathrm{sk}}. \tag{38}$$

This restriction is because, otherwise, $\mathbf{A}$ is given private secret keys for A^* on a single $\mathtt{gid}_{i^*}$ for some i^*, $q_I < i^* \le q_I + q_{\mathrm{sk}}$, and then $\mathbf{A}$ can trivially be accepted in the attacking phase.

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

$$\mathbf{Expr}^{\text{ano}}_{\mathtt{a\text{-}auth},\mathbf{A}}(1^\lambda): \tag{39}$$

$$q_A \leftarrow \mathbf{A}(1^\lambda), A := \{1, \dots, q_A\}, \mathtt{PP} \leftarrow \mathtt{Setup}(1^\lambda), \text{For } a \in A : (\mathrm{PK}^a, \mathrm{MSK}^a) \leftarrow \mathtt{AuthKG}(\mathtt{PP}, a) \tag{40}$$

$$\mathtt{gid}_0, \mathtt{gid}_1 \leftarrow \mathbf{A}(\mathtt{PP}, (\mathrm{PK}^a)^{a\in A}), \text{For } a \in A : \text{For } i \in 0,1 : \mathrm{sk}^a_{\mathtt{gid}_i} \leftarrow \mathtt{PrivKG}(\mathtt{PP}, \mathrm{PK}^a, \mathrm{MSK}^a, \mathtt{gid}_i) \tag{41}$$

$$b \in_R \{0,1\}, b^* \leftarrow \mathbf{A}^{\mathtt{P}(\mathtt{PP},(\mathrm{PK}^a,\mathrm{sk}^a_{\mathtt{gid}_b})^{a\in A})}(\mathtt{PP}, (\mathrm{PK}^a, \mathrm{sk}^a_{\mathtt{gid}_0}, \mathrm{sk}^a_{\mathtt{gid}_1})^{a\in A}) \tag{42}$$

$$\text{If } b = b^*,\ \text{then Return } \textsc{Win},\ \text{else Return } \textsc{Lose} \tag{43}$$

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 $\mathtt{P}$ 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 $\textsc{Win}$ and otherwise, returns $\textsc{Lose}$.

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

4.2 Generic Construction

We give a generic construction of our authentication scheme $\mathtt{a\text{-}auth}$. 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 $\mathtt{Sig}$. We note that a commit-and-prove scheme $\mathtt{CmtPrv}$ is employed in $\Sigma^{a\in A}_{\text{bnd}}$.

• $\mathtt{Setup}(1^\lambda) \rightarrow \mathtt{PP}$. 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 $\mathtt{Sig}$, the interactive argument systems $(\Pi^a_0)^{a\in A}$, and the commitment generation algorithm $\mathtt{Cmt.Com}$. 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 $\mathtt{PP}$.

• $\mathtt{AuthKG}(\mathtt{PP}, a) \rightarrow (\mathrm{PK}^a, \mathrm{MSK}^a)$. On input the set of public parameter values $\mathtt{PP}$ 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 $(\mathrm{PK}^a, \mathrm{MSK}^a)$.

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

```
$\texttt{Setup}(1^\lambda)$
  $\texttt{PP}_{\texttt{Sig}} \leftarrow \texttt{Sig.Setup}(1^\lambda)$
  $\texttt{PP}_{\Pi} \leftarrow \Pi\texttt{.Setup}(1^\lambda)$
  $\texttt{PP}_{\texttt{CmtPrv}} \leftarrow \texttt{CmtPrv.Setup}(1^\lambda)$
  $\texttt{PP} := (\texttt{PP}_{\Pi}, \texttt{PP}_{\texttt{CmtPrv}}, \texttt{PP}_{\texttt{Sig}})$
  Return $\texttt{PP}$

$\texttt{AuthKG}(\texttt{PP}, a)$
  $(\mathrm{SK}, \mathrm{PK}) \leftarrow \texttt{Sig.KG}(\texttt{PP}_{\texttt{Sig}})$
  $\mathrm{PK}^a := \mathrm{PK}, \mathrm{MSK}^a := \mathrm{SK}$
  Return $(\mathrm{PK}^a, \mathrm{MSK}^a)$

$\texttt{PrivKG}(\texttt{PP}, \mathrm{PK}^a, \mathrm{MSK}^a, \texttt{gid})$
  $\sigma^a_{\texttt{gid}} \leftarrow \texttt{Sig.Sign}(\texttt{PP}_{\texttt{Sig}}, \mathrm{PK}^a, \mathrm{MSK}^a, \texttt{gid})$
  $\mathrm{sk}^a_{\texttt{gid}} := \sigma^a_{\texttt{gid}}$
  Return $\mathrm{sk}^a_{\texttt{gid}}$

$\texttt{P}(\texttt{PP}, (\mathrm{PK}^a)^{a \in A}, (\mathrm{sk}^a_{\texttt{gid}})^{a \in A})$
  For $a \in A$: $x^a := \mathrm{PK}^a, w^a_1 := \mathrm{sk}^a_{\texttt{gid}}$
  $w_0 := \texttt{gid}$

                    (Execute $\Sigma^{a \in A}_{\mathrm{bnd}}$)

$\texttt{V}(\texttt{PP}, (\mathrm{PK}^a)^{a \in A})$
  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.

