Bulletin of Networking, Computing, Systems, and Software – www.bncss.org, ISSN 2186-5140
Volume 14, Number 1, pages 27–33, January 2025

Issuer-Hiding Attribute-Based Credentials with Constant-Size Attribute Proofs for CNF Formulas with Negations

Yuta Hamada*, Toru Nakanishi*, and Teruaki Kitasuka*
* Graduate School of Advanced Science and Engineering, Hiroshima University, Higashi-Hiroshima 739-8527, Japan

***Abstract*—Attribute-based credential (ABC) systems allow a user to anonymously prove attributes of the user to a verifier. In advance, the user is issued a credential of the attributes from an issuer. The user can select disclosed attributes, and prove the possession of the attributes without revealing other unnecessary attributes. However, the issuer's public key is required in the verification of the attribute proof, and thus the verifier can know who is the issuer issuing the credential. An issuer-hiding ABC system was proposed, where the issuer can be hidden in the attribute proof, and the issuer's attributes can be proved. Thus, the verifier can verify the anonymous issuer flexibly based on the issuer's attributes. In this paper, we propose an issuer-hiding ABC system by combining the previous system and an accumulator. In the proposed system, CNF formulas with negations can be used as the attribute proof. Since the attributes of the user and the issuer are accumulated, the constant proof size and verification time are achieved. We show the practicality by an implementation on a PC.**

***Index Terms*—attribute-based credential (ABC), issuer-hiding, pairing, accumulator.**

I. Introduction

An attribute-based credential (ABC) system that is an extension of anonymous credential system [3], [4] allows a user to anonymously prove attributes of the user (e.g., age, name, and affiliation) to a verifier. In advance, the user needs to obtain the credential of the attributes from an issuer. The user can select disclosed attributes, and prove the possession of the attributes without revealing other unnecessary attributes.

A problem of the conventional ABC systems is that the public key of the issuer is required in the verification of the attribute presentation proof, and thus the verifier can know who is the issuer issuing the credential. If the attributes of the user are relevant to the issuer, the verifier can presume the user information through the public key of the issuer (i.e., who is the issuer). For example, when the issuer is a university, the verifier can know that the user belongs to the university.

As a solution of this problem, an *issuer-hiding* ABC system was proposed [1]. In this system, each verifier defines a set of accepted issuers, and sends users signatures of the issuers' public keys. The public key of the issuer issuing the proved credential can be concealed in the attribute presentation proof by a zero-knowledge proof on the signature, while it is ensured that the issuer is included in the set of the accepted issuers. Thus, the above-mentioned problem that the issuer's information is revealed is solved.

In [6], an extended issuer-hiding ABC system was proposed, where the issuer's attributes can be proved, in addition to the user's attributes. For example, suppose that an issuer is ranked in the top 100 in a certain industry. The user can selectively disclose the attributes that indicate that the issuer has the “Top 100” attributes in “a certain industry” and keep the rest of the attributes secret. In this system, each verifier defines a set of accepted issuers with the attributes of the issuers, and sends users signatures, called policy signatures, for the set. The user can prove selected attributes of the user and the issuer. Thus, the verifier can verify the anonymous issuer flexibly based on the issuer's attributes. In the previous issuer-hiding ABC systems [1], [6], only the selective disclosure is available. However, in conventional ABC systems, more complex relations on attributes can be proved. In [7], by using an accumulator to verify a CNF (Conjunctive Normal Form) formula with negations, an ABC system was constructed, where a user can prove such a CNF formula on attributes. For example, consider the authentication to verify the absence of the specified attribute for the evaluation of the organization of the attribute. The ABC system can address such an authentication, in addition to the authentications of AND and OR relations on attributes.

In this paper, we propose an issuer-hiding ABC system by combining the previous system [6] and the accumulator [7]. In the previous system [6], Groth's structure-preserving signature [5] is used as a credential and a policy signature, and the verification can be proved by a Signature based on Proof of Knowledge (SPK), where disclosed attributes are public values and hidden attributes are secret values. In the proposed system, the accumulator [7] to verify CNF formulas is used as the attribute proof, and attributes of the user and the issuer are accumulated. In the user's attribute presentation protocol, SPK-based attribute proofs for the verification of accumulators are performed, where values for the accumulator verification are blinded. The verification relations do not depend on the numbers of the attributes of the user and the issuer and the CNF formula size, and the constant proof size and verification cost are achieved. We show the practicality by an implementation on a PC.

II. Preliminaries

A. Bilinear Groups

In this paper, we use bilinear groups with a bilinear map. Let $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ be cyclic groups of the same prime order

p. Let G and $\tilde{G}$ be generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. The bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ satisfies the following bilinearity and non-degeneracy.

- **bilinearity:** For any $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$, and $a, b \in \mathbb{Z}_p$, $e(P^a, Q^b) = e(P, Q)^{ab}$.
- **non-degeneracy:** $e(G, \tilde{G})$ is not the identity element in $\mathbb{G}_T$.

The above bilinear map can be realized by a pairing on elliptic curves.

B. Assumption

The security of our system is based on n-DHE (DH Exponent) assumption [2] on the asymmetric type of bilinear map.
Definition 1 (n-DHE assumption). For any PPT algorithm $\mathcal{A}$, the probability

$$Pr\left[\mathcal{A}\begin{pmatrix} G, G^{a}, \ldots, G^{a^n}, G^{a^{n+2}}, \ldots, G^{a^{2n}}, \\ \tilde{G}, \tilde{G}^{a}, \ldots, \tilde{G}^{a^n}, \tilde{G}^{a^{n+2}}, \ldots, \tilde{G}^{a^{2n}} \end{pmatrix} = \tilde{G}^{a^{n+1}}\right]$$

is negligible, where $a \xleftarrow{\$} \mathbb{Z}_p$.

C. Signatures Based on Proofs of Knowledge (SPKs)

We adopt signatuers based on proofs of knowledge (SPKs). A zero-knowledge proof of knowledge is an interactive protocol between a prover P and a verifier V, where P proves secrets satisfying certain relations without revealing the secrets. In this paper, we use proofs of knowledge for discrete logarithms. The proof of knowledge can be transformed to the corresponding non-interactive SPK via Fiat-Shamir heuristic by applying a hash fanction to the random challenge and a signed message.

D. Structure-Preserving Signatures (Groth Signatures)

In this paper, we utilize Groth signatures [5]. Groth signatures are randomizable structure-preserving signatures to sign multiple group elements, and the verification on the bilinear map can be proved using the SPKs. In this scheme, there are two variants: $\mathbf{Groth}_1$ signs $\mathbb{G}_1$-elements and the public key is a $\mathbb{G}_2$-element. Also, $\mathbf{Groth}_2$ signs $\mathbb{G}_2$-elements and the public key is a $\mathbb{G}_1$-element. The algorithms of $\mathbf{Groth}_1$ for n messages are as follows.

