A Study of Efficient Proofs for Conjunctive Normal Forms on Attributes in Anonymous Authentication Systems

September, 2014

Nasima Begum

Graduate School of
Natural Science and Technology

(Doctor's Course)
Okayama University

Dissertation submitted to
Graduate School of Natural Science and Technology
of
Okayama University
for
partial fulfillment of the requirements
for the degree of
Doctor of Philosophy.

Written under the supervision of

Professor Nobuo Funabiki

and co-supervised by
Professor Toru Nakanishi
Professor Satoshi Denno
and
Professor Yasuyuki Nogami

OKAYAMA UNIVERSITY, September 2014.

To Whom It May Concern

We hereby certify that this is a typical copy of the original doctor thesis of
Mrs. Nasima Begum

Signature of

the Supervisor

Prof. Nobuo Funabiki

Seal of

Graduate School of

Natural Science and Technology

Abstract

Recently, attribute-based user authentications using user's attributes such as the gender, the profession and the date of birth have been used to ensure the secure access from the valid user to network services. However, in these ID-based systems, Service Providers (SPs) can identify the user, record the user's behaviors, and extract the user's profile. As a result, the conventional systems may cause a serious privacy problem. As a cryptographic solution to this privacy problem, an anonymous credential system has been intensively researched. This system allows an issuer to issue a certificate to a user containing the user's attributes. Then, based on this certificate, the user can anonymously convince a verifier of the possession of such a certificate, where only the selected attributes can be disclosed without revealing any other information about the user's privacy.

Previously, an anonymous credential system with constant-size proofs was proposed. This system supports the proofs of the inner product relations on attributes to handle the complex logical relations on attributes as the CNF (Conjunctive Normal Form) and DNF (Disjunctive Normal Form) formulas. However, this system suffers from the computational cost: the proof generation needs exponentiations whose number depends on the number of the literals in OR relations.

In this thesis, firstly, we propose a pairing-based anonymous credential system with the constant-size proofs for CNF formulas and the efficient proof generation. In the system, the proof generation needs only multiplications whose number depends on the number of literals, and thus it is more efficient than the previous system. The key idea of our construction is to use a pairing-based extended accumulator, by which we can verify that multiple attributes are included in multiple sets, all at once. This leads to the verification of CNF formulas on attributes. Since the accumulator is mainly obtained through by multiplications, we achieve the better computational cost. To show the practicality of the proposed system, we implemented it using the fast pairing library. The experimental result shows that the proof generation time and the verification time are less than one second even for $100,000$ literals in OR relations. This indicates that our system is sufficiently practical.

Secondly, we propose an extension of the anonymous credential system with the constant-size proofs for CNF formulas to reduce the public key size. The key idea behind the extension is to separate the set of candidates into two sets. In the basic system, to ensure the correctness of a value u in the verification, the issuer publishes signatures on all candidates of the value. In our extension, we consider two values for the value u, u_1 and u_2 such that $u = u_1 + u_2$, and signatures on u_1 and u_2 are separately published. In the attribute proof protocol, the user proves the knowledge of the signatures on both u_1 and u_2, for the verification of the accumulator. The experimental result shows that the public key size is reduced to $2\sqrt{N}$ for the original size N, although the computational costs are increased by about 20%. We consider it as a trade-off to reduce the public key size.

Thirdly, we propose an efficiency improvement of the computational overhead based on online/offline precomputation technique to reduce the online computational costs of the proof generation in case of lots of AND relations in the proved CNF formulas. All exponentiations that can be used for the accumulator and witness computations are executed in advance in the precomputation algorithm. Thus, exponentiations in the online accumulator and witness computations are excluded, and only multiplications are needed. The experimental result shows that the computational costs of the proof generation in the case of using lots of AND relations are greatly reduced than our basic system. Hence, it is practical for mobile users.

One future work is to propose a system allowing proofs beyond CNF formulas. Although our proposed system focuses only on the CNF formulas, in some real applications, we may need some other logical relations beyond CNF formulas, such as monotone relations or even negations. Another future work is the implementation of our system on smart phones such as Android devices.

Acknowledgements

It is my great pleasure to thank those who made this thesis possible.

I owe my deepest gratitude to my supervisor, Prof. Nobuo Funabiki, who has supported me throughout my thesis with his patience and knowledge. I am greatly indebted to him, whose encouragement, advice, and support from the beginning enabled me to develop an understanding of the subject, not only in scientific but also in life. He gave me a wonderful advice, comments, and guidance when writing papers and presenting them.

I am greatly indebted to my co-supervisor, Prof. Toru Nakanishi, who patiently gave me advices, wonderful guidance, insightful comments, helpful discussions and suggestions, a lots of inputs and directions during the completion of my research which inspired many ideas in this thesis. He was always ready to hear my problems, discuss them, and give me wonderful suggestions, directions, and solutions without feeling disturbed. Prof. Nakanishi taught me how to face the problems, how to investigate them, and how to solve them. He assisted me in improving my knowledge and inspiring new ideas, and taught me a lot of scientific knowledge especially in cryptography and it's applications. He supplied me the fundamentals and many ideas of this thesis, and guided me how to write quality papers and present them in an excellent way. Moreover, he did not only supply me the scientific knowledge, but also enlightened me with the important knowledge in life.

I am heartily thankful to my co-supervisor, Prof. Satoshi Denno for his support, guidance, and proofreading of this research work.

I would like to show my gratitude to my another co-supervisor professor Yasuyuki Nogami for taking the time to deliver lectures on elliptic curve cryptography, supply me the tutorial about the pairing techniques, give me advices, suggestions & comments in doing this research and proofreading of this research work.

I am heartily thankful to Prof. Tokumi Yokohira for his lectures, continuous support, and guidance for doing this research work.

I would also like to show my gratitude to the course teachers during my Doctor's Course study for enlightening me with wonderful knowledge.

I would like to thank for the helpful discussions from many people, a non-exhaustive list includes: Daiki Ikeda, Tomoya Nomura, Toru Mishima, Yuji Hamada, Nakada Hiroyuki, Kyosuke Horichi and Sahidatul Sadiah. I would like to thank to all FUNABIKI's Lab members for their continual supports during the period of this study. They are my wonderful friends.

I am eternally grateful to my husband Md. Ezharul Islam, my son Nahian, my brothers and sisters, my beloved parents and my parents in law. They always encourage me and support the spirit either in an atmosphere of happiness or sadness. I dedicate this thesis to them.

List of Publications

The following papers have been published and/or presented. The content of this thesis is based on the papers.

Journal Paper

1. **Nasima Begum**, Toru Nakanishi, and Nobuo Funabiki, "Efficient proofs for CNF formulas on attributes in pairing-based anonymous credential system," IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences, vol. E96-A, No. 12, pp. 2422-2433, December 2013.

International Conference Papers

2. **Nasima Begum**, Toru Nakanishi, and Nobuo Funabiki, "Reducing public-key size in an anonymous credential system for CNF formulas with constant-size proofs," Proc. 2nd Global Conference on Consumer Electronics (IEEE GCCE 2013), pp. 530-533. IEEE, 1-4 October, 2013 (Tokyo, Japan).
3. **Nasima Begum**, Toru Nakanishi, and Nobuo Funabiki, "Implementation and evaluation of an pairing-based anonymous credential system with constant-size proofs and efficient proof generations," Proc. 3rd International Workshop on Advances in Networking and Computing (WANC2012), pp. 264-268, 5-7 December, 2012 (Okinawa, Japan).
4. **Nasima Begum**, Toru Nakanishi, and Nobuo Funabiki, "Efficient proofs for CNF formulas on attributes in pairing-based anonymous credential system," Proc. 15th Annual International Conference on Information Security and Cryptology(ICISC 2012), LNCS 7839, Springer-Verlag, pp. 495-509, November 2012 (Seoul, Korea).

Other Paper

5. **Nasima Begum**, Toru Nakanishi, and Nobuo Funabiki, "Efficient proofs for CNF formulas on attributes in pairing-based anonymous credential system," IEICE Technical Report, vol. 112, no. 126, ISEC2012-38, pp. 207212, 19-20 July, 2012 (Hokkaido, Japan).

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Backgrounds

In recent years, due to the wide-spread use of Internet and wireless networks, Web services have become very popular all over the world. Users can access various kinds of Web services even through small mobile devices from any place at any time. It is possible to use many cloud applications such as Gmail, skype, twitter, YouTube, google Voice from a smartphone. In such services, to ensure the secure access only from the valid user, a user authentication is greatly required.

Currently, for the authentication, an electronic identity (eID) such as an eID card has been often used. The eID is issued by a trusted organization such as a government, a company or a university. The eID includes attributes of the user such as the gender, the occupation and the date of birth. Indeed, in many commercial applications, an attribute-based authentication is highly desired instead of the user authentication. For instance, when distributing violent video contents, the service provider needs to deny any access from the underage users by checking the age attribute in the eID, as illustrated in Fig. 1.1. In the current eHealth networks [1], patients are assigned to multiple attributes which directly reflect their symptoms, undergoing treatments, etc. Those life-threatened attributes must be verified by authorized medical facilities, such as hospitals and clinics. When needing medical services, patients have to be authenticated by showing their corresponding attributes in order to take appropriate healthcare actions.

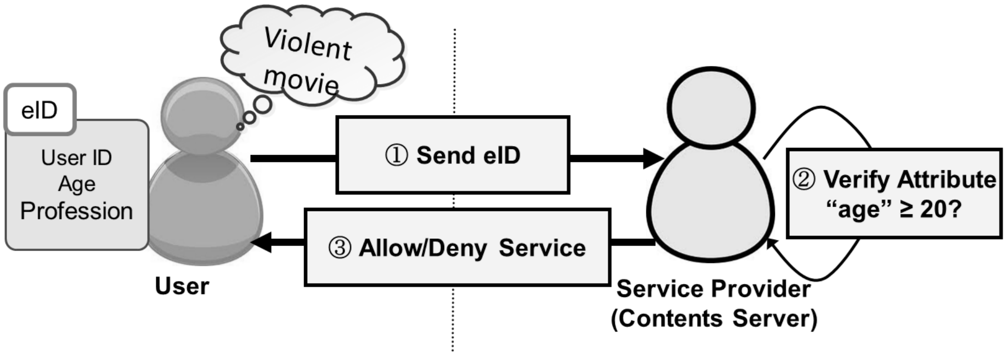

Figure 1.1: An example of eID applications.

One of the serious issues of existing attribute-based authentication systems is the user's privacy. The eID may reveal the user's identity and furthermore it reveals the full set of user's attributes during the authentication. In conventional ID-based authentications, a user is registered in the server, where the attributes are stored in the database. During accessing the service, a user is authenticated with his/her ID and password by the service provider (SP). Then, the SP verifies the attributes of the corresponding ID from the database. However, in this case, the SP can trace the user's service history by linking the ID and extract the user's profile using his/her attributes. These privacy information may be managed in a misbehavior way. Thus, the conventional authentication systems may cause serious privacy issues. Therefore, an attribute-based authentication with the strong privacy protection is in demand, where users can disclose only the minimal amount of the personal information necessary for the service instead of his/her ID.

To address this privacy issue, anonymous credential systems [2, 3, 4, 5, 6] have been intensively researched. In these systems, the user can anonymously convince a verifier about the possession of the specified attributes included in the certificate, as shown in Fig. 1.2. Each certificate is a proof of membership, qualification, or privilege, and contains users' attributes. There are three entities in an anonymous credential system: an issuer, a user, and an SP. A user is issued a certificate including the attributes certified by a trusted issuer. To get the access to the service, the user can anonymously prove a relation on the certified attributes to the SP without revealing his/her original identity.

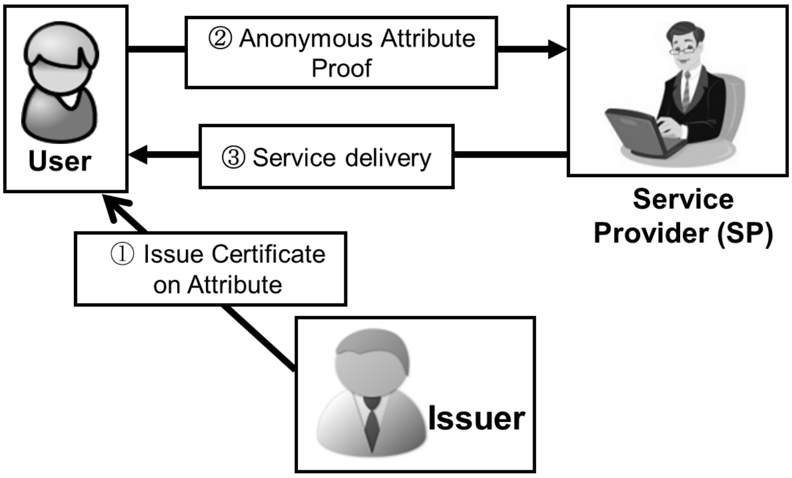

Figure 1.2: Model of anonymous credential system.

In the proof protocol of the anonymous credential system, only the selected attributes can be disclosed without revealing any other information about the users privacy. Proofs of more complex relation on attributes are also available. The AND relation is used when proving the possession of all of the multiple attributes. For example, the user can prove that he/she belongs to the department and is a professor, when entering the room of examination papers. The OR relation represents the proof for possession of one of multiple attributes. For example, he/she can prove that he/she is a technical staff, an assistant, or a professor, when using a copy machine in a laboratory. An implementation on a standard Java card is shown in [7].

In [3], Camenisch and Groß proposed an RSA-based anonymous credential system with proofs for the AND and OR relation, where the proofs have constant size with respect to the number of attributes. In [5], the pairing-based system with the constant-size proofs was proposed to achieve the short data size by excluding the the RSA-related assumptions. The pairing [8, 9, 10] is a bilinear map constructed in Elliptic Curve Cryptography (ECC), and a recent key technique to achieve highly functional cryptosystem such as anonymous credential systems. However, both systems have a drawback: They allow us to prove only simple AND or OR relations on attributes. Namely, we cannot prove any combination of AND and OR relations simultaneously.

In [6], a pairing-based system with the constant-size proofs was proposed, where inner-product relations on attributes can be proved. This means that we can handle CNF (Conjunctive Normal Form) or DNF (Disjunctive Normal Form) formulas on attributes via the polynomial-based encoding shown in [11]. However, this system has a drawback of the computational cost: The proof generation needs exponentiations whose number depends on the number of the literals in OR relations. In usual cases that the formulas include OR relations for lots of literals, the user devices with the limited computational power such as electronic ID cards need long time for processing the authentication.

1.2 Contributions

In this dissertation, we propose a pairing-based anonymous credential system with constant size of proofs, such that the combinations of AND and OR relations on attributes can be proved as CNF (Conjunctive Normal Form) formulas [12, 13, 14]. In our system, the proof generation cost is more efficient than the system in [6], since only multiplications whose number depends on the number of literals are needed and the multiplication's cost is much more less than the exponentiation's one. The key idea of this proposal is the use of a pairing-based accumulator in [4, 5, 15], which outputs a constant-size value from a large set of input values. We consider that the input values are assigned to attributes and utilize a zero-knowledge-proof friendly signature scheme [16] to certify a set of attributes as the accumulator. As the underlying anonymous credential system, our system is derived from the group signature scheme [17], and utilizes zero-knowledge proof technique [18] to anonymously prove the ownership of the certificate to the SP. We extend the efficient accumulators in [4, 5] to handle the proof of the CNF formula for the construction. In our extended accumulator, we can verify that multiple attributes are included in multiple sets, all at once. This leads to the verification of CNF formulas on attributes, where a CNF formula consists of AND relations of OR relations on users attributes. Let V_ℓ be the set of attributes in the ℓ-th OR clause in the proved CNF formula, and let U be the set of user's certified attributes. To prove the CNF formula using accumulator, an attribute value from U must be included in each V_ℓ, i.e., $U \cap V_\ell \neq \emptyset$, all at once. To hold the CNF formula, there must be at least one common attribute between the user attribute set U and the CNF clauses V_ℓ. One demerit of our system is the increase of public parameters. This increase happens when the maximum number of matched attribute (i.e., $|V_\ell \cap U|$) is large for multiple ℓ.

Then, to confirm the practicality of our proposed system, we implemented our scheme using the fast pairing and ECC (Elliptic Curve Cryptography) library [19, 20]. In our implemented system, we measured the computational processing times. The experimental results show that the proof generation time and the verification time depend on the size of

CNF formula. When we consider the size of CNF formula that is the maximum number of OR clauses upto 50, the proof generation and the verification time are at most 215 ms and 339 ms, respectively. Even for large number of attributes in an OR clause ($|V_\ell| = 100,000$), the proof generation time is only 228 ms and the verification time is only 371 ms. These indicates that our system is sufficiently practical.

