Abstract. We propose a decentralized multi-authority anonymous authentication scheme in which a prover and a verifier are non-interactive. We give two security definitions; resistance against collusion attacks that cause misauthentication, and anonymity for privacy protection. Then we give a construction under a principle of “commit-to-ID”. We employ two building blocks; the structure-preserving signature scheme and the Groth-Sahai non-interactive proof system, the both of which are based on bilinear groups. We give security proofs in the standard model, which reduce to the security of the building blocks.

Keywords: anonymous authentication, attribute, collusion attack, identity, non-interactive

1 Introduction

Privacy protection in cyber-physical space is a function that should be pursued in authentication. The growth of companies (such as GAFA) in the areas of the IT infrastructures made protecting privacy of vital importance for involved users in recent years because we use search engines, digital devices, social networking services and e-shopping services everyday. Considering the change of our life and business going the cyber-physical space, one of the critical aspects that our future authentication framework should attain is anonymous authentication via attributes. For example, connected-to-the-internet vehicles, bicycles and even human beings with embedded devices will use plural services like GPS, availability of nearby places, disclosed data of other movable entities, and suitable options of business strategy. There is no need of identity information of the user, but instead, the user should be authenticated in anonymous way using her attribute certificates issued beforehand by independent administration authorities related to the service providers. Another aspect is optimization of a service based on plural other services. That is, there will be a compound service model which involves the independent administration authorities at a time. For example, location data from GPS, availability information of nearby places and disclosed data of other entities can be thrown into the input of smart computation of optimizing business strategies.

However, there is a threat on such a framework of anonymous authentication using plural independent attribute certificates; collusion attack. That is, malicious users of different identities bring together their attribute certificates. They try to make a verifier accept anonymously using the merged attribute certificates. Here the very anonymity is a critical potential drawback from the view point of the collusion attack that causes misauthentication.

1.1 Related Work and Our Contribution

In this paper, we propose a decentralized multi-authority anonymous authentication scheme to resolve the above problem. Our scheme is a special case of a decentralized multi-authority attribute-based
signature scheme (DMA-ABS). One of the state-of-the-art schemes of DMA-ABS is due to Okamoto-Takashima [OT13]. Their scheme can treat general non-monotone access structures, while our scheme treats only the all-AND structure. Nonetheless, a feature of our scheme is that, when a prover wants the authorities to issue private secret keys as attribute certificates, the authorities simply generate digital signatures on her global identity group element. This feature is useful when her global identity group element is easy to be validated in registration phase by the authorities. The second feature of our scheme is that the authorities are independent each other, while the computational amount and the proof length of a prover grows linearly to the number of authorities involved. There is also previous work by Anada-Arita [AA18a,AA18b] which shares the above two features. However, their authentication scheme is interactive between a prover and a verifier in accordance with the $\Sigma$-protocol [Dam10], whereas our scheme is non-interactive based on the Groth-Sahai proof system [GS08] yielding better availability for applications which need quick authentication.

Theoretically, we note that our scheme is captured as a variant of multi-prover proof system [Gol01]. A novel point is that, in our scheme, the verifier is convinced that the multi-provers are actually a single prover who knows each witness of each corresponding statement, while anonymity is kept.

1.2 Overview of Our Construction and Security Proofs

In Section 6, we will construct a decentralized multi-authority anonymous authentication scheme with non-interactive proofs (NI-DMA-A-AUTH). There we employ two building blocks. One is the structure-preserving signature scheme [AFG10,AFG16] (see Section 2.2). Each decentralized authority indexed by ‘a’ issues a private secret key $sk_{gid}^a$ by signing a global identity group element $gid$ of a prover. Here $gid$ is a group element of one of the source groups of bilinear groups. As a remark, a realistic constraint is that the authorities have to refer to a common set of public parameters $pp$, as is usual in the case of NIST Standard (like NIST.FIPS.186-4 [NIS13]). The other building block is the non-interactive commit-and-prove scheme of the fine-tuned Groth-Sahai proof system [GS08,EG14]. We give a description of a version adopted to the case of proving knowledge of the structure-preserving signatures (see Section 3). Our construction is under a principle of “commit-to-ID”. That is, in the commit-phase the prover commits to an identity group element $gid$ (“global identity”). She also commits to the components of the structure-preserving signatures, which are also group elements. In the prove-phase the prover generates a proof $\pi$ of knowing a solution of the verification equation system of the structure-preserving signatures. Here the common single commitment $c_0$ to $gid$ works for proving knowledge of bundled witnesses (see “bundled product of the language” in Section 4). Thus, the collusion attack becomes impossible due to the binding property of the commitment $c_0$ because $c_0$ is common in the verification equation system. In other words, our construction builds on the “plug-in” mechanism of commitments in generating a proof $\pi$ in the Groth-Sahai proof system (see [GS08]).

In Section 5, after giving the syntax of our NI-DMA-A-AUTH, we give two security definitions. One is resistance against collusion attacks that cause misauthentication, and the other is anonymity for privacy protection. There we capture the requirements described above. In Section 6, security proofs for the above construction are given in the standard model. The resistance against collusion attacks is due to knowledge extraction property of the Groth-Sahai proof system and existential unforgeability of the structure-preserving signature scheme. The anonymity is due to perfectly hiding property of commitments and perfectly witness indistinguishable property of proofs of the Groth-Sahai proof system, where the both properties hold in the simulation mode of the dual mode commitment.