- $\texttt{P}(\texttt{PP}, (\mathrm{PK}^a)^{a \in A}, (\mathrm{sk}^a_{\texttt{gid}})^{a \in A})$ and $\texttt{V}(\texttt{PP}, (\mathrm{PK}^a)^{a \in A})$. 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 $(\mathrm{sk}^a_{\texttt{gid}})^{a \in A}$ 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 $w_0 := \texttt{gid}$ and $w^a_1 := \mathrm{sk}^a_{\texttt{gid}}$ for $a \in A$. The witness spaces $W^a, a \in A$ are described as follows.

$$W^a = W_0 \times W^a_1, \tag{44}$$

$$W_0 = \{\texttt{gid} \mid \text{string of length } \lambda\} = \{0,1\}^\lambda, \tag{45}$$

$$W^a_1 = \{\sigma^a_{\texttt{gid}} \mid \sigma^a_{\texttt{gid}} \leftarrow \texttt{Sig.Sign}(\texttt{PP}_{\texttt{Sig}}, \mathrm{PK}^a, \mathrm{MSK}^a, \texttt{gid}) \text{ for some } \texttt{gid} \in W_0\}. \tag{46}$$

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

$$\mathbf{Adv}^{\textit{conc-coll}}_{\textit{a-auth},\mathbf{A}}(\lambda) \leq \frac{1}{\mathrm{CHASP}(1^\lambda)} + \sqrt{\frac{2^\lambda}{2^\lambda - 1} \cdot q_A \cdot \mathbf{Adv}^{\textit{euf-cma}}_{\textit{Sig},\mathbf{F}}(\lambda) + \mathbf{Adv}^{\textit{bind}}_{\textit{Cmt},\mathbf{B}}(\lambda)}. \tag{47}$$

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

$$\text{For } a \in A \text{ s.t. } a \neq a^* : (\mathrm{PK}^a, \mathrm{MSK}^a) \leftarrow \texttt{AuthKG}(\texttt{PP}, a), \tag{48}$$

$$\mathrm{PK}^{a^*} := \mathrm{PK}_{\texttt{Sig}}. \tag{49}$$

F inputs the set of public parameter values $\mathtt{PP}$ and the public keys $(\mathrm{PK}^a)^{a\in A}$ into **A** to obtain the number q_I of concurrent provers. **F** sets I as $I := \{1, \ldots, q_I\}$.
Simulation of Concurrent Provers **F** chooses a *single* identity string $\tilde{\mathtt{gid}} \in_R \{0,1\}^\lambda$. For $a \in A$ *except* the target index a^*, **F** runs the private secret key generation algorithm with $\tilde{\mathtt{gid}}$ honestly. As for a^*, **F** issues a signing query with $\tilde{\mathtt{gid}}$:

$$\text{For } a \in A \text{ s.t. } a \neq a^* : \mathrm{sk}^a_{\tilde{\mathtt{gid}}} \leftarrow \mathtt{PrivKG}(\mathtt{PP}, \mathrm{PK}^a, \mathrm{MSK}^a, \tilde{\mathtt{gid}}), \tag{50}$$

$$\mathrm{sk}^{a^*}_{\tilde{\mathtt{gid}}} \leftarrow \mathbf{SignO}(\mathtt{PP}_{\mathtt{Sig}}, \mathrm{PK}_{\mathtt{Sig}}, \mathrm{SK}_{\mathtt{Sig}}, \tilde{\mathtt{gid}}). \tag{51}$$

In the simulation of concurrent provers $\mathtt{P}(\mathtt{PP}, (\mathrm{PK}^a, \mathrm{sk}^a_{\mathtt{gid}_i})^{a\in A})|_{i\in I}$ which **A** interacts with, **F** uses the private secret keys $(\mathrm{sk}^a_{\tilde{\mathtt{gid}}})^{a\in A}$. Note that this is a perfect simulation. *This is because* of the perfect witness-indistinguishability of our $\Sigma^{a\in A}_{\mathrm{bnd}}$ (Theorem 2).
