Attribute-based Anonymous Credential: Optimization for Single-Use and Multi-Use

Kwan Yin Chan, Tsz Hon Yuen

The University of Hong Kong
{kychan, thyuen}@cs.hku.hk

Abstract. User attributes can be authenticated by an attribute-based anonymous credential while keeping the anonymity of the user. Most attribute-based anonymous credential schemes are designed specifically for either multi-use or single-use. In this paper, we propose a unified attribute-based anonymous credential system, in which users always obtain the same format of credential from the issuer. The user can choose to use it for an efficient multi-use or single-use show proof. It is a more user-centric approach than the existing schemes.
Technically, we propose an interactive approach to the credential issuance protocol using a two-party computation with an additive homomorphic encryption. At the same time, it keeps the security property of impersonation resilience, anonymity, and unlinkability. Apart from the interactive protocol, we further design the show proofs for efficient single-use credentials which maintain the user anonymity.

Keywords: Anonymous credential · zero-knowledge proof · single-use · multi-use.

1 Introduction

Anonymous credentials provide a secure approach for transmitting trust signals while protecting the anonymity of the users simultaneously in a cryptographic way. Starting from the idea from Chaum [11], the development of anonymous credentials has been last for decades. Attribute-based access control is already implemented by the industry[1]. Hence, attribute-based anonymous credential (ABC) with efficient show proofs for complex statements are important [40].

Looking at the typical design of an attribute-based credential, it contains several attributes from the user who is being authenticated (e.g., age = 20, gender = male, ZIP code = 123456). A full disclosure of attributes may compromise the identity of the user. For example, 87% of the population in U.S. can be uniquely identified based on ZIP code, date of birth, and gender [36]. With the help of zero-knowledge show proofs, ABC allows the minimal number of user

[1] AWS: https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction_attribute-based-access-control.html. Azure: https://docs.microsoft.com/en-us/azure/role-based-access-control/conditions-overview

attributes to be disclosed during authentication and maintains strong privacy. Verifiers could verify the zero-knowledge proof from a user without knowing the content of the credential. Therefore, the credentials can be used multiple times with anonymity.

***q*-SDH-based anonymous credential system.** Tan *et al.* [40] introduced an attribute-based anonymous credential (ABC) system, which supports the needs of showing proofs for complex statements. The system generates credentials with the self-introduced *MoniPoly* commitment scheme and the SDH-based Camenisch-Lysyanskaya (CL) signature [7]. The proposed scheme includes a series of zero-knowledge proofs, which proves the relationship between the attribute set in the credential A and the attribute set of the verifier A'. The authors proposed AND, ANY/OR, NAND/NOT and NANY proofs concretely. The ABC system also achieves impersonation resilience, anonymity, and unlinkability.

Single-use and multi-use credentials. Considering the (un)linkability of the anonymous credentials, they could be classified into *single-use* credentials and *multi-use* credentials. A *multi-use* credential can be used more than once without revealing any linkage between different authenticated sessions. We could find many protocols are in the type of *multi-use* credentials in recent research [12,23,40] which aim to reduce the number of credentials issued from the issuer.

If an *single-use* credential is used to authenticate two different sessions, the two sessions are linked (while the anonymity of the attributes is still preserved). The traditional application of *single-use* credentials can be shown in U-Prove [32], also it could be very useful under the situation of e-voting (double voting can be detected). Existing research work of *single-use* credentials could be found in [14, 18, 26], and they are usually more efficient than *multi-use* credentials during the authentication protocol. On the other hand, if the linkage of sessions is undesirable in the application, the issuer has to generate a new credential for each communication. The cost of the authentication protocol is switched to the credential generation protocol in this case.

1.1 Motivation

In this paper, we design a unified attribute-based anonymous credential system supporting efficient single-use and multi-use credential. In the existing ABC systems, the choice of single-use or multi-use credential is selected by the one who design and implement the system (which may be the issuer). In this paper, we design a unified ABC system, in which this choice is chosen by the *user* or depending on the real world application. The issuer, which is usually an authority managing identities, does not need to care about the underlying use case of the credential system.

Consider the example of anonymous discussion forum. Users can be authenticated by anonymous credential before posting messages. The user can choose to use a multi-use credential if he wants all his posts remain unlinked to each other. The user can also choose to use a single-use credential to keep all his posts linked in the same thread/topic. In this case, the choice of single-use of multi-use is chosen by the user.

Building an efficient attribute-based anonymous credential system supporting both single-use and multi-use credential is a non-trivial task. Many existing single-use credentials are constructed from symmetric key primitives [14,26] and hence it is not efficient to use a zero-knowledge proof to convert it into a multi-use credential. On the other hand, most existing multi-use credentials are fully or partially known by the issuer [7,12,17,23,27,40]. Simply presenting this credential as a show proof of a single-use credential compromise the anonymity with respect to the issuer.

1.2 Our Contributions

In this paper, we propose an attribute-based anonymous credential system which support efficient single-use and multi-use show proof. In particular, the choice of single-use or multi-use is chosen by the user, instead of the issuer. As compared to the state-of-the-art scheme from Tan *et al.* [40], our system has the contributions below:

User-centric anonymous credentials. Based on the system proposed by Tan *et al.* [40], we propose a interactive credential issuance protocol using two-party computation and additive homomorphic encryption. Following this interactive approach, the full content of the credential is only known by the user, not by the issuer. It enables the user to use the credential as a single-use or multi-use later. The interactive approach does not affect any of the multi-use show proof presentation protocol in [40]. Our multi-use show proof credential system achieves impersonation resilience, anonymity, and unlinkability.

If the user choose to use a single-use show proof, he can also choose to use a subset of his attributes when generating that single-use credential. For example, if Alice has 100 attributes and she is going to use 3 attributes in a single-use anonymous authentication, she just needs to request a credential with these 3 attributes only. It can significantly reduce the computation and communication complexity, especially when the user has a large number of attributes.

Clear role for the Issuer. In [40], the attributes from users are committed using the *MoniPoly* commitment scheme. The commitment is sent to the issuer and the issuer generates credential upon the commitment. Tan *et al.* claimed that their scheme achieved perfectly hiding of attributes with respect to the issuer. However, we find it unrealistic for the issuer (e.g., the identity authority) for not knowing the attributes (e.g., gender, date of birth, etc.) when it issues credential. Generally, the issuer knows the content of the attribute set from a user, and what attribute(s) that the user would like to authenticate in each credential. This prevents some malicious users issue the credentials with some unexpected attributes.

In our proposed protocol, we set that the issuer knows the attributes of a particular user. Although this approach breaks the perfectly hiding of attributes among the issuing protocol between the issuer and the user, it maintains the power that the issuer understands what it is authenticating and that is its only role. The issuer does not need to care about the choice of single-use or multi-use in the system.

Show proofs for single-use credentials. We further design the show proofs with logic statements for single-use credentials. Since the single-use credentials are aimed to use once only, a prover could hide less information in order to decrease computation costs while maintaining anonymity. With the single-use setting, we further proposed batch credentials issuance in order to minimize the communication rounds.

1.3 Overview of Our Scheme

To fill the aforementioned research gap, we propose a modified attribute-based credential system from Tan *et al.* [40], with a new two-party computation approach to generating the credential.

The protocol proposed by Tan *et al.* [40] is a state-of-the-art attribute-based anonymous credential system, with *multi-use* credentials. Furthermore, it provides different show proofs to prove the credential with a statement, comparing the requirement from the verifiers and the authenticated attributes from the users. The credential in [40] is essentially a Camenisch-Lysyanskaya (CL) signature [7]. Suppose that C is the commitment of the user attributes, x is the issuer's secret key, b, c are system parameters. The credential is

$$(t, \quad s, \quad v = (C \cdot b^s \cdot c)^{\frac{1}{x+t}}),$$

in which t and s are jointly chosen by the issuer and the user. Simply presenting (t, s, v) as a single-use credential violates anonymity since t is known to the issuer.

In this paper, we design a new two-party computation protocol to generate the CL signature, such that the issuer has no information about the entire credential (t, s, v). If we can obtain such credential, we can design an efficient show proof for the single-use credentials. The major technical difficulty is the computation of $1/(x + t)$ in the exponent without the full knowledge of t by the issuer. We take advantage of the multiplicative to additive (MtA) protocol [20] to achieve such computations between the two parties.

Share Conversion Protocol. Gennaro *et al.* [20] generalized the multiplicative to additive (MtA) share conversion protocol using additive homomorphic encryption. This protocol shares secrets between two parties in the form of $\alpha + \beta = ab$, where a (and α) and b (and β) are the secrets (and additive shares) kept by Alice and Bob, respectively.

In our scheme, we adopt the MtA protocol during the interactive generation of credentials between the issuer and the user. Now the issuer knows the secret key x and the user chooses a random t. They engage in the MtA protocol and obtain a, b such that $x + t = ab$. The issuer computes $v' = (C \cdot b^s \cdot c)^{\frac{1}{a}}$ and the user can recover $v = v'^{\frac{1}{b}}$. By the security of the MtA protocol, the issuer has no information of t and v.

1.4 Related Works

We remark that Tan *et al.* [40] provides a detailed comparisons between recent works. Apart from the q-SDH-based anonymous credential system, various systems are using different cryptographic approaches. Recently, some research [12,17,23] focus on the anonymous credential system using the structure-preserving signatures on equivalence classes. Moreover, some are designed using different kind of signatures such as Camenisch and Lysyanskaya (CL) Signature [7, 27, 40], Abe-Haralambiev-Ohkubo signature [1, 31] and BLS signature [21, 35]. Moreover, there are various systems are designed with special functionalities, such as multi-authority credentials [2,24], anonymous credentials with redactable signature [37], blacklistable anonymous credentials [1,30,41,42], anonymous credentials with accumulator-based revocation [3], updatable anonymous credentials [22], and delegatable anonymous credentials [3, 4, 13]. Taking advantages from the anonymous credential protocols, many applications were being proposed and applied on different aspects such as direct anonymous attestation (DAA) [6], application on Smart City [5,28,29,34], smart cards [16], IoT devices [8, 33, 39], and blockchain [15]. And, the real-time message application *Signal* adopts anonymous credential [9].

Organization. The organization of this paper is as follows. In Section 2, we briefly introduce the mathematical background and the secure models. We define the security requirement of our ABC scheme in Section 3, the construction in Section 4, and the security proofs in Section 5. We further propose the zero-knowledge proofs for single-use credentials and the security proofs in Section 6. We give a conclusion in Section 7.

2 Preliminaries

2.1 Bilinear Pairing

Consider $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ be cyclic groups of prime order p such that $\mathsf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Assuming $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$ and $x, y \in \mathbb{Z}_p$, the bilinear pairing function follows the properties below:

1. Bilinearity: $\mathsf{e}(g_1^x, g_2^y) = \mathsf{e}(g_1^y, g_2^x) = \mathsf{e}(g_1, g_2)^{xy}$
2. Non-degeneracy: $\mathsf{e}(g_1, g_2) \neq 1$
3. Efficiency: e is efficiently computable.

Galbraith *et al.* [19] classified the types of pairing: (1) Type 1: $\mathbb{G}_1 = \mathbb{G}_2$; (2) Type 2: $\mathbb{G}_1 \neq \mathbb{G}_2$ where there exists an efficient isomorphism ψ; and (3) Type 3: $\mathbb{G}_1 \neq \mathbb{G}_2$ with no isomorphism exists. In this work, we take Type 3 pairings.

2.2 Security Assumptions

Definition 1. *Discrete logarithm Assumption (DLOG): An algorithm $\mathcal{C}$ $(t_{\mathsf{dlog}}, \epsilon_{\mathsf{dlog}})$-breaks the DLOG assumption if $\mathcal{C}$ runs with a negligible probability ϵ_{dlog} such that:*

$$\Pr[x \in \mathbb{Z}_p : \mathcal{C}(g, g^x) = x)] \geq \epsilon_{\mathsf{dlog}}$$

and runs in time at most t_{dlog}*. It is said that the DLOG assumption is* $(t_{\mathsf{dlog}}, \epsilon_{\mathsf{dlog}})$*-secure is there are no algorithm* $(t_{\mathsf{dlog}}, \epsilon_{\mathsf{dlog}})$*-solves the DLOG problem.*

Definition 2. *q-Strong Diffie-Hellman Assumption (SDH) [38]: An algorithm* $\mathcal{C}$ $(t_{\mathsf{sdh}}, \epsilon_{\mathsf{sdh}})$*-breaks the SDH assumption if* $\mathcal{C}$ *runs with a negligible probability* ϵ_{sdh} *such that:*

$$\Pr[x \in \mathbb{Z}_p, c \in \mathbb{Z}_p \setminus \{-x\} : \mathcal{C}(g_1, g_1^x, ..., g_1^{x^q}, g_2, g_2^x) = (g_1^{\frac{1}{x+c}}, c)] \geq \epsilon_{\mathsf{sdh}}$$

and runs in time at most t_{sdh}*. It is said that the SDH assumption is* $(t_{\mathsf{sdh}}, \epsilon_{\mathsf{sdh}})$*-secure is there are no algorithm* $(t_{\mathsf{sdh}}, \epsilon_{\mathsf{sdh}})$*-solves the SDH problem.*

Definition 3. *q-co-Strong Diffie-Hellman Assumption (co-SDH) [10]: An algorithm* $\mathcal{C}$ $(t_{\mathsf{cosdh}}, \epsilon_{\mathsf{cosdh}})$*-breaks the co-SDH assumption if* $\mathcal{C}$ *runs with a negligible probability* $\epsilon_{\mathsf{cosdh}}$ *such that:*

$$\Pr[x \in \mathbb{Z}_p, c \in \mathbb{Z}_p \setminus \{-x\} : \mathcal{C}(g_1, g_1^x, ..., g_1^{x^q}, g_2, g_2^x, ..., g_2^{x^q}) = (g_1^{\frac{1}{x+c}}, c)] \geq \epsilon_{\mathsf{cosdh}}$$

and runs in time at most t_{cosdh}*. It is said that the co-SDH assumption is* $(t_{\mathsf{cosdh}}, \epsilon_{\mathsf{cosdh}})$*-secure is there are no algorithm* $(t_{\mathsf{cosdh}}, \epsilon_{\mathsf{cosdh}})$*-solves the co-SDH problem.*

2.3 The SDH-based Camenisch and Lysyanskaya (CL) Signature

We recall a pairing-based signature schemes introduced by Camenisch and Lysyanskaya [7] as follows:

$\mathsf{KeyGen}(1^k)$: Construct three cyclic groups $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ of order p based on an bilinear-based elliptic curve with the bilinear pairing $\mathsf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. Sample randon generators $a, b, c \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$ and a secret value $x \in \mathbb{Z}_p^*$. This alrogithm outputs the public key $pk = (\mathsf{e}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, a, b, c, g_2, X = g_2^x)$ and the secret key $sk = x$.