- $\mathbf{Groth}_1.\mathbf{ParGen}(1^\lambda)$: Generate a bilinear group parameters $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, \tilde{G})$ with $p \geq 2^\lambda$. Set public parameters $pp = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, \tilde{G}, \{Y_i\}_{i=1}^n)$ with random elements $Y_i \xleftarrow{\$} \mathbb{G}_1$ for all $i \in [1, n]$.
- $\mathbf{Groth}_1.\mathbf{KGen}(pp)$: Randomly select $sk \xleftarrow{\$} \mathbb{Z}_p$ and compute $pk = \tilde{G}^{sk}$. Output secret key and public key (sk, pk).
- $\mathbf{Groth}_1.\mathbf{Sign}(pp, sk, \{M_i\}_{i=1}^n)$: Calculate a $\mathbf{Groth_1}$ signature on the messages $\{M_i\}_{i=1}^n$ as the credential $\sigma = (\tilde{R}, S, \{T_i\}_{i=1}^n) = (\tilde{G}^r, (Y \cdot G^{sk})^{1/\gamma}, \{(Y_i^{sk} \cdot M_i)^{1/\gamma}\}_{i=1}^n)$ for $\gamma \xleftarrow{\$} \mathbb{Z}_p^*$.
- $\mathbf{Groth}_1.\mathbf{Rand}(pp, \sigma)$: Given $\mathbf{Groth_1}$ signature $\sigma = (\tilde{R}, S, \{T_i\}_{i=1}^n)$, and compute and output $\sigma' = (\tilde{R}^{\gamma'}, S^{1/\gamma'}, \{T_i^{1/\gamma'}\}_{i=1}^n)$ for $\gamma' \xleftarrow{\$} \mathbb{Z}_p^*$.
- $\mathbf{Groth}_1.\mathbf{Verify}(pp, pk, \sigma, \{M_i\}_{i=1}^n)$: Given $\sigma = (\tilde{R}, S, \{T_i\}_{i=1}^n)$, verify the validity by $e(S, \tilde{R}) = e(Y, \tilde{G}) \cdot e(G, pk)$ and $e(T_i, \tilde{R}) = e(Y_i, pk) \cdot e(M_i, \tilde{G})$ for $1 \leq i \leq n$.

$\mathbf{Groth_2}$ is obtained by switching the roles of $\mathbb{G}_1$-elements and $\mathbb{G}_2$-elements.
This scheme is EUF-CMA secure in the generic group model [5].

E. Accumulator to Verify CNF Formulas with Negations

In this paper, we use a pairing-based accumulator. In the accumulator, a set of elements is accumulated to a single value called an accumulator, and one can prove that an element is included in the set. In [7], an extended accumulator was proposed, where a CNF formula with negations can be verified. Consider a CNF formula $\Psi = \wedge_l \vee_j \breve{a}_{lj}$ where $\breve{a}_{lj}$ is a literal of a non-negated attribute a_{lj} or a negated attribute $\neg a_{lj}$. Let V_l^+ (resp. V_l^-) be a set of a_{lj}'s ($\neg a_{lj}$'s). Let U be a set of attributes. Then, the formula Ψ is accumulated, and it can be verified that Ψ is satisfied by U. Each attribute is an index in set of $\{1, \ldots, n\}$, and U, V_l^+, V_l^- are subsets of $\{1, \ldots, n\}$. When $U \cap V_l^+ \neq \emptyset$ or $U \cap V_l^- \neq V_l^-$ for all $1 \leq l \leq L$, a user with the attributes of U owns an attribute in V_l^+ or does not own an attribute in V_l^- for all $1 \leq l \leq L$, i.e., the CNF formula Ψ is satisfied. The algorithms are as follows.

- $\mathbf{AccSetup}(1^\lambda, L, \{\eta_l\}_{1 \leq l \leq L})$: Given security parameter λ, the number L of clauses of CNF formulas, and η_l as the maximum value of $|V_l^+ \cup V_l^-|$ for all $1 \leq l \leq L$. Generate bilinear group parameters $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, \tilde{G})$ with $p \geq 2^\lambda$. Set $c_1 = 1$, $c_l = (\eta_{l-1} + 1) \cdot c_{l-1}$ $(2 \leq l \leq L)$, and $\mathcal{C} = (c_1, \ldots, c_L)$. Choose $\gamma \xleftarrow{\$} \mathbb{Z}_p$ and compute $G_i = G^{\gamma^i}$ for all $1 \leq i \leq 2n$ except $i = n + 1$ (resp. $\tilde{G}_i = \tilde{G}^{\gamma^i}$ for the same condition), and $z = e(G, \tilde{G})^{\gamma^{n+1}}$. Output public parameters $pp = (\mathcal{C}, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, \{G_i\}_{i=1, i \neq n+1}^{2n}, \tilde{G}, \{\tilde{G}_i\}_{i=1, i \neq n+1}^{2n}, z)$.
- $\mathbf{AccGen}(pp, \mathcal{V})$: For $1 \leq l \leq L$, $V_l^+ \subseteq \{1, \ldots, n\}$ is the set of non-negated attributes in the l-th OR clause, $V_l^- \subseteq \{1, \ldots, n\}$ is the set of negated attributes, and $\mathcal{V} = (V_1^+, V_1^-, \ldots, V_L^+, V_L^-)$. Calculate an accumulator $acc_{\mathcal{V}}$ as follows.

$$acc_{\mathcal{V}} = \prod_{1 \leq l \leq L} \Big(\prod_{j \in V_l^+} \tilde{G}_{n+1-j}\Big)^{c_l} \Big(\prod_{j \in V_l^-} \tilde{G}_{n+1-j}\Big)^{-c_l}$$

- $\mathbf{AccWitGen}(pp, \mathcal{V}, U)$: $U \subseteq \{1, \ldots, n\}$ is the set of attributes. Calculate the witness W as follows.

$$W = \prod_{i \in U} \prod_{1 \leq l \leq L} \Big(\prod_{j \in V_l^+}^{j \neq i} \tilde{G}_{n+1-j+i}\Big)^{c_l} \Big(\prod_{j \in V_l^-}^{j \neq i} \tilde{G}_{n+1-j+i}\Big)^{-c_l}$$

  Output the witness W, and $\delta_l = |U \cap V_l^+| - |U \cap V_l^-|$ for all $1 \leq l \leq L$ as auxiliary parameters.
- $\mathbf{AccVerify}(pp, \mathcal{V}, acc_{\mathcal{V}}, U, W, \{\delta_l\}_{1 \leq l \leq L})$: Verify $U \cap V_l^+ \neq \emptyset$ or $U \cap V_l^- \neq V_l^-$ for all $1 \leq l \leq L$ as follows.