2 Preliminaries

The set of natural numbers is denoted by $\mathbb{N}$. We put $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$. The residue class ring of integers modulo a prime number $p$ is denoted by $\mathbb{Z}_p$. The security parameter is denoted by $\lambda$, where $\lambda \in \mathbb{N}$. A probability $P$ is said to be negligible in $\lambda$ if for any given positive polynomial $\text{poly}(\lambda)$, $P < 1/\text{poly}(\lambda)$ for sufficiently large $\lambda \in \mathbb{N}$. Two probabilities $P$ and $Q$ are said to be computationally indistinguishable if $|P - Q|$ is negligible in $\lambda$, which is denoted as $P \approx_c Q$. $St$ means the inner state of an algorithm.
2.1 Bilinear Groups [GPS08,EG14]

Let \( \mathcal{G} \) be a generation algorithm of bilinear groups \([GPS08]\): \( \mathcal{G}(1^\lambda) \rightarrow (p, \hat{G}, \hat{H}, T, e, \hat{G}, \hat{H}) \). Here \( p \) is a prime number of bit-length \( \lambda \), \( \hat{G}, \hat{H} \) and \( T \) are cyclic groups of order \( p \), and \( \hat{G} \) and \( \hat{H} \) are generators of \( \hat{G} \) and \( \hat{H} \), respectively. We denote operations in \( \hat{G}, \hat{H} \) and \( T \) multiplicatively. \( e \) is a map \( e: \hat{G} \times \hat{H} \rightarrow T \) with the following two properties:

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

Hereafter we denote an element in \( \hat{G} \) and \( \hat{H} \) with hat: \( ' \) and check: \( \check{\cdot} \), respectively.

2.2 Structure-Preserving Signature Scheme [AFG+10,AFG+16]

The structure-preserving signature scheme \( \text{Sig} \) is a digital signature scheme based on bilinear groups, in which a message is a vector whose entries belong to one of the two source groups \( \hat{G} \) and \( \hat{H} \), and a signature is a vector whose entries belong to \( \hat{G} \) and \( \hat{H} \). Based on Abe et al. \([AFG+10,AFG+16]\), we survey the four ppt algorithms of the structure-preserving signature scheme \( \text{Sig} = (\text{Sig.Setup, Sig.KG}_{pp}, \text{Sig.Sign}_{pp}, \text{Sig.Vrf}_{pp}) \).

\( \text{Sig.Setup}(1^\lambda) \rightarrow pp \). On input the security parameter \( 1^\lambda \), this ppt algorithm executes the generation algorithm of bilinear groups, and it puts the output as a set of public parameters: \( \mathcal{G}(1^\lambda) \rightarrow (p, \hat{G}, \hat{H}, T, e, \hat{G}, \hat{H}) =: pp \). It returns \( pp \).

\( \text{Sig.KG}_{pp}() \rightarrow (PK, SK) \). Based on the set of public parameters \( pp \), this ppt algorithm generates a signing key \( SK \) and the corresponding public key \( PK \) as follows: \( \hat{G}_u \in_R \hat{G}, \gamma_1, \delta_1 \in_R \mathbb{Z}_p^*, \hat{G}_1 := \hat{G}^{\gamma_1}, \hat{G}_{u,1} := \hat{G}_u^{\delta_1}, \gamma_2, \delta_2 \in_R \mathbb{Z}_p^*, \hat{G}_z := \hat{G}^{\gamma_2}, \hat{G}_{u,z} := \hat{G}_u^{\delta_2}, \alpha, \beta \in_R \mathbb{Z}_p^*, (\hat{A}_1, \hat{A}_i)_{i=0}^1 \leftarrow \text{Extend}(\hat{G}, \hat{H}^\alpha), (\hat{B}_1, \hat{B}_i)_{i=0}^1 \leftarrow \text{Extend}(\hat{G}_z, \hat{H}^\beta) \) (for \( \text{Extend} \), see \([AFG+16]\)). It puts \( PK := (\hat{G}_z, \hat{G}_{u,z}, \hat{G}_u, \hat{G}_1, \hat{G}_{u,1}, (\hat{A}_i, \hat{A}_1), (\hat{B}_i, \hat{B}_1))_{i=0}^1 \) and \( SK := (\alpha, \beta, \gamma_2, \delta_2, \gamma_1, \delta_1) \). It returns \( PK, SK \).

\( \text{Sig.Sign}_{pp}(PK, SK, m) \rightarrow \sigma \). On input the public key \( PK \), the secret key \( SK \) and a message \( m = \hat{M} \in \hat{H} \), this ppt algorithm generates a signature \( \sigma \) as follows.

\( \zeta, \rho, \tau, \phi, \omega \in_R \mathbb{Z}_p, \hat{Z} := \hat{H}^\zeta, \hat{R} := \hat{H}^{\alpha - \rho \tau - \gamma_1 \zeta} \hat{M}^{-\gamma_1} \hat{S} := \hat{G}^\rho, \hat{T} := \hat{H}^\tau, \hat{U} := \hat{H}^{\beta - \phi \omega - \delta_1 \zeta} \hat{M}^{-\delta_1} \hat{V} := \hat{G}_u^\phi \hat{W} := \hat{H}^\omega \).

It returns \( \sigma := (\hat{Z}, \hat{R}, \hat{S}, \hat{T}, \hat{U}, \hat{V}, \hat{W}) \).

\( \text{Sig.Vrf}_{pp}(PK, m, \sigma) \rightarrow d \). On input the public key \( PK \), a message \( m = \hat{M} \in \hat{H} \) and a signature \( \sigma = (\hat{Z}, \hat{R}, \hat{S}, \hat{T}, \hat{U}, \hat{V}, \hat{W}) \), this deterministic algorithm checks whether the following verification equation system holds or not.

\[
e(\hat{G}_z, \hat{Z}) e(\hat{G}, \hat{R}) e(\hat{S}, \hat{T}) e(\hat{G}_1, \hat{M}) e(\hat{A}_0, \hat{A}_0)^{-1} e(\hat{A}_1, \hat{A}_1)^{-1} = 1_T, \quad \text{and} \quad (1)
\]

\[
e(\hat{G}_{u,z}, \hat{Z}) e(\hat{G}_u, \hat{U}) e(\hat{V}, \hat{W}) e(\hat{G}_{u,1}, \hat{M}) e(\hat{B}_0, \hat{B}_0)^{-1} e(\hat{B}_1, \hat{B}_1)^{-1} = 1_T. \quad (2)
\]

It returns a boolean decision \( d \).

