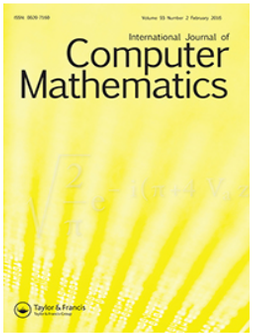

International Journal of Computer Mathematics

ISSN: 0020-7160 (Print) 1029-0265 (Online) Journal homepage: https://www.tandfonline.com/loi/gcom20

Privacy-preserving and efficient attributes proof based on selective aggregate CL-signature scheme

Nan Guo, Tianhan Gao & Jia Wang

To cite this article: Nan Guo, Tianhan Gao & Jia Wang (2016) Privacy-preserving and efficient attributes proof based on selective aggregate CL-signature scheme, International Journal of Computer Mathematics, 93:2, 273-288, DOI: 10.1080/00207160.2014.915961

To link to this article: https://doi.org/10.1080/00207160.2014.915961

Published online: 22 May 2014.

International Journal of Computer Mathematics, 2016
Vol. 93, No. 2, 273–288, http://dx.doi.org/10.1080/00207160.2014.915961

Taylor & Francis
Taylor & Francis Group

Privacy-preserving and efficient attributes proof based on selective aggregate CL-signature scheme

Nan Guo[a], Tianhan Gao[b*] and Jia Wang[a]

[a]*College of Information Science and Engineering, Northeastern University, Shenyang, China;* [b]*College of Software, Northeastern University, Shenyang, China*

(*Received 6 January 2014; revised version received 18 March 2014; accepted 9 April 2014*)

We propose efficient attributes proof protocols in an anonymous and unlinkable fashion. The core idea is issuing anonymous credentials for each single attribute and proving relations over attributes by selectively aggregating individual anonymous credentials. A selective aggregate Camenisch–Lysyanskaya (CL)-signature scheme is presented to construct anonymous credentials. It is existentially unforgeable against adaptively chosen-message attack under CL-signature scheme on the Lysyanskaya–Rivest–Sahai–Wolf assumption. It has constant complexity in verification of multiple signatures. Users can select which attributes and the corresponding individual anonymous credentials are involved in the proof. They can prove the possession of attributes over logic relations including AND and OR, and the possession of a single attribute over comparison relations including inequality to a given value and belonging to a given interval. The efficiency analysis shows that the resulting protocols have advantages in computation cost; the AND relation proof and comparison relation proofs have constant complexity w.r.t. the number of attributes, and the OR relation proof has linear complexity only w.r.t. the number of attributes as required.

Keywords: privacy; anonymous credential; attributes proof; aggregate signature

2010 AMS Subject Classifications: 11T71; 94A60

1. Introduction

The information exchanged via the Internet has dramatically changed, from exchanging scientific and professional information to enormous amount of personal information. The management of identity attributes raises a number of challenges. On one hand, attributes need to be shared to facilitate user authentication and access control. Service providers authorize the access request by a user's attributes which are more available than identities or roles. On the other hand, individuals' attributes need to be protected from privacy leakage as they may convey sensitive information and be the target of attack. As far as privacy issue is concerned, we cannot exclude the attacks from insiders in the potentially untrusted authorities [16,22]; they have become threats of identity theft and abuse [2].

An anonymous credential is a privacy-preserving technology that can meet privacy requirements for attribute-based authentication and access control. Users obtain a signature from an issuing authority on a number of attributes and, at later time, can convince verifiers that they indeed

*Corresponding author. Email: gaoth@mail.neu.edu.cn

possess a signature on those attributes. Individual transactions are anonymous and unlinkable by default and users can select which portion of a credential to reveal, which portions to keep hidden, and what relations between certified attributes to expose [10]. The user can also prove complex relations of the attributes using logic relations, such as AND and OR, and comparison relations, such as $=, \neq, \leq$ and $\geq$. AND relation is used when proving the possession of all of the multiple attributes. OR relation is used when proving the possession of one of multiple attributes.

In the literature, there exit various anonymous credential systems. However, there are only two kinds of practical anonymous credential systems, the Camenisch–Lysyanskaya (CL) Idemix based on group signature and the Brands U-Prove based on blind signature. Camenisch and Lysyanskaya [9] came up with an efficient CL-signature scheme and constructed a multi-use CL-anonymous credential system from bilinear mappings. It also introduces a construction based on the Boneh–Boyen–Shacham group signature, which enables selective disclosure and unlinkable multi-use. However, attributes proof with such anonymous credentials suffers from a linear complexity w.r.t. the total number of attributes. This limitation makes them unfit for many practical applications.

The existing approaches [1,7,8,10,14,21] to solve the linear complexity of attributes proof is focus on binary and finite-set attributes and use cryptographic accumulator to compress this type of attributes into a single one. However, attributes proof with accumulator-based anonymous credentials requires many extra pairings to verify accumulator, and the size of public key is dependent on the number of attributes as well.

As far as efficiency issue is concerned, aggregate signature schemes [5,15] are worth mentioning. They enable us to compress a number of signatures, which are on distinct messages issued by distinct parities, into a single one. They have short public key size, short aggregate signature size and efficient aggregate verification. However, to the best of our knowledge, there are no literatures about constructing efficient attributes proof based on aggregate signature. In this paper, we present a selective aggregate signature scheme based on the CL-signature scheme [9] under the Lysyanskaya–Rivest–Sahai–Wolf (LRSW) assumption and construct anonymous credentials and attributes proof protocols to solve linear complexity.

Our contribution is to use the concept of aggregate signature to solve linear complexity of attributes proof. The core idea of our proposal is that given l attributes to be certified by an issuer, each single attribute is certified in an individual credential; later on users can selectively disclose any n out of l attributes and aggregate the corresponding individual signatures, then prove the possession of the aggregate signature on n disclosed attributes all at once. The efficiency analysis shows that the aggregate-based attributes proof has advantages on computation cost w.r.t. the number of pairings and exponentiations.

This paper is a revised and expanded version of [12] which extends the original CL-anonymous credential [9] with AND, OR, Equality and Interval proof over attributes, while this paper mainly focus on efficiency issue of attributes proof. It proposes a selective aggregate signature scheme as the building block to construct efficient attributes proof protocols.

The organization of the remained is as follows. Section 2 covers related literature for anonymous credential systems as well as existing approaches for attributes proof. Section 3 gives the preliminaries about bilinear maps, Pedersen commitment scheme, discrete logarithm representation, the CL-signature scheme and the Boudot-interval proof protocols. In Section 4, we present a selective aggregate CL-signature scheme and prove the security. In Section 5, we analyse privacy requirements of anonymous credentials and give the issuance protocol. Section 6 presents AND and OR relation proof over multiple attributes, as well as interval and inequality proof over a single attribute, respectively. Section 7 shows efficiency analysis. Finally, Section 8 is the conclusion.

2. Related work

Camenisch and Groß [7,8], Sudarsono *et al.* [21], Herranz *et al.* [13], Begum *et al.* [1] focused on attributes proof over multiple attributes. In 2008 and 2012, Camenisch and Groß [7,8] proposed a RSA-based anonymous credential with efficient attributes proof. It encodes discrete binary and finite-set attribute values as prime numbers and use the divisibility property for efficient proofs of their presence or absence. The complexity only depends on the number of string/integer attributes, and binary and finite-set attributes are free. In 2011, Sudarsono [21] utilized extended BBS+ signatures to certify a set of attributes as the accumulator, and used zero-knowledge proofs of BBS+ signatures and accumulators to prove AND and OR relations with constant complexity in the number of finite-set attributes. In 2013, Begum *et al.* [1] handled the complex logical relations on attributes as conjunctive normal form and disjunctive normal form formulas. However, the size of public key is dependent on the number of attribute values, and the extra number of pairings involved in the accumulator-based anonymous credentials increases a lot.

There are some researches [13,18,20] about the attribute-based signature introducing attributes proofs such as NOT, AND, OR, and threshold gates. However, they work in a traceable and linkable way and are not available for anonymous environment. Li and Li [17] constructed Oblivious Commitment Based Envelope (OCBE) protocols which offers proofs of comparison predicates such as $=$, $\neq$, $\geq$ and $\leq$. Unfortunately, the protocols for predicates suffer from linear complexity in the binary number of user's attribute values. Bichsel *et al.* [3] showed the details of comparison predicates supported in the Identity Mixer and the U-Prove technologies, which are implemented using Boudot-interval proofs [6] with constant complexity.