Set $u = \delta_1 c_1 + \ldots + \delta_L c_L$. Accept if the following relations hold.

$$\frac{e(\prod_{i \in U} G_i, acc_{\mathcal{V}})}{e(G, W)} = z^u, \text{ and}$$

$$1 \leq \delta_l + |V_l^-| \leq \eta_l \text{ for all } 1 \leq l \leq L$$

The following security of the accumulator is proved in [7].

Theorem 1. *Under n-DHE assumption, given the public parameters, any PPT adversary cannot output U,* $\mathcal{V} = \{V_l^+, V_l^-\}_{1 \leq l \leq L}$*, W, and* $\{\delta_l\}_{1 \leq l \leq L}$ *which satisfy the following with a non-negligible probability.*

- For $acc_{\mathcal{V}}$ correctly computed from $\mathcal{V}$, **AccVerify** accepts $\mathcal{V}$, $acc_{\mathcal{V}}$, U, W, and $\{\delta_l\}_{1 \leq l \leq L}$.
- For some l, $U \cap V_l^+ = \emptyset$ and $U \cap V_l^- = V_l^-$.

III. Previous Issuer-Hiding ABC System

We review the previous systems [1], [6] of issuer-hiding ABC systems. In the previous system [1], a user obtains a credential on the user's attributes by an issuer, which is a Groth signature ($\mathbf{Groth}_1$) [5] on the attributes where the attributes is accumulated to one group element as $\Pi_{1 \leq i \leq L} H_i^{a_i}$ for public parameter H_i and the i-th attribute a_i. On the other hand, as a policy signature, a verifier sends a Groth signature ($\mathbf{Groth}_2$) on a public key of each accepted issuer. The public key is a public key of $\mathbf{Groth}_1$, i.e., $\mathbb{G}_2$-element, and thus $\mathbf{Groth}_2$ can sign it as a message. In the presentation of a credential, by a zero-knowledge proof on Groth signatures, a user proves that the verifications of the credential and the policy signature, where the public key of the credential is signed by the policy signature, and the disclosed attributes are signed by the credential. In the extended issuer-hiding ABC system [6], issuer's attribute verification is added. In addition to the issuer public key, the attributes are signed in the policy signature, where the attributes are accumulated as well as the user's attributes. In the attribute presentation proof, selected attributes are proved in the similar way to [1].

IV. Model

We show the model of our issuer-hiding ABC system.

A. Syntax

The algorithms of our issuer-hiding ABC system are as follows.

- $\mathbf{ParGen}(1^\lambda, L, n, \{\eta_l\}_{1 \leq l \leq L})$: Given a security parameter 1^λ, L as the number of clauses of CNF formulas, n as the total number of attributes, and η_l as the upper bound of $|V_l^+ \cup V_l^-|$ for all $1 \leq l \leq L$, it outputs public parameter pp.
- $\mathbf{IKGen}(pp)$: Given the public parameter pp, it outputs an issuer's secret and public key (isk, ipk).
- $\mathbf{UKGen}(pp)$: Given the public parameter pp, it outputs an user's secret and public key (usk, upk).
- $\mathbf{Issue}(pp, isk, U, upk)$: Given the issuer's secret key isk, the set $U \subseteq \{1, \ldots, n\}$ of a user's attributes, and the user's public key upk, it outputs a credential $cred$ on the attributes and upk.
- $\mathbf{VfCred}(pp, cred, ipk, U, upk)$: Given the credential $cred$, the issuer's public key ipk, the set U of the user's attributes, and the user's public key upk, it outputs the validity of the credential.
- $\mathbf{PresPolicy}(pp, \{(ipk_j, I_j)\})$: Given a set of accepted issuer's public key ipk_j and its attributes set $I_j \subseteq \{1, \ldots, n\}$, it outputs a policy pol that consists of a verifier's public key vpk and a set of (ipk_j, I_j, σ_j), where σ_j is a policy signature on ipk_j and I_j with vpk.
- $\mathbf{VfPolicy}(pp, pol, \{(ipk_j, I_j)\})$: Given the policy pol, the set of issuer's public key and its attributes, it outputs the validity of the policy.
- $\mathbf{Present}(pp, cred, ipk, U, I, usk, pol, \psi_U, \psi_I, \mathbf{ctx})$: Given the credential $cred$, the issuer's public key ipk, the user's attributes set U, the issuer's attributes set I, the user's secret key usk, the policy pol, CNF formulas ψ_U, ψ_I on the user's and issuer's attributes, and $\mathbf{ctx}$ to define a context where the present protocol is accepted (i.e., sessoin ID or a nonce), it outputs a presentation token pt.
- $\mathbf{Verify}(pp, pt, pol, \psi_U, \psi_I, \mathbf{ctx})$: Given the token pt, the policy pol, the formulas ψ_U, ψ_I, and $\mathbf{ctx}$, it outputs the validity of the token.

B. Security

Security requirements for our issuer-hiding ABC system are defined, which are derived from the original paper [1], as follows.

- *Correctness.* Correctness requires that if every party follows the protocols, any presentation token is accepted by the verifier.
- *Unforgeability.* Unforgeability requires that it is infeasible for any adversary to generate a valid proof when the adversary does not receive any credential on the disclosed user's and issuer's attributes from one of accepted issuers.
- *Unlinkability.* Unlinkability requires that any adversary cannot determine whether any two presentation tokens are generated by the same user, which implies stronger anonymity.

V. Proposed System

We construct our proposed system by combining the previous system [6] and the accumulator [7] to verify CNF formulas with negations with the constant-size attribute proofs. In the previous system, attributes of a user and an issuer are represented as vectors, such as $(a_1, \cdots, a_L) \in \mathbb{Z}_p^L$, and an accumulated value of $\Pi_{i=1}^L H^{a_i}$ is signed as a credential. As the signature scheme, Groth's structure-preserving signature [5] is used, where the verification can be proved by an SPK.