The correctness should hold for the scheme \( \text{Sig} \): For any security parameter \( 1^\lambda \), any set of public parameters \( pp \leftarrow \text{Sig.Setup}(1^\lambda) \) and any message \( m = \hat{M} \in \hat{H} \), \( \Pr[d = 1 \mid (PK, SK) \leftarrow \text{Sig.KG}_{pp}(), \sigma \leftarrow \text{Sig.Sign}_{pp}(PK, SK, m), d \leftarrow \text{Sig.Vrf}_{pp}(PK, m, \sigma)] = 1. \)
Adaptive chosen-message attack of an existential forgery on the scheme $\Sigma$ by a forger algorithm $F$ is defined by the following algorithm of experiment.

$$\text{Exp}_{\Sigma,F}^{\text{euf-cma}}(1^\lambda):$$

$\text{pp} \leftarrow \Sigma.\text{Setup}(1^\lambda), (\text{PK}, \text{SK}) \leftarrow \Sigma.\text{KG}_{\text{pp}}()$

$$\text{pp} \leftarrow \text{pp}, (m^*, \sigma^*) \leftarrow \text{F}.\text{SignO}_{\text{pp}}((\text{PK}, \text{SK}), \text{pp})$$

If $m^* \notin \{m_j\}_{1 \leq j \leq q_0}$ and $\Sigma.\text{Vrf}_{\text{pp}}(\text{PK}, m^*, \sigma^*) = 1$, then Return $\text{WIN}$ else Return $\text{LOSE}$

In the experiment, $F$ issues a signing query to its signing oracle $\text{SignO}_{\text{pp}}((\text{PK}, \text{SK}), \cdot)$ by sending a message $m_j$ at most $q_0$ times ($1 \leq j \leq q_0$). As a reply, $F$ receives a valid signature $\sigma_j$ on $m_j$. After receiving replies, $F$ returns a pair of a message and a signature $(m^*, \sigma^*)$. A restriction is imposed on the algorithm $F$: The set of queried messages $\{m_j\}_{1 \leq j \leq q_0}$ should not contain the message $m^*$. The advantage of $F$ over $\Sigma$ is defined as $\text{Adv}_{\Sigma,F}^{\text{euf-cma}}(\lambda) := \text{Pr}[\text{Exp}_{\Sigma,F}^{\text{euf-cma}}(1^\lambda) \text{ returns } \text{WIN}].$ The scheme $\Sigma$ is said to be existentially unforgeable against adaptive chosen-message attacks (EUF-CMA) if for any PPT algorithm $F$ and any $q_0$ bounded by a polynomial in $\lambda$, the advantage $\text{Adv}_{\Sigma,F}^{\text{euf-cma}}(\lambda)$ is negligible in $\lambda$. The structure-preserving signature scheme [AFG+10, AFG+16] is known to be EUF-CMA under the $q$-SFP assumption.

### 3 Non-interactive Commit-and-Prove Scheme for Structure-Preserving Signatures

In this section, we give a description of the non-interactive commit-and-prove scheme of the fine-tuned Groth-Sahai proof system [EG14] adapted to the case of our specific group-dependent language; that is, the language of the verification equation system of the structure-preserving signature. A commit-and-prove scheme $\text{CmtPrv}$ consists of six PPT algorithms: $\text{CmtPrv} = (\text{CmtPrv.Setup}, \text{Cmt}_{pp} = (\text{Cmt.\text{KG}}_{pp}, \text{Cmt.\text{Com}}_{pp}, \text{Cmt.\text{Vrf}}_{pp}), \Pi_{pp} = (\text{P}_{pp}, \text{V}_{pp}))$.

#### 3.1 Commitment Part

The commitment part $(\text{CmtPrv.Setup}, \text{Cmt}_{pp})$ is described as follows.

- **CmtPrv.Setup$(1^\lambda)$** $\rightarrow$ $\text{pp}$. On input the security parameter $1^\lambda$, this PPT algorithm executes a generation algorithm of bilinear groups, and it puts the output as a set of public parameters: $G(1^\lambda) \rightarrow (p, G, H, T, e, \bar{G}, \bar{H}) =: \text{pp}$. It returns $\text{pp}$.

- **Cmt.\text{KG}_{pp}(\text{mode})** $\rightarrow$ $\text{key}$. On input a string $\text{mode}$, this PPT algorithm generates a $\text{key}$. If $\text{mode} = \text{nor}$, then $\text{key} = \text{ck}$ which is a commitment key. If $\text{mode} = \text{ext}$, then $\text{key} = (\text{ck}, \text{zk})$ which is a pair of $\text{ck}$ and an extraction key $\text{zk}$. If $\text{mode} = \text{sim}$, then $\text{key} = (\text{ck}, \text{tk})$ which is a pair of $\text{ck}$ and a trapdoor key $\text{tk}$. It returns $\text{key}$.

We put $\text{pp} := (\text{pp}, \text{ck})$ because the commitment key $\text{ck}$ is treated as a public parameter.

- **Cmt.\text{Com}_{pp}(w, r)$** $\rightarrow (c, r)$. 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 $\lambda$), 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 $r_i$, $i = 0, \ldots, n - 1$.

