SCHOOL OF
COMPUTING

Title: A Pairing-Based Anonymous Credential System with Efficient Attribute Encoding

Names: Syh-Yuah Tan and Thomas Gross

TECHNICAL REPORT SERIES

No. CS-TR- 1527 June 2019

TECHNICAL REPORT SERIES

No: CS-TR- 1527 Date: June 2019

Title: A Pairing-Based Anonymous Credential System with Efficient Attribute Encoding

Authors: Syh-Yuah Tan and Thomas Gross

Abstract: We propose new multi-show anonymous attribute-based credential system that is provably secure in the standard model under the q-SDH assumption. The corresponding proof system incorporates efficient show proofs for logical statements **AND**, **OR** and **NOT**. Each statement requires only a constant number of bilinear pairing operations by the verifier and none by the prover. The system is based on the pairing-based Camenisch-Lysyanskaya signature scheme and offers a novel efficient attribute encoding method, which is of independent interest.

Printed and published by Newcastle University,
School of Computing, Urban Sciences Building,
Newcastle upon Tyne, NE4 5TG, England.

Bibliographical Details

Title and Authors: A Pairing-Based Anonymous Credential System with Efficient Attribute Encoding

Syh-Yuah Tan and Thomas Gross

NEWCASTLE UNIVERSITY
School of Computing. Technical Report Series. CS-TR- 1527

Abstract: We propose new multi-show anonymous attribute-based credential system that is provably secure in the standard model under the q-SDH assumption. The corresponding proof system incorporates efficient show proofs for logical statements **AND**, **OR** and **NOT**. Each statement requires only a constant number of bilinear pairing operations by the verifier and none by the prover. The system is based on the pairing-based Camenisch-Lysyanskaya signature scheme and offers a novel efficient attribute encoding method, which is of independent interest.

About the authors: Syh-Yuan Tan received his PhD in Engineering from Universiti Tunku Abdul Rahman, Malaysia in 2015. He is currently attached to the School of Computing, Newcastle University UK as a postdoctoral researcher. He worked as a senior lecturer at Multimedia University, Malaysia from 2012 to 2018 and served as a committee in the Malaysian cryptographic standards (MySEAL) from 2016 to 2018. He is a member of the Malaysia national mirror committee for ISO/IEC on Cryptography and Security Mechanisms as well as a member of the Malaysian Society for Cryptology Research. His research interests include cryptography and information security, particularly on the provable security techniques.

Thomas Gross: Current Research: Cyber Security, Privacy and Evidence-based Methods for Security
I am a Tenured Reader in System Security (Associate Professor) at the Newcastle University. I am the Director of the Centre for Cybercrime and Computer Security (CCCS), a UK Academic Centre of Excellence in Cyber Security Research (ACE-CSR). I am a member of the Secure and Resilient Systems group and the Centre for Software Reliability (CSR).

Suggested keywords:

The keywords are
- anonymous credential system
- q-SDH assumption
- efficient zero-knowledge proof system
- logical statements
- pairing-based Camenisch-Lysyanskaya signature scheme

A Pairing-Based Anonymous Credential System with Efficient Attribute Encoding (Technical Report)*

Syh-Yuan Tan and Thomas Groß

Newcastle University, UK

Abstract. We propose new multi-show anonymous attribute-based credential system that is provably secure in the standard model under the q-SDH assumption. The corresponding proof system incorporates efficient show proofs for logical statements AND, OR and NOT. Each statement requires only a constant number of bilinear pairing operations by the verifier and none by the prover. The system is based on the pairing-based Camenisch-Lysyanskaya signature scheme and offers a novel efficient attribute encoding method which is of independent interest.

1 Introduction

An attribute-based anonymous credential (ABC) system allows a user to obtain credentials, that is, certified sets of attributes, from issuers and to anonymously prove the possession of these credentials. ABC system was first proposed by Chaum [22] while the first practical ABC system was introduced by Camenisch and Lysyanskaya (CL) [17] in which the user can demonstrate the possession of his credential for multiple times while retaining unlinkability. The same authors also proposed signature schemes with special algebraic properties [19, 20] which allow the construction of multi-show ABC system that can bind multiple attributes into a single credential. Their Strong Diffie-Hellman (SDH)-based signature [20] ported from Boneh et al.'s group signature [10] is the most efficient among all and has been applied in a number of anonymous credential systems [6, 33, 2, 16, 31, 34, 7].

Au et al. [6] extended SDH-based CL signature [20] scheme to sign on blocks of messages and due to the algebraic properties of the signature, the extended scheme has the same signature size as the original scheme. They also showed that their signature scheme can provide proof of possession protocol which resemble an ABC system.

There have been a number of approaches to extend anonymous credential schemes with different encodings. The *traditional encoding* employed by Camenisch and Lysyanskaya [19] and other schemes on blocks of messages fixes

* Preliminary version published as Newcastle University technical report, TR-1527, 2019. A subsequent version was published in the Cryptology ePrint Archive: Report 2020/587, ia.cr/2020/587

the i-th attribute as the i-th exponent to the i-th base and it has complexity, in terms of the scalar multiplication operation, linear to the number of attributes in the credential. A *prime encoding* first suggested by Camenisch and Groß [14] for the SRSA CL Signature Scheme [19] offers show proofs for AND, NOT and OR statements with a constant number of exponentiations for the specially encoded attributes.

Exploiting the compression feature in SDH-based CL signature [20] and combining with a newly proposed pairing-based accumulator, Camenisch et al. [16] presented an ABC system that supports revocation. Later, Sudarsono et al. [31] applied the accumulator on the attributes instead of user identity and showed that the resulted ABC system can support show proof for AND and OR statements with constant complexity. In the subsequent work, Zhang and Feng [34] replaced the accumulator with a threshold encryption scheme to construct an ABC system that can support show proof for threshold statement. Compared to Au et al.'s basic ABC system [6], the public key size and credential size in the two ABC systems are doubled. Besides, the ABC systems require a signing operation for each attribute and involve at least 10 additional bilinear pairing operations in the proof of possession protocol and show proofs.

To the best of our knowledge, the only ABC system in the standard model that has show proof for AND, OR and NOT statements with constant complexity is the ABC system proposed by Camenisch and Groß [14, 15]. Specifically, the Camenisch-Groß encoding uses a product of prime numbers to represent binary and finite-set attributes in a single exponent, a technique subsequently employed to more complex data structures [26]. The proof of possession protocol and show proof has linear computational complexity in terms of multiplications of prime exponents, however only has a constant number of exponentiations. The show proof for AND and OR statements can be efficiently constructed based on the non-divisibility among prime numbers which represent an attribute each; while the NOT statement is based on the co-primality among the prime numbers. The authors pointed out that it is possible to apply the prime encoding on non-RSA-based ABC system to support logical statements with constant complexity. However, the suggested workaround comes with the price of additional zero-knowledge proofs for RSA parameters and yields an ABC system which is slower than their prime encoded RSA-based ABC system. It is thus interesting to know whether there exists a non-RSA-based ABC system which is provably secure in the standard model and can support show proof for AND, OR and NOT statements efficiently, i.e., more efficient than adopting the prime encoding or accumulators as in [31, 34].

Our Contribution. We present a novel efficient encoding method for ABC system. Unlike the Camenisch-Groß encoding [15], it does not rely on pre-certified prime numbers, yet is able to support the same expressiveness in logical statements AND, NOT, and OR. Specifically, we encode the message m as the exponent $x' + m$ such that x' is a secret privy only to the issuer. Applying the encoding to the SDH-based Camenisch-Lysyanskaya signature [20], we obtain a signature scheme with an efficient attribute encoding which is at least as secure as the

former. We further show that the encoding is collision resistant under the SDH assumption.

Next, we extend the signature scheme with attribute encoding into a multi-show ABC system which efficiently supports the proof of possession protocol as well as show proofs for AND, OR and NOT statements. Subsequently, we give a direct proof in the standard model for the security against impersonation and anonymity of the ABC system.

As a by-product from the proposed ABC system, we rigorously define a new security model for the anonymity notion of ABC system. Our anonymity notion is compatible to that of KVABC system [21, 7, 11] and covers the unlinkability as well as the anonymity required by the state-of-the-art ABC systems [12, 24], both of which we will discuss in related works (Section 2).

Organization. We organize the paper as follows. Section 2 contains related work, especially ABCs from transformation frameworks and KVABCs. In Section 3, we briefly describe the mathematical assumptions and the security notions of attribute credentials system. In Section 4, we present the encoding method and its application SDH-based CL signature scheme [20]. We present our ABC system in Section 5 followed by discussion in Section 6. We conclude the paper in Section 7.

2 Related Works

ABC from Transformation Frameworks. Camenisch et al. [12] as well as Fuchbauer et al. [24] proposed transformation frameworks for ABC system that are relevant to our work. Camenisch et al.'s framework allows one to obtain an ABC system from structure-preserving signature and vector commitment scheme which has:

- non-interactive zero-knowledge (NIZK) proofs without random oracle
- show proof of constant size
- universal composability in the common reference string model

where their most efficient ABC system is based on the construction by Abe et al. [1]. Fuchbauer et al.'s framework also uses the same ingredients but does not yield universal composability and is proven secure in the generic group model only. Camenisch et al. [12] explained that these three advantages in their ABC system are the limitations of the ABC systems based on CL signatures, be it RSA-based or SDH-based. Putting aside the universal composability and NIZK proofs, we are skeptical on the view of the constant size protocol as there is an exception [15] which can provide constant size show proof. Moreover, Camenisch and Groß' ABC system [15] also achieves constant complexity for the show proofs but the frameworks [12, 24] cannot.

Keyed-Verification ABC (KVABC). Chase et al. [21] opened another door for ABC system, namely, keyed-verification ABC system (KVABC system) in which the credential issuer and verifier share the same secret key using algebraic MAC. Barki et al. [7] proposed a more efficient KVABC system based on a newly proposed MAC scheme provably secure in the standard model. We see that their MAC scheme is actually the SDH-based CL signature but in the use case scenario of MAC where both issuer and verifier knows the signing key. Due to this, Barki et al.'s KVABC system can be switched into a public key ABC system assuming the verifier does not know the signing key, which is in fact the basic ABC system by Akagi et al. and Au et al. Recently, Camenisch et al. constructed a KVABC system [11] based on an algebraic MAC ported from the weakly secure Boneh-Boyen signature [9] scheme. In view of the similar structure of the weakly secure Boneh-Boyen signature and SDH-based CL signature, applying the traditional encoding [15], the two KVABC systems also can provide show proof with linear complexity.

3 Preliminaries

3.1 Mathematical Tools

Bilinear Pairing. Let $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ be groups of prime order p. Let $g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2$ and $x, y \in \mathbb{Z}_p$ where g_1, g_2 are the generators, the bilinear pairing function is $\mathsf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ with the following properties:

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.

Throughout this work, we will assume Type-3 pairing which has $\mathbb{G}_1 \neq \mathbb{G}_2$.

Definition 1. *Discrete Logarithm Assumption (DLOG). An algorithm $\mathcal{C}$ is said to $(t_{\mathsf{dlog}}, \varepsilon_{\mathsf{dlog}})$-solve the DLOG assumption if $\mathcal{C}$ runs in time at most t_{dlog} and furthermore:*

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

We say that the DLOG assumption is $(t_{\mathsf{dlog}}, \varepsilon_{\mathsf{dlog}})$-hard if no algorithm $(t_{\mathsf{dlog}}, \varepsilon_{\mathsf{dlog}})$-solves the DLOG assumption.

Definition 2. *$q-$Strong Diffie-Hellman Assumption (SDH) [9]. An algorithm $\mathcal{C}$ is said to $(t_{\mathsf{sdh}}, \varepsilon_{\mathsf{sdh}})$-solve the SDH assumption if $\mathcal{C}$ runs in time at most t_{sdh} and furthermore:*

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

We say that the SDH assumption is $(t_{\mathsf{sdh}}, \varepsilon_{\mathsf{sdh}})$-hard if no algorithm $(t_{\mathsf{sdh}}, \varepsilon_{\mathsf{sdh}})$-solves the SDH assumption.

Definition 3. *Relation ($\mathcal{R}$) [23]. Let $\mathcal{R}$ be a relation $\{(x, w(x))\}$ testable in polynomial time, where x is the secret and $w(x) = \{w_1, \ldots, |w(x)|\}$ is the corresponding set of witnesses such that $(x, w(x)) \in \mathcal{R}$ and $|x| = |w_i|$ for $1 \leq i \leq |w(x)|$.*

Definition 4. *Proof System (P, V) [23]. A zero knowledge proof system (P, V) over $\mathcal{R}$ is a pair of algorithms (P, V) satisfying:*

1. *Completeness: The verifier V always accepts a true statement $P(x, w_i \in w(x))$.*
2. *Soundness: The verifier V always rejects a false statement $P(x, w_i \notin w(x))$ except with a negligible probability.*
3. *Zero-knowledge: A simulator S exists for every true statement such that $(P(x, w_i \in w(x)), V)$ and $S(x, V)$ are polynomially indistinguishable.*

Definition 5. *Witness Indistinguishability (WI) [23]. An algorithm $\mathcal{C}$ is said to $(t_{\mathsf{wi}}, \varepsilon_{\mathsf{wi}})$-break the WI property of a proof system (P, V) over a relation $\mathcal{R}$ if $\mathcal{C}$ runs in time at most t_{wi} and furthermore:*

$$|\Pr[x, \{w_1 \in w(x)\} \in \mathcal{R} : \mathcal{C}(P(x, w_1), V) = 1] - \Pr[x, \{w_2 \in w(x)\} \in R : \mathcal{C}(P(x, w_2), V) = 1]| \geq \varepsilon_{\mathsf{wi}}$$

We say that a proof system is $(t_{\mathsf{wi}}, \varepsilon_{\mathsf{wi}})$-secure if no algorithm $(t_{\mathsf{wi}}, \varepsilon_{\mathsf{wi}})$-break its WI property.

3.2 Digital Signature Scheme

A digital signature scheme is defined by three algorithm as $\mathsf{DS} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ as follows:

1. $\mathsf{KeyGen}.(1^k) \rightarrow (pk, sk)$: A pair of public and secret keys are generated based on the security parameter input 1^k. The public key pk can be made known to the public while the secret key sk is kept secret by the signer.
2. $\mathsf{Sign}.(m, pk, sk) \rightarrow \sigma$: The signer uses the secret key sk to sign on a message m, generating a signature σ.
3. $\mathsf{Verify}.(m, \sigma, pk) \rightarrow 1/0$: The verifier takes the signer's public key pk and σ as the input to ensure that the signature is genuinely signed by the signer. If the signature is verified, the algorithm returns 1 and 0 otherwise.

3.2.1 Unforgeability We refer to the security notion of *strong existential unforgeability under chosen message attacks* (seuf-cma) [9]. The security model is defined as the following game between a forger $\mathcal{F}$ and a challenger $\mathcal{C}$:

Game 1 ($\mathsf{seufcma}(\mathcal{F}, \mathcal{C})$)

1. ***Setup:*** *$\mathcal{C}$ runs KeyGen and sends pk $\mathcal{F}$.*
2. ***Phase 1:*** *$\mathcal{F}$ is allowed to issue queries to the Sign oracle.*

3. ***Challenge:*** *$\mathcal{F}$ outputs a challenge message m^* which may have not been queried to* Sign *oracle previously.*
4. ***Phase 2:*** *$\mathcal{F}$ can continue to query the* Sign *oracle as in Phase 1.*
5. ***Forgery.*** *$\mathcal{F}$ outputs a message and signature pair (m^*, σ^*) which is different from all the previous replies from the* Sign *oracle. $\mathcal{F}$ wins the game if* Verify(m^*, σ^*, pk) *outputs 1.*

Definition 6. *A forger $\mathcal{F}$ is said to $(t_{\mathsf{sig}}, \varepsilon_{\mathsf{sig}})$-break the* seuf-cma *security of a signature scheme if $\mathcal{F}$ runs in time at most t_{sig} and furthermore:*

$$\Pr[\mathsf{Verify}(m^*, \sigma^*, pk) = 1] \geq \varepsilon_{\mathsf{sig}}.$$

We say that a signature scheme is $(t_{\mathsf{sig}}, \varepsilon_{\mathsf{sig}})$-secure if no forger $(t_{\mathsf{sig}}, \varepsilon_{\mathsf{sig}})$-breaks the seuf-cma *security of the signature scheme.*

3.3 SDH-based CL Signature Scheme

The SDH-based CL signature scheme was first proposed in [20] and proven to be seuf-cma-secure [6, 28] with a tight reduction [30] to the SDH assumption in the standard model. We describe the signature scheme 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 elliptic curve whose bilinear pairing is $\mathsf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Select random generators $a, b, c \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$ and a secret value $x \in \mathbb{Z}_p^*$. Output 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 $sk = x$.

$\mathsf{Sign}(m, pk, sk)$: On input m, choose the random values $s, t \in \mathbb{Z}_p^*$ to compute:

$$v = (a^m b^s c)^{\frac{1}{x+t}}$$

In the unlikely case where $x + t = 0 \mod p$ occurs, reselect a random t. Output the signature as $\sigma = (t, s, v)$.