Simulation of Private Secret Key Oracle When **A** issues a private secret key query with $a \in A$ and $\mathtt{gid}_j \in \{0,1\}^\lambda$ $(q_I + 1 \leq j \leq q_I + q_{\mathrm{sk}})$, if $a \neq a^*$, then **F** runs the private secret key generation algorithm with $\mathtt{gid}_j$ honestly, and otherwise (i.e. $a = a^*$), **F** issues a signing query with $\mathtt{gid}_j$:

$$\text{If } a \in A \text{ s.t. } a \neq a^* : \mathrm{sk}^a_{\mathtt{gid}_j} \leftarrow \mathtt{PrivKG}(\mathtt{PP}, \mathrm{PK}^a, \mathrm{MSK}^a, \mathtt{gid}_j), \tag{52}$$

$$\text{otherwise (i.e. } a = a^*) \ \mathrm{sk}^{a^*}_{\mathtt{gid}_j} \leftarrow \mathbf{SignO}(\mathtt{PP}_{\mathtt{Sig}}, \mathrm{PK}_{\mathtt{Sig}}, \mathrm{SK}_{\mathtt{Sig}}, \mathtt{gid}_j). \tag{53}$$

F replies to **A** with the secret key $\mathrm{sk}^a_{\mathtt{gid}_j}$. This is also a perfect simulation.

At the end of the "learning phase" **A** outputs a target set of authority indices A^* and its inner state St^*.
Generating Existential Forgery Next, in the "attacking phase", on input the inner state St^*, the adversary **A** interacts with the verifier. That is, **F** runs a verifier $\mathtt{V}$ with input $(\mathtt{PP}, (\mathrm{PK}^a)^{a\in A^*})$. If the decision d of $\mathtt{V}$ is 1, then **F** *rewinds* (Bellare-Palacio [BP02]) **A** back to the timing at which **A** had sent the challenge message of the Σ-protocol $\Sigma^{a\in A}_{\mathrm{bnd}}$. If the decision d of $\mathtt{V}$ is again 1, **F** runs 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 $\mathtt{gid}^*$ as $\mathtt{gid}^* := \hat{w}_0$, and a signature σ^* as $\sigma^* := \hat{w}^{a^*}_1$. **F** returns $(\mathtt{gid}^*, \sigma^*)$. This completes the description of **F**.

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

$$\textsc{Acc} : \text{the event that } \mathtt{V} \text{ accepts } \mathbf{A}, \tag{54}$$

$$\textsc{Rst} : \text{the event that } \mathtt{V} \text{ accepts } \mathbf{A} \text{ both before and after the rewinding with different } \mathrm{CHA}, \tag{55}$$

$$\textsc{TgtAuthIdx} : \text{the event that } \hat{a} = a^*, \tag{56}$$

$$\textsc{Ext} : \text{the event that } \Sigma^{a\in A}_{\mathrm{bnd,ext}} \text{ returns a witness } \hat{w} := (\hat{w}_0, (\hat{w}^a_1)^{a\in A}), \tag{57}$$

$$\textsc{NewID} : \text{the event that } \mathtt{gid}^* \neq \tilde{\mathtt{gid}}, \tag{58}$$

$$\textsc{Forge} : \text{the event that } (\mathtt{gid}^*, \sigma^*) \text{ is an existential forgery on } \mathtt{Sig}. \tag{59}$$

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

$$\Pr[\textsc{Acc}] \leq \frac{1}{\mathrm{CHA}\textsc{Sp}(1^\lambda)} + \sqrt{\Pr[\textsc{Rst}]}. \tag{60}$$

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

$$\mathbf{Adv}^{\text{conc-coll}}_{\text{a-auth},\mathbf{A}}(\lambda) = \Pr[\textsc{Acc}], \tag{61}$$

$$\Pr[\textsc{TgtAuthIdx}, \textsc{Rst}, \textsc{Ext}, \textsc{NewID}] = \Pr[\textsc{Forge}], \tag{62}$$

$$\Pr[\textsc{Forge}] = \mathbf{Adv}^{\text{euf-cma}}_{\mathtt{Sig},\mathbf{F}}(\lambda). \tag{63}$$

The left-hand side of the equality (62) is expanded as follows.