The compensation of our scheme is the increase of public parameters, which brings a large communication cost to the system. Hence, to overcome this overhead, we propose an extension to reduce the public key size. In the previous system, to ensure the correctness of a value u in the verification, the issuer publishes signatures on all candidates of u. In this extension, we consider two values u_1 and u_2 such that $u = u_1 + u_2$, and signatures on u_1 and u_2 are separately published. This modification reduces the public key size to $2\sqrt{N}$ for the original size N. However, this trick increases the computational costs by about 20% compared to our previous system, which we consider as a trade off in order to reduce the key size. In our implementation, the proving time and the verification time are less than 200 ms and 500 ms respectively in a usual PC, which is still practical.

Finally, we propose an efficiency improvement based on online/offline precomputation technique to reduce the online computational costs of the proof generation in case of lots of AND relations in the proved CNF formulas. In the precomputation, all exponentiations that can be used for the accumulator and witness computations are executed in advance. Thus, exponentiations in the online accumulator and witness computations are excluded, and only multiplications are needed. The experimental result shows that the computational costs of the online proof generation in the user side are greatly reduced than our previous system, and hence it is practical for mobile users. One demerit of this proposal is the storage cost for lots of precomputed values. But the current small mobile devices have sufficiently large storage.

1.3 Contents of This Dissertation

The remaining of this dissertation is organized as follows.

In Chapter 2, we begin with preliminaries where all the building blocks of our anonymous credential scheme are defined. This chapter reviews the mathematical fundamentals for this dissertation which covers the introduction of the mathematical setting such as the groups, bilinear maps and the basic of pairings. Then the complexity assumptions, the structure-preserving signature and Groth-Sahai proofs that are used in this dissertation are illustrated.

In Chapter 3, we extend the accumulator to fit the CNF formulas, describe the construction idea and propose our scheme.

In Chapter 4, to show the practicality and effectiveness of our proposed scheme, we explain the implementation and experiments.

In Chapter 5, we describe the construction idea of the extended algorithm to reduce the public parameters, and explain the implementation and experiments.

In Chapter 6, we propose an improvement based on online-offline technique.

Finally, Chapter 7 concludes this dissertation with some future works.

Chapter 2

Preliminaries

2.1 Setting of Mathematics

2.1.1 Groups

In this research, the implemented anonymous credential system is mainly constructed based on the bilinear groups and bilinear map. Our scheme utilizes the following bilinear groups:

1. $\mathcal{G}_1$ and $\mathcal{G}_2$ are multiplicative cyclic groups of prime order p,

2. g_1 and g_2 are randomly chosen generators of $\mathcal{G}_1$ and $\mathcal{G}_2$, respectively.

2.1.2 Bilinear Maps

Throughout this dissertation, we also employ bilinear maps and use the following notations:

1. $\mathcal{G}_T$ is a multiplicative cyclic group of order p.

2. e is an efficiently computable bilinear map $e : \mathcal{G}_1 \times \mathcal{G}_2 \to \mathcal{G}_T$ with the following properties:

   (a) **Bilinearity:** for all $u, u' \in \mathcal{G}_1$ and $v, v' \in \mathcal{G}_2$, $e(uu', v) = e(u, v)e(u', v)$ and $e(u, vv') = e(u, v)e(u, v')$, and thus for all $u, \in \mathcal{G}_1$, $v, \in \mathcal{G}_2$ and $a, b \in \mathcal{Z}$, $e(u^a, v^b) = e(u, v)^{ab}$

   (b) **Non-degeneracy:** $e(g_1, g_2) \neq 1_{\mathcal{T}}$, where $1_{\mathcal{T}}$ is the identity element of group $\mathcal{G}_{\mathcal{T}}$.

2.2 Pairing-Based Cryptography

The underlying secure technique of the pairing-based cryptosystem is Elliptic Curve Cryptography (ECC), which is based on Discrete Logarithm Problem known as ECDLP (Elliptic Curve Discrete Logarithm Problem). In this research, our main concentration is anonymous credential system by utilizing the group signature, ECC, and pairing.

The pairing-based cryptography [8, 9, 10] is a technique to construct a cryptosystem by mapping between elements of two cryptographic groups of elliptic curve rational points into another group of vectors in extension field. Usually, the two groups are in the same group, and they may come from different groups. Here, pairing means a mapping from two rational

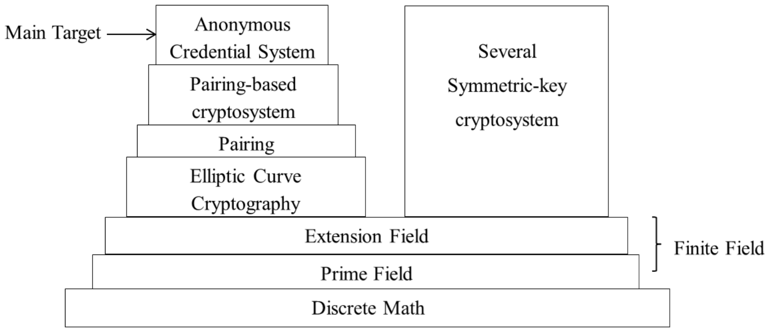
Main Target
Anonymous Credential System
Pairing-based cryptosystem
Pairing
Elliptic Curve Cryptography
Several Symmetric-key cryptosystem
Extension Field
Prime Field
Finite Field
Discrete Math

Figure 2.1: The research layers of cryptographies.

points in the same or different group to a vector in another group. In this dissertation, we describe only the pairing notation and its classes related to our work.

2.2.1 Notation and Fundamental

Fundamentally, the underlying of pairing-based cryptography is an elliptic curve defined over finite field $\mathbb{F}_p$ where is generally defined by:

$$E/\mathcal{F}_p : y^2 = x^3 + ax + b \wedge a, b \in \mathcal{F}_p. \tag{2.1}$$

$\mathcal{F}_p$ and $E/\mathcal{F}_p$ denote a prime finite field and an elliptic curve over $\mathcal{F}_p$. Additionally, a set of $\mathcal{F}_p$-rational points on the curve forms an additive abelian group $E(\mathcal{F}_p)$. This group includes a special point called *infinity point* $\mathcal{O}$ and its order is denoted by $\#E(\mathcal{F}_p)$. Let us consider there exists a large prime r that divides $\#E(\mathcal{F}_p)$ such that r does not divide p. In addition, there exists a subgroup $\mathcal{G}[r]$ which has a smallest positive integer k called *embedding degree*, such that r divides $p^k - 1$ but does not divide $p^i - 1 \wedge 1 \leq i < k$. Let us suppose the subgroup $E[r] \cong \mathcal{G}[r] \times \mathcal{G}[r]$ of r-torsion points lies in the elliptic curve $E(\mathcal{F}_{p^k})$ defined over $\mathcal{F}_{p^k}$ [21].

2.2.2 Types of Pairing

The bilinear map can be efficiently implemented with the pairings. There are two types of bilinear pairings, symmetric ($\mathcal{G}_1 = \mathcal{G}_2$) and asymmetric ($\mathcal{G}_1 \neq \mathcal{G}_2$). The symmetric pairings can be called as type-1 pairings [22, 23]. As commented in [23], at the 128-bit security level, the asymmetric type is faster than the symmetric type. Thus, in this dissertation, we concentrate on the use of the asymmetric type. There are two types of asymmetric pairing on bilinear groups $(\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_T)$: asymmetric pairings for which an efficiently-computable homomorphism between $\mathcal{G}_1$ and $\mathcal{G}_2$ is known are called as type-2 pairings and asymmetric pairings for which no efficiently-computable homomorphism is known between $\mathcal{G}_1$ and $\mathcal{G}_2$ are called type-3 pairings [22, 23], where homomorphism denotes the map between two groups (i.e., $\mathcal{G}_1$ and $\mathcal{G}_2$).

2.2.3 Tate Pairing

Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be subgroups of the order r in $E(\mathcal{F}_p)$ and $(\mathcal{F}_{p^k})$, respectively, and $\mathcal{G}_1 \neq \mathcal{G}_2$. Let us say rational points $P \in \mathcal{G}_1$ and $Q \in \mathcal{G}_2$, Tate pairing is a map $e : \mathcal{G}_1 \times \mathcal{G}_2 \to \mathcal{G}_T$ with the following properties:

1. **Bilinearity:** $e(P^a, Q^b) = e(P, Q)^{ab}$, where $0 \leq a, b \leq r$; r: order of ECC of group $\mathcal{G}_1$, $\mathcal{G}_2$, $\mathcal{G}_T$.

2. **Non-degeneracy:** $e(P, Q) \neq 1$.

2.2.4 Barreto-Naehrig Curve

Barreto-Naehrig curve or BN curve is a family of ordinary curves. BN curve was discovered by Barreto and Naehrig in 2005 which is an elegant method in constructing elliptic curves $E(\mathcal{F}_p)$ with embedding degree $k = 12$. Now, it becomes a well-known *pairing-friendly* curve with embedding degree $k = 12$. Note that, a pairing-friendly curve is an elliptic curve which has a small embedding degree and a large prime order subgroup. BN curve is a parameterized curve with the following parameters:

$$p(x) = 36x^4 - 36x^3 + 24x^2 - 6x + 1, \tag{2.2}$$

$$r(x) = 36x^4 - 36x^3 + 18x^2 - 6x + 1. \tag{2.3}$$

2.2.5 Cross-Twisted χ-Based Ate (Xt-Xate) Pairing

This pairing is based on the GMP library [24]. The group order is 254 bits and the embedding degree is 12 (Barreto-Naehrig curve [25, 26]). This pairing library gives the fast pairing called "*Cross-twisted* χ-based Ate (Xt-Xate) pairing" with *subfield-twisted* curve [19, 27]. The number of iterations of Miller's algorithm for the Xt-Xate pairing is about one-quarter of the plain Tate pairing. In addition, using efficiently-computable endomorphism's and isomorphism's, elliptic curve operations are accelerated [28, 29]. Thus, based on good properties of Barreto-Naehrig curve, this library accelerates not only pairings but also the other elliptic curve operations together with Gauss Period Normal Bases (GNB).

This pairing library has capability to calculate a product of several pairings by *multi-pairing* technique that has the following computational efficiencies.

1. *N final exponentiations* are bracketed.

2. Squarings in Miller's algorithm are bracketed.

3. Montgomery trick is efficiently applied for elliptic curve doublings and additions.

The security level is equivalent to the 3000-bit RSA. The library is implemented by C language due to the pursuit of the fastness.

2.3 Complexity Assumptions

As in the underlying system [17], the security of our system is based on the DLIN (Decision LINear) assumption [30], and the q-SFP (Simultaneous Flexible Pairing) assumption [16, 31]. We also adopt n-DHE (DH Exponent) assumption [4] for the accumulator and Symmetric External Diffie-Hellman (SXDH) Assumption for Groth-Sahai (GS) Proofs [18]. Hereafter, we use the notation $a \in_R A$ as sampling a from the set A according to the uniform distribution. Firstly, we describe the measured running time related to some computational algorithms. Here, we define the Probabilistic Polynomial Time (PPT) as a probabilistic Turing machine that takes the random decisions.

Definition 1 (Decision Linear (DLIN) assumption) *For all PPT algorithm $\mathcal{A}$,*

$$|\Pr[\mathcal{A}(g, g^a, g^b, g^{ac}, g^{bd}, g^{c+d}) = 1] - \Pr[\mathcal{A}(g, g^a, g^b, g^{ac}, g^{bd}, g^z) = 1]|$$

is negligible, where $g \in_R \mathcal{G}$ and $a, b, c, d, z \in_R Z_p$.

Definition 2 (q- Simultaneous Flexible Pairing (q-SFP) assumption) *For all PPT algorithm $\mathcal{A}$, the probability*

$$\begin{aligned}
&\Pr[\mathcal{A}(g_z, h_z, g_r, h_r, a, \tilde{a}, b, \tilde{b}, \{(z_j, r_j, s_j, t_j, u_j, v_j, w_j)\}_{j=1}^q)\\
&= (z^*, r^*, s^*, t^*, u^*, v^*, w^*) \in \mathcal{G}^7 \wedge e(a, \tilde{a})\\
&= e(g_z, z^*)e(g_r, r^*)e(s^*, t^*) \wedge e(b, \tilde{b})\\
&= e(h_z, z^*)e(h_r, u^*)e(v^*, w^*)\\
&\wedge z^* \neq 1_{\mathcal{G}} \wedge z^* \neq z_j \text{ for all } 1 \leq j \leq q]
\end{aligned}$$

is negligible, where $(g_z, h_z, g_r, h_r, a, \tilde{a}, b, \tilde{b}) \in \mathcal{G}^8$ and all tuples $\{(z_j, r_j, s_j, t_j, u_j,\ v_j, w_j)\}_{j=1}^q$ satisfy

$$\begin{aligned}
&e(a, \tilde{a}) = e(g_z, z_j)e(g_r, r_j)e(s_j, t_j)\\
&\wedge e(b, \tilde{b}) = e(h_z, z_j)e(h_r, u_j)e(v_j, w_j),
\end{aligned}$$

and $1_{\mathcal{G}}$ is the identity element of group $\mathcal{G}$.

Definition 3 (n-Diffie Hellman Exponent (n-DHE) assumption) *For all PPT algorithm $\mathcal{A}$, the probability*

$$\Pr[\mathcal{A}(g, g^a, \ldots, g^{a^n}, g^{a^{n+2}}, \ldots, g^{a^{2n}}) = g^{a^{n+1}}]$$

is negligible, where $g \in_R \mathcal{G}$ and $a \in_R Z_p$.

2.3.1 Symmetric External Diffie-Hellman (SXDH) Assumption

We assume the decisional Diffie-Hellman problem is hard in both of the groups $\mathbb{G}_1$ and $\mathbb{G}_2$, which is known as the Symmetric External Diffie-Hellman (SXDH) Assumption.

Definition 4 (SXDH assumption) *We say the SXDH assumption holds for $\mathcal{G}_{SXDH}$ if for all non-uniform polynomial time $\mathcal{A}$ and all $b \in \{1, 2\}$, we have*

$$\begin{aligned}
&\Pr[gk \leftarrow \mathcal{G}_{SXDH}(1)^k; \alpha, t \leftarrow \mathbb{Z}_p^* : \mathcal{A}(gk, \alpha\mathcal{P}_b, t\mathcal{P}_b, \alpha t\mathcal{P}_b = 1)]\\
\approx\ &\Pr[gk \leftarrow \mathcal{G}_{SXDH}(1)^k; \alpha, t, r \leftarrow \mathbb{Z}_p^* : \mathcal{A}(gk, \alpha\mathcal{P}_b, t\mathcal{P}_b, \alpha r\mathcal{P}_b = 1)]
\end{aligned}$$

where $gk = (\mathcal{P}_b, G_1, G_2, G_T, e, \mathcal{P}_1, \mathcal{P}_2)$.

2.4 Structure-Preserving Signatures (AHO signatures)

In our system, we utilize the structure-preserving signatures. In the structure-preserving signatures, the verification keys, messages, and signatures are the elements of bilinear groups, and the verification predicate is a conjunction of pairing products. Thus, the knowledge of the signature and messages can be proved by Groth-Sahai proofs. As in [17], we adopt the AHO (Abe, Haralambiev and Ohkubo) signature scheme in [16, 31]. Using the AHO signature scheme, we can sign multiple group elements to obtain a constant-size signature. In the construction, a single group element is signed, and thus we describe the case of single message to be signed.

AHOKeyGen: Select bilinear groups $\mathcal{G}, \mathcal{T}$ with a prime order p and a bilinear map e. Select $g, G_r, H_r \in_R \mathcal{G}$, and $\mu_z, \nu_z, \mu, \nu, \alpha_a, \alpha_b \in_R Z_p$. Compute $G_z = G_r^{\mu_z}, H_z = H_r^{\nu_z}, G = G_r^{\mu}, H = H_r^{\nu}$, $A = e(G_r, g^{\alpha_a}), B = e(H_r, g^{\alpha_b})$. Output the public key as $pk = (\mathcal{G}, \mathcal{T}, p, e, g, G_r, H_r, G_z, H_z, G, H, A, B)$, and the secret key as $sk = (\alpha_a, \alpha_b, \mu_z, \nu_z, \mu, \nu)$.

AHOSign: Given message $M \in \mathcal{G}$ to be signed together with sk, choose $\beta, \epsilon, \eta, \iota, \kappa \in_R Z_p$, and compute $\theta_1 = g^{\beta}$, and

$$\theta_2 = g^{\epsilon-\mu_z\beta} M^{-\mu}, \quad \theta_3 = G_r^{\eta}, \quad \theta_4 = g^{(\alpha_a-\epsilon)/\eta},$$
$$\theta_5 = g^{\iota-\nu_z\beta} M^{-\nu}, \quad \theta_6 = H_r^{\kappa}, \quad \theta_7 = g^{(\alpha_b-\iota)/\kappa}.$$

Output the signature $\sigma = (\theta_1, \ldots, \theta_7)$.