$\mathsf{Verify}(m, \sigma, pk)$: Given $\sigma = (t, s, v)$, accept the signature if the following holds:

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

Theorem 1. *[30] SDH-based CL signature scheme is $(t_{\mathsf{sig}}, \varepsilon_{\mathsf{sig}})$-secure in the standard model under the Strong Diffie-Hellman assumption.*

3.4 Anonymous Credential System

To the best of our knowledge, there are only four rigorous definitions that come in pair with their security models for anonymous credential system (ABC system) in the literature [24, 2, 12, 29] but none of them suit our need in this work for

a fair comparison on security and performance later on. After a careful consideration, using Fuchsbauer et al.'s work [24] as the base, we build a more general definition for ABC system and the corresponding security model. Firstly, we embed the user key generation into the credential issuing protocol and exclude the proprietary feature such as pseudonymization in Camenisch et al.'s ABC system [12]. Secondly, we do not consider non-blind credential issuing protocol as in Akagi et al.'s ABC system [2]. Thirdly, besides viewing the protocol transcript as one of the output from the credential issuing and proof of possessions protocols as in Camenisch's ABC system, we also view attributes as a part of the credential. We highlight that this is important to allow our anonymity notion to cover the non-standardized notions of anonymity and unlinkability to-date.

An anonymous credential system can be defined by five algorithms $\mathsf{ABCsystem} = \{\mathsf{KeyGen}, \mathsf{Obtain}, \mathsf{Issue}, \mathsf{Prove}, \mathsf{Verify}\}$ as follows:

$\mathsf{KeyGen}(1^k, 1^n) \rightarrow (pk, sk)$. This algorithm is executed by the signer. On the input of the security parameter k and the upper bound n of attribute sets, it generates a key pair (pk, sk) for issuer.

$(\mathsf{Obtain}(pk, A), \mathsf{Issue}(pk, sk)) \rightarrow (\pi_1, cred/\perp)$. The first algorithm is executed by the user. On the input of issuer's public key pk and an attribute set A, it generates a user secret key usk. The second algorithm is executed by the issuer and takes as input the issuer's public key pk and secret key sk. At the end of this protocol, a proof π_1 on (usk, A) is produced. Obtain outputs the credential $cred$ if the proof π_1 is accepted by Issue or $\perp$ otherwise.

$(\mathsf{Prove}(pk, A, cred, A'), \mathsf{Verify}(pk, A')) \rightarrow (\pi_2, 1/0)$. The first algorithm is executed by the credential holder which takes as input the issuer's public key pk, user secret key usk, signed attribute set A, user's credential $cred$ and a disclosure attribute set A'. The second algorithm is executed by the credential verifier which takes as input the issuer's public key pk and the disclosure attribute set A'. The verifier has the right to decide which protocol π_2 to run, either a proof of possession protocol where $A = A'$ or a show proof on the $\{\mathsf{AND}, \mathsf{OR}, \mathsf{NOT}\}$ statements. Verify outputs 1 if π_2 is valid or 0 otherwise.

3.4.1 Impersonation The security goal of an ABC system requires that it is infeasible for an adversary to get accepted by the verifier, neither in the proof of possesion nor the show proof. This security goal is termed as *unforgeability* in the literature because the goal seems to boil down to the credential $cred$ which is a blinded signature. Specifically, an adversary should not be able to complete a valid proof of possessions protocol without knowing a valid $cred$. Correspondingly, it should be infeasible for an adversary to complete a valid show proof without knowing a valid $cred$ or a valid attribute set A, or both. However, we highlight that the goal is only to have adversary rejected by the verifier. So, examining adversary's ability to possess a valid $cred$ and A alone is not sufficient. For instance, an adversary can be accepted even without a valid

cred if the protocols are flawed. As a matter of fact, one can see that this security goal is closely related to the security against impersonation in attribute-based identification (ABI) scheme [32, 3] which requires that it is infeasible for an adversary to impersonate as a user, i.e., an adversary cannot get accepted by the verifier through the identification protocol.

The strongest attack which can be perfomed by an ABI adersary is the man-in-the-middle attack, followed by the concurrent attacks, active attack and passive attack. It is well-known that three-move identification protocols such as the sigma protocol cannot resist the man-in-the-middle attack but we argue that the security against impersonation under concurrent attacks is sufficient in the practice. This security notion allows the ABI adversary to concurrently corrupt the user of his choice and play the role of verifier in the identification protocol. Mapping this security notion to the ABC system scenario, it means ABC system adversary should be allowed to query Obtain, Prove and Verify oracles, which is exactly the state-of-the-art security notion of *unforgeability* for ABC system in the literature [12, 24].

Therefore, it is natural for us to term the security model for ABC system as the security against impersonation, instead of unforgeability, in the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$.

Game 2 ($\mathsf{Impersonation}(\mathcal{A}, \mathcal{C})$)

1. ***Setup:*** *$\mathcal{C}$ runs* $\mathsf{KeyGen}(1^k, 1^n)$ *and sends pk to $\mathcal{A}$.*
2. ***Phase 1:*** *$\mathcal{A}$ is able to issue queries to the* Obtain, Prove *and* Verify *oracles where he plays the role of user, prover and verifier respectively on any attribute sets A of his choice.*
3. ***Challenge:*** *$\mathcal{A}$ outputs a challenge attribute set $A^* \not\subseteq A$ for every A queried to the* Obtain *oracle during Phase 1.*
4. ***Phase 2:*** *$\mathcal{A}$ can continue to query the oracles as in Phase 1 with the restriction that it cannot query A to* Obtain *such that $A^* \subseteq A$.*
5. ***Forge:*** *$\mathcal{A}$ completes a proof of possession or a show proof for a credential $cred^*$ of A^* and wins the game if* Verify *outputs 1.*

Definition 7. *An adversary $\mathcal{A}$ is said to $(t_{\mathsf{imp}}, \varepsilon_{\mathsf{imp}})$-break the security against impersonation of an anonymous credential system if $\mathcal{A}$ runs in time at most t_{imp} and furthermore:*

$$\Pr[(\mathcal{A}(pk, A^*, cred^*, \cdot), \mathsf{Verify}(pk, \cdot)) = 1] \leq \varepsilon_{\mathsf{imp}}$$

We say that an anonymous credential system is $(t_{\mathsf{imp}}, \varepsilon_{\mathsf{imp}})$-secure if no adversary $(t_{\mathsf{imp}}, \varepsilon_{\mathsf{imp}})$-breaks the security against impersonation of the anonymous credential system.

3.4.2 Anonymity This security property requires that given polynomially many valid pairs $(\pi^*_{2,b}, A^*)$ which belong to either user u_0 or user u_1 such that $A' \subseteq A_0 \bigcap A_1$, it is infeasible for an adversary to decide the ownership of $\pi^*_{2,b}$

where $b \in \{0,1\}$. Inspired by a recent finding [25] on anonymity issues in the ABC systems, we craft a new notion of anonymity such that it covers the anonymity during the issuing protocol as well as during the proof of possession protocol. We name this notion as full anonymity and it covers the well-accepted anonymity properties in the literature [24, 2, 12, 29] where we view the attribute set as a part of the credential and the adversary is allowed to corrupt both issuer and verifier. Besides, our anonymity notion is stronger than the existing ABC systems [24, 2, 12, 29] as we allow adversary to hold the issuer's secret key sk as required in KVABC [7, 11]. To be precise, these previous works also assume adversary can collude with issuer but the sk is not known to the adversary. The security model is defined as the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$.

Game 3 ($\mathsf{Anonymity}(\mathcal{A}, \mathcal{C})$)

1. ***Setup:*** *$\mathcal{C}$ runs* KeyGen *and sends pk, sk to $\mathcal{A}$.*
2. ***Phase 1:*** *$\mathcal{A}$ is able to make queries to the* Obtain, Issue, Prove *and* Verify *oracles where he plays the role of user, issuer, prover and verifier respectively.*
3. ***Challenge:*** *$\mathcal{A}$ decides the two users u_0, u_1 and the non-empty attribute set $A^* \subseteq A_0 \bigcap A_1$ which he wishes to challenge upon. u_0, u_1 and the corresponding A_0, A_1 can be taken from the existing queries in Phase 1. $\mathcal{C}$ responds by randomly choosing the challenge bit $b \in \{0,1\}$ and interacts as the prover with $\mathcal{A}$ as the verifier to complete the protocol*

$$\pi_{2,b} = (\mathsf{Prove}(pk, A_b, cred_b, A^*), \mathsf{Verify}(pk, A^*))$$

   for a polynomially many time as requested by $\mathcal{A}$.
4. ***Phase 2:*** *$\mathcal{A}$ can continue to query the oracles as in Phase 1.*
5. ***Guess:*** *$\mathcal{A}$ outputs a guess b' and wins the game if $b' = b$.*

T1

Definition 8. *An adversary $\mathcal{A}$ is said to $(t_{\mathsf{ano}}, \varepsilon_{\mathsf{ano}})$-break the security against full anonymity of an anonymous credential system if $\mathcal{A}$ runs in time at most t_{ano} and furthermore:*

$$|\Pr[b = b'] - \frac{1}{2}| \leq \varepsilon_{\mathsf{ano}}$$

We say that an anonymous credential system is $(t_{\mathsf{ano}}, \varepsilon_{\mathsf{ano}})$-secure if no adversary $(t_{\mathsf{ano}}, \varepsilon_{\mathsf{ano}})$-breaks the security against full anonymity of the anonymous credential system.

In the weaker notion [18] which assumes the issuer as a trusted party, the adversary $\mathcal{A}$ is not allowed to access the Issue oracle throughout the game, we term such notion as *weak anonymity*.

T1 **TGr:** Walk through finite-set anonymity.

Definition 9. *An adversary $\mathcal{A}$ is said to $(t_{\mathsf{wano}}, \varepsilon_{\mathsf{wano}})$-break the security against weak anonymity of an anomymous credential system if $\mathcal{A}$ runs in time at most t_{wano} and furthermore:*

$$|\Pr[b = b'] - \frac{1}{2}| \leq \varepsilon_{\mathsf{wano}}$$

We say that an anonymous credential system is $(t_{\mathsf{wano}}, \varepsilon_{\mathsf{wano}})$-secure if no adversary $(t_{\mathsf{wano}}, \varepsilon_{\mathsf{wano}})$-breaks the security against weak anonymity of the anonymous credential system.

4 Efficient Attributes Encoding

In this section, we show how to encode and compress an attribute set $A = \{m_1, \ldots, m_n\}$ into a single message m of the SDH-based CL signature scheme. Recall that an attribute m_i is encoded as $(x' + m_i)$ where x' is not known to the user, multiple attributes then has the form $m = (x' + m_1)(x' + m_2) \ldots (x' + m_n)$. This encoding mechanism has several useful features. The first is the encoded attribute length where an attribute set of arbitrary length n is always a single element in $\mathbb{Z}_p$. Secondly, the existing signing optimization [28] is still applicable since signer knows x'. Thirdly, the encoded attribute set can be viewed as a monic polynomial $m = \sum_{i=0}^{n} \alpha_i x'^i$ where every $\alpha_i \in \mathbb{Z}_p^*$ can be found within $2^n - n - 1$ multiplications and $\sum_{i=1}^{n} \binom{n}{i} - 1$ additions in $\mathbb{Z}_p$ using the recursive searching algorithm as shown in Algorithm 1.

This enables anyone who knows A to verify the signature without the need to know x' and is also the main ingredient to realize the logical statements in our proposed ABC system. The latter is from the fact that $\sum_{i}^{n} \alpha_i x'^i = c(x') \sum_{i}^{n-k} \beta_i x'^i + d$ where $c(x'), d \in \mathbb{Z}_p^*$ can be computed from long division such that $c(x') = \sum_{i}^{k} \omega_i x'^i$ and k is the number of attributes disclosed in a show proof. We can see that although our show proof is not as efficient as that based on the prime encoding [15], ours yields the first pairing-based ABC system whose show proof can efficiently support the AND, OR and NOT logical statements. Lastly, the encoding supports both non-indexed and indexed attributes because the attributes does not need to fix to the bases as in the traditional encoding.

4.1 SDH-based Encoded-Message CL Signature Scheme

We apply the message encoding on SDH-based CL signature [20] T2 scheme and evaluate its security after the modification. This encoded-message signature scheme will be the building block for our multi-show anonymous credential system in the next section.

T2 **TGr:** Does this scheme operate on SDH-CL [20] itself or use Au et al.'s version. Should apply to both.

Algorithm 1 convertToAlphas(): Convert attribute set into αs

Input: Attribute set $A = \{m_1, \ldots, m_n\}$ and prime order p.
Output: $L = \{\alpha_0, \ldots, \alpha_n\}$.
Post-conditions: $\sum_{i=0}^{n} \alpha_i x'^i = (x' + m_1)(x' + m_2) \ldots (x' + m_n)$

```
1: i ← 1
2: n ← |A|
3: L[n + 1] ← {1}
4: val ← 1
5: lvl ← 1
6: procedure toAlphas(val, lvl, i)
7:     if i = n then
8:         return (val, lvl)
9:     end if
10:    while i < n do
11:        temp ← toAlpha(val × A[i] mod p, lvl + 1, i + 1)
12:        if L[n − temp.lvl + 1] =⊥ then
13:            L[n − temp.lvl + 1] ← L[n − temp.lvl + 1] + temp.val
14:        else
15:            L[n − temp.lvl + 1] ← temp.val
16:        end if
17:        i ← i + 1
18:    end while
19:    return (val, lvl)
20: end procedure
21: return L
```

$\mathsf{KeyGen}(1^k)$: Construct three cyclic groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ of order p based on an elliptic curve whose bilinear pairing is $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Select 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 the values $a^{x'}, \ldots, a^{x'^n}$ to output the public key $pk = (\mathsf{e}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, a, a^{x'}, \ldots, a^{x'^n}, b, c, g_2, X = g_2^x)$ and the secret key $sk = (x, x')$.

$\mathsf{Sign}(m_1, \ldots, m_n, pk, sk)$: Select a random $m_1 \in \mathbb{Z}_p^*$ as the user secret key. On the input of attributes $m_2, \ldots, m_n$, choose the random values $s, t \in \mathbb{Z}_p^*$ to compute:

$$v = \left(a^{\prod_{i=1}^{n}(x'+m_i)} b^s c\right)^{\frac{1}{x+t}}$$

and output the signature as $\sigma = (t, s, v)$.

$\mathsf{Verify}(m, \sigma, pk)$: Given $\sigma = (t, s, v)$, compute $\alpha_i \in \mathbb{Z}_p^*$ for $0 \leq i \leq n$ such that $\prod_{i=1}^{n}(x' + m_i) = \sum_{i=0}^{n} \alpha_i x'^i$. Output 1 if the following holds:

$$\begin{aligned} \mathsf{e}(v, Xg_2^t) &= e\left(\left(\prod_{i=0}^{n}(a^{x'^i})^{\alpha_i} b^s c\right)^{\frac{1}{x+t}}, g_2^x g_2^t\right) \\ &= e\left(\prod_{i=0}^{n}(a_1^{x'^i})^{\alpha_i} b^s c, g_2\right) \end{aligned}$$

and output 0 otherwise.

4.2 Security Analysis

We use the naming 'basic signature scheme' as the shorthand for the original SDH-based CL signature scheme [20], and 'encoded-message signature scheme' as the shorthand for the signature scheme with the new message encoding proposed in Section 4.1, that is, original SDH-based CL signature scheme with the proposed encoding applied on the messages.

Theorem 2. *If the encoded-message signature scheme is* $(t_{\mathsf{encode}}, \varepsilon_{\mathsf{encode}})$*-secure, then the basic signature scheme is* $(t_{\mathsf{basic}}, \varepsilon_{\mathsf{basic}})$*-secure such that:*

$$\varepsilon_{\mathsf{encode}} = \varepsilon_{\mathsf{basic}}, t_{\mathsf{encode}} = t_{\mathsf{basic}}$$

where $t_{\mathsf{basic}}, t_{\mathsf{encode}}$ *are the time taken to forge a basic signature and an encoded-message signature, respectively.*

Proof. Given the public key of basic scheme as $(\mathsf{e}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, a, b, c, g_2, X)$, we show that there exists a forger $\mathcal{F}_{\mathsf{basic}}$ which can break the basic scheme with the help of the forger $\mathcal{F}_{\mathsf{encode}}$ of the encoded-message signature scheme.

Game$_0$. This is an attack on the original encoded-message signature scheme. Let S_0 denote a successful forging attempt, we have:

$$\Pr[S_0] = \varepsilon_{\mathsf{encode}} \tag{1}$$

by assumption.

