The preliminary version of this paper appeared in the proceedings of *"The 3rd IEEE International Workshop on Big Data and IoT Security in Smart Computing - IEEE BITS 2019 -"* (co-held with *"IEEE SMARTCOMP 2019"*) under the same title. This is the full version.

Decentralized Multi-authority Anonymous Authentication for Global Identities with Non-interactive Proofs

Hiroaki Anada

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

July 6, 2022

Abstract. A decentralized multi-authority anonymous authentication scheme that is suitable for IoT and blockchains is proposed, in which a prover and a verifier are non-interactive. The proposed scheme can treat dynamically increasing/decreasing independent attribute authorities. When an entity wants the authorities to issue attribute credentials, the authorities only have to generate digital signatures on her global identity. Two security definitions are given; resistance against eavesdrop-and-collude attacks that cause misauthentication, and anonymity for privacy protection. Then a construction of our scheme is described under a principle of "commit-to-ID" to attain resistance against the collusion attacks. There are two building blocks; the structure-preserving signature scheme and the Groth-Sahai non-interactive proof system, the both of which are in the setting of bilinear groups. The proposed scheme is proved to be secure in the standard model.

Keywords: decentralized authorities, anonymous authentication, collusion resistance, non-interactive, blockchain

1 Introduction

Email addresses that are issued by a reliable organization can be identification data on the internet. Also, universally unique identifiers (UUID) that are in accordance with the standard ISO/IEC 11578:1996 are global identifiers for devices having MAC addresses. Those data can be called global identities on the internet of things (IoT). In the trend that human beings, machines and computer programs are connected to IoT with the global identities, each entity is likely to have plural *attribute credentials* issued by authorities that are also on IoT. Due to the trend, there arises possibility that some combinations of those attribute credentials enable the entity to enjoy services of smarter strategies. The reason why the possibility is expected to be realized is that those strategies are basically not only for individuals but also for ecology and economy towards global optimization. Note here that the global identities would be actually useful because possibly independent flat authorities which we call *decentralized multi-authorities* can refer the identity data.

On the other hand, privacy protection is rapidly growing demand on IoT because most of entities' activities on IoT will not need identity information. Suppose that a person buys some goods in a convenience store using bitcoins [1] with her smartphone. In the case the record should not be linked to the UUID or the phone number; only the data of paying bitcoins and having the right should be sent to the server of the store. Here the data will contain transaction data of bitcoins and a proof of having attribute credentials that are issued legitimately.

Thus, we need anonymous attribute authentication scheme on IoT where decentralized multi-authorities issue attribute credentials to entities on IoT. Then, in the scheme, when an entity wants to prove the possession of the attribute credentials to a verifier, it generates a proof of the credentials in a zero-knowledge or witness-indistinguishable way [2]. However, one technical obstacle in realizing such a scheme is *collusion attacks*. Adversaries having different identities might bring together their credentials of different attributes. Then they send a proof of those merged credentials to a verifier and try to get accepted.

1.1 Our Contribution and Related Work

In this paper, we propose a decentralized multi-authority anonymous authentication scheme a-auth that is secure against the collusion attacks, and in which a prover and a verifier are non-interactive. Our a-auth scheme can treat dynamically increasing/decreasing decentralized multi-authorities of attributes. Here the idea to attain collusion resistance is to "commit-to-ID". Intuitively, when a prover having attribute credentials for her identity generates a proof, she first generates a commitment to her identity string. The commitment works as a confirmation that the owner of the plural credentials is certainly a single person (see the technical explanation below for detail). Another feature of our a-auth is that, when a prover wants the authorities to issue private secret keys as attribute credentials, the authorities only have to generate digital signatures on her identity. The feature is useful when her identity data are easy to be validated by the authorities in the registration phase, which is actually the case for a legitimately issued global identity.

Camenisch et al. [3, 4] proposed a scheme which is similar to our a-auth in its construction. One of the strong points of their scheme is its universal composability with other cryptographic primitives [3]. However, the case of decentralized multi-authorities and the property of collusion resistance were not discussed in [3, 4], which is our prime target in this paper. Anada-Arita [5, 6] shares the above features of our a-auth. However, in their authentication scheme a prover and a verifier is interactive, while our a-auth is non-interactive. As for performance, we note that our a-auth does not have a mechanism of accumulators (see, for example, [7]). Therefore, it is a drawback of our a-auth that the length of a proof grows linearly to number of attributes which a prover wants to prove. Also, the computational overhead of a verifier grows linearly to the number of attributes.

1.2 Remark on Replay Attack and Application to Blockchain

In compensation of the merit of non-interactiveness, naive use of our a-auth is vulnerable to the *replay attack* [8]. That is, if an adversary captures a proof of the credentials which a prover sends to a verifier on the internet, then the adversary possibly forces the verifier accept him by sending the same proof. (The attack works even when the communication between the prover and the verifier is encrypted.) Since a proof in our a-auth is generated with randomness, we can avoid the attack by making verifiers stateful to detect replays. But we cannot avoid the attack if the adversary sends the same proof to *other* verifiers.

Recently, the replay attack has been studied with use of a blockchain [1]. For example, IBM [9] proposed a mechanism to avoid the replay attack on a permissioned and privacy-preserving blockchain network. Since our a-auth scheme contains decentralized multi-authorities and anonymous provers, the mechanism fits to our a-auth. Roughly speaking, a hash value of a proof in our a-auth is took in to the related transaction, and the transaction is involved in a blockchain. Then the verifiers in our a-auth are able to detect replay attacks by examining the blockchain. Development in this direction is still under construction in the research community (for example, [10]). Thus, we hope that our decentralized multi-authority anonymous authentication scheme a-auth serves as a privacy-protecting authentication framework which is non-interactive and which has resistance against the replay attack on future blockchain networks.

1.3 Overview of Our Technical Construction and Security Proofs

After giving the syntax of our a-auth scheme, we give two security definitions in Section 3. One is resistance against eavesdrop-and-collude attacks that cause misauthentication, and the other is anonymity for privacy protection. Then a construction of our a-auth is described. There are two building blocks; the structure-preserving signature scheme [11, 12] and the Groth-Sahai non-interactive proof system [2, 13], the both of which are in the setting of bilinear groups. (We note that other types of the structure-preserving signatures such as [14] can be employed instead of [11, 12].) In Section 4, security proofs for the above construction are given in the standard model. The resistance against eavesdrop-and-collude is due to knowledge extraction property of the Groth-Sahai proof system and existential unforgeability of the structure-preserving signature scheme. Besides, our design principle "commit-to-ID" works to exclude the collusion attacks because a commitment to an identity string cannot be opened in two ways. The anonymity is due to perfectly hiding property of commitments and perfectly witness indistinguishable property of proofs generated in the Groth-Sahai proof system, where the both properties hold in the simulation mode of the dual mode commitment.

2 Preliminaries

We survey here the building blocks of our a-auth scheme. $\mathbb{N}$ and $\mathbb{Z}$ means the set of natural numbers and integers, respectively. p means a prime number. $\mathbb{Z}_p$ means the ring $\mathbb{Z}/p\mathbb{Z}$. λ means the security parameter, where $\lambda \in \mathbb{N}$. A probability P is said to be negligible in λ if for any given positive polynomial poly(λ) $P < 1/\mathrm{poly}(\lambda)$ for sufficiently large $\lambda \in \mathbb{N}$. Two probabilities P and Q are said to be computationally indistinguishable if $|P - Q|$ is negligible in λ, which is denoted as $P \approx_{\mathrm{c}} Q$. A uniform random sampling of an element a from a set S is denoted as $a \in_R S$. When a probabilistic algorithm A with an input a and a randomness r on a random tape returns z, we denote it as $z \leftarrow A(a; r)$. St is the inner state of the concerned algorithm.

2.1 Bilinear Groups [15, 13]

Let p be a prime number of bit-length λ and $\hat{\mathbb{G}}, \check{\mathbb{H}}$ and $\mathbb{T}$ be cyclic groups of order p, $\hat{G}$ and $\check{H}$ be generators of $\hat{\mathbb{G}}$ and $\check{\mathbb{H}}$, respectively. We denote operations in the groups multiplicatively. Let e be a map $e : \hat{\mathbb{G}} \times \check{\mathbb{H}} \to \mathbb{T}$ with:

- Non-degeneracy : $e(\hat{G}, \check{H}) \neq 1_{\mathbb{T}}$,
- Bilinearity : $e(\hat{X}^a, \check{Y}^a) = e(\hat{X}, \check{Y})^{ab}$ for $a, b \in \mathbb{Z}_p, \hat{X} \in \hat{\mathbb{G}}, \check{Y} \in \check{\mathbb{H}}$.