As far as privacy is concerned, it is crucial that insider threats are carefully taken into consideration when designing security and privacy of credentials. Slamanig *et al.* [19] discussed insider threats in eHealth application that the user has no guarantee that the provider always preserve the users' privacy claims; a person and her requested data are linkable to draw potentially compromising conclusion about her. Bjones *et al.* [4] states that an electronic identity server under control of an insider attacker has the ability to impersonate every user at applications using eIDs for authentication. For example, insiders can copy or alter user's credentials and as such steal the identity of a user. In general, in a federation scenario, the insiders or outsiders who learn a user's credentials can impersonate the user and get access to the assets at different applications involved in the federation. If the properties of anonymity, unlinkability and selective disclosure of attributes, provided by anonymous credentials, are realized, an insider is not able to learn the user's identity, and link a set of transactions accomplished by a user either. As a consequence, a person who uses the privacy sensitive applications does not need to rely on trusting the provider anymore, e.g. concerning the divulgement of her data.

However, when a person uses an anonymous credential to accomplish attributes proof, if attributes proof protocol has computation intensive cryptographic building blocks, it will turn out to be very resource consuming. In this paper we use the concept of aggregate signature to solve the linear complexity of attributes proof. In 2003, Boneh, Gentry, Lynn and Shacham [5] constructed an efficient aggregate signature from a Boneh–Lynn–Shacham short signature scheme based on bilinear maps. Aggregate signatures are useful for reducing the size of certification chains (by aggregating all signatures in the chain) and for reducing message size in secure routing protocols such as Secure Border Gateway Protocol. Lee *et al.* [15] proposed two aggregate signature schemes based on the CL-signature scheme. The first one is an efficient sequential aggregate signature scheme. The second one is an efficient synchronized aggregate signature scheme. They both take constant number of pairings and l number of exponentiations (l being the number of signers). However, to the best of our knowledge, there is not any aggregate signature scheme to construct attributes proof protocols for efficiency reason. In

this paper, the proposed attributes proof protocols based on aggregate signature outperforms the accumulator-based protocols w.r.t. the number of exponentiations and pairings, and satisfies more relation proof over attributes than prime number-based approach as well. Thus it is more practical to be utilized for attributes-based authentication and access control in the context of insider threats.

3. Preliminaries

Before presenting the proposed protocols, we first review a few cryptographic primitives consisting of bilinear maps, Pedersen commitment, discrete logarithm representation, the CL-signature scheme and Boudot-Interval proofs.

3.1 *Bilinear maps*

Let G_1 and G_2 be two (multiplicative) cyclic groups of the prime order p, with an additional group G_T such that $|G_1| = |G_2| = |G_T|$. g_1 is a generator of G_1 and g_2 is the generator of G_2. A bilinear map is a map e: $G_1 \times G_2 \rightarrow G_T$ with the following properties:

- *Bilinear*: for all $u \in G_1, v \in G_2$, and $a, b \in \mathbb{Z}_p, e(u^a, v^b) = e(u, v)^{ab}$.
- *Non-degenerate*: $e(g_1, g_2) \neq 1$.
- *Computability*: There is an efficient algorithm for computing e.

3.2 *Pedersen commitment*

In the Pedersen Commitment scheme, which is an unconditionally hiding and computational binding commitment scheme and based on the discrete logarithm problem, there is a finite multiplicative cyclic group G of prime order q involved along with a generator $g \in G$ and an element $h \in G$ such that it is hard to find an integer α such that $h = g^\alpha$. Given a message x, the User picks $r \in_R \mathbb{Z}_p$ and computes the commitment $M = g^x h^r$. The User runs a zero-knowledge proof of knowledge protocol to open the commitment without showing the values(x, r):

$$\mathrm{PK}\{(x, r) : M = g^x h^r\}. \tag{1}$$

3.3 *Discrete logarithm representation, DLREP*

G is a multiplicative cyclic group of prime order q, let $g_0, g_1, \ldots, g_l$ and y be element of group G. The tuple $(x_0, x_1, \ldots, x_l) \in \mathbb{Z}_q$ is called a DL-representation of the product $y = {g_0}^{x_0} {g_1}^{x_1} \ldots {g_l}^{x_l}$ mod q with respect to the generators $(g_0, g_1, \ldots, g_l)$. The User runs the zero-knowledge proof of knowledge protocol to prove the DL-representation of y.

$$PK\{(x_0, x_1, \ldots, x_l) : y = {g_0}^{x_0} {g_1}^{x_1} \ldots {g_l}^{x_l}\}. \tag{2}$$

3.4 *The CL-signature scheme under the LRSW assumption*

Camenisch and Lysyanskaya [9] proposed a signature scheme which is correct and secure under the LRSW assumption. Suppose a setup algorithm *Setup* that, on input the security parameter 1^k, outputs the setup for $G = \langle g \rangle$ and $\mathsf{G} = \langle \mathsf{g} \rangle$, two groups of prime order $q = \theta(2^k)$ that have a non-degenerate efficiently computable bilinear map $e : G \times G \rightarrow \mathsf{G}$. It consists of the following algorithms:

Key generation. Run the *Setup* algorithm to generate $(q, G, \mathsf{G}, g, \mathsf{g}, e)$. Choose $x, y, z \in_R \mathbb{Z}_q$. Let $X = g^x, Y = g^y, Z = g^z$. Set $sk = (x, y, z), pk = (q, G, \mathsf{G}, g, \mathsf{g}, e, X, Y, Z)$.

Signature. On input message (m, r), secret key $sk = (x, y, z)$, and public key $pk = (q, G, \mathsf{G}, g, \mathsf{g}, e, X, Y, Z)$, choose a random $a \in_R G$, let $A = a^z, b = a^y, B = A^y$ and $c = a^{x+xym} A^{xyr}$, output $\sigma = (a, b, A, B, c)$.

Verification. On input $pk = (q, G, \mathsf{G}, g, \mathsf{g}, e, X, Y, Z)$, message (m, r), and purported signature $\sigma = (a, b, A, B, c)$, check the following: (1) A was formed correctly: $e(a, Z) = e(g, A)$. (2) b and B were formed correctly: $e(a, Y) = e(g, b)$ and $e(A, y) = e(g, B)$. (3) c was formed correctly:

$$e(X, a) \cdot e(X, b)^m \cdot e(X, B)^r = e(g, c). \tag{3}$$

Note that the signature itself can be distributed in a way that is information-theoretically independent of the message being signed in the case that what is being signed is an information-theoretically secure Pedersen commitment of the message. Thus, the values $(g^m Z^r, a, A, b, B, c)$ are information-theoretically independent of m if r is chosen randomly. This will become crucial when using this signature scheme in the context of an anonymous credential system.

3.5 *Boudot-interval proofs*

For the proofs that a committed number belongs to an interval, we now list a few proofs of knowledge protocols introduced in [6].

Prove that two commitments hide the same secret. Given two commitments $E = E(x, r_1) = g^x h^{r_1}$ and $F = E(x, r_2) = g^x h^{r_2}$ to the message x. The Prover proves to the Verifier that E and F hide the same secret x as follows:

$$\mathrm{PK}\{(x, r_1, r_2) : E = g^x h^{r_1} \wedge F = g^x h^{r_2}\}. \tag{4}$$

Prove that a committed number belongs to an interval. Given a commitment $E = E(x, r) = g^x h^r$, the Prover proves to the Verifier that the committed number x lies in $[a, b]$ as follows:

$$\mathrm{PK}\{(x, r) : E = g^x h^r \wedge x \in [a, b]\}. \tag{5}$$

4. Construction of the selective aggregate CL-signature scheme

In this section, we give a novel selective aggregate CL-signature scheme which is extended from the original CL signature. The goal of this signature scheme is to construct an anonymous credential with efficient attributes proof.

Suppose that we have a setup algorithm *Setup*: $(q, G, \mathsf{G}, g, \mathsf{g}, e) \leftarrow Setup(1^k)$, that on input the security parameter 1^k, outputs the setup for $G = \langle g \rangle$ and $\mathsf{G} = \langle \mathsf{g} \rangle$, two groups of prime order $q = \theta(2^k)$ that have a non-degenerate efficiently computable bilinear map $e : G \times G \to \mathsf{G}$.