Game$_1$. $\mathcal{F}_{\mathsf{basic}}$ selects uniformly random $x' \in \mathbb{Z}_p^*$ to computes $a^{x'}, \ldots, a^{x'^n}$. $\mathcal{F}_{\mathsf{basic}}$ sets and sends the public key of encoded-message signature scheme as $pk = (\mathsf{e}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, a, a^{x'}, \ldots, a^{x'^n}, b, c, g_2, X)$ to $\mathcal{F}_{\mathsf{encode}}$. Since the distribution of the simulated pk is identical to that of the original pk, this gives:

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

Game$_2$. $\mathcal{F}_{\mathsf{basic}}$ simulates the Sign oracle of encoded-message signature scheme in this game. When $\mathcal{F}_{\mathsf{encode}}$ queries $m_i = (m_{i,1}, \ldots, m_{i,n})$ during the i-th query,

$\mathcal{F}_{\mathsf{basic}}$ queries its basic signature Sign oracle with $m'_i = \prod_{j=1}^{n}(x' + m_{i,j})$ and gets $\sigma'_i = (t_i, s_i, v_i = (a^{m'_i} b^{s_i} c)^{1/(x+t_i)})$ in return. $\mathcal{F}_{\mathsf{basic}}$ gives σ'_i as the signature of m_i to $\mathcal{F}_{\mathsf{encode}}$ where:

$$\begin{aligned} v_i &= (a^{m'_i} b^{s_i} c)^{1/(x+t_i)} \\ &= \left(a^{\prod_{j=1}^{n}(x'+m_{i,j})} b^{s_i} c\right)^{1/(x+t_i)} \end{aligned}$$

as required. As $\mathcal{F}_{\mathsf{encode}}$ perfectly simulates the Sign for $\mathcal{F}_{\mathsf{basic}}$, we have:

$$\Pr[S_2] = \Pr[S_1]. \tag{3}$$

Challenge. At some point, $\mathcal{F}_{\mathsf{encode}}$ decided the messages $m^* = (m_1^*, \ldots, m_n^*)$ to be challenged on. $\mathcal{F}_{\mathsf{basic}}$ then sets $m' = \prod_{j=1}^{n}(x' + m_j^*)$ as his challenge message. $\mathcal{F}_{\mathsf{encode}}$ and $\mathcal{F}_{\mathsf{basic}}$ can continue to interact as in $\mathbf{Game}_2$ but we do not repeat it them here as the probability of a successful forging attempt is the same.

$\mathbf{Game}_3$. When $\mathcal{F}_{\mathsf{encode}}$ outputs a valid forgery $\sigma^* = (t^*, s^*, v^*)$ for the encoded-message signature scheme on the challenge message $m^* = (m_1^*, \ldots, m_n^*)$, $\mathcal{F}_{\mathsf{basic}}$ outputs σ^* as the forgery for the basic signature scheme on the message $m' = \prod_{j=1}^{n}(x' + m_j^*)$. $\mathcal{F}_{\mathsf{basic}}$ success in producing a forgery for the basic signature scheme because σ^* is also a valid signature on m':

$$\begin{aligned} v^* &= \left(a^{\prod_{j=1}^{n}(x'+m_j^*)} b^{s^*} c\right)^{1/(x+t^*)} \\ &= (a^{m'} b^{s^*} c)^{1/(x+t^*)} \end{aligned}$$

as required. The probability of this successful forgery is then:

$$\Pr[S_3] = \varepsilon_{\mathsf{basic}}. \tag{4}$$

Summing up the probability from equation (1) to (4), we have $\varepsilon_{\mathsf{encode}} = \varepsilon_{\mathsf{basic}}$ and $t_{\mathsf{encode}} = t_{\mathsf{basic}}$ as needed. □

We do not know how to construct the reduction in the opposite direction, i.e., from the basic signature scheme to the encoded-message signature scheme without breaking the discrete logarithm assumption of $a^{x'}$. On the other hand, we see that the reduction exists if the basic signature scheme also encodes its message such that the signature is having the form $\sigma = (t, s, v = (a^{x'+m} b^s c)^{1/(x+t)})$ and this is exactly the singleton of the encoded-message signature. This shows that breaking the encoded-message signature scheme is at least as hard as breaking the basic signature scheme.

It remains to prove that the encoding method is collision resistant. This is to ensure a collision m^* cannot be found such that a signature σ is valid on two different messages m and m^*. When there is only one encoded message m in the signature, it is impossible to find a collision as shown below:

$$\begin{aligned} & a^{x'+m} = a^{x'+m^*} \\ \therefore \quad & x' + m \equiv x' + m^* \mod p \\ & m \equiv m^* \mod p \end{aligned}$$

where there is only one $m \in \mathbb{Z}_p$ for the value $a^{x'+m} \in \mathbb{G}_1$. For the completeness of the security analysis, we also consider the case of multiple encoded messages and prove its collision resistant as follows.

Theorem 3. *If the encoding is $(t_{\mathsf{col}}, \varepsilon_{\mathsf{col}})$-collision resistant, then the Strong Diffie-Hellman assumption is $(t_{\mathsf{sdh}}, \varepsilon_{\mathsf{sdh}})$-hard.*

Proof. We show that if there exists an adversary $\mathcal{A}$ which can find a collision of the encoded messages, there exists a challenger $\mathcal{C}$ which can break the SDH assumption with the help of $\mathcal{A}$. $\mathcal{C}$ sets the SDH challenge $(a, a^{x'}, \ldots, a^{x'^n})$ as the parameters for encoding method and sends to $\mathcal{A}$.

When $\mathcal{A}$ found a collision $\{m_1^*, \ldots, m_{t'}^*\}$ on some chosen messages $\{m_1, \ldots, m_t\}$ such that $a^{\prod_{i=1}^{t}(x'+m_i)} = a^{\prod_{i=1}^{t'}(x'+m_i^*)}$ where $1 \leq t, t' \leq n$, there exists at least one $m_j^* \notin \{m_1, \ldots, m_n\}$ such that $a^{\prod_{i=1}^{t}(x'+m_i)} = a^{c(x')(x'+m_j^*)+d}$ where $c(x'), d \in \mathbb{Z}_p^*$ can be computed from long division. $\mathcal{C}$ can extract a solution $(m_j^*, a^{\frac{1}{x'+m_j^*}})$ for the SDH problem as follow:

$$
\begin{aligned}
& a^{\prod_{i=1}^{t}(x'+m_i)} = a^{\prod_{i=1}^{t'}(x'+m_i^*)} \\
\Leftrightarrow & a^{c(x')(x'+m_j^*)+d} = a^{\prod_{i=1}^{t'}(x'+m_i^*)} \\
\Leftrightarrow & a^{c(x')+\frac{d}{x'+m_j^*}} = a^{\prod_{i=1, i\neq j}^{t'}(x'+m_i^*)} \\
\Leftrightarrow & a^{\frac{1}{x'+m_j^*}} = \left(a^{\prod_{i=1, i\neq j}^{t'}(x'+m_i^*)} / a^{c(x')}\right)^{d^{-1}}.
\end{aligned}
$$

□

4.3 Optimization

The optimization on signing described by Boneh and Boyen [9] and Okamoto [28] still applicable here by viewing the messages as $m = \prod_{i=1}^{n}(x' + m_i)$. It requires the knowledge of the discrete logarithms $\beta^{-1}, \gamma \in \mathbb{Z}_p^*$ such that $b = a^\beta$ and $c = a^\gamma$. During signing, the signer selects a random $\delta \in \mathbb{Z}_p^*$ to compute:

$$
\begin{aligned}
v &= \left(a^{\prod_{j=1}^{n}(x'+m_{1,j})} b^s c\right)^{\frac{1}{x+t}} \\
&= \left(a^{\prod_{i=1}^{n}(x'+m_i)+s\beta+\gamma}\right)^{\frac{1}{x+t}} \\
&= \left(a^{\prod_{i=1}^{n}(x'+m_i)+\frac{\delta-r_0(x'+m_0)-r_1\prod_{i=1}^{n}(x'+m_i)-\gamma}{\beta}\beta+\gamma}\right)^{\frac{1}{x+t}} \\
&= a^{\frac{\delta}{x+t}}
\end{aligned}
$$

where $\sigma = (t, s = (\delta - \prod_{i=1}^{n}(x' + m_i) - \gamma)\beta^{-1}, v)$. This replaces two scalar multiplications and two point addition in $\mathbb{G}_1$ with two subtraction and one multiplication in $\mathbb{Z}_p$ in the Sign algorithm.

On the other hand, optimization on the verification can be done by performing the verification as $\mathsf{e}(v, X) = \mathsf{e}(\prod_{i=0}^{n}(a^{x'^i})^{\alpha_i} b^s c v^{-t}, g_2)$ instead. This switches one scalar multiplication and one point addition from $\mathbb{G}_2$ to $\mathbb{G}_1$ in the Verify algorithm. The correctness still hold as shown below:

$$\begin{aligned}\mathsf{e}(v, X) &= \mathsf{e}(v, g_2^{x+t})\mathsf{e}(v, g_2^{-t}) \\ &= e\left(\prod_{i=0}^{n}(a^{x'^i})^{\alpha_i} b^s c, g_2\right)\mathsf{e}(v, g_2^{-t}) \\ &= e\left(\prod_{i=0}^{n}(a^{x'^i})^{\alpha_i} b^s c v^{-t}, g_2\right).\end{aligned}$$

If the signature is initiated with Type-1 pairing, we can take a step further to precompute and store $(n+1)+3$ elements $\{A_i = \mathsf{e}(a^{x'^i}, g_2)\}_{0 \le i \le n}, B = \mathsf{e}(b, g_2), C = \mathsf{e}(c, g_2)$ in $\mathbb{G}_T$ into the public key. This sacrifice allows us to replace $(n+1)+3$ scalar multiplications, $n+3$ point additions in $\mathbb{G}_1$, with $(n+1)+2$ exponentiation and $n+3$ additions in $\mathbb{G}_T$ in the Verify algorithm such that:

$$\begin{aligned}\mathsf{e}(v, X) &= e\left(\prod_{i=0}^{n}(a^{x'^i})^{\alpha_i} b^s c v^{-t}, g_2\right) \\ &= \prod_{i=0}^{n} A_i^{\alpha_i} B^s C \mathsf{e}(v^{-t}, g_2).\end{aligned}$$

This precomputation is not effective for Type-3 pairing because $|\mathbb{G}_T| = 2|\mathbb{G}_1|$ for Type-1 pairing but $|\mathbb{G}_T| \ge 6|\mathbb{G}_1|$ for Type-3 pairing.

5 Anonymous Credential System

It is clear that the security of SDH-based CL signature is not affected after applying the encoding mechanism on the messages. Now we are ready to extend it to a multi-show anonymous credential system 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 elliptic curve whose bilinear pairing is $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Select 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 the values $a^{x'}, \ldots, a^{x'^n}, X_1 = g_2^{x'}, \ldots, X_n = g_2^{x'^n}$ to output the public key $pk = (\mathsf{e}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, a, a^{x'}, \ldots, a^{x'^n}, b, c, g_2, X = g_2^x, X_1, \ldots, X_n)$ and the secret key $sk = (x, x')$.

$(\mathsf{Obtain}(pk, A), \mathsf{Issue}(pk, sk))$: User and verifier interact as follows to generate a user credential *cred* on an attribute set A.

1. User chooses random $m_1, s', r' \in \mathbb{Z}_p^*$ and computes the commitment on the attributes $m_1, \ldots, m_n$ as $C = a^{r'} \prod_{j=1}^{n}(x'+m_j) b^{s'} = (\prod_{j=0}^{n}(a^{x'^j})^{\alpha_j})^{r'} b^{s'}$.

2. User proves the discrete logarithms representation $(s', r'\alpha_0, \ldots, r'\alpha_n)$ to signer as follows:
   (a) User randomly selects $r_{s'}, r_0, \ldots, r_n \in \mathbb{Z}_p^*$ and sends $C, R = \prod_{j=0}^{n}(a^{x'^j})^{r_j} b^{r_{s'}}$ to issuer.
   (b) issuer replies with a challenge $e \in \mathbb{Z}_p^*$.
   (c) User sends the response $z_{s'} = r_{s'} + es', z_0 = r_0 + er'\alpha_0, \ldots, z_n = r_n + er'\alpha_n$ to issuer.
   (d) Issuer proceeds to the next step if:

$$\begin{aligned}\prod_{j=0}^{n}(a^{x'^j})^{z_j} b^{z_{s'}} &= \prod_{j=0}^{n}(a^{x'^j})^{r_j + er'\alpha_j} b^{r_{s'}+es'} \\ &= \prod_{j=0}^{n}(a^{x'^j})^{r_j} b^{r_{s'}} \left(\prod_{j=0}^{n}(a^{x'^j})^{r'\alpha_j} b^{s'}\right)^{e} \\ &= RC^e\end{aligned}$$

   holds. Else, issuer outputs $\perp$ and stops.
3. Issuer generates the signature for user as $\sigma = (t, s'', v = (Cb^{s''}c)^{1/(x+t)})$.
4. User outputs the user secret key as $usk = m_1$ and credential as $cred = (t, s, v, r', m_1, \ldots, m_n)$ where:

$$\begin{aligned}s &= s' + s'' \\ v &= \left(a^{r'\prod_{j=1}^{n}(x'+m_j)} b^s c\right)^{1/(x+t)}\end{aligned}$$

   as required.

Remark: The value α_n always equal to 1 and so we have $z_n = r_n + er'$. This also indicates that the prover actually proves the representation of $(s', r'\alpha_0, \ldots, r'\alpha_{n-1}, r')$.

($\mathsf{Prove}(pk, A, cred, \perp)$,$\mathsf{Verify}(pk, \perp)$): The proof of possession protocol proves the ownership of a credential. This is done by the prover in showing that he owns a credential $cred$ which contains the attribute set $A = \{m_1, \ldots, m_n\}$ without disclosing any attribute. Recall that the commitment $C = \left(\prod_{j=0}^{n} a^{x'^j \alpha_j}\right)^{r'} b^s$, the Prove and Verify algorithms interact as follows:

1. Prover chooses $r, y, r_r, r_y, r_{ty}, r_0, \ldots, r_n, r_s \in \mathbb{Z}_p^*$ and sends $v' = v^{ry^{-1}}, v'' = v'^y, V = v'^{r_y}, Y = \prod_{j=0}^{n}(a^{x'^j})^{r_j} b^{r_s} c^{r_r} v'^{r_{ty}}$ to verifier.
2. Verifier replies with a random challenge $e \in \mathbb{Z}_p^*$.
3. Prover responds with $z_r = r_r + er, z_y = r_y + ey, z_{ty} = r_{ty} - ety, z_0 = r_0 + err'\alpha_0, \ldots, z_n = r_n + err'\alpha_n, z_s = r_s + ers$.

4. Verifier outputs 1 if the equation $\mathsf{e}(\prod_{j=0}^{n}(a^{x'^j})^{z_j} b^{z_s} c^{z_r} v'^{z_{ty}}, g_2) = \mathsf{e}(Y, g_2)\mathsf{e}((v'^{z_y}/V), X)$ holds such that:

$$\begin{aligned}
\mathsf{e}(\prod_{j=0}^{n}(a^{x'^j})^{z_j} b^{z_s} c^{z_r} v'^{z_{ty}}, g_2) &= \mathsf{e}\left(\prod_{j=0}^{n}(a^{x'^j})^{r_j + err'\alpha_j} b^{r_s + ers} c^{r_r + er} v'^{r_{ty} - ety}, g_2\right) \\
&= \mathsf{e}\left(Y\left(\prod_{j=0}^{n}(a^{x'^j})^{rr'\alpha_j} b^{rs} c^{r} v^{-ty}\right)^{e}, g_2\right) \\
&= \mathsf{e}\left(Y\left(\left(\left(a^{\prod_{j=1}^{n}(x'+m_j)}\right)^{r'} b^s c\right)^{r} v'^{-ty}\right)^{e}, g_2\right) \\
&= \mathsf{e}(Y, g_2)\mathsf{e}\left(v^{(x+t)r} v'^{-ty}, g_2^e\right) \\
&= \mathsf{e}(Y, g_2)\mathsf{e}(v'^{(x+t)y} v'^{-ty}, g_2^e) \\
&= \mathsf{e}(Y, g_2)\mathsf{e}(v'^{xy}, g_2^e) \\
&= \mathsf{e}(Y, g_2)\mathsf{e}(v'^{y}, X^e) \\
&= \mathsf{e}(Y, g_2)\mathsf{e}((v'^{z_y}/V), X)
\end{aligned}$$

and 0 otherwise.