AHOVerify: Given the message M and the signature $\sigma = (\theta_1, \ldots, \theta_7)$, accept these if the following equations hold:

$$\begin{aligned} A &= e(G_z, \theta_1) \cdot e(G_r, \theta_2) \cdot e(\theta_3, \theta_4) \cdot e(G, M), \\ B &= e(H_z, \theta_1) \cdot e(H_r, \theta_5) \cdot e(\theta_6, \theta_7) \cdot e(H, M). \end{aligned}$$

Under the q-SFP assumption [16, 31], this signature is existentially unforgeable against chosen-message attack. This attack generates a pair of a secret key and the corresponding public key, and runs the forging adversary given the public key. In the run, the adversary can request a message, and the signing oracle responds the signature on the message. Finally, the adversary outputs a message and the forged signature. If the signature is valid and a signature on the outputted message has never been responded by the signing oracle, the adversary wins this attack.

Using the re-randomization algorithm in [16, 31], this signature can be publicly randomized to obtain another signature $(\theta'_1, \ldots, \theta'_7)$ on the same message. As a result, in the following Groth-Sahai proof, $(\theta'_i)_{i=3,4,6,7}$ can be safely revealed, while $(\theta'_i)_{i=1,2,5}$ have to be committed, as mentioned in [17].

2.5 Groth-Sahai (GS) Proofs

To prove the secret knowledge in relations of the bilinear maps, we utilize Groth-Sahai (GS) Non-Interactive Witness Indistinguishable (NIWI) proofs [18]. The witness-indistinguishability means that the proof does not reveal the witnesses (satisfying the proved relations) the prover has used. As in [17], we adopt the instantiation based on DLIN assumption. For the bilinear groups, the proof system needs a Common Reference String (CRS) $(\vec{f_1}, \vec{f_2}, \vec{f_3}) \in (\mathcal{G}^3)^3$ for $\vec{f_1} = (f_1, 1, g), \vec{f_2} = (1, f_2, g)$ for some $f_1, f_2 \in \mathcal{G}$, where two types of CRS are used separately for the construction and the security proofs. In the perfect soundness setting, $\vec{f_3} = \vec{f_1}^{\xi_1} \cdot \vec{f_2}^{\xi_2}$ for $\xi_1, \xi_2 \in_R Z_p^*$. The commitment to an element X is computed as $\vec{C} = (1, 1, X) \cdot \vec{f_1}^r \cdot \vec{f_2}^s \cdot \vec{f_3}^t$ for $r, s, t \in_R Z_p^*$. In the perfect soundness setting, the commitment $\vec{C} = (f_1^{r+\xi_1 t}, f_2^{s+\xi_2 t}, X g^{r+s+t(\xi_1+\xi_2)})$ is the linear encryption in [30], and thus it is extractable using $\log_g f_1, \log_g f_2$. On the other hand, in the witness indistinguishability (WI) setting, $\vec{f_1}, \vec{f_2}, \vec{f_3}$ are linearly independent, and thus $\vec{C}$ is perfectly hiding. The DLIN assumption implies the indistinguishability of two types of CRS.

To prove that the committed variables satisfy the pairing relations, the prover prepares the commitments, and replaces the variables in the pairing relations by the commitments. The GS proof allows us to prove the set of pairing product equations:

$$\prod_{i=1}^{n} e(A_i, X_i) \cdot \prod_{i=1}^{n} \prod_{j=1}^{n} e(X_i, X_j)^{a_{ij}} = t$$

for variables $X_1, \ldots, X_n \in \mathcal{G}$ and constants $A_1, \ldots, A_n \in \mathcal{G}, a_{ij} \in Z_p, t \in \mathcal{T}$.

2.6 Summary

In this chapter, we have briefly reviewed the underlying fundamentals of this dissertation including mathematical backgrounds (e.g., groups, bilinear maps, and pairing-based cryptography), the complexity assumptions (e.g., Decision Linear assumption, Diffie-Hellman Exponent assumption and Symmetric External Diffie-Hellman assumption), and cryptographic primitives (e.g., AHO signatures and Groth-Sahai Proofs).

Chapter 3

Proposal of Efficient Proofs for CNF Formulas on Attributes in Pairing-Based Anonymous Credential System

3.1 Introduction

In this chapter, we propose a pairing-based anonymous credential system with the constant size of proofs, such that the combinations of AND and OR relations on attributes can be proved as CNF formulas. In our system, the proof generation cost is more efficient than the system in [6], since only multiplications are needed depending on the number of literals. We extend the efficient accumulators in [4, 5] to handle the proof of the CNF formulas for the construction. Using the accumulator, lots of attributes are accumulated to one value, and we can verify that a value (or multiple values) is included in the accumulator. In the extended accumulator, we can verify that multiple attributes are included in multiple sets, all at once. This leads to the verification of CNF formulas on attributes. As the underlying anonymous credential system, our system is derived from the group signature scheme [17], which adopts structure-preserving signature in [31] as the certificate and Groth-Sahai proofs [18] as non-interactive witness-indistinguishable proofs. As a result, our system is secure in the standard model, as in [6]. In addition, due to the non-interactive proofs, our system is non-interactive where a user can generate the proof on certified attributes by himself and the verifier can verify the proof by himself, as in [6].

One demerit of our system is the increase of public parameters. Let V_ℓ be the set of attributes in the ℓ-th OR clause in the proved CNF formula, and let U be the set of user's certified attributes. This increase happens when the maximum of $|V_\ell \cap U|$ is large for multiple ℓ. We demonstrate that the increase of the public parameters is not so huge in a likely example of CNFs formula used in eID applications.

3.2 Extended Accumulator for Inclusions in Multiple Sets

In [4], an efficient pairing-based accumulator is proposed. The accumulator is generated from a set of values, and we can verify that a single value is included in the set. In [5], the extended version is proposed, where we can verify that multiple values are included in the specified set, all at once. They furthermore extends the accumulator, where we can verify that, for a set U, for all multiple sets $V_1, \ldots, V_L$, a value from U is included in each V_ℓ, i.e., $U \cap V_\ell \neq \emptyset$, all at once. The verification of this type is applied to our construction of the anonymous credential system with proofs for CNF formulas on attributes.

3.2.1 Proposed Construction

Let $V_1, \ldots, V_L$ be L subsets of $\{1, \ldots, n\}$. Define $\mathcal{V} = (V_1, \ldots, V_L)$. Let U be a subset of $\{1, \ldots, n\}$ of size L' satisfying $U \cap V_\ell \neq \emptyset$ for all $1 \leq \ell \leq L$. In the application of our anonymous credential system, each V_ℓ is correspondent to the ℓ-th OR clause in the proved CNF formula. The number of OR clauses may be variable depending on the formula. On the other hand, in the following accumulator, the number of V_ℓ, i.e., L, has to be fixed in the setup phase. Thus, we consider the maximum of OR clauses used in the application, and that L is set as the maximum. Similarly, the maximum of $|V_\ell|$ and the maximum of $|U \cap V_\ell|$ have to be fixed in the setup phase also. Let η_ℓ be the maximum of $|V_\ell|$ in each $1 \leq \ell \leq L$, and let ζ_ℓ be the maximum of $|U \cap V_\ell|$ in each $1 \leq \ell \leq L$.

Then, the following accumulator allows us to confirm $U \cap V_\ell \neq \emptyset$ for all $1 \leq \ell \leq L$, all at once.

AccSetup: This is the algorithm to output the public parameters. Set $c_1 = 1$. For all l with $2 \leq \ell \leq L$, compute $c_\ell = (\eta_{\ell-1} + 1) \cdot c_{\ell-1}$ and set $\mathcal{C} = (c_1, \ldots, c_L)$. We assume that $(\eta_L + 1)c_L < p$. Select bilinear groups $\mathcal{G}, \mathcal{T}$ with a prime order p and a bilinear map e. Select $g \in_R \mathcal{G}$. Select $\gamma \in_R Z_p$, and compute and publish $\{\zeta_\ell\}_{1 \leq \ell \leq L}, \mathcal{C}, p, \mathcal{G}, \mathcal{T}, e, g, g_1 = g^{\gamma^1}, \ldots, g_n = g^{\gamma^n}, g_{n+2} = g^{\gamma^{n+2}}, \ldots, g_{2n} = g^{\gamma^{2n}}$ and $z = e(g,g)^{\gamma^{n+1}}$ as the public parameters.

AccGen: This is the algorithm to compute the accumulator using the public parameters. The accumulator $acc_\mathcal{V}$ of $\mathcal{V}$ is computed as

$$acc_\mathcal{V} = \prod_{1 \leq \ell \leq L} (\prod_{j \in V_\ell} g_{n+1-j})^{c_\ell}.$$

AccWitGen: This is the algorithm to compute the witness that $U \cap V_\ell \neq \emptyset$ for all $1 \leq \ell \leq L$, using the public parameters. Given U, $\mathcal{V}$, and the accumulator $acc_\mathcal{V}$, the witness is computed as $W = \prod_{i \in U} \prod_{1 \leq \ell \leq L} (\prod_{j \in V_\ell}^{j \neq i} g_{n+1-j+i})^{c_\ell}$. Furthermore, the auxiliary parameters are computed as $\delta_\ell = |U \cap V_\ell|$ for all $1 \leq \ell \leq L$.

AccVerify: This is the algorithm to verify that $U \cap V_\ell \neq \emptyset$ for all l with $1 \leq \ell \leq L$, using the witness, the auxiliary parameters, and the public parameters. Given $acc_\mathcal{V}, U$, W and $\{\delta_\ell\}_{1 \leq \ell \leq L}$, set $u = \delta_1 c_1 + \ldots + \delta_L c_L$. Then, accept if

$$\frac{e(\prod_{i \in U} g_i, acc_\mathcal{V})}{e(g, W)} = z^u, \quad 1 \leq \delta_\ell \leq \zeta_\ell, \quad forall 1 \leq \ell \leq L.$$

Remark 1 *The witness W can be efficiently computed by*

$$W = \prod_{1 \le \ell \le L} (\prod_{i \in U} \prod_{j \in V_\ell}^{j \ne i} g_{n+1-j+i})^{c_\ell}$$

.

Remark 2 **AccSetup** *and* **AccGen** *have to be divided into two algorithms. The reason of the separation is as follows. In our application,* **AccSetup** *is used only one time for the key generation. But* **AccGen** *can be used multiple times in attribute proof protocols for different $\mathcal{V} = (V_1, V_2, ...)$, which corresponds to the proved formula.*

3.2.2 Correctness and Security

We can show the correctness and the security as follows.

Theorem 1 *Assume that* **AccSetup***,* **AccGen***,* **AccWitGen** *correctly compute all parameters. Then,* **AccVerify** *accepts $U, acc_{\mathcal{V}}, W, \{\delta_\ell\}_{1 \le \ell \le L}$ that they outputs.*

Proof. We have

$$acc_{\mathcal{V}} = \prod_{1 \le \ell \le L} (\prod_{j \in V_\ell} g_{n+1-j})^{c_\ell},$$

$$W = \prod_{i \in U} \prod_{1 \le \ell \le L} (\prod_{j \in V_\ell}^{j \ne i} g_{n+1-j+i})^{c_\ell}.$$

Thus, the left hand side of the verification equation is as follows.

$$\begin{aligned}
&\frac{e(\prod_{i \in U} g_i, acc_{\mathcal{V}})}{e(g, W)} \\
&\quad = \frac{e(\prod_{i \in U} g_i, \prod_{1 \le \ell \le L} (\prod_{j \in V_\ell} g_{n+1-j})^{c_\ell})}{e(g, \prod_{i \in U} \prod_{1 \le \ell \le L} (\prod_{j \in V_\ell}^{j \ne i} g_{n+1-j+i})^{c_\ell})} \\
&\quad = \frac{e(g, \prod_{i \in U} \prod_{1 \le \ell \le L} (\prod_{j \in V_\ell} g_{n+1-j+i})^{c_\ell})}{e(g, \prod_{i \in U} \prod_{1 \le \ell \le L} (\prod_{j \in V_\ell}^{j \ne i} g_{n+1-j+i})^{c_\ell})} \\
&\quad = e(g, \prod_{i \in U} \prod_{1 \le \ell \le L} (\prod_{j \in V_\ell}^{j = i} g_{n+1-j+i})^{c_\ell}).
\end{aligned}$$

Set $\delta_\ell = |U \cap V_\ell|$ for $1 \le \ell \le L$. Then, the above expression is equal to $e(g, \prod_{1 \le \ell \le L} g_{n+1}^{\delta_\ell c_\ell}) = e(g, g_{n+1})^u = z^u$ for $u = \delta_1 c_1 + \ldots + \delta_L c_L$. Due to $U \cap V_\ell \ne \emptyset$ and $\delta_\ell \le \zeta_\ell$, we obtain $1 \le \delta_\ell \le \zeta_\ell$, for all $1 \le \ell \le L$.

For proving the security of the accumulator, we prepare the following lemma.

Lemma 1 *For any $\bar{\ell}$ $(2 \le \bar{\ell} \le L)$, $c_{\bar{\ell}} > \sum_{1 \le \ell \le \bar{\ell}-1} \eta_\ell \cdot c_\ell$.*

Proof. In case of $\bar{\ell} = 2$, $c_2 = (\eta_1 + 1) \cdot c_1 > \eta_1 \cdot c_1$. For $\bar{\ell} \geq 3$, we assume the case of $\bar{\ell} - 1$, that is $c_{\bar{\ell}-1} > \sum_{1 \leq \ell \leq \bar{\ell}-2} \eta_\ell \cdot c_\ell$, and we will prove the case of $\bar{\ell}$. Using the assumption and $c_{\bar{\ell}} = (\eta_{\bar{\ell}-1} + 1) \cdot c_{\bar{\ell}-1}$, we have

$$\begin{aligned}
& \sum_{1 \leq \ell \leq \bar{\ell}-1} \eta_\ell \cdot c_\ell \\
= \; & \eta_{\bar{\ell}-1} \cdot c_{\bar{\ell}-1} + \sum_{1 \leq \ell \leq \bar{\ell}-2} \eta_\ell \cdot c_\ell \\
< \; & \eta_{\bar{\ell}-1} \cdot c_{\bar{\ell}-1} + c_{\bar{\ell}-1} \\
= \; & (\eta_{\bar{\ell}-1} + 1) \cdot c_{\bar{\ell}-1} \\
= \; & c_{\bar{\ell}}
\end{aligned}$$

Thus, for any $\bar{\ell}$ $(2 \leq \bar{\ell} \leq L)$, we obtain $c_{\bar{\ell}} > \sum_{1 \leq \ell \leq \bar{\ell}-1} \eta_\ell \cdot c_\ell$.

Theorem 2 *Under the n-DHE assumption, any adversary cannot output* $(U, \mathcal{V} = \{V_\ell\}_{1 \leq \ell \leq L}, W, \{\delta_\ell\}_{1 \leq \ell \leq L}$ *where* $U, V_1, \ldots, V_L$ *are subsets of* $\{1, \ldots, n\}$ *and* $\delta_\ell \in Z_p$*, on inputs* $\{\zeta_\ell\}_{1 \leq \ell \leq L}, \mathcal{C}, p, \mathcal{G}, \mathcal{T}, e, g, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}$ *and* z *satisfying the followings:*

- **AccVerify** *accepts* $U, acc_{\mathcal{V}}, W, \{\delta_\ell\}_{1 \leq \ell \leq L}$*,*
- *there exists some* V_ℓ *satisfying* $U \cap V_\ell = \emptyset$*.*

Proof. Assume an adversary which outputs $(U, \mathcal{V}, W, \{\delta_\ell\}_{1 \leq \ell \leq L})$ s.t. **AccVerify** accepts $U, acc_{\mathcal{V}}, W$ and there exists some V_ℓ satisfying $U \cap V_\ell = \emptyset$. Since **AccVerify** accepts these, for $u = \delta_1 c_1 + \ldots + \delta_L c_L$ satisfying $1 \leq \delta_\ell \leq \zeta_\ell$,

$$\frac{e(\prod_{i \in U} g_i, acc_{\mathcal{V}})}{e(g, W)} = z^u = e(g, g_{n+1})^u,$$

where $g_{n+1} = g^{\gamma^{n+1}}$.