In the proposed system, using the accumulator to verify CNF formulas, attributes of a user are accumulated to $P_U = \Pi_{l \in U} G_l$ for the set U of the attributes, and the attributes of the issuer are accumulated similarly. Groth's structure-preserving signatures are used as in the previous system [6], and for signatures required for the verification of accumulators, we use Groth's signatures instead of the AHO signatures [2] used in the previous system [7]. In the user's attribute presentation protocol, SPK-based attribute proofs using accumulators are performed, where values for the accumulator verification are blinded as in the previous system [1], and the SPK on the blinded values is constructed.

A. Proposed Algorithm

ParGen$(1^\lambda, n, L, \{\eta_l\}_{1\leq l\leq L})$.
Given n that is the total number of attribute values, L that is the maximum value of clauses of proved CNF formulas, and η_l that are the upper bound of $|V_{U,l}^+ \cup V_{U,l}^-|$ and $|V_{I,l}^+ \cup V_{I,l}^-|$.

(i) Generate a bilinear map parameters $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, \tilde{G})$.

(ii) Set parameters of the accumulator as follows. Calculate $c_1 = 1, c_l = (\eta_{l-1} + 1) \cdot c_{l-1}$ for $2 \leq l \leq L$, and set $\mathcal{C} = (c_1, ..., c_L)$. Choose $\gamma \overset{\$}{\leftarrow} \mathbb{Z}_p$, and calculate and set $pk_{acc} = (\mathcal{C}, G_1 = G^{\gamma^1}, \ldots, G_n = G^{\gamma^n}, G_{n+2} = G^{\gamma^{n+2}}, \ldots, G_{2n} = G^{\gamma^{2n}}, \tilde{G}_1 = \tilde{G}^{\gamma^1}, \ldots, \tilde{G}_n = \tilde{G}^{\gamma^n}, \tilde{G}_{n+2} = \tilde{G}^{\gamma^{n+2}}, \ldots, \tilde{G}_{2n} = \tilde{G}^{\gamma^{2n}}, z = (G, \tilde{G})^{\gamma^{n+1}})$.

(iii) For Groth signatures, select $sk_{Groth_1} \overset{\$}{\leftarrow} x$ and calculate $pk_{Groth_1} = \tilde{G}^x$ with $\mathbf{Groth}_1.\mathbf{KGen}$, select $sk_{Groth_2} \overset{\$}{\leftarrow} y$ and calculate $pk_{Groth_2} = G^y$ with $\mathbf{Groth}_2.\mathbf{KGen}$. Select $Y_1, Y_2 \overset{\$}{\leftarrow} \mathbb{G}_1$, $\tilde{Y}_1$ and $\tilde{Y}_2 \overset{\$}{\leftarrow} \mathbb{G}_2$.

(iv) As in the ABC system [7] using the accumulator, to ensure the range of $\delta'_{U,l} = \delta_{U,l} + |V_{U,l}^-|$ and $\delta'_{I,l} = \delta_{I,l} + |V_{I,l}^-|$ in each attribute proof of the accumulator, valid $\delta'_{U,l}$'s and $\delta'_{I,l}$'s are signed by Groth signatures as follows, and the signatures are included in the public parameters pp. Calculate $\Phi_U = \{u'_U = \sum_{l=1}^{L} \delta'_{U,l} c_l \mid 1 \leq \delta'_{U,l} \leq \eta_l\}$ and $\Phi_I = \{u'_I = \sum_{l=1}^{L} \delta'_{I,l} c_l \mid 1 \leq \delta'_{I,l} \leq \eta_l\}$. For each $u'_U \in \Phi_U, u'_I \in \Phi_I$, generate $\tilde{\sigma}_{u'_U} = (\tilde{R}_U, S_U, T_U) = (\tilde{G}^r, (Y_1 \cdot G^x)^{1/r}, (Y_1^x \cdot G_1^{u'_U})^{1/r})$ as a signature on $G_1^{u'_U}$ with $\mathbf{Groth}_1.\mathbf{Sign}$, and $\tilde{\sigma}_{u'_I} = (R_I, \tilde{S}_I, \tilde{T}_I) = (G^r, (\tilde{Y}_1 \cdot \tilde{G}^y)^{1/r}, (\tilde{Y}_1^y \cdot \tilde{G}_1^{u'_I})^{1/r})$ as a signature on $\tilde{G}_1^{u'_I}$ with $\mathbf{Groth}_2.\mathbf{Sign}$ for $r \overset{\$}{\leftarrow} \mathbb{Z}_p^*$.

(v) Output the following parameters pp.
$pp = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, p, G, \tilde{G}, \{Y_i\}_{i=1}^2, \{\tilde{Y}_i\}_{i=1}^2, pk_{Groth_1}, pk_{Groth_2}, pk_{acc}, \{\tilde{\sigma}_{u'_U}\}_{u'_U \in \Phi_U}, \{\tilde{\sigma}_{u'_I}\}_{u'_I \in \Phi_I})$

IKGen(pp). Output issuser's secret key and public key (isk, ipk) using $\mathbf{Groth_1}.\mathbf{KGen}$.

$$(isk, ipk) = (a, \tilde{A}) = (a, \tilde{G}^a)$$

UKGen(pp). Output user's secret key and public key (usk, upk) using $\mathbf{Groth_2}.\mathbf{KGen}$.

$$(usk, upk) = (b, B) = (b, G^b)$$

Issue(pp, isk, U, upk). Compute the attribute parameter $P_U = \prod_{l\in U} G_l$ for the user's attribute set $U \subseteq \{1, \ldots, n\}$. Calculate a $\mathbf{Groth_1}$ signature on the attribute P_U and the user's public key upk, as the credential $cred$ by $\mathbf{Groth}_1.\mathbf{Sign}$ with ipk, as follows.

$$cred = (\tilde{R}, S, T_1, T_2)$$
$$= (\tilde{G}^r, (Y_1 \cdot G^{isk})^{1/r}, (Y_1^{isk} \cdot P_U)^{1/r}, (Y_2^{isk} \cdot upk)^{1/r})$$

VfCred$(pp, cred, ipk, U, upk)$. Compute $P_U = \Pi_{l\in U} G_l$, and verify the validity of the credential $cred$ by using $\mathbf{Groth}_1.\mathbf{Verify}$ with ipk, as follows.

$$e(S, \tilde{R}) = e(Y_1, \tilde{G}) \cdot e(G, ipk)$$
$$e(T_1, \tilde{R}) = e(Y_1, ipk) \cdot e(P_U, \tilde{G})$$
$$e(T_2, \tilde{R}) = e(Y_2, ipk) \cdot e(upk, \tilde{G})$$

PresPolicy$(pp, \{(ipk_j, I_j)\})$. Generate a verifier key pair $(vsk, vpk) = (c, C) = (c, G^c)$ using $\mathbf{Groth_2}.\mathbf{KGen}$. For every accepted issuer j, calculate the issuer j's attribute parameter $P_{I_j} = \prod_{l\in I_j} \tilde{G}_l$ for the issuer's attribute set I_j, and compute a $\mathbf{Groth}_2$ signature σ_j on the issuer public key ipk_j and the issuer attribute P_{I_j} using $\mathbf{Groth_2}.\mathbf{Sign}$ with vsk, as follows.

$$\sigma_j = (R_j, \tilde{S}_j, \tilde{T}_{j,1}, \tilde{T}_{j,2})$$
$$= (G^r, (\tilde{Y}_1 \cdot \tilde{G}^{vsk})^{1/r}, (\tilde{Y}_1^{vsk} \cdot ipk_j)^{1/r}, (\tilde{Y}_2^{vsk} \cdot P_{I_j})^{1/r})$$

Output the set $pol = (vpk, \{(ipk_j, I_j, \sigma_j)\})$ for the every accepted issuer j.