($\mathsf{Prove}(pk, A, cred, A')$,$\mathsf{Verify}(pk, A')$): This is the show proof for AND statement. It is a variant of the proof of possession protocol where prover discloses an attribute $A' = m_i$ upon the request from verifier, and proves that his credential $cred$ contains $A' = m_i$. In order to do so, prover has to prove that $(x'+m_i)$ divides the encoded attribute set $\prod_{j=1}^{n}(x' + m_j)$ in $cred$ such that $\prod_{j=1}^{n}(x' + m_j) = (x' + m_i)\sum_{j=0}^{n-1}\beta_j x'^j$. We can extend this disclosure to multiple attributes such that $A' = \{m_1, \ldots, m_\iota\}$ and $\prod_{i=1}^{\iota}(x'+m_i) = \sum_{i=0}^{\iota}\delta_i x'^i$ where $\prod_{j=1}^{n}(x'+m_j) = \sum_{i=0}^{\iota}\delta_i x'^i \cdot \sum_{j=0}^{n-\iota}\beta_j x'^j$. The show proof for the AND statement is as follows:

1. Verifier requests a, AND show proof for the attribute set $A' = \{m_1, \ldots, m_\iota\}$.
2. Prover randomly selects $r, y, r_r, r_{ty}, r_{\beta_0}, \ldots, r_{\beta_{n-\iota}}, r_s \in \mathbb{Z}_p^*$ and sends $v' = v^{ry^{-1}}, v'' = v'^{y}, V = v'^{r_y}, Y_1 = b^{r_{rs}} c^{r_r} v'^{r_{ty}}, Y_2 = \prod_{j=0}^{n-\iota}(a^{x'^j})^{r_{\beta_j}}$.
3. Verifier replies with a random challenge $e \in \mathbb{Z}_p^*$.
4. Prover responds with $z_r = r_r + er, z_y = r_y + ey, z_{ty} = r_{ty} - ety, z_{\beta_0} = r_{\beta_0} + err'\beta_0, \ldots, z_{\beta_{n-\iota}} = r_{\beta_{n-\iota}} + err'\beta_{n-\iota}, z_s = r_s + ers$.
5. Verifier outputs 1 if the equation:

$$\mathsf{e}(b^{z_s} c^{z_r} v'^{z_{ty}}, g_2) e\left(\prod_{j=0}^{n-\iota}(a^{x'^j})^{z_{\beta_j}}, \prod_{i=0}^{\iota}(g_2^{x'^i})^{\delta_i}\right) = \mathsf{e}(Y_1, g_2)\mathsf{e}(Y_2, \prod_{i=0}^{\iota}(g_2^{x'^i})^{\delta_i})\mathsf{e}((v'^{z_y}/V), X)$$

holds such that:

$$
\begin{aligned}
&\mathsf{e}(b^{z_s}c^{z_r}v'^{z_{ty}}, g_2)e\left(\prod_{j=0}^{n-\iota}(a^{x'^j})^{z_{\beta_j}}, \prod_{i=0}^{\iota}(g_2^{x'^i})^{\delta_i}\right)\\
&= \mathsf{e}(b^{r_s}c^{r_r}v'^{r_{ty}}, g_2)e\left(\prod_{j=0}^{n-\iota}(a^{x'^j})^{r_{\beta_j}}, \prod_{i=0}^{\iota}(g_2^{x'^i})^{\delta_i}\right)\left(\mathsf{e}(b^{rs}c^{r}v'^{-ty}, g_2)e\left(\prod_{j=0}^{n-\iota}(a^{x'^j})^{rr'\beta_j}, \prod_{i=0}^{\iota-1}(g_2^{x'^i})^{\delta_i}\right)\right)^e\\
&= \mathsf{e}(Y_1, g_2)\mathsf{e}(Y_2, \prod_{i=0}^{\iota}(g_2^{x'^i})^{\delta_i})\left(\mathsf{e}((b^s)^r c^r v'^{-ty}, g_2)e\left(a^{rr'\sum_{j=0}^{n-\iota}x'^j\beta_j}, g_2^{\sum_{i=0}^{\iota}x'^i\delta_i}\right)\right)^e\\
&= \mathsf{e}(Y_1, g_2)\mathsf{e}(Y_2, \prod_{i=0}^{\iota}(g_2^{x'^i})^{\delta_i})\left(\mathsf{e}((b^s c)^r v'^{-ty}, g_2)e\left(a^{rr'\prod_{j=1}^{n}(x'+m_j)}, g_2\right)\right)^e\\
&= \mathsf{e}(Y_1, g_2)\mathsf{e}(Y_2, \prod_{i=0}^{\iota}(g_2^{x'^i})^{\delta_i})\mathsf{e}((a^{\prod_{j=1}^{n}(x'+m_j)})^{r'}b^s c)^r v'^{-ty}, g_2)^e\\
&= \mathsf{e}(Y_1, g_2)\mathsf{e}(Y_2, \prod_{i=0}^{\iota}(g_2^{x'^i})^{\delta_i})\mathsf{e}(v^{(x+t)r}v'^{-ty}, g_2)^e\\
&= \mathsf{e}(Y_1, g_2)\mathsf{e}(Y_2, \prod_{i=0}^{\iota}(g_2^{x'^i})^{\delta_i})\mathsf{e}(v'^{(x+t)y}v'^{-ty}, g_2^e)\\
&= \mathsf{e}(Y_1, g_2)\mathsf{e}(Y_2, \prod_{i=0}^{\iota}(g_2^{x'^i})^{\delta_i})\mathsf{e}(v'^{xy}, g_2^e)\\
&= \mathsf{e}(Y_1, g_2)\mathsf{e}(Y_2, \prod_{i=0}^{\iota}(g_2^{x'^i})^{\delta_i})\mathsf{e}(v'^{y}, X^e)\\
&= \mathsf{e}(Y_1, g_2)\mathsf{e}(Y_2, \prod_{i=0}^{\iota}(g_2^{x'^i})^{\delta_i})\mathsf{e}((v'^{z_y}/V), X).
\end{aligned}
$$

and 0 otherwise.

Remark: The disclosed attributes are known by verifier and thus $\prod_{i=0}^{\iota}(g_2^{x'^i})^{\delta_i} \in \mathbb{G}_2$ can be calculated by the verifier. If the verifier requests for a show proof on one attribute only, we can set $\iota = 1$ and run the protocol.

($\mathsf{Prove}(pk, A, cred, L)$,$\mathsf{Verify}(pk, L)$): This is the show proof for OR statement. Consider the scenario where the prover is given a list of attributes $L = \{m_1, \ldots, m_l\}$ and he needs to prove that he has one $m_i \in L$ in his *cred* without the verifier knowing which m_i he is proving. Firstly, the prover encodes the attributes in L such that $\prod_{j=1}^{l}(x' + m_j) = (x' + m_i)\sum_{j=1, j\neq i}^{l}\beta'_j x'^j$. Next, the prover proves that the same $(x' + m_i)$ divides the encoded attributes in his *cred* such that $\prod_{j=1}^{n}(x' + m_j) = (x' + m_i)\sum_{j=0}^{n-1}\beta_j x'^j$. We can also extend this protocol to prove the possession on multiple attributes $\{m_1, \ldots, m_\iota\} \in L$ such that $\prod_{j=i}^{\iota}(x' + m_i) = \sum_{i=0}^{\iota}\delta_i x'^i$ as in the AND statement. The show proof for the OR statement is as follows:

1. Verifier sends $Z = a^{\prod_{j=1}^{l}(x'+m_j)} = \prod_{j=0}^{l}(a^{x'^j})^{\alpha_j}$ and requests an OR show proof for ι-many attributes from $L = \{m_1, \ldots, m_l\}$.
2. Prover randomly selects $r, y, r_r, r_{r'}, r_{rr'}, r_{ty}, r_{\delta_1}, \ldots, r_{\delta_\iota}, r_{\beta'_0} \ldots, r_{\beta'_{l-\iota}}, r_{\beta_0}, \ldots, r_{\beta_{n-\iota}}, r_s \in \mathbb{Z}_p^*$ and sends $v' = v^{ry^{-1}}, v'' = v'^y, V = v'^{r_y}, Y_1 = b^{r_{rs}} c^{r_r} v'^{r_{ty}}, Y_2 = \prod_{j=0}^{n-\iota}(a^{x'^j})^{r_{\beta_j}}, Y_3 = \prod_{i=0}^{\iota}(g_2^{x'^i})^{r_{r'}}, Z_1 = Z^{r_{rr'}}, Z_2 = \prod_{j=0}^{l-1}(a^{x'^j})^{r_{\beta'_j}}$.
3. Verifier replies with a random challenge $e \in \mathbb{Z}_p^*$.
4. Prover responds with $z_r = r_r + er, z_y = r_y + ey, z_{r'} = r_{r'} + er', z_{rr'} = r_{rr'} + err', z_{ty} = r_{ty} - ety, z_{\delta_0} = r_{\delta_0} + er'\delta_0, \ldots, z_{\delta_\iota} = r_{\delta_\iota} + er'\delta_\iota, z_{\beta_0} = r_{\beta_0} + er\beta_0, \ldots, z_{\beta_{n-\iota}} = r_{\beta_{n-\iota}} + er\beta_{n-\iota}, z_{\beta'_0} = r_{\beta'_0} + er\beta'_0, \ldots, z_{\beta'_{l-\iota}} = r_{\beta'_{l-\iota}} + er\beta'_{l-\iota}, z_s = r_s + ers$.
5. Verifier outputs 1 if the two equations hold:
   (a) $\mathsf{e}(b^{z_s} c^{z_r} v'^{z_{ty}}, g_2) e\left(\prod_{j=0}^{n-1}(a^{x'^j})^{z_{\beta_j}}, \prod_{i=0}^{\iota}(g_2^{x'^i})^{z_{r'}}\right) = \mathsf{e}(Y_1, g_2)\mathsf{e}(Y_2, Y_3)\mathsf{e}((v'^{z_y}/V), X)$
   (b) $\mathsf{e}\left(\prod_{j=0}^{l-\iota}(a^{x'^j})^{z_{\beta'_j}}, \prod_{i=0}^{\iota}(g_2^{x'^i})^{z_{r'}}\right) = \mathsf{e}(Z_2, Y_3) e\left((Z^{z_{rr'}}/Z_1), g_2\right)$
   such that the correctness of the first equation is:

$$
\begin{aligned}
&\mathsf{e}(b^{z_s} c^{z_r} v'^{z_{ty}}, g_2)\mathsf{e}\left(\prod_{j=0}^{n-\iota}(a^{x'^j})^{z_{\beta_j}}, \prod_{i=0}^{\iota}(g_2^{x'^i})^{z_{r'}}\right) \\
&= \mathsf{e}(b^{r_s} c^{r_r} v'^{r_{ty}}, g_2)\mathsf{e}\left(\prod_{j=0}^{n-\iota}(a^{x'^j})^{r_{\beta_j}}, \prod_{i=0}^{\iota}(g_2^{x'^i})^{r_{r'}}\right)\left(\mathsf{e}(b^{rs} c^r v'^{-ty}, g_2)\mathsf{e}\left(\prod_{j=0}^{n-\iota}(a^{x'^j})^{r\beta_j}, \prod_{i=0}^{\iota}(g_2^{x'^i})^{r'\delta_i}\right)\right)^e \\
&= \mathsf{e}(Y_1, g_2)\mathsf{e}(Y_2, Y_3)\left(\mathsf{e}((b^s c)^r v'^{-ty}, g_2)\mathsf{e}\left(a_1^{r\sum_{j=0}^{n-1} x'^j \beta_j}, g_2^{r'\sum_{i=0}^{\iota} x'^i \delta_i}\right)\right)^e \\
&= \mathsf{e}(Y_1, g_2)\mathsf{e}(Y_2, Y_3)\left(\mathsf{e}((b^s c)^r v'^{-ty}, g_2)\mathsf{e}\left(a^{rr'\prod_{j=1}^{n}(x'+m_j)}, g_2\right)\right)^e \\
&= \mathsf{e}(Y_1, g_2)\mathsf{e}(Y_2, Y_3)\mathsf{e}((a^{\prod_{j=1}^{n}(x'+m_j)})^{r'} b^s c)^r v'^{-ty}, g_2)^e \\
&= \mathsf{e}(Y_1, g_2)\mathsf{e}(Y_2, Y_3)\mathsf{e}(v^{(x+t)r} v'^{-ty}, g_2)^e \\
&= \mathsf{e}(Y_1, g_2)\mathsf{e}(Y_2, Y_3)\mathsf{e}(v'', X^e) \\
&= \mathsf{e}(Y_1, g_2)\mathsf{e}(Y_2, Y_3)\mathsf{e}((v'^{z_y}/V), X)
\end{aligned}
$$

   while the correctness of the second equation is:

$$
\begin{aligned}
\mathsf{e}\left(\prod_{j=0}^{l-\iota}(a^{x'^j})^{z_{\beta'_j}}, \prod_{i=0}^{\iota}(g_2^{x'^i})^{z_{r'}}\right) &= \mathsf{e}\left(\prod_{j=0}^{l-\iota}(a^{x'^j})^{r_{\beta'_j}}, \prod_{i=0}^{\iota}(g_2^{x'^i})^{r'}\right)\mathsf{e}\left(\prod_{j=0}^{l-\iota}(a^{x'^j})^{r\beta'_j}, \prod_{i=0}^{\iota}(g_2^{x'^i})^{r'\delta_i}\right)^e \\
&= \mathsf{e}(Z_2, Y_3)\mathsf{e}\left(a^{r\sum_{j=0}^{l-\iota} x'^j \beta'_j}, g_2^{r'\sum_{i=0}^{\iota} x'^j \delta_i}\right)^e \\
&= \mathsf{e}(Z_2, Y_3)\mathsf{e}\left(Z^{rr'}, g_2\right)^e \\
&= \mathsf{e}(Z_2, Y_3)\mathsf{e}\left((Z^{z_{rr'}}/Z_1), g_2\right)
\end{aligned}
$$

   and verifier outputs 0 otherwise.

Remark: If the verifier requests for a show proof on one attribute only, we can set $\iota = 1$ and run the protocol.

($\mathsf{Prove}(pk, A, cred, A')$,$\mathsf{Verify}(pk, A')$): This is the show proof for NOT statement. In order to show that an attribute $A' = m_i$ is not in the encoded-message signature, we use the fact that when $(x' + m_i) \nmid \prod_{j=1}^{n}(x' + m_j)$, there exist a unique polynomial $\prod_{j=1,j\neq i}^{n-1}(x' + m_j) = \sum_{j=0}^{n-1} \omega_j x'^j$ such that $\prod_{j=1}^{n}(x' + m_j) = (x' + m_i)\sum_{j=0}^{n-1} \omega_j x'^j + d$ where $\omega_j, d \in \mathbb{Z}_p$. Note that ω_j and d can be computed using long division and $d \neq 1$ is always the case when $(x' + m_i) \nmid \prod_{j=1}^{n}(x' + m_j)$. We can also extend this protocol to prove the possession on multiple attributes $A' = \{m_1, \ldots, m_\iota\}$ such that $\prod_{j=i}^{\iota}(x' + m_i) = \sum_{i=0}^{\iota} \delta_i x'^i$ as in the AND statement. The show proof on the NOT statement is as below:

1. Verifier request a NOT show proof for the attributes $A' = \{m_1, \ldots, m_\iota\}$.
2. Prover randomly selects $r, y, r_r, r_{rr'}, r_{ty}, r_{\omega_0}, \ldots, r_{\omega_{n-\iota}}, r_d, r_s \in \mathbb{Z}_p^*$ and sends $v' = v^{ry^{-1}}, v'' = v'^y, V = v'^{r_y}, Y_1 = a^{r_d} b^{r_{rs}} c^{r_r} v'^{r_{ty}}, Y_2 = \prod_{j=0}^{n-\iota}(a^{x'^j})^{r_{\omega_j}}$.
3. Verifier replies with a random challenge $e \in \mathbb{Z}_p^*$.
4. Prover responds with $z_r = r_r + er, z_y = r_y + ey, z_{rr'} = r_{rr'} + err', z_{ty} = r_{ty} - ety, z_{\omega_0} = r_{\omega_0} + err'\omega_0, \ldots, z_{\omega_{n-\iota}} = r_{\omega_{n-\iota}} + err'\omega_{n-\iota}, z_d = r_d + err'd, z_s = r_s + ers$.
5. Verifier accepts if the equation

$$\mathsf{e}(a^{z_d} b^{z_s} c^{z_r} v'^{z_{ty}}, g_2) e\left(\prod_{j=0}^{n-\iota}(a^{x'^j})^{z_{\omega_j}}, \prod_{i=0}^{\iota}(g_2^{x'^i})^{\delta_i}\right) = \mathsf{e}(Y_1, g_2)\mathsf{e}(Y_2, g_2^{\sum_{i=0}^{\iota} x'^i \delta_i})\mathsf{e}((v'^{z_y}/V), X)$$

holds such that:

$$\begin{aligned}
&\mathsf{e}(a^{z_d} b^{z_s} c^{z_r} v'^{z_{ty}}, g_2) e\left(\prod_{j=0}^{n-\iota}(a^{x'^j})^{z_{\omega_j}}, \prod_{i=0}^{\iota}(g_2^{x'^i})^{\delta_i}\right) \\
&= \mathsf{e}(a^{r_d} b^{r_s} c^{r_r} v'^{r_{ty}}, g_2) e\left(\prod_{j=0}^{n-\iota}(a^{x'^j})^{r_{\omega_j}}, \prod_{i=0}^{\iota}(g_2^{x'^i})^{\delta_i}\right) \cdot \\
&\quad \left(\mathsf{e}(a^{rr'd} b^{rs} c^{r} v'^{-ty}, g_2) e\left(\prod_{j=0}^{n-\iota}(a^{x'^j})^{rr'\omega_j}, \prod_{i=0}^{\iota}(g_2^{x'^i})^{\delta_i}\right)\right)^e \\
&= \mathsf{e}(Y_1, g_2)\mathsf{e}(Y_2, g_2^{\sum_{i=0}^{\iota} x'^i \delta_i}) \left(\mathsf{e}(a^{dr'} b^s)^r c^r v'^{-ty}, g_2) e\left(a^{rr' \sum_{j=0}^{n-\iota} x'^j \omega_j}, g_2^{\sum_{i=0}^{\iota} x'^i \delta_i}\right)\right)^e \\
&= \mathsf{e}(Y_1, g_2)\mathsf{e}(Y_2, g_2^{\sum_{i=0}^{\iota} x'^i \delta_i}) \left(\mathsf{e}((a^{dr'} b^s c)^r v'^{-ty}, g_2) e\left(a^{rr' \prod_{j=1}^{n}(x'+m_j)-d}, g_2\right)\right)^e \\
&= \mathsf{e}(Y_1, g_2)\mathsf{e}(Y_2, g_2^{\sum_{i=0}^{\iota} x'^i \delta_i})\mathsf{e}((a^{\prod_{j=1}^{n}(x'+m_j)})^{r'} b^s c)^r v'^{-ty}, g_2)^e \\
&= \mathsf{e}(Y_1, g_2)\mathsf{e}(Y_2, g_2^{\sum_{i=0}^{\iota} x'^i \delta_i})\mathsf{e}(v^{(x+t)r} r v'^{-ty}, g_2)^e \\
&= \mathsf{e}(Y_1, g_2)\mathsf{e}(Y_2, g_2^{\sum_{i=0}^{\iota} x'^i \delta_i})\mathsf{e}(v'', X^e) \\
&= \mathsf{e}(Y_1, g_2)\mathsf{e}(Y_2, g_2^{\sum_{i=0}^{\iota} x'^i \delta_i})\mathsf{e}((v'^{z_y}/V), X).
\end{aligned}$$

Remark: If the attributes should not be disclosed, we can run the protocol as in OR statement where each δ_i is masked by r' while each ω_j is masked by r. If the verifier requests for a show proof on one attribute only, we can set $\iota = 1$ and run the protocol.