$$\Pr[\textsc{TgtAuthIdx}, \textsc{Rst}, \textsc{Ext}, \textsc{NewID}] = \Pr[\textsc{TgtAuthIdx}] \cdot \Pr[\textsc{Rst}, \textsc{Ext}, \textsc{NewID}] \tag{64}$$

$$= \Pr[\textsc{TgtAuthIdx}] \cdot \Pr[\textsc{Rst}, \textsc{Ext}] \cdot \Pr[\textsc{NewID} \mid \textsc{Rst}, \textsc{Ext}]. \tag{65}$$

Lemma 1

$$\Pr[\textsc{TgtAuthIdx}] = 1/q_A. \tag{66}$$

Proof. The restriction (38) of the experiment assures that there exists an authority index $\hat{a}$ such that $\hat{a} \in A^*, \hat{a} \notin A_j, q_I \leq \forall j \leq q_I + q_{\mathrm{sk}}$. Besides, $\hat{a}$ coincides with a^* with probability $1/q_A$. Therefore, the string $\mathtt{gid}^*$ is different from all the queried strings $\mathtt{gid}_j, q_I \leq \forall j \leq q_I + q_{\mathrm{sk}}$ with probability $1/q_A$. □

Lemma 2

$$\Pr[\textsc{NewID} \mid \textsc{Rst}, \textsc{Ext}] = \frac{2^\lambda - 1}{2^\lambda}. \tag{67}$$

Proof. The string $\tilde{\mathtt{gid}}$ for the simulation of concurrent provers is hidden from the view of $\mathbf{A}$. Therefore $\mathtt{gid}^*$ is different from $\tilde{\mathtt{gid}}$ with probability $\frac{2^\lambda-1}{2^\lambda}$. □

Lemma 3 *For any given* PPT *algorithm* $\mathbf{A}$ *that executes a concurrent and collusion attack on our* **a-auth** *in accordance with the experiment* $\mathbf{Expr}^{\textit{conc-coll}}_{\textit{a-auth},\mathbf{A}}(1^\lambda)$*, there exists a* PPT *algorithm* $\mathbf{B}$ *that breaks the bandaging property of* `Cmt` *in accordance with the experiment* $\mathbf{Exp}^{\textit{bind}}_{\textit{Cmt},\mathbf{B}}(1^\lambda)$ *satisfying the following equality.*

$$\Pr[\textsc{Rst}, \overline{\textsc{Ext}}] = \mathbf{Adv}^{\textit{bind}}_{\textit{Cmt},\mathbf{B}}(\lambda). \tag{68}$$

Proof. Given any PPT algorithm $\mathbf{A}$ on $\mathbf{Expr}^{\text{conc-coll}}_{\text{a-auth},\mathbf{A}}(1^\lambda)$, we construct a PPT algorithm $\mathbf{B}$ that breaks the binding property of `Cmt` in accordance with the experiment $\mathbf{Exp}^{\text{bind}}_{\mathtt{Cmt},\mathbf{B}}(1^\lambda)$. $\mathbf{B}$ is given as input the set of public parameter values $\mathtt{PP}_{\mathtt{CmtPrv}}$. $\mathbf{B}$ first reads out the security parameter 1^λ from $\mathtt{PP}_{\mathtt{CmtPrv}}$, and runs the setup algorithms $\mathit{\Pi}.\mathtt{Setup}(1^\lambda)$ and $\mathtt{Sig.Setup}(1^\lambda)$ to obtain the sets of public parameter values $\mathtt{PP}_{\mathit{\Pi}}$ and $\mathtt{PP}_{\mathtt{Sig}}$, respectively. $\mathbf{B}$ merges the sets of public parameter values as $\mathtt{PP} := (\mathtt{PP}_{\mathtt{Sig}}, \mathtt{PP}_{\mathit{\Pi}}, \mathtt{PP}_{\mathtt{CmtPrv}})$. Then $\mathbf{B}$ invokes the algorithm $\mathbf{A}$ with 1^λ to obtain 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", $\mathbf{B}$ runs a verifier `V` with input $(\mathtt{PP}, (\mathrm{PK}^a)^{a \in A^*})$. 