VfPolicy$(pp, pol, \{(ipk_j, I_j)\})$. Compute $P_{I_j} = \Pi_{i\in I_j} \tilde{G}_i$ for every I_j in pol, and verify the validity of its signature σ_j using $\mathbf{Groth}_2.\mathbf{Verify}$ with vpk, as follows.

$$e(R_j, \tilde{S}_j) = e(G, \tilde{Y}_1) \cdot e(vpk, \tilde{G})$$
$$e(R_j, \tilde{T}_{j,1}) = e(vpk, \tilde{Y}_1) \cdot e(G, ipk_j)$$
$$e(R_j, \tilde{T}_{j,2}) = e(vpk, \tilde{Y}_2) \cdot e(G, P_{I_j})$$

Present$(pp, cred, ipk, U, I, usk, pol, \psi_U, \psi_I, \mathbf{ctx})$. Given pp including signatures $\tilde{\sigma}_{u'_U}, \tilde{\sigma}_{u'_I}$ to certify the correctness of the ranges of vaild u'_U, u'_I, a $cred$ that shows the correctness of the user's attributes of P_U and the user's secret key usk, the issuer's public key ipk, and pol including a verifier's signature σ_j on $ipk_j = ipk$ and the attributes of P_{I_j}, and a proved CNF formula ψ_U (resp., ψ_I) on user's (resp., issuer's) attributes, generate a zero-knowledge proof pt to prove the satisfication of the CNF formulas on attributes that are certified by $cred$ and σ_j. First, generate the accumulators $acc_{\mathcal{V}_U}, acc_{\mathcal{V}_I}$, the witnesses W_U, W_I, and the parameters of the accumulators. Then, perform the randomization $\mathbf{Groth}_1.\mathbf{Rand}$, $\mathbf{Groth}_2.\mathbf{Rand}$ for the credential $cred$ and signatures $\sigma_j, \tilde{\sigma}_{u'_U}, \tilde{\sigma}_{u'_U}$. In addition, blind values that the randomizations output, and signed messages, and compute SPKs to prove the verifications of the blinded signatures and accumulators.

(i) Generate accumulators $acc_{\mathcal{V}_U}$, $acc_{\mathcal{V}_I}$ for the CNF formulas $\psi_U = (V_{U,1}^+, V_{U,1}^-, \cdots, V_{U,L}^+, V_{U,L}^-)$, and $\psi_I = (V_{I,1}^+, V_{I,1}^-, \cdots, V_{I,L}^+, V_{I,L}^-)$.

$$acc_{\mathcal{V}_U} = \prod_{1\leq l\leq L} \left(\prod_{j\in V_{U,l}^+} \tilde{G}_{n+1-j} \right)^{c_l} \left(\prod_{j\in V_{U,l}^-} \tilde{G}_{n+1-j} \right)^{-c_l}$$

$$acc_{\mathcal{V}_I} = \prod_{1\leq l\leq L} \left(\prod_{j\in V_{I,l}^+} G_{n+1-j} \right)^{c_l} \left(\prod_{j\in V_{I,l}^-} G_{n+1-j} \right)^{-c_l}$$

(ii) Generate the witnesses $W_{\mathcal{V}_U}, W_{\mathcal{V}_I}$.

$$W_U = \prod_{i\in U}\prod_{1\le l\le L}\left(\prod_{j\in V^+_{U,l}}^{j\neq i}\tilde{G}_{n+1-j+i}\right)^{c_l}\left(\prod_{j\in V^-_{U,l}}^{j\neq i}\tilde{G}_{n+1-j+i}\right)^{-c_l}$$

$$W_I = \prod_{i\in I}\prod_{1\le l\le L}\left(\prod_{j\in V^+_{I,l}}^{j\neq i}G_{n+1-j+i}\right)^{c_l}\left(\prod_{j\in V^-_{I,l}}^{j\neq i}G_{n+1-j+i}\right)^{-c_l}$$

(iii) Calculate parameters for each accumulator as follows.

$$\delta'_{U,l} = \delta_{U,l} + |V^-_{U,l}| \text{ for } \delta_{U,l} = |U\cap V^+_{U,l}| - |U\cap V^-_{U,l}| \quad (1\le l\le L)$$

$$u'_U = \delta'_{U,1}c_1 + \ldots + \delta'_{U,L}c_L,\ \tau_{u'_U} = G_1^{u'_U}$$

$$\delta'_{I,l} = \delta_{I,l} + |V^-_{I,l}| \text{ for } \delta_{I,l} = |I\cap V^+_{I,l}| - |I\cap V^-_{I,l}| \quad (1\le l\le L)$$

$$u'_I = \delta'_{I,1}c_1 + \ldots + \delta'_{I,L}c_L,\ \tau_{u'_I} = \tilde{G}_1^{u'_I}$$

$$\tilde{u}_U = |V^-_{U,1}|c_1 + \ldots + |V^-_{U,L}|c_L,\ \tau_{\tilde{u}_U} = G_1^{\tilde{u}_U}$$

$$\tilde{u}_I = |V^-_{I,1}|c_1 + \ldots + |V^-_{I,L}|c_L,\ \tau_{\tilde{u}_I} = \tilde{G}_1^{\tilde{u}_I}$$

From pp, pick up the signatures $\tilde{\sigma}_{u'_U} = (\tilde{R}_U, S_U, T_U)$ on u'_U, $\tilde{\sigma}_{u'_I} = (R_I, \tilde{S}_I, \tilde{T}_I)$ on u'_I. As in the previous system, using $\mathbf{Groth}_1.\mathbf{Rand}$, $\mathbf{Groth}_2.\mathbf{Rand}$, rerandomize $cred$ and σ_j for $ipk_j = ipk$ of the issuer j. Furthermore, rerandomize the signatures $\tilde{\sigma}_{u'_U}$, $\tilde{\sigma}_{u'_I}$ on the accumulator parameters u'_U, u'_I.