5.1 Security Analysis

In this section, we analyze the security of the proposed anonymous credential system. From the first glance, it seems that the seuf-cma security from the encoded-message signature is sufficient to ensure the security against impersonation of the ABC system. However, the opposite is not true where a successful impersonation of the ABC system does not necessarily yield a forgery of the encoded-message signature. It can be that the proof of possession protocol is not properly constructed. Moreover, even if we assume the adversary impersonates successfully by forging a credential, the credential contains the signature on a set of attributes A which is now a group element in $\mathbb{G}_1$ but not the exponents in $\mathbb{Z}_p^*$ as in the encoded-message signature scheme.

Table 1. Types of impersonation and the corresponding assumptions.

Type	A s t v	Adversary	**Assumption**
1	0 0 0 0	$\mathcal{A}_1$	SDH
2	0 0 0 1	$\mathcal{A}_1$	DLOG
3	0 0 1 0	$\mathcal{A}_2$	SDH
4	0 0 1 1	N/A	N/A
5	0 1 0 0	$\mathcal{A}_1$	SDH
6	0 1 0 1	$\mathcal{A}_1$	DLOG
7	0 1 1 0	$\mathcal{A}_3$	SDH
8	0 1 1 1	N/A	N/A
9	1 0 0 0	$\mathcal{A}_1$	SDH
10	1 0 0 1	$\mathcal{A}_1$	DLOG
11	1 0 1 0	$\mathcal{A}_2$	SDH
12	1 0 1 1	N/A	N/A
13	1 1 0 0	$\mathcal{A}_1$	SDH
14	1 1 0 1	$\mathcal{A}_1$	DLOG
15	1 1 1 0	$\mathcal{A}_3$	SDH

Note: 1 = equal, 0 = unequal, N/A = not available

Therefore, we give a direct proof for the ABC system by reducing its security against impersonation to the hardness of SDH assumption. As an adversary impersonates by either forges a credential or breaks the proof of possession protocol in the ABC system, we categorize the way an adversary impersonates by the types of forgeries on the credential as shown in Table 1. The bit 1 indicates the forged credential element appears in an output of the Obtain oracle, while 0 indicates the credential element does not appear before. Subsequently, based

on the forgery types, we differentiate the adversary $\mathcal{A}$ into $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3\}$ corresponding to three different simulation strategies by the SDH challenger $\mathcal{C}$.

The N/A cases need further explanation. Let $M^* = \prod_{j=1}^{n}(x' + m_j^*)$ and $M_i = \prod_{j=1}^{n}(x' + m_{i,j})$, if the forgery (v^*, t^*) produced by $\mathcal{A}$ equals to a pair (v_i, t_i) which has been generated by $\mathcal{C}$, we have forgeries of Type 4, 8, 12 such that:

$$
\begin{aligned}
\because v^* &\equiv v_i \\
(a^{M^*} b^{s^*} c)^{\frac{1}{x+t^*}} &\equiv (a^{M_i} b^{s_i} c)^{\frac{1}{x+t^*}} \\
(a^{M^*+s^*\beta+\gamma})^{\frac{1}{x+t^*}} &\equiv (a^{M_i+s_i\beta+\gamma})^{\frac{1}{x+t^*}} \\
\therefore \frac{M^* + s^*\beta + \gamma}{x + t^*} &\equiv \frac{M_i + s_i\beta + \gamma}{x + t^*} \mod p \\
(x + t^*)(M^* + s^*\beta + \gamma) &\equiv (x + t^*)(M_i + s_i\beta + \gamma) \mod p \\
x(M^* - M_i + \beta(s^* - s_i)) &\equiv t^*(M_i - M^* + \beta(s_i - s^*) + \gamma - \gamma) \mod p \\
x &\equiv \frac{t^*(M_i - M^* + \beta(s_i - s^*))}{M^* - M_i + \beta(s - s_i)} \mod p \\
x &\equiv \frac{-t^*((M^* - M_i) + \beta(s^* - s_i))}{M^* - M_i + \beta(s^* - s_i)} \mod p \\
x &\equiv -t^* \equiv t_i \mod p
\end{aligned}
$$

and $\mathcal{C}$ can break the SDH assumption using x. Notice that this calculation holds as well when $M^* = M_i$ or $s^* = s_i$. In order to simplify $\mathcal{C}$'s simulation strategy, we design the challenger $\mathcal{C}$ to avoid these three forgeries such that it checks whether $X = g_2^{-t_i}$ every time after selecting a random t_i. On the other hand, Type 16 forgery will not occur by definition as it is not a forgery but a credential obtained by $\mathcal{A}$ previously from its Obtain oracle. We describe Lemma 1, 2 and 3 which correspond to adversary $\mathcal{A}_1, \mathcal{A}_2$ and $\mathcal{A}_3$ as follows.

Lemma 1. *If an adversary $\mathcal{A}_1$ $(t_{\mathsf{imp}}, \varepsilon_{\mathsf{imp}})$-breaks the anonymous credential system, then there exists an algorithm $\mathcal{C}$ which $(t_{\mathsf{sdh}}, \varepsilon_{\mathsf{sdh}})$-breaks the SDH assumption such that:*

$$
\begin{aligned}
\varepsilon_{\mathsf{imp}} &\leq \sqrt{\varepsilon_{\mathsf{sdh}}} + \mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}} + (\mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}} + 1)/2^{-p}, \\
t_{\mathsf{imp}} &\leq 2t_{\mathsf{sdh}} + (2^{n-1} - n - 2)t_{\mathsf{mul}} + 2t_{\mathsf{smul}} + t_{\mathsf{add}}
\end{aligned}
$$

where $\mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}}$ is the total query made to the Obtain oracle for n attributes, while $t_{\mathsf{mul}}, t_{\mathsf{smul}}, t_{\mathsf{add}}$ are the time required for a multiplication in $\mathbb{Z}_p^*$ and a scalar multiplication and an addition in $\mathbb{G}_1$.

Proof. Given a SDH instance $(g_1, g_1^x, g_1^{x^2}, \ldots, g_1^{x^q}, g_2, g_2^x)$ where $q = \mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}} + \mathsf{q}_{\mathsf{p}\leftrightarrow\mathsf{v}}$ is the total query $\mathcal{A}_1$ can issue to the Obtain and Verify oracles, we show that if $\mathcal{A}_1$ exists, there exists an algorithm $\mathcal{C}$ which can output $(g_1^{\frac{1}{x+t}}, t)$ by acting as the simulator for the ABC system as follows:

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

$$\Pr[S_0] = \varepsilon_{\mathsf{imp}} \tag{5}$$

Game$_1$. In order to simulate the environment of the ABC system, $\mathcal{C}$ uniformly random selects $t_0, t_0', t_0'', x', t_1, \ldots, t_q \in \mathbb{Z}_p$ and checks whether $X = g_2^{-t_i}$ for $i \in \{1, \ldots, q\}$. If such t_i is found, $\mathcal{C}$ outputs the solution of SDH using the discrete logarithm $x = -t_i$. Else, let $f(x)$ denotes the polynomial $f(x) = \prod_{k=1}^{q}(x + t_k) = \sum_{k=0}^{q} \theta_k x^k$ and $f_i(x)$ denotes the polynomial $f_i(x) = \prod_{k=1, k\neq i}^{q}(x + t_k) = \sum_{k=0}^{q-1} \lambda_k x^k$ where $\theta_0, \ldots, \theta_q \in \mathbb{Z}_p^*$ and $\lambda_0, \ldots, \lambda_q \in \mathbb{Z}_p^*$ can be found in $2^q - q - 1$ and $2^{q-1} - q - 2$ multiplications in $\mathbb{Z}_p^*$, respectively. Let $g_1^{f(x)} = \prod_{k=0}^{q}(g_1^{x^k})^{\theta_k}$, $\mathcal{C}$ sends $(\mathsf{e}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, a = g_1^{f(x)t_0}, a^{x'}, \ldots, a^{x'^n}, b = g_1^{f(x)t_0'}, c = g_1^{f(x)t_0''}, g_2, X = g_2^x, X_1, \ldots, X_n)$ as the public key to $\mathcal{A}_1$. $\mathcal{C}$ also create two empty lists $Q_{o\leftrightarrow i}$ and $Q_{p\leftrightarrow v}$ where the former stores the corrupted credentials simulated during the issuing protocol while the latter stores the non-corrupted credentials simulated during the proof of possession protocol. Since t_0, t_0', t_0'' are uniformly random, the distribution of the public key is the same as in the original scheme due to the self-reducibility of SDH parameters [9] and we have:

$$\Pr[S_1] = \Pr[S_0] \tag{6}$$

Game$_2$. In this game, $\mathcal{A}_1$ plays the role of user while $\mathcal{C}$ plays the role of issuer in the issuing protocol. $\mathcal{C}$ simulates the Issue oracle which interacts with $\mathcal{A}_1$ to produce a credential $cred_i$ for $\mathcal{A}_1$'s chosen hidden attribute set $A_i = \{m_{i,1}, \ldots, m_{i,n}\}$. Their interaction is as follows:

1. $\mathcal{A}_1$ runs the issuing protocol with $\mathcal{C}$ as follows:
   (a) $\mathcal{A}_1$ sends the commitment C_i of the hidden attribute set A_i and the corresponding witness R_i to $\mathcal{C}_i$.
   (b) $\mathcal{C}$ replies with a challenge $e_i \in \mathbb{Z}_p^*$.
   (c) $\mathcal{A}_1$ returns the response corresponding to the commitment and witness $z_{i,s'}, z_{i,0}, \ldots, z_{i,n}$ to $\mathcal{C}$.
   (d) If $\prod_{j=0}^{n}(a^{x'^j})^{z_{i,j}} b^{z_{i,s'}} = R_i C_i^{e_i}$ holds, $\mathcal{C}$ resets $\mathcal{A}_1$ to where it just sent C_i, R_i. If $\mathcal{A}_1$ passes the check again, $\mathcal{C}$ obtains two transcripts $\{C_i, R_i, (e_{i,1}, z_{i,s',1}, z_{i,0,1}, \ldots, z_{i,n,1}), (e_{i,2}, z_{i,s',2}, z_{i,0,2}, \ldots, z_{i,n,2})\}$. Subsequently, $\mathcal{C}$ can extract the secret exponents used in constructing the witness and commitment. If at either round the check fails, $\mathcal{C}$ outputs $\perp$.
2. $\mathcal{C}$ chooses a random value $s_i'' \in \mathbb{Z}_p^*$ and sets $v_i = (\prod_{j=0}^{n} a_i^{\alpha_{i,j} x'^j})^{r_i'} b_i^{s_i' + s_i''} c_i$ where $a_i = g_1^{f_i(x)t_0}, b_i = g_1^{f_i(x)t_0'}, c_i = g_1^{f_i(x)t_0''}$. If $(\alpha_{i,0}, \ldots, \alpha_{i,n}) \in Q_{p\leftrightarrow v}$, $\mathcal{C}$ removes it from $Q_{p\leftrightarrow v}$. $\mathcal{C}$ adds $(\alpha_{i,0}, \ldots, \alpha_{i,n}, t_i, s_i, v_i, r_i')$ to $Q_{o\leftrightarrow i}$ and returns (t_i, s_i'', v_i) to $\mathcal{A}_1$ as the encoded-message signature on C_i. The simulated

signature is valid because:

$$\begin{aligned}
\mathsf{e}(v_i, Xg_2^{t_i}) &= \mathsf{e}((a_i^{\prod_{j=1}^{n}(x'+m_{i,j})})^{r'_i} b_i^{s'_i+s''_i} c_i, g_2^{x+t_i}) \\
&= \mathsf{e}(((a^{r'_i \prod_{j=1}^{n}(x'+m_{i,j})} b_i^{s'_i+s''_i} c_i)^{\frac{1}{x+t_i}}, g_2^{x+t_i}) \\
&= \mathsf{e}(a^{r'_i \prod_{j=1}^{n}(x'+m_{i,j})} b^{s'_i+s''_i} c, g_2) \\
&= \mathsf{e}(C_i b^{s''_i} c, g_2)
\end{aligned}$$

as required.

Since $\mathcal{C}$ simulates the Issue oracle perfectly, $\mathcal{A}_1$ can formulate its credential $cred = (t_i, s_i = s'_i + s''_i, v_i, r'_i, usk_i, A_i)$ as in the original issuing protocol. By Theorem 3, the encoding is collision resistant and so the extracted values $\alpha_{i,0}, \ldots, \alpha_{i,n}$ from the reset must be the A_i used by $\mathcal{A}_1$ throughout the issuing protocol. Next, since $\mathcal{A}_1$ knows the secret exponents A_i, $\mathcal{C}$ always reset successfully. Let $\mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}}$ be the total rounds of successful issuing protocol initiated by $\mathcal{A}_1$, by Reset Lemma [8], we have:

$$\Pr[S_2] \leq \Pr[S_1] + \mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}}(1 + 2^{-p}) \tag{7}$$

Game$_3$. In this game, $\mathcal{A}_1$ plays the role of prover while $\mathcal{C}$ plays the role of verifier in the proof of possession protocol. $\mathcal{C}$ simulates the Verify oracle which communicates with $\mathcal{A}_1$ to confirm the possession on $cred_i$ for A_i:

1. $\mathcal{A}_1$ sends the witnesses $v'_i, v''_i, C'_i, R'_i, V_i, Y_i$ to $\mathcal{C}$.
2. $\mathcal{C}$ replies with a random challenge $e_i \in \mathbb{Z}_p^*$.
3. $\mathcal{A}_1$ responds with $z_{i,r}, z_{i,y}, z_{i,ty}, z_{i,0}, \ldots, z_{i,n}, z_{i,s}$.
4. Verifier accepts if the following equation holds:

$$\mathsf{e}(\prod_{j=0}^{n}(a^{x'^j})^{z_{i,j}} b^{z_{i,s}} c^{z_{i,r}} v'^{z_{i,ty}}, g_2) = \mathsf{e}(Y_i, g_2)\mathsf{e}((v_i'^{z_{i,y}}/V_i), X)$$

and outputs $\perp$ otherwise.

If this is a show proof for the AND, OR or NOT statement, $\mathcal{C}$ decides a random disclosure attribute set A'_i before the protocol starts. The verification at step 4 is then changed accordingly, corresponding to the logical statements being proved. $\mathcal{C}$ simulates the Verify oracle correctly and this gives:

$$\Pr[S_3] = \Pr[S_2] \tag{8}$$

Game$_4$. In this game, $\mathcal{A}_1$ plays the role of verifier while $\mathcal{C}$ is required to simulate the Prove oracle. If $\mathcal{A}_1$ asks for proof of possession protocol, $\mathcal{C}$ interacts with $\mathcal{A}_1$ using a $cred_i$ which contains a random attribute set A_i. Else if $\mathcal{A}_1$ asks for a show proof for the AND, OR or NOT statement on a disclosure attribute set A'_i, $\mathcal{C}$ interacts with $\mathcal{A}_1$ using a $cred_i$ which contains A'_i. Without loss of generality, we assume $\mathcal{C}$ already has the appropriate credentials on his hand for these purposes.

Else, $\mathcal{C}$ simulates $(\alpha_{i,0}, \ldots, \alpha_{i,n}, t_i, s_i, v_i, r'_i)$ and adds it to $Q_{p \leftrightarrow v}$ before doing what $\mathcal{A}_1$ did in $\mathbf{Game}_3$. This gives:

$$\Pr[S_4] = \Pr[S_3] \tag{9}$$

Challenge: $\mathcal{A}_1$ decided to announce the challenge attribute set $A^* \not\subseteq Q_{o \leftrightarrow i}$. $\mathcal{A}_1$ is still allowed to query the oracles as in $\mathbf{Game}_2$, $\mathbf{Game}_3$ and $\mathbf{Game}_4$ but with the restriction $A^* \not\subseteq Q_{o \leftrightarrow i}$ in $\mathbf{Game}_2$. Since the success probability does not change, we do not reiterate the games involved.