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 $\mathit{\Sigma}$-protocol $\mathit{\Sigma}^{a \in A}_{\mathrm{bnd}}$. If the decision d of `V` is again 1, $\mathbf{B}$ runs the knowledge extractor $\mathit{\Sigma}^{a \in A}_{\mathrm{bnd,ext}}$ 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 $\mathit{\Sigma}^{a \in A}_{\mathrm{bnd,ext}}$ 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 `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 $\textsc{Win}$ in $\mathbf{Exp}^{\text{bind}}_{\mathtt{Cmt},\mathbf{B}}(1^\lambda)$. This completes the description of $\mathbf{B}$, and $\mathbf{B}$ satisfies (68). □

Note that we have the equality:

$$\Pr[\textsc{Rst}] = \Pr[\textsc{Rst}, \textsc{Ext}] + \Pr[\textsc{Rst}, \overline{\textsc{Ext}}]. \tag{69}$$

Combining (62), (65), (66), (67), (68) and (69), we have:

$$\Pr[\textsc{Rst}] = \frac{2^\lambda}{2^\lambda - 1} \cdot q_A \cdot \Pr[\textsc{Forge}] + \mathbf{Adv}^{\text{bind}}_{\mathtt{Cmt},\mathbf{B}}(\lambda). \tag{70}$$

Combining (60), (61), (70) and (63), we have:

$$\mathbf{Adv}^{\text{conc-coll}}_{\text{a-auth},\mathbf{A}}(\lambda) \leq \frac{1}{\mathrm{CHA}\textsc{Sp}(1^\lambda)} + \sqrt{\frac{2^\lambda}{2^\lambda - 1} \cdot q_A \cdot \mathbf{Adv}^{\text{euf-cma}}_{\mathtt{Sig},\mathbf{F}}(\lambda) + \mathbf{Adv}^{\text{bind}}_{\mathtt{Cmt},\mathbf{B}}(\lambda)}. \tag{71}$$

□

Theorem 4 *If the component proof system* $\mathit{\Pi}^a_0$ *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* PPT *algorithm* $\mathbf{A}$ *that executes the anonymity game on our* **a-auth** *in accordance with the experiment* $\mathbf{Expr}^{\textit{ano}}_{\textit{a-auth},\mathbf{A}}(1^\lambda)$*, the following equality holds.*

$$\mathbf{Adv}^{\textit{ano}}_{\textit{a-auth},\mathbf{A}}(\lambda) = 0. \tag{72}$$

Proof. The perfect witness-indistinguishability of $\mathit{\Pi}_0^a$ for each $a \in A$ and the perfectly hiding property of the commitment scheme `Cmt` assure that our proof system $\mathit{\Pi}_{\mathrm{bnd}}^{a \in A}$ is perfectly witness-indistinguishable by Theorem 2. Then the two distribution $\mathit{dist}^{a \in A} := \mathit{dist}\big((c_0, (\mathrm{COM}^a, \mathrm{COM}_{a,0})^{a \in A}), \mathrm{CHA}, (\mathrm{RES}^a, \mathrm{RES}_{a,0})^{a \in A}\big)$ is identical even if the auxiliary input z is private secret keys $(\mathrm{sk}_{\mathsf{gid}_0}^a, \mathrm{sk}_{\mathsf{gid}_1}^a)^{a \in A}$. Therefore, the advantage $\mathbf{Adv}_{\mathsf{a\text{-}auth},\mathbf{A}}^{\mathrm{ano}}(\lambda)$ is zero. □

Note. Relation with 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 gives a decentralized multi-authority attribute-based signature scheme [OT13].

5 Conclusion

We proposed a generic construction of a $\mathit{\Sigma}$-protocol of commit-and-prove type, which is an AND-composition of $\mathit{\Sigma}$-protocols on the statements that include a common commitment. When the component $\mathit{\Sigma}$-protocols are of witness-indistinguishable argument systems, our $\mathit{\Sigma}$-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.

A post-quantum instantiation should be our future work.