$$(\tilde{R}, S, T_1, T_2) \overset{\$}{\leftarrow} \mathbf{Groth}_1.\mathbf{Rand}(pp, cred)$$

$$(R_j, \tilde{S}_j, \tilde{T}_{j,1}, \tilde{T}_{j,2}) \overset{\$}{\leftarrow} \mathbf{Groth}_2.\mathbf{Rand}(pp, \sigma_j)$$

$$(\tilde{R}_U, S_U, T_U) \overset{\$}{\leftarrow} \mathbf{Groth}_1.\mathbf{Rand}(pp, \tilde{\sigma}_{u'_U})$$

$$(R_I, \tilde{S}_I, \tilde{T}_I) \overset{\$}{\leftarrow} \mathbf{Groth}_2.\mathbf{Rand}(pp, \tilde{\sigma}_{u'_I})$$

(iv) Select $\alpha_{U_1}, \alpha_{U_2}, \beta_U, \gamma_U, \delta_U, \alpha_{I_1}, \alpha_{I_2}, \beta_I, \gamma_I, \delta_I, \alpha, \beta_1, \beta_2, \gamma, \delta_1, \delta_2 \overset{\$}{\leftarrow} Z_p^*$, and blind the parameters in the rerandamized signatures and the signed messages as follows.

Blinded signatures:

$$\tilde{\sigma}'_{u'_U} = (\tilde{R}_U, S'_U, T'_U) = (\tilde{R}_U, S_U^{1/\alpha_{U_1}}, T_U^{1/\alpha_{U_2}}),$$

$$\tilde{\sigma}'_{u'_I} = (R_I, \tilde{S}'_I, \tilde{T}'_I) = (R_I, \tilde{S}_I^{1/\alpha_{I_1}}, \tilde{T}_I^{1/\alpha_{I_2}})$$

Blinded credential:

$$cred' = (\tilde{R}, S', T'_1, T'_2) = (\tilde{R}, S^{1/\alpha}, T^{1/\beta_1}, T^{1/\beta_2})$$

The blinded issuer's public key: $ipk'_j = ipk_j^{1/\gamma}$

Blinded policy signature:

$$\sigma'_j = (R_j, \tilde{S}_j, \tilde{T}'_{j,1}, \tilde{T}'_{j,2}) = (R_j, \tilde{S}_j, \tilde{T}_{j,1}^{1/\delta_1}, \tilde{T}_{j,2}^{1/\delta_2})$$

Blinded attributes: $P'_U = P_U^{1/\beta_U},\ P'_I = P_I^{1/\beta_I}$

Blinded witnesses: $W'_U = W_U^{1/\gamma_U},\ W'_I = W_I^{1/\gamma_I}$

Blinded range conditions: $\tau'_{u'_U} = \tau_{u'_U}^{1/\delta_U},\ \tau'_{u'_I} = \tau_{u'_I}^{1/\delta_I}$

(v) Generate the following SPK.
$\pi \leftarrow SPK[(\alpha_{U_1}, \alpha_{U_2}, \beta_U, \gamma_U, \delta_U, \alpha_{I_1}, \alpha_{I_2}, \beta_I, \gamma_I, \delta_I, \alpha, \beta_1, \beta_2, \gamma, \delta_1, \delta_2, usk)$:

$\mathbf{Groth}_1$ credential check :

$$e(Y_1, \tilde{G})^{-1} = \{e(S', \tilde{R})^{-1}\}^{\alpha} \cdot e(G, ipk'_j)^{\gamma} \tag{1}$$

$\mathbf{Groth}_1$ credential check :

$$1 = \{e(T'_1, \tilde{R})^{-1}\}^{\beta_1} \cdot e(Y_1, ipk'_j)^{\gamma} \cdot e(P'_U, \tilde{G})^{\beta_U} \tag{2}$$

$\mathbf{Groth}_1$ credential check :

$$1 = \{e(T'_2, \tilde{R})^{-1}\}^{\beta_2} \cdot e(Y_2, ipk'_j)^{\gamma} \cdot e(G, \tilde{G})^{usk} \tag{3}$$

$\mathbf{Groth}_2$ policy check :

$$e(R_j, \tilde{S}_j)^{-1} \cdot e(G, \tilde{Y}_1) \cdot e(vpk, \tilde{G}) = 1 \tag{4}$$

$\mathbf{Groth}_2$ policy check :

$$e(vpk, \tilde{Y}_1)^{-1} = \{e(R_j, \tilde{T}'_{j,1})^{-1}\}^{\delta_1} \cdot e(G, ipk'_j)^{\gamma} \tag{5}$$

$\mathbf{Groth}_2$ policy check :

$$e(vpk, \tilde{Y}_2)^{-1} = \{e(R_j, \tilde{T}'_{j,2})^{-1}\}^{\delta_2} \cdot e(G, P'_I)^{\beta_I} \tag{6}$$

$\mathbf{Groth}_1$ range check :

$$e(Y_1, \tilde{G}) \cdot e(G, pk_{Groth_1}) = e(S'_U, \tilde{R}_U)^{\alpha_{U_1}} \tag{7}$$

$\mathbf{Groth}_1$ range check :

$$e(Y_1, pk_{Groth_1})^{-1} = e(\tau'_{u'_U}, \tilde{G})^{\delta_U} \cdot \{e(T'_U, \tilde{R}_U)^{-1}\}^{\alpha_{U_2}} \tag{8}$$

$\mathbf{Groth}_2$ range check :

$$e(G, \tilde{Y}_1) \cdot e(pk_{Groth_2}, \tilde{G}) = e(R_I, \tilde{S}'_I)^{\alpha_{I_1}} \tag{9}$$

$\mathbf{Groth}_2$ range check :

$$e(pk_{Groth_2}, \tilde{Y}_1)^{-1} = e(G, \tau'_{u'_I})^{\delta_I} \cdot \{e(R_I, \tilde{T}'_I)^{-1}\}^{\alpha_{I_2}} \tag{10}$$

user acumulator check :

$$e(G_1^{\tilde{u}_U}, \tilde{G}_n) = \{e(P'_U, acc_{\mathcal{V}_U})^{-1}\}^{\beta_U} \cdot e(G, W'_U)^{\gamma_U} \cdot e(\tau'_{u'_U}, \tilde{G}_n)^{\delta_U} \tag{11}$$

issuer acumulator check :

$$e(G_n, G_1^{\tilde{u}_I}) = \{e(acc_{\mathcal{V}_I}, P'_I)^{-1}\}^{\beta_I} \cdot e(W'_I, \tilde{G})^{\gamma_I} \cdot e(G_n, \tau'_{u'_I})^{\delta_I}](pol, \psi_U, \psi_I, \mathbf{ctx}) \tag{12}$$

The zero-knowledge proof using this SPK verifies that the following conditions are satisfied.