Key generation. Run the *Setup* algorithm to generate $(q, G, \mathsf{G}, g, \mathsf{g}, e)$. Choose $y, z \in_R \mathbb{Z}_q$, and for $1 \le i \le l$, $x_i \in_R \mathbb{Z}_q$. Let $Y = g^y, Z = g^z$ and, for $1 \le i \le l, X_i = g^{x_i}$. Set $sk = (y, z, x_1, \ldots, x_l), pk = (q, G, \mathsf{G}, g, \mathsf{g}, e, \{X_i\}, Y, Z)$.

Sign. On input message $(r, m_1, \ldots, m_l)$, secret key $sk = (y, z, x_1, \ldots, x_l)$, and public key $pk = (q, G, \mathsf{G}, g, \mathsf{g}, e, \{X_i\}, Y, Z)$, choose a random $a \in_R G$, compute $b \leftarrow a^y, A \leftarrow a^z, B \leftarrow A^y$; for each $i \in \{1, l\}$, let $c_i \leftarrow a^{x_i + x_i yr} A^{x_i y m_i}$, output $(a, b, A, B, \{c_i\})$ as signature.

Verify. On input public key $pk = (q, G, \mathsf{G}, g, \mathsf{g}, e, \{X_i\}, Y, Z)$, ith message (r, m_i) and ith signature (a, b, A, B, c_i) where $1 \le i \le l$, check if $e(a, Y) = e(b, g), e(a, Z) = e(A, g), e(A, Y) = e(B, g)$ and $e(X_i, a) \cdot e(X_i, b^r) \cdot e(X_i, B^{m_i}) = e(g, c_i)$ hold.

Aggregate. On input k signatures indexed by $\{j_1, \ldots, j_k\} \subseteq \{1, \ldots, l\}$ as required, compute $c \leftarrow \prod_{i=1}^{k} c_{j_i}$, and output the aggregate signature (a, b, A, B, c).

AggVerify. On input public key $pk = (q, G, \mathsf{G}, g, \mathsf{g}, e, \{X_i\}, Y, Z)$, k messages indexed by $\{j_1, \ldots, j_k\} \subseteq \{1, \ldots, l\}$ as required, and aggregate signature (a, b, A, B, c), check if $e(a, Y) = e(g, b), e(a, Z) = e(g, A), e(A, Y) = e(g, B)$ and $e(\prod_{i=1}^{k} X_{j_i}, a) \cdot e(\prod_{i=1}^{k} X_{j_i}, b^r) \cdot e(\prod_{i=1}^{k} X_{j_i}^{m_{j_i}}, B) = e(g, c)$ hold.

Note that the values $(g^r Z^{m_i}, a, b, A, B, c_i)$ are information-theoretically independent of m_i if r is chosen randomly. This will become crucial when using this signature scheme to construct anonymous credentials.

THEOREM 4.1 *The selective aggregate CL-signature scheme is existentially unforgeable against adaptively chosen-message attack under CL-signature scheme.*

Proof Suppose $\mathcal{A}$ is a forger algorithm that breaks the selective aggregate CL-signature scheme. We show how to construct an algorithm $\mathcal{B}$ breaking CL signatures that are secure under the LRSW assumption. Algorithm $\mathcal{B}$ is given $(q, G, \mathsf{G}, g, \mathsf{g}, e, X, Y, Z)$, where (x, y, z) is the private key set up by the CL-signature scheme. $\mathcal{B}$ simulates the challenger and interacts with forger $\mathcal{A}$ as follows.

Setup. $\mathcal{B}$ chooses $\alpha \in_R \mathbb{Z}_q$, and set $X^* \leftarrow g^\alpha X$. Next, it initializes a key pair list *KeyList* as an empty one and starts by giving $\mathcal{A}$ the public key $(q, G, \mathsf{G}, g, \mathsf{g}, e, X^*, Y, Z)$.

Certification query. For $2 \le i \le l$, $\mathcal{A}$ adaptively requests the certification of a public key by providing a public key X_i and its private key x_i. $\mathcal{B}$ checks the private key and adds the key pair (X_i, x_i) to *KeyList*.

Signature queries. $\mathcal{A}$ adaptively requests a set of signatures by providing messages $(r, m_1, \ldots, m_l)$ to sign under the public key $(X^*, X_2, \ldots, X_l, Y, Z)$. $\mathcal{B}$ proceeds the signature query as follows: First, for m_1, $\mathcal{B}$ is given access to CL sign oracle to obtain the signature (a, b, A, B, c) on (r, m_1) under the private key (x, y, z), where $a \in G, b = a^y, A = a^z, B = A^y, c = a^{x+xyr} A^{xym_1}$. Then $\mathcal{B}$ defines $c_1 \leftarrow c \cdot (ab^r B^{m_1})^\alpha$. Observe that $e(a, Y) = e(g, b), e(a, Z) = e(g, A), e(A, y) = e(g, B)$ and $e(g, c_1) = e(X^*, a) \cdot e(X^*, b^r) e(X^*, B^{m_1})$ and therefore (a, b, A, B, c_1) is a valid signature on (r, m_1) under the public key(X^*, Y, Z). Next, for each $m_i, 2 \le i \le l$, $\mathcal{B}$ retrieves $\mathcal{A}$'s private key x_i from *KeyList* and defines $c_i \leftarrow a^{x_i} b^{x_i r} B^{x_i m_i}$. Observe that $e(g, c_i) = e(X_i, a) \cdot e(X_i, b^r) e(X_i, B^{m_i})$ and therefore (a, b, A, B, c_i) is a valid signature on (r, m_i) under the public key (X_i, Y, Z). $\mathcal{B}$ gives $(a, b, A, B, \{c_1, \ldots, c_l\})$, to $\mathcal{A}$.

Output. $\mathcal{A}$ produces the indices $(j_1, \ldots, j_k)$ and a forged aggregate signature $(a^*, b^*, A^*, B^*, c^*)$ on messages $(R, M_{j_1}, \ldots, M_{j_k})$ under public keys $(X_{j_1}, \ldots, X_{j_k}, Y, Z)$. The forged signature should not be trivial: the challenge public key X^* must be included, and the message (R, M_{j_1}) must not be queried by $\mathcal{A}$ to the CL sign oracle. Without loss of generality, we assume that $X_{j_1} = X^*$. From the verification equations, we have $e(a^*, Y) = e(b^*, g), e(a^*, Z) = e(A^*, g), e(A^*, Y) = e(B^*, g)$ and $e(\prod_{i=1}^{k} X_{j_i}, a^*) \cdot e(\prod_{i=1}^{k} X_{j_i}, (b^*)^R) e(\prod_{i=1}^{k} X_{j_i}^{M_{j_i}}, B^*) = e(g, c^*)$. $\mathcal{B}$ proceeds as follows: for each $j_i, 2 \le i \le k$, $\mathcal{B}$ retrieves the j_ith private key x_{j_i} from *KeyList*. Then $\mathcal{B}$ computes $c_{j_i}^* \leftarrow (a^*)^{x_{j_i}} (b^*)^{x_{j_i} R} (B^*)^{x_{j_i} M_{j_i}}$. Observe that $e(g, c_{j_i}^*) = e(X_{j_i}, a^*) \cdot e(X_{j_i}, (b^*)^R) e(X_{j_i}, (B^*)^{M_{j_i}})$ and therefore $(a^*, b^*, A^*, B^*, c_{j_i}^*)$ is a valid signature on (R, M_{j_i}) under the public key (X_{j_i}, Y, Z). Now $\mathcal{B}$ constructs the signature for the message (R, M_{j_1}): $c_{j_1}^* \leftarrow c^* / \prod_{i=2}^{k} c_{j_i}^*$. Then

$$
\begin{aligned}
e(g, c_{j_1}^*) &= e(g, c^*) \cdot \prod_{i=2}^{k} e(g, c_{j_i}^*)^{-1} \\
&= e\left(\prod_{i=1}^{k} X_{j_i}, a^*\right) \cdot e\left(\prod_{i=1}^{k} X_{j_i}, (b^*)^R\right) \cdot e\left(\prod_{i=1}^{k} X_{j_i}^{M_{j_i}}, B^*\right)
\end{aligned}
$$

$$\cdot \left(\prod_{i=2}^{k} e(X_{j_i}, a^*) \cdot e(X_{j_i}, (b^*)^R) e(X_{j_i}, (B^*)^{M_{j_i}}) \right)^{-1}$$