Let $\mathcal{G}$ be a bilinear-groups generation algorithm: $\mathcal{G}(1^\lambda) \to (p, \hat{\mathbb{G}}, \check{\mathbb{H}}, \mathbb{T}, e, \hat{G}, \check{H})$. $\hat{\mathbb{G}}$ and $\check{\mathbb{H}}$ are called source groups and $\mathbb{T}$ is called a target group. We denote an element in $\hat{\mathbb{G}}$ and $\check{\mathbb{H}}$ with hat ' ˆ ' and check ' ˇ ' , respectively.

2.2 Structure-Preserving Signature Scheme [11, 12]

The structure-preserving signature scheme Sig is a digital signature scheme which is based on bilinear groups. A message is a vector with its entries being in $\hat{\mathbb{G}}$ and $\check{\mathbb{H}}$. A signature is a vector with its entries being in $\hat{\mathbb{G}}$ and $\check{\mathbb{H}}$. Sig consists of four PPT algorithms (Sig.Setup, Sig.KG$_{pp}$, Sig.Sign$_{pp}$, Sig.Vrf$_{pp}$). The description below is in accordance with [12].
Sig.Setup(1^λ) $\to pp$. On input the security parameter 1^λ, this PPT algorithm executes the bilinear-groups generation algorithm. It puts the output as a set of public parameters: $\mathcal{G}(1^\lambda) \to (p, \hat{\mathbb{G}}, \check{\mathbb{H}}, \mathbb{T}, e, \hat{G}, \check{H}) =: pp$. It returns pp.
Sig.KG$_{pp}$() $\to$ (PK, SK). Based on the set of public parameters pp, this PPT algorithm generates a signing key SK and the corresponding public key PK as follows: $\hat{G}_u \in_R \hat{\mathbb{G}}$, $\gamma_1, \delta_1 \in_R \mathbb{Z}_p^*$, $\hat{G}_1 := \hat{G}^{\gamma_1}, \hat{G}_{u,1} := \hat{G}_u^{\delta_1}$. $\gamma_z, \delta_z \in_R \mathbb{Z}_p^*$, $\hat{G}_z := \hat{G}^{\gamma_z}, \hat{G}_{u,z} := \hat{G}_u^{\delta_z}$. $\alpha, \beta \in_R \mathbb{Z}_p^*$, $(\hat{A}_i, \check{A}_i)_{i=0}^1 \leftarrow \mathsf{Extend}(\hat{G}, \check{H}^\alpha)$, $(\hat{B}_i, \check{B}_i)_{i=0}^1 \leftarrow \mathsf{Extend}(\hat{G}_u, \check{H}^\beta)$ (for the description of the algorithm Extend, see [12]). It puts PK $:= (\hat{G}_z, \hat{G}_{u,z}, \hat{G}_u, \hat{G}_1, \hat{G}_{u,1}, (\hat{A}_i, \check{A}_i, \hat{B}_i, \check{B}_i)_{i=0}^1)$ and SK $:= (\alpha, \beta, \gamma_z, \delta_z, \gamma_1, \delta_1)$. It returns (PK, SK).

$\mathsf{Sig.Sign}_{pp}(\mathrm{PK}, \mathrm{SK}, m) \to \sigma$. On input the public key PK, the secret key SK and a message $m = \check{M} \in \check{\mathbb{H}}$, this PPT algorithm generates a signature σ as follows.

$$\begin{aligned} &\zeta, \rho, \tau, \phi, \omega \in_R \mathbb{Z}_p, \\ &\check{Z} := \check{H}^{\zeta}, \check{R} := \check{H}^{\alpha-\rho\tau-\gamma_z\zeta}\check{M}^{-\gamma_1}, \hat{S} := \hat{G}^{\rho}, \check{T} := \check{H}^{\tau}, \\ &\qquad \check{U} := \check{H}^{\beta-\phi\omega-\delta_z\zeta}\check{M}^{-\delta_1}, \hat{V} := \hat{G}_u^{\phi}, \check{W} := \check{H}^{\omega}. \end{aligned}$$

It returns $\sigma := (\check{Z}, \check{R}, \hat{S}, \check{T}, \check{U}, \hat{V}, \check{W})$.
$\mathsf{Sig.Vrf}_{pp}(\mathrm{PK}, m, \sigma) \to d$. On input the public key PK, a message $m = \check{M} \in \check{\mathbb{H}}$ and a signature $\sigma = (\check{Z}, \check{R}, \hat{S}, \check{T}, \check{U}, \hat{V}, \check{W})$, this deterministic algorithm checks whether the following verification equations hold or not.

$$e(\hat{G}_z, \check{Z})e(\hat{G}, \check{R})e(\hat{S}, \check{T})e(\hat{G}_1, \check{M})e(\hat{A}_0, \check{A}_0)^{-1}e(\hat{A}_1, \check{A}_1)^{-1} = 1_{\mathbb{T}}, \text{ and} \tag{1}$$

$$e(\hat{G}_{u,z}, \check{Z})e(\hat{G}_u, \check{U})e(\hat{V}, \check{W})e(\hat{G}_{u,1}, \check{M})e(\hat{B}_0, \check{B}_0)^{-1}e(\hat{B}_1, \check{B}_1)^{-1} = 1_{\mathbb{T}}. \tag{2}$$

It returns a boolean decision d.

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

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

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

In the experiment, $\mathbf{F}$ issues a signing query to its signing oracle $\mathbf{SignO}_{pp}(\mathrm{PK}, \mathrm{SK}, \cdot)$ by sending a message m_j at most q_{s} times ($1 \le j \le q_{\mathrm{s}}$). As a reply, $\mathbf{F}$ receives a valid signature σ_j on m_j. After receiving replies, $\mathbf{F}$ returns a pair of a message and a signature (m^*, σ^*). A restriction is imposed on the algorithm $\mathbf{F}$: The set of queried messages $\{m_j\}_{1 \le j \le q_{\mathrm{s}}}$ should not contain the message m^*. The advantage of $\mathbf{F}$ over Sig is defined as $\mathbf{Adv}^{\text{euf-cma}}_{\mathsf{Sig},\mathbf{F}}(\lambda) := \Pr[\mathsf{Exp}^{\text{euf-cma}}_{\mathsf{Sig},\mathbf{F}}(1^{\lambda}) \text{ returns } \textsc{Win}]$. The scheme Sig is said to be *existentially unforgeable against adaptive chosen-message attacks (EUF-CMA)* if for any PPT algorithm $\mathbf{F}$ and any q_{s} bounded by a polynomial in λ, the advantage $\mathbf{Adv}^{\text{euf-cma}}_{\mathsf{Sig},\mathbf{F}}(\lambda)$ is negligible in λ. The structure-preserving signature scheme [11, 12] is known to be EUF-CMA under the q-SFP assumption [12].

3 Decentralized Multi-authority Anonymous Authentication with Non-interactive Proofs

In this section, we give a syntax and security definitions of our decentralized multi-authority non-interactive anonymous authentication scheme $\mathsf{a\text{-}auth}$. Then we introduce two security definitions. One is resistance against eavesdrop-and-collude attacks that cause misauthentication. The other is anonymity for privacy protection.

3.1 Syntax

Our $\mathsf{a\text{-}auth}$ consists of five PPT algorithms, (Setup, AuthKG_{pp}, PrivKG_{pp}, Prover_{pp}, $\mathsf{Verifier}_{pp}$).
• $\mathsf{Setup}(1^{\lambda}) \to pp$. This PPT algorithm is needed to generate a set of public parameters pp. On input the security parameter 1^{λ}, it generates the set pp. It returns pp.