From $acc_{\mathcal{V}} = \prod_{1 \leq \ell \leq L} (\prod_{j \in V_\ell} g_{n+1-j})^{c_\ell}$,

$$\frac{e(\prod_{i \in U} g_i, \prod_{1 \leq \ell \leq L} (\prod_{j \in V_\ell} g_{n+1-j})^{c_\ell})}{e(g, W)} = e(g, g_{n+1})^u,$$

$$e(g, \prod_{i \in U} \prod_{1 \leq \ell \leq L} (\prod_{j \in V_\ell} g_{n+1-j+i})^{c_\ell}) = e(g, W {g_{n+1}}^u).$$

Thus, we have

$$\prod_{i \in U} \prod_{1 \leq \ell \leq L} (\prod_{j \in V_\ell} g_{n+1-j+i})^{c_\ell} = W {g_{n+1}}^u.$$

Here, let $\mathcal{L}_1 \subseteq \{1, \ldots, L\}$ be a set of ℓ s.t. V_ℓ includes an element of U, and let $\mathcal{L}_2 \subseteq \{1, \ldots, L\}$ be a set of ℓ s.t. V_ℓ includes no element of U. Let λ_ℓ be $|U \cap V_\ell|$. Then, we have

$$\begin{aligned}
&\prod_{\ell\in\mathcal{L}_1}\prod_{i\in U}(\prod_{j\in V_\ell} g_{n+1-j+i})^{c_\ell}\cdot\prod_{\ell\in\mathcal{L}_2}\prod_{i\in U}(\prod_{j\in V_\ell} g_{n+1-j+i})^{c_\ell}=W{g_{n+1}}^{u},\\
&\prod_{\ell\in\mathcal{L}_1}\prod_{i\in U}(\prod_{j\in V_\ell, j\neq i} g_{n+1-j+i})^{c_\ell}\cdot\prod_{\ell\in\mathcal{L}_1}{g_{n+1}}^{\lambda_\ell c_\ell}\\
&\qquad\qquad\cdot\prod_{\ell\in\mathcal{L}_2}\prod_{i\in U}(\prod_{j\in V_\ell} g_{n+1-j+i})^{c_\ell}=W{g_{n+1}}^{u},\\
&\prod_{\ell\in\mathcal{L}_1}\prod_{i\in U}(\prod_{j\in V_\ell, j\neq i} g_{n+1-j+i})^{c_\ell}\\
&\qquad\cdot\prod_{\ell\in\mathcal{L}_2}\prod_{i\in U}(\prod_{j\in V_\ell} g_{n+1-j+i})^{c_\ell}=W{g_{n+1}}^{u-\sum_{\ell\in\mathcal{L}_1}\lambda_\ell c_\ell}
\end{aligned}\tag{3.1}$$

By setting $\Delta = u - \sum_{\ell\in\mathcal{L}_1}\lambda_\ell c_\ell$,

$$\Delta \;=\; \sum_{1\leq\ell\leq L}\delta_\ell c_\ell - \sum_{\ell\in\mathcal{L}_1}\lambda_\ell c_\ell = \sum_{\ell\in\mathcal{L}_1}(\delta_\ell-\lambda_\ell)c_\ell + \sum_{\ell\in\mathcal{L}_2}\delta_\ell c_\ell.$$

Furthermore, separate $\mathcal{L}_1$ to $\mathcal{L}_1^>$, $\mathcal{L}_1^<$ and $\mathcal{L}_1^=$, where $\mathcal{L}_1^>$ consists of ℓ s.t. $\delta_\ell > \lambda_\ell$, $\mathcal{L}_1^<$ consists of ℓ s.t. $\delta_\ell < \lambda_\ell$, and $\mathcal{L}_1^=$ consists of ℓ s.t. $\delta_\ell = \lambda_\ell$. We can obtain

$$\Delta \;=\; \sum_{\ell\in\mathcal{L}_1^>}(\delta_\ell-\lambda_\ell)c_\ell + \sum_{\ell\in\mathcal{L}_1^<}(\delta_\ell-\lambda_\ell)c_\ell + \sum_{\ell\in\mathcal{L}_2}\delta_\ell c_\ell.$$

Let $\tilde{\ell}$ be the maximum of ℓ s.t. $\ell\notin\mathcal{L}_1^=$ (i.e., $\tilde{\ell}\in\mathcal{L}_1^>$, $\tilde{\ell}\in\mathcal{L}_1^<$, or $\tilde{\ell}\in\mathcal{L}_2$).

Consider two cases.

(i) The first case is that $\tilde{\ell}\in\mathcal{L}_1^<$ (i.e., $\delta_{\tilde{\ell}} < \lambda_{\tilde{\ell}}$). Then, $(\delta_{\tilde{\ell}}-\lambda_{\tilde{\ell}})c_{\tilde{\ell}} \leq -c_{\tilde{\ell}}$. This is why

$$\Delta \leq -c_{\tilde{\ell}} + \sum_{\ell\in\mathcal{L}_1^>}(\delta_\ell-\lambda_\ell)c_\ell + \sum_{\ell\in\mathcal{L}_1^<,\ell\neq\tilde{\ell}}(\delta_\ell-\lambda_\ell)c_\ell + \sum_{\ell\in\mathcal{L}_2}\delta_\ell c_\ell.$$

For $\ell\in\mathcal{L}_1^>$, due to $\lambda_\ell\geq 1$ and $\delta_\ell\leq\zeta_\ell\leq\eta_\ell$, we have $\delta_\ell-\lambda_\ell<\zeta_\ell\leq\eta_\ell$. For $\ell\in\mathcal{L}_1^<$, we have $\delta_\ell-\lambda_\ell<0$, and for $\ell\in\mathcal{L}_2$, we have $\delta_\ell\leq\zeta_\ell\leq\eta_\ell$. Thus,

$$\Delta < -c_{\tilde{\ell}} + \sum_{\ell\in\mathcal{L}_1^>}\eta_\ell c_\ell + \sum_{\ell\in\mathcal{L}_2}\eta_\ell c_\ell.$$

From Lemma 1, we have $c_{\tilde{\ell}} > \sum_{\ell\in\mathcal{L}_1^>}\eta_\ell c_\ell + \sum_{\ell\in\mathcal{L}_2}\eta_\ell c_\ell$, and thus $-c_{\tilde{\ell}} + \sum_{\ell\in\mathcal{L}_1^>}\eta_\ell c_\ell + \sum_{\ell\in\mathcal{L}_2}\eta_\ell c_\ell < 0$. Therefore, we can obtain $\Delta<0$. On the other hand, we obtain

$$\Delta \;=\; \sum_{1\leq\ell\leq L}\delta_\ell c_\ell - \sum_{\ell\in\mathcal{L}_1}\lambda_\ell c_\ell > -\sum_{\ell\in\mathcal{L}_1}\eta_\ell c_\ell,$$

due to $\delta_\ell>0$ and $\lambda_\ell\leq\zeta_\ell\leq\eta_\ell$. From Lemma 1, we have $\sum_{\ell\in\mathcal{L}_1,\ell\neq\tilde{\ell}}\eta_\ell c_\ell < c_{\tilde{\ell}}$, and

$$\sum_{\ell\in\mathcal{L}_1}\eta_\ell c_\ell < c_{\tilde{\ell}} + \eta_{\tilde{\ell}}c_{\tilde{\ell}} < (\eta_L+1)c_L < p.$$

Thus, $\Delta > -p$. Therefore, we have $\Delta\neq 0 \pmod{p}$.

(ii) The other case is that $\tilde{\ell} \in \mathcal{L}_1^>$ (i.e., $\delta_{\tilde{\ell}} > \lambda_{\tilde{\ell}}$) or $\tilde{\ell} \in \mathcal{L}_2$ (i.e., $\lambda_{\tilde{\ell}} = 0$). Then, in case of $\tilde{\ell} \in \mathcal{L}_1^>$, due to $(\delta_{\tilde{\ell}} - \lambda_{\tilde{\ell}})c_{\tilde{\ell}} \geq c_{\tilde{\ell}}$, we obtain

$$\Delta \geq c_{\tilde{\ell}} + \sum_{\ell \in \mathcal{L}_1^>, \ell \neq \tilde{\ell}} (\delta_\ell - \lambda_\ell)c_\ell + \sum_{\ell \in \mathcal{L}_1^<} (\delta_\ell - \lambda_\ell)c_\ell + \sum_{\ell \in \mathcal{L}_2} \delta_\ell c_\ell.$$

In case of $\tilde{\ell} \in \mathcal{L}_2$, due to $\delta_{\tilde{\ell}} \geq 1$, we obtain

$$\Delta \geq c_{\tilde{\ell}} + \sum_{\ell \in \mathcal{L}_1^>} (\delta_\ell - \lambda_\ell)c_\ell + \sum_{\ell \in \mathcal{L}_1^<} (\delta_\ell - \lambda_\ell)c_\ell + \sum_{\ell \in \mathcal{L}_2, \ell \neq \tilde{\ell}} \delta_\ell c_\ell.$$

In the both cases (i.e., $\tilde{\ell} \in \mathcal{L}_1^>$ and $\tilde{\ell} \in \mathcal{L}_2$), for any $\ell \in \mathcal{L}_1^>$, we have $\delta_\ell - \lambda_\ell > 0$, and for any $\ell \in \mathcal{L}_2$, $\delta_\ell > 0$, and thus

$$\Delta > c_{\tilde{\ell}} + \sum_{\ell \in \mathcal{L}_1^<} (\delta_\ell - \lambda_\ell)c_\ell.$$

Due to $\tilde{\ell} > \ell$ for any $\ell \in \mathcal{L}_1^<$, from Lemma 1 and $\lambda_\ell - \delta_\ell < \zeta_\ell < \eta_\ell$,

$$c_{\tilde{\ell}} > \sum_{\ell \in \mathcal{L}_1^<} \eta_\ell c_\ell > \sum_{\ell \in \mathcal{L}_1^<} (\lambda_\ell - \delta_\ell)c_\ell,$$

and thus

$$c_{\tilde{\ell}} + \sum_{\ell \in \mathcal{L}_1^<} (\delta_\ell - \lambda_\ell)c_\ell > 0.$$

This is why we obtain $\Delta > 0$. On the other hand,

$$\Delta \leq \sum_{1 \leq \ell \leq L} \delta_\ell c_\ell = \sum_{1 \leq \ell \leq L-1} \delta_\ell c_\ell + \delta_L c_L \leq \sum_{1 \leq \ell \leq L-1} \eta_\ell c_\ell + \eta_L c_L.$$

From Lemma 1, $\Delta \leq (\eta_L + 1)c_L < p$. Therefore, in this case, also $\Delta \neq 0 \pmod{p}$.

Thus, from equation (3.1), we obtain

$$g_{n+1} = (W^{-1} \cdot \prod_{\ell \in \mathcal{L}_1} \prod_{i \in U} (\prod_{j \in V_\ell, j \neq i} g_{n+1-j+i})^{c_\ell}$$

$$\cdot \prod_{\ell \in \mathcal{L}_2} \prod_{i \in U} (\prod_{j \in V_\ell} g_{n+1-j+i})^{c_\ell})^{1/\Delta},$$

due to $\Delta \neq 0 \pmod{p}$.

For any $i \in U$ and any $j \in V_\ell$ with $\ell \in \mathcal{L}_1$ satisfying $j \neq i$, $g_{n+1-j+i} \neq g_{n+1}$. For any $i \in U$ and any $j \in V_\ell$ with $\ell \in \mathcal{L}_2$, $g_{n+1-j+i} \neq g_{n+1}$, since such V_ℓ does not include elements in U. Therefore, we can compute g_{n+1} given $g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}$, which contradicts n-DHE assumption.

Remark 3 *Note that, in Theorem 2, the adversary is allowed to output $\delta_\ell \neq |U \cap V_\ell|$, since the condition in* **AccVerify** *is only $1 \leq \delta_\ell \leq \zeta_\ell$. This implies that, under the n-DHE assumption, for any $u' = \delta_1 c_1 + \cdots + \delta_L c_L$ s.t. $u' \neq u = |U \cap V_1|c_1 + \cdots + |U \cap V_L|c_L$ and $1 \leq \delta_\ell \leq \zeta_\ell$, the adversary cannot output $(U, \mathcal{V}, W, \{\delta_\ell\}_{1 \leq \ell \leq L})$ s.t. $\frac{e(\prod_{i \in U} g_i, acc_{\mathcal{V}})}{e(g,W)} = z^{u'}$, when there exists some V_ℓ satisfying $U \cap V_\ell = \emptyset$.*

3.3 Syntax and Security Model of Anonymous Credential System

We consider the *non-interactive* anonymous credential system, where a user can generate the proof on certified attributes by himself/herself and the verifier can verify the proof by himself/herself. This is similar to the group signature scheme [32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44], and thus our security model is derived from that of the group signature scheme.

The security model of the group signature scheme consists of traceability, non-frameability, and anonymity. The traceability means that once a group signature is opened, it identifies a group member who joined the group. The non-frameability means that no one except a group member can issue a valid group signature that can be identified to the member. In this dissertation, since we concentrate on the function of the anonymous attribute proof, we do not care about the tracing. Thus, in the following model, we omit the functions on the tracing. This is why the non-frameability is omitted, and the traceability is replaced by the similar requirement *misauthentication resistance*. The misauthentication resistance means the soundness of the attribute proof. Note that the combination of the group signature scheme [17] can achieve the tracing.

3.3.1 Syntax

The attribute value is indexed by an integer from $\{1, \ldots, n\}$, where n is the total number of attribute values. As in [5], all attribute values in all attribute types are indexed by using the universal set $\{1, \ldots, n\}$. We describe a CNF formula Ψ on attributes using the indexes as follows: $(a_{11} \vee a_{12} \vee \cdots) \wedge (a_{21} \vee a_{22} \vee \cdots) \wedge \cdots$ with $a_{11}, a_{12}, \ldots, a_{21}, a_{22}, \ldots \in \{1, \ldots, n\}$. Each literal $a_{11}, a_{21}, \cdots$ means that the proving user owns the attribute of the index. Set $V_1 = (a_{11}, a_{12}, \ldots)$, $V_2 = (a_{21}, a_{22}, \ldots), \ldots,$ and $V_L = (a_{L1}, a_{L2}, \ldots)$. Set U be the set of attributes (indexes) of the proving user. We assume that $|V_\ell|$ has the upper bound, η_ℓ, for all $1 \leq \ell \leq L$, and assume that the size of $|U \cap V_\ell|$ has the upper bound, ζ_ℓ, for each $1 \leq \ell \leq L$. Also, we assume the maximum number of clauses in any CNF formula, L. Although n is an assumptive upper bound of both $|V_l|$ and $|U \cap V_l|$, we need each maximum number of $|V_\ell|$ and $|U \cap V_\ell|$, as η_ℓ, ζ_ℓ, respectively. This is because the efficiency of our system depends on the sizes of η_ℓ and ζ_ℓ, and thus we want to set the sizes of η_ℓ and ζ_ℓ as small values as possible.

The anonymous credential system consists of the following algorithms:

IssuerKeyGen: The inputs of this algorithm are $n, L, \eta_\ell, \zeta_\ell$ for all $1 \leq \ell \leq L$. The outputs are issuer's public key *ipk* and issuer's secret key *isk*.

CertObtain: This is an interactive protocol between a probabilistic algorithm **CertObtain-**$\mathcal{U}_k$ for the k-th user and a probabilistic algorithm **CertObtain-**$\mathcal{I}$ for an issuer, where the issuer issues the certificate including the attributes to the user. **CertObtain-**$\mathcal{U}_k$, on input *ipk* and $U_k \subset \{1, \ldots, N\}$ that is indexes corresponding to the attribute values of the user, outputs the certificate $cert_k$ ensuring the attributes of the user. On the other hand, **CertObtain-**$\mathcal{I}$ is given ipk, isk as inputs.

ProofGen: This probabilistic algorithm, on inputs *ipk*, U_k, $cert_k$, Ψ that is the predicate on attributes to be proved, outputs the proof σ.

Verify: This is a deterministic algorithm for verification. The input is ipk, a proof σ, and the predicate Ψ. Then the output is 'valid' if the attributes in U_k satisfy Ψ, or 'invalid' otherwise.

3.3.2 Security Model

The security model consists of misauthentication resistance and anonymity. The misauthentication resistance requirement captures the soundness of the attribute proof. This means that an adversary $\mathcal{A}$ cannot forge a proof for a predicate, where the attributes of any user corrupted by $\mathcal{A}$ do not satisfy the predicate. The anonymity requirement captures the anonymity and unlinkability of proofs, as in the group signatures. The unlinkability is a stronger anonymity. In linkable anonymous setting, the verier can collect the use history of an anonymous user, since the verier can determine the sameness of the prover. Then, the history may de-anonymize the prover by relating the dates or frequency. Also, if one transaction is de-anonymized by some other method, all the transactions of the prover are de-anonymized. Therefore, the unlinkability is needed.

Misauthentication Resistance.

Consider the following misauthentication resistance game.

Misauthentication Resistance Game: The challenger runs **IssuerKeyGen**, and obtains ipk(issuer's public key) and isk(issuer's secret key). He provides $\mathcal{A}$ with ipk, and run $\mathcal{A}$. He sets CU with empty, where CU denotes the set of IDs of users corrupted by $\mathcal{A}$. In the run, $\mathcal{A}$ can query the challenger about the following issuing query:

C-Issuing: $\mathcal{A}$ can request the k-th user's certificate on U_k. Then, $\mathcal{A}$ as the user executes **CertObtain** protocol with the challenger as the issuer. The challenger adds k to CU.

Finally, $\mathcal{A}$ outputs a predicate Ψ^*, and a proof σ^*.

Then, $\mathcal{A}$ wins if

1. $\textbf{Verify}(ipk, \sigma^*, \Psi^*) =$ valid, and
2. for all $k \in CU$, U_k does not satisfy Ψ^*.

Misauthentication resistance requires that for all PPT $\mathcal{A}$, the probability that $\mathcal{A}$ wins the misauthentication resistance game is negligible.

Anonymity.

Consider the following anonymity game.

Anonymity Game: The challenger runs **IssuerKeyGen**, and obtains ipk, isk. He provides $\mathcal{A}$ with ipk, isk, and run $\mathcal{A}$. He sets HU with empty, where HU denotes the set of IDs of users who are not corrupted by $\mathcal{A}$. In the run, $\mathcal{A}$ can query the challenger, as follows.