$$= e\left(\prod_{i=1}^{k} X_{j_i}, a^* \right) \cdot e\left(\prod_{i=1}^{k} X_{j_i}, (b^*)^R \right) \cdot e\left(\prod_{i=1}^{k} X_{j_i}^{M_{j_i}}, B^* \right)$$

$$\cdot e\left(\prod_{i=2}^{k} X_{j_i}, a^* \right)^{-1} \cdot e\left(\prod_{i=2}^{k} X_{j_i}, (b^*)^R \right)^{-1} \cdot e\left(\prod_{i=2}^{k} X_{j_i}^{M_{j_i}}, B^* \right)^{-1}$$

$$= e(X^*, a^*) \cdot e(X^*, (b^*)^R) e(X^*, (B^*)^{M_{j_1}}).$$

It follows that $(a^*, b^*, A^*, B^*, c^*_{j_1})$ is a valid signature on the message (R, M_{j_1}) under the challenge public key (X^*, Y, Z). B outputs $(a^*)^{x+xyR}(A^*)^{xyM_{j_1}}$ as $(a^*)^{x+xyR}(A^*)^{xyM_{j_1}} = c^*_{j_1}/(a^* \cdot (b^*)^R \cdot (B^*)^{M_{j_1}})^{\alpha}$. This means that a CL signature for a new message (R, M_{j_1}) is forged, which contradicts the LRSW assumption. ■

5. Construction of anonymous credentials

In this section, we first define the privacy requirements of anonymous credentials, then present a novel attributes encoding method based on the concept of aggregation. Next, we show the issuance protocol on how to issue anonymous credentials to users.

The anonymous credential system allows a user to obtain a credential from an issuer (also denoted as *Identity Provider*) on a number of attributes and prove the possession of a credential to a verifier (also denoted as *Relying Party*). They also enable a user to only release and prove a subset of the certified attributes while others are hidden completely.

As far as privacy issue is concerned, attributes proof needs to guarantee such requirements as follows.

- *Untraceability.* Issuers are unable to trace issued attributes and their owners. In the other word, the issuance of a credential and the showing of a credential are mutually unlinkable. It is able to prevent the insiders of relying parties from tracing the user's transactions.
- *Unlinkability.* Multiple attributes proof sessions of a single user are mutually unlinkable by the Verifiers even they collude.
- *Selective disclosure of attributes.* Users can select which portions of a credential to reveal, which portions to keep hidden, and what relations between certified items are exposed during attribute proofs. It is able to avoid the users from disclose more personal information than necessary.

5.1 *Attributes encoding*

In general cases when we talk about an attribute, it implies a tuple (*id*, *attribute type*, *attribute value*). The *id* is the identifier of the credential holder. It may be real name, pseudonym, any attribute value being an identifier, or signature. Such identifiers are different for each credential holder and can be identified by the issuer. Setting up an *id* with attributes makes credential issuance more practical, because in the physical world issuing authorities tend to identify the user before assert his attributes and issue him a credential. However, the user does not always need to reveal his *id* when showing a credential or proving attributes as far as anonymity is concerned. Therefore an attribute just implies a tuple (*attribute type*, *attribute value*).

In general, there are multiple attribute values corresponding to a single attribute type in the universal attributes field. If an attribute value is pre-defined in a finite set, it is in this case. For example, a person's gender is pre-defined in {male, female} and the particular one may have the realization of attribute (gender, male). In the policies of attribute-based authentication and access control, relying parties generally require users prove either possession of all of the multiple attributes or possession of one of the multiple attributes. For example, when submitting a resume, a person has to show a credential with the multiple attributes (gender, female), (nationality, French) and (degree, Ph.D) all together embedded, while in the other scenario, one person can enjoy the free tickets with his ID-card only if any one of the multiple attributes (minority, blind), (social_benefit, *unemployed*) or (type, kids_card) is embedded. For simplifying attributes proof, we assume there are not any two attribute labels assigned with identical values. It means we can distinguish an attribute from the value. Back to the above examples, when submitting a resume, a person has to show a credential with all of the multiple attribute values female, French, Ph.D embedded, while one person can enjoy the free tickets with any one of the multiple attribute values blind, unemployed, kids_card in his ID-card.

To solve linear complexity of attributes proof, the proposed anonymous credential only embeds a single one attribute instead of a number of attributes. Each attribute is encoded in one base, and the proof of multiple attributes is done by aggregating the corresponding individual signatures into a single one and verify it in one round. The ith anonymous credential asserts and embeds ith attribute. Users have a number of individual anonymous credentials regarding to the corresponding attribute. Each individual anonymous credential is fundamentally a CL signature formed as (a, A, b, B, c_i), which is signed on the discrete logarithm representation $g^r Z^{m_i}$ of attribute values m_i and a secret value r. Note that it is practical for all the attributes of one person to bind the same secret value, such that the aggregate CL-signature scheme works for efficient verification. Each private key x_i is designated to sign the ith attribute. Such association between the public key X_i and the attribute they represent is public for anyone. Back to the above examples again, when submitting a resume a person proves all of the required attributes, for example indexed by (i, j, k), with values $m_i = \text{female}, \text{m}_\text{j} = \text{French}, \text{m}_\text{k} = \text{Ph.D}$ and the corresponding aggregate signature $(a, b, A, B, c = c_i c_j c_k)$ satisfy the verification equation for *AggVerify*; while one person can enjoy the free tickets with any one of the required attributes $m_i = \text{blind}, \text{m}_\text{j} = \text{unemployed}, \text{m}_\text{k} = \text{kids\_card}$ and the corresponding aggregate signature $(a, b, A, B, c = c_i c_j c_k)$ satisfy the verification equation for *AggVerify*.

5.2 *Issuance protocol*

The selective aggregate CL-signature scheme can be used to obtain a signature on a committed value. It is sufficient for the signer to know $M_i = g^r Z^{m_i}$. The values (a, b, A, B) are not a function of (r, m_i), so the signer need not know (r, m_i) to generate them. Suppose that the signer chooses $\alpha \in_R \mathbb{Z}_q$, and let $a \leftarrow g^\alpha$, compute b, A, B as described by the sign algorithm. Finally, the signer computes $c_i \leftarrow a^{x_i} M_i^{\alpha x_i y}$. In order to obtain a signature on a committed value, the issuance protocol requires a recipient of the signature prove that he knows the representation of M_i in bases g and Z.

Common Input. The public key $pk = (q, G, \mathsf{G}, g, \mathsf{g}, e, Y, Z, W, \{X_i\})$ where $W = Y^z$, and commitments $M_1, \ldots, M_l$.
User's Input. Values $r, m_1, \ldots, m_l$ such that $M_i = g^r Z^{m_i}$.
Issuer's Input. Signing key $sk = (y, z, \{x_i\})$.

(1) The User gives a zero-knowledge proof of the opening of the commitments:

$$\text{PK}\{(\alpha_l, \ldots, \alpha_l, \beta) : M_1 = g^\beta Z^{\alpha_1}, \ldots, M_l = g^\beta Z^{\alpha_l}\}. \tag{6}$$

Note that for identity assurance, the commitments $M_1, \ldots, M_l$ are suggested to be shown with the corresponding certificates $\psi_1, \ldots, \psi_l$ issued by authorities. The certificates $\psi_1, \ldots, \psi_l$ are fundamentally some kind of signatures on $M_1, \ldots, M_l$. As a consequence, the zero-knowledge proof of commitments $M_1, \ldots, M_l$ and verification of certificates $\psi_1, \ldots, \psi_l$ can be aggregated all at once, as referred to as [2].

(2) The Issuer chooses a random value $\alpha \in_R \mathbb{Z}_q$, sets $a \leftarrow g^\alpha, A \leftarrow a^z, b \leftarrow a^y, B \leftarrow A^y$. Then for $1 \le i \le l$, sets $c_i \leftarrow a^{x_i} M_i^{\alpha x_i y}$ Then the Issuer outputs a signature $(a, b, A, B, \{c_i\})$ as an anonymous credential where attributes $m_1, \ldots, m_l$ are asserted.

THEOREM 5.1 *The issuance protocol is a secure two-party computation of a signature on a discrete logarithm representation of $g^r Z^m$ under the signer's public key.*

Proof From the signer's point of view, this protocol is as secure as when the user submits his signature queries in the clear. This is because of the proof of knowledge: there exists an extractor that can discover the value of the message being signed, and ask it to the signer in the clear.