$\mathsf{Sign}(m, pk, sk)$: The algorithm intakes message m, chooses random values $s, t \in \mathbb{Z}_p$ for computing $v = (a^m b^s c)^{\frac{1}{x+t}}$. If the rare case with $x + t = 0 \mod p$ occurs, reselect random t. It outputs the signature $sig = (t, s, v)$.

$\mathsf{Verify}(m, sig, pk)$: The algorithm verify the signature sig with

$$\mathsf{e}(v, Xg_2^t) = \mathsf{e}((a^m b^s c)^{\frac{1}{x+t}}, g_2^{x+t}) = \mathsf{e}(a^m b^s c, g_2).$$

It outputs 1 for a successful verification, 0 otherwise.

2.4 Share Conversion Protocol

Gennaro *et al.* [20] generalized the multiplicative to additive (MtA) share conversion protocol. Assume that Alice and Bob holding a, b respectively, attempt to share the secret in the form of $\alpha + \beta = ab$ using the homomorphic encryption with the mechanism below. We omit the details of the range proof and the zero-knowledge proof without loss of generality. However, we emphasize that all the proofs are required for a particular homomorphic cryptosystem.

1. Alice computes $c_A = \mathsf{Enc}_A(a)$, and sends c_A to Bob.
2. Bob picks β' uniformly randomly and computes $c_B = b \times c_A + \mathsf{Enc}_A(\beta') = \mathsf{Enc}_A(ab + \beta')$, where $\beta = -\beta'$. Bob replies Alice with c_B.
3. Alice decrypts c_B and gets α', such that $\alpha = \alpha'$.

Alice and Bob eventually reveal α and β to each other and compute $\alpha + \beta$ on their own.

3 Security Requirements for Attribute-based Anonymous Credential

In this section, we generally introduce the security models of impersonation resilience, anonymity, and unlinkability. The attribute-based anonymous credential system is divided into the algorithms as follows:

1. $\mathsf{KeyGen}(1^k, 1^n) \rightarrow (pk, sk)$: Executed by the issuer with the security parameter k and the attributes upper bound n, it generates a key pair (pk, sk).
2. $(\mathsf{Obtain}(pk, A),\ \mathsf{Issue}(pk, sk, T)) \rightarrow (cred \text{ or } \perp)$: Interactively executed by the issuer and the user, these two algorithms form the credential issuing protocol. Obtain is invoked by the user with the public key of issuer pk and an attribute set A. With the request from user, the issuer executed the Issue algorithm with the public key pk, the secret key sk, and the attributes table T. The protocol outputs a valid credential $cred$ or a null value $\perp$ otherwise.
3. $(\mathsf{Prove}(pk, cred, \phi_{\mathsf{stmt}}),\ \mathsf{Verify}(pk, \phi_{\mathsf{stmt}}) \rightarrow b)$: These two algorithms form the credential presentation protocol. An access policy ϕ is formed by an attribute set A from prover, with a statement stmt that specifies the relation between A and A' from the verifier. The details of this protocol remains unchanged in this work, please refer to [40] for details.

3.1 Impersonation Resilience

The property of impersonation resilience requires that it is infeasible to get accepted by the verifier for an adversary in the show proof. We recap the definition the security model as the security against impersonation under active and concurrent attacks (imp-aca) between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ in **Game 1** from Tan *et al.* [40] as follows:

Game 1 (imp-aca($\mathcal{A}$, $\mathcal{C}$))

1. ***Setup:*** *$\mathcal{C}$ runs* $\mathsf{KeyGen}(1^k, n)$ *and sends pk to $\mathcal{A}$.*
2. ***Phase 1:*** *$\mathcal{A}$ is able to play the role of user, prover and verifier, respectively. He can issue concurrent queries to the* Obtain, Prove *and* Verify *oracles on any attribute set A_i of his choice in the i-th query. And, $\mathcal{A}$ can issue queries to the* IssueTranscript *oracle which takes in A_i and returns the corresponding transcripts of the issuing protocol.*

3. ***Challenge:*** *$\mathcal{A}$ output the challenge attribute set A^* and its corresponding access policy ϕ^*_{stmt} such that $\phi^*_{\mathsf{stmt}}(A_i) = 0$ and $\phi^*_{\mathsf{stmt}}(A^*) = 1$ for every A_i queried to the* Obtain *oracle during Phase 1.*
4. ***Phase 2:*** *With the restriction that $\mathcal{A}$ cannot query an attribute set A_i to* Obtain *such that $\phi^*_{\mathsf{stmt}}(A_i) = 1$, it can continue to query the oracles as in Phase 1.*
5. ***Impersonate:*** *$\mathcal{A}$ completes a show proof as the prover with $\mathcal{C}$ as the verifier for the access policy $\phi^*_{\mathsf{stmt}}(A^*) = 1$. $\phi^*_{\mathsf{stmt}}(A^*) = 1$ wins the game if $\mathcal{C}$ outputs* 1*, otherwise* 0.

Definition 4. *An adversary $\mathcal{A}$ is said to ($t_{\mathsf{imp}}, \epsilon_{\mathsf{imp}}$)-break the* imp-aca *security of an ABC system if $\mathcal{A}$ runs in time at most t_{imp} and wins **Game 1** for a negligible probability ϵ_{imp} such that:*

$$\Pr[(\mathcal{A}, \mathsf{Verify}(pk, \phi^*_{\mathsf{stmt}})) = 1] \geq \epsilon_{\mathsf{imp}}.$$

A particular ABC system is imp-aca*-secure if there are no adversary ($t_{\mathsf{imp}}, \epsilon_{\mathsf{imp}}$)-wins **Game 1**.*

3.2 Anonymity

The property of anonymity requires an adversary cannot recover the identity of a user from the show proofs. We introduce the security model for anonymity under active and concurrent attacks (anon-aca) in **Game 2** between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$:

Game 2 (anon-aca($\mathcal{A}$, $\mathcal{C}$))

1. ***Setup:*** *$\mathcal{C}$ runs* $\mathsf{KeyGen}(1^k, n)$ *and sends pk, sk to $\mathcal{A}$.*
2. ***Phase 1:*** *$\mathcal{A}$ is able to play the role of user, issuer, prover and verifier, respectively. He can issue concurrent queries to the* Obtain*,* Issue*,* Prove *and* Verify *oracles on any attribute set A_i of his choice in the i-th query. And, $\mathcal{A}$ can issue queries to the* Corrupt *oracle which takes in transcript of presentation protocol from prover which is $\mathcal{C}$ and returns the entire internal state, including the random seed used by $\mathcal{C}$ in the transcript.*
3. ***Challenge:*** *$\mathcal{C}$ selects two credentials from* IssueTranscript *oracle which contains the credentials created with $\mathcal{A}$ as a Issuer.*

   $$(\mathsf{Otain}(pk, A_0), \mathsf{Issue}(pk, sk)) \rightarrow cred_0, \quad (\mathsf{Otain}(pk, A_1), \mathsf{Issue}(pk, sk)) \rightarrow cred_1.$$

   The two selected credentials hold with equal length, and the access policy ϕ^*_{stmt} which aim to challenge such that $\phi^*_{\mathsf{stmt}}(A_0) = \phi^*_{\mathsf{stmt}}(A_1) = 1$. $\mathcal{C}$ responds a random bit $b \in \{0,1\}$ and interacts as the prover with $\mathcal{A}$ as verifier to complete the protocol:

   $$(\mathsf{Prove}(pk, cred_b, \phi^*_{\mathsf{stmt}}), \mathsf{Verify}(pk, \phi^*_{\mathsf{stmt}})) \rightarrow 1.$$
4. ***Phase 2:*** *With the restriction that $\mathcal{A}$ cannot query transcript of the challenged show proofs to* Corrupt*, it can continue to query the oracles as in Phase 1.*

5. ***Guess:*** *$\mathcal{A}$ wins the game with guess b' if $b' = b$.*

Definition 5. *An adversary $\mathcal{A}$ is said to $(t_{\mathsf{ano}}, \epsilon_{\mathsf{ano}})$-break the* anon-aca *security of an ABC system if $\mathcal{A}$ runs in time at most t_{ano} and wins **Game 2** for a negligible probability ϵ_{ano} such that:*

$$\Pr[b = b'] - \frac{1}{2} \geq \epsilon_{\mathsf{ano}}$$

A particular ABC system is anon-aca*-secure if there are no adversary $(t_{\mathsf{ano}}, \epsilon_{\mathsf{ano}})$-wins **Game 2**.*

3.3 Unlinkability

The property of unlinkability requires an adversary cannot link the attributes or instances among the presentation protocols. We introduce the security model for unlinkability under active and concurrent attacks (unl-aca) in **Game 3** between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$, which requires an adversary cannot distinguish the sequence of two attribute sets after being involved in the generation of a list of credentials:

Game 3 (unl-aca($\mathcal{A}$, $\mathcal{C}$))

1. ***Setup:*** *$\mathcal{C}$ runs* KeyGen(1^k, n) *and sends pk, sk to $\mathcal{A}$.*
2. ***Phase 1:*** *$\mathcal{A}$ is able to play the role of user, issuer, prover and verifier, respectively. He can issue concurrent queries to the* Obtain*,* Issue*,* Prove *and* Verify *oracles on any attribute set A_i of his choice in the i-th query. And, $\mathcal{A}$ can issue queries to the* Corrupt *oracle which takes in transcript of presentation protocol from prover which is $\mathcal{C}$ and returns the entire internal state, including the random seed used by $\mathcal{C}$ in the transcript.*
3. ***Challenge:*** *$\mathcal{C}$ responds a random bit $b \in \{0, 1\}$ and selects two credentials from* IssueTranscript *oracle which contains the credentials created with $\mathcal{A}$ as a Issuer.*

   $$(\mathsf{Otain}(pk, A_b), \mathsf{Issue}(pk, sk)) \rightarrow cred_b, \quad (\mathsf{Otain}(pk, A_{1-b}), \mathsf{Issue}(pk, sk)) \rightarrow cred_{1-b}.$$

   The two selected credentials hold with equal length, and the access policy ϕ^*_{stmt} which aim to challenge such that $\phi^*_{\mathsf{stmt}}(A_0) = \phi^*_{\mathsf{stmt}}(A_1) = 1$. $\mathcal{C}$ interacts as the prover with $\mathcal{A}$ as verifier to complete the protocol:

   $$(\mathsf{Prove}(pk, cred_b, \phi^*_{\mathsf{stmt}}), \mathsf{Verify}(pk, \phi^*_{\mathsf{stmt}})) \rightarrow 1,$$

   $$(\mathsf{Prove}(pk, cred_{1-b}, \phi^*_{\mathsf{stmt}}), \mathsf{Verify}(pk, \phi^*_{\mathsf{stmt}})) \rightarrow 1.$$

4. ***Phase 2:*** *With the restriction that $\mathcal{A}$ cannot query transcript of the challenged show proofs to* Corrupt*, it can continue to query the oracles as in Phase 1.*
5. ***Guess:*** *$\mathcal{A}$ wins the game with guess b' if $b' = b$.*

Definition 6. *An adversary $\mathcal{A}$ is said to $(t_{\mathsf{unl}}, \epsilon_{\mathsf{unl}})$-break the* unl-aca *security of an ABC system if $\mathcal{A}$ runs in time at most t_{imp} and wins **Game 3** for a negligible probability ϵ_{unl} such that:*

$$\Pr[b = b'] - \frac{1}{2} \geq \epsilon_{\mathsf{unl}}$$

A particular ABC system is unl-aca*-secure if there are no adversary $(t_{\mathsf{unl}}, \epsilon_{\mathsf{unl}})$-wins **Game 3**.*

4 Our Construction on Credential Issuance

In this section, we illustrate our modification towards Tan *et al.* [40]. In our construction, we utilize the share conversion protocol (MtA), trying to share secrets between the issuer and the user during the credential issuance protocol.