• $\mathsf{AuthKG}_{pp}(a) \rightarrow (\mathrm{PK}^a, \mathrm{MSK}^a)$. This PPT algorithm is executed by a key-issuing authority indexed by a. On input 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 $(\mathrm{PK}^a, \mathrm{MSK}^a)$.
• $\mathsf{PrivKG}_{pp}(\mathrm{PK}^a, \mathrm{MSK}^a, \mathtt{i}) \rightarrow \mathrm{sk}^a_{\mathtt{i}}$. This PPT algorithm is executed by the a-th key-issuing authority. On input the a-th public and master secret keys $(\mathrm{PK}^a, \mathrm{MSK}^a)$ and an element $\mathtt{i} \in \check{\mathbb{H}}$ (that is an identifier of a prover), it generates a private secret key $\mathrm{sk}^a_{\mathtt{i}}$ of a prover. It returns $\mathrm{sk}^a_{\mathtt{i}}$.
• $\mathsf{Prover}_{pp}((\mathrm{PK}^a, \mathrm{sk}^a_{\mathtt{i}})^{a \in A'}) \rightarrow \pi$. This PPT algorithm is executed by a prover who is to be authenticated, where A' denotes a subset of the set A of all the authority indices. On input the public keys $(\mathrm{PK}^a)^{a \in A'}$ and the corresponding private secret keys $(\mathrm{sk}^a_{\mathtt{i}})^{a \in A'}$, it returns a proof π.
• $\mathsf{Verifier}_{pp}((\mathrm{PK}^a)^{a \in A'}, \pi) \rightarrow d$. This deterministic polynomial-time algorithm is executed by a verifier who confirms that the prover certainly knows the secret keys for indices $a \in A'$. On input the public keys $(\mathrm{PK}^a)^{a \in A'}$ and the proof π, it returns $d := 1$ ("accept") or $d := 0$ ("reject").

3.2 Security Definitions

Resistance against Eavesdrop-and-Collude Attack of Misauthentication We define an algorithm of experiment of eavesdrop-and-collude attack on a-auth and an adversary algorithm $\mathbf{A}$, as follows.

```
$\mathsf{Exp}^{\text{eaves-coll}}_{\mathsf{a\text{-}auth},\mathbf{A}}(1^\lambda)$ :
  $pp \leftarrow \mathsf{Setup}(1^\lambda), A := \{1, \ldots, \mu\}$
  For $a \in A : (\mathrm{PK}^a, \mathrm{MSK}^a) \leftarrow \mathsf{AuthKG}_{pp}(a)$
  $(q_I, St) \leftarrow \mathbf{A}(pp, (\mathrm{PK}^a)^{a \in A}), I := \{1, \ldots, q_I\}$
  For $i \in I : \mathtt{i}_i \in_R \check{\mathbb{H}}$
    For $a \in A : \mathrm{sk}^a_{\mathtt{i}_i} \leftarrow \mathsf{PrivKG}_{pp}(\mathrm{PK}^a, \mathrm{MSK}^a, \mathtt{i}_i)$
  $(\tilde{A}, St) \leftarrow \mathbf{A}(St), \bar{\tilde{A}} := A \backslash \tilde{A}$
  $(\pi^*, A^*) \leftarrow \mathbf{A}^{\mathsf{Prover}_{pp}((\mathrm{PK}^a, \mathrm{sk}^a_{\mathtt{i}_i})^{a \in \bar{\tilde{A}}})|_{i \in I}, \mathbf{PrivKGO}_{pp}(\mathrm{PK}^\cdot, \mathrm{MSK}^\cdot, \cdot)}(St, (\mathrm{MSK}^a)^{a \in \tilde{A}})$
  $\mathsf{Verifier}_{pp}((\mathrm{PK}^a)^{a \in A^*}, \pi^*) \rightarrow d$
  If $d = 1$ then return WIN else return LOSE
```

Intuitively, the attack described in the above experiment has the following meaning. On input the set of public parameters pp and the public keys $(\mathrm{PK}^a)^{a \in A}$, $\mathbf{A}$ outputs the number q_I of provers. Then $\mathbf{A}$ outputs a set of indices of corrupted authorities $\tilde{A}$. $\mathbf{A}$ eavesdrops and intercepts proofs from provers Prover_{pp} with $\mathtt{i}_i, i = 1, \ldots, q_I$ and $a \in \bar{\tilde{A}} := A \backslash \tilde{A}$. In addition, $\mathbf{A}$ collects at most q_{sk} private secret keys by issuing queries to the private secret key oracle $\mathbf{PrivKGO}_{pp}(\mathrm{PK}^\cdot, \mathrm{MSK}^\cdot, \cdot)$ with an authority index $a \in \bar{\tilde{A}}$ and an identifier element $\mathtt{i}_j \in \check{\mathbb{H}}$ for $j \in J := \{q_I + 1, \ldots, q_I + q_{\mathrm{sk}}\}$. We denote by A_j the set of authority indices for which the private secret key queries were issued with $\mathtt{i}_j$:

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

Note that the maximum number of private secret key queries is μq_{sk}. We require that the numbers μ, q_I and q_{sk} are bounded by a polynomial in λ. At the end $\mathbf{A}$ returns a forgery proof π^* together with the target set of authority indices A^* that is a subset of $\bar{\tilde{A}}$:

$$A^* \subset \bar{\tilde{A}}. \tag{3}$$

If the decision d on π^* by $\mathsf{Verifier}_{pp}$ is 1 under $(\mathrm{PK}^a)^{a \in A^*}$, then the experiment returns WIN; otherwise it returns LOSE.

Two types of restrictions are imposed on the adversary $\mathbf{A}$. One type is that mere *replay* of proofs is ruled out[1]. The other type is that the queried $\mathtt{i}_j$s are pairwise different, and any A_j is a proper subset of the

[1] As is mentioned in Section 1, detecting replay attacks is currently studied by researchers and developers [9, 10].

target set A^*:

$$\mathtt{i}_{j_1} \neq \mathtt{i}_{j_2} \text{ for } j_1, j_2 \in J, j_1 \neq j_2, \tag{4}$$

$$A_j \subsetneq A^*,\ j \in J. \tag{5}$$

These restrictions are because, otherwise, $\mathbf{A}$ can trivially succeed in causing misauthentication.

The advantage of an adversary $\mathbf{A}$ over an anonymous authentication scheme $\mathsf{a\text{-}auth}$ in the experiment is defined as: $\mathbf{Adv}^{\text{eaves-coll}}_{\mathsf{a\text{-}auth},\mathbf{A}}(\lambda) \overset{\text{def}}{=} \Pr[\mathsf{Exp}^{\text{eaves-coll}}_{\mathsf{a\text{-}auth},\mathbf{A}}(1^\lambda) = \textsc{Win}]$. A scheme $\mathsf{a\text{-}auth}$ is called secure against eavesdrop-and-collude attacks that cause misauthentication. if, for any PPT algorithm $\mathbf{A}$, the advantage $\mathbf{Adv}^{\text{eaves-coll}}_{\mathsf{a\text{-}auth},\mathbf{A}}(\lambda)$ is negligible in λ.

Anonymity We define an algorithm of experiment of anonymity game on $\mathsf{a\text{-}auth}$ and an adversary algorithm $\mathbf{A}$, as follows.

```
$\mathsf{Exp}^{\text{ano}}_{\mathsf{a\text{-}auth},\mathbf{A}}(1^\lambda)$ :
  $pp \leftarrow \mathsf{Setup}(1^\lambda), A := \{1, \ldots, \mu\}$
  For $a \in A : (\mathrm{PK}^a, \mathrm{MSK}^a) \leftarrow \mathsf{AuthKG}_{pp}(a)$
  $(\mathtt{i}_0, \mathtt{i}_1, St) \leftarrow \mathbf{A}(pp, (\mathrm{PK}^a)^{a \in A})$
  For $a \in A$ : For $i = 0, 1 : \mathrm{sk}^a_{\mathtt{i}_i} \leftarrow \mathsf{PrivKG}_{pp}(\mathrm{PK}^a, \mathrm{MSK}^a, \mathtt{i}_i)$
  $b \in_R \{0,1\}, b' \leftarrow \mathbf{A}^{\mathsf{Prover}_{pp}((\mathrm{PK}^a, \mathrm{sk}^a_{\mathtt{i}_b})^{a \in A})}(St, (\mathrm{sk}^a_{\mathtt{i}_0}, \mathrm{sk}^a_{\mathtt{i}_1})^{a \in A})$
  If $b = b'$, then return $\textsc{Win}$, else return $\textsc{Lose}$
```

Intuitively, the game described in the above experiment has the following meaning. On input the set of public parameters pp and the issued public keys $(\mathrm{PK}^a)^{a \in A}$, $\mathbf{A}$ designates two identity elements $\mathtt{i}_0$ and $\mathtt{i}_1$, and $\mathbf{A}$ is given private secret keys $(\mathrm{sk}^a_{\mathtt{i}_0}, \mathrm{sk}^a_{\mathtt{i}_1})$ for all $a \in A$. Next, for randomly chosen b that is hidden from $\mathbf{A}$, $\mathbf{A}$ does oracle-access to a prover Prover_{pp} that is with input the private secret keys $(\mathrm{sk}^a_{\mathtt{i}_b})^{a \in A}$. If the decision b' of $\mathbf{A}$ is equal to b, then the experiment returns $\textsc{Win}$; otherwise it returns $\textsc{Lose}$.