- **Cmt.\text{Vrf}_{pp}(c, w, r)$** $\rightarrow d$. 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 $(\text{CmtPrv.Setup}, \text{Cmt}_{pp})$ of the Groth-Sahai proof system has the following four properties.
Definition 1 (Correct [GS08]) A commitment scheme $\text{Cmt}_pp$ is said to be correct if it satisfies the following condition: For any security parameter $1^\lambda$, any set of public parameters $pp \leftarrow \text{CmtPrv.Setup}(1^\lambda)$, any commitment key $ck \leftarrow \text{Cmt.KG}_pp(\text{mode})$ where $\text{mode} = \text{nor}$ or $\text{ext}$ or $\text{sim}$, and any message $w$, 
\[
\Pr[d = 1 \mid (c, r) \leftarrow \text{Cmt.Com}_pp(w), d \leftarrow \text{Cmt.Vrf}_pp(c, w, r)] = 1.
\]

Definition 2 (Dual Mode [GS08]) A commitment scheme $\text{Cmt}_pp$ is said to be dual mode if it satisfies the following condition: For any security parameter $1^\lambda$, any set of public parameters $pp \leftarrow \text{CmtPrv.Setup}(1^\lambda)$ and any PPT algorithm $A$,
\[
\Pr[A(pp, ck) = 1 \mid ck \leftarrow \text{Cmt.KG}_pp(\text{nor})] \\
= \Pr[A(pp, ck) = 1 \mid (ck, ck) \leftarrow \text{Cmt.KG}_pp(\text{ext})], \\
\Pr[A(pp, ck) = 1 \mid ck \leftarrow \text{Cmt.KG}_pp(\text{nor})] \\
\approx \Pr[A(pp, ck) = 1 \mid (ck, tk) \leftarrow \text{Cmt.KG}_pp(\text{sim})].
\]

The above computational indistinguishability holds under the SXDH assumption [GS08,EG14].

Definition 3 (Perfectly Binding [GS08]) A commitment scheme $\text{Cmt}_pp$ is said to be perfectly binding if it satisfies the following condition for some unbounded algorithm $\text{Cmt.Open}_pp$: For any security parameter $1^\lambda$, any set of public parameters $pp \leftarrow \text{CmtPrv.Setup}(1^\lambda)$, any commitment key $ck \leftarrow \text{Cmt.KG}_pp(\text{nor})$ and any message $w$,
\[
\Pr[w = w' \mid (c, r) \leftarrow \text{Cmt.Com}_pp(w; r), w' \leftarrow \text{Cmt.Open}_pp(c)] = 1.
\]

Definition 4 (Perfectly Hiding [GS08]) A commitment scheme $\text{Cmt}_pp$ is said to be perfectly hiding if it satisfies the following condition: For any security parameter $1^\lambda$, any set of public parameters $pp \leftarrow \text{CmtPrv.Setup}(1^\lambda)$, any commitment key $ck \leftarrow \text{Cmt.KG}_pp(\text{nor})$ and any PPT algorithm $A$,
\[
\Pr[A((St, c) = 1 \mid (w, w', St) \leftarrow A(pp, ck, tk)(c, r) \leftarrow \text{Cmt.Com}_pp(w)] \\
= \Pr[A((St, c') = 1 \mid (w, w', St) \leftarrow A(pp, ck, tk)(c', r') \leftarrow \text{Cmt.Com}_pp(w')]]
\]

3.2 Proof Part

The proof-part ($\text{CmtPrv.Setup}$, $\Pi_{pp}$) 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 $x \in \mathcal{X}_{pp}$:

\[
\mathcal{CK}_pp = \{ck \mid ck \leftarrow \text{Cmt.KG}_pp(\text{mode}) \text{ for mode = nor or ext or sim}\}, \\
\mathcal{X}_{pp} = \{x \mid (PK, SK) \leftarrow \text{Sig.KG}_pp(), x = PK\}, \\
\mathcal{W}_{pp} = \{w \mid w = (w_0, w_1, \ldots, w_7) \in \mathbb{F}_p^3 \times \hat{G} \times \hat{H} \times \hat{G} \times \hat{H} \text{ s.t. (1) and (2) hold for } \exists x \in \mathcal{X}, \exists 0 = m = \tilde{M}, (w_1, \ldots, w_7) = \sigma = (\tilde{Z}, \tilde{R}, \tilde{S}, \tilde{T}, \tilde{U}, \tilde{V}, \tilde{W})\}. 
\]

Then we define the following ternary relation $R_{pp}$.
\[
R_{pp} \overset{\text{def}}{=} \{(ck, x, w) \in \mathcal{CK}_pp \times \mathcal{X}_{pp} \times \mathcal{W}_{pp} \mid w \text{ can be committed by $\text{Cmt.Com}_pp$ under $ck$ and (1) and (2) hold for } (x, w)\}.
\]

A group-dependent language $L_{pp,ck}$ parametrized by $ck \in \mathcal{CK}$ is defined as follows.
\[
L_{pp,ck} \overset{\text{def}}{=} \{x \in \mathcal{X}_{pp} \mid \exists w \in \mathcal{W}_{pp} \text{ s.t. } (ck, x, w) \in R_{pp}\}.
\]
We put \( pp := (pp, ck) \) because the commitment key \( ck \) is treated as a public parameter.

- \( P_{pp}(x, c, w, r) \rightarrow \pi \). 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 \( \pi \) (see [EG14] for the details). It returns \( \pi \).
- \( V_{pp}(x, c, \pi) \rightarrow d \). On input a statement \( x \), a commitment \( c \) and a proof \( \pi \), this deterministic algorithm executes the verification algorithm of the Groth-Sahai proof system to obtain a boolean decision \( d \) (see [EG14] for the details). It returns \( d \).

The proof-part (\( CmtPrv.Setup, II_{pp} \)) of the Groth-Sahai proof system have the following four properties.