H-Issuing: $\mathcal{A}$ can request the k-th user's certificate on U_k. Then, $\mathcal{A}$ as the issuer executes **CertObtain** protocol with the challenger as the user. The challenger adds k to HU.

Proving: $\mathcal{A}$ can request the k-th user's proof on a predicate Ψ. Then, the challenger responds the proof on Ψ of user k, if $k \in HU$.

During the run, as the challenge, $\mathcal{A}$ outputs a predicate Ψ^*, and two users k_0 and k_1, such that both U_{k_0} and U_{k_1} satisfy Ψ^*. If $k_0 \in HU$ and $k_1 \in HU$, the challenger chooses $\phi \in_R \{0, 1\}$, and responds the proof on Ψ^* of user k_ϕ. After that, similarly, $\mathcal{A}$ can make the queries.

Finally, $\mathcal{A}$ outputs a bit ϕ' indicating its guess of ϕ. If $\phi' = \phi$, $\mathcal{A}$ wins. We define the advantage of $\mathcal{A}$ as $|\Pr[\phi' = \phi] - 1/2|$.

Anonymity requires that for all PPT $\mathcal{A}$, the advantage of $\mathcal{A}$ on the anonymity game is negligible.

Remark 4 *Ψ denotes each predicate queried in the proving query, and Ψ^* denotes a predicate in the attacker's output.*

3.4 Proposed Anonymous Credential System

3.4.1 Construction Idea

For the verification of CNF formulas, we use our extended accumulator. Consider the following CNF formula: $(a_{11} \vee a_{12} \vee \ldots) \wedge (a_{21} \vee a_{22} \ldots) \wedge \ldots$, where $a_{11}, a_{12} \ldots a_{21}, a_{22} \ldots \in \{1, \ldots, n\}$ means that the proving user owns the corresponding attribute. Set $V_1 = (a_{11}, a_{12}, \ldots)$, $V_2 = (a_{21}, a_{22}, \ldots)$,..., and $V_L = (a_{L1}, a_{L2}, \ldots)$. Let U be the set of attributes (indexes) of the proving user. Then, using the accumulator, we can confirm that $U \cap V_\ell \neq \emptyset$ for any V_ℓ. This means that the attributes in U satisfies this CNF formula, since some attribute in U is one of attributes in every OR clause expressed by V_ℓ.

Our construction is based on the anonymous credential system using the AHO signatures and GS proofs. This is derived from the construction of a group signature scheme in [17], which is secure in the standard model. In the underlying system, the certificate is an AHO signature, where attributes of the user are unified to one element and embedded as $\prod_{i \in U} g_i$, to apply it to the verification of accumulator.

In our system, a part of accumulator verification, $\frac{e(\prod_{i \in U} g_i, acc_\mathcal{V})}{e(g, W)} = z^u$, can be proved without revealing secret information by directly using the GS proof for the pairing relation. However, the other part of the verification, $1 \leq \delta_\ell \leq \zeta_\ell$ where $u = \delta_1 c_1 + \ldots + \delta_L c_L$, needs another technique. We utilize the set membership proof technique [45] to prove this relation, while hiding parameters δ_ℓ. Since δ_ℓ may indicate some secret information (i.e., $|U \cap V_\ell|$) of the user, we need the zero-knowledge type of proof. As the preparation, the issuer publishes signatures on ${g_1}^u$ for all $u \in \{\sum_{\ell=1}^{L} \delta_\ell c_\ell | 1 \leq \delta_\ell \leq \zeta_\ell$ for all $1 \leq \ell \leq L\}$. In the attribute proof, the user proves the knowledge of the signature on a committed value to convince the verifier that the committed value is g_1^u such that u satisfies the conditions.

To use the accumulator, L, which is the number of V_ℓ, has to be fixed. On the other hand, in the CNF formula of the input, the number of clauses, L' is less than or equal to L. Then, we introduce a special attribute a_{SP} that every user always owns. To the CNF formula with L' clauses, a_{SP} is added such as the number of clauses becomes L. Namely, given formula $\Psi = (a_{11} \vee a_{12} \vee \cdots) \wedge (a_{21} \vee a_{22} \vee \cdots) \wedge \cdots (a_{L'1} \vee a_{L'2} \vee \cdots)$ is extended to $\Psi' = (a_{11} \vee a_{12} \vee \cdots) \wedge (a_{21} \vee a_{22} \vee \cdots) \wedge \cdots (a_{L'1} \vee a_{L'2} \vee \cdots) \wedge a_{\mathrm{SP}} \wedge \cdots a_{\mathrm{SP}}$. The literal a_{SP} is always true, this extended formula is the same as the original.

3.4.2 Proposed Construction

The details of this construction are described as follows.

IssuerKeyGen

The input of this algorithm consists of n, L, η_ℓ, and ζ_ℓ for all $1 \leq \ell \leq L$. This algorithm executes **AccSetup** to obtain the public parameters of the extended accumulator, generates key pairs of AHO signatures, generates CRS for GS NIWI proof, and prepares AHO signatures on ${g_1}^u$ for all $u \in \{\sum_{\ell=1}^{L} \delta_\ell c_\ell | 1 \leq \delta_\ell \leq \zeta_\ell \text{ for all } 1 \leq \ell \leq L\}$. The outputs are *ipk* and *isk*.

1. Select bilinear groups $\mathcal{G}$, $\mathcal{T}$ with the same order p and the bilinear map e, and $g \in_R \mathcal{G}$.

2. Generate public parameters of the extended accumulator: Set $c_1 = 1$. For all $2 \leq \ell \leq L$, compute $c_\ell = (\eta_{\ell-1} + 1) \cdot c_{\ell-1}$ and set $\mathcal{C} = (c_1, \ldots, c_L)$. Select $\gamma \in_R Z_p$, and compute

$$\begin{aligned} pk_{\mathrm{acc}} = (&\mathcal{C}, g_1 = g^{\gamma^1}, \ldots, g_n = g^{\gamma^n}, \\ &g_{n+2} = g^{\gamma^{n+2}}, \ldots, g_{2n} = g^{\gamma^{2n}}, \\ &z = (g, g)^{\gamma^{n+1}}). \end{aligned}$$

3. Generate two key pairs for the AHO signature:

$$\begin{aligned} pk_{\mathrm{AHO}}^{(d)} &= (G_r^{(d)}, H_r^{(d)}, G_z^{(d)}, H_z^{(d)}, G^{(d)}, H^{(d)}, A^{(d)}, B^{(d)}), \\ sk_{\mathrm{AHO}}^{(d)} &= (\alpha_a^{(d)}, \alpha_b^{(d)}, \mu_z^{(d)}, \nu_z^{(d)}, \mu, \nu), \end{aligned}$$

   for $d = 0$ and $d = 1$.

4. Generate a CRS for the GS NIWI proof: select $\vec{f} = (\vec{f_1}, \vec{f_2}, \vec{f_3})$, where $\vec{f_1} = (f_1, 1, g)$, $\vec{f_2} = (1, f_2, g)$, $\vec{f_3} = \vec{f_1}^{\xi_1} \cdot \vec{f_2}^{\xi_2}$ for $f_1, f_2 \in_R \mathcal{G}$ and $\xi_1, \xi_2 \in_R Z_p^*$.

5. For $\mathcal{C}$, define set $\Phi = \{u = \sum_{\ell=1}^{L} \delta_\ell c_\ell | 1 \leq \delta_\ell \leq \zeta_\ell \text{ for all } 1 \leq \ell \leq L\}$, where $|\Phi| = \prod_{1 \leq \ell \leq L} \zeta_\ell$. For every $u \in \Phi$, generate the AHO signature on g_1^u, using $sk_{\mathrm{AHO}}^{(0)}$. The signature is denoted as $\tilde{\sigma}_u = (\tilde{\theta}_{u1}, \ldots, \tilde{\theta}_{u7})$.

6. Output the issuer public key

$$ipk = (p, \mathcal{G}, \mathcal{T}, e, g, pk_{\mathrm{AHO}}^{(0)}, pk_{\mathrm{AHO}}^{(1)}, pk_{\mathrm{acc}}, \vec{f}, \{\tilde{\sigma}_u\}_{u \in \Phi}),$$

   and the issuer secret key $isk = (sk_{\mathrm{AHO}}^{(0)}, sk_{\mathrm{AHO}}^{(1)})$.

CertObtain

This is an interactive protocol between **CertObtain-$\mathcal{U}_k$** (user) and **CertObtain-$\mathcal{I}$** (issuer). The common inputs of this protocol consists of *ipk*, and U_k that is the indexes of attribute values of the user. The input of **CertObtain-$\mathcal{I}$** is *isk*. In this protocol, to the user, the issuer sends a certificate $cert_k$ including the AHO signature σ_k on $P_k = \prod_{i \in U_k} g_i$. We introduce a special attribute value a_{SP}. Every user has a_{SP}. The output of **User** is $cert_k$.

1. **CertObtain-$\mathcal{I}$**: Generate $P_k = \prod_{i \in U_k} g_i$.

2. **CertObtain-$\mathcal{I}$**: Using $sk^{(1)}_{\mathrm{AHO}}$, generate an AHO signature $\sigma_k = (\theta_1, \ldots, \theta_7)$ on message P_k. Return σ_k to **CertObtain-$\mathcal{U}_k$** as the certificate.

3. **CertObtain-$\mathcal{U}_k$**: Compute $P_k = \prod_{i \in U_k} g_i$, and verify the AHO signature σ_k on P_k. Output $cert_k = (P_k, \sigma_k)$.

ProofGen

The inputs of this algorithm are *ipk*, U_k, $cert_k$, and the CNF formula Ψ. For a given formula $\Psi = (a_{11} \vee a_{12} \vee \cdots) \wedge (a_{21} \vee a_{22} \vee \cdots) \wedge \cdots (a_{L'1} \vee a_{L'2} \vee \cdots)$ with $a_{11}, a_{12}, \ldots, a_{21}, a_{22}, \ldots \in \{1, \ldots, n\}$, define $V_1 = \{a_{11}, a_{12}, \ldots\}, V_2 = \{a_{21}, a_{22}, \ldots\}, V_L = \{a_{L1}, a_{L2}, \ldots\}$. If $L' < L$, define $V_{L'+1} = \cdots = V_L = \{a_{\mathrm{SP}}\}$. This algorithm generates the GS NIWI proof proving that P_k satisfies the accumulator verification for the accumulator $acc_{\mathcal{V}}$ indicating the proved predicate Ψ, and proving P_k is signed as a AHO signature σ_k by the issuer's public key. In the accumulator verification, the GS proof for an AHO signature on g_1^u is also utilized.

1. Compute the accumulator:
$$acc_{\mathcal{V}} = \prod_{1 \le \ell \le L} (\prod_{j \in V_\ell} g_{n+1-j})^{c_\ell}.$$

2. Compute the witness
$$W_{\mathcal{V}} = \prod_{i \in U_k} \prod_{1 \le \ell \le L} (\prod_{j \in V_\ell}^{j \neq i} g_{n+1-j+i})^{c_\ell}$$
that U_k satisfies $\mathcal{V}$ for $acc_{\mathcal{V}}$, and sets $u = \delta_1 c_1 + \ldots + \delta_L c_L$, where $\delta_\ell = |U_k \cap V_\ell|$ for all $1 \le \ell \le L$.

3. Set $\tau_u = g_1^u$. From *ipk*, select the AHO signature $\tilde{\sigma}_u = (\tilde{\theta}_{u1}, \ldots, \tilde{\theta}_{u7})$ on the g_1^u.

4. Compute GS commitments $com_{P_k}, com_{W_{\mathcal{V}}}, com_{\tau_u}$ to $P_k, W_{\mathcal{V}}, \tau_u$. Then, re-randomize the AHO signature σ_k to obtain $\sigma'_k = \{\theta'_1, \ldots, \theta'_7\}$, and compute GS commitments $\{com_{\theta'_i}\}_{i \in \{1,2,5\}}$ to $\{\theta'_i\}_{i \in \{1,2,5\}}$. Similarly, re-randomize the AHO signature $\tilde{\sigma}_u$ to obtain $\tilde{\sigma}'_u = \{\tilde{\theta'}_{u1}, \ldots, \tilde{\theta'}_{u7}\}$, and compute GS commitments $\{com_{\tilde{\theta'}_{ui}}\}_{i \in \{1,2,5\}}$ to $\{\tilde{\theta'}_{ui}\}_{i \in \{1,2,5\}}$.

5. Generate the GS proofs $\{\pi_i\}_{i=1}^{5}$ s.t.

$$
\begin{aligned}
1_{\mathcal{T}} &= e(P_k, acc_{\mathcal{V}}) \cdot e(g, W_{\mathcal{V}})^{-1} \cdot e(\tau_u, g_n)^{-1}, &(3.2)\\
A^{(1)} \cdot e(\theta'_3, \theta'_4)^{-1} &= e(G_z^{(1)}, \theta'_1) \cdot e(G_r^{(1)}, \theta'_2) \cdot e(G^{(1)}, P_k), &(3.3)\\
B^{(1)} \cdot e(\theta'_6, \theta'_7)^{-1} &= e(H_z^{(1)}, \theta'_1) \cdot e(H_r^{(1)}, \theta'_5) \cdot e(H^{(1)}, P_k), &(3.4)\\
A^{(0)} \cdot e(\tilde{\theta}'_{u3}, \tilde{\theta}'_{u4})^{-1} &= e(G_z^{(0)}, \tilde{\theta}'_{u1}) \cdot e(G_r^{(0)}, \tilde{\theta}'_{u2}) \cdot e(G^{(0)}, \tau_u), &(3.5)\\
B^{(0)} \cdot e(\tilde{\theta}'_{u6}, \tilde{\theta}'_{u7})^{-1} &= e(H_z^{(0)}, \tilde{\theta}'_{u1}) \cdot e(H_r^{(0)}, \tilde{\theta}'_{u5}) \cdot e(H^{(0)}, \tau_u), &(3.6)
\end{aligned}
$$

6. Output $\sigma = (\{\theta'_i\}_{i=3,4,6,7}, \{\tilde{\theta'}_{ui}\}_{i=3,4,6,7}, com_{P_k}, com_{W_{\mathcal{V}}}, com_{\tau_u}, \{com_{\theta'_i}\}_{i=1,2,5}, \{com_{\tilde{\theta'}_{ui}}\}_{i=1,2,5}, \{\pi_i\}_{i=1}^{5})$.

Equation (3.2) shows the verification relations of accumulator:

$$\frac{e(\prod_{i \in U_k} g_i, acc_{\mathcal{V}})}{e(g, W_{\mathcal{V}})} = e(g_1^u, g_n) = z^u,$$

where $P_k = \prod_{i \in U_k} g_i$ and $\tau_u = g_1^u$. The equations (3.3), (3.4) show the knowledge of the AHO signature of P_k, i.e., the certificate $cert_k$. The equations (3.5), (3.6) shows the knowledge of the AHO signature of τ_u. This ensures that $1 \leq \delta_\ell \leq \zeta_\ell$ where $u = \delta_1 c_1 + \ldots + \delta_L c_L$. Thus, together with the equation (3.2), it ensures the verification of the accumulator. This is why the verifier is ensured that $U_k \cap V_\ell \neq \emptyset$, i.e, attributes in U_k satisfies the CNF formula Ψ.

Verify

The inputs of this algorithm are ipk, the proof σ, and the CNF formula Ψ.

1. Compute the accumulator $acc_{\mathcal{V}}$, as in **ProofGen**.

2. Accept σ, if the verifications of all GS proofs $\{\pi_i\}_{i=1}^{5}$ are successful.

3.4.3 Security

We can prove the following security of our construction.

Theorem 3 *The proposed system satisfies the misauthentication resistance under the security of the AHO signatures and the extended accumulators.*

Proof. To win the misauthentication resistance game, the adversary $\mathcal{A}$ must output a predicate Ψ^* and a proof σ^* satisfying

1. **Verify**(ipk, σ^*, Ψ^*) = valid, and

2. for all $k \in CU$, U_k does not satisfy Ψ^*.