$\mathbf{Game}_5$. Finally, if $\mathcal{A}_1$ completes a successful proof of possession or show proof for an unknown $cred^*$ of $A^* = \{m_1^*, \ldots, m_n^*\} \not\subseteq Q_{o \leftrightarrow i}$ such that $(\mathcal{A}_1(pk, A^*, cred^*, \cdot), \mathcal{C}(pk, \cdot)) = 1, \mathcal{C}$ resets $\mathcal{A}_1$ to the time where it has just sent the witnesses. If $(\mathcal{A}_1(pk, A^*, \cdot, \cdot), \mathcal{C}(pk, \cdot)) = 1$ again the second time, $\mathcal{C}$ can obtain two valid transcripts $(v'^*, v''^*, C'^*, R'^*, V^*, Y^*, (e_1, z^*_{r,1}, z^*_{y,1}, z^*_{ty,1}, z^*_{0,1}, \ldots, z^*_{n,1}, z^*_{s,1}), (e_2, z^*_{r,2}, z^*_{y,2}, z^*_{ty,2}, z^*_{0,2}, \ldots, z^*_{n,2}, z^*_{s,2}))$ and recover the following:

1. $r^* = \frac{z^*_{r,2} - z^*_{r,1}}{e_2 - e_1}$
2. $y^* = \frac{z^*_{y,2} - z^*_{y,1}}{e_2 - e_1}$
3. $r^* r'^* = \alpha^*_n = \frac{z^*_{n,2} - z^*_{n,1}}{e_2 - e_1}$
4. $\alpha^*_0 = \frac{z^*_{0,2} - z^*_{0,1}}{(e_2 - e_1) r r'}, \ldots, \alpha^*_{n-1} = \frac{z^*_{n-1,2} - z^*_{n-1,1}}{(e_2 - e_1) r^* r'^*}$
5. $s^* = \frac{z^*_{s,2} - z^*_{s,1}}{(e_2 - e_1) r}$
6. $t^* = \frac{z^*_{ty,2} - z^*_{ty,1}}{(e_2 - e_1) y}$
7. $v^* = v''^{* r^{*-1}}$

to extract the elements (t^*, s^*, v^*) from $cred^*$. Since $\mathcal{A}_1$ must output $t^* \notin \{t_1, \ldots, t_q\}$, if $v^* \notin Q_{o \leftrightarrow i} \cup Q_{p \leftrightarrow v}$, $\mathcal{C}$ can construct a polynomial $c(x) = \sum_{k=0}^{n-1} \omega_i x^i$ of degree $n-1$ such that $f(x) = c(x)(x+t) + d$ where $\omega_i, d \in \mathbb{Z}_p^*$ to calculate:

$$\begin{aligned} v^{* \frac{1}{(t_0 \sum_{i=0}^n \alpha_i^* x'^i + t'_0 s^* + t''_0) d}} g_1^{-\frac{c(x)}{d}} &= g_1^{\frac{(t_0 \sum_{i=0}^n \alpha_i^* x'^i + t'_0 s^* + t''_0) f(x)}{(t_0 \sum_{i=0}^n \alpha_i^* x'^i + t'_0 s^* + t''_0)(x+t^*) d} - \frac{c(x)}{d}} \\ &= g_1^{\frac{c(x)(x+t^*)+d}{d(x+t^*)} - \frac{c(x)}{d}} \\ &= g_1^{\frac{1}{x+t^*}} \end{aligned}$$

and output $(g^{\frac{1}{x+t^*}}, t^*)$ as the solution for the SDH instance.

On the other hand, if we have $v^* \in Q_{o \leftrightarrow i} \cup Q_{p \leftrightarrow v}$, it implies $\frac{(t_0 \sum_{i=0}^n \alpha_i^* x'^i + t'_0 s^* + t''_0) f(x)}{x+t^*} \equiv \frac{(t_0 \sum_{i=0}^n \alpha_{j,i} x'^i + t'_0 s_j + t''_0) f(x)}{x+t_j} \mod p$ for a $j \in \{1, \ldots, q\}$. Let T^* and T_j be the numerator (excluding $f(x)$) for the two fractions, respectively, $\mathcal{C}$ can extract the

discrete logarithm x such that:

$$\begin{aligned}
\frac{T^* f(x)}{x+t^*} &\equiv \frac{T_i f(x)}{x+t_j} \mod p \\
\frac{T^*}{x+t^*} &\equiv \frac{T_j}{x+t_j} \mod p \\
T^*(x+t_j) &\equiv T_j(x+t^*) \mod p \\
T^*x + T^*t_j &\equiv T_j x + T_j t^* \mod p \\
x(T^* - T_j) &\equiv T_j t^* - T^* t_j \mod p \\
x &\equiv \frac{T_j t^* - T^* t_j}{T^* - T_j} \mod p
\end{aligned}$$

to solve the SDH instance. Notice that x can be found even when we have $s^* \in Q_{o\leftrightarrow i} \cup Q_{p\leftrightarrow v}$ for the same $j \in \{1, \ldots, q\}$. Furthermore, $\mathcal{C}$ will not get denominator $T^* - T_j = 0$ as this is the Type 8 forgery in Table 1. Therefore, we have:

$$\Pr[S_5] \leq \Pr[S_4] + \sqrt{\varepsilon_{\mathsf{sdh}}} + 2^{-p} \tag{10}$$

and summing up the probability from (5) to (10), we have $\varepsilon_{\mathsf{imp}} \leq \sqrt{\varepsilon_{\mathsf{sdh}}} + \mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}} + (\mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}} + 1)/2^{-p}$ as required. The time taken by $\mathcal{C}$ is at most $2t_{\mathsf{sdh}}$ due to reset, besides the final SDH solution extraction which costs $2^{n-1} - n - 2$ multiplications in $\mathbb{Z}_p^*$ plus two scalar multiplications and one addition in $\mathbb{G}_1$. □

Lemma 2. *If an adversary $\mathcal{A}_2$ $(t_{\mathsf{imp}}, \varepsilon_{\mathsf{imp}})$-breaks the anonymous credential system, then there exists an algorithm $\mathcal{C}$ which $(t_{\mathsf{sdh}}, \varepsilon_{\mathsf{sdh}})$-breaks the SDH assumption such that:*

$$\begin{aligned}
&\varepsilon_{\mathsf{imp}} \leq \sqrt{\varepsilon_{\mathsf{sdh}}} + \mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}} + (\mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}} + 1)/2^{-p}, \\
&t_{\mathsf{imp}} \leq 2t_{\mathsf{sdh}} + (2^{n-2} - n - 3)t_{\mathsf{mul}} + (\mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}} + \mathsf{q}_{\mathsf{p}\leftrightarrow\mathsf{v}} + 1)t_{\mathsf{smul}} + t_{\mathsf{add}}
\end{aligned}$$

where $\mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}}, \mathsf{q}_{\mathsf{p}\leftrightarrow\mathsf{v}}$ are the total query made to the Obtain *and* Verify *oracles for n attributes, respectively, while $t_{\mathsf{mul}}, t_{\mathsf{smul}}, t_{\mathsf{add}}$ are the time required for a multiplication in $\mathbb{Z}_p^*$ and a scalar multiplication and an addition in $\mathbb{G}_1$.*

Proof. Given a SDH instance $(g_1, g_1^x, g_1^{x^2}, \ldots, g_1^{x^q}, g_2, g_2^x)$ where $q = \mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}} + \mathsf{q}_{\mathsf{p}\leftrightarrow\mathsf{v}}$ is the total query $\mathcal{A}_1$ can issue to the Obtain and Verify oracles, there exists an algorithm $\mathcal{C}$ which can output $(g_1^{\frac{1}{x+t}}, t)$ by acting as the simulator for the ABC system as follows:

Game$_0$. This is the same as the **Game$_0$** in Lemma 1 where we have:

$$\Pr[S_0] = \varepsilon_{\mathsf{imp}} \tag{11}$$

Game$_1$. This is the same as the **Game$_1$** in Lemma 1 except that $\mathcal{C}$ additionally computes $f_{i,j}(x) = \prod_{k=1, k\neq i,j}^{q}(x + t_k) = \sum_{k=0}^{q-2} \gamma_k x^k$ where $\gamma_0, \ldots, \gamma_{q-2}$ can

be found with $2^{q-2}-q-3$ multiplications in $\mathbb{Z}_p^*$. $\mathcal{C}$ sends $(\mathsf{e}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, a = g_1^{f(x)t_0}, a^{x'}, \ldots, a^{x'^n}, b = g_1^{f(x)t_0' - \sum_{i=1}^q f_i(x)}, c = g_1^{f(x)t_0'' + \sum_{i=1}^q s_i f_i(x)}, g_2, X = g_2^x, X_1, \ldots, X_n)$ as the public key to $\mathcal{A}_2$ where $s_1, \ldots, s_q \in \mathbb{Z}_p^*$ are uniformly random integers. This gives:

$$\Pr[S_1] = \Pr[S_0] \tag{12}$$

Game$_2$. This is the same as the **Game**$_2$ in Lemma 1 except that, after resetting $\mathcal{A}_2$, $\mathcal{C}$ simulates the encoded-message signature (t_j, s_j, v_j) on $C_j = (a^{\prod_{i=1}^n (x'+m_{j,i})})^{r_j'} b^{s_j'}$ for $A_j = \{m_{j,1}, \ldots, m_{j,n}\}$ such that:

$$\begin{aligned}
v_j &= ((a^{\prod_{i=1}^n (x'+m_{j,i})})^{r_j'} b^{s_j' + (s_j - s_j')} c)^{1/(x+t_j)} \\
&= ((g_1^{f(x)t_0 \prod_{i=1}^n (x'+m_{j,i})})^{r_j'} g_1^{s_j(f(x)t_0' - \sum_{i=1}^q f_i(x))} g_1^{f(x)t_0'' + \sum_{i=1}^q s_i f_i(x)})^{1/(x+t_j)} \\
&= (g_1^{f(x)(r_j' t_0 \prod_{i=1}^n (x'+m_{j,i}) + s_j t_0' + t_0'') - s_j \sum_{i=1}^q f_i(x) + \sum_{i=1}^q s_i f_i(x)})^{1/(x+t_j)} \\
&= g_1^{f_j(x)(r_j' t_0 \prod_{k=1}^n (x'+m_{j,i}) + s_j t_0' + t_0'') - s_j \sum_{i=1, i\neq j}^q f_{i,j}(x) + \sum_{i=1, i\neq j}^q s_i f_{i,j}(x)} g_1^{\frac{f_j(x)(s_j - s_j)}{(x+t_j)}} \\
&= g_1^{f_j(x)(r_j' t_0 \prod_{i=1}^n (x'+m_{k,i}) + s_j t_0' + t_0'') + \sum_{i=1, i\neq j}^q (s_i - s_j) f_{i,j}(x)}
\end{aligned}$$

and $s_j'' = s_j - s_j'$. When the protocol ends, $\mathcal{A}_2$ can compile the credential as $cred_j = (t_j, s_j = s_j' + s_j'', v_j, r_j', usk_j, A_j)$. As $\mathcal{C}$ simulates the Issue oracle perfectly, by Reset Lemma [8], we have:

$$\Pr[S_2] = \Pr[S_1] + \mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}}(1 + 2^{-p}) \tag{13}$$

Game$_3$**.** This is the same as the **Game**$_3$ in Lemma 1 and we have:

$$\Pr[S_3] = \Pr[S_2] \tag{14}$$

Game$_4$**.** This is the same as the **Game**$_4$ in Lemma 1 and we have:

$$\Pr[S_4] = \Pr[S_3] \tag{15}$$

Challenge: $\mathcal{A}_2$ decided to announce the challenge attribute set $A^* \not\subseteq Q_{o\leftrightarrow i}$. $\mathcal{A}_2$ is still allowed to query the oracles as in **Game**$_2$, **Game**$_3$ and **Game**$_4$ but with the restriction $A^* \not\subseteq Q_{o\leftrightarrow i}$ in **Game**$_2$. Since the success probability does not change, we do not reiterate the games involved.

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 $cred^*$. Since $\mathcal{A}_2$ must output $t^* \in Q_{o\leftrightarrow i} \cup Q_{p\leftrightarrow v}$ but $s^* \notin Q_{o\leftrightarrow i} \cup Q_{p\leftrightarrow v}$ for a $j \in \{1, \ldots, q\}$, $\mathcal{C}$ proceeds to compute $c(x)$ of degree $n-2$ and $d \in \mathbb{Z}_p^*$ from the knowledge of $\{t_1, \ldots, t_q\}$ such that $f_j(x) = c(x)(x+t_j) + d$. Recall that $\mathcal{C}$ will further have the case $v \notin Q_{o\leftrightarrow i} \cup Q_{p\leftrightarrow v}$ or $\mathcal{C}$ already found

$x = -t_j$ during $\mathbf{Game}_1$. Subsequently, $\mathcal{C}$ calculates:

$$\begin{aligned}
&\left(v/g_1^{f_j(x)(r'^* t_0 \prod_{i=1}^{n}(x'+m_i^*)+s^* t_0'+t_0'')+\sum_{i=1,i\neq j}^{q}(s_i-s^*)f_{i,j}(x)}\right)^{\frac{1}{d(s_j-s^*)}} g_1^{-\frac{c(x)}{d}} \\
&= g_1^{\frac{f_j(x)(s_j-s^*)}{d(s_j-s^*)}} g_1^{-\frac{c(x)}{d}} \\
&= g_1^{\frac{f_j(x)}{d}} g_1^{-\frac{c(x)}{d}} \\
&= g_1^{\frac{c(x)(x+t_j)+d}{d(x+t_j)}} g_1^{-\frac{c(x)}{d}} \\
&= g_1^{\frac{c(x)}{d}+\frac{1}{x+t_j}-\frac{c(x)}{d}} \\
&= g_1^{\frac{1}{x+t_j}}
\end{aligned}$$

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

$$\Pr[S_5] \leq \Pr[S_4] + \sqrt{\varepsilon_{\mathsf{sdh}}} + 2^{-p} \tag{16}$$

and summing up the probability from (12) to (16), we have $\varepsilon_{\mathsf{imp}} \leq \sqrt{\varepsilon_{\mathsf{sdh}}} + \mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}} + (\mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}} + 1)/2^{-p}$ as required. The time taken by $\mathcal{C}$ is at most $2t_{\mathsf{sdh}}$ due to reset, besides the final SDH solution extraction which costs $2^{n-2} - n - 3$ multiplications in $\mathbb{Z}_p^*$ plus $q+1$ scalar multiplications and one addition in $\mathbb{G}_1$. □

Lemma 3. *If an adversary $\mathcal{A}_3$ $(t_{\mathsf{imp}}, \varepsilon_{\mathsf{imp}})$-breaks the anonymous credential system, then there exists an algorithm $\mathcal{C}$ which $(t_{\mathsf{sdh}}, \varepsilon_{\mathsf{sdh}})$-breaks the SDH assumption such that:*

$$\begin{aligned}
&\varepsilon_{\mathsf{imp}} \leq \sqrt{\varepsilon_{\mathsf{sdh}}} + \mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}} + (\mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}} + 1)/2^{-p}, \\
&t_{\mathsf{imp}} \leq 2t_{\mathsf{sdh}} + (2^{n-2} - n - 3)t_{\mathsf{mul}} + (\mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}} + \mathsf{q}_{\mathsf{p}\leftrightarrow\mathsf{v}} + 1)t_{\mathsf{smul}} + t_{\mathsf{add}}
\end{aligned}$$

where $\mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}}, \mathsf{q}_{\mathsf{p}\leftrightarrow\mathsf{v}}$ are the total query made to the Obtain *and* Verify *oracles for n attributes, respectively, while $t_{\mathsf{mul}}, t_{\mathsf{smul}}, t_{\mathsf{add}}$ are the time required for a multiplication in $\mathbb{Z}_p^*$ and a scalar multiplication and an addition in $\mathbb{G}_1$.*

Proof. Given a SDH instance $(g_1, g_1^x, g_1^{x^2}, \ldots, g_1^{x^q}, g_2, g_2^x)$ where $q = \mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}} + \mathsf{q}_{\mathsf{p}\leftrightarrow\mathsf{v}}$ is the total query $\mathcal{A}_1$ can make to the Obtain and Verify oracles, there exists an algorithm $\mathcal{C}$ which can output $(g_1^{\frac{1}{x+t}}, t)$ by acting as the simulator for the ABC system as follows:

$\mathbf{Game}_0$. This is the same as the $\mathbf{Game}_0$ in Lemma 1 and we have:

$$\Pr[S_0] = \varepsilon_{\mathsf{imp}} \tag{17}$$

$\mathbf{Game}_1$. The precomputations are the same as the $\mathbf{Game}_1$ in Lemma 2 but $(\mathsf{e}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, a = g_1^{f(x)t_0 - \sum_{i=1}^{q_s} f_i(x)}, a^{x'}, \ldots, a^{x'^n}, b = g_1^{f(x)t_0' - \sum_{i=1}^{q_s} f_i(x)}, c =$