References

[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 *ASIAPKC2014*, volume 2 of *ACM-ASIAPKC*, pages 49–58. ACM, 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 *CRYPTO '94*, pages 174–187. Springer-Verlag, 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] T. Okamoto and K. Takashima. Decentralized attribute-based signatures. In *PKC 2013*, volume 7778 of *LNCS*, pages 125–142. Springer, 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

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

Basically, we can employ any three building blocks that satisfy requirements stated in Section 4. Below we 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 $\mathtt{Sig}^{\mathtt{CL}}$ (See Appendix C) [Oka06,SNF11,TF12], the pairing version of the Camenisch-Lysyanskaya perfectly witness-indistinguishable argument of knowledge system $\Pi^{\mathtt{CL}}$ (See Appendix D) [Oka06,SNF11,TF12], and the Pedersen-Okamoto commit-and-prove scheme $\mathtt{CmtPrv}^{\mathtt{PO}}$ (See Appendix E) which is a combination of the perfectly hiding commitment scheme of Pedersen [Ped91] and the perfectly witness-indistinguishable argument of knowledge system by Okamoto [Oka92].

We obtain the following propositions and theorems. (The details and proofs are omitted.)

Proposition 4 *$\Sigma^{\mathit{CL},a}_{\mathit{PO},0}$ is a Σ-protocol.*

Proposition 5 *$\Pi^{\mathit{CL},a}_{\mathit{PO},0}$ is perfectly witness indistinguishable.*

Theorem 5 *If $\mathtt{CmtPrv}^{\mathtt{PO}}$ is perfectly hiding and computationally binding, and if $\mathtt{Sig}^{\mathtt{CL}}$ is existentially unforgeable against chosen-message attacks, then `a-auth` is secure against concurrent and collusion attacks.*

Theorem 6 *`a-auth` is anonymous.*

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:

$$\mathbf{Adv}^{\mathrm{dl}}_{\mathcal{G},\mathbf{S}}(\lambda) := \Pr[\gamma = \gamma^* \mid \Lambda \leftarrow \mathcal{G}(1^\lambda), \gamma \in_R \mathbb{Z}_p, \gamma^* \leftarrow \mathbf{S}(\Lambda, G, G^\gamma)]. \tag{73}$$

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:

$$\mathbf{Adv}^{\mathrm{sdh}}_{\mathcal{BG},\mathbf{S}}(\lambda) := \Pr[V^{\gamma+e} = G \mid \Lambda \leftarrow \mathcal{BG}(1^\lambda), \gamma \in_R \mathbb{Z}_p, (V, e) \leftarrow \mathbf{S}(\Lambda, (\tilde{G}^\gamma, \tilde{G}^{\gamma^2}, \ldots, \tilde{G}^{\gamma^q}))]. \tag{74}$$

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 $\mathtt{Sig}^{\mathtt{CL}}$, 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]. $\mathtt{Sig}^{\mathtt{CL}}$ consists of four PPT algorithms, $\mathtt{Sig}^{\mathtt{CL}} := (\mathtt{Sig}^{\mathtt{CL}}.\mathtt{Setup}, \mathtt{Sig}^{\mathtt{CL}}.\mathtt{KG}, \mathtt{Sig}^{\mathtt{CL}}.\mathtt{Sign}, \mathtt{Sig}^{\mathtt{CL}}.\mathtt{Vrf})$.

- $\mathtt{Sig}^{\mathtt{CL}}.\mathtt{Setup}(1^\lambda) \to \mathtt{PP}$. On input the security parameter 1^λ, this PPT algorithm generates a set of public parameter values. That is, it runs 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 $\mathtt{PP} := (\lambda, G_0, G_1, G_2, \tilde{G}_0)$.
- $\mathtt{Sig}^{\mathtt{CL}}.\mathtt{KG}(\mathtt{PP}) \to (\mathrm{PK}, \mathrm{SK})$. On input $\mathtt{PP}$, 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 $(\mathrm{PK}, \mathrm{SK})$.
- $\mathtt{Sig}^{\mathtt{CL}}.\mathtt{Sign}(\mathrm{PK}, \mathrm{SK}, m) \to \sigma$. On input $\mathtt{PP}$, PK, SK and a message $m \in \mathbb{Z}_p$, this PPT algorithm chooses two randomnesses $\gamma, \delta \in \mathbb{Z}_p$. It computes $V := (G_0 G_1^m G_2^\gamma)^{1/(\delta+\alpha)}$. It sets a signature $\sigma := (V, \gamma, \delta)$. It returns σ.