Let $\sigma^* = (\{\theta'^*_i\}_{i=3,4,6,7}, \{\tilde{\theta'}^*_{ui}\}_{i=3,4,6,7}, com^*_{P_k}, com^*_{W_{\mathcal{V}}}, com^*_{\tau_u}, \{com^*_{\theta'_i}\}_{i=1,2,5}, \{com^*_{\tilde{\theta'}_{ui}}\}_{i=1,2,5}, \{\pi^*_i\}_{i=1}^{5})$. By utilizing the CRS for the perfect soundness setting, the GS commitments $com^*_{P_k}, com^*_{W_{\mathcal{V}}}, com^*_{\tau_u}, \{com^*_{\theta'_i}\}_{i=1,2,5}, \{com^*_{\tilde{\theta'}_{ui}}\}_{i=1,2,5}$ are extractable, as mentioned in Section 2.5. Thus, we can extract $P^*_k, W^*_{\mathcal{V}}, \tau^*_u$ satisfying the equation (3.2) for accumulator verification with $acc^*_{\mathcal{V}}$ that is derived from Ψ^*, the re-randomized AHO signature $\sigma'^*_k =$

$\{\theta'^*_1, \ldots, \theta'^*_7\}$ for P^*_k satisfying the equations (3.3),(3.4), and the re-randomized AHO signature $\tilde{\sigma'}^*_u = \{\tilde{\theta'}^*_{u1}, \ldots, \tilde{\theta'}^*_{u7}\}$ for τ^*_u satisfying the equations (3.5),(3.6). Set $\tau^*_u = g_1^{u^*}$ for unknown u^*. From the equation (3.2), we obtain $1_{\mathcal{T}} = e(P^*_k, acc^*_{\mathcal{V}}) \cdot e(g, W^*_{\mathcal{V}})^{-1} \cdot e(g_1^{u^*}, g_n)^{-1}$, and thus the verification relation of our accumulator:

$$\frac{e(P^*_k, acc^*_{\mathcal{V}})}{e(g, W^*_{\mathcal{V}})} = z^{u^*}.$$

We distinguish the following cases from each other.

- **Type 1 forgeries.** This is the case that the AHO signature on P^*_k was never issued to any corrupted user k (i.e., $k \in CU$).
- **Type 2 forgeries.** This is the case that the AHO signature on P^*_k was issued to a corrupted user k. In this case, we have two sub-cases:

  a. The AHO signature of τ^*_u was never issued.

  b. The AHO signature of τ^*_u was issued.

Using Type 1 and Type 2.a forgeries, we can obtain a forger against the AHO signatures, as follows.

Type 1 forgeries. With the adversary $\mathcal{A}$, simulate the misauthentication resistance game, as follows. The public key of AHO signatures is given to the AHO signature forger. This is set as $pk^{(1)}_{\text{AHO}}$ in **IssuerKeyGen**. Choose and compute other parameters in *ipk*, as in the real algorithm of **IssuerKeyGen**. Note that the CRS for the GS NIWI proof $\vec{f}$ is for the perfect soundness setting. Then, run $\mathcal{A}$ on *ipk*. In the misauthentication resistance game, $\mathcal{A}$ can request **C-Issuing** queries. For the **C-Issuing** query on U_k, compute $P_k = \prod_{i \in U_k} g_i$, and request the AHO signature on P_k to the signing oracle of the AHO signatures. Respond the AHO signature as $cert_k$. Finally, $\mathcal{A}$ outputs a predicate Ψ^*, and a proof σ^*. As mentioned above, we can extract the AHO signature σ'^*_k on P^*_k. In this case, since the AHO signature σ'^*_k was never issued for P^*_k, this implies the forgery against the AHO signature.

Type 2.a forgeries. Setting the given public key as $pk^{(0)}_{\text{AHO}}$ and executing the similar way to the case of Type 1 forgeries, we can obtain the forger against the AHO signature.

Using Type 2.b forgeries, we can obtain an adversary against the extended accumulator, as follows.

Type2.b forgeries. The public parameters of the extended accumulator are given to this accumulator adversary. Then, choose and compute other parameters in *ipk*, as the real algorithm of **IssuerKeyGen**, and run $\mathcal{A}$. In the run, each **C-Issuing** query is responded as in the real algorithm, since $sk^{(0)}_{\text{AHO}}$ and $sk^{(1)}_{\text{AHO}}$ are generated as usual. Finally, $\mathcal{A}$ outputs a predicate Ψ^*, and a proof σ^*. As mentioned above, we can extract the AHO signature σ'^*_k on P^*_k, and the AHO signature $\tilde{\sigma'}^*_u$ on τ^*_u, together with witness $W^*_{\mathcal{V}}$. In this case, the AHO signature σ'^*_k on P^*_k was correctly issued to some corrupted user k, and thus $P^*_k = \prod_{i \in U^*_k} g_i$ for U^*_k and U^*_k does not satisfy Ψ^*. On the other hand, the AHO signature on τ^*_u was correctly issued, and thus $\tau^*_u = g_1^{u^*}$ for $u^* \in \Phi$. This means $u^* = \delta^*_1 c_1 + \cdots + \delta^*_L c_L$, and $1 \leq \delta^*_\ell \leq \zeta_\ell$ with all $1 \leq \ell \leq L$. This is why **AccVerify** accepts $U^*_k, acc^*_{\mathcal{V}}, W^*_{\mathcal{V}}, \{\delta^*_\ell\}_{1 \leq \ell \leq L}$. Therefore, we can forge the witness $W^*_{\mathcal{V}}$, when U^*_k does not satisfy Ψ^* (i.e., $U^*_k \cap V_\ell = \emptyset$ for some ℓ).

Theorem 4 *The proposed system satisfies the anonymity under the DLIN assumption.*

Proof. Consider the sequence of games, as follows.

Game 1. This is the anonymity game for the proposed system. The challenger generates ipk and isk using **IssuerKeyGen** algorithm, where the CRS is prepared for the perfect soundness setting. The challenger runs the adversary $\mathcal{A}$ with ipk, isk. $\mathcal{A}$ can request **H-Issuing** queries and **Proving** queries. In the response of **H-Issuing** on U_k, the challenger records $cert_k$. For **Proving** queries and the challenge, the challenger responds using ipk and $cert_k$.

Game 2. In **IssuerKeyGen** algorithm, the challenger generates the CRS for the WI setting. Namely, choose linear independent $\vec{f_1}, \vec{f_2}, \vec{f_3}$. The others are the same as Game 1.

Let S_1, S_2 denote the events that $\phi' = \phi$ in Game 1, 2, respectively. In Game 2, the proof of responded in the challenge consists of the GS commitments that are perfectly hiding in the WI setting, the GS proofs that reveal no information about the underlying witness, and the randomized AHO signatures $\{\theta'_i\}_{i=3,4,6,7}$, $\{\tilde{\theta'}_{ui}\}_{i=3,4,6,7}$ that are information-theoretically independent of the signed messages and the remaining AHO signatures. Thus, we have $\Pr[S_2] = 1/2$. On the other hand, $|\Pr[S_1] - \Pr[S_2]|$ is negligible under the DLIN assumption, as in [18]. Therefore, the advantage of $\mathcal{A}$, i.e., $|\Pr[S_1] - 1/2|$, is negligible, which means that the proposed system is anonymous.

3.4.4 Protection against Replay Attack

In the authentication, the re-use of the proof should be prevented. In our system, the proof depends on the predicate, as the signature depends on the message. Thus, to prevent the re-use, we can make the predicate including a random nonce, as follows. Consider T-bit nonce $b_1 \cdots b_T$ with $b_t \in \{0, 1\}$. Then, we introduce virtual attributes $\tilde{a}_{1,0}, \tilde{a}_{1,1}, \ldots, \tilde{a}_{T,0}, \tilde{a}_{T,1}$. To the original CNF formula Ψ, we append the following OR clause, $(a_{\mathrm{SP}} \vee \tilde{a}_{1,b_1} \vee \cdots \vee \tilde{a}_{T,b_T})$. Since this clause includes a_{SP}, this clause is meaningless in the attribute proof (i.e,. this clause is satisfied anytime). On the other hand, an appended formula is different from other appended formulas, due to the random nonce. In this method, the public parameters of the accumulator for the virtual attributes are additionally required. The number of the parameters is only $2T$.

3.5 Design of Protocols

In this section, we describe the protocols of the eID application of our anonymous credential system. The system model is shown in Fig. 3.1. At first, the issuer publishes the public key ipk. Then, the user registers himself/herself along with a set of his/her particular attributes U to the issuer for certification by using the **CertObtain** protocol via a secure channel (①∼③). The user, utilizing the issued certificate, requests a service to the Service Provider (SP) (④). Then, the SP specifies some attributes to the user that the SP wants to be proved by the user (⑤). This specification forms the logical relation of CNF formula consisting of AND and OR relations, depending on the SP's requirement. Then, the user generates a proof of knowledge for the possession of the certificate w.r.t. the specified attribute(s) and

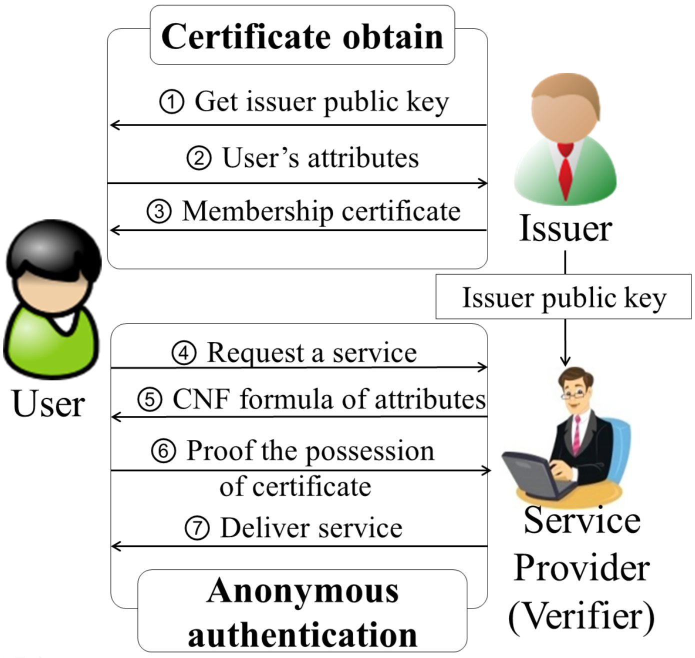

Figure 3.1: System model.

prove the proof to the SP (verifier) anonymously by using **ProveGen** and **Verify** (⑥). The SP grants the user to access a requested service (⑦), if and only if the verification of user's proof is valid. In the anonymous credential system model, we can consider two types of attributes for representation: one type is a small finite-set attribute such as the gender or the occupation, and another type is a string attribute such as the full-name or the address. An example of those attributes used in the eID is depicted in Table 3.1. We design our protocols: the user registration and anonymous authentication as follows.

3.5.1 User Registration

Figure 3.2 shows the user registration through **ObtainCert** protocol. This protocol comprises the following two steps. In advance, the user has to fetch the issuer public key *ipk*, for example by downloading it from the official web-site of issuer. We assume that there is an existing access control application (e.g., based on username and password) to permit the user to use a communication channel (which is out of scope of this dissertation). Then, the user can use such channel to perform the **ObtainCert** protocol together with the issuer.

(1) Registration of user's attributes.
The user requests the registration. He/she sends the set of his/her attributes, U for certification (①∼③).

(2) Issuing the certificate.

(a) The issuer computes the accumulator, certifies it into the certificate *cert* and sends the *cert* to the user (③∼⑤).

Table 3.1: Example of string and small finite-set attributes.

String	Small Finite-set	Example Values
1) full-name	8) gender	male,female
2) address	9) day of birth	1–31
3) phone number	10) month of birth	1–12
4) identity number	11) year of birth	1930–2005
5) issuance date	12) marital status	single,marriage
6) expiration date	13) nationality	193 recognized states
7) email address	14) hometown	200 allocated cities
	15) city living	200 allocated cities
	16) residence status	citizen,imigrant,...
	17) religion	moslem,christian,...
	18) blood type	A,B,O,AB
	19) occupation	student,teacher,...
	20) academic degree	B.S.,M.S,Ph.D.,...
	21) major	science,economic,...
	22) year of graduated	1970–2005
	23) workplace	200 allocated cities
	24) main language	100 allocated lang.
	25) 2nd language	100 allocated lang.
	26) topic of interest	music,sport,...
	27) favorite color	red,green,blue,...
	28) favorite music	pop,rock,jazz,...
	29) favorite sport	baseball,tennis,...

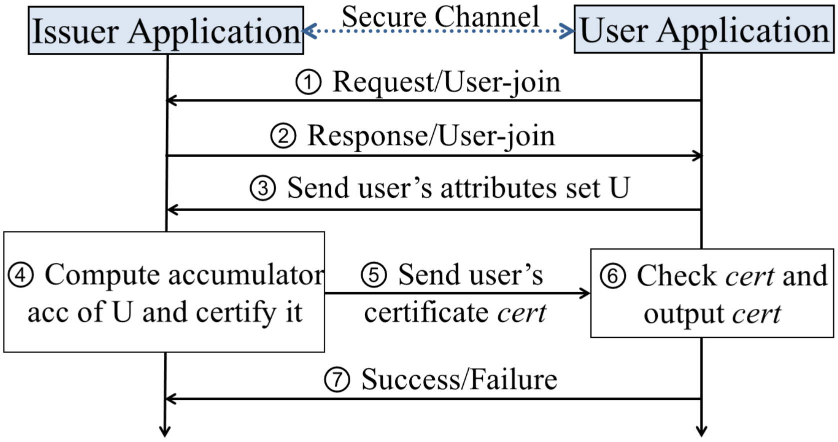
Issuer Application
Secure Channel
User Application
① Request/User-join
② Response/User-join
③ Send user's attributes set U
④ Compute accumulator acc of U and certify it
⑤ Send user's certificate *cert*
⑥ Check *cert* and output *cert*
⑦ Success/Failure

Figure 3.2: User registration protocol.

(b) The user checks the validity of *cert* to ensure whether the *cert* was sent by the legitimate issuer or not. If it is valid, the user outputs the users' certificate *cert* (⑥∼⑦).

3.5.2 Anonymous Authentication

The anonymous authentication based on attributes is performed by using the **ProveGen** and **Verify**. Figure 3.3 shows the authentication process to allow the user to access the service provided by an SP through wireless networks. This protocol comprises the following two steps. In this protocol, the user can prove his/her possession of particular attributes by proving the given CNF formula of specific attributes.

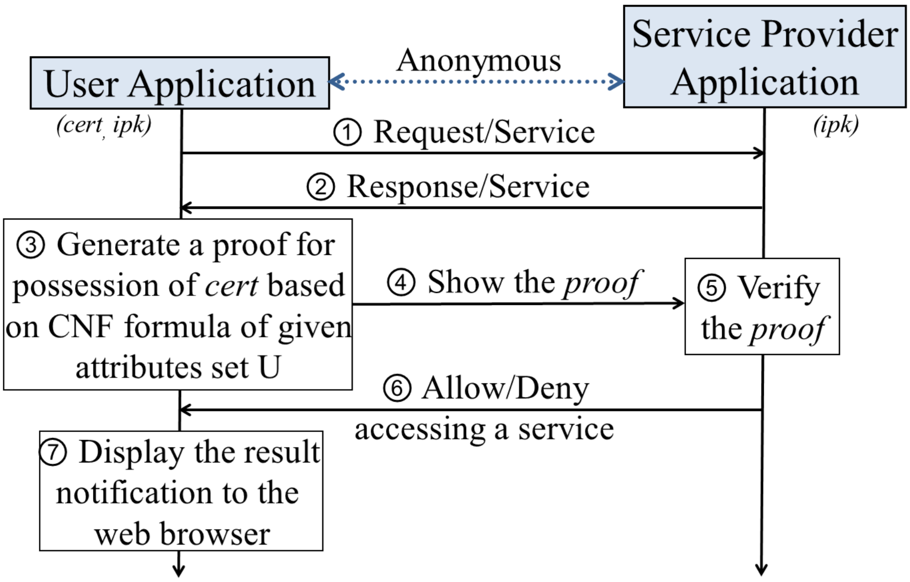
User Application
Anonymous
Service Provider Application
(cert, ipk)
(ipk)
① Request/Service
② Response/Service
③ Generate a proof for possession of *cert* based on CNF formula of given attributes set U
④ Show the *proof*
⑤ Verify the *proof*
⑥ Allow/Deny accessing a service
⑦ Display the result notification to the web browser

Figure 3.3: Authentication protocol.

(1) Generation and transmission of a proof for possession of certificate.

(a) The user requests a service to the SP (service provider). Then, the SP provides the user a CNF formula of the specified attributes. Depending on the SP's requirement, one or multiple attributes can be selected from the specified set of attributes all at once. (①∼②).

(b) By using *cert* and the selected attribute(s), the user generates the proof for the possession of such attribute(s). This proof means that such selected attribute(s) is (are) included in the certificate and is satisfied by the CNF formula. Then, he shows the proof to the SP (③∼④).

(2) Verification of a proof for possession of certificate.

(a) The SP verifies the proof. If it is valid, the SP grants the user to access the service (resp. reject the service) (⑤∼⑥).

(b) The verification result is displayed in the web browser of the user which indicates either accept or reject to access the service (⑦).

3.6 Comparisons

We compare our system with the system in [6] that allows us to prove inner product relations on attributes. Since both systems achieve constant-size proofs, we mainly concentrate on the computational costs for generating the proofs.