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

4 Construction and Security Proofs

In this section, we give a construction of an $\mathsf{a\text{-}auth}$ scheme. Each decentralized authority indexed by 'a' issues a private secret key $\mathrm{sk}^a_{\mathtt{i}}$ for an identity element $\mathtt{i}$ by generating a structure-preserving signature on $\mathtt{i}$. Next, in the committing-phase a prover generates commitments to the identity element $\mathtt{i}$ and the components of the structure-preserving signatures $(\sigma^a)^{a \in A'} = (\sigma^a_1, \ldots, \sigma^a_7)^{a \in A'}$. In the proving-phase the prover generates a proof π of the "bundled" witness [5]. Here the bundled witness means that the identity element $\mathtt{i} = \check{M}$ is common for all $a \in A'$, and, for each $a \in A'$ $\mathtt{i}$ and the elements $(\sigma^a_1, \ldots, \sigma^a_7)$ satisfy the verification equation system "(1) and (2)".

4.1 Construction

The scheme $\mathsf{a\text{-}auth} = (\mathsf{Setup}, \mathsf{AuthKG}_{pp}, \mathsf{PrivKG}_{pp}, \mathsf{Prover}_{pp}, \mathsf{Verifier}_{pp})$ is described as follows.

• $\mathsf{Setup}(1^\lambda) \rightarrow pp$. On input the security parameter 1^λ, this PPT algorithm runs the bilinear-groups generation algorithm. It puts the output as a set of public parameters: $\mathcal{G}(1^\lambda) \rightarrow (p, \hat{\mathbb{G}}, \check{\mathbb{H}}, \mathbb{T}, e, \hat{G}, \check{H}) =: pp$. Note that pp is common for both the structure-preserving signature scheme Sig and the commit-and-prove scheme CmtPrv. Besides, it runs the generation algorithm of commitment key: $\mathsf{Cmt.KG}_{pp}(\mathtt{nor}) \rightarrow ck$. It returns $pp := (pp, ck)$.

• $\mathsf{AuthKG}_{pp}(a) \rightarrow (\mathrm{PK}^a, \mathrm{MSK}^a)$. On input an authority index a, this PPT algorithm executes the key-generation algorithm $\mathsf{Sig.KG}_{pp}()$ to obtain $(\mathrm{PK}, \mathrm{SK})$. It puts $\mathrm{PK}^a := \mathrm{PK}$ and $\mathrm{MSK}^a := \mathrm{SK}$. It returns $(\mathrm{PK}^a, \mathrm{MSK}^a)$.

• $\mathsf{PrivKG}_{pp}(\mathrm{PK}^a, \mathrm{MSK}^a, \mathtt{i}) \to \mathrm{sk}^a_{\mathtt{i}}$. On input PK^a, MSK^a and an element $\mathtt{i} \in \check{\mathbb{H}}$, this PPT algorithm puts $\mathrm{PK}^a := \mathrm{PK}^a$ and $\mathrm{SK}^a := \mathrm{MSK}^a$ and $m = \check{M} := \mathtt{i}$. It executes the signing algorithm $\mathsf{Sig.Sign}_{pp}(\mathrm{PK}^a, \mathrm{SK}^a, m)$ to obtain a signature σ^a. It puts $\mathrm{sk}^a_{\mathtt{i}} := (\mathtt{i}, \sigma^a)$. It returns $\mathrm{sk}^a_{\mathtt{i}}$.
• $\mathsf{Prover}_{pp}((\mathrm{PK}^a, \mathrm{sk}^a_{\mathtt{i}})^{a \in A'}) \to \pi$. On input $(\mathrm{PK}^a, \mathrm{sk}^a_{\mathtt{i}})^{a \in A'}$, first, this PPT algorithm commits to $\mathtt{i}$:

$$c_0 \leftarrow \mathsf{Cmt.Com}_{pp}(\mathtt{i}; r_0).$$

Second, for each $a \in A'$, it commits to the components $\sigma^a_1, \ldots, \sigma^a_7$ of the signature σ^a in the componentwise way.

$$(c^a_1, \ldots, c^a_7) \leftarrow \mathsf{Cmt.Com}_{pp}((\sigma^a_1, \ldots, \sigma^a_7); (r^a_1, \ldots, r^a_7)).$$

Then, for each $a \in A'$, it puts $x^a := (\hat{G}^a_z, \hat{G}^a_{u,z}, \hat{G}^a_u, \hat{G}^a_1, \hat{G}^a_{u,1}, (\hat{A}^a_i, (\check{A}^a_i)^{-1}, \hat{B}^a_i, (\check{B}^a_i)^{-1})^1_{i=0})$ by using PK^a. It also puts $c^a := (c_0, c^a_1, \ldots, c^a_7)$, $w^a := (w_0, w^a_1, \ldots, w^a_7) := (\mathtt{i}, \sigma^a_1, \ldots, \sigma^a_7)$ and $r^a := (r_0, r^a_1, \ldots, r^a_7)$. It executes the proof generation algorithm P_{pp} to obtain a proof:

$$\pi^a \leftarrow \mathsf{P}_{pp}(x^a, c^a, w^a, r^a), a \in A'.$$

It puts $\bar{\pi}^a := ((c^a_1, \ldots, c^a_7), \pi^a)$ for each $a \in A'$, and it merges all the $\bar{\pi}^a$s and the single commitment c_0 into a proof π. That is, $\pi := (c_0, (\bar{\pi}^a)^{a \in A'})$. It returns π.
• $\mathsf{Verifier}_{pp}((\mathrm{PK}^a)^{a \in A'}, \pi) \to d$. On input $((\mathrm{PK}^a)^{a \in A'}, \pi)$, this deterministic polynomial-time algorithm converts PK^a into x^a and it puts $c^a := (c_0, c^a_1, \ldots, c^a_7)$ for each $a \in A'$. Then it executes the verification algorithm V_{pp} for each $a \in A'$ to obtain the decisions:

$$d^a \leftarrow \mathsf{V}_{pp}(x^a, c^a, \pi^a), a \in A'.$$

If all the decisions d^as are 1, then it returns $d := 1$; otherwise it returns $d := 0$.

4.2 Security Proofs

Theorem 1 (Security against Eavesdrop-and-Collude Attacks) *For any* PPT *algorithm* $\mathbf{A}$ *that is in accordance with the experiment* $\mathsf{Exp}^{\textit{eaves-coll}}_{\textit{a-auth},\mathbf{A}}(1^\lambda)$*, there exists a* PPT *algorithm* $\mathbf{F}$ *that is in accordance with the experiment* $\mathsf{Exp}^{\textit{euf-cma}}_{\textit{Sig},\mathbf{F}}(1^\lambda)$ *and the following inequality holds.*

$$\mathbf{Adv}^{\textit{eaves-coll}}_{\textit{a-auth},\mathbf{A}}(\lambda) = \frac{p}{p-1} \cdot \mu \cdot \mathbf{Adv}^{\textit{euf-cma}}_{\textit{Sig},\mathbf{F}}(\lambda).$$

The meaning of this theorem is that, if the structure-preserving signature scheme Sig is EUF-CMA, then our $\mathsf{a\text{-}auth}$ is secure against eavesdrop-and-collude attacks.
Proof. Given any PPT algorithm $\mathbf{A}$ that is in accordance with the experiment $\mathsf{Exp}^{\text{eaves-coll}}_{\mathsf{a\text{-}auth},\mathbf{A}}(1^\lambda)$, we construct a PPT algorithm $\mathbf{F}$ that generates an existential forgery of Sig according to the experiment $\mathsf{Exp}^{\text{euf-cma}}_{\mathsf{Sig},\mathbf{F}}(1^\lambda)$. $\mathbf{F}$ is given as input the set of public parameters pp and a public key $\mathrm{PK}_{\mathsf{Sig}}$. $\mathbf{F}$ is also given an auxiliary input μ. $\mathbf{F}$ executes $\mathsf{Cmt.KG}_{pp}(\mathtt{ext})$ to obtain a pair (ck, xk). $\mathbf{F}$ puts $pp := (pp, ck)$. $\mathbf{F}$ chooses a *target index* $a^\dagger$ from the set $A := \{1, \ldots, \mu\}$ *uniformly at random*. $\mathbf{F}$ executes the authority key generation algorithm honestly for $a \in A$ *except* the target index $a^\dagger$. As for $a^\dagger$, $\mathbf{F}$ uses the input public key:

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

$\mathbf{F}$ inputs pp and the public keys $(\mathrm{PK}^a)^{a \in A}$ into $\mathbf{A}$ to obtain the number q_I. $\mathbf{F}$ sets I as $I := \{1, \ldots, q_I\}$. $\mathbf{F}$ inputs St into $\mathbf{A}$. Then $\mathbf{F}$ obtains a set of corrupted authority indices $\tilde{A}$ from $\mathbf{A}$. $\mathbf{F}$ puts $\bar{\tilde{A}} := A \backslash \tilde{A}$. If $a^\dagger \in \bar{\tilde{A}}$ (the case TgtIdx), then $a^\dagger$ is not in $\tilde{A}$ and $\mathbf{F}$ is able to input $(St, (\mathrm{MSK}^a)^{a \in \tilde{A}})$ into $\mathbf{A}$. Otherwise $\mathbf{F}$ aborts.
Simulation of Provers. When $\mathbf{A}$ tries to intercept proofs from q_I provers $\mathsf{Prover}_{pp}((\mathrm{PK}^a, \mathrm{sk}^a_{\mathtt{i}_i})^{a \in \bar{\tilde{A}}})|_{i \in I}$, $\mathbf{F}$ chooses $\mathtt{i}^\dagger \in_R \check{\mathbb{H}}$. $\mathbf{F}$ executes the private secret key generation algorithm with input $\mathtt{i}^\dagger$ honestly for $a \in \bar{\tilde{A}}$ where $a \neq a^\dagger$. As for $a = a^\dagger$, $\mathbf{F}$ issues a signing query to its oracle with $\mathtt{i}^\dagger$:

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

In the simulation of provers $\mathsf{Prover}_{pp}((\mathrm{PK}^a, \mathrm{sk}^a_{\mathtt{i}_i})^{a\in\bar{\bar{A}}})|_{i\in I}$, $\mathbf{F}$ uses the *single* private secret key $(\mathrm{sk}^a_{\mathtt{i}^\dagger})^{a\in\bar{\bar{A}}}$. This is perfect simulation due to the perfect witness-indistinguishability of the Groth-Sahai proof system (see Definition 8 in Appendix).

Simulation of Private Secret Key Oracle. When $\mathbf{A}$ issues a private secret key query with $a \in A_j \subsetneq \bar{\bar{A}}$ and $\mathtt{i}_j \in \mathbb{Z}_p (j \in J)$, $\mathbf{F}$ executes the private secret key generation algorithm with $\mathtt{i}_j$ honestly for $a \in \bar{\bar{A}}$ such that $a \neq a^\dagger$. As for $a = a^\dagger$, $\mathbf{F}$ issues a signing query to its oracle with $\mathtt{i}_j$:

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

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

At the end $\mathbf{A}$ returns a forgery proof and the target set of authority indices (π^*, A^*). Note that $A^* \subset \bar{\bar{A}}$ as in the definition (3).

Generating Existential Forgery. Next, $\mathbf{F}$ runs a verifier $\mathsf{Verifier}_{pp}$ with an input $((\mathrm{PK}^a)^{a\in A^*}, \pi^*)$. If the decision d of $\mathsf{Verifier}_{pp}$ is 1, then $\mathbf{F}$ executes for each $a \in A^*$ the extraction algorithm $\mathsf{Cmt.Ext}_{pp}(xk, c^a)$ to obtain a committed message $(w^a)^* = ((w^a_0)^*, ((w^a_k)^*)_k)$ (see Definition 7 in Appendix). Note here that, for all $a \in A^*$, $(w^a_0)^*$ is equal to a single element $(w_0)^*$ in $\check{\mathbb{H}}$. This is because of the *perfectly binding property* of Cmt_{pp}. Then $\mathbf{F}$ puts $\mathtt{i}^* := (w_0)^*$. Here the restriction (4)(5) assures that, if $q_{\mathrm{sk}} > 0$, then there exists at least one $\hat{a} \in A^* \backslash A_j$ for some $j \in J$. If $q_{\mathrm{sk}} = 0$, then there exists at least one $\hat{a} \in A^*$. $\mathbf{F}$ chooses such an $\hat{a}$ and puts $\sigma^* := (\sigma^{\hat{a}})^* := ((w^{\hat{a}}_k)^*)_k$. $\mathbf{F}$ returns a forgery pair of a message and a signature $(\mathtt{i}^*, \sigma^*)$. This completes the description of $\mathbf{F}$.

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

$$\begin{aligned}
\textsc{Acc} &: d = 1,\\
\textsc{Ext} &: \mathsf{Cmt.Ext}_{pp} \text{ returns a witness } (w^a)^*,\\
\textsc{TgtIdx} &: \hat{a} = a^\dagger,\\
\textsc{NewID} &: \mathtt{i}^* \neq \mathtt{i}^\dagger,\\
\textsc{Forge} &: (\mathtt{i}^*, \sigma^*) \text{ is an existential forgery on } \mathsf{Sig}.
\end{aligned}$$

We have the following equalities.

$$\mathbf{Adv}^{\text{eaves-coll}}_{\text{a-auth},\mathbf{A}}(\lambda) = \Pr[\textsc{Acc}], \tag{6}$$
$$\Pr[\textsc{Acc}, \textsc{Ext}, \textsc{TgtIdx}, \textsc{NewID}] = \Pr[\textsc{Forge}], \tag{7}$$
$$\Pr[\textsc{Forge}] = \mathbf{Adv}^{\text{euf-cma}}_{\mathsf{Sig},\mathbf{F}}(\lambda). \tag{8}$$

The left-hand side of the equality (7) is expanded as follows.

$$\begin{aligned}
&\Pr[\textsc{Acc}, \textsc{Ext}, \textsc{TgtIdx}, \textsc{NewID}]\\
=&\Pr[\textsc{TgtIdx}] \cdot \Pr[\textsc{Acc}, \textsc{Ext}, \textsc{NewID}]\\
=&\Pr[\textsc{TgtIdx}] \cdot \Pr[\textsc{Acc}, \textsc{Ext}] \cdot \Pr[\textsc{NewID} \mid \textsc{Acc}, \textsc{Ext}]\\
=&\Pr[\textsc{TgtIdx}] \cdot \Pr[\textsc{Acc}] \cdot \Pr[\textsc{Ext} \mid \textsc{Acc}] \cdot \Pr[\textsc{NewID} \mid \textsc{Acc}, \textsc{Ext}].
\end{aligned} \tag{9}$$

Claim 1

$$\Pr[\textsc{TgtIdx}] = 1/|A| = 1/\mu. \tag{10}$$

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

Claim 2

$$\Pr[\textsc{NewID} \mid \textsc{Acc}, \textsc{Ext}] = \frac{p-1}{p}. \tag{11}$$

Proof. $\mathtt{i}^*$ is not $\mathtt{i}^\dagger$ with probability $\frac{p-1}{p}$. This is because $\mathtt{i}^\dagger$ is chosen uniformly at random from $\check{\mathbb{H}}$ by $\mathbf{F}$. Note here that, though the information of the whole witness space might leak to $\mathbf{A}$, the information that identify a witness including $\mathtt{i}^\dagger$ does not leak due to the perfect witness-indistinguishability of the Groth-Sahai proof system (see Definition 8 in Appendix). □

Claim 3 *If* TGTIDX *and* NEWID *occurs, then $\mathtt{i}^*$ is not queried by* $\mathbf{F}$ *to its oracle* $\mathbf{SignO}_{pp}$.