$g_1^{f(x)t_0''+\sum_{i=1}^{q_s} z_i f_i(x)}, g_2, X = g_2^x, X_1, \ldots, X_n)$ as the public key to $\mathcal{A}_3$ where $z_1, \ldots, z_{q_s} \in \mathbb{Z}_p^*$. This gives:

$$\Pr[S_1] = \Pr[S_0] \tag{18}$$

Game$_2$. This is the same as the **Game**$_2$ in Lemma 1 except that, after resetting $\mathcal{A}_2, \mathcal{C}$ simulates the encoded-message signature (t_j, s_j, v_j) on $C_j = (a^{\prod_{i=1}^n (x'+m_{j,i})})^{r'_j} b^{s'_j}$ for $A_j = \{m_{j,1}, \ldots, m_{j,n}\}$ by letting $s_j = z_j - r'_j \prod_{i=1}^n (x' + m_{j,i})$ where:

$$\begin{aligned}
v_j &= ((a^{\prod_{i=1}^n (x'+m_{j,i})})^{r'_j} b^{s'_j + (z_j - r'_j \prod_{i=1}^n (x'+m_{j,i}) - s'_j)} c)^{1/(x+t_j)} \\
&= ((g_1^{\prod_{i=1}^n (x'+m_{j,i})(f(x)t_0 - \sum_{i=1}^q f_i(x))})^{r'_j} g_1^{s_j(f(x)t'_0 - \sum_{i=1}^q f_i(x))} g_1^{f(x)t''_0 + \sum_{i=1}^q z_i f_i(x)})^{1/(x+t_j)} \\
&= (g_1^{f(x)(r'_j t_0 \prod_{i=1}^n (x'+m_{j,i}) + s_j t'_0 + t''_0) - z_j \sum_{i=1}^q f_i(x) + \sum_{i=1}^q z_i f_i(x)})^{1/(x+t_j)} \\
&= g_1^{f_j(x)(r' t_0 \prod_{i=1}^n (x'+m_{j,i}) + s_j t'_0 + t''_0) - z_j \sum_{i=1, i\neq j}^q f_{i,j}(x) + \sum_{i=1, i\neq j}^q z_i f_{i,j}(x)} g_1^{\frac{f_j(x)(z_j - z_j)}{x+t_j}} \\
&= g_1^{f_j(x)(r'_j t_0 \prod_{i=1}^n (x'+m_{j,i}) + s_j t'_0 + t''_0) + \sum_{i=1, i\neq j}^q (z_i - z_j) f_{i,j}(x)}
\end{aligned}$$

and $s''_j = s_j - s'_j$. When the protocol ends, $\mathcal{A}_2$ compiles the credential as $cred_j = (t_j, s_j = s'_j + s''_j, v_j, r'_j, usk_j, A_j)$. As $\mathcal{C}$ simulates the Issue oracle perfectly, by Reset Lemma [8], we have:

$$\Pr[S_2] = \Pr[S_1] + \mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}}(1 + 2^{-p}) \tag{19}$$

Game$_3$**.** This is the same as the **Game**$_3$ in Lemma 1 and we have:

$$\Pr[S_3] = \Pr[S_2] \tag{20}$$

Game$_4$**.** This is the same as the **Game**$_4$ in Lemma 1 and we have:

$$\Pr[S_4] = \Pr[S_3] \tag{21}$$

Challenge: $\mathcal{A}_2$ decided to announce the challenge attribute set $A^* \not\subseteq Q_{o\leftrightarrow i}$. $\mathcal{A}_2$ is still allowed to query the oracles as in **Game**$_2$, **Game**$_3$ and **Game**$_4$ but with the exception $A^* \not\subseteq Q_{o\leftrightarrow i}$ in **Game**$_2$. Since the success probability does not change, we do not reiterate the games involved.

Game$_5$**.** Similar to the **Game**$_5$ in Lemma 1, $\mathcal{F}$ can reset $\mathcal{A}_2$ to extract the elements (t^*, s^*, v^*) of $cred^*$. Since $\mathcal{A}_2$ must output $t^*, s^* \in Q_{o\leftrightarrow i} \cup Q_{p\leftrightarrow v}$ for a $j \in \{1, \ldots, q\}$, $\mathcal{C}$ proceeds to compute $c(x)$ of degree $n-2$ and $d \in \mathbb{Z}_p^*$ from the knowledge of $\{t_1, \ldots, t_q\}$ such that $f_j(x) = c(x)(x + t_j) + d$. Recall that $\mathcal{C}$ will further have the case $v \notin Q_{o\leftrightarrow i} \cup Q_{p\leftrightarrow v}$ or $\mathcal{C}$ already found $x = -t_j$ during

Game$_1$. Subsequently, $\mathcal{C}$ calculates:

$$\begin{aligned}
&\left(v^*/g_1^{f_j(x)(r'^* t_0 \prod_{i=1}^{n}(x'+m_i^*)+s^* t_0'+t_0'')+\sum_{i=1,i\neq j}^{q}(z_i-z^*)f_{i,j}(x)}\right)^{\frac{1}{d(z_j-z^*)}} g_1^{-\frac{c(x)}{d}} \\
&= g_1^{\frac{f_j(x)(z_j-z^*)}{d(z_j-z^*)}} g_1^{-\frac{c(x)}{d}} \\
&= g_1^{\frac{f_j(x)}{d}} g_1^{-\frac{c(x)}{d}} \\
&= g_1^{\frac{c(x)(x+t_j)+d}{d(x+t_j)}} g_1^{-\frac{c(x)}{d}} \\
&= g_1^{\frac{c(x)}{d}+\frac{1}{x+t_j}-\frac{c(x)}{d}} \\
&= g_1^{\frac{1}{x+t_j}}
\end{aligned}$$

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

$$\Pr[S_5] \leq \Pr[S_4] + \sqrt{\varepsilon_{\mathsf{sdh}}} + 2^{-p} \tag{22}$$

and summing up the probability from (17) to (22), we have $\varepsilon_{\mathsf{imp}} \leq \sqrt{\varepsilon_{\mathsf{sdh}}} + \mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}} + (\mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}}+1)/2^{-p}$ as required. The time taken by $\mathcal{C}$ is at most $2t_{\mathsf{sdh}}$ due to reset, besides the final SDH solution extraction which costs $2^{n-2}-n-3$ multiplications in $\mathbb{Z}_p^*$ plus $q+1$ scalar multiplications and one addition in $\mathbb{G}_1$. □

Combining Lemma 1, 2, 3 gives us the following theorem.

Theorem 4. *If an adversary $\mathcal{A}$ $(t_{\mathsf{imp}}, \varepsilon_{\mathsf{imp}})$-breaks the anonymous credential system, then there exists an algorithm $\mathcal{C}$ which $(t_{\mathsf{sdh}}, \varepsilon_{\mathsf{sdh}})$-breaks the SDH assumption such that:*

$$\begin{aligned}
&\varepsilon_{\mathsf{imp}} \leq 3(\sqrt{\varepsilon_{\mathsf{sdh}}} + \mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}} + (\mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}}+1)/2^{-p}), \\
&t_{\mathsf{imp}} \leq 3\,(2t_{\mathsf{sdh}} + t_{\mathsf{add}}) + (2^n - 3n - 8)t_{\mathsf{mul}} + (2(\mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}} + \mathsf{q}_{\mathsf{p}\leftrightarrow\mathsf{v}}) + 4)t_{\mathsf{smul}}
\end{aligned}$$

where $\mathsf{q}_{\mathsf{o}\leftrightarrow\mathsf{i}}, \mathsf{q}_{\mathsf{p}\leftrightarrow\mathsf{v}}$ are the total query made to the Obtain *and* Verify *oracles for n attributes, respectively, while $t_{\mathsf{mul}}, t_{\mathsf{smul}}, t_{\mathsf{add}}$ are the time required for a multiplication in $\mathbb{Z}_p^*$ and a scalar multiplication and an addition in $\mathbb{G}_1$.*

From the proof of Theorem 4, it is clear that the issuing and proof of possession protocols are zero-knowledge protocols, or the reset lemma would fail. Given the fact that a zero-knowledge protocol also possesses witness indistinguishability (WI) [23], it is sufficient to show that the proof of possession protocols are witness indistinguishable in order to prove the anonymity of the ABC system. The WI property implies the adversary $\mathcal{A}$ has only negligible probability in making a correct guess on the identity of the prover, even given access to the Obtain and Issue oracles in the issuing protocol, as well as the Prove and Verify oracles in the proof of possession protocol.

Theorem 5. *If an adversary $\mathcal{A}$ $(t_{\mathsf{ano}}, \varepsilon_{\mathsf{ano}})$-breaks the anonymous credential system, then there exists an algorithm $\mathcal{C}$ which $(t_{\mathsf{wi}}, \varepsilon_{\mathsf{wi}})$-breaks the witness indistinguishability of either the proof of possession protocol or the show proof such that:*

$$\varepsilon_{\mathsf{ano}} = \varepsilon_{\mathsf{wi}}, t_{\mathsf{ano}} = t_{\mathsf{wi}}$$

Proof. Given a WI instance (pk, sk) for the prove system $(P = \mathsf{Prove}, V = \mathsf{Verify})$ in the proof of possession protocol, we show that if $\mathcal{A}$ exists, there exists an algorithm $\mathcal{C}$ which can break the witness indistinguishability of (P, V) with the help of $\mathcal{A}$.

Game$_0$. This is the attack by $\mathcal{A}$ on the real anonymous credential system. Let S be the event of a successful distinguishing, by assumption, we have:

$$\Pr[S_0] = \varepsilon_{\mathsf{ano}} \tag{23}$$

Game$_1$. In this game, $\mathcal{C}$ gives (pk, sk) to $\mathcal{A}$ so that the latter can play the role of user and issuer. When $\mathcal{A}$ acts as issuer to interact with $\mathcal{C}$ to produce a credential $cred_i$ for $\mathcal{C}$'s chosen hidden attribute set $A_i = \{m_{i,1}, \ldots, m_{i,n}\}$. Their interaction is as follows:

1. $\mathcal{C}$ runs the issuing protocol as an honest user u_i with $\mathcal{A}$ as follows:
   (a) $\mathcal{C}$ randomly selects $r_{i,s'}, r_{i,0}, \ldots, r_{i,n} \in \mathbb{Z}_p^*$ and sends the witness $C_i, R_i = \prod_{j=0}^{n} (a^{x'^j})^{r_{i,j}} b^{r_{i,s'}}$ to $\mathcal{A}$.
   (b) $\mathcal{A}$ replies with a challenge $e_i \in \mathbb{Z}_p^*$.
   (c) $\mathcal{C}$ returns the response corresponding to the commitment and witness $z_{i,s'} = r_{i,s'} + e_i s'_i, z_{i,0} = r_{i,0} + e_i r'_{i,0} \alpha_{i,0}, \ldots, z_{i,n} = r_{i,n} + e_i r'_{i,n} \alpha_{i,n}$ to $\mathcal{C}$.
   (d) $\mathcal{A}$ proceeds to next step if $\prod_{j=0}^{n} (a^{x'^j})^{z_{i,j}} b^{z_{i,s'}} = R_i C_i^{e_i}$ holds. Else, $\mathcal{A}$ outputs $\perp$ and stops.
2. $\mathcal{A}$ returns (t_i, s''_i, v_i) to $\mathcal{C}$ as the encoded-message signature on C_i.

Notice that for any two representations $(s'_1, r'_1 \alpha_0, \ldots, r'_1 \alpha_{n-1}), (s'_2, r'_2 \alpha_0, \ldots, r'_2 \alpha_{n-1})$ in the issuing protocol, the distribution of their transcripts are identical to each other from the view of $\mathcal{A}$. This is true even for the non-uniformly distributed attributes $\alpha_0, \ldots, \alpha_{n-1}$ as they have been randomized by r'. To be precise, the witness (C, R) sent by $\mathcal{C}$ in the first step is independent of the representation being used while the response $(z_{s'}, z_0, \ldots, z_{n-1})$ sent by $\mathcal{C}$ in the third step is conditioned on the uniformly distributed witness and the challenge value e selected by $\mathcal{A}$. If $\mathcal{C}$ is using the first representation, the response are distributed uniformly over the representations of (C_1, R_1) and then uniformly distributed over the representation of $R_1 C_1^{e_1}$. If the second representation is used, the response from $\mathcal{C}$ also distributed uniformly over the representation of $R_2 C_2^{e_2}$ as required where the distribution of response value is uniformly random even if $e_1 = e_2$. We do not describe $\mathcal{A}$'s actions as user because that does not affect the distribution of the representations chosen by a $\mathcal{C}$ in the issuing protocol. Therefore, we have:

$$\Pr[S_1] = \Pr[S_0] \tag{24}$$

Game$_2$. In this game, $\mathcal{A}$ acts as verifier to interact with $\mathcal{C}$ to run proof of possession protocol on $cred_i$ for attribute set $A_i = \{m_{i,1}, \ldots, m_{i,n}\}$. $\mathcal{A}$ can also request to run show proof on on $cred_i$ for the AND, OR and NOT statements on a disclosure attribute set A'_i. $\mathcal{C}$ interacts as an honest prover u_i with $\mathcal{A}$ as follows:

1. $\mathcal{C}$ chooses $r_i, y_i, r_{i,r}, r_{i,y}, r_{i,ty}, r_{i,0}, \ldots, r_{i,n}, r_{i,s} \in \mathbb{Z}_p^*$ and sends the witness $v'_i = v^{r_i y_i^{-1}}, v''_i = v'^{r_{i,y}}_i, V_i = v'^{r_{i,y}}_i, Y_i = \prod_{j=0}^{n} (a^{x'^j})^{r_{i,j}} b^{r_{i,s}} c^{r_{i,r}} v'^{r_{i,ty}}$ to $\mathcal{A}$.
2. $\mathcal{A}$ replies with a random challenge $e_i \in \mathbb{Z}_p^*$.
3. $\mathcal{C}$ responds with $z_{i,r} = r_{i,r} + e_i r_i, z_{i,y} = r_{i,y} + e_i y_i, z_{i,ty} = r_{i,ty} - e_i t_i y_i, z_{i,0} = r_{i,0} + e_i r_i r'_i \alpha_{i,0}, \ldots, z_{i,n} = r_{i,n} + e_i r_i r'_i \alpha_{i,n}, z_{i,s} = r_{i,s} + e_i r_i s_i$.
4. $\mathcal{A}$ outputs 1 if the equation $\mathsf{e}(\prod_{j=0}^{n} (a^{x'^j})^{z_{i,j}} b^{z_{i,s}} c^{z_{i,r}} v'^{z_{i,ty}}_i, g_2) = \mathsf{e}(Y_i, g_2)\mathsf{e}((v'^{z_{i,y}}_i / V_i), X)$ holds and 0 otherwise.

The proof of possession protocol is an extension to the issuing protocol where $\mathcal{C}$ proves the additional representation (t, s, r') in the credential. Specifically, $\mathcal{C}$ proves the representation which consists of the randomized representation from the previous issuing protocol $(rs, rr'\alpha_0, \ldots, rr'\alpha_n)$, the blinded credential elements $(t' = ty, s = rs)$ and the blinding factors (r, y). Therefore, for any two representations $(r_1, y_1, ty_1, r_1 s, r_1 r', r_1 s, r_1 r' \alpha_0, \ldots, r_1 r' \alpha_n), (r_2, y_2, ty_2, r_2 s, r_2 r', r_2 s, r_2 r' \alpha_0, \ldots, r_2 r' \alpha_n)$ in the proof of possession protocol, the distribution of their transcripts are identical to each other from the view of $\mathcal{A}$. This is true even if $\mathcal{A}$ knows (t, s) that have been exposed during the issuing protocol, which now have been randomized by r. Thus, the witness (v', v'', V, Y) is independent from the representation being used and the response $(z_r, z_y, z_{ty}, z_0, \ldots, z_n, z_s)$ is conditioned on the uniformly distributed witness and the challenge value e selected by $\mathcal{A}$. The response is thus uniformly distributed over the representation of (v', v'', V, Y) and the representation of $\mathsf{e}(Y, g_2)\mathsf{e}((v'^{z_y}/V), X)$ as required. The same argument applies on the show proof for logical statements, which are the subset of proof of possession protocols. We do not describe $\mathcal{A}$'s actions as prover because that does not affect the distribution of the representations chosen by $\mathcal{C}$ in the proof of possession protocol. Thus, we have:

$$\Pr[S_2] = \Pr[S_1] \tag{25}$$

Game$_3$. $\mathcal{A}$ decides the two user identities u_1, u_2 and the attribute set $A^* \subseteq A_0 \bigcap A_1$ which he wishes to challenge upon. $\mathcal{C}$ follows $\mathcal{A}$ to announce u_1, u_2 and A^*, A_0, A_1 as its challenge to $\mathcal{C}$'s challenge oracle and receives the challenge bit $b \in \{0, 1\}$ in return. $\mathcal{C}$ acts as the man in the middle to pass the messages in between its challenge oracle as the prover u_b and $\mathcal{A}$ as the verifier to complete the protocol $\pi_{2,b} = (\mathsf{Prove}(pk, A_b, cred_b, A^*), \mathsf{Verify}(pk, A^*))$ for a polynomially many time as requested by $\mathcal{A}$. $\mathcal{A}$ can also continue to act as the issuer and verifier as in **Game$_1$** and **Game$_2$** from time to time. By assumption, witnesses and responses generated by $\mathcal{C}$'s challenge oracle are uniformly distributed. This gives:

$$\Pr[S_3] = \Pr[S_2] \tag{26}$$

Game$_4$. Finally, $\mathcal{A}$ outputs a guess b' on user identity and $\mathcal{C}$ sends b' to its challenge oracle. If $b' = b$, $\mathcal{C}$ breaks the WI of the proof of possession protocol and this gives:

$$\Pr[S_4] = \varepsilon_{\mathsf{wi}} \tag{27}$$

Summing up the equation from (23) to (27), we have $\varepsilon_{\mathsf{ano}} = \varepsilon_{\mathsf{wi}}$. Since $\mathcal{C}$ and $\mathcal{A}$ took the same time in the games, we have $t_{\mathsf{ano}} = t_{\mathsf{wi}}$ as required. □

6 Discussion

In this section, we highlight the advantages and limitation of our proposed ABC system against those built on the SDH-based CL signature in the literature.