- Whether the credential $cred$ is signed by the issuer j on P_U and upk. (eq. (1)-(3)).
- Whether the policy signature σ_j is the verifier's signature on ipk_j and P_{I_j}. (eq. (4)-(6)).
- The signatures $\tilde{\sigma}_{u'_U}$, $\tilde{\sigma}_{u'_I}$ are the signatures on $\tau_{u'_U} = G_1^{u'_U},\ \tau_{u'_I} = G_1^{u'_I}$. (eq. (7)-(10)).
- P_k of user's attributes and P_{I_j} of issuer's attributes satisfy the verifications of the accumulators (eq. (11)-(12)).

(vi) Output $pt = ((\tilde{R}, S', T'_1, T'_2), ipk'_j, (R_j, \tilde{S}_j, \tilde{T}'_{1,j}, \tilde{T}'_{2,j}), (\tilde{R}_U, S'_U, T'_U), (R_I, \tilde{S}'_I, \tilde{T}'_I), P'_U, P'_I, W'_U, W'_I, \tau'_{u'_U}, \tau'_{u'_I}, \pi)$.

$\underline{\mathbf{Verify}(pp, pt, pol, \psi_U, \psi_I, \mathbf{ctx})}$. SPK π in pt is verified on the formulas ψ_U, ψ_I, and pt is accepted if π is valid, and otherwise it is rejected.

VI. Security

Here, we discuss that the proposed system satisfies the requirements of unforgeability and unlinkability shown in the Section IV.B. As for the unforgeability, the verification formulas of Groth signatures $cred$, σ_j, $\tilde{\sigma}_{u'_U}$ and $\tilde{\sigma}_{u'_I}$ are proved with the SPK π in pt, and thus the unforgeability of Groth signatures and the soundness of SPK imply that the issuer's public key $ipk_j = ipk$ is signed by the policy signature as in the previous systems [1], [6]. Furthermore, the correctness of $P_U = \Pi_{l\in U} G_l$, $P_{I_j} = \Pi_{l\in I_j} \tilde{G}_l$, $u'_U \in \Phi_U$, and $u'_I \in \Phi_I$ are ensured for blinded $P'_U = P_U^{-\beta_U}$, $P'_I = P_I^{-\beta_I}$, $\tau'_{u'_U} = \tau_{u'_U}^{-\delta_U}$, $\tau'_{u'_I} = \tau_{u'_I}^{-\delta_I}$ where $\tau_{u'_U} = G_1^{u'_U}$ and $\tau_{u'_I} = G_1^{u'_I}$. Thus, $u'_U = \Sigma_{l=1}^{L} \delta'_{U,l} c_l$ s.t. $1 \le \delta'_{U,l} \le \eta_l$ holds. Since the SPK proves eq.(11), we have

$$e(\tau_{\tilde{u}_U}, \tilde{G}_n) = e(P'_U, acc_{\mathcal{V}_U})^{-\beta_U} \cdot \\ e(G, W'_U)^{\gamma_U} \cdot e(\tau'_{u'_U}, \tilde{G}_n)^{\delta_U}$$

From $P_U = {P'_U}^{\beta_U}$, $W_U = {W'_U}^{\gamma_U}$, and $\tau_{u'_U} = {\tau'_{u'_U}}^{\delta_U}$, we have

$$e(\tau_{\tilde{u}_U}, \tilde{G}_n) = e(P_U, acc_{\mathcal{V}_U})^{-1} \cdot \\ e(G, W_U) \cdot e(\tau_{u'_U}, \tilde{G}_n), \\ e(P_U, acc_{\mathcal{V}_U}) \cdot e(G, W_U)^{-1} = e(\tau_{u'_U} \cdot \tau_{\tilde{u}_U}^{-1}, \tilde{G}_n).$$

From $\tau_{\tilde{u}_U} = G_1^{|V_{U,1}^-|c_1 + \cdots + |V_{U,L}^-|c_L}$ and $\tau_{u'_U} = G_1^{\delta'_{U,1}c_1 + \cdots + \delta'_{U,L}c_L}$, we have

$$e(P_U, acc_{\mathcal{V}_U}) \cdot e(G, W_U)^{-1} = e(G_1, \tilde{G}_n)^{\Sigma_{i=1}^{L}(\delta'_{U,l} - |V_{U,l}^-|)\cdot c_l}.$$

For $\delta_{U,l} = \delta'_{U,l} - |V_{U,l}^-|$, due to $1 \le \delta'_{U,l} \le \eta_l$, we have $1 \le \delta_{U,l} + |V_{U,l}^-| \le \eta_l$. Therefore, the accumulator verification is proved, and thus the CNF formula ψ_U is satisfied by U. For the issuer's attributes, we can prove that ψ_I is satisfied similarly.

As for the unlinkability, it follows from the hiding property of the rerandomization and blinding, and the zero-knowledge property of SPK.

VII. Implementation and Experimental Results

To evaluate our system, we implemented it on a PC (WSL2 Ubuntu 22.04.4 LTS, AMD Ryzen 7 5700X 8-Core Processor, 32.0GB) using C language with GMP library and pairing library ELiPS [**?**]. A Barreto-Lynn-Scott curve of embedding degree 12 over a 461-bit prime field is used.

We measured the processing times of **Issue**/**VfCred**, **PresPolicy**/**VfPolicy**, and **Present**/**Verify**. Unless otherwise noted, the number of issuers in the policy is 10, the maximum number n of attributes is 1000, the numbers of user's and issuer's attributes are 10, and the number of literals of proved CNF formula is 10.

A. Comparisons of Performances between Previous Systems and Proposed System

We show comparisons of performances of algorithms between the previous systems [1], [6] and the proposed system in Fig.1. In the proposed system, the processing times of **Present** and **Verify** are increased, since the verifications for the user's and the issuer's attributes using accumulator are added. However, the times are less than 100ms, and those are practical on a PC for general use.