Proof. This is because of the restriction (4)(5). □

Claim 4

$$\Pr[\textsc{Ext} \mid \textsc{Acc}] = 1. \tag{12}$$

Proof. This is because of the perfect knowledge extraction of Π_{pp} (see Definition 7 in Appendix). □

Combining (6), (7), (8), (9), (10), (11) and (12) we have:

$$\mathbf{Adv}^{\text{eaves-coll}}_{\mathsf{a\text{-}auth},\mathbf{A}}(\lambda) = \frac{p}{p-1} \cdot \mu \cdot \mathbf{Adv}^{\text{euf-cma}}_{\mathsf{Sig},\mathbf{F}}(\lambda).$$

□

Theorem 2 (Anonymity) *Assume the computational indistinguishability between commitment keys $\{ck\}$ of the mode* $\mathtt{nor}$ *and commitment keys $\{ck\}$ of the mode* $\mathtt{sim}$. *For any* PPT *algorithm* $\mathbf{A}$ *that is in accordance with the experiment* $\mathsf{Exp}^{ano}_{\mathsf{a\text{-}auth},\mathbf{A}}(1^\lambda)$*, there exists a* PPT *algorithm* $\mathbf{D}$ *and the following equality holds.*

$$\mathbf{Adv}^{ano}_{\mathsf{a\text{-}auth},\mathbf{A}}(\lambda) \le \mathbf{Adv}^{ind\text{-}dual}_{\mathsf{Cmt}_{pp},\mathbf{D}}(\lambda).$$

(For the definition of $\mathbf{Adv}^{ind\text{-}dual}_{\mathsf{Cmt}_{pp},\mathbf{D}}(\lambda)$*, see Definition 2 in Appendix.)*

The meaning of this theorem is that, if the dual-mode commitment keys are indistinguishable, then our $\mathsf{a\text{-}auth}$ has anonymity.

Proof. Suppose that any PPT algorithm $\mathbf{A}$ that is in accordance with the experiment $\mathsf{Exp}^{\text{ano}}_{\mathsf{a\text{-}auth},\mathbf{A}}(1^\lambda)$ is given. We set a sequence of games, Game_0 and Game_1, as follows. Game_0 is exactly the same as $\mathsf{Exp}^{\text{ano}}_{\mathsf{a\text{-}auth},\mathbf{A}}(1^\lambda)$. Note that when a set of public parameters $pp = (pp', ck)$ is given to $\mathbf{A}$ where pp' is for bilinear groups, the commitment key ck is chosen as a commitment key ck of the mode $\mathtt{nor}$. We denote the probability that Game_0 returns WIN as $\Pr[\textsc{Win}_0]$.

Game_1 is the same as Game_0 except that, when a set of public parameters $pp = (pp', ck)$ is given to $\mathbf{A}$, the commitment key ck is chosen as a commitment key ck of the mode $\mathtt{sim}$. We denote the probability that Game_1 returns WIN as $\Pr[\textsc{Win}_1]$. The values in Game_1 distribute identically for both $\mathtt{i}_0$ and $\mathtt{i}_1$ due to the perfectly hiding property (18) and the witness-indistinguishability (28). Therefore, $\Pr[\textsc{Win}_1] = 1/2$.

Employing $\mathbf{A}$ as a subroutine, we construct a PPT distinguisher algorithm $\mathbf{D}$ as follows. Given input pp, ck and an auxiliary input μ $\mathbf{D}$ reads out the security parameter. $\mathbf{D}$ simulates the environment of $\mathbf{A}$ in Game_0 or Game_1 honestly except that $\mathbf{D}$ puts $pp := (pp, ck)$ instead of executing $\mathsf{Setup}(1^\lambda)$. If $b = b'$, then $\mathbf{D}$ returns 1, and otherwise, 0. By the definition of (16), $\Pr[\mathbf{D}(pp, ck) = 1 \mid ck \leftarrow \mathsf{Cmt.KG}_{pp}(\mathtt{nor})] = \Pr[\textsc{Win}_0]$ and $\Pr[\mathbf{D}(pp, ck) = 1 \mid (ck, tk) \leftarrow \mathsf{Cmt.KG}_{pp}(\mathtt{sim})] = \Pr[\textsc{Win}_1]$, and

$$\mathbf{Adv}^{\text{ind-dual}}_{\mathsf{Cmt}_{pp},\mathbf{D}}(\lambda) = |\Pr[\textsc{Win}_0] - \Pr[\textsc{Win}_1]|.$$

Therefore,

$$\begin{aligned}\mathbf{Adv}^{\text{ano}}_{\mathsf{a\text{-}auth},\mathbf{A}}(\lambda) &= |\Pr[\textsc{Win}_0] - (1/2)| \\ &\le |\Pr[\textsc{Win}_0] - \Pr[\textsc{Win}_1]| + |\Pr[\textsc{Win}_1] - (1/2)| \\ &= \mathbf{Adv}^{\text{ind-dual}}_{\mathsf{Cmt}_{pp},\mathbf{D}}(\lambda) + 0 = \mathbf{Adv}^{\text{ind-dual}}_{\mathsf{Cmt}_{pp},\mathbf{D}}(\lambda).\end{aligned}$$

□

5 Conclusion and Future Work

In our future IoT, a decentralized multi-authority anonymous authentication scheme `a-auth` would be needed. In this paper, we gave a construction of `a-auth` that attains the collusion resistance; a prover is able to convince a verifier that a single anonymous prover has the knowledge of attribute credentials related to public keys. Another feature is that, when a prover wants the authorities to issue attribute credentials, the authorities only have to generate digital signatures on her identity `i`. In the case of legitimately issued global identities, the second feature is useful.

Technically, under the mode $ck = \mathtt{nor}$ or $\mathtt{ext}$, perfectly binding property of the commitment to `i` works as a proof of simultaneous satisfiability of the verification equations of structure-preserving signatures, and hence, the collusion attacks are prevented. On the other hand, under the mode $ck = \mathtt{sim}$, perfectly hiding property of commitments and perfectly witness-indistinguishable property of proofs yield anonymity.

In naive use, our `a-auth` is vulnerable to the replay attack. However, as was explained in Section 1, the replay attack will be avoided when our decentralized scheme `a-auth` is used as an authentication framework on a peer-to-peer network with a blockchain. This direction of research should be pursued.

Our future work should be to resolve the drawback that the length of a proof grows linearly to the number of attributes which a prover wants to prove. Developing a cryptographic accumulator would be a candidate direction.

Acknowledgments

The author would like to thank Jens Groth who gave me comments on simultaneous satisfiability of pairing product equations at "The 9th BIU Winter School on Cryptography".

References

1. Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. `http://www.bitcoin.org/bitcoin.pdf`, 2009.
2. Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In *Proc. of the Theory and Applications of Cryptographic Techniques 27th Annual International Conference on Advances in Cryptology, EUROCRYPT'08, Istanbul, Turkey, LNCS*, volume 4965, pages 415–432. Springer-Verlag, April 2008.
3. Jan Camenisch, Maria Dubovitskaya, Kristiyan Haralambiev, and Markulf Kohlweiss. Composable and modular anonymous credentials: Definitions and practical constructions. In *Proc. of Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, LNCS*, volume 9452, pages 262–288. Springer-Verlag, November-December 2015.
4. Jan Camenisch, Manu Drijvers, and Björn Tackmann. Multi-protocol UC and its use for building modular and efficient protocols. *IACR Cryptology ePrint Archive*, 2019/065, 2019.
5. Hiroaki Anada and Seiko Arita. Witness-indistinguishable arguments with Σ-protocols for bundled witness spaces and its application to global identities. In *Proc. of Information and Communications Security - 20th International Conference, ICICS 2018, Lille, France, LNCS*, volume 11149, pages 530–547. Springer-Verlag, October 2018.
6. Hiroaki Anada and Seiko Arita. Witness-indistinguishable arguments with Σ-protocols for bundled witness spaces and its application to global identities. *IACR Cryptology ePrint Archive*, 2018/742, 2018.
7. Ryo Okishima and Toru Nakanishi. An anonymous credential system with constant-size attribute proofs for CNF formulas with negations. In *Proc. of Advances in Information and Computer Security - 14th International Workshop on Security, IWSEC 2019, Tokyo, Japan, LNCS*, volume 11689, pages 89–106. Springer-Verlag, August 2019.
8. AO Kaspersky Lab. What is a replay attack? `https://www.kaspersky.com/resource-center/definitions/replay-attack`. Accessed: 2020-10-24.
9. International Business Machines Corporation (IBM). Resisting replay attacks efficiently in a permissioned and privacy-preserving blockchain network. US 20170149819 A1, United States Patent and Trademark Office, May 2017.
10. Alibaba Group Holding Limited. System and method for detecting replay attack. US 20200128043 A1, United States Patent and Trademark Office, June 2020.
11. Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Structure-preserving signatures and commitments to group elements. In *Proc. of Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, LNCS*, volume 6223, pages 209–236. Springer-Verlag, August 2010.

12. Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Structure-preserving signatures and commitments to group elements. *Journal of Cryptology*, 29(2):363–421, 2016.
13. Alex Escala and Jens Groth. Fine-tuning Groth-Sahai proofs. In *Proc. of Public-Key Cryptography - PKC 2014 - 17th International Conference on Practice and Theory in Public-Key Cryptography, Buenos Aires, Argentina, LNCS*, volume 8383, pages 630–649. Springer-Verlag, March 2014.
14. Masayuki Abe, Dennis Hofheinz, Ryo Nishimaki, Miyako Ohkubo, and Jiaxin Pan. Compact structure-preserving signatures with almost tight security. In *Proc. of Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, LNCS*, volume 10402, pages 548–580. Springer-Verlag, August 2017.
15. Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

Appendix

A Non-interactive Commit-and-Prove Scheme for Structure-Preserving Signatures

In this appendix, using the fine-tuned Groth-Sahai proof system [13], we describe the non-interactive commit-and-prove scheme CmtPrv that is adapted to the case of our specific language of the verification equation system of the structure-preserving signatures [11, 12]. CmtPrv consists of six PPT algorithms $(\mathsf{CmtPrv.Setup}, \mathsf{Cmt}_{pp} = (\mathsf{Cmt.KG}_{pp}, \mathsf{Cmt.Com}_{pp}, \mathsf{Cmt.Vrf}_{pp}), \mathit{\Pi}_{pp} = (\mathsf{P}_{pp}, \mathsf{V}_{pp}))$.

A.1 Commitment-Part

The commitment-part $(\mathsf{CmtPrv.Setup}, \mathsf{Cmt}_{pp})$ is described as follows.
• $\mathsf{CmtPrv.Setup}(1^\lambda) \to pp$. On input the security parameter 1^λ, this PPT algorithm executes a bilinear-groups generation algorithm, and it puts the output as a set of public parameters: $\mathcal{G}(1^\lambda) \to (p, \hat{\mathbb{G}}, \check{\mathbb{H}}, \mathbb{T}, e, \hat{G}, \check{H}) =: pp$. It returns pp.
• $\mathsf{Cmt.KG}_{pp}(\mathtt{mode}) \to key$. On input a string $\mathtt{mode}$, this PPT algorithm generates a key. If $\mathtt{mode} = \mathtt{nor}$, then $key = ck$ which is a commitment key. If $\mathtt{mode} = \mathtt{ext}$, then $key = (ck, xk)$ which is a pair of ck and an extraction key xk. If $\mathtt{mode} = \mathtt{sim}$, then $key = (ck, tk)$ which is a pair of ck and a trapdoor key tk. It returns key.

We put $pp := (pp, ck)$ because the commitment key ck is treated as a public parameter.
• $\mathsf{Cmt.Com}_{pp}(w; r) \to (c, r)$. On input a message w which may be a vector, this PPT algorithm generates a commitment c with a randomness r. It returns (c, r). If w is a vector $w = (w_0, \ldots, w_{n-1})$ (for some $n \in \mathbb{N}$ bounded by a polynomial in λ), then c and r are also vectors of the same number of components: $c = (c_0, \ldots, c_{n-1})$ and $(r_0, \ldots, r_{n-1})$, respectively. Note also that computation is executed in *componentwise way*; c_i is generated from w_i and r_i, $i = 0, \ldots, n-1$.
• $\mathsf{Cmt.Vrf}_{pp}(c, w, r) \to d$. On input a commitment c, a message w and a verification key r, this deterministic algorithm generates a boolean decision d. It returns d.

The commitment-part $(\mathsf{CmtPrv.Setup}, \mathsf{Cmt}_{pp})$ of the Groth-Sahai proof system has the following four properties.

Definition 1 (Correctness [2]) *A commitment scheme Cmt_{pp} is said to be correct if it satisfies the following condition: For any security parameter 1^λ, any set of public parameters $pp \leftarrow \mathsf{CmtPrv.Setup}(1^\lambda)$, any commitment key $ck \leftarrow \mathsf{Cmt.KG}_{pp}(\mathtt{mode})$ where $\mathtt{mode} = \mathtt{nor}$ or $\mathtt{ext}$ or $\mathtt{sim}$, and any message w,*

$$\Pr[d = 1 \mid (c, r) \leftarrow \mathsf{Cmt.Com}_{pp}(w), d \leftarrow \mathsf{Cmt.Vrf}_{pp}(c, w, r)] = 1. \tag{13}$$

Definition 2 (Dual Mode [2]) *A commitment scheme Cmt_{pp} is said to be dual mode if it satisfies the following condition: For any security parameter 1^λ, any set of public parameters $pp \leftarrow \mathsf{CmtPrv.Setup}(1^\lambda)$ and any PPT algorithm $\mathbf{A}$,*

$$\begin{aligned} &\Pr[\mathbf{A}(pp, ck) = 1 \mid ck \leftarrow \mathsf{Cmt.KG}_{pp}(\mathtt{nor})] \\ &= \Pr[\mathbf{A}(pp, ck) = 1 \mid (ck, xk) \leftarrow \mathsf{Cmt.KG}_{pp}(\mathtt{ext})], \qquad (14) \\ &\Pr[\mathbf{A}(pp, ck) = 1 \mid ck \leftarrow \mathsf{Cmt.KG}_{pp}(\mathtt{nor})] \\ &\approx_c \Pr[\mathbf{A}(pp, ck) = 1 \mid (ck, tk) \leftarrow \mathsf{Cmt.KG}_{pp}(\mathtt{sim})]. \qquad (15) \end{aligned}$$

The computational indistinguishability (15) is equivalent to the following: For any security parameter 1^λ, for any set of public parameters $pp \leftarrow \mathsf{CmtPrv.Setup}(1^\lambda)$ and any PPT algorithm $\mathbf{A}$, the advantage $\mathbf{Adv}^{\text{ind-dual}}_{\mathsf{Cmt}_{pp},\mathbf{A}}(\lambda)$ of $\mathbf{A}$ over Cmt_{pp} defined by the difference below is negligible in λ:

$$\begin{aligned}\mathbf{Adv}^{\text{ind-dual}}_{\mathsf{Cmt}_{pp},\mathbf{A}}(\lambda) \overset{\text{def}}{=} |\Pr[\mathbf{A}(pp, ck) = 1 \,|ck \leftarrow \mathsf{Cmt.KG}_{pp}(\mathtt{nor})] \\ - \Pr[\mathbf{A}(pp, ck) = 1 \,|(ck, tk) \leftarrow \mathsf{Cmt.KG}_{pp}(\mathtt{sim})]|. \end{aligned} \tag{16}$$

The indistinguishability holds, for example, for an instance of the Groth-Sahai proof system under the SXDH assumption [2, 13].

Definition 3 (Perfectly Binding [2]) *A commitment scheme Cmt_{pp} is said to be perfectly binding if it satisfies the following condition for some unbounded algorithm $\mathsf{Cmt.Open}_{pp}$: For any security parameter 1^λ, any set of public parameters $pp \leftarrow \mathsf{CmtPrv.Setup}(1^\lambda)$, any commitment key $ck \leftarrow \mathsf{Cmt.KG}_{pp}(\mathtt{nor})$ and any message w,*

$$\Pr[w = w' \mid (c, r) \leftarrow \mathsf{Cmt.Com}_{pp}(w; r), w' \leftarrow \mathsf{Cmt.Open}_{pp}(c)] = 1. \tag{17}$$

Definition 4 (Perfectly Hiding [2]) *A commitment scheme Cmt_{pp} is said to be perfectly hiding if it satisfies the following condition: For any security parameter 1^λ, any set of public parameters $pp \leftarrow \mathsf{CmtPrv.Setup}(1^\lambda)$, any commitment key ck s.t. $(ck, tk) \leftarrow \mathsf{Cmt.KG}_{pp}(\mathtt{sim})$ and any PPT algorithm $\mathbf{A}$,*

$$\begin{aligned}&\Pr[\mathbf{A}(St, c) = 1 \,|(w, w', St) \leftarrow \mathbf{A}(pp, ck, tk); \\ &\qquad (c, r) \leftarrow \mathsf{Cmt.Com}_{pp}(w)] \\ &= \Pr[\mathbf{A}(St, c') = 1 \,|(w, w', St) \leftarrow \mathbf{A}(pp, ck, tk); \\ &\qquad (c', r') \leftarrow \mathsf{Cmt.Com}_{pp}(w')].\end{aligned} \tag{18}$$