As mentioned in [6], using the inner product relations and suitable attribute encoding, CNF formulas (also DNF formulas) can be expressed (the encoding is shown in [11]). As the encoding, a polynomial is used. Consider the polynomial $f(x) = c_d x^d + \cdots + c_0 x^0$ and the coefficients vector $\vec{p} = (c_d, \ldots, c_0)$. In the system of [6], the user's attribute is expressed by $\omega \in Z_p$. Let $\vec{\omega} = (\omega^d \bmod p, \ldots, \omega^0 \bmod p)$. Using the inner product proof system, the user can prove $\vec{\omega} \cdot \vec{p} = 0$. This means that $f(\omega) = c_d \omega^d + \cdots + c_0 \omega^0 = 0$. In the computations of the system, each element of the vector is set as the exponent on some base parameter. This means that an exponentiation for each element is needed, and thus the computational cost depends on the size of the vector. In the encoding of OR relation, for example, $(x = a_1) \vee (x = a_2)$ can be encoded as the univariate polynomial $(x - a_1) \cdot (x - a_2)$. The case of more literals is similar. Let d be the number of attribute values that are used in the OR relation. Then, this encoding needs d coefficients and the vector size becomes d. Thus, the computational cost of the proof is d exponentiations. Consider a CNF formula such that the ℓ-th OR clause has d_ℓ literals $(1 \leq \ell \leq L)$, where L is the number of OR clauses. Then, by the encoding for AND relation in [11], the computational cost becomes $\sum_{1 \leq \ell \leq L} d_\ell$ exponentiations.

In our system, the computations of $acc_\mathcal{V}$ and $W_\mathcal{V}$ need only $O(d)$ multiplications, while L exponentiations by c_ℓ are needed. In cases that some OR clauses have lots of literals, the cost of our system is much more efficient than the system [6].

On the other hand, our system has disadvantages against the system [6]: The inner product proofs can be converted to the proofs of DNF formulas, while our system cannot support the DNF formulas directly. In some applications, DNF formulas may be better. Another disadvantage is the public key size. In our scheme, we need to publish signatures for set Φ, where the size $|\Phi|$ is $\prod_{1 \leq \ell \leq L} \zeta_\ell$. The size may become large based-on application.

To show the effectiveness of our system, we discuss a concrete application. Consider the eID application as mentioned in Introduction. In such an application, a user often proves the following CNF formula on user's attributes.

$$gender = male \wedge birth\_year \in \{1900, \ldots, 1992\}$$
$$\wedge profession \in \{student, teacher, professor, \ldots\} \wedge \cdots .$$

Namely, for each attribute type, the user's attribute value is included in a set of attribute values. This example considers that a service provider grasps user's profile that can be useful for marketing, while serious private data are concealed. By the OR relation of $birth\_year$, the user proves that he/she is adult, but the concrete age is concealed. As in this example, for the proof including OR relations with lots of literals, the system [6] needs heavy computations of $\sum_{1 \leq \ell \leq L} d_\ell$ exponentiations. On the other hand, as shown above, our system has the additional public key size. However, in this eID application, ζ_ℓ (i.e., the maximum of $|U \cap V_\ell|$) can be 1 for the attribute types such that the user owns a single attribute value such as gender and birth_year. For the attribute types such as the user owns multiple attribute values such as the professions, ζ_ℓ can be more than 1. Most attribute types are former, and for the latter type, a user does not own lots of attribute values. Thus, in this application, the public key size depending on $\prod_{1 \leq \ell \leq L} \zeta_\ell$ is not so huge.

3.7 Summary

In this chapter, we propose and describe a pairing-based anonymous credential system with the constant-size proofs of CNF formulas using our extended accumulator. Our system has the great advantage that the proof generation cost is more efficient than the system [6], since only multiplications are needed whose number depends on the number of literals. The compensation is the increase of public parameters. We demonstrate that, for the CNF formulas that can be often used in eID applications, the increase is not so huge.

Chapter 4

Implementation and Evaluation of an Anonymous Credential System with Constant-Size Proofs on CNF Formulas and Efficient Proof Generations

4.1 Introduction

In Chapter 3, we proposed a pairing-based anonymous credential system with the constant-size proofs for CNF formulas, where the proof generation cost is more efficient than the system in [6], since only multiplications are needed depending on the number of literals. However, since the proposed system has never been implemented yet, the practicality in the current PC environment is unknown. Thus, the evaluation based on the implementation is required.

In this chapter, we implement our system using a fast pairing library. In the pairing-based construction, the pairing calculation's cost is dominant, compared to other operations such as multiplications and exponentiations in the underlying Elliptic Curve Cryptosystem (ECC). We adopt the pairing library in [19] where the pairing can be computed fast using "Cross-twisted χ-based Ate (Xt-Xate) pairing."

To evaluate the practicality of our implemented system, we measured the processing time in the proof protocol and the data size of the proof. In the measurements, we are changing parameters such as the maximum number of OR clauses in the proved CNF formula and the number of the user's own attributes. From the results of the measurements, we confirm that the proof size is constant w.r.t. the parameters, and the verification time is constant and very fast (about 300 ms in a usual PC). However, the result shows that the time of the proof generation in the prover increases non-linearly, when the maximum number of OR clauses and the number of the user's attribute increase simultaneously. In case that both the numbers are less than 30, the processing time is less than about 1 sec., and thus our system is practical. In case of more numbers, it is inefficient. The reduction of the processing time is one of our future works.

In the implemented system, the public key size also depends on the parameters: the total number of attributes, and the number of attributes matched between the user's attribute set and clauses in the CNF formula. Then, we measured the data size of the public key, when changing these parameters. The results show that the public key size greatly increases when these parameters increase. In case of 10,000 attributes, the size amounts to more than 3 MBytes. In case that the matched number increases, the public key size increases. The reduction is shown in Chapter 5.

4.2 Outline of Cryptographic Construction

Generally, the anonymous credential system can be constructed as follows. The certificate is a cryptographic digital signature, where multiple attribute values are signed. The signature is denoted as $Sign(a_1, a_2, \ldots, a_k)$ that is the signing function on attribute values $a_1, a_2, \ldots, a_k$. By using a zero-knowledge proof technique, the user can prove only the knowledge that he/she owns $Sign(a_1, a_2, \ldots, a_k)$, where the attributes can be secret.

In [31], a signature scheme using a bilinear map called pairings is proposed. By using zero-knowledge proofs for pairings in [18] called GS (Groth-Sahai) proofs, the knowledge of the signature can be proved. Thus, our system adopts the signatures and zero-knowledge proofs.

To obtain the constant-size proof, multiple attributes in the user and the proved formula have to be compressed. For the compression, our system utilizes a cryptographic technique, an accumulator. A basic accumulator is generated from a set of values. We can verify that a single value is included in the set. In our construction, an extended accumulator is used, where we can verify that, for a set U, for all multiple sets $V_1, \ldots, V_L$, a value from U is included in each V_ℓ, i.e., $U \cap V_\ell \neq \emptyset$, all at once. Consider the following CNF formula:

$$(a_{11} \wedge a_{12} \wedge \ldots) \vee (a_{21} \wedge a_{22} \ldots) \vee \ldots$$

where $a_{11}, a_{12} \ldots a_{21}, a_{22} \ldots \in \{1, \ldots, n\}$ means that the proving user owns the corresponding attribute. Set $V_1 = (a_{11}, a_{12}, \ldots)$, $V_2 = (a_{21}, a_{22}, \ldots) \ldots$ and $V_L = (a_{L1}, a_{L2}, \ldots)$. Set U be the set of attributes (indexes) of the proving user. Then, using the accumulator, we can confirm that $U \cap V_\ell \neq \emptyset$ for any V_ℓ. This means that the attributes in U satisfies this CNF formula, since some attribute in U is one of attributes in every OR clause expressed by V_ℓ. The attributes of U are compressed to one value, and the CNF formulas $V_1 \ldots, V_L$ are also compressed.

In the verification of the accumulator, the relation $u = |U \cap V_1|c_1 + \cdots + |U \cap V_L|c_L$ has to be proved, where u is an integer parameter output from the pairing relation of the accumulator, and $c_1, \ldots, c_L$ are public parameters that are computed in the key generation of the accumulator. However, values $|U \cap V_1|, \ldots, |U \cap V_L|$ have to be secret, since these values reveal some information on U (user's attributes). In our system, in **IssuerKeyGen**, the issuer generates signatures for all candidates of u. Due to $U \cap V_\ell \neq \emptyset$, and the maximum ζ_ℓ of $|U \cap V_\ell|$, the set of the candidates is $\Phi = \{u = \sum_{\ell=1}^{L} \delta_\ell c_\ell | 1 \leq \delta_\ell \leq \zeta_\ell \text{ for all } 1 \leq \ell \leq L\}$. Namely, the issuer generates signatures for all the elements of this set, which are included in the public key, *ipk*. In **AttributeProof**, the user proves the knowledge of a signature of u generated in the accumulator, where the GS proof brings the zero-knowledge, i.e., u is secret.

However, this trick brings the communication cost to the system. The size of $|\Phi|$ is $\prod_{1 \leq \ell \leq L} \zeta_\ell$. The public key size increases by $|\Phi|$ signatures. This cost is evaluated later based on the implementation.

4.3 Implementation

In this section, we describe the implementation of our proposed anonymous credential system. To show the practicality of our system, we implemented the system and measured the computational overhead.

4.3.1 Utilized Pairing Library

The construction of the implemented anonymous credential system is mainly based on the bilinear groups and bilinear map. We utilize the library based on "Cross-twisted χ-based Ate (Xt-Xate) pairing" [19] with 254-bit group order and the embedding degree is 12. The security level is equivalent to the 128-bit AES. The library is based on the GMP library and implemented by C language due to the pursuit of the fastness.

4.3.2 Instantiation of GS Proof

For the implementation, we need to instantiate the GS zero-knowledge proof [18] concretely. Based on the utilized pairing and cryptographic assumption, there are three types of instantiations. Since we utilize asymmetric pairing from the viewpoint of efficiency, we adopt the GS proof based on the SXDH assumption for the asymmetric pairing.

The GS proof deals with any pairing equation to be proved, but the size of the proof is relatively large in case of the general pairing equation. For one pairing equation, four $\mathcal{G}_1$-elements and four $\mathcal{G}_2$-elements are needed. On the other hand, in the simplified relation called linear equation, the size is reduced to only two $\mathcal{G}_1$-elements. In the implemented system, there are five pairing equations to be proved. Among them, four equations are for proving the knowledge of signatures [31], which are linear equations. The other equation is for the verification of the accumulator. The accumulator is verified by the following pairing equation:

$$1_{\mathcal{G}_T} = e(P_U, acc_{\mathcal{V}}) \cdot e(g, W_{\mathcal{V}})^{-1} \cdot e(\tau_u, g_n)^{-1},$$

where $P_U, W_{\mathcal{V}}, \tau_u$ are secret parameters and the others are public parameters. In the linear equation, the first inputs must be public parameters and the second inputs must be secret ones. In our implementation, the above equation is modified as follows:

$$1_{\mathcal{G}_T} = e(acc_{\mathcal{V}}, P_U) \cdot e(g, W_{\mathcal{V}})^{-1} \cdot e(g_n, \tau_u)^{-1}.$$

From the property of the bilinear map, the modification is valid. This modification allows us to utilize the GS proof of the linear equation to reduce the proof size.

4.4 Experiments for Evaluations

In this section, we present the experimental results to show the effectiveness and practicality of our proposed scheme. To assume the mobile environments, we measured the computation time and the data sizes of this scheme using a desktop PC for both the signer and the verifier. The environments of the experiments are shown in Table 4.1. For measuring the time, we utilize gettimeofday_sec() method that is a JAVA API.

Table 4.1: Environments of implementation and experiments.

CPU	Intel Core2 Duo (3GHz)
Memory	3.9 GBytes
OS	Ubuntu 12.04 (kernel Linux-3.2.0-59-generic)
Compiler	GCC-4.5.2
Library	GMP-5.0.2 (Multiple Precision Arithmetic Library) ELiPS (Pairing Library)

In this experiment, we suppose the eID application that is mentioned in Chapter 1. In the application, we can consider the popular example of the following CNF formula:

$$gender = male \wedge birth\_year \in \{1900, \ldots, 1992\}$$

$$\wedge profession \in \{student, teacher, professor, \ldots\} \wedge \cdots .$$

This example means that a service provider grasps the user's profile that can be useful for marketing, while serious private data are concealed. By the OR relation of *birth_year*, the user proves that he is an adult, but the concrete age is concealed. In the measurements, we evaluate the dependency on L (the number of clauses), $|V_\ell|$ (the size of a clause), ζ_ℓ (the size of $|U \cap V_\ell|$). Thus, we performed three types of measurements: *Measurement 1*, *Measurement 2* and *Measurement 3*. In each measurement, except for the changed parameters, we set $L = 10$, $|V_\ell| = 10$, $\zeta_\ell = 1$.

4.4.1 Measurement 1

In Measurement 1, we measured the processing time of **Attribute Proof** protocol for the user and verifier, in case of changing the maximum number of OR clauses (L) in the proved CNF formula. In this measurement, for simplicity, we set $L = |U|$. This is a popular setting in the eID application. As the above example, an OR clause is set up for a single attribute type of the user, such as the gender and birth year. Thus, it is likely that the maximum number of clauses is similar to the number of user's attributes. The measured times on the variation of L (also, $|U|$) are shown in Fig. 4.1. This figure shows that the user time increases non-linearly w.r.t. L. On the other hand, the verification time is constant and about 300 ms. This is because the verification consists of only the verification of the GS proofs, whose cost does not depend on the size of $U, V_1, \ldots, V_L$ and L. The size of proof sent between the user and verifier in **AttributeProof** protocol is also constant for L, and concretely 3.8 Kbytes only. Therefore, the verification time and the size indicate the practicality of our system. On the other hand, the user time is practical in case of $L \leq 30$, but inefficient in case of $L > 30$.

To explore why the user time increases, we measured the detailed processing time of user in the **AttributeProof** protocol. The proof generation in the user consists of three parts: the accumulator computation, the witness computation, and the GS proof generation. Fig. 4.2 shows the processing time in each part. The figure shows that the accumulator computation is very fast, compared to the others, and the variation is linear w.r.t. L, because the computation consists of only multiplications and the loop depending on L is once. The GS proof generation is constant as in the verification, but needs more time than the accumulator computation, since the generation needs the exponentiations. On the other hand,

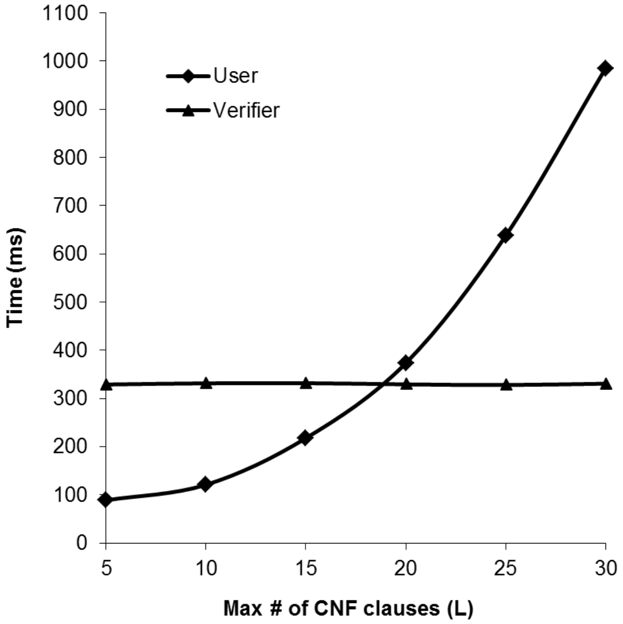

Figure 4.1: Processing times of **AttributeProof** for prover and verifier.

the processing time for the witness computation increases greatly and non-linearly w.r.t. L, because the witness computation requires two loops depending on L (i.e, L and $|U|$). If U contains more than L elements, the time increases more. Hence, to reduce the time of the witness computation is one of our future works.

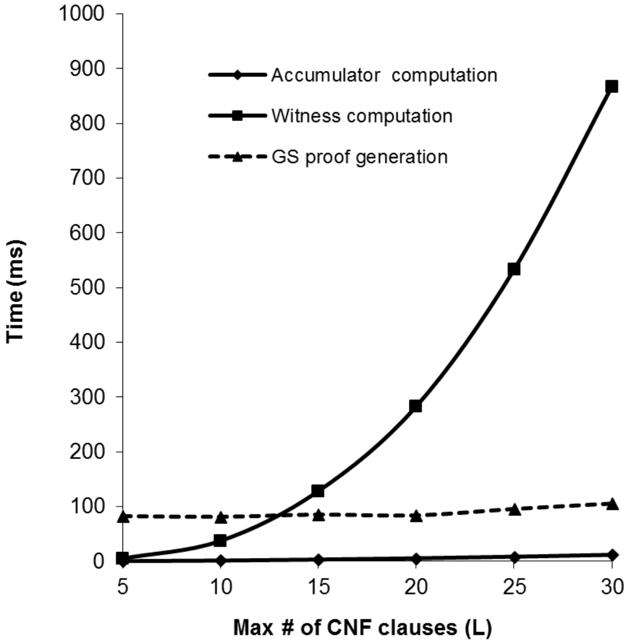

Figure 4.2: Detailed processing times in prover.