- $\mathtt{Sig}^{\mathtt{CL}}.\mathtt{Vrf}(\mathrm{PK}, m, \sigma) \to 1/0$. On input $\mathtt{PP}$, PK, m and σ, this deterministic polynomial time algorithm returns a boolean decision 1 if the following holds. Otherwise, 0: $e(G_0 G_1^m G_2^\gamma) =_? e(V, \tilde{G}_0^\delta \tilde{G}_1)$.

The pairing version of the Camenisch-Lysyanskaya signature scheme $\mathtt{Sig}^{\mathtt{CL}}$ 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 $\Pi^{\mathtt{CL}}$, 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]. $\Pi^{\mathtt{CL}} = (\Pi^{\mathtt{CL}}.\mathtt{Setup}, \mathtt{P}, \mathtt{V})$ is executed in accordance with a Σ-protocol $\Sigma^{\mathtt{CL}} = (\Sigma^{\mathtt{CL}}_{\mathrm{com}}, \Sigma^{\mathtt{CL}}_{\mathrm{cha}}, \Sigma^{\mathtt{CL}}_{\mathrm{res}}, \Sigma^{\mathtt{CL}}_{\mathrm{vrf}}, \Sigma^{\mathtt{CL}}_{\mathrm{ext}}, \Sigma^{\mathtt{CL}}_{\mathrm{sim}})$.

The setup algorithm $\Pi^{\mathtt{CL}}.\mathtt{Setup}$ is the same as $\mathtt{Sig}^{\mathtt{CL}}.\mathtt{Setup}(1^\lambda)$. The set of public parameter values $\mathtt{PP}$ is common.

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

$$e(G_0 G_1^{\mathtt{gid}} G_2^\gamma, \tilde{G}_0) = e(V, \tilde{G}_0^\delta \tilde{G}_1). \tag{75}$$

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, $(\mathtt{gid}, \gamma, \delta)$. In other words, this number is the number of the solutions of the *equation* (75) determined by $\mathtt{PP}$ and x.

The protocol between $\mathtt{P}$ and $\mathtt{V}$ is a Σ-protocol. It goes as follows.

- $\Sigma^{\mathtt{CL}}_{\mathrm{com}}(x, w) \to (\textsc{com}, St)$. This PPT algorithm is executed by $\mathtt{P}$. On input a statement x and a witness w, it chooses $v \in_R \mathbb{Z}_p$ and re-randomize the secret element V as $R := VG_2^v$. It puts $z := \gamma + v\delta$. It chooses $r_{\mathtt{gid}}, r_z, r_v, r_\delta \in_R \mathbb{Z}_p$ and computes $T := e(G_1, \tilde{G}_0)^{r_{\mathtt{gid}}} e(G_2, \tilde{G}_0)^{r_z} e(G_2, \tilde{G}_1)^{r_v} e(R, \tilde{G}_0)^{-r_\delta}$. It puts the commitment message as $\textsc{com} := (R, T)$. It returns COM and its inner state St. $\mathtt{P}$ sends COM to $\mathtt{V}$. Note that the following equality holds after the re-randomization.

$$e(G_1, \tilde{G}_0)^{\mathtt{gid}} e(G_2, \tilde{G}_0)^{z} e(G_2, \tilde{G}_1)^{v} e(R, \tilde{G}_0)^{-\delta} = e(R, \tilde{G}_1)/e(G_0, \tilde{G}_0). \tag{76}$$

- $\Sigma^{\mathtt{CL}}_{\mathrm{cha}}(x) \to \textsc{cha}$. This PPT algorithm is executed by $\mathtt{V}$. 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. $\mathtt{V}$ sends CHA to $\mathtt{P}$.
- $\Sigma^{\mathtt{CL}}_{\mathrm{res}}(St, \textsc{cha}) \to \textsc{res}$. This PPT algorithm is executed by $\mathtt{P}$. On input the state St^a and the challenge message CHA, it computes $s_{\mathtt{gid}} := r_{\mathtt{gid}} + c\mathtt{gid}, s_z := r_z + cz, s_v := r_v + cv, s_\delta := r_\delta + c\delta$. It sets the response message as $\textsc{res} := (s_{\mathtt{gid}}, s_z, s_v, s_\delta)$. It returns RES. $\mathtt{P}$ sends RES to $\mathtt{V}$.
- $\Sigma^{\mathtt{CL}}_{\mathrm{vrf}}(x, \textsc{com}, \textsc{cha}, \textsc{res}) \to d$. This deterministic algorithm is executed by $\mathtt{V}$. 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").