6.1 Efficiency

To the best of our knowledge, our ABC system is the first ABC system from SDH-based CL signature which can efficiently support the logical statements in the show proof. The previous ABC system has high complexity in the show proof based on the traditional encoding [15] and they do not support the NOT statement. Recalling the credential from Au et al.'s basic ABC system [6] as an example, the traditional encoding fixes each attribute to a specific base such that:

$$A = (g_2^{m_1} \dots g_{L+1}^{m_L} g_1^s g_0)^{\frac{1}{x+e}}$$

whose show proof has the same complexity on the AND statement but not on the OR statements. For the OR statement in the traditional encoding, given the list L, a prover has to run a complex witness indistinguishability (WI) protocol that covers every attribute in L. To be precise, the WI protocol requires the prover to run one and simulate $|L| - 1$ other proof of possession protocols. Our show proof for the OR and NOT statements on the contrary, has a significantly lower complexity due to its algebraic properties. The comparison of the techniques to support multiple attributes in an ABC system based on SDH-based CL signature is shown in Table 2 while the concrete protocol complexity is shown in Table 3. The single attribute ABC systems [33, 2, 29] are excluded but some state-of-the-art ABC systems [15, 12, 24] are included as a benchmark.

Although the complexity analysis in the tables may not be accurate due to the different natures of the covered ABC systems, we argue that the result is adequately generated. Table 2 is a normalized view where we consider only the asymptotic computational complexity for expensive operations such as the scalar multiplication (or modular exponentiation in RSA) and pairing operations in a protocol. The more comprehensive comparison for our proposed ABC system and other ABC systems is depicted in Table 3 based on the ABC system definitions

Table 2. Asymptotic protocol complexity in ABC systems based on SDH-based CL Signature scheme.

Property			**ABC system**				
Technique			SDH-based CL			RSA-based CL	SPS
			Trad. Encd.	Accumulator	This Work	Prime Encd.	Commitment
Possession Complexity			$O(n)$	$O(n)$	$O(n)$	$O(1)$	$O(n)$
Show Proof Complexity (l-of-n attributes)	AND	Prover	$O(n-l)$	$O(n-l)$	$O(n-l)$	$O(1)$	$O(n-l)$
		Verifier	$O(l)$	$O(l)$	$O(l)$	$O(1)$	$O(l)$
	OR	Prover	$O(nl)$	$O(n-l)$	$O(n-l)$	$O(1)$	$O(n-l)$
		Verifier	$O(l)$	$O(l)$	$O(l)$	$O(1)$	$O(l)$
	NOT	Prover	✗	✗	$O(n-l)$	$O(1)$	$O(n-l)$
		Verifier	✗	✗	$O(l)$	$O(1)$	$O(l)$
Flexible Attribute Indexing			✗	✗	✓	✓	✓
Schemes			[6, 16, 7]	[31, 34]	Ours	[15]	[12, 24]

Table 3. Comparison of ABC systems based on SDH-based CL Signature scheme.

ABC system	**Credential Size**	**Proof of Possession Complexity**
ASM [6]	$1\|\mathbb{G}_1\| + (n+2)\|\mathbb{Z}_p\|$	$3M_1(1) + 2M_1(n+4) + 3P$
CKS [16]	$2\|\mathbb{G}_1\| + (n+2)\|\mathbb{Z}_p\|$	$3M_1(1) + 16M_1(2) + 4M_1(3) + 2M_1(n+4) + 12P$
SNF [31]	$5\|\mathbb{G}_1\| + (n+2)\|\mathbb{Z}_p\|$	$8M_1(1) + 12M_1(2) + 6M_1(3) + 2M_1(5) + 2M_1(n) + 15P$
ZF [34]	$7\|\mathbb{G}_1\| + (n+2)\|\mathbb{Z}_p\|$	$8M_1(1) + 8M_1(2) + 6M_1(3) + 2M_1(n+6) + 16P$
BBDT [7]	$2\|\mathbb{G}_1\| + (n+2)\|\mathbb{Z}_p\|$	$M_1(1) + 6M_1(2) + 2M_1(n+2)$
CG [15]	$1\|\mathbb{Z}_N\| + 1\|\mathbb{Z}_{NMk}\| + 1\|\mathbb{Z}_{M+2}\| + n\|\mathbb{Z}_{M/n}\|$	$2E(4) + (21k+3)E(1)$
CDHK [12]	$(n+5)\|\mathbb{G}_1\| + 2\|\mathbb{G}_2\|$	$4M_1(1) + (30+6n)M_1(2) + M_1(n) + 4M_2(1) + 12M_2(2) + 42P$
FHS [24]	$(n+3)\|\mathbb{G}_1\| + 1\|\mathbb{G}_2\| + 2\|\mathbb{Z}_p\|$	$4M_1(1) + 30M_1(2) + M_1(n) + M_2(1) + 6M_2(2) + 41P$
Ours	$1\|\mathbb{G}_1\| + (n+3)\|\mathbb{Z}_p\|$	$3M_1(1) + 2M_1(n+4) + 3P$

Note: We view n as the total of string attributes and finite set attributes in [16, 31, 34].

in Section 3.4. For instance, we include attributes in the credential for every ABC system where the credential size may be higher than what it was in the original works. Besides, we exclude computations for proprietary properties such as user revocation, pseudonymization, accumulator initialization, prime hashing and so on which are not covered by our definition. In the tables, q and n are the prime group order and total attributes in a credential. The symbol $|\cdot|$ denotes the size of an element while $M_1(\cdot), M_2(\cdot), P$ denotes denote the multi-scalar multiplications in $\mathbb{G}_1, \mathbb{G}_2$ and pairing operation respectively. For an example, $|\mathbb{G}_1|$ is the size of the bilinear group $\mathbb{G}_1$ and $M_1(1)$ is a scalar multiplication in $\mathbb{G}_1$ with no point addition. For the only RSA-based ABC system [15] in the table, we use N, M and k to represent the RSA modulus, size of attribute space and

security parameter respectively. Similarly, we denote the multi-exponentiation in RSA by the symbol $E(\cdot)$.

From Table 2 and Table 3, we can conclude that our ABC system has a construction as efficient as that of the ABC system based on traditional encoding, yet the show proofs are as expressive as the state-of-the-art schemes. This is justified from our proof of possession and show proofs which, besides the ABC system based on prime encoding [15], record the lowest asymptotic complexity in Table 2, and lowest concrete complexity in Table 3. Our ABC system requires the verifier to compute only three pairings for proof of possession protocol, five pairings for the show proof on AND and NOT statements, and eight pairings for OR statement, all independent from the value l. Moreover, our ABC system supports flexible attribute indexing and proven secure to be fully anonymous. We also summarize the security properties of ABC systems which are based on SDH-based CL signature scheme and the signature schemes which share the similar algebraic property in Table 4[1]. If an ABC system claims only weak anonymity such that it does not consider blind issuing or assumes a trusted issuer, we mark ○ for the anonymity in the issuing protocol.

Table 4. Security properties of related ABC systems.

ABC System	Signature	ABC System	Anonymity		Security
	Unforgeability	Impersonation	Issue	Possession	Model
ASM [6]	●	●	●	●	RO
TAKS [33]	●	●	●	●	RO
AMO [2]	●	●	○	●	Standard
CKS [16]	●	●	○	○	Standard
SNF [31]	●	●	○	○	Standard
ZF [34]	●	●	◐	◐	Standard
BBDT [7]	●	●	●	●	Standard
RVH [29]	●	●	○	○	Standard
CDDH [11]	●	●	○	●	Standard
CG [15]	●	●	○	○	Standard
CDHK [12]	●	●	○	○	CRS
FHS [24]	●	●	●	●	Generic
Ours	●	●	●	●	Standard

●: security proof provided ◐: security claim provided ○: no security claim

6.2 Other Applications

We see that our encoding method is applicable not only to ABC system [6, 16, 31, 34, 7, 33, 29, 11], but also to other primitives based on SDH-based CL

[1] Camenisch et al.'s [12] and Ringers et al.'s [29] ABC systems consider the security property of unlinkability but not anonymity.

signature. For instance, our encoding can be applied on oblivious transfer [13] and e-cash [5] to support signing on multiple messages. It also allows efficient construction of access policy for attribute-based signature [4] and access control [27] schemes. Last but not least, the encoding method works on RSA-based CL signature as well. Although our encoding does not yield a show proof which is as efficient as the prime encoding, it avoids the hash to prime procedure which may be of independent interest.

6.3 Limitation

Although our proposed ABC system is secure in the standard model, its security will be downgraded to the random oracle model when the proof of possessions protocol are converted into the NIZK protocol. As observed by Camenisch et al. [12], this is a limitation of the CL signature and the solution is in the structure-preserving signature schemes. Abe et al. [1] proposed an automorphic signature whose NIZK proofs on signature can be proven secure in the common reference string model. We notice that their automorphic signature is an extension of the SDH-based CL signature which our ABC system is based on. It is interesting to know whether our ABC system can enjoy NIZK proof by applying the same extension to our underlying encoded-message signature scheme.

7 Conclusion

We introduced a collision resistant encoding which resulted in an efficient SDH-based CL signature for multiple messages. Subsequently, we extend the encoded-message signature scheme into a multi-show anonymous credential systems and proves its security against impersonation and anonymity in the standard model. The proposed ABC system can support show proof on AND, OR and NOT statements besides being the most efficient and secure ABC system built upon SDH-based CL signature to-date.

8 Acknowledgement

This work was supported by the ERC Starting Grant Confidentiality-Preserving Security Assurance (CASCAde, GA n°716980).

Bibliography

[1] Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Structure-preserving signatures and commitments to group elements. In Tal Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, pages 209–236, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[2] Norio Akagi, Yoshifumi Manabe, and Tatsuaki Okamoto. An efficient anonymous credential system. In Gene Tsudik, editor, *Financial Cryptography and Data Security*, pages 272–286, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[3] Hiroaki Anada, Seiko Arita, Sari Handa, and Yosuke Iwabuchi. Attribute-based identification: Definitions and efficient constructions. In Colin Boyd and Leonie Simpson, editors, *Information Security and Privacy*, pages 168–186, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[4] Hiroaki Anada, Seiko Arita, and Kouichi Sakurai. Attribute-based two-tier signatures: Definition and construction. In Soonhak Kwon and Aaram Yun, editors, *Information Security and Cryptology - ICISC 2015*, pages 36–49, Cham, 2016. Springer International Publishing.

[5] M. H. Au, J. K. Liu, J. Fang, Z. L. Jiang, W. Susilo, and J. Zhou. A new payment system for enhancing location privacy of electric vehicles. *IEEE Transactions on Vehicular Technology*, 63(1):3–18, Jan 2014.

[6] Man Ho Au, Willy Susilo, and Yi Mu. Constant-size dynamic k-taa. In Roberto De Prisco and Moti Yung, editors, *Security and Cryptography for Networks*, pages 111–125, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[7] Amira Barki, Solenn Brunet, Nicolas Desmoulins, and Jacques Traoré. Improved algebraic macs and practical keyed-verification anonymous credentials. In Roberto Avanzi and Howard Heys, editors, *Selected Areas in Cryptography – SAC 2016*, pages 360–380, Cham, 2017. Springer International Publishing.

[8] Mihir Bellare and Adriana Palacio. Gq and schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In Moti Yung, editor, *Advances in Cryptology — CRYPTO 2002*, pages 162–177, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[9] Dan Boneh and Xavier Boyen. Short signatures without random oracles and the sdh assumption in bilinear groups. *Journal of Cryptology*, 21(2):149–177, Apr 2008.

[10] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matt Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, pages 41–55, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[11] Jan Camenisch, Manu Drijvers, Petr Dzurenda, and Jan Hajny. Fast keyed-verification anonymous credentials on standard smart cards. In Gurpreet Dhillon, Fredrik Karlsson, Karin Hedström, and André Zúquete, editors,

ICT Systems Security and Privacy Protection, pages 286–298, Cham, 2019. Springer International Publishing.

[12] Jan Camenisch, Maria Dubovitskaya, Kristiyan Haralambiev, and Markulf Kohlweiss. Composable and modular anonymous credentials: Definitions and practical constructions. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology – ASIACRYPT 2015*, pages 262–288, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[13] Jan Camenisch, Maria Dubovitskaya, and Gregory Neven. Oblivious transfer with access control. In *Proceedings of the 16th ACM Conference on Computer and Communications Security*, CCS '09, pages 131–140, New York, NY, USA, 2009. ACM.

[14] Jan Camenisch and Thomas Groß. Efficient attributes for anonymous credentials. In *Proceedings of the 15th ACM conference on Computer and communications security*, pages 345–356. ACM, 2008.

[15] Jan Camenisch and Thomas Groß. Efficient attributes for anonymous credentials. *ACM Trans. Inf. Syst. Secur.*, 15(1):4:1–4:30, March 2012.

[16] Jan Camenisch, Markulf Kohlweiss, and Claudio Soriente. An accumulator based on bilinear maps and efficient revocation for anonymous credentials. In Stanisław Jarecki and Gene Tsudik, editors, *Public Key Cryptography – PKC 2009*, pages 481–500, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[17] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In Birgit Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001*, pages 93–118, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[18] Jan Camenisch and Anna Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In Moti Yung, editor, *Advances in Cryptology — CRYPTO 2002*, pages 61–76, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[19] Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In Stelvio Cimato, Giuseppe Persiano, and Clemente Galdi, editors, *Security in Communication Networks*, pages 268–289, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[20] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Matt Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, pages 56–72, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[21] Melissa Chase, Sarah Meiklejohn, and Greg Zaverucha. Algebraic macs and keyed-verification anonymous credentials. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 1205–1216, New York, NY, USA, 2014. ACM.

[22] David Chaum. Security without identification: Transaction systems to make big brother obsolete. *Commun. ACM*, 28(10):1030–1044, October 1985.

[23] U. Feige and A. Shamir. Witness indistinguishable and witness hiding protocols. In *Proceedings of the Twenty-second Annual ACM Symposium on*

Theory of Computing, STOC '90, pages 416–426, New York, NY, USA, 1990. ACM.

[24] Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. Structure-preserving signatures on equivalence classes and constant-size anonymous credentials. *Journal of Cryptology*, 32(2):498–546, Apr 2019.

[25] T. Groß and S.-Y. Tan. On the anonymity of a graph signature scheme. Newcastle University ePrint Archive, 2019. https://www.ncl.ac.uk/media/wwwnclacuk/schoolofcomputingscience/files/trs/1527.pdf.

[26] Thomas Groß. Signatures and efficient proofs on committed graphs and np-statements. In Rainer Böhme and Tatsuaki Okamoto, editors, *Financial Cryptography and Data Security*, pages 293–314, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[27] J. K. Liu, M. H. Au, X. Huang, R. Lu, and J. Li. Fine-grained two-factor access control for web-based cloud computing services. *IEEE Transactions on Information Forensics and Security*, 11(3):484–497, March 2016.

[28] Tatsuaki Okamoto. Efficient blind and partially blind signatures without random oracles. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, pages 80–99, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[29] Sietse Ringers, Eric Verheul, and Jaap-Henk Hoepman. An efficient self-blindable attribute-based credential scheme. In Aggelos Kiayias, editor, *Financial Cryptography and Data Security*, pages 3–20, Cham, 2017. Springer International Publishing.

[30] Sven Schäge. Tight proofs for signature schemes without random oracles. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, pages 189–206, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[31] Amang Sudarsono, Toru Nakanishi, and Nobuo Funabiki. Efficient proofs of attributes in pairing-based anonymous credential system. In Simone Fischer-Hübner and Nicholas Hopper, editors, *Privacy Enhancing Technologies*, pages 246–263, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[32] Syh-Yuan Tan, Swee-Huay Heng, Bok-Min Goi, and SangJae Moon. Fuzzy identity-based identification scheme. In Dominik Ślęzak, Tai-hoon Kim, Jianhua Ma, Wai-Chi Fang, Frode Eika Sandnes, Byeong-Ho Kang, and Bongen Gu, editors, *U- and E-Service, Science and Technology*, pages 123–130, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[33] Patrick P. Tsang, Man Ho Au, Apu Kapadia, and Sean W. Smith. Blacklistable anonymous credentials: Blocking misbehaving users without ttps. In *Proceedings of the 14th ACM Conference on Computer and Communications Security*, CCS '07, pages 72–81, New York, NY, USA, 2007. ACM.

[34] Yan Zhang and Dengguo Feng. Efficient attribute proofs in anonymous credential using attribute-based cryptography. In Tat Wing Chim and Tsz Hon Yuen, editors, *Information and Communications Security*, pages 408–415, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.