From the user's point of view, since the user's secret input r is only used in the zero-knowledge proof of knowledge of it, the only thing that the signer finds out about the value r is the input value $M = g^r Z^m$. The hardness of discrete logarithm problem makes $r = \log_g(M/Z^m)$ unknown. ■

6. Attributes proof protocols

In this section, we describe a series of attributes proof protocols based on the proposed anonymous credential system. Before presenting the protocols, we define a set RI. According to the verifier's policy, the prover indicates the indices of n required attributes and the corresponding signatures in $\mathrm{RI} = \{j_1, \ldots, j_n\}$. Given l attributes certified by the Issuer, we have $\mathrm{RI} \subseteq \{1, \ldots, l\}$.

Each time before the prover shows a credential he will generate a blinded version of the originally issued signature, in order to avoid being traced by the issuer and linked by multiple relying parties. Precisely, given ith signature (a, b, A, B, c_i) as required, the Prover randomly chooses $r', r'' \in_R \mathbb{Z}_q$, then sets:

$$\tilde{a} = a^{r'}, \quad \tilde{A} = A^{r'}, \quad \tilde{b} = b^{r'}, \quad \tilde{B} = B^{r'}, \quad \tilde{c}_i = c_i^{r'}, \quad \hat{c}_i = \tilde{c}_i^{r''}, \quad 1 \le i \le l. \tag{7}$$

The blinded ith signature $\tilde{\sigma} = (\tilde{a}, \tilde{A}, \tilde{b}, \tilde{B}, \hat{c}_i)$ is distributed independently of everything else.

6.1 *AND relation proof*

The Prover is needed to prove that a subset of attributes are all embedded into the user's credential. We define RA to indicate the attribute values specified in the verifier's policy. Accordingly, the prover specifies $\mathrm{RI} = \{j_1, \ldots, j_n\}$ to indicate the indices of required attributes, and we define $\mathrm{RA} = \{a_{j_1}, \ldots, a_{j_n}\}$. Given l attributes are certified by the issuer, the AND relation proof implies to prove the possession of a combination of anonymous credentials with n attributes revealed.

Common Input. The public key $pk = (q, G, \mathsf{G}, g, \mathsf{g}, e, Y, Z, \{X_i\}, W)$, $\mathrm{RI} = \{j_1, \ldots, j_n\} \subseteq \{1, \ldots, l\}$, $\mathrm{RA} = \{a_{j_1}, \ldots, a_{j_n}\}$.
Prover's Input. The signature $\sigma = (a, A, b, B, c_{j_1}, \ldots, c_{j_n})$.
Protocol.

(1) The Prover aggregates the corresponding signatures of the required attributes, $c \leftarrow \prod_{i=1}^{n} c_{j_i}$.

(2) The Prover generates the blinded signature $\tilde{\sigma} = (\tilde{a}, \tilde{A}, \tilde{b}, \tilde{B}, \hat{c})$ as Equation (7), then sends the blinded signature $\tilde{\sigma}$ to the Verifier.
(3) The Prover carries out a zero-knowledge proof of a blinded signature $\tilde{\sigma}$ with the Verifier as follows.

$$\mathrm{PK}\left\{(\alpha,\beta) : e\left(\prod_{i=1}^{n} X_{j_i}, \tilde{a}\right) e\left(\prod_{i=1}^{n} X_{j_i}, \tilde{b}\right)^{\beta} e\left(\prod_{i=1}^{n} {X_{j_i}}^{a_{j_i}}, \tilde{B}\right) = e(g,\hat{c})^{\alpha}\right\}. \tag{8}$$

The Verifier accepts if it accepts the proof above and (a) $\tilde{A}$ were formed correctly: $e(\tilde{a}, Z) = e(g, \tilde{A})$; and (b) $\tilde{b}$ and $\tilde{B}$ were formed correctly: $e(\tilde{a}, Y) = e(g, \tilde{b}), e(\tilde{A}, Y) = e(g, \tilde{B})$.

THEOREM 6.1 *The AND relation proof protocol is a zero-knowledge proof of knowledge of a selective aggregate CL-signature.*

Proof First, we prove the zero-knowledge property. Consider the following simulator S: chooses random values r, r' and set $\tilde{a} = g^r, \tilde{b} = Y^r, \tilde{A} = Z^r, \tilde{B} = W^r$ and $\hat{c} = g^{r'}$, these values are independent of the actual signature and satisfy $e(\tilde{a}, Z) = e(g, \tilde{A}), e(\tilde{a}, Y) = e(g, \tilde{b}), e(\tilde{A}, Y) = e(g, \tilde{B})$, so step 1 is simulated correctly. Then since in step 2, the Prover and Verifier execute a zero-knowledge proof, it follows that there exists a simulator S'; just run S'. Therefore, S constructed this way is the zero-knowledge simulator for this protocol.

Next, we prove that this protocol is a proof of knowledge. We must exhibit a knowledge extractor E that, given access to a Prover such that the Verifier's acceptance probability is non-negligible, outputs $(a_{j_1}, \ldots, a_{j_n}, r, \sigma)$, such that σ is a valid signature on $(a_{j_1}, \ldots, a_{j_n}, r)$. Suppose that we are given such a prover. The extractor proceeds as follows: first, it runs the extractor for the proof of knowledge protocol of step 2. As a result, it obtains the values r, r' such that $e(\prod_{i=1}^{n} X_{j_i}, \tilde{a}) e(\prod_{i=1}^{n} X_{j_i}, \tilde{b})^r e(\prod_{i=1}^{n} {X_{j_i}}^{a_{j_i}}, \tilde{B}) = e(g, \hat{c})^{r'}$. We wish to show that $(a_{j_1}, \ldots, a_{j_n}, r)$ and $\sigma = (\tilde{a}, \tilde{A}, \tilde{b}, \tilde{B}, \hat{c}^{r'})$ satisfy the verification equation for selective aggregate CL-signature scheme. We have:

$$e\left(\prod_{i=1}^{n} X_{j_i}, \tilde{a}\right) e\left(\prod_{i=1}^{n} X_{j_i}, \tilde{b}\right)^{r} e\left(\prod_{i=1}^{n} {X_{j_i}}^{a_{j_i}}, \tilde{B}\right) = e(g,\hat{c})^{r'},$$

$$e\left(\prod_{i=1}^{n} X_{j_i}, \tilde{a}\right) \cdot e\left(\prod_{i=1}^{n} X_{j_i}, \tilde{b}^{r}\right) e\left(\prod_{i=1}^{n} X_{j_i}^{a_{j_i}}, \tilde{B}\right) = e(g,\hat{c}^{r'}).$$

■

6.2 *OR relation proof*

The Prover needs to prove that one of the subset of attributes is signed in the credential. Given an OR relation over n values $(a_{j_1} \vee \cdots \vee a_{j_n})$, the OR relation proof is to convince one out of the specified attributes $a_{j_1}, \ldots, a_{j_n}$ is embedded into the user's credential. Regarding minimal information disclosure principle, it is required that the Verifier not recognize which the particular one of the User's attributes does satisfy the verification equation. We use a three round public coin, witness indistinguishable proof of knowledge [11] to meet such requirement. The protocol requires n commitments to the relevant attributes and a linear relationship proof.

Common Input. The public key $pk = (q, G, \mathsf{G}, g, \mathsf{g}, e, Y, Z, \{X_i\}, W, h)$ where $h \in G$, $\mathrm{RI} = \{j_1, \ldots, j_n\} \subseteq \{1, \ldots, l\}$, $\mathrm{RA} = \{a_{j_1}, \ldots, a_{j_n}\}$.
Prover's Input. The signature $\sigma = (a, A, b, B, c_{j_1}, \ldots, c_{j_n})$.

Protocol.