Generally, the issuer knows the content of the attribute set from a user, and what attribute(s) that the user would like to authenticate in each credential. This prevents some malicious users issue the credentials with some unexpected attributes. Therefore, we further allow the issuer holds table T which maintains the attributes of a particular user.

The user credential is an q-SDH-based CL signature, with the use of *MoniPoly* commitment C with its attribute set A. Precisely, an attribute m_i is an attribute-value pair (attribute=value) and $A = \{m_1, ..., m_{n-1}\}$ is an attribute set. And, we assume A'' be a set of attributes without value (attribute), where the attributes in A and A'' are the same. An access policy ϕ is formed by an attribute set A with a statement that specifies the relation between A and A'.

4.1 Key Generation

$\mathsf{KeyGen}(1^k,\ n)$: Construct three cyclic groups $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ of order p based on an bilinear-based elliptic curve with the bilinear pairing $\mathsf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. Sample random generators $a, b, c \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$ and two secret values $x, x' \in \mathbb{Z}_p$. Compute parameters $a_0 = a, a_1 = a^{x'}, ..., a_n = a^{x'^n}$, $X = g_2^x, X_0 = g_2, X_1 = g_2^{x'}, ..., X_n = g_2^{x'^n}$. The Issuer also generates the key pair for homomorphic encryption E by running $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{E.KeyGen}()$. The homomorphic key pair is generated with corresponding range proofs and zero-knowledge proofs, depending on the adopted homomorphic cryptosystem. This algorithm outputs the public key $pk = (\mathsf{e}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, b, c, \{a_i, X_i\}_{0 \leq i \leq n}, X, \mathsf{pk})$ and the secret key $sk = (x, x', \mathsf{sk})$. The Issuer also maintains a table T of its users and their corresponding attributes.

4.2 Issue of Credentials

$(\mathsf{Obtain}(pk,\ A),\ \mathsf{Issue}(pk, sk, T))$: The algorithm instantiates an interaction between the user and the issuer, generating a user credential *cred* on an attribute set $A = \{m_1, ..., m_{n-1}\}$. The algorithm operates in the following steps:

1. User samples a random opening value $o \in \mathbb{Z}_p$ to compute $C = \prod_{j=0}^{n} a_j^{\mathsf{m}_j} = \mathsf{Commit}(pk, A, o)$. Subsequently, user selects random $s_1 \in \mathbb{Z}_p$ to initialize the issuing protocol by completing the ZK protocol π_s with Issuer:
$$PK\{s_1 : M = C \cdot b^{s_1}\}.$$
User sends (A'', M, C, o, π_s) to Issuer.
2. Issuer receives (A'', M, C, o, π_s) and generates the corresponding attribute set A from A'' using its own table T. Issuer proceeds to next step if π_s is verified and $\mathsf{Open}(pk, C, A, o) = 1$. Else, issuer outputs $\perp$ and halts.
3. Issuer chooses random values $u_1, \mathit{\Delta} \in \mathbb{Z}_p$, and sets
$$z_{u_1} = \mathsf{Enc}_{\mathsf{pk}}(u_1), \quad z_{\hat{x}} = \mathsf{Enc}_{\mathsf{pk}}(x + \mathit{\Delta}).$$
Issuer sends z_{u_1} and $z_{\hat{x}}$ to User.
4. User selects $t', u_2, \gamma_2 \xleftarrow{\$} \mathbb{Z}_p$. User computes
$$z_1 = \mathsf{Enc}_{\mathsf{pk}}(t' \cdot u_1 + (x + \mathit{\Delta}) \cdot u_2 - \gamma_2) = t' \cdot z_{u_1} + u_2 \cdot z_{\hat{x}} + \mathsf{Enc}_{\mathsf{pk}}(-\gamma_2),$$
$$\omega_2 = \gamma_2 + t' \cdot u_2.$$
User sends z_1, ω_2 and a ZK proof π_U of (t', u_2, γ_2) to Issuer.
5. After validating π_U, Issuer calculates
$$\gamma_1 = \mathsf{Dec}_{\mathsf{sk}}(z_1), \quad \omega_1 = \gamma_1 + (x + \mathit{\Delta}) \cdot u_1, \quad \beta = \omega_1 + \omega_2, \quad \delta_I = u_1 \cdot \beta^{-1}.$$
Issuer selects $s_2 \xleftarrow{\$} \mathbb{Z}_p$ and generates a partial SDH-CL signature for M as
$$sig' = (\mathit{\Delta}, s_2, v_1 = (Mb^{s_2}c)^{\delta_I}).$$
Issuer sends sig' and β to User.
6. User calculates $v_2 = (Mb^{s_2}c)^{\delta_U}$, where $\delta_U = u_2 \cdot \beta^{-1}$. User sets
$$t = t' + \mathit{\Delta}, \quad v = v_1 \cdot v_2 = (Mb^{s_2}c)^{\delta_I + \delta_U} = (Mb^{s_2}c)^{\frac{1}{(x+\mathit{\Delta})+t'}} = (Mb^{s_2}c)^{\frac{1}{x+t}}$$
and generates a full SDH-CL signature for M as $sig = (t, s_2, v)$.
If sig is not a valid signature on $A \cup \{o\}$, User outputs $\perp$ and stops. Else, user outputs the credential as $cred = (t, s, v, A = A \cup \{o\})$ where:
$$s = s_1 + s_2, \quad v = \left(a_0^{\prod_{j=1}^{n}(x'+m_j)} b^s c\right)^{1/(x+t)}.$$

4.3 Zero-Knowledge Proofs for Multi-Use Show Proof

Tan *et al.* [40] proposed the proof of possession, AND, ANY/OR, NAND/NOT and NANY show proofs concretely. The details of these proofs are remain unchanged with the aforementioned modifications, i.e. the presentation protocol $((\mathsf{Prove}(pk, cred, \phi_{\mathsf{stmt}}), \mathsf{Verify}(pk, \phi_{\mathsf{stmt}})) \to b$ unchanged, where $b \in \{0, 1\}$. Under the perspective of the verifier, there are no differences between the original t in Tan *et al.* [40] and the t in this work, the MtA protocol is independent to the verifier.

5 Security

5.1 Impersonation Resilience

In the security proof, we first classify 3 different types of adversary and use different simulation strategies for each of them to solve the SDH problem. In all proofs, we will extract the credential (s^*, t^*, v^*, A^*) that the adversary used to win the security game. Denote (s_i, t_i, v_i, A_i) as the credential used in the i-th Obtain or Verify oracle query. We have:

- Adversary $\mathcal{A}_1$: $t^* \neq t_i$ for all i.
- Adversary $\mathcal{A}_2$: $t^* = t_i$ for some i and $s^* \neq s_i$.
- Adversary $\mathcal{A}_3$: $t^* = t_i$ for some i and $s^* = s_i$.

We will use any of these adversaries to solve the SDH problem.

In the simulation, there is a special case that is common for all types of adversaries: $v^* = v_i$ for some i. We first describe it here and we do not repeat it throughout the simulation with $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_3$. This case is handled by solving the DLOG problem first. Assuming $M^* = \prod_{j=i}^{n}(x' + m_j^*)$ and $M_i = \prod_{j=i}^{n}(x' + m_{i,j})$ for attribute sets $A^* = \{m_j^*\}$ and $A_i = \{m_j\}$ respectively. Assume that $b = a_0^\tau$ and $c = a_0^\gamma$ for some $\tau, \gamma \in \mathbb{Z}_p$. Whenever v^* produced by $\mathcal{A}$ is the same as v_i produced by $\mathcal{C}$, the DLOG problem can be solved such that:

$$\begin{aligned} \because v^* &= v_i \\ (a_0^{M^*} b^{s^*} c)^{\frac{1}{x+t^*}} &\equiv (a_0^{M_i} b^{s_i} c)^{\frac{1}{x+t_i}} \\ (a_0^{M^* + s^*\tau + \gamma})^{\frac{1}{x+t^*}} &\equiv (a_0^{M_i + s_i\tau + \gamma})^{\frac{1}{x+t_i}} \\ \therefore \frac{M^* + s^*\tau + \gamma}{x + t^*} &\equiv \frac{M_i + s_i\tau + \gamma}{x + t_i} \quad \bmod p, \end{aligned}$$

which leads to:

$$x \equiv \frac{t^* M_i - t_i M^* + \tau(t^* s_i - t_i s^*) + \gamma(t^* - t_i)}{M^* - M_i + \tau(s^* - s_i)} \quad \bmod p.$$

$\mathcal{C}$ can solve the SDH problem using x if $M^* - M_i + \tau(s^* - s_i) \neq 0$. Next consider two cases for $M^* - M_i + \tau(s^* - s_i) = 0$.

1. $M^* \neq M_i$ or $s^* \neq s_i$: It happens with negligible probability of $1/p$ with the random choice of τ.
2. $M^* = M_i$ and $s^* = s_i$. This case only applies to $\mathcal{A}_1$ and $\mathcal{A}_3$ defined above. It implies $A^* = A_i$. In the security model, it is restricted that A^* cannot be the same as A_i used in the Obtain oracle. On the other hand, if $A^* = A_i$ used in the IssueTranscript queries or Verify queries. If the view of $\mathcal{A}$ is independent of the choice s_i of $\mathcal{C}$, there exists a probability of $1 - 1/p$ such that $s^* \neq s_i$, which means that it happens with negligible probability $1/p$ such that the simulation fails.

We present Lemma 1, 2 and 3 representing the adversaries $\mathcal{A}_1$, $\mathcal{A}_2$, and $\mathcal{A}_3$ as follows. Please refer to Appendix A for the proofs.

Theorem 1. *If an adversary $\mathcal{A}$ $(t_{\mathsf{imp}}, \epsilon_{\mathsf{imp}})$-breaks the* imp-aca*-security of the proposed anonymous credential system, then there exists an algorithm $\mathcal{C}$ which $(t_{\mathsf{cosdh}}, \epsilon_{\mathsf{cosdh}})$-breaks the co-SDH problem such that:*

$$\frac{\epsilon_{\mathsf{cosdh}}}{t_{\mathsf{cosdh}}} = \frac{\epsilon_{\mathsf{imp}}}{t_{\mathsf{imp}}},$$

or an algorithm $\mathcal{C}$ which $(t_{\mathsf{sdh}}, \epsilon_{\mathsf{sdh}})$-breaks the SDH problem such that:

$$\epsilon_{\mathsf{imp}} \leq \sqrt[N]{\sqrt{\epsilon_{\mathsf{sdh}}} - 1} + \frac{1 + (q-1)!/p^{q-2}}{p} + 1, \quad t_{\mathsf{imp}} \leq t_{\mathsf{sdh}}/2N - T(q^2),$$

where N is the total adversary instance, $q = Q_{(O,I)} + Q_{(P,V)}$ is the total query made to the Obtain *and* Verify *oracles, while $T(q^2)$ is the time parameterized by q to setup the simulation environment and to extract the SDH solution. Consider the dominant time elements t_{imp} and t_{sdh} only, we have:*

$$\left(1 - \left(1 - \epsilon_{\mathsf{imp}} + \frac{1 + (q-1)!/p^{q-2}}{p}\right)^N\right)^2 \leq \epsilon_{\mathsf{sdh}}, \quad 2Nt_{\mathsf{imp}} \approx t_{\mathsf{sdh}}.$$

Let $N = (\epsilon_{\mathsf{imp}} - \frac{1+(q-1)!/p^{q-2}}{p})^{-1}$, we get $\epsilon_{\mathsf{sdh}} \geq (1 - e^{-1})^2 \geq 1/3$ and the success ratio is:

$$\frac{\epsilon_{\mathsf{sdh}}}{t_{\mathsf{sdh}}} \geq \frac{1}{3 \cdot 2Nt_{\mathsf{imp}}}$$
$$\frac{6\epsilon_{\mathsf{sdh}}}{t_{\mathsf{sdh}}} \geq \frac{\epsilon_{\mathsf{imp}}}{t_{\mathsf{imp}}} - \frac{1 + (q-1)!/p^{q-2}}{t_{\mathsf{imp}}p}$$

which gives a tight reduction.

We follow the setting from Tan *et al.* [40] to use Multi-Instance Reset Lemma [25] as the knowledge extractor which requires an adversary $\mathcal{A}_1$ to run N parallel instances under active and concurrent attacks. The challenger can fulfil this requirement by simulating the $N-1$ instances from the SDH instance. It suffices to describe the simulation for a single instance of impersonation. Our security reduction proof is as follows.