A.2 Proof-Part

The proof-part $(\mathsf{CmtPrv.Setup}, \mathit{\Pi}_{pp})$ is described as follows. Let $\mathcal{CK}_{pp}$ denote the set of commitment keys, $\mathcal{X}_{pp}$ denote the set of coefficients of the verification equation system (1) and (2), and $\mathcal{W}_{pp}$ denote the set of the pairs of messages and signatures for some $x \in \mathcal{X}_{pp}$:

$$\mathcal{CK}_{pp} = \{ck \mid ck \leftarrow \mathsf{Cmt.KG}_{pp}(\mathtt{mode}) \text{ for } \mathtt{mode} = \mathtt{nor}, \mathtt{ext}, \mathtt{sim}\}, \tag{19}$$

$$\mathcal{X}_{pp} = \{x \mid (\mathrm{PK}, \mathrm{SK}) \leftarrow \mathsf{Sig.KG}_{pp}(), x = \mathrm{PK}\}, \tag{20}$$

$$\begin{aligned}\mathcal{W}_{pp} = \{w \mid w = (w_0, w_1, \ldots, w_7) \in \check{\mathbb{H}}^3 \times \hat{\mathbb{G}} \times \check{\mathbb{H}}^2 \times \hat{\mathbb{G}} \times \check{\mathbb{H}} \\ \text{s.t. (1) and (2) hold for } \exists x \in \mathcal{X}_{pp}, \\ w_0 = m = \check{M},\ (w_1, \ldots, w_7) = \sigma = (\check{Z}, \check{R}, \hat{S}, \check{T}, \check{U}, \hat{V}, \check{W})\}.\end{aligned} \tag{21}$$

Then we define the following ternary relation R_{pp}.

$$\begin{aligned}R_{pp} \overset{\text{def}}{=} \{(ck, x, w) \in \mathcal{CK}_{pp} \times \mathcal{X}_{pp} \times \mathcal{W}_{pp} \mid \\ w \text{ can be committed by } \mathsf{Cmt.Com}_{pp} \text{ under } ck \\ \text{and (1) and (2) hold for } (x, w)\}.\end{aligned} \tag{22}$$

A group-dependent language $L_{pp,ck}$ parametrized by $ck \in \mathcal{CK}$ is defined as follows.

$$L_{pp,ck} \overset{\text{def}}{=} \{x \in \mathcal{X}_{pp} \mid \exists w \in \mathcal{W}_{pp} \text{ s.t. } (ck, x, w) \in R_{pp}\}. \tag{23}$$

We put $pp := (pp, ck)$ because the commitment key ck is treated as a public parameter.
• $\mathsf{P}_{pp}(x, c, w, r) \rightarrow \pi$. On input a statement x, a commitment c, a witness w and a randomness r which was used to generate a commitment c, this PPT algorithm executes the proof-generation algorithm of the Groth-Sahai proof system to obtain a proof π (see [13] for the details). It returns π.
• $\mathsf{V}_{pp}(x, c, \pi) \rightarrow d$. On input a statement x, a commitment c and a proof π, this deterministic algorithm executes the verification algorithm of the Groth-Sahai proof system to obtain a boolean decision d (see [13] for the details). It returns d.

The proof-part $(\mathsf{CmtPrv.Setup}, \mathit{\Pi}_{pp})$ of the Groth-Sahai proof system have the following four properties.

Definition 5 (Perfect Correctness [2]) *A commit-and-prove scheme* CmtPrv *is said to be perfectly correct if it satisfies the following condition: For any security parameter* 1^λ*, any set of public parameters* $pp \leftarrow \mathsf{CmtPrv.Setup}(1^\lambda)$*, any commitment key* $ck \leftarrow \mathsf{Cmt.KG}_{pp}(\mathtt{mode})$ *where* $\mathtt{mode} = \mathtt{nor}$ *or* $\mathtt{ext}$ *or* $\mathtt{sim}$*, and any* PPT *algorithm* $\mathbf{A}$*,*

$$\begin{aligned}&\Pr[\mathsf{V}_{pp}(x,c,\pi) = 1 \textit{ if } (ck,x,w) \in R_{pp} \mid \\ &\quad (x,w) \leftarrow \mathbf{A}(pp), (c,r) \leftarrow \mathsf{Cmt.Com}_{pp}(w), \\ &\quad \pi \leftarrow \mathsf{P}_{pp}(x,c,w,r)] = 1. \end{aligned} \tag{24}$$

Definition 6 (Perfect Soundness [2]) *A commit-and-prove scheme* CmtPrv *is said to be perfectly sound if it satisfies the following condition for some unbounded algorithm* $\mathsf{Cmt.Open}_{pp}$*: For any security parameter* 1^λ*, any set of public parameters* $pp \leftarrow \mathsf{CmtPrv.Setup}(1^\lambda)$*, any commitment key* $ck \leftarrow \mathsf{Cmt.KG}_{pp}(\mathtt{nor})$ *and any* PPT *algorithm* $\mathbf{A}$*,*

$$\begin{aligned}&\Pr[\mathsf{V}_{pp}(x,c,\pi) = 0 \textit{ or } (ck,x,w) \in R_{pp} \mid \\ &\quad (x,c,\pi) \leftarrow \mathbf{A}(pp); w \leftarrow \mathsf{Cmt.Open}_{pp}(c)] = 1. \end{aligned} \tag{25}$$

Let $\mathcal{C}_{ck}$ be the set of commitments under ck to some message w.

Definition 7 (Perfect Knowledge Extraction [2]) *A commit-and-prove scheme* CmtPrv *is said to be perfectly knowledge extractable if it satisfies the following condition for some* PPT *algorithm* $\mathsf{Cmt.Ext}_{pp}$*: For any security parameter* 1^λ*, any set of public parameters* $pp \leftarrow \mathsf{CmtPrv.Setup}(1^\lambda)$*, any commitment key* $(ck, xk) \leftarrow \mathsf{Cmt.KG}_{pp}(\mathtt{ext})$ *and any* PPT *algorithm* $\mathbf{A}$*,*

$$\begin{aligned}&\Pr[c \notin \mathcal{C}_{ck} \textit{ or } \mathsf{Cmt.Ext}_{pp}(xk,c) = \mathsf{Cmt.Open}_{pp}(c) \mid \\ &\quad c \leftarrow \mathbf{A}(pp,ck,xk)] = 1. \end{aligned} \tag{26}$$

Definition 8 (Composable Witness-Indistinguishability [2]) *A commit-and-prove scheme* CmtPrv *is said to be composably witness-indistinguishable if it satisfies the following condition: For any security parameter* 1^λ*, any set of public parameters* $pp \leftarrow \mathsf{CmtPrv.Setup}(1^\lambda)$ *and any* PPT *algorithm* $\mathbf{A}$*,*

$$\begin{aligned}&\Pr[\mathbf{A}(pp,ck) = 1 \mid ck \leftarrow \mathsf{Cmt.KG}_{pp}(\mathtt{nor})] \\ \approx_c\ &\Pr[\mathbf{A}(pp,ck) = 1 \mid (ck,tk) \leftarrow \mathsf{Cmt.KG}_{pp}(\mathtt{sim})], \end{aligned} \tag{27}$$

$$\begin{aligned}&\Pr[(ck,x,w),(ck,x,w') \in R_{pp} \textit{ and } \mathbf{A}(St,\pi) = 1 \mid \\ &\quad (ck,tk) \leftarrow \mathsf{Cmt.KG}_{pp}(\mathtt{sim}); pp := (pp,ck); \\ &\quad (x,w,w',St) \leftarrow \mathbf{A}^{\mathsf{Cmt.Com}_{pp}(\cdot)}(pp,ck,tk); \\ &\quad (c,r) \leftarrow \mathsf{Cmt.Com}_{pp}(w); \pi \leftarrow \mathsf{P}_{pp}(x,c,w,r)] \\ =\ &\Pr[(ck,x,w),(ck,x,w') \in R_{pp} \textit{ and } \mathbf{A}(St,\pi') = 1 \mid \\ &\quad (ck,tk) \leftarrow \mathsf{Cmt.KG}_{pp}(\mathtt{sim}); pp := (pp,ck); \\ &\quad (x,w,w',St) \leftarrow \mathbf{A}^{\mathsf{Cmt.Com}_{pp}(\cdot)}(pp,ck,tk); \\ &\quad (c',r') \leftarrow \mathsf{Cmt.Com}_{pp}(w'); \pi' \leftarrow \mathsf{P}_{pp}(x,c',w',r')]. \end{aligned} \tag{28}$$