(1) For each required attribute, the Prover chooses a random value $r_{j_i} \in_R \mathbb{Z}_q$ and computes the commitment $M_{j_i} = g^{m_{j_i}} h^{r_{j_i}}$, where $j_i \in \mathrm{RI}$. Then, the Prover aggregates the corresponding signatures of the required attributes, $c \leftarrow \prod_{i=1}^{n} c_{j_i}$. Next, the Prover generates the blinded signature $\tilde{\sigma} = (\tilde{a}, \tilde{A}, \tilde{b}, \tilde{B}, \hat{c})$ as Equation (7), then sends the blinded signature $\tilde{\sigma}$ and commitments $M_{j_1}, \ldots, M_{j_n}$ to the Verifier.
(2) The Prover carries out a zero-knowledge proof with the Verifier as follows.

$$PK\left\{(\alpha_{j_1}, \ldots, \alpha_{j_n}, \theta_{j_1}, \ldots, \theta_{j_n}, \beta, \gamma) : e\left(\prod_{i=1}^{n} X_{j_i}, \tilde{a}\right) e\left(\prod_{i=1}^{n} X_{j_i}, \tilde{b}\right)^{\beta} \times \prod_{i=1}^{n} e(X_{j_i}, \tilde{B})^{\alpha_{j_i}} = e(g, \hat{c})^{\gamma}, M_{j_i} = g^{\alpha_{j_i}} h^{\theta_{j_i}} \text{ for each } j_i \in \mathrm{RI}\right\}. \tag{9}$$

The Verifier accepts if it accepts the proof above and (a) $\tilde{A}$ were formed correctly: $e(\tilde{a}, Z) = e(g, \tilde{A})$; and (b) $\tilde{b}$ and $\tilde{B}$ were formed correctly: $e(\tilde{a}, Y) = e(g, \tilde{b}), e(\tilde{A}, Y) = e(g, \tilde{B})$.
(3) The Prover and the Verifier then facilitate proofs of knowledge over the committed attribute values, in this case a disjunction of equality proofs as follows.

$$PK\left\{(\theta_{j_1}, \ldots, \theta_{j_n}) : \frac{M_{j_1}}{g^{a_{j_1}}} = h^{\theta_{j_1}} \vee \cdots \vee \frac{M_{j_n}}{g^{a_{j_n}}} = h^{\theta_{j_n}}\right\}. \tag{10}$$

THEOREM 6.2 *The OR relation proof protocol is a zero-knowledge proof of knowledge of a selective aggregate CL-signature and a witness indistinguishable proof of partial knowledge.*

Proof First, we prove the zero-knowledge property. Consider the following simulator S: chooses random values r, r' and set $\tilde{a} = g^r, \tilde{b} = Y^r, \tilde{A} = Z^r, \tilde{B} = W^r$ and $\hat{c} = g^{r'}$, these values are independent of the actual signature and satisfy $e(\tilde{a}, Z) = e(g, \tilde{A}), e(\tilde{a}, Y) = e(g, \tilde{b}), e(\tilde{A}, Y) = e(g, \tilde{B})$; then for $1 \le i \le n$, chooses a random value $m_{j_i}, r_{j_i} \in_R \mathbb{Z}_q$, sets $M_{j_i} = g^{m_{j_i}} h^{r_{j_i}}$, and M_{j_i} is distributed correctly. So step 1 is simulated correctly. Then, since in steps 2 and 3, the Prover and Verifier execute zero-knowledge proofs, it follows that there exist two simulator S', S''; just run S', S''. Therefore, S constructed this way is the zero-knowledge simulator for this protocol.

Next, we prove that this protocol is a proof of knowledge. We must exhibit two knowledge extractors. Given access to a Prover such that the Verifier's acceptance probability is non-negligible, the first one E_1 is to output $(m_{j_1}, \ldots, m_{j_n}, r_{j_1}, \ldots, r_{j_n}, r_\beta, r_\gamma, \sigma)$ such that σ is a valid signature on $(m_{j_1}, \ldots, m_{j_n}, r_\beta)$ and (m_{j_i}, r_{j_i}) can open the commitment M_{j_i} for each $1 \le i \le n$. The second knowledge extractor E_2 is to output $(r_{j_1}, \ldots, r_{j_n})$ such that there exists a commitment M_{j_k} hiding the secret value a_{j_k} with an indistinguishable witness.

Suppose that we are given such a prover. The extractor E_1 proceeds as follows: first, it runs the extractor for the proof of knowledge protocol of step 2. As a result, it obtains the values $m_{j_1}, \ldots, m_{j_n}, r_{j_1}, \ldots, r_{j_n}, r_\beta, r_\gamma$ such that $e(\prod_{i=1}^{n} X_{j_i}, \tilde{a}) e(\prod_{i=1}^{n} X_{j_i}, \tilde{b})^{r_\beta} \prod_{i=1}^{n} e(X_{j_i}, \tilde{B})^{m_{j_i}} = e(g, \hat{c})^{r_\gamma}$ and $M_{j_i} = g^{m_{j_i}} h^{r_{j_i}}$ for each $1 \le i \le n$. We wish to show that $(m_{j_1}, \ldots, m_{j_n}, r_\beta)$ and $\sigma = (\tilde{a}, \tilde{A}, \tilde{b}, \tilde{B}, \hat{c}^{r_\gamma})$ satisfy the verification equation for the selective

aggregate CL-signature scheme. We have:

$$e\left(\prod_{i=1}^{n} X_{j_i}, \tilde{a}\right) e\left(\prod_{i=1}^{n} X_{j_i}, \tilde{b}\right)^{r_\beta} e\left(\prod_{i=1}^{n} {X_{j_i}}^{m_i}, \tilde{B}\right) = e(g, \hat{c})^{r_\gamma},$$

$$e\left(\prod_{i=1}^{n} X_{j_i}, \tilde{a}\right) e\left(\prod_{i=1}^{n} X_{j_i}, \tilde{b}^{r_\beta}\right) e\left(\prod_{i=1}^{n} X_{j_i}^{m_{j_i}}, \tilde{B}\right) = e(g, \hat{c}^{r_\gamma}).$$

The extractor E_2 proceeds as follows: first, it runs the extractor for the witness distinguishable proof of partial knowledge protocol of step 3. Assume the value a_{j_k} is committed in M_{j_k}, i.e. $M_{j_k} = g^{a_{j_k}} h^{r_{j_k}}$. It means that the Prover knows the witness r_{j_k} of $h^{r_{j_k}} = M_{j_k}/g^{a_{j_k}}$. As in [11], assume that the Prover can answer correctly a non-negligible fraction of the possible choices of the challenge. This means that by rewinding the Prover, we can efficiently get correct answers to two different challenges s and s'. Let the shares of s and s' be share(c_{j_i}) and share(c'_{j_i}), $1 \le i \le n$, respectively. Then for $j_1, \ldots, j_n$ messages, the j_kth message must have share$(c_{j_k}) \neq$ share(c'_{j_k}) since otherwise it would follow that $s = s'$. But then we also have $c_{j_k} \neq c'_{j_k}$ and can compute a witness r_{j_k} for $M_{j_k}/g^{a_{j_k}}$. ■

6.3 *Interval proof*

The interval proof expresses that the value of a given attribute lies into a given interval. For privacy protection reason, the Prover is able to prove interval predicate without revealing the value of the attribute in the clear. The Boudot-interval proofs [6] are applied to construct the protocol. It is required that the proved attribute be committed in an information-semantically secure way and retrieved from the credential.

Common Input. The public key $pk = (q, G, \mathsf{G}, g, \mathsf{g}, e, Y, Z, \{X_i\}, W, h)$ where $h \in G$, RI $= j, 1 \le j \le l$, two given values a and b, $a < b$.
Prover's Input. The signature $\sigma = (a, A, b, B, c_j)$.
Protocol.

(1) The Prover chooses a random value $r_j \in_R \mathbb{Z}_q$ and computes the commitment $M = g^{m_j} h^{r_j}$ on the jth attribute. Then it generates the blinded signature $\tilde{\sigma} = (\tilde{a}, \tilde{A}, \tilde{b}, \tilde{B}, \hat{c})$ as Equation (7). Finally it sends M and $\tilde{\sigma}$ to the Verifier.
(2) The Prover carries out a zero-knowledge proof with the Verifier as follows.

$$\mathrm{PK}\{(\alpha, \beta, \gamma, \theta) : e(X_j, \tilde{a}) e(X_j, \tilde{b})^{\beta} e(X_j, \tilde{B})^{\alpha} = e(g, \hat{c})^{\gamma}, M = g^{\alpha} h^{\theta}, \alpha \in [a, b]\}. \tag{11}$$

The Verifier accepts if it accepts the proof above and (a) $\tilde{A}$ were formed correctly: $e(\tilde{a}, Z) = e(g, \tilde{A})$; and (b) $\tilde{b}$ and $\tilde{B}$ were formed correctly: $e(\tilde{a}, Y) = e(g, \tilde{b}), e(\tilde{A}, Y) = e(g, \tilde{B})$.