Lemma 1. *If an adversary $\mathcal{A}_1$ $(t_{\mathsf{imp}}, \epsilon_{\mathsf{imp}})$-breaks the* imp-aca*-security of the proposed anonymous credential system, then there exists an algorithm $\mathcal{C}$ which $(t_{\mathsf{cosdh}}, \epsilon_{\mathsf{cosdh}})$-breaks the co-SDH problem such that:*

$$\epsilon_{\mathsf{imp}} \leq \sqrt[N]{\sqrt{\epsilon_{\mathsf{sdh}}} - 1} + \frac{1 + (q-1)!/p^{q-2}}{p} + 1, \quad t_{\mathsf{imp}} \leq t_{\mathsf{sdh}}/2N - T(q^2),$$

where N is the total adversary instance, $q = Q_{(O,I)} + Q_{(P,V)}$ is the total number of query made to the Obtain *and* Verify *oracles, while $T(q^2)$ is the time parameterized by q to setup the simulation environment and to extract the SDH solution.*

Lemma 2. *If an adversary $\mathcal{A}_2$ $(t_{\mathsf{imp}}, \epsilon_{\mathsf{imp}})$-breaks the* imp-aca*-security of the proposed anonymous credential system, then there exists an algorithm $\mathcal{C}$ which $(t_{\mathsf{cosdh}}, \epsilon_{\mathsf{cosdh}})$-breaks the co-SDH problem such that:*

$$\epsilon_{\mathsf{imp}} \leq \sqrt[N]{\sqrt{\epsilon_{\mathsf{sdh}}} - 1} + \frac{1 + (q-1)!/p^{q-2}}{p} + 1, \quad t_{\mathsf{imp}} \leq t_{\mathsf{sdh}}/2N - T(q^2),$$

where N is the total adversary instance, $q = Q_{(O,I)} + Q_{(P,V)}$ is the total number of query made to the Obtain *and* Verify *oracles, while $T(q^2)$ is the time parameterized by q to setup the simulation environment and to extract the SDH solution.*

Lemma 3. *If an adversary $\mathcal{A}_3$ $(t_{\mathsf{imp}}, \epsilon_{\mathsf{imp}})$-breaks the* imp-aca*-security of the proposed anonymous credential system, then there exists an algorithm $\mathcal{C}$ which $(t_{\mathsf{cosdh}}, \epsilon_{\mathsf{cosdh}})$-breaks the co-SDH problem such that:*

$$\epsilon_{\mathsf{imp}} \leq \sqrt[N]{\sqrt{\epsilon_{\mathsf{sdh}}} - 1} + \frac{1 + (q-1)!/p^{q-2}}{p} + 1, \quad t_{\mathsf{imp}} \leq t_{\mathsf{sdh}}/2N - T(q^2),$$

where N is the total adversary instance, $q = Q_{(O,I)} + Q_{(P,V)}$ is the total number of query made to the Obtain *and* Verify *oracles, while $T(q^2)$ is the time parameterized by q to setup the simulation environment and to extract the SDH solution.*

Combining Theorem 3 from [40], Lemma 1, 2, and 3 in this work gives Theorem 1 as required.

5.2 Anonymity

In this security proof, we prove that the proposed interactive anonymous credential system achieves anonymity under active and concurrent attack (anon-aca). In our modification, the Issuer contains the attribute sets of Users, therefore, the anonymity among the issuing protocol is exploited, while we keep the anonymity among the presentation protocol.

Before proving the anonymity and unlinkability of the new proposed interactive ABC system, we note that committed attributes and the randomized credentials maintain the properties of perfectly hiding. Similarly, the issuing protocol and the presentation protocol achieve self-reducibility [40]. Please refer to Appendix B for the proof.

Theorem 2. *The proposed interactive anonymous credential system is* anon-aca*-secure under the presentation protocol.*

5.3 Unlinkability

With a similar approach, we prove that the proposed interactive anonymous credential system achieves unlinkability under active and concurrent attack (unl-aca). As aforementioned, the unlinkability among the issuing protocol is exploited, while we keep the unlinkability among the presentation protocol. Please refer to Appendix C for the proof.

Theorem 3. *The proposed interactive anonymous credential system is* unl-aca*-secure under the presentation protocol.*

6 Single-Use Credential and Anonymity Proof

Recall that a user credential *cred* is an SDH-CL signature on the MoniPoly Commitment C of his attribute set A and opening o (i.e., $C \leftarrow \mathsf{MoniPoly.Commit}(pk, A, o)$). The show proofs of a multi-use ABC system is a proof of knowledge for some predicate $Pred = \{\mathsf{Intersection}, \mathsf{Difference}\}$ with respect to an attribute set A' and length l:

$$\begin{aligned} PK\{(cred, C, P, W) : & 1 = \mathsf{SDH{-}CL.Verify}(pk, C, cred) \wedge \\ & 1 = \mathsf{MoniPoly.Verify}Pred(pk, C, P, W, (A', l))\}, \end{aligned}$$

where $(P, W) \leftarrow \mathsf{MoniPoly.Open}Pred(pk, C, A, o, (A', l))$.

For the credential system in [40], the credential is $cred = (t, s, v)$, in which t, s, v are all known to the issuer. Hence, a zero-knowledge proof on (t, s, v) should be used in order to provide anonymity against the issuer, even if the credential is used once only. Hence, the zero-knowledge proof in [40] is complicated if we apply it to a single-use credential system.

6.1 Proof of Possession of Single-Use Credential

In our Issue and Obtain protocol, the issuer and the user runs a two-party computation protocol, such that the issuer cannot obtain any information about C and $cred = (t, s, v)$ for the attribute set A. As a result, we can formulate an efficient show proof for a single-use credential *cred*.

In order to give a show proof to an attribute set A' and length l, the prover first runs $(P, W) \leftarrow \mathsf{MoniPoly.Open}Pred(pk, C, A, o, (A', l))$ for some predicate $Pred = \{\mathsf{Intersection}, \mathsf{Difference}\}$. The prover can simply output $(cred, C, P, W)$ as the show proof. The verifier can validate the show by checking if $1 = \mathsf{SDH\text{-}CL.Verify}(pk, C, cred)$ and $1 = \mathsf{MoniPoly.Verify}Pred(pk, C, P, W, (A', l))$.

6.1.1 AND Proof. We give a simple demonstration on a single AND clause. The prover can prove its ownership without disclosing any attribute as follows.

1. Verifer requests a proof of possession with an AND proof for the attribute set $A' = \{m_1, \dots, m_l\}$ with length l. Denote $\{\mathsf{m}_j\}_{0 \le j \le l} = \mathsf{MPEncode}(A')$.
2. If A' is not a subset of the prover's attribute set A (where $|A| = n$), the prover aborts and the verifier outputs 0.
3. The prover computes $\{\mathsf{w}'_j\}_{0 \le j \le n-l} = \mathsf{MPEncode}(A - A')$ and sets $W = \prod_{j=0}^{n-l} a_j^{\mathsf{w}'_j}$. The prover outputs the intersection set $P = A'$, the witness W, the commitment C and the credential (t, s, v).
4. The verifier outputs 1 if:

$$\hat{e}(W, \prod_{j=0}^{l} X_j^{\mathsf{m}_j}) = \hat{e}(C, X_0).$$

Otherwise, it outputs 0. The correctness for the equation is shown in Appendix E.1.

6.1.2 ANY and OR Proof. We give a simple implementation on a single ANY/OR clause, proving that the prover has l attributes $\{m_j\}_{1\leq j\leq l} \in (A' \cap A)$, where $|A| = n$ and $|A'| = k$. The prover can prove its ownership without disclosing any attribute as follows.

1. Verifier requests a proof of possession with an ANY/OR proof for the attribute set $A' = \{m_1, \dots, m_k\}$ with length k.
2. The prover tries to generate a l-attribute intersection set $I \in (A' \cap A)$, the prover aborts and the verifier outputs 0 if no such I can be formed.
3. The prover computes $\{\mathsf{w}'_j\}_{0\leq j\leq n-l} = \mathsf{MPEncode}(A - I)$ and sets $W = \prod_{j=0}^{n-l} a_j^{\mathsf{w}'_j}$. Also, the prover computes $\{\mathsf{m}'_{2,j}\}_{0\leq j\leq k-l} = \mathsf{MPEncode}(A' - I)$ and sets $W' = \prod_{j=0}^{k-l} a_j^{\mathsf{m}'_{2,j}}$. The prover outputs the witness W, W', the outputs of $\mathsf{MPEncode}(I)$, the commitment C and the credential (t, s, v).
4. The verifier outputs 1 if:

$$\hat{e}(W'W, \prod_{j=0}^{l} X_j^{\iota_j}) = \hat{e}(C \cdot \prod_{j=0}^{k} a_j^{\mathsf{m}_{1,j}}, X_0)$$

   where $\{\mathsf{m}'_{1,j}\}_{0\leq j\leq k} = \mathsf{MPEncode}(A')$, $\{\iota'_j\}_{0\leq j\leq l} = \mathsf{MPEncode}(I)$. Otherwise, it outputs 0. The correctness for the equation is shown in Appendix E.2.

6.1.3 NAND and NOT Proof. We give a simple demonstration on a single NAND clause, proving that an attribute set A' is disjoint with the set A in his credential, where $|A| = n$ and $|A'| = k$. If $|A'| = 1$, the NAND proof becomes a NOT proof. The prover can prove its ownership without disclosing any attribute as follows.

1. Verifier requests a proof of possession with an NAND proof for the attribute set $A' = \{m_1, \dots, m_k\}$ with length k. Denote $\{\mathsf{m}_j\}_{0\leq j\leq k} = \mathsf{MPEncode}(A')$.
2. If $|A' - A| < k$, the prover aborts and the verifier outputs 0.
3. The prover computes $(\{\mathsf{w}'_j\}_{0\leq j\leq n-k}, \{\mathsf{r}'_j\}_{0\leq j\leq k-1}) = \mathsf{MPEncode}(A)/\mathsf{MPEncode}(A')$, and set $W = \prod_{j=0}^{n-k} a_j^{\mathsf{w}'_j}$. The prover outputs the witness $(W, \{\mathsf{r}'_j\}_{0\leq j\leq k-1})$, the commitment C and the credential (t, s, v) to the verifier.
4. The verifier outputs 1 if:

$$\prod_{j=0}^{k-1} a_j^{\mathsf{r'}_j} \neq 1_{\mathbb{G}_1}, \quad \hat{e}(W, \prod_{j=0}^{k} X_j^{\mathsf{m}_j}) = \hat{e}(C \cdot \prod_{j=0}^{k-1} a_j^{-\mathsf{r'}_j}, X_0)$$

   Otherwise, it outputs 0. The correctness for the equation is shown in Appendix E.3.

6.1.4 NANY Proof. We give a simple implementation on a single NANY clause, proving that the prover has an l-attribute set $D \subseteq (A' - A)$ are not in the credential, where $|A| = n$ and $|A'| = k$. The prover can prove its ownership without disclosing any attribute as follows.

1. Verifier requests a proof of possession with an NANY proof for the attribute set $A' = \{m_1, \ldots, m_k\}$ with length k.
2. Prover generates a l-attribute difference set $D \in (A' - A)$. The prover aborts and the verifier outputs 0 if no such D can be formed.
3. The prover computes $(\{\mathsf{w}'_j\}_{0 \leq j \leq n-l}, \{\mathsf{r}'_j\}_{0 \leq j \leq l-1}) = \mathsf{MPEncode}(A)/\mathsf{MPEncode}(D)$, and set $W = \prod_{j=0}^{n-l} a_j^{\mathsf{w}'_j}$. Also, the prover computes $\{\mathsf{m}'_{2,j}\}_{0 \leq j \leq k-l} = \mathsf{MPEncode}(A' - D)$ and sets $W' = \prod_{j=0}^{k-l} a_j^{\mathsf{m}'_{2,j}}$. The prover outputs the witness $(W', W, \{\mathsf{r}'_j\}_{0 \leq j \leq l-1})$, the commitment C and the credential (t, s, v).
4. With $\{\mathsf{m}'_{1,j}\}_{0 \leq j \leq k} = \mathsf{MPEncode}(A')$, $\{\delta_j\}_{0 \leq j \leq l} = \mathsf{MPEncode}(D)$, the verifier outputs 1 if:

$$\prod_{j=0}^{l-1} a_j^{\mathsf{r'}_j} \neq 1_{\mathbb{G}_1}, \quad \hat{e}(W'W, \prod_{j=0}^{l} X_j^{\delta_j}) = \hat{e}(C \cdot \prod_{j=0}^{k} a_j^{\mathsf{m}_{1,j}} \cdot \prod_{j=0}^{l-1} a_j^{-\mathsf{r'}_j}, X_0)$$

   Otherwise, it outputs 0. The correctness for the equation is shown in Appendix E.4.

6.2 Proof of Anonymity

In this subsection, we give the full prove of anonymity towards the aforementioned single use credentials. Please refer to Appendix D for the proof.

Theorem 4. *The proposed single use interactive anonymous credential is* $\mathsf{anon\text{-}aca}$*-secure under the presentation protocol.*

6.3 Batch the Single-Use Credentials

The single-use credential can only be used once in order to maintain the unlinkability. Therefore, under the scenario that the single-use credentials are being adopted, the batched version of our proposed credential system may alleviate the tedious operations of creating credentials one by one. Suppose the User would like to invoke l creations of the proposed single-use credentials, using the same attribute set. This action could be done by batch MtA, with only one instance of the key generation algorithm. Instead of sending a single value of a variable in each communication, a list (vector) with length l of each variable could be used in the batched single-use credential issuance. This approach could decrease the number of communication rounds from $4l$ to 4.

7 Conclusion and Further Extensions

In this work, we optimize the existing q-SDH-based attribute-based anonymous credential system with an interactive credential issuance. Moreover, we further design the show proofs between credentials in a single-use manner. We further point out that our interactive approach could be extended to other q-SDH-based credential systems.

Interactive setup. In this work, we proposed an interactive approach between the Issuer and User under the q-SDH-based anonymous credential system [40]. This setup decentralizes the original centralized approach in the existing literature. The interactive approach could be applicable to other q-SDH-based anonymous credential systems and maybe some related protocols. Here, we give an example of the possible protocol which is the q-SDH-based direct anonymous attestation [6]. However, due to the limitation of the computation power of the TPM, the interactive setup could not be adopted at this moment.