**Definition 5 (Perfectly Correct [GS08])** A commit-and-prove scheme \( CmtPrv \) is said to be perfectly correct if it satisfies the following condition: For any security parameter \( 1^\lambda \), any set of public parameters \( pp \leftarrow CmtPrv.Setup(1^\lambda) \), any commitment key \( ck \leftarrow Cmt.KG_{pp}(\text{nor} \text{ or ext or sim}) \) and any PPT algorithm \( A \),

\[
\Pr[V_{pp}(x, c, \pi) = 1 | (ck, x, w) \in R_{pp} | (x, w) \leftarrow A(pp), (c, r) \leftarrow Cmt.Com_{pp}(w), \pi \leftarrow P_{pp}(x, c, w, r)] = 1.
\]

**Definition 6 (Perfectly Sound [GS08])** A commit-and-prove scheme \( CmtPrv \) is said to be perfectly sound if it satisfies the following condition for some unbounded algorithm \( Cmt.Open_{pp} : \) For any security parameter \( 1^\lambda \), any set of public parameters \( pp \leftarrow CmtPrv.Setup(1^\lambda) \), any commitment key \( ck \leftarrow Cmt.KG_{pp}(\text{nor}) \) and any PPT algorithm \( A \),

\[
\Pr[V_{pp}(x, c, \pi) = 0 \text{ or } (ck, x, w) \in R_{pp} | (x, c, \pi) \leftarrow A(pp), w \leftarrow Cmt.Open_{pp}(c)] = 1.
\]

Let \( C_{ck} \) be the set of commitments under \( ck \) to some message \( w \).

**Definition 7 (Perfectly Knowledge Extractable [GS08])** A commit-and-prove scheme \( CmtPrv \) is said to be perfectly knowledge extractable if it satisfies the following condition for some PPT algorithm \( Cmt.Ext_{pp} : \) For any security parameter \( 1^\lambda \), any set of public parameters \( pp \leftarrow CmtPrv.Setup(1^\lambda) \), any commitment key \( (ck, xk) \leftarrow Cmt.KG_{pp}(\text{ext}) \) and any PPT algorithm \( A \),

\[
\Pr[c \notin C_{ck} \text{ or } Cmt.Ext_{pp}(xk, c) = Cmt.Open_{pp}(c) | c \leftarrow A(pp, ck, xk)] = 1.
\]

**Definition 8 (Composably Witness-Indistinguishable [GS08])** A commit-and-prove scheme \( CmtPrv \) is said to be composably witness-indistinguishable if it satisfies the following condition: For any security parameter \( 1^\lambda \), any set of public parameters \( pp \leftarrow CmtPrv.Setup(1^\lambda) \) and any PPT algorithm \( A \),

\[
\Pr[A(pp, ck) = 1 | ck \leftarrow Cmt.KG_{pp}(\text{nor})] \\
\approx_c \Pr[A(pp, ck) = 1 | (ck, tk) \leftarrow Cmt.KG_{pp}(\text{sim})], \text{ and} \\
\Pr[(ck, x, w), (ck, x, w') \in R_{pp} \text{ and } A(St, \pi) = 1 | (ck, tk) \leftarrow Cmt.KG_{pp}(\text{sim}), pp := (pp, ck), (x, w, w', St) \leftarrow A^{\text{Cmt.Com}_{pp}(\lambda)(pp, ck, tk), (c, r) \leftarrow Cmt.Com_{pp}(w), \pi \leftarrow P_{pp}(x, c, w, r)] = 1] = \Pr[(ck, x, w), (ck, x, w') \in R_{pp} \text{ and } A(St, \pi') = 1 | (ck, tk) \leftarrow Cmt.KG_{pp}(\text{sim}), pp := (pp, ck), (x, w, w', St) \leftarrow A^{\text{Cmt.Com}_{pp}(\lambda)(pp, ck, tk), (c', r') \leftarrow Cmt.Com_{pp}(w'), \pi' \leftarrow P_{pp}(x, c', w', r')].
\]

### 4 Bundled Product of Language

In this section, we define a general notion of a bundled product of a language for the case of the group-dependent language [GS08]. Intuitively, the notion determines a subset of the Cartesian product of a language by the condition that the corresponding witnesses have a fixed number of common components in the former part.
5 Decentralized Multi-authority Anonymous Authentication with Non-interactive Proofs

In this section, we give a syntax and security definitions of a decentralized multi-authority anonymous authentication scheme with non-interactive proofs, which we call NI-DMA-A-AUTH for short. In the security definitions we capture the requirements described in Section 1: One is resistance against collusion attacks that cause misauthentication, and the other is anonymity for privacy protection.

5.1 Syntax

Our scheme \texttt{a-auth} consists of five PPT algorithms, (\texttt{Setup}, \texttt{AuthKG}_{pp}, \texttt{SKG}_{pp}, \texttt{Prover}_{pp}, \texttt{Verifier}_{pp}).

- \texttt{Setup}(1^{\lambda}) \rightarrow pp. This PPT algorithm is needed to generate a set of public parameters \texttt{pp}. On input the security parameter \(1^{\lambda}\), it generates the set \(pp\). It returns \(pp\).
- \texttt{AuthKG}_{pp}(a) \rightarrow (PK^a, MSK^a). This PPT algorithm is executed by a key-issuing authority indexed by a positive integer \(a\). 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 \((PK^a, MSK^a)\).
- \texttt{SKG}_{pp}(PK^a, MSK^a, gid) \rightarrow sk^a_{gid}. This PPT algorithm is executed by the \(a\)-th key-issuing authority. On input the \(a\)-th public and master secret keys \((PK^a, MSK^a)\) and an element \(gid \in G\) of a prover, it generates a private secret key \(sk^a_{gid}\) of a prover. It returns \(sk^a_{gid}\).
- \texttt{Prover}_{pp}((PK^a, sk^a_{gid})_{a \in A'}) \rightarrow \pi. This PPT algorithm is executed by a prover who is to be authenticated, where \(A'\) denotes a subset of all indices at which the prover is issued her private secret keys by authorities. On input the public keys \((PK^a)_{a \in A'}\) and the corresponding private secret keys \((sk^a_{gid})_{a \in A'}\), it returns a proof \(\pi\).
- \texttt{Verifier}_{pp}((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 \((PK^a)_{a \in A'}\) and the proof \(\pi\), it returns \(d := 1\) ("accept") or \(d := 0\) ("reject").