THEOREM 6.3 *The Interval proof protocol is a zero-knowledge proof of knowledge of a CL signature, and two commitments hiding the same secret and a committed number belonging to an interval under Boudot-interval proof.*

Proof First, we prove the zero-knowledge property. Consider the following simulator S: chooses random values r, r' and set $\tilde{a} = g^r, \tilde{b} = Y^r, \tilde{A} = Z^r, \tilde{B} = W^r$ and $\hat{c} = g^{r'}$, these values are independent of the actual signature and satisfy $e(\tilde{a}, Z) = e(g, \tilde{A}), e(\tilde{a}, Y) = e(g, \tilde{b}), e(\tilde{A}, Y) = e(g, \tilde{B})$; then chooses a random value $r \in_R \mathbb{Z}_q$, sets $M = g^r$, and M is distributed correctly, so step 1 is simulated correctly. Then, since in step 2, the Prover and Verifier execute a zero-knowledge proof,

it follows that there exists a simulator S'; just run S'. Therefore, S constructed this way is the zero-knowledge simulator for this protocol.

Next, we prove that this protocol is a proof of knowledge. We must exhibit a knowledge extractor E that, given access to a Prover such that the Verifier's acceptance probability is non-negligible, outputs (m, r, r_β, σ), such that σ is a valid signature on (m, r_β), (m, r) can open the commitment M and m lies in $[a, b]$. Suppose that we are given such a prover. The extractor proceeds as follows: first, it runs the extractor for the proof of knowledge protocol of step 2. As a result, it obtains the values m, r, r_β, r_γ such that $M = g^m h^r, m \in [a, b]$ and $e(X_j, \tilde{a})e(X_j, \tilde{b})^{r_\beta} e(X_j, \tilde{B})^m = e(g, \hat{c})^{r_\gamma}$. We wish to show that (m, r_β) and $\sigma = (\tilde{a}, \tilde{A}, \tilde{b}, \tilde{B}, \hat{c}^{r_\gamma})$ satisfy the verification equation for CL-signature scheme. We have:

$$e(X_j, \tilde{a})e(X_j, \tilde{b})^{r_\beta} e(X_j, \tilde{B})^m = e(g, \hat{c})^{r_\gamma},$$
$$e(X_j, \tilde{a})e(X_j, \tilde{b})^{r_\beta} e(X_j, \tilde{B})^m = e(g, \hat{c}^{r_\gamma}).$$

■

6.4 *Inequality proof*

The statements express that the required attribute value is not equal to a given value. For privacy protection reason, the Prover is able to prove the inequality predicate without revealing the value of the attribute in the clear.

Common Input. The public key $pk = (q, G, \mathsf{G}, g, \mathsf{g}, e, Y, Z, \{X_i\}, W, h)$ where $h \in G$, $RI = j$, $1 \leq j \leq l$, a given value a.
Prover's Input. $\sigma = (a, A, b, B, c_j)$.
Protocol.

(1) The Prover chooses a random value $r_j \in_R \mathbb{Z}_q$, computes the commitment $M = g^{m_j} h^{r_j}$ on the jth attribute. Then it generates the blinded signature $\tilde{\sigma} = (\tilde{a}, \tilde{A}, \tilde{b}, \tilde{B}, \hat{c})$ as Equation (7). Finally it sends M and $\tilde{\sigma}$ to the Verifier.
(2) The Prover carries out a zero-knowledge proof with the Verifier as follows.

$$\mathrm{PK}\left\{(\alpha, \beta, \gamma, \theta, \pi, \rho) : e(X_j, \tilde{a})e(X_j, \tilde{b})^{\beta} e(X_j, \tilde{B})^{\alpha} = e(g, \hat{c})^{\gamma},\ M = g^{\alpha} h^{\theta}, g = \left(\frac{M}{g^a}\right)^{\pi} h^{\rho}\right\}. \tag{12}$$

The Verifier accepts if it accepts the proof above and (a) $\tilde{A}$ were formed correctly: $e(\tilde{a}, Z) = e(g, \tilde{A})$; and (b) $\tilde{b}$ and $\tilde{B}$ were formed correctly: $e(\tilde{a}, Y) = e(g, \tilde{b}), e(\tilde{A}, Y) = e(g, \tilde{B})$.

THEOREM 6.4 *The Inequality proof protocol is a zero-knowledge proof of knowledge of a CL signature and two commitments hiding the same secret.*

Proof First, we prove the zero-knowledge property. Consider the following simulator S: chooses random values r, r' and set $\tilde{a} = g^r, \tilde{b} = Y^r, \tilde{A} = Z^r, \tilde{B} = W^r$ and $\hat{c} = g^{r'}$, these values are independent of the actual signature and satisfy $e(\tilde{a}, Z) = e(g, \tilde{A}), e(\tilde{a}, Y) = e(g, \tilde{b}), e(\tilde{A}, Y) = e(g, \tilde{B})$; then chooses a random value $r \in_R \mathbb{Z}_q$, sets $M = g^r$, and M is distributed correctly, so step 1 is simulated correctly. Then, since in step 2, the Prover and Verifier execute a zero-knowledge proof, it follows that there exists a simulator S'; just run S'. Therefore, S constructed this way is the zero-knowledge simulator for this protocol.

Next, we prove that this protocol is a proof of knowledge. We must exhibit a knowledge extractor E that, given access to a Prover such that the Verifier's acceptance probability is non-negligible, outputs $(m, r, r_\beta, r_\gamma, r_\pi, r_\rho, \sigma)$, such that σ is a valid signature on (m, r_β), (m, r) can

open the commitment M and $g = (M/g^a)^{r_\pi} h^{r_\rho}$. Suppose that we are given such a prover. The extractor proceeds as follows: first, it runs the extractor for the proof of knowledge protocol of step 2. As a result, it obtains the values $m, r, r_\beta, r_\gamma, r_\pi, r_\rho$ such that $M = g^m h^r, g = (M/g^a)^{r_\pi} h^{r_\rho}$ and $e(X_j, \tilde{a})e(X_j, \tilde{b})^{r_\beta} e(X_j, \tilde{B})^m = e(g, \hat{c})^{r_\gamma}$. We wish to show that (m, r_β) and $\sigma = (\tilde{a}, \tilde{A}, \tilde{b}, \tilde{B}, \hat{c}^{r_\gamma})$ satisfy the verification equation for CL-signature scheme. We have:

$$e(X_j, \tilde{a})e(X_j, \tilde{b})^{r_\beta} e(X_j, \tilde{B})^m = e(g, \hat{c})^{r_\gamma},$$
$$e(X_j, \tilde{a})e(X_j, \tilde{b})^{r_\beta} e(X_j, \tilde{B})^m = e(g, \hat{c}^{r_\gamma}).$$

■

7. Efficiency

In this section, we analyse the efficiency of AND relation proof, OR relation proof, interval proof and inequality proof w.r.t. the number of exponentiations and pairings. N is the number of the attributes referenced in a proof.

Prior to attributes proof, the prover pre-computes the pairings as follows: $V_a \leftarrow e(\prod_{i=1}^N X_{j_i}, a), V_b \leftarrow e(\prod_{i=1}^N X_{j_i}, b) = V_a^y, V_A \leftarrow e(\prod_{i=1}^N X_{j_i}, A) = V_a^z, V_B \leftarrow e(\prod_{i=1}^N X_{j_i}, B) = V_A^y, V_c = e(g, c)$. Thus, we can omit most pairings with adding some slight exponentiations.

The computation cost of attributes proof is the sum of three parts which are *signature randomization*, *proof generation* and *verification*. The first two parts are related to the Prover, while the last one is related to the Verifier. Each time before the prover shows a credential, there are five exponentiations for signature randomization as Equation (7). The additional number of exponentiations and pairings in attributes proof are given respectively as follows.

- The AND relation proof protocol has constant complexity with the number of required attributes. Concisely, it takes zero pairing and 2 exponentiations for proof generation; while 10 pairings and 3 exponentiations for verification.
- The OR proof protocol has linear complexity with the number of required attributes. Concisely, it takes zero pairing and $2 + 6N$ exponentiations to generate the relevant commitments and the proof; while 10 pairings and $3 + 6N$ exponentiations for verification.
- The interval proof protocol has constant complexity with the number of required attributes. Concisely, it takes zero pairing and 7 exponentiations to generate the relevant commitment and the proof; while 10 pairings and 7 exponentiations for verification.
- The inequality proof protocol has constant complexity with the number of required attributes. Concisely, it takes zero pairing and 9 exponentiations to generate the relevant commitment and the proof; while 10 pairings and 10 exponentiations for verification.