$$e(G_1, \tilde{G}_0)^{s_{\mathtt{gid}}} e(G_2, \tilde{G}_0)^{s_z} e(G_2, \tilde{G}_1)^{s_v} e(R, \tilde{G}_0)^{-s_\delta} =_? T(e(R, \tilde{G}_1)/e(G_0, \tilde{G}_0))^c. \tag{77}$$

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

$\Pi^{\mathtt{CL}}$ 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 CHA deviates from the uniform random distribution.

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

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

- $\mathtt{Cmt}^{\mathtt{Ped}}.\mathtt{Setup}(1^\lambda) \to \mathtt{PP}$. On input the security parameter 1^λ, this PPT algorithm generates a set of public parameter values. That is, it runs 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 $\mathtt{PP} := (p, \mathbb{G}, G, H)$.
- $\mathtt{Cmt.Com}^{\mathtt{Ped}}(\mathtt{PP}, m) \to (C, \kappa)$. On input $\mathtt{PP}$ and 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).
- $\mathtt{Cmt.Vrf}^{\mathtt{Ped}}(\mathtt{PP}, C, m, \kappa) \to d$. On input $\mathtt{PP}$, 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, $d := 0$.

$\mathtt{Cmt}^{\mathtt{Ped}}$ is *perfectly hiding.* the distribution of the commitment C is independent of the committed message m. $\mathtt{Cmt}^{\mathtt{Ped}}$ 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 $\Pi^{\mathtt{Oka}} = (\Pi^{\mathtt{Oka}}.\mathtt{Setup}, \mathtt{P}, \mathtt{V})$ [Oka92] is executed in accordance with a Σ-protocol $\Sigma^{\mathtt{Oka}} = (\Sigma^{\mathtt{Oka}}_{\mathrm{com}}, \Sigma^{\mathtt{Oka}}_{\mathrm{cha}}, \Sigma^{\mathtt{Oka}}_{\mathrm{res}}, \Sigma^{\mathtt{Oka}}_{\mathrm{vrf}}, \Sigma^{\mathtt{Oka}}_{\mathrm{ext}}, \Sigma^{\mathtt{Oka}}_{\mathrm{sim}})$.

The setup algorithm $\Pi^{\mathtt{Oka}}.\mathtt{Setup}$ is the same as $\mathtt{Cmt}^{\mathtt{Ped}}.\mathtt{Setup}(1^\lambda)$. The set of public parameter values $\mathtt{PP}$ 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 $\mathtt{PP}$ and x.

The protocol between $\mathtt{P}$ and $\mathtt{V}$ is a Σ-protocol. It goes as follows.

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

- $\Sigma^{\mathtt{Oka}}_{\mathrm{res}}(St, \textsc{cha}) \to \textsc{res}$. This PPT algorithm is executed by $\mathtt{P}$. On input the state St^a and the challenge message $\textsc{cha}$, it computes $s_t := r_t + ct, s_u := r_u + cu$. It sets the response message as $\textsc{res} := (s_t, s_u)$. $\mathtt{P}$ sends $\textsc{res}$ to $\mathtt{V}$.
- $\Sigma^{\mathtt{Oka}}_{\mathrm{vrf}}(x, \textsc{com}, \textsc{cha}, \textsc{res}) \to d$. This deterministic algorithm is executed by $\mathtt{V}$. On input the statement x and all the messages $(\textsc{com}, \textsc{cha}, \textsc{res})$, it checks whether the following equality holds: $G^{s_t} H^{s_u} =_? AX^c$.

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

$\Pi^{\mathtt{Oka}}$ 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 $\textsc{cha}$ deviates from the uniform random distribution.

Combining the Pedersen commitment scheme $\mathtt{Cmt}^{\mathtt{Ped}}$ and the Okamoto interactive argument system $\Pi^{\mathtt{Oka}}$ with the Σ-protocol $\Sigma^{\mathtt{Oka}}$, we obtain the Pedersen-Okamoto commit-and-prove scheme $\mathtt{CmtPrv}^{\mathtt{PO}} = (\mathtt{CmtPrv.Setup}, \mathtt{Cmt}^{\mathtt{Ped}} = (\mathtt{Cmt.Com}^{\mathtt{Ped}}, \mathtt{Cmt.Vrf}^{\mathtt{Ped}}), \Pi^{\mathtt{Oka}} = (\mathtt{P}, \mathtt{V}))$.