5.2 Security Definitions

We discuss two security notions for our authentication scheme \texttt{a-auth}: security against collusion attacks that yield misauthentication, and anonymity for privacy of provers’ global identities.
Resistance against Concurrent and Collusion Attack of Misauthentication. One of the strongest attacks to cause misauthentication is the concurrent and collusion attack on our a-auth. For a formal treatment we define the following experiment on a-auth and an adversary algorithm $A$.

$$\text{Expr}_{a\text{-auth}, A}(1^\lambda) :$$

$$(q_A, St) \leftarrow A(1^\lambda), A := \{1, \ldots, q_A\}, pp \leftarrow \text{Setup}(1^\lambda)$$

For $a \in A : (PK^a, MSK^a) \leftarrow \text{AuthKG}_{pp}(a)$

$$((\text{gid}_j)_{i=1}^{q_I}, St) \leftarrow A(St, pp, (PK^a)_{a \in A}), I := \{1, \ldots, q_I\}$$

For $a \in A : I : i \in I : sk_{\text{gid}_i}^a \leftarrow \text{SKG}_{pp}(PK^a, MSK^a, \text{gid}_i)$

$$(A^*, \pi^*) \leftarrow A^{\text{Prover}_{pp}((PK^a, sk_{\text{gid}_j}^a))_{j \in J}, \text{SKO}_{pp}(PK^a, MSK^a, \cdot)}(St)$$

$$\text{Verifier}_{pp}((PK^a)_{a \in A^*}, \pi^*) \rightarrow d$$

If $d = 1$ then return Win else return Lose

Intuitively, the above experiment describes the attack as follows. The adversary algorithm $A$, on input the security parameter $1^\lambda$, first outputs the number $q_A$ of key-issuing authorities. Then, on input the set of public parameters $pp$ and the issued public keys $(PK^a)_{a \in A}$, $A$ outputs global identity element $\text{gid}_i \in \mathcal{G}, i = 1, \ldots, q_I$. $A$ invokes prover algorithm $\text{Prover}_{pp}$ with $\text{gid}_i, i = 1, \ldots, q_I$ to obtain proofs. In addition, $A$ collects at most $q_A$ private secret keys by issuing queries to the private secret key oracle $\text{SKO}_{pp}(PK^a, MSK^a, \cdot)$ with an authority index $a \in A$ and a global identity element $\text{gid}_j \in \mathcal{G}$ for $j = q_I + 1, \ldots, q_I + q_{sk}$. We denote by $A_j$ the set of authority indices for which the queries with $\text{gid}_j$ were issued. That is,

$$A_j := \{a \in A \mid \text{A receives } sk_{\text{gid}_j}^a\}, j = q_I + 1, \ldots, q_I + q_{sk}.$$

We here require that the numbers $q_A, q_I$ and $q_{sk}$ are bounded by a polynomial in $\lambda$. At the end $A$ returns a target set of authority indices and a forgery proof $(A^*, \pi^*)$. If the decision $d$ on $\pi^*$ by $\text{Verifier}_{pp}$ is 1 under $(PK^a)_{a \in A^*}$, then the experiment returns Win; otherwise it returns Lose.

A restriction is imposed on the adversary $A$: The target set of authority indices $A^*$ should not be a subset of any single set $A_j$:

$$A^* \not\subseteq A_j, j = q_I + 1, \ldots, q_I + q_{sk}. \quad (8)$$

This restriction is because otherwise, $A$ is given private secret keys for $A^*$ on a single $\text{gid}_j, \ast$ for some $j^*$, $q_I < j^* \leq q_I + q_{sk}$, and then $A$ can trivially succeed in causing misauthentication.

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

$$\text{Adv}_{a\text{-auth}, A}(\lambda) \equiv \Pr[\text{Expr}_{a\text{-auth}, A}(1^\lambda) = \text{Win}].$$

A scheme $a\text{-auth}$ is called secure against concurrent and collusion attacks that cause misauthentication. if, for any PPT algorithm $A$, the advantage $\text{Adv}_{a\text{-auth}, A}(\lambda)$ is negligible in $\lambda$.

Anonymity. As is explained in Section 1, a critical feature to be attained is provers’ anonymity on global identities when the provers are authenticated. Formally we define the following experiment on $a\text{-auth}$ and an adversary algorithm $A$.

$$\text{Expr}_{a\text{-auth}, A}(1^\lambda) :$$

$$(q_A, St) \leftarrow A(1^\lambda), A := \{1, \ldots, q_A\}, pp \leftarrow \text{Setup}(1^\lambda)$$

For $a \in A : (PK^a, MSK^a) \leftarrow \text{AuthKG}_{pp}(1^\lambda, a)$

$$(\text{gid}_0, \text{gid}_1, St) \leftarrow A(St, pp, (PK^a)_{a \in A})$$

For $a \in A : I : i \in I : sk_{\text{gid}_i}^a \leftarrow \text{SKG}_{pp}(PK^a, MSK^a, \text{gid}_i)$

$$b \in R \{0,1\}, b^* \leftarrow A^{\text{Prover}_{pp}((PK^a, sk_{\text{gid}_j}^a))_{j \in A}}(St, (sk_{\text{gid}_0}^a, sk_{\text{gid}_1}^a)_{a \in A})$$

If $b = b^*$, then return Win, else return Lose.
In this section, we construct a scheme of NI-DMA-A-AUTH. We employ two building blocks. One is the structure-preserving signature scheme [AFG10, AFG16] (see Section 2.2). Each decentralized authority indexed by ‘a’ issues a private secret key sk^a_gid for a global identity element gid. The other building block is the non-interactive commit-and-prove scheme of the fine-tuned Groth-Sahai proof system [GS08, EG14] adapted to the case of the structure-preserving signature (see Section 3). In the commit-phase a prover generates commitments to the global identity element gid and the components of the structure-preserving signatures (σ^1_gid, ..., σ^n_gid). In the prove-phase the prover generates a proof π using the bundled witnesses. That is, w_0 = gid is the common component, and for (w^1, ..., w^n) = (σ^1_gid, ..., σ^n_gid), (w_0, w^1, ..., w^n) is a whole witness of the a-th component. The proof π is a proof for the bundled product of the language (see Section 4).