8. Conclusion

The proposed aggregate-based attributes proof protocols are novel to solve linear complexity of attributes proof. Distinct with the existing attribute encoding method, each single attribute is certified in an individual credential. Later on users can select a subset of individual signatures as required and prove the combination of attributes all at once by aggregating the corresponding individual signatures. Based on CL-signature scheme, we present a selective aggregate CL-signature scheme and use it as the building block to construct anonymous credentials. Then we construct AND relation proof protocol, OR relation proof protocol, interval proof protocol and inequality proof protocol respectively. Fundamentally these protocols are primarily zero-knowledge proof of a blinded aggregate signature on the required attributes. The efficiency analysis shows that the

resulting protocols, except for OR relation proof, have constant complexity w.r.t. the number of pairings and exponentiations. OR relation proof has linear complexity only w.r.t. the number of attributes as required and the concrete number of pairings and exponentiations are smaller.

Acknowledgements

This work was supported by the Fundamental Research Funds for the Central Universities [Grant number N100404004 and N120404010], Major National Scientific and Technolog- ical Projects [Grant number 2013ZX03002006], and China Natural Science Foundation [Grant number 61300196].

References

[1] N. Begum, T. Nakanishi, and N. Funabiki, *Efficient proofs for CNF formulas on attributes in pairing-based anonymous credential system*, in *Information Security and Cryptology C ICISC 2012*, T. Kwon, M.-K. Lee, and D. Kwon, eds., Vol. LNCS 7839, Springer, Berlin, Heidelberg, 2012, pp. 495–509.

[2] A. Bhargav-Spantzel, A.C. Squicciarini, R. Xue, and E. Bertino, *Multifactor identity verification using aggregated proof of knowledge*, IEEE Trans. Syst. Man Cybern. C, Appl. Rev. 40 (2010), pp. 372–383.

[3] P. Bichsel, J. Camenisch, and F.S. Preiss, *A comprehensive framework enabling data-minimizing authentication*, Proc. of the 7th ACM Workshop on Digital Identity Management(DIM'11), Berlin, Germany, July, ACM, 2011, pp. 13–22.

[4] R. Bjones, I. Krontiris, P. Paillier, and K. Rannenberg, *Integrating anonymous credentials with aids for privacy-respecting online authentication*, in *Privacy Technologies and Policy*, B. Preneel and D. Ikonomou, eds., Springer, Berlin, Heidelberg, 2014, pp. 111–124.

[5] B.D. Boneh, C. Gentry, and H. Shacham, *Aggregate and verifiably encrypted signatures from bilinear maps*, in *Advances in Cryptology*, E. Biham, ed., Vol. LNCS 2656, Springer, Berlin, Heidelberg, 2003, pp. 416–432.

[6] F. Boudot, *Efficient proofs that a committed number lies in an interval*, in *Proc. of the International Conference on the Theory and Application of Cryptographic Techniques Advances in Cryptology(EUROCRYPT'00)*, Bruges, Belgium, B. Preneel, ed., Vol. LNCS 1807, May, Springer, Berlin, Heidelberg, 2000, pp. 431–444.

[7] J. Camenisch and T. Groß, *Efficient attributes for anonymous credentials*, Proc. ACM Conference on Computer and Communications Security 2008, Alexandria, Virginia, October, ACM, 2008, pp. 345–356.

[8] J. Camenisch and T. Groß, *Efficient attributes for anonymous credentials*, ACM Trans. Inf. Syst. Secur., (TISSEC) – Special Issue on Computer and Communications Security, Vol. 15, March, ACM, 2012, pp. 4:1–4:30.

[9] J. Camenisch and A. Lysyanskaya, *Signature schemes and anonymous credentials from bilinear maps*, in *Proc. of the 24th Annual International Cryptology Conference Advances in Cryptology (CRYPTO'04)*, Santa Barbara, California, M. Franklin, ed., Vol. LNCS 3152, August, Springer, Berlin, Heidelberg, 2004, pp. 56–72.

[10] J. Camenisch, M. Kohlweiss, and C. Soriente, *An accumulator based on bilinear maps and efficient revocation for anonymous credentials*, in *Irvine Proceedings of the 12th International Conference on Practice and Theory in Public Key Cryptography: PKC '09*, S. Jarecki and G. Tsudik, eds., Vol. LNCS 5443, Springer, Berlin, Heidelberg, 2009, pp. 481–500.

[11] R. Cramer, I. Damgård, and B. Schoenmakers, *Proofs of partial knowledge and simplified design of witness hiding protocols*, in *Advances in Cryptology*, Y.G. Desmedt, ed., Vol. LNCS 839, May, Springer, Berlin, Heidelberg, 1994, pp. 174–187.

[12] N. Guo, J. Wang, T. Gao, and K. Yim, *Privacy-preserving predicate proof of attributes with CL-anonymous credential*, J. Internet Serv. Inf. Secur. 4 (2014), pp. 37–46.

[13] J. Herranz, F. Laguillaumie, B. Libert, and C. Rafols, *Short attribute-based signatures for threshold predicates*, in *Proc. of the The Cryptographers' Track at the RSA Conference 2012 Topics in Cryptology (CT-RSA'12)*, San Francisco, CA, USA, O. Dunkelman, ed., Vol. LNCS 7178, February–March, Springer, Berlin, Heidelberg, 2012, pp. 51–67.

[14] M. Izabachène, B. Libert, and D. Vergnaud, *Block-wise P-signatures and non-interactive anonymous credentials with efficient attributes*, in *Cryptography and Coding*, L. Chen, ed., Vol. LNCS 7089, Springer, Berlin, Heidelberg, 2011, pp. 431–450.

[15] K. Lee, D.H. Lee, and M. Yung, *Aggregating CL-signatures revisited: Extended functionality and better efficiency*, in *Financial Cryptography and Data Security*, A.-R. Sadeghi, ed., Vol. LNCS 7859, February, Springer, Berlin, Heidelberg, 2013, pp. 171–188.

[16] P. Legg, N. Moffat, J.R. Nurse, J. Happa, I. Agrafiotis, M. Goldsmith, and S. Creese, *Towards a conceptual model and reasoning structure for insider threat detection*, JoWUA 4 (2013), pp. 20–37.

[17] J. Li and N. Li, *OACerts: Oblivious attribute certificates*, in *Proc. of the 3th International Conference Applied Cryptography and Network Security (ACNS'05)*, New York, NY, J. Ioannidis and A. Keromytis, eds., Vol. LNCS 3531, June, Springer, Berlin, Heidelberg, 2005, pp. 301–317.

[18] H.K. Maji, M. Prabhakaran, and M. Rosulek, *Attribute-based signatures*, in *Proc. of the Cryptographers'Track at the RSA Conference 2011 Topics in Cryptology (CT-RSA'11)*, San Francisco, CA, A. Kiayias, ed., Vol. LNCS 6558, February, Springer, Berlin, Heidelberg, 2011, pp. 376–392.

[19] D. Slamanig, C. Stingl, C. Menard, M. Heiligenbrunner, and J. Thierry, *Anonymity and application privacy in context of mobile computing in ehealth*, in *Mobile Response*, J. Löffler and M. Klann, eds., Springer, Berlin, Heidelberg, 2009, pp. 148–157.
[20] J. Su, D. Cao, B. Zhao, X. Wang, and I. You, *epass: An expressive attribute-based signature scheme with privacy and an unforgeability guarantee for the internet of things*, Future Gener. Comput. Syst. 33 (2014), pp. 11–18.
[21] A. Sudarsono, T. Nakanishi, and N. Funabiki, *Efficient proofs of attributes in pairing-based anonymous credential system*, in *Proc. of the 11th International Symposium Privacy Enhancing Technologies (PETS'11)*, Waterloo, ON, Canada, D. Ślęzak, ed., Vol. LNCS 6794, July, Springer, Berlin, Heidelberg, 2011, pp. 246–263.
[22] S.M. Takashi Nishide and K. Sakurai, *Security analysis of offline e-cash systems with malicious insider*, JoWUA 3 (2012), pp. 55–71.