From the perspective of security requirements, we exploit the full anonymity, full attribute unlinkability, and full protocol unlinkability in [40], since our interactive approach requires both the issuer and the user to know the content inside a particular credential. With this setting, we further prove that our protocol achieves anonymity and unlinkability among the presentation protocol.

References

1. Aikou, Y., Sadiah, S., Nakanishi, T.: An efficient blacklistable anonymous credentials without ttps using pairing-based accumulator. In: 2017 IEEE 31st International Conference on AINA. pp. 780–786 (2017)
2. Anada, H.: Decentralized multi-authority anonymous credential system with bundled languages on identifiers. In: Maimut, D., Oprina, A.G., Sauveron, D. (eds.) SecITC 2020. p. 71–90. Springer (2021)
3. Begum, N., Nakanishi, T.: An accumulator-based revocation in delegatable anonymous credentials. In: CANDARW 2020. pp. 314–320 (2020)
4. Blömer, J., Bobolz, J.: Delegatable attribute-based anonymous credentials from dynamically malleable signatures. In: Preneel, B., Vercauteren, F. (eds.) ACNS. p. 221–239. Springer (2018)
5. Camenisch, J., Drijvers, M., Dzurenda, P., Hajny, J.: Fast keyed-verification anonymous credentials on standard smart cards. In: Dhillon, G., Karlsson, F., Hedström, K., Zúquete, A. (eds.) IFIP SEC. p. 286–298. Springer (2019)
6. Camenisch, J., Drijvers, M., Lehmann, A.: Anonymous attestation using the strong diffie hellman assumption revisited. In: Franz, M., Papadimitratos, P. (eds.) TRUST. pp. 1–20. Springer, Cham (2016)
7. Camenisch, J., Lysyanskaya, A.: Signature schemes and anonymous credentials from bilinear maps. In: Franklin, M. (ed.) CRYPTO 2004. p. 56–72 (2004)
8. Casanova-Marqués., R., Pascacio., P., Hajny., J., Torres-Sospedra., J.: Anonymous attribute-based credentials in collaborative indoor positioning systems. In: SECRYPT,. pp. 791–797. INSTICC, SciTePress (2021)
9. Chase, M., Perrin, T., Zaverucha, G.: The signal private group system and anonymous credentials supporting efficient verifiable encryption. p. 1445–1459. CCS '20, ACM, New York, NY, USA (2020)
10. Chatterjee, S., Menezes, A.: On cryptographic protocols employing asymmetric pairings — the role of ψ revisited. Discrete Applied Mathematics **159**(13), 1311–1322 (2011)
11. Chaum, D.: Security without identification: Transaction systems to make big brother obsolete. Commun. ACM **28**(10), 1030–1044 (1985)

12. Connolly, A., Lafourcade, P., Perez Kempner, O.: Improved constructions of anonymous credentials from structure-preserving signatures on equivalence classes. In: Hanaoka, G., Shikata, J., Watanabe, Y. (eds.) Public-Key Cryptography – PKC 2022. pp. 409–438. Springer, Cham (2022)
13. Crites, E.C., Lysyanskaya, A.: Delegatable anonymous credentials from mercurial signatures. In: Matsui, M. (ed.) Topics in Cryptology – CT-RSA 2019. p. 535–555. Springer (2019)
14. Davidson, A., Goldberg, I., Sullivan, N., Tankersley, G., Valsorda, F.: Privacy pass: Bypassing internet challenges anonymously. Proceedings on Privacy Enhancing Technologies **2018**, 164 – 180 (2018)
15. Deng, X., Tian, C., Chen, F., Xian, H.: Designated-verifier anonymous credential for identity management in decentralized systems. Mobile Information Systems **2021**, article ID 2807395
16. Dzurenda, P., Hajny, J., Malina, L., Ricci, S.: Anonymous credentials with practical revocation using elliptic curves. In: SECRYPT 2017. pp. 534–539
17. Fuchsbauer, G., Hanser, C., Slamanig, D.: Structure-preserving signatures on equivalence classes and constant-size anonymous credentials. JOC 2019 **32**, 498–546
18. Fuchsbauer, G., Hanser, C., Slamanig, D.: Practical round-optimal blind signatures in the standard model. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. p. 233–253 (2015)
19. Galbraith, S.D., Paterson, K.G., Smart, N.P.: Pairings for cryptographers. Discret. Appl. Math. **156**, 3113–3121 (2006)
20. Gennaro, R., Goldfeder, S.: Fast multiparty threshold ecdsa with fast trustless setup. In: CCS 2018. p. 1179–1194. ACM (2018)
21. Guo, N., Gao, T., Park, H.: Random oracle-based anonymous credential system for efficient attributes proof on smart devices. Soft Computing **20**, 1781–1791 (2016)
22. Haböck, U., Krenn, S.: Breaking and fixing anonymous credentials for the cloud. In: Mu, Y., Deng, R.H., Huang, X. (eds.) CANS. p. 249–269. Springer (2019)
23. Hanzlik, L., Slamanig, D.: With a little help from my friends: Constructing practical anonymous credentials. In: CCS 2021. p. 2004–2023. CCS '21, ACM (2021)
24. Hébant, C., Pointcheval, D.: Traceable constant-size multi-authority credentials. Cryptology ePrint Archive, Paper 2020/657 (2020), `https://eprint.iacr.org/2020/657`
25. Kiltz, E., Masny, D., Pan, J.: Optimal security proofs for signatures from identification schemes. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. pp. 33–61. Springer Berlin Heidelberg, Berlin, Heidelberg (2016)
26. Kreuter, B., Lepoint, T., Orrù, M., Raykova, M.: Anonymous tokens with private metadata bit. In: CRYPTO 2020. p. 308–336. Springer-Verlag (2020)
27. Krzywiecki, Ł., Wszoła, M., Kutyłowski, M.: Brief announcement: Anonymous credentials secure to ephemeral leakage. In: Dolev, S., Lodha, S. (eds.) CSCML. p. 96–98. Springer (2017)
28. Lin, C., He, D., Zhang, H., Shao, L., Huang, X.: Privacy-enhancing decentralized anonymous credential in smart grids. Computer Standards & Interfaces **75** (2021), article ID: 103505
29. Liu, D., Wu, H., Ni, J., Shen, X.: Efficient and anonymous authentication with succinct multi-subscription credential in sagvn. IEEE Transactions on Intelligent Transportation Systems **23**(3), 2863–2873 (2022)
30. Nakanishi, T., Kanatani, T.: Efficient blacklistable anonymous credential system with reputation using a pairing-based accumulator. IET Information Security **14**(6), 613–624 (2020)

31. Okishima, R., Nakanishi, T.: An anonymous credential system with constant-size attribute proofs for cnf formulas with negations. In: Attrapadung, N., Yagi, T. (eds.) IWSEC. p. 89–106. Springer (2019)
32. Paquin, C., Zaverucha, G.: U-prove cryptographic specification v1.1 (revision 3) (December 2013), https://www.microsoft.com/en-us/research/publication/u-prove-cryptographic-specification-v1-1-revision-3/
33. Pinjala, S.K., Vivek, S.S., Sivalingam, K.M.: Delegated anonymous credentials with revocation capability for iot service chains (dancis). IEEE Internet of Things Journal **9**(5), 3729–3742 (2022)
34. Pussewalage, H.S.G., Oleshchuk, V.A.: An anonymous delegatable attribute-based credential scheme for a collaborative e-health environment. ACM Trans. Internet Technol. **19**(3), 1–22 (sep 2019)
35. Rondelet, A.: A note on anonymous credentials using BLS signatures. CoRR **abs/2006.05201** (2020), https://arxiv.org/abs/2006.05201
36. Samarati, P., Sweeney, L.: Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Tech. rep. (1998)
37. Sanders, O.: Efficient redactable signature and application to anonymous credentials. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) PKC 2020. p. 628–656 (2020)
38. Schäge, S.: Tight proofs for signature schemes without random oracles. In: EUROCRYPT. p. 189–206. EUROCRYPT'11, Springer-Verlag, Berlin, Heidelberg (2011)
39. Srinivas, J., Das, A.K., Kumar, N., Rodrigues, J.J.P.C.: Tcalas: Temporal credential-based anonymous lightweight authentication scheme for internet of drones environment. IEEE Transactions on Vehicular Technology **68**(7), 6903–6916 (2019)
40. Tan, S.Y., Groß, T.: Monipoly—an expressive q-sdh-based anonymous attribute-based credential system. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020. p. 498–526. Springer (2020)
41. Wang, W., Liu, J., Qin, Y., Feng, D.: Formal analysis of a ttp-free blacklistable anonymous credentials system. In: Qing, S., Mitchell, C., Chen, L., Liu, D. (eds.) Information and Communications Security. pp. 3–16. Springer, Cham (2018)
42. Yang, R., Au, M.H., Xu, Q., Yu, Z.: Decentralized blacklistable anonymous credentials with reputation. Computers & Security **85**, 353–371 (2019)

A Proof of Theorem 1

A.1 Proof of Lemma 1

Proof. In this proof, we show that if $\mathcal{A}_1$ exists, there exists an algorithm $\mathcal{C}$ which output $(g_1^{\frac{1}{x+t}}, t)$ by acting the simulator for the ABC system. Given a q-SDH instance $(g_1, g_1^x, g_1^{x^2}, ..., g_1^{x^q}, g_2, g_2^x)$ where $q = Q_{(O,I)} + Q_{(P,V)}$ is the number of queries made to the Obtain and Verify oracles. The reduction games are as follows:

Game$_0$. Let S be the event of a successful impersonation. Attacking by $\mathcal{A}$ on N real instances of anonymous credential system, by assumption, we have:

$$\Pr[S_0] = \epsilon_{\mathsf{imp}}. \tag{1}$$

Game$_1$. This game simulates the environment of the modified ABC system. $\mathcal{C}$ uniformly selects unique $t_0, t_0', t_0'', x', t_1, ..., t_q \in \mathbb{Z}_p^*$. Next, let $f(x)$ be the polynomial $f(x) = \prod_{k=1}^{q}(x + t_k) := \sum_{k=0}^{q} \rho_k x^k$ for some coefficient ρ_k, $f_i(x)$ be the polynomial $f_i(x) = \prod_{k=1,k\neq i}^{q}(x + t_k) := \sum_{k=0}^{q-1} \lambda_k x_k$ for some coefficient λ_k, and thus $g_1^{f(x)} = \prod_{k=0}^{q}(g_1^{x^k})^{\rho_k}$. $\mathcal{C}$ additionally generates the key pair $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{E.KeyGen}()$ under the key generation protocol of homomorphic cryptosystem E and sends $(\mathsf{e}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, a_0 = g_1^{f(x)t_0}, a_1 = a_0^{x'}, ..., a_n = a_0^{x'^n}, b = g_1^{f(x)t_0'}, c = g_1^{f(x)t_0''}, X = g_2^x, X_0 = g_2, X_1 = X_0^{x'}, ..., X_n = X_0^{x'^n}, \mathsf{pk})$ as the public key to $\mathcal{A}_1$. $\mathcal{C}$ also creates two empty lists $L_{(O,I)}$ and $L_{(P,V)}$ which stores the corrupted credentials simulated during the issuing protocol and the non-corrupted credentials simulated during the presentation protocol, respectively. Note that t_0, t_0', t_0'', x' are uniformly random (including the corresponding random self-reducible $N-1$ instances and the variables in the homomorphic encryption schemes), the distribution of the simulated public key is the same as that of the original scheme, thus:

$$\Pr[S_1] = \Pr[S_0]. \tag{2}$$

Game$_2$. $\mathcal{A}_1$ plays the role of multiple users in this game, and concurrently interact with the issuer simulated by $\mathcal{C}$. We assume every user i uses different attribute set A_i without loss of generality. $\mathcal{C}$ produces a credential $cred_i$ for $\mathcal{A}_1$'s chosen $A_i = \{m_{1,i}, ...m_{n-1,i}, o_i\}$, if the i-th session of an issuing protocol ends successfully. Their interaction is as follows:

1. $\mathcal{A}_1$ concurrently initializes the issuing protocol with $\mathcal{C}$ by running the zero-knowledge protocol $\pi_{s,i}$:

   $$PK\{s_{1,i} : M_i = C_i \cdot b^{s_{1,i}}\}$$

   $\mathcal{A}_1$ sends $(A_i'', M_i, C_i, o_i, \pi_{s,i})$ to $\mathcal{C}$. If the ZK proof $\pi_{s,i}$ is valid, $\mathcal{C}$ can successfully extract the secret exponents $s_{1,i}$ used by $\mathcal{A}_1$ in the protocol.
2. $\mathcal{C}$ validates (A_i'', C_i, o_i) with its own data set T. The protocol proceed if $\mathsf{Open}(pk, C_i, A_i, o_i) = 1$ where A_i is generated locally by $\mathcal{C}$ using A_i'' and T.
3. $\mathcal{C}$ chooses a random value $u_{1,i} \in \mathbb{Z}_p$, and sets

   $$z_{u_{1,i}} = \mathsf{Enc}_{\mathsf{pk}}(u_{1,i}).$$

   $\mathcal{C}$ chooses a random ciphertext $z_{\hat{x},i}$. $\mathcal{C}$ sends $(z_{u_{1.i}}, z_{\hat{x},i})$ to $\mathcal{A}_1$.
4. $\mathcal{A}_1$ sends $z_{1,i}, \omega_{2,i}$ and a ZK proof $\pi_{U,i}$ to $\mathcal{C}$. By the knowledge extractor of the ZK proof, $\mathcal{C}$ can obtain $(t_i', u_{2,i}, \gamma_{2,i})$.