6 Construction and Security Proofs of NI-DMA-A-AUTH

According to the syntax in Section 5, the scheme a-auth consists of five PPT algorithms: a-auth = (Setup, AuthKG_{pp}, SKG_{pp}, Prover_{pp}, Verifier_{pp}).

• Setup(λ) → pp. On input the security parameter λ, it runs the generation algorithm of bilinear groups, and it puts the output as a set of public parameters: (p, 1^λ, H, T, e, G, H) := pp. Note that pp is a 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: Cmt.KG_{pp}(noz) → ck. It returns pp := (pp, ck).

• AuthKG_{pp}(a) → (PK^a, MSK^a). On input an authority index a, it executes the key-generation algorithm Sig.KG_{pp}() to obtain (PK, SK). It puts PK^a := PK and MSK^a := SK. It returns (PK^a, MSK^a).

• SKG_{pp}(PK^a, MSK^a, gid) → sk^a_gid. On input PK^a, MSK^a and an element gid ∈ H, it puts PK^a := PK^a and SK^a := MSK^a and m := M := gid. It executes the signing algorithm Sig.Sign_{pp}(PK^a, SK^a, m) to obtain a signature σ^a. It puts sk^a_gid := (gid, σ^a). It returns sk^a_gid.

• Prover_{pp}((PK^a, sk^a_gid)_{a ∈ A'}) → π. On input (PK^a, sk^a_gid)_{a ∈ A'}, first, it commits to gid:

   c_0 ← Cmt.Com_{pp}(gid, r_0).

Second, for each a ∈ A', it commits to the components σ^a_1, ..., σ^a_n of the signature σ^a in the componentwise way.

   (c^a_1, ..., c^a_n) ← Cmt.Com_{pp}((σ^a_1, ..., σ^a_n); (r^a_1, ..., r^a_n)).

Then, for each a ∈ A', it puts a := (G^a_z, G^a_{u,z}, G^a_u, G^a_{u,1}, (A^a_i, (A^a_i)^{-1}, B^a_i, (B^a_i)^{-1})_{i=0}^l) by using PK^a. It also puts c^a := (c^a_1, ..., c^a_n), w^a := (w_0, w^a_1, ..., w^a_n) := (gid, σ^a_1, ..., σ^a_n) and r^a := (r_0, r^a_1, ..., r^a_n). It executes the prove-algorithm to obtain a proof:

   π^a ← P_{pp}(a^a, c^a, w^a, r^a), a ∈ A'.
It puts $\tilde{\pi}^a := ((c_1^a, \ldots, c_2^a), \pi^a)$ for each $a \in A'$, and it merges all the $\tilde{\pi}^a$s and the single commitment $c_0$ to $\text{gid}$: $\pi := (c_0, (\tilde{\pi}^a)_{a \in A'})$. It returns $\pi$.

- **Verifier**$_{pp}((PK^a)_{a \in A'}, \pi) \rightarrow d$. On input $(PK^a)_{a \in A'}, \pi)$, it converts $PK^a$ into $x^a$ and it puts $c^a := (c_0, c_1^a, \ldots, c_2^a)$ for each $a \in A'$. Then it executes the verify-algorithm for each $a \in A'$ to obtain the decisions:

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

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

### 6.2 Security Proofs

**Theorem 1 (Resistance against Concurrent and Collusion Attacks)** For any PPT algorithm $A$ that is in accordance with the experiment $\text{Expr}^{\text{conc-coll}}_{a, \text{auth}}(1^\lambda)$, there exists a PPT algorithm $F$ that is in accordance with the experiment $\text{Exp}s^{\text{euf-cma}}_{\text{Sig}, F}(1^\lambda)$ and the following inequality holds.

$$\text{Adv}^{\text{conc-coll}}_{a, \text{auth}, A}(\lambda) \leq \frac{p}{p - q_I - q_{sk}} \cdot q_A \cdot \text{Adv}^{\text{euf-cma}}_{\text{Sig}, F}(\lambda).$$

This theorem means that, if the structure-preserving signature scheme Sig is existentially unforgeable against adaptive chosen-message attacks, then our $a$-auth is secure against concurrent and collusion attacks.

**Proof.** Given any PPT algorithm $A$ that is in accordance with the experiment $\text{Expr}^{\text{conc-coll}}_{a, \text{auth}, A}(1^\lambda)$, we construct a PPT algorithm $F$ that generates an existential forgery of Sig in accordance with the experiment $\text{Exp}s^{\text{euf-cma}}_{\text{Sig}, F}(1^\lambda)$. $F$ is given as input the set of public parameters $pp$ and a public key $PK$. $F$ executes $\text{Cmt.KG}_{pp}(\text{ext})$ to obtain a pair $(ck, xk)$. $F$ puts $pp := (pp, ck)$. $F$ invokes the algorithm $A$ with $1^\lambda$ to obtain the number $q_A$ of authorities. $F$ chooses a target index $a^*$ from the set $A := \{1, \ldots, q_A\}$ uniformly at random. $F$ runs the authority key generation algorithm honestly for $a \in A$ except the target index $a^*$. As for $a^*$, $F$ uses the input public key:

For $a \in A$ s.t. $a \neq a^*$: $(PK^a, MSK^a) \leftarrow \text{AuthKG}_{pp}(a),

For a = a^*: PK^* := PK.$

$F$ inputs $pp$ and $(PK^a)_{a \in A}$ into $A$ to obtain the number $q_I$ of concurrent provers. $F$ puts $I := \{1, \ldots, q_I\}$.