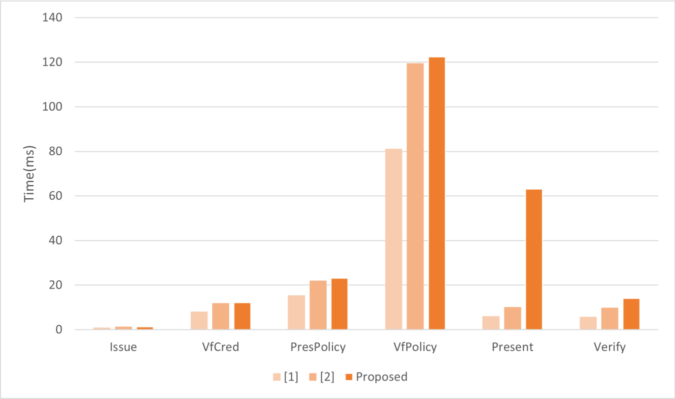

Fig. 1. Comparisons of performances between the previous systems and the proposed system

B. Performance for Number of User's or Issuer's Attributes

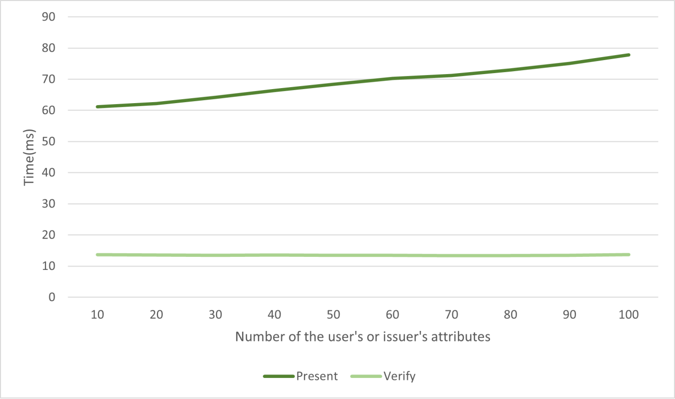

Fig. 2. Performance for the number of the user's or issuer's attributes

In Fig.2, we show the processing times of **Present** and **Verify**, when the number of the user's or issuer's attributes are increased by 10. The **Verify** time is constant, due to the accumulator. The **Present** time increases linearly, since the generation of the witnesses W_U, W_I depends on the number of the attributes. However, the amount of increase is gradual, and the time is practical in case of even 100.

C. Performance for Number of CNF Literals

In Fig.3, we show the processing times of **Present** and **Verify**, when the number of literals in a proved CNF formula is increased from 10 to 1000. Similarly to Fig.2, the **Present** time increases linearly, since the generations of the accumulators $acc_{\mathcal{V}_U}$, $acc_{\mathcal{V}_I}$ and the witnesses W_U, W_I depend on the number of the literals. As the CNF formula is more complex, the more present time is headed. However, the **Verify** time is constant and very fast.

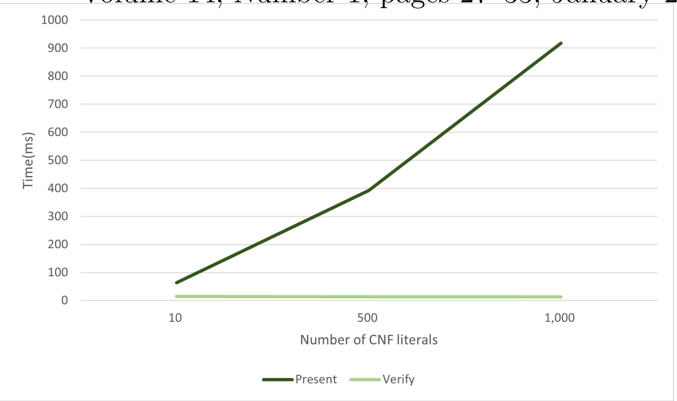

Fig. 3. Performance for the number of CNF literals

D. Comparisons of Data Size between Previous Systems and Proposed System

We discuss the data size of the public parameter in **ParGen**, the policy in **PresPolicy** and the presentation token in **Present**. The order of the data size of the public parameter is $O(n + \Pi_{l=1}^{L} \eta_l)$, since the accumulator's setup parameter depends on the maximum size of attributes n, and the numbers for Φ_U, Φ_I are $\Pi_{l=1}^{L} \eta_l$. The size of the policy pol is $O(N_I)$ for the number N_I of accepted issuers, since the policy includes signatures for the accepted issuers. Since the number of verifications using the accumulator in the SPK is constant, the size of the presentation token pt is constant.

VIII. Conclusion

In this paper, we have proposed an extended issuer-hiding ABC system, where the attribute proof can be verified using the accumulator [7]. The processing time of the verification do not depend on the number of attributes of user and issuer, and the size of CNF formula, and the proof size is also constant. On the other hand, the processing time of the attribute proof generation is increased.

Our future work includes an implementation of an application system using the proposed system, and its evaluation.

References

[1] J. Bobolz, F. Eidens, S. Krenn, S. Ramacher and K. Samelin: *Issuer-Hiding Attribute-Based Credentials*, CANS 2021, pp.158–178, 2021.

[2] J. Camenisch, M. Kohlweiss, and C. Soriente: *An accumulator based on bilinear maps and efficient revocation for anonymous credentials*, 12th International Conference on Practice and Theory in Public Key Cryptography (PKC 2009), LNCS 5443, pp.481-500, 2009.

[3] J. Camenisch and A. Lysyanskaya: *An efficient system for non-transferable anonymous credentials with optional anonymity revocation*, Advances in Cryptology - EUROCRYPT 2001, LNCS 2045 pp.93-118, 2001.

[4] D. Chaum: *Untraceable electronic mail, return addresses, and digital pseudonyms*, Commun. ACM, Vol.24, No.2, pp.84-88, 1981.

[5] J. Groth: *Efficient fully structure-preserving signatures for large messages*, Advances in Cryptology, ASIACRYPT 2015, Part I, LNCS 9452, pp.239–259, 2015.

[6] Y. Hamada, T. Nakanishi, T. Kitasuka: *Issuer-Hiding Attribute-Based Credentials with Verification of Issuer's Attributes*, ICCE-TW 2024, pp.389-390, 2024.

[7] R. Okishima, T. Nakanishi: *An Anonymous Credential System with Constant-Size Attribute Proofs for CNF Formulas with Negations*, IEICE TRANS.FUNDAMENTALS, vol.e103-a, no.12, pp.1381-1392, 2020.

[8] Y. Takahashi, Y.Nanjo, T. Kusaka, Y. Nogami, T. Kaneari, T. Tatara: *An implementation and evaluatioin of pairing library ELiPS for BLS curve with several techniques*, 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), 2019.