5. $\mathcal{C}$ picks a random $\beta_i, s_{2,i} \xleftarrow{\$} \mathbb{Z}_p$, calculates $\Delta_i = t_i - t'_i$ and generates a partial SDH-CL signature for M_i as $sig'_i = (\Delta_i, s_{2,i}, v_{1,i})$, where

$$\begin{aligned} v_{1,i} &= (M_i b^{s_{2,i}} c)^{\delta_{I,i}} \\ &= (\prod_{j=0}^{n} a_j^{\alpha_{j,i}} b^{\sigma_i} b^{s_{2,i}} c)^{\frac{1}{x+t_i} - \delta_{U,i}} \\ &= (\prod_{j=0}^{n} g_1^{f(x)t_0 x'^j \alpha_{j,i} + f(x)t'_0(\sigma_i + s_{2,i}) + f(x)t''_0})^{\frac{1}{x+t_i} - \frac{u_{2,i}}{\beta_i}} \\ &= (\prod_{j=0}^{n} g_1^{f(x)[t_0 x'^j \alpha_{j,i} + t'_0(\sigma_i + s_{2,i}) + t''_0]})^{\frac{1}{x+t_i} - \frac{u_{2,i}}{\beta_i}} . \end{aligned}$$

Observe that the above $v_{1,i}$ can be calculated by using $g_1^{f_i(x)}$. $\mathcal{C}$ sends sig'_i and β_i to $\mathcal{A}_1$.
Since $\mathcal{C}$ has extracted t'_i and $u_{2,i}$ from the ZK proof $\pi_{U,i}$, $\mathcal{C}$ can also calculate the full SDH-CL signature (t_i, s_i, v_i). $\mathcal{C}$ will check if $(\mathsf{m}_{0,i}, ..., \mathsf{m}_{n,i}, t_i, s_i, v_i) \in L_{(P,V)}$, $\mathcal{C}$ removes it from $L_{(P,V)}$ and adds to $L_{(O,I)}$.

Since $\mathcal{C}$'s choices of $s_{2,i}$ are independent of $\mathcal{A}_1$'s view, we have $s_i \neq s_j$ for some $i, j \leq q$ with overwhelming probability. Since $t_i \neq t_j$, a collision $v_i = v_j$ for some $i, j \leq q$ in $\mathcal{A}$'s concurrent queries happens with a negligible probability of $\Pr[Col] = 1/p$ in which $\mathcal{A}_1$ can compute the discrete logarithm x. Else, $\mathcal{C}$ simulates the issuing protocol perfectly for every concurrent query and $\mathcal{A}_1$ can formulate its credential $cred_i$ as in the original issuing protocol. This gives:

$$\begin{aligned} \Pr[S_2] &= \Pr[S_1] + \Pr[Col] \\ &\leq \Pr[S_1] + \prod_{i=1}^{q-1} 1/p \\ &\leq \Pr[S_1] + (q-1)!/p^{q-1}. \end{aligned} \tag{3}$$

Game$_3$. This game $\mathcal{C}$ plays as a verifier and $\mathcal{A}_1$ plays as multiple provers which concurrently interact with $\mathcal{C}$. Without loss of generality, assume that every prover i uses a valid $cred_i$ to execute the corresponding show proof on $\phi_{\mathsf{stmt_i}}$ such that $\phi_{\mathsf{stmt_i}}(A_i) = 1$. $\mathcal{C}$ simulates Verify orcale accordingly and thus:

$$\Pr[S_3] = \Pr[S_2]. \tag{4}$$

Game$_4$. This game $\mathcal{A}_1$ plays as a verifier and $\mathcal{C}$ plays as multiple provers which concurrently interact with $\mathcal{A}_1$. $\mathcal{C}$ interacts with $\mathcal{A}_1$ using a $cred_i$ where $\phi_{\mathsf{stmt_i}}(A_i) = 1$ when $\mathcal{A}_1$ requests for a show proof. With the assumption that $\mathcal{C}$ has appropriate credentials for these queries already. Also, $\mathcal{C}$ simulates $(\mathsf{m}_{0,i}, ..., \mathsf{m}_{n,i}, \Delta_i, s_{2,i}, v_{1,i})$ as in **Game**$_2$ and adds it to $L_{(P,V)}$ before the interaction with $\mathcal{A}_1$. This results:

$$\Pr[S_4] = \Pr[S_3]. \tag{5}$$

Game$_5$. $\mathcal{A}_1$ tries to impersonate the prover in this game. The attribute set of the prover is $A^* = \{m_1^*, ..., m_n^*\} \neq A_i \in L_{(O,I)}$ using the access policy $\phi^*_{\mathsf{stmt_i}}$ where $\phi^*_{\mathsf{stmt_i}}(A^*) = 1$ and $\phi^*_{\mathsf{stmt_i}}(A_i) = 0$. $\mathcal{A}_1$ is still allowed to query the oracles as in **Game**$_2$, **Game**$_3$ and **Game**$_4$ with restriction that $\phi^*_{\mathsf{stmt_i}}(A_i) \neq 1$ for the Obtain oracle. The aim of $\mathcal{A}_1$ is to complete the proof with $(\mathcal{A}_1^{\mathsf{Prove}}(pk, \cdot, \phi^*_{\mathsf{stmt_i}}(A^*)), \mathcal{C}^{\mathsf{Verify}}(pk, \phi^*_{\mathsf{stmt_i}}(A^*))) = 1$. $\mathcal{C}$ may further obtain two valid transcripts and regenerate the secret values to extract the credentials components (t^*, s^*, v^*) if the show proof could be verified again after resetting $\mathcal{A}_1$ by $\mathcal{C}$ to the time after sending witnesses.

$\mathcal{A}_1$ is required to output $t^* \notin \{t_1, ..., t_q\}$. If $v^* \notin L_{(O,I)} \cup L_{(P,V)}$, $\mathcal{C}$ can construct a polynomial $c(x)$ of degree $n-1$ where $f(x) = c(x)(x+t^*) + d$ to compute:

$$\begin{aligned} v^{*\frac{1}{(t_0 \sum_{j=0}^{n} x'^j \mathsf{m}_j^* + t_0' s^* + t_0'') \cdot d}} g_1^{\frac{-c(x)}{d}} &= g_1^{\frac{f(x) \cdot (t_0 \sum_{j=0}^{n} x'^j \mathsf{m}_j^* + t_0' s^* + t_0'')}{(x+t^*) \cdot (t_0 \sum_{j=0}^{n} x'^j \mathsf{m}_j^* + t_0' s^* + t_0'') \cdot d} - \frac{c(x)}{d}} \\ &= g_1^{\frac{c(x)(x+t^*)+d}{d \cdot (x+t^*)} - \frac{c(x)}{d}} \\ &= g_1^{\frac{1}{x+t^*}} \end{aligned}$$

and outputs $(g^{\frac{1}{x+t^*}}, t^*)$ as the solution for the SDH instance. On the other hand, if we have $v^* \in L_{(O,I)} \cup L_{(P,V)}$, $\mathcal{C}$ can extract the discrete logarithm x to break the SDH assumption.

Let $\Pr[Res]$ be the probability of $\mathcal{C}$ resets successfully, and $\Pr[Acc]$ be the probability of $\mathcal{C}$ outputs 1 in the presentation protocol with $\mathcal{A}_1$, by Multi-Instance Reset Lemma [25], we have:

$$\begin{aligned} \Pr[S_5] &\leq \Pr[S_4] + \Pr[Acc] \\ &\leq \Pr[S_4] + \sqrt[N]{\Pr[Res] - 1} + \frac{1}{p} + 1 \\ &\leq \Pr[S_4] + \sqrt[N]{\sqrt{\epsilon_{\mathsf{sdh}}} - 1} + \frac{1}{p} + 1, \end{aligned} \tag{6}$$

and summing up the probability from (1) to (6), we have $\epsilon_{\mathsf{imp}} \leq \sqrt[N]{\sqrt{\epsilon_{\mathsf{sdh}}} - 1} + \frac{1 + (q-1)!/p^{q-2}}{p} + 1$ as required. The time taken by $\mathcal{C}$ is at least $2Nt_{\mathsf{imp}}$ due to reset and interacting with N parallel impersonation instances, in additional to the environment setup and the final SDH soltion extraction that cost $T(q^2)$. □

A.2 Proof of Lemma 2

Proof. In this proof, we show that if $\mathcal{A}_2$ exists, there exists an algorithm $\mathcal{C}$ which output $(g_1^{\frac{1}{x+t}}, t)$ by acting the simulator for the ABC system. Given a q-SDH instance $(g_1, g_1^x, g_1^{x^2}, ..., g_1^{x^q}, g_2, g_2^x)$ where $q = Q_{(O,I)} + Q_{(P,V)}$ is the number of queries made to the Obtain and Verify oracles. The reduction games are as follows:

Game$_0$. There are no differences between this game and **Game**$_0$ in Lemma 1 such that:

$$\Pr[S_0] = \epsilon_{\mathsf{imp}}. \tag{7}$$

Game$_1$. This game follows **Game**$_1$ in Lemma 1 with exceptions that $\mathcal{C}$ additionally checks $X = g_2^{-t_i}$ for $i \in \{1, ..., q\}$. If t_i is found, $\mathcal{C}$ outputs the solution towards the SDH instance using the discrete logarithm $x = -t_i$. $\mathcal{C}$ also computes $f_{i,j}(x) = \prod_{k=1,k\neq i,j}^{q}(x+t_k) = \sum_{k=0}^{q-2}\gamma_k x^k$ and uniformly selects random distinct $s_1, ..., s_q \in \mathbb{Z}_p$. $\mathcal{C}$ sends $(\mathsf{e}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, a_0 = g_1^{f(x)t_0}, a_1 = a_0^{x'}, ..., a_n = a_0^{x'^n}, b = g_1^{f(x)t'_0 - \sum_{j=1}^{q} f_j(x)}, c = g_1^{f(x)t''_0 + \sum_{j=1}^{q} s_j f_j(x)}, X = g_2^x, X_0 = g_2, X_1 = X_0^{x'}, ..., X_n = X_0^{x'^n}, \mathsf{pk})$ as the public key to $\mathcal{A}_2$. Thus,

$$\Pr[S_1] \leq \Pr[S_0]. \tag{8}$$

Game$_2$. This game is the same as **Game**$_2$ in Lemma 1 except in step 5, $\mathcal{C}$ picks a random $\beta_i \xleftarrow{\$} \mathbb{Z}_p$ and calculates $\Delta_i = t_i - t'_i$, $s_{2,i} = s_i - s_{1,i}$. $\mathcal{C}$ simulates the partial SDH-CL signature $sig'_i = (\Delta_i, s_{2,i}, v_{1,i})$ on $M_i = a_0^{(x'+o_i)\prod_{j=1}^{n-1}(x'+m_{j,i})} b^{s_{1,i}}$ for $A_i = \{m_{1,i}, ..., m_{n-1,i}, o_i\}$ after reset of $\mathcal{A}_2$ such that:

$$\begin{aligned}
v_{1,i} &= (M_i b^{s_{2,i}} c)^{\delta_{I,i}} \\
&= (a_0^{\prod_{j=1}^{n}(x'+m_{j,i})} b^{s_{1,i}+(s_i - s_{1,i})} c)^{\frac{1}{x+t_i} - \delta_{U,i}} \\
&= \left(g_1^{f(x)t_0 \prod_{j=1}^{n}(x'+m_{j,i})} g_1^{s_i(f(x)t'_0 - \sum_{j=1}^{q} f_j(x))} g_1^{f(x)t''_0 + \sum_{j=1}^{q} s_j f_j(x)}\right)^{\frac{1}{x+t_i} - \frac{u_{2,i}}{\beta_i}} \\
&= \left(g_1^{f(x)[t_0 \prod_{j=1}^{n}(x'+m_{j,i}) + s_i t'_0 + t''_0]} g_1^{\sum_{j=1,j\neq i}^{q}(s_j - s_i) f_j(x)}\right)^{\frac{1}{x+t_i} - \frac{u_{2,i}}{\beta_i}}.
\end{aligned}$$

Observe that the above $v_{1,i}$ can be calculated by using $g_1^{f_i(x)}$ and $g_1^{f_{i,j}(x)}$. $\mathcal{C}$ sends sig'_i and β_i to $\mathcal{A}_2$. Since $\mathcal{C}$ simulates the issuing protocol perfectly, this gives:

$$\Pr[S_2] \leq \Pr[S_1] + (q-1)!/p^{q-1}. \tag{9}$$

Game$_3$. There are no differences between this game and **Game**$_3$ in Lemma 1 such that:

$$\Pr[S_3] = \Pr[S_2]. \tag{10}$$

Game$_4$. There are no differences between this game and **Game**$_4$ in Lemma 1 such that:

$$\Pr[S_4] = \Pr[S_3]. \tag{11}$$

Game$_5$. Similar to the **Game**$_5$ in Lemma 1, $\mathcal{C}$ can reset $\mathcal{A}_2$ to extract the elements (t^*, s^*, v^*) of $cred^*$ where v^* has the form:

$$v^* = \left(g_1^{f(x)[t_0 \prod_{j=1}^{n}(x'+m_{j,i}) + s^* t'_0 + t''_0] + \sum_{j=1,j\neq i}^{q}(s_j - s^*) f_j(x) + (s_i - s^*) f_i(x)}\right)^{\frac{1}{x+t_i}}.$$