**Simulation of Concurrent Provers.** When $A$ invokes a prover with $\text{gid}_i \in G$, $i = 1, \ldots, q_I$, $F$ runs the generation algorithm of a private secret key with $\text{gid}_i$ honestly for $a \in A$ except the target index $a^*$. As for $a^*$, $F$ issues a signing query with $\text{gid}_i$ to its oracle:

For $a \in A$ s.t. $a \neq a^*$: $sk^a_{\text{ gid}_i} \leftarrow \text{SKG}_{pp}(PK^a, MSK^a, \text{gid}_i),

For a = a^*: sk^a_{\text{ gid}_i} \leftarrow \text{SignO}_{pp}(PK, SK, \text{gid}_i).$

In the simulation of concurrent provers $\text{Prover}_{pp}((PK^a, sk^a_{\text{ gid}_i})_{a \in A})|_{i \in I}$. $F$ uses the private secret keys $(sk^a_{\text{ gid}_i})_{a \in A}, i \in I$. Note that this is a perfect simulation.

**Simulation of Private Secret Key Oracle.** When $A$ issues a private secret key query with $A_j \subset A$ and $\text{gid}_j \in G$, $j = q_I + 1, \ldots, q_1 + q_{sk}$, $F$ runs the generation algorithm of private secret key with $\text{gid}_j$ honestly for $a \in A$ except the target index $a^*$. As for $a^*$, $F$ issues a signing query with $\text{gid}_j$ to its oracle:

For $a \in A$ s.t. $a \neq a^*$: $sk^a_{\text{ gid}_j} \leftarrow \text{SKG}_{pp}(PK^a, MSK^a, \text{gid}_j),

For a = a^*: sk^a_{\text{ gid}_j} \leftarrow \text{SignO}_{pp}(PK, SK, \text{gid}_j).$
Claim 4

A This is because of the perfect knowledge extraction of Proof.

Proof.

Claim 5

For the left-hand side of the equality (10) is expanded as follows.

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

(9)

\[
\Pr[\text{Acc, Ext, TGTIDX, NEWID}] = \Pr[\text{Forge}],
\]

(10)

\[
\Pr[\text{Forge}] = \text{Adv}^{\text{euf-cma}}_{\text{Sig,F}}(\lambda).
\]

(11)

The left-hand side of the equality (10) is expanded as follows.

\[
\Pr[\text{TGTIDX}] = 1/q_A.
\]

(13)

Proof. The restriction (8) assures that there exists an authority index \( \hat{a} \) such that \( \hat{a} \in A^* \) and \( \hat{a} \notin A_j \) for \( j = q_l + 1, \ldots, q_l + q_{sk} \). \( \hat{a} \) coincides with \( a^* \) with probability \( 1/q_A \).

Claim 4

\[
\Pr[\text{NEWID | Acc, Ext}] \geq \frac{p - q_l - q_{sk}}{p}.
\]

(14)

Proof. \( \text{gid}^* \) is not in \( \{\text{gid}_i\}_{i=1}^{q_l+q_{sk}} \) with probability at least \( \frac{p - q_l - q_{sk}}{p} \).

Claim 5

\[
\Pr[\text{Ext | Acc}] = 1.
\]

(15)

Proof. This is because of the perfect knowledge extraction of \( \Pi_{pp} \).
Combining (9), (10), (11), (12), (13), (14) and (15) we have:

\[
\text{Adv}_{\text{a-auth}, A}^{\text{conc-coll}}(\lambda) \leq \frac{p}{p - q_1 - q_{sk}} \cdot q_A \cdot \text{Adv}_{\text{Sig}_F}^{\text{euf-cma}}(\lambda).
\]

□

**Theorem 2 (Anonymity)** Assuming the computational indistinguishability between commitment keys \(\{ck\}\) of the mode \(\text{nor}\) and commitment keys \(\{ck\}\) of the mode \(\text{sim}\), the following equality holds for any PPT algorithm \(A\) that is in accordance with the experiment \(\text{Expr}_{\text{a-auth}, A}^{\text{ano}}(1^\lambda)\).

\[
\text{Adv}_{\text{a-auth}, A}^{\text{ano}}(\lambda) = 0.
\]

**Proof.** Suppose that any PPT algorithm \(A\) that is in accordance with the experiment \(\text{Expr}_{\text{a-auth}, A}^{\text{ano}}(1^\lambda)\) is given. When a set of public parameters \(pp = (pp', ck)\) is given to \(A\), where \(pp'\) is for bilinear groups, the commitment key \(ck\) can be chosen as a commitment key \(ck\) of the mode \(\text{sim}\) under the assumption of the computational indistinguishability (4). Then the rest of values in \(\text{Expr}_{\text{a-auth}, A}^{\text{ano}}(1^\lambda)\) distribute identically for both \(\text{gid}_0\) and \(\text{gid}_1\) due to the perfectly hiding property (5) and the witness-indistinguishability (6). Therefore, \(\text{Adv}_{\text{a-auth}, A}^{\text{ano}}(\lambda) = 0\) holds.

□

7 Conclusion

We gave a NI-DMA-A-AUTH scheme, in which a prover is able to convince a verifier that a single anonymous prover has the knowledge of plural attribute certificates issued by independent authorities. Perfect binding property of the commitment to \(\text{gid}\) works as a proof of simultaneous satisfiability of the verification equations of structure-preserving signatures. Hence the collusion attacks are prevented. On the other hand, perfectly hiding property of commitments and perfect witness-indistinguishable property of proofs of the Groth-Sahai proof system yields anonymity, and hence, assures privacy. Other structure-preserving signature schemes such as [AHN17] can be employed instead of that of [AFG10,AFG16].

References