Since $\mathcal{A}_2$ must output $t^* = t_i \in \{t_1, ..., t_q\}$ but $s^* \neq s_i \in \{s_1, ..., s_q\}$ for $i \in \{1, ..., q\}$, $\mathcal{C}$ proceeds to compute $c(x)$ of degree $q-2$ and $d \in \mathbb{Z}_p$ from the knowledge of $\{t_1, ..., t_q\}$ such that $f_i(x) = c(x)(x + t_i) + d$ (since $x \neq -t_i$ in $\mathbf{Game}_1$). Moreover, it will be the case that $v \notin L_{(O,I)} \cup L_{(P,V)}$ as discussed in the special case. $\mathcal{C}$ then calculates:

$$\left(v^{*\frac{1}{f_i(x)[t_0 \sum_{j=0}^{n} \mathsf{m}_j^* x'^j + s^* t'_0 + t''_0] + \sum_{j=1, j\neq i}^{q} (s_j - s^*) f_{j,i}(x) + (s_i - s^*) c(x)}} \right)^{\frac{1}{d(s_i - s^*)}}$$
$$= g_1^{\frac{(f_i(x) - c(x)(x+t_i))(s_i - s^*)}{d(s_i - s^*)(x+t_i)}}$$
$$= g_1^{\frac{1}{x+t_i}},$$

and outputs $(g_1^{\frac{1}{x+t_i}}, t_i)$ as the solution for the SDH instance. Therefore, we have:

$$\Pr[S_5] \leq \Pr[S_4] + \sqrt[N]{\sqrt{\epsilon_{\mathsf{sdh}}} - 1} + \frac{1}{p} + 1, \tag{12}$$

and summing up the probability from (7) to (12), we have $\epsilon_{\mathsf{imp}} \leq \sqrt[N]{\sqrt{\epsilon_{\mathsf{sdh}}} - 1} + \frac{1 + (q-1)!/p^{q-2}}{p} + 1$ as required. The time taken by $\mathcal{C}$ is at least $2Nt_{\mathsf{imp}}$ due to reset and interacting with N parallel impersonation instances, in additional to the environment setup and the final SDH solution extraction that cost $T(q^2)$. □

A.3 Proof of Lemma 3

Proof. In this game, we show that if $\mathcal{A}_3$ exists, there exists an algorithm $\mathcal{C}$ which output $(g_1^{\frac{1}{x+t}}, t)$ by acting the simulator for the ABC system. Given a q-SDH instance $(g_1, g_1^x, g_1^{x^2}, ..., g_1^{x^q}, g_2, g_2^x)$ where $q = Q_{(O,I)} + Q_{(P,V)}$ is the number of queries made to the Obtain and Verify oracles, The reduction games are as follows:

$\mathbf{Game}_0$. There are no differences between this game and $\mathbf{Game}_0$ in Lemma 1 such that:

$$\Pr[S_0] = \epsilon_{\mathsf{imp}}. \tag{13}$$

$\mathbf{Game}_1$. This game follows $\mathbf{Game}_1$ in Lemma 2 with exceptions that $\mathcal{C}$ sends $(\mathsf{e}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, a_0 = g_1^{f(x)t_0 - \sum_{j=1}^{q} f_j(x)}, a_1 = a_0^{x'}, ..., a_n = a_0^{x'^n}, b = g_1^{f(x)t'_0 - \sum_{j=1}^{q} f_j(x)}, c = g_1^{f(x)t''_0 + \sum_{j=1}^{q} z_j f_j(x)}, X = g_2^x, X_0 = g_2, X_1 = X_0^{x'}, ..., X_n = X_0^{x'^n}, \mathsf{pk})$ as the public key to $\mathcal{A}_3$, where $z_1, ..., z_q \in \mathbb{Z}_p$ is randomly chosen with uniform distribution. Thus,

$$\Pr[S_1] \leq \Pr[S_0]. \tag{14}$$

$\mathbf{Game}_2$. This game is the same as $\mathbf{Game}_2$ in Lemma 1 except in step 5, $\mathcal{C}$ picks a random $\beta_i \xleftarrow{\$} \mathbb{Z}_p$ and calculates $\Delta_i = t_i - t'_i$, $s_{2,i} = s_i - s_{1,i}$. $\mathcal{C}$ simulates the partial SDH-CL signature $sig'_i = (\Delta_i, s_{2,i}, v_{1,i})$ on $M_i = a_0^{(x'+o_i)\prod_{j=1}^{n-1}(x'+m_{j,i})} b^{s_{1,i}}$ for

$A_i = \{m_{1,i}, ..., m_{n-1,i}, o_i\}$ after reset of $\mathcal{A}_3$ such that:

$$\begin{aligned} v_{1,i} &= (a_0^{\prod_{j=1}^{n}(x'+m_{j,i})} b^{s_{1,i}+(s_i-s_{1,i})} c)^{\frac{1}{x+t_i}-\delta_{U,i}} \\ &= \left(g_1^{(f(x)t_0-\sum_{j=1}^{q} f_j(x))(\prod_{j=1}^{n}(x'+m_{j,i}))} g_1^{s_i(f(x)t'_0-\sum_{j=1}^{q} f_j(x))} g_1^{f(x)t''_0+\sum_{j=1}^{q} z_j f_j(x)} \right)^{\frac{1}{x+t_i}-\frac{u_{2,i}}{\beta_i}} \\ &= \left(g_1^{f(x)[t_0 \prod_{j=1}^{n}(x'+m_{j,i})+s_i t'_0+t''_0]} g_1^{\sum_{j=1,j\neq i}^{q}(z_j-z_i)f_j(x)} \right)^{\frac{1}{x+t_i}-\frac{u_{2,i}}{\beta_i}}. \end{aligned}$$

Observe that the above $v_{1,i}$ can be calculated by using $g_1^{f_i(x)}$. $\mathcal{C}$ sends sig'_i and β_i to $\mathcal{A}_2$. Since $\mathcal{C}$ simulates the Issue oracle perfectly, this gives:

$$\Pr[S_2] \leq \Pr[S_1] + (q-1)!/p^{q-1}. \tag{15}$$

Game$_3$. There are no differences between this game and **Game**$_3$ in Lemma 1 such that:

$$\Pr[S_3] = \Pr[S_2]. \tag{16}$$

Game$_4$. There are no differences between this game and **Game**$_4$ in Lemma 1 such that:

$$\Pr[S_4] = \Pr[S_3]. \tag{17}$$

Game$_5$. This game requires $\mathcal{A}_3$ to output $t^* = t_i \in \{t_1, ..., t_q\}$ and $s^* = s_i \in \{s_1, ..., s_q\}$ for $i \in \{1, ..., q\}$. Note that the output must be the case that $v \notin L_{(O,I)} \cup L_{(P,V)}$ or $\mathcal{C}$ already found $x = -t_i$ during **Game**$_1$. $\mathcal{C}$ aborts with the unlikely case of forgery $(A^*, s^*, t^*, v^*) \in L_{(P,V)}$ which happens with probability $1/p$. Similar to the **Game**$_5$ in Lemma 1, $\mathcal{C}$ can reset $\mathcal{A}_3$ to extract the elements (t^*, s^*, v^*) of $cred^*$ where v^* is in form of

$$v^* = \left(g_1^{f(x)(t_0 \sum_{j=0}^{n} \mathsf{m}_{j,i} x'^j + s_i t'_0 + t''_0)} g_1^{\sum_{j=1,j\neq i}^{q}(z_j-z^*)f_j(x)+(z_i-z^*)f_i(x)} \right)^{\frac{1}{x+t_i}}.$$

$\mathcal{C}$ proceeds to compute $c(x)$ of degree $q-2$ and $d \in \mathbb{Z}_p$ from the knowledge of $\{t_1, ..., t_q\}$ such that $f_i(x) = c(x)(x+t_i) + d$. $\mathcal{C}$ subsequently computes:

$$\begin{aligned} &\left(v^{*\frac{1}{f_i(x)(t_0 \sum_{j=0}^{n} \mathsf{m}^*_j x'^j + s^* t'_0 + t''_0) + \sum_{j=1,j\neq i}^{q}(z_j-z^*)f_{j,i}(x)+(z_i-z^*)c(x)}} \right)^{d(z_i-z^*)} \\ &= g_1^{\frac{(f_i(x)-c(x)(x+t_i))(z_i-z^*)}{d(z_i-z^*)(x+t_i)}} \\ &= g_1^{\frac{1}{x+t_i}}, \end{aligned}$$

and outputs $(g_1^{\frac{1}{x+t_i}}, t_i)$ as the solution for the SDH instance. Therefore, we have:

$$Pr[S_5] \leq Pr[S_4] + \sqrt[N]{\sqrt{\epsilon_{\mathsf{sdh}}} - 1} + 1, \tag{18}$$

and summing up the probability from (13) to (18), we have $\epsilon_{\mathsf{imp}} \leq \sqrt[N]{\sqrt{\epsilon_{\mathsf{sdh}}} - 1} + \frac{(q-1)!}{p^{q-1}} + 1$ as required. The time taken by $\mathcal{C}$ is at least $2Nt_{\mathsf{imp}}$ due to reset and interacting with N parallel impersonation instances, in additional to the environment setup and the final SDH soltion extraction that cost $T(q^2)$. □

B Proof of Theorem 2

Proof. We prove that with respect to the ABC system simulator $\mathcal{C}$, an adversary $\mathcal{A}$ wins the game with $\mathsf{anon\text{-}aca}$-security only with a negligible advantage ϵ_{aunl}.
Game$_0$. Attacking on the original ABC system, we have

$$\Pr[S_0] \leq \epsilon_{\mathsf{anon}} + \frac{1}{2} \tag{19}$$

by definition, where S_0 is denoted as a successful distinguishing attempt.
Game$_1$. As in the original algorithm, $\mathcal{C}$ generates (pk, sk). The key pair is forwarded to $\mathcal{A}$ so that it can play the role as user and issuer. Moreover, $\mathcal{C}$ holds two lists $L_{(O,I)}$ and $L_{(P,V)}$ for corrupted issuing protocol and presentation protocol, respectively. Since the actions do not alter the key generation algorithm, thus:

$$\Pr[S_1] = \Pr[S_0]. \tag{20}$$

Game$_2$. Acting as an issuer, $\mathcal{A}$ interact with $\mathcal{C}$ concurrently, who simulates the Obtain oracle to produce a credential $cred_i$ for the user in the i-th session. Assume every user uses different attributes set $A_i = \{m_{1,i}, ..., m_{n-1,i}, o_i\}$ without loss of generality. The interaction is as follows:

1. $\mathcal{C}$ initials the issuing protocol for use in the i-th session of the concurrent interactions by running the zero-knowledge protocol:

   $$PK\{s_{1,i} : M_i = C_i \cdot b^{s_{1,i}}\}.$$

   $\mathcal{C}$ sends $(A''_i, M_i, C_i, o_i, \pi_{s,i})$ to $\mathcal{A}$. If the ZK proof $\pi_{s,i}$ is valid, $\mathcal{A}$ can successfully extract the secret exponents $s_{1,i}$ used by $\mathcal{C}$ in the protocol.
2. $\mathcal{A}$ validates (A''_i, C_i, o_i) with its own data set T. The protocol proceed if $\mathsf{Open}(pk, C_i, A_i, o_i) = 1$ where A_i is generated locally by $\mathcal{A}$ using A''_i and T.
3. $\mathcal{A}$ chooses a random value $u_{1,i} \in \mathbb{Z}_p$, and sets

   $$z_{u_{1,i}} = \mathsf{Enc}_{\mathsf{pk}}(u_{1,i}).$$

   $\mathcal{A}$ chooses a random ciphertext $z_{\hat{x},i}$. $\mathcal{A}$ sends $(z_{u_{1,i}}, z_{\hat{x},i})$ to $\mathcal{C}$.
4. $\mathcal{C}$ sends $z_{1,i}, \omega_{2,i}$ and a ZK proof $\pi_{U,i}$ to $\mathcal{A}$. By the knowledge extractor of the ZK proof, $\mathcal{A}$ can obtain $(t'_i, u_{2,i}, \gamma_{2,i})$.
5. $\mathcal{A}$ picks a random $\beta_i, s_{2,i} \xleftarrow{\$} \mathbb{Z}_p$, calculates $\Delta_i = t_i - t'_i$ and generates a partial SDH-CL signature for M_i as $sig'_i = (\Delta_i, s_{2,i}, v_{1,i})$, where $v_{1,i} = (M_i b^{s_{2,i}} c)^{\delta_{I,i}}$. $\mathcal{A}$ sends sig'_i and β_i to $\mathcal{C}$. Since $\mathcal{A}$ has extracted t'_i and $u_{2,i}$ from the ZK proof $\pi_{U,i}$, $\mathcal{A}$ can also calculate the full SDH-CL signature (t_i, s_i, v_i). $\mathcal{C}$ adds $cred_i = (t_i, s_i, v_i, A_i)$ to $L_{(O,I)}$.

From the view of $\mathcal{A}$, the issuing protocol is the same as the original one. For every M_i and its witness, they achieve the property of perfectly hiding. Also, each protocol session is uniformly distributed. The arguments are valid for the case where $\mathcal{A}$ concurrently runs the issuing protocol on the same attribute set.

Since $\mathcal{A}$ do not gain more information than acting as an issuer, we ignore the case where $\mathcal{A}$ acts as a user in the issuing protocol. This gives

$$\Pr[S_2] = \Pr[S_1]. \tag{21}$$

Game$_3$. $\mathcal{A}$ queries the issuing protocol transcript of the i-th sesion to the $\mathsf{Corrupt}$ oracle additionally. $\mathcal{C}$ searches in $L_{(O,I)}$ and returns the internal state and the random exponents used. Since the issuing protocol achieves self-reducibility [40], for any two witness sets:

$$(\tilde{s}_{1,i,1}, \tilde{\mathsf{m}}_{0,i,1}, ..., \tilde{\mathsf{m}}_{n,i,1}), (\tilde{s}_{1,i,2}, \tilde{\mathsf{m}}_{0,i,2}, ..., \tilde{\mathsf{m}}_{n,i,2})$$

in the issuing protocol returned by $\mathsf{Corrupt}$, the distribution of their transcripts are identical to each other from the view of $\mathcal{A}$. Following the perfectly hiding property among the committed attributes and the corresponding witness, o_i and $s_{1,i}$ are hidden, thus the non-uniformed distributes attributes are also identical to each other from the view of $\mathcal{A}$. Since $\mathcal{A}$ does not gain any advantage, we have:

$$\Pr[S_3] = \Pr[S_2]. \tag{22}$$

Game$_4$. $\mathcal{A}$ acts as the verifier and concurrently interact with $\mathcal{C}$ as the prover for multiple credentials. $\mathcal{C}$ runs the i-th session of a show proof for $cred_i = (t_i, s_i, v_i, A_i = \{m_{1,i}, ...m_{n-1,i}, o_i\})$. We assume the $\mathcal{A}$ always request for successful show proofs where $\phi_{\mathsf{stmt}}(A_i) = 1$. From the view of $\mathcal{A}$, the interaction is the same as the original show proofs. With the property that the randomized credential in the presentation protocol is perfectly hiding, and the presentation protocol of the credential system offers random self-reducibility [40], this gives:

$$\Pr[S_4] = \Pr[S_3]. \tag{23}$$

Game$_5$. $\mathcal{A}$ also queries to the presentation transcript of the i-th session to the $\mathsf{Corrupt}$ oracle. $\mathcal{C}$ searches in $L_{(P,V)}$ to return the internal state and the random exponents used in completing the protocol. The presentation protocol is an extension to the initialization in the issuing protocol where $\mathcal{C}$ additionally prove the knowledge of the blinding factors used to randomize the credential. Specifically, $\mathcal{C}$ proves the validity of the randomized credential element v'_i in a witness-hiding protocol, such that it consists of the corresponding randomized attributes $(\mathsf{m}'_{0,i}, ..., \mathsf{m}'_{n,i})$, the blinded credential elements t'_i, s'_i and the blinding factors (r_i, y_i). Therefore, following the property that the presentation protocol of the credential system offers random self-reducibility [40], for any two witness sets in a presentation protocol returned by $\mathsf{Corrupt}$, the distribution of their transcripts are identical form the view of $\mathcal{A}$. Following the property that the randomized credential in the presentation protocol is perfectly hiding [40], this is true even if $\mathcal{A}$ knows $(t_i, s_{2,i}, v_i)$ that have been exposed during the issuing protocol, which now have been perfectly hidden by (r_i, y_i). $\mathcal{A}$ also act as a prover in which it does not gain useful information, and any advantage from the query. The same argument applies on show proofs with access policy of composite clauses and thus:

$$\Pr[S_5] = \Pr[S_4], \tag{24}$$

Game$_6$. $\mathcal{C}$ plays the role of user to run the challenge issuing protocol with A_0 and A_1, respectively. When the issuing protocol is completed, $\mathcal{C}$ obtains two credentials $cred_0$ and $cred_1$. $\mathcal{C}$ randomly selects a bit $b \in \{0,1\}$ and completes the challenge show proof with $\mathcal{A}$ as the verifier using $cred_b$. $\mathcal{A}$ can request polynomially many times of show proofs. With the restriction that $\mathcal{A}$ cannot query to the challenge transcripts to $\mathsf{Corrupt}$, $\mathcal{A}$ can query to oracles as before. Finally, $\mathcal{A}$ is requested to make a guess on b. It breaks the anonymity of the ABC system with correctly guessing $b' = b$ with the probability that:

$$\begin{aligned} \Pr[S_6] &= \Pr[S_5] \\ &= \Pr[b' = b] \\ &= \frac{1}{2} + \epsilon_{\mathsf{anon}}. \end{aligned} \tag{25}$$

Combining the equations 19 to 25, we have a negligible ϵ_{anon} as required and $\mathcal{A}$ runs in time t_{anon}. □

C Proof of Theorem 3

Proof. The proof is the same as that of Theorem 2 except **Game**$_6$.

Game$_6$. $\mathcal{C}$ randomly selects a bit $b \in \{0,1\}$ and plays the role of user to run the challenge issuing protocol with A_b and A_{1-b}, respectively. When the issuing protocol is completed, $\mathcal{C}$ obtains two credentials $cred_b$ and $cred_{1-b}$. $\mathcal{C}$ completes the challenge show proof with $\mathcal{A}$ as the verifier using the same order of $cred_b$ and $cred_{1-b}$. $\mathcal{A}$ can request polynomially many times of show proofs. With the restriction that $\mathcal{A}$ cannot query to the challenge transcripts to $\mathsf{Corrupt}$, $\mathcal{A}$ can query to oracles as before. Finally, $\mathcal{A}$ is requested to make a guess on b. It breaks the anonymity of the ABC system with correctly guessing $b' = b$ with the probability that:

$$\begin{aligned} \Pr[S_6] &= \Pr[S_5] \\ &= \Pr[b' = b] \\ &= \frac{1}{2} + \epsilon_{\mathsf{unl}}. \end{aligned} \tag{26}$$

Therefore, we have a negligible ϵ_{unl} as required and $\mathcal{A}$ runs in time t_{unl}. □

D Proof of Theorem 4

Proof. The proof is the same as that of Theorem 2 except **Game**$_4$ to **Game**$_6$.

Game$_4$. $\mathcal{A}$ acts as the verifier and concurrently interact with $\mathcal{C}$ as the provers for multiple credentials. $\mathcal{C}$ runs the i-th session of a show proof for $cred_i = (t_i, s_i, v_i, A_i = \{m_{1,i}, ...m_{n-1,i}, o_i\})$. We assume the $\mathcal{A}$ always request for successful show proofs where $\phi_{\mathsf{stmt}}(A_i) = 1$. From the view of $\mathcal{A}$, the interaction is the same as the original show proofs. We note that the credential in the presentation protocol is being revealed to the verifier, thus the perfectly hiding property is exploited. With the property that the presentation protocol of the credential system offers random self-reducibility [40], this gives: This gives:

$$\Pr[S_4] = \Pr[S_3]. \tag{27}$$

Game$_5$. $\mathcal{A}$ also queries to the presentation transcript of the i-th session to the $\mathsf{Corrupt}$ oracle. $\mathcal{C}$ searches in $L_{(P,V)}$ to return the internal state and the random exponents used in completing the protocol. $\mathcal{A}$ also act as a prover in which it does not gain useful information, and any advantage from the query. The same argument applies on show proofs with access policy of composite clauses and thus:

$$\Pr[S_5] = \Pr[S_4], \tag{28}$$

Game$_6$. It is the same as **Game**$_6$ in Theorem 2.

Therefore, we have a negligible ϵ_{anon} as required and $\mathcal{A}$ runs in time t_{anon}.

E Correctness for the equations from Section 6.1

E.1 AND proof (Section 6.1.1)

$$\begin{aligned}
\hat{e}(W, \prod_{j=0}^{l} X_j^{\mathsf{m}_j}) &= \hat{e}(\prod_{j=0}^{n-l} a_j^{\mathsf{w}'_j}, \prod_{j=0}^{l} (g_2^{x'^j})^{\mathsf{m}_j}) \\
&= \hat{e}(a^{\prod_{j=l+1}^{n}(x'+m_j)}, g_2^{\prod_{j=1}^{l}(x'+m_j)}) \\
&= \hat{e}(a^{\prod_{j=1}^{n}(x'+m_j)}, g_2) \\
&= \hat{e}(\prod_{j=0}^{n} a_j^{\mathsf{m}_j}, g_2) \\
&= \hat{e}(C, X_0)
\end{aligned}$$

E.2 ANY proof (Section 6.1.2)

$$\begin{aligned}
\hat{e}(W'W, \prod_{j=0}^{l} X_j^{\iota_j}) &= \hat{e}(\prod_{j=0}^{k-l} a_j^{\mathsf{m}'_{2,j}} \cdot \prod_{j=0}^{n-l} a_j^{\mathsf{w}'_j}, \prod_{j=0}^{l} X_j^{\iota_j}) \\
&= \hat{e}(\prod_{j=0}^{k-l} a_j^{\mathsf{m}'_{2,j}}, \prod_{j=0}^{l} X_j^{\iota_j}) \cdot \hat{e}(\prod_{j=0}^{n-l} a_j^{\mathsf{w}'_j}, \prod_{j=0}^{l} X_j^{\iota_j}) \\
&= \hat{e}(a^{\sum_{j=l+1}^{n} x'^j \mathsf{m}'_{2,j}}, g_2{}^{\sum_{j=0}^{l} x'^j \iota_j}) \cdot \hat{e}(a^{\sum_{j=l+1}^{n} x'^j \mathsf{w}'_j}, g_2{}^{\sum_{j=0}^{l} x'^j \iota_j}) \\
&= \hat{e}(a^{\prod_{j=1}^{n}(x'+m_j)}, g_2) \cdot \hat{e}(a^{\prod_{j=1}^{n}(x'+\hat{m}_j)}, g_2) \\
&= \hat{e}(\prod_{j=0}^{k} a_j^{\mathsf{m}_{1,j}}, g_2) \cdot \hat{e}(C, g_2) \\
&= \hat{e}(C \cdot \prod_{j=0}^{k} a_j^{\mathsf{m}_{1,j}}, X_0)
\end{aligned}$$

E.3 NAND proof (Section 6.1.3)

Note that C could be rewritten as $C = a_0^{f(x')}$ where $f(x') = d(x')q(x') + r(x')$. Let $d(x') = \sum_{j=0}^{n-k} x'^j \mathsf{w}'_j$, $q(x') = \sum_{j=0}^{k} x'^j \mathsf{m}_j$, and $r(x') = \sum_{j=0}^{k-1} x'^j \mathsf{r}'_j$.

$$\begin{aligned}
\hat{e}(W, \prod_{j=0}^{k} X_j^{\mathsf{m}_j}) &= \hat{e}(\prod_{j=0}^{n-k} a_j^{\mathsf{w}'_j}, \prod_{j=0}^{k} X_j^{\mathsf{m}_j}) \\
&= \hat{e}(a^{\sum_{j=0}^{n} x'^j \mathsf{w}'_j}, g_2{}^{\sum_{j=0}^{k} x'^j \mathsf{m}_j}) \\
&= \hat{e}(a^{d(x')q(x')+r(x')} \cdot a^{-r(x')}, X_0) \\
&= \hat{e}(C \cdot \prod_{j=0}^{k-1} a_j^{-\mathsf{r}'{}_j}, X_0)
\end{aligned}$$

E.4 NANY proof (Section 6.1.4)

$$
\begin{aligned}
\hat{e}(W'W, \prod_{j=0}^{l} X_j^{\delta_j}) &= \hat{e}(\prod_{j=0}^{k-l} a_j^{\mathsf{m}'_{2,j}} \cdot \prod_{j=0}^{n-l} a_j^{\mathsf{w}'_j}, \prod_{j=0}^{l} X_j^{\delta_j}) \\
&= \hat{e}(\prod_{j=0}^{k-l} a_j^{\mathsf{m}'_{2,j}}, \prod_{j=0}^{l} X_j^{\delta_j}) \cdot \hat{e}(\prod_{j=0}^{n-l} a_j^{\mathsf{w}'_j}, \prod_{j=0}^{l} X_j^{\delta_j}) \\
&= \hat{e}(a_0^{\prod_{m_j \in (A'-D)}(x'+m_j)}, X_0^{\prod_{m_j \in (D)}(x'+m_j)}) \cdot \hat{e}(a_0^{d(x')q(x')}, X_0) \\
&= \hat{e}(a_0^{\prod_{m_j \in A'}(x'+m_j)}, X_0) \cdot \hat{e}(a_0^{d(x')q(x')+r(x')} \cdot a_0^{-r(x')}, X_0) \\
&= \hat{e}(\prod_{j=0}^{k} a_j^{\mathsf{m}_{1,j}}, X_0) \cdot \hat{e}(C \cdot \prod_{j=0}^{l-1} a_j^{-\mathsf{r}'_j}, X_0) \\
&= \hat{e}(C \cdot \prod_{j=0}^{k} a_j^{\mathsf{m}_{1,j}} \cdot \prod_{j=0}^{l-1} a_j^{-\mathsf{r}'_j}, X_0)
\end{aligned}
$$