4.4.2 Measurement 2

Next, we target the dependency on $|V_\ell|$ (i.e., the number of literals in an OR clause in the predicate). The processing time is almost constant, although the user time is changing from 160 ms – 200 ms. In this case, the witness computation in the prover does not influence the processing time so much, since the loop depending on $|V_\ell|$ is once and thus the variation of the processing time is similar to that in the accumulator computation. On the other hand, $|V_\ell|$ influences the size of the issuer public key, *ipk*. Thus, in Measurement 2, the size of *ipk* is measured for different sizes of V_1. Fig. 4.3 shows the variation of the size of *ipk*. From this figure, we can confirm that the size of *ipk* increases almost linearly w.r.t. the sizes of V_1. This is because the increase of $|V_\ell|$ means the increase of the number of attributes, and thus the increase of public parameters of the accumulator. In case of $|V_1|$ = 10,000 and keeping the sizes of other V_ℓ as 10, the size of *ipk* is more than 3 MBytes. Hence, our future modification will cover the reduction of *ipk* size in such a lots of literals.

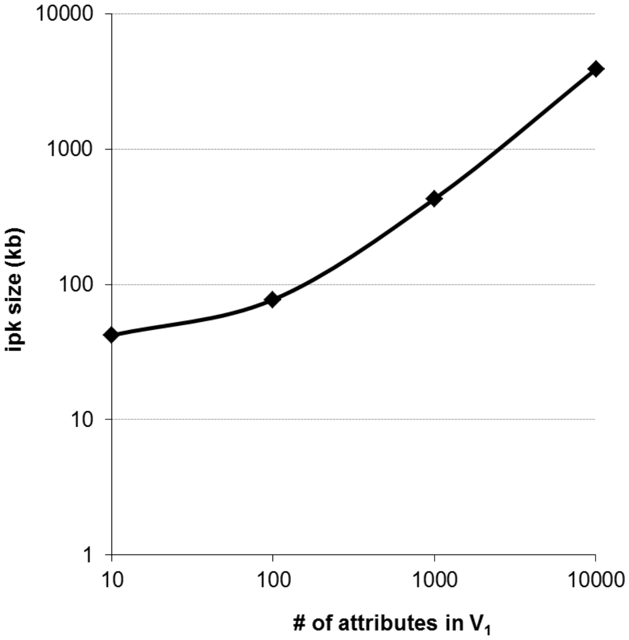

Figure 4.3: Size of *ipk* for size of V_1.

4.4.3 Measurement 3

Finally, we target the dependency on ζ_ℓ (i.e., $|U \cap V_\ell|$). In this case, the variation of ζ_ℓ does not influence the processing time at all, but it impacts on the size of *ipk*. Thus, in this measurement, when we change ζ_ℓ, we measured the size of *ipk*. The results of Measurement 3 are shown in Table 4.2. This table shows that *ipk* size increases when each ζ_ℓ increases. Here we consider four different cases of ζ_ℓ and the values of rest of the ζ_ℓ is 1 (i.e., $\zeta_5 \sim \zeta_{10}$ = 1). When ζ_1 = 5, ζ_2 = 10, ζ_3 =15 and ζ_4 = 20, the size of *ipk* amounts to more than 7 MBytes. However, in the eID applications, ζ_ℓ tends to be 1 for most attribute types. In the above example, a user owns a single birth year and thus $|U \cap V_\ell| = 1$. For the attribute type of profession, a user may own multiple professions such as a Ph.D student and a company researcher. In this case, it may be $|U \cap V_\ell| = 2$. However, usually the maximum number $|U \cap V_\ell|$ is not so large.

Table 4.2: Size of *ipk* for ζ_ℓ

ζ_1	ζ_2	ζ_3	ζ_4	Size of *ipk* (KBytes)
5	1	1	1	45
5	10	1	1	86
5	10	15	1	539
5	10	15	20	7516

4.5 Summary

In this chapter, we show the implementation of the anonymous authentication system which is proposed in Chapter 3. The experimental measurements show the practicality of the proposed system in case that the parameters $L, |V_\ell|$ are not so large. The future works include the reduction of the witness computation time and the reduction of *ipk* size in case of large parameters. The reduction in case of larger ζ_ℓ (the number of matched attributes between user's attribute set and the clause in the CNF formula) is shown in Chapter 5.

Chapter 5

Extension for Public-Key Size Reduction

5.1 Introduction

In Chapter 3, we proposed an anonymous credential system with the constant size of proofs for CNF formulas. In the proposed system, the proof generation cost is smaller than the similar existing system in [6], since only multiplications depending on the number of literals are needed. One demerit of our previous system is the increase of public parameters, which brings a large communication cost to the system.

Hence, in this chapter, we propose an extension to reduce the public key size and evaluate the efficiency based on an implementation. In the previous system, to ensure the correctness of a value u in the verification, the issuer publishes signatures on all candidates of u, which causes the pubic key size increase. In our extension, we consider two values u_1 and u_2 s. t. $u = u_1 + u_2$, and signatures on u_1 and u_2 are separately published. This modification reduces the public key size to $2\sqrt{N}$ for the original size N. However, this trick increases the computational cost by about 20%, compared to our previous system. For our implementation, the proving and verification times are less than 200 ms and 500 ms respectively in a usual PC, which is still practical.

5.2 Previous Anonymous Credential System for CNF Formulas and the Problem

Generally, the anonymous credential system can be constructed as follows. The certificate is a cryptographic digital signature, where multiple attribute values are signed. The signature is denoted as $Sign(a_1, a_2, \ldots, a_k)$ that is a signing function on attribute values $a_1, a_2, \ldots, a_k$. By using a *zero-knowledge proof* technique, the user can prove only the knowledge that he owns $Sign(a_1, a_2, \ldots, a_k)$, where the attributes can be secret.

In [31], a signature scheme using a bilinear map called pairings is proposed. By using zero-knowledge proofs for pairings in [18] called GS (Groth-Sahai) proofs, the knowledge of the signature can be proved. Thus, this system adopts this group signatures and zero-knowledge proofs.

To obtain the constant-size proof, multiple attributes in the user and the proved formula have to be compressed using an accumulator. In the accumulator, it can be verified that, for a set U, for all multiple sets $V_1, \ldots, V_L$, a value from U is included in each V_ℓ, i.e., $U \cap V_\ell \neq \emptyset$, all at once. Consider the following CNF formula:

$$(a_{11} \wedge a_{12} \wedge \ldots) \vee (a_{21} \wedge a_{22} \ldots) \vee \ldots$$

where $a_{11}, a_{12} \ldots\ a_{21}, a_{22} \ldots \in \{1, \ldots, n\}$. Set $V_1 = (a_{11}, a_{12}, \ldots)$, $V_2 = (a_{21}, a_{22}, \ldots)$ and $V_l = (a_{l1}, a_{l2}, \ldots)$. Set U be the set of attributes (indexes) of the proving user. Then, using the accumulator, we can confirm that $U \cap V_\ell \neq \emptyset$ for any V_ℓ. This means that the attributes in U satisfy this CNF formula, since some attribute in U is one of the attributes in every OR clause expressed by V_ℓ. The attributes of U are compressed to one value, and the CNF formulas $V_1 \ldots, V_L$ are also compressed.

The accumulator is verified by

$$\frac{e(P_U, acc_{\mathcal{V}})}{e(g, W)} = z^u \tag{5.1}$$

where P_U and $acc_{\mathcal{V}}$ are the accumulated values of U and V_ℓ's, W is the witness value, g and z are public parameters, and e is the pairing function. Furthermore, the relation $u = |U \cap V_1|c_1 + \cdots + |U \cap V_L|c_L$ has to be proved, where $c_1, \ldots, c_L$ are public parameters that are computed in the key generation of the accumulator.

Values $|U \cap V_1|, \ldots, |U \cap V_L|$ have to be secret, since the values reveal some information on U (user's attributes). For the zero-knowledge proof, the *set membership proof* of [45] is adopted as follows. In advance, the issuer generates signatures for all candidates of u. Assuming ζ_ℓ, that is the maximum of $|U \cap V_\ell|$, the set of the candidates is $\Phi = \{u = \sum_{\ell=1}^{L} \delta_\ell c_\ell | 1 \leq \delta_\ell \leq \zeta_\ell \text{ for all } 1 \leq \ell \leq L\}$. Namely, the issuer generates signatures for all elements of this set, which are included in the public key. In the attribute proof protocol, the user proves the knowledge of a signature of u generated in the accumulator, where the GS proof brings the zero-knowledge, i.e., u is secret.

However, this trick brings the communication costs to the system. The size $|\Phi|$ is $\prod_{1 \leq \ell \leq L} \zeta_\ell$. The public key size increases by $|\Phi|$ signatures.

5.3 Proposed Extended System to Reduce Public-Key Size

In our previous system, the bottleneck was the large public key size. To overcome this problem, we will extend this system. The idea behind the extension is to separate the candidates of the set Φ into two sets of Φ_1 and Φ_2. We consider $\Phi_1 = \{u_1 = \sum_{\ell=1}^{\tilde{L}} \delta_\ell c_\ell | 1 \leq \delta_\ell \leq \zeta_\ell \text{ for all } 1 \leq \ell \leq \tilde{L}\}$ and $\Phi_2 = \{u_2 = \sum_{\ell=\tilde{L}+1}^{L} \delta_\ell c_\ell | 1 \leq \delta_\ell \leq \zeta_\ell \text{ for all } \tilde{L}+1 \leq \ell \leq L\}$, where $\tilde{L}$ is set such that $|\Phi_1| \simeq |\Phi_2|$. If all the values of ζ_ℓ are equal, then $\tilde{L}$ is set such that $\tilde{L} = \lceil \frac{L}{2} \rceil$. Then, the issuer generates signatures for all the candidates of sets Φ_1 and Φ_2, where the signatures are generated using the public key for Φ_1 and the key for Φ_2 independently. In the attribute proof protocol, the user proves the knowledge of the signatures on both u_1 and u_2, used in the accumulator verification in the GS proof. The accumulator is verified by the following pairing equation:

$$\frac{e(P_U, acc_{\mathcal{V}})}{e(g, W)} = z^{u_1} \cdot z^{u_2} = z^{u_1+u_2}.$$

Since the signatures on u_1 and u_2 ensure $u_1 = |U \cap V_1|c_1 + \cdots + |U \cap V_{\tilde{L}}|c_{\tilde{L}}$ and $u_2 = |U \cap V_{\tilde{L}+1}|c_{\tilde{L}+1} + \cdots + |U \cap V_L|c_L$, the above relations imply equation (5.1). Thus the accumulator relation is correct. The size of $|\Phi_1|$ is $\prod_{1 \le \ell \le \tilde{L}} \zeta_\ell$ and that of $|\Phi_2|$ is $\prod_{\tilde{L}+1 \le \ell \le L} \zeta_\ell$, which greatly reduces the size of signatures. Therefore, the public key size is also greatly reduced than the previous system.

5.3.1 Proposed Construction of Extended System

The details of the construction of the extended system are described as follows:

IssuerKeyGen

n, L, η, and ζ_ℓ are given for all $1 \le \ell \le L$.

1. Select bilinear groups $\mathcal{G}$ and $\mathcal{T}$ with the same order p and the bilinear map e, and $g \in_R \mathcal{G}$.

2. Generate the public parameters of the extended accumulator: for all $1 \le \ell \le L$, compute $c_\ell = (\eta + 1)^{\ell - 1}$ and set $\mathcal{C} = (c_1, \ldots, c_L)$. Select $\gamma \in_R Z_p$, and compute

$$\begin{aligned} pk_{\text{acc}} = (\mathcal{C}, g_1 = g^{\gamma^1}, \ldots, g_n = g^{\gamma^n}, \\ g_{n+2} = g^{\gamma^{n+2}}, \ldots, g_{2n} = g^{\gamma^{2n}}, \\ z = (g, g)^{\gamma^{n+1}}). \end{aligned}$$

3. Generate two key pairs for the AHO signature:

$$\begin{aligned} pk_{\text{AHO}}^{(d)} &= (G_r^{(d)}, H_r^{(d)}, G_z^{(d)}, H_z^{(d)}, G^{(d)}, H^{(d)}, A^{(d)}, B^{(d)}), \\ sk_{\text{AHO}}^{(d)} &= (\alpha_a^{(d)}, \alpha_b^{(d)}, \mu_z^{(d)}, \nu_z^{(d)}, \mu, \nu), \end{aligned}$$

where $d \in \{0, 1\}$.

4. Generate a CRS for the GS NIWI proof: select $\vec{f} = (\vec{f_1}, \vec{f_2}, \vec{f_3})$, where $\vec{f_1} = (f_1, 1, g)$, $\vec{f_2} = (1, f_2, g)$, $\vec{f_3} = \vec{f_1}^{\xi_1} \cdot \vec{f_2}^{\xi_2}$ for $f_1, f_2 \in_R \mathcal{G}$ and $\xi_1, \xi_2 \in_R Z_p^*$.

5. For $\mathcal{C}$ and $\tilde{L} = \lceil \frac{L}{2} \rceil$, define $\Phi_1 = \{u_1 = \sum_{\ell=1}^{\tilde{L}} \delta_\ell c_\ell | 1 \le \delta_\ell \le \zeta_\ell \text{ for all } 1 \le \ell \le \tilde{L}\}$ and $\Phi_2 = \{u_2 = \sum_{\ell=\tilde{L}+1}^{L} \delta_\ell c_\ell | 1 \le \delta_\ell \le \zeta_\ell \text{ for all } \tilde{L}+1 \le \ell \le L\}$, where $|\Phi_1| = \prod_{1 \le \ell \le \tilde{L}} \zeta_\ell$ and $|\Phi_2| = \prod_{\tilde{L}+1 \le \ell \le L} \zeta_\ell$. For every $u_1 \in \Phi_1$ and $u_2 \in \Phi_2$, generate the AHO signatures on $g_1^{u_1}$ and $g_1^{u_2}$. The signatures are denoted as $\tilde{\sigma}_{u_1} = (\tilde{\theta}_{u_1 1}, \ldots, \tilde{\theta}_{u_1 7})$ and $\tilde{\sigma}_{u_2} = (\tilde{\theta}_{u_2 1}, \ldots, \tilde{\theta}_{u_2 7})$ respectively.

6. Output the issuer public key

$$\begin{aligned} ipk = (p, \mathcal{G}, \mathcal{T}, e, g, pk_{\text{acc}}^{(0)}, pk_{\text{acc}}^{(1)}, pk_{\text{AHO}}, \vec{f}, \{\tilde{\sigma}_{u_1}\}_{u_1 \in \Phi_1}, \\ \{\tilde{\sigma}_{u_2}\}_{u_2 \in \Phi_2}) \end{aligned}$$

and the issuer secret key $isk = (sk_{\text{AHO}}^{(0)}, sk_{\text{AHO}}^{(1)})$.

CertObtain

This is an interactive protocol between **CertObtain-$\mathcal{U}_k$** (user) and **CertObtain-$\mathcal{I}$** (issuer). The common inputs of this protocol consist of ipk and U_k that are the indexes of attribute values of the user. The input of **CertObtain-$\mathcal{I}$** is isk. We introduce a special attribute value a_{SP}. Every user has a_{SP}.

1. **CertObtain-$\mathcal{I}$**: Generate $P_k = \prod_{i \in U_k} g_i$.

2. **CertObtain-$\mathcal{I}$**: Using $sk_{\mathrm{AHO}}^{(1)}$, generate an AHO signature $\sigma_k = (\theta_1, \ldots, \theta_7)$ on message P_k. Return σ_k to **CertObtain-$\mathcal{U}_k$** as the certificate.

3. **CertObtain-$\mathcal{U}_k$**: Compute $P_k = \prod_{i \in U_k} g_i$, and verify the AHO signature σ_k on P_k. Output $cert_k = (P_k, \sigma_k)$.

ProofGen

The inputs are ipk, U_k, $cert_k$, and the CNF formula Ψ. For the given formula $\Psi = (a_{11} \vee a_{12} \vee \cdots) \wedge (a_{21} \vee a_{22} \vee \cdots) \wedge \cdots (a_{L'1} \vee a_{L'2} \vee \cdots)$ with $a_{11}, a_{12}, \ldots, a_{21}, a_{22}, \ldots \in \{1, \ldots, n\}$, define $V_1 = \{a_{11}, a_{12}, \ldots\}, V_2 = \{a_{21}, a_{22}, \ldots\}, V_l = \{a_{l1}, a_{l2}, \ldots\}$. If $L' < L$, define $V_{L'+1} = \cdots = V_L = \{a_{\mathrm{SP}}\}$.

1. Compute the accumulator:

$$acc_{\mathcal{V}} = \prod_{1 \le \ell \le L} (\prod_{j \in V_\ell} g_{n+1-j})^{c_\ell}.$$

2. Compute the witness:

$$W_{\mathcal{V}} = \prod_{i \in U_k} \prod_{1 \le \ell \le L} (\prod_{j \in V_\ell}^{j \neq i} g_{n+1-j+i})^{c_\ell}$$

   U_k satisfies $\mathcal{V}$ for $acc_{\mathcal{V}}$, and sets $u_1 = \delta_1 c_1 + \ldots + \delta_{\tilde{L}} c_{\tilde{L}}$ and $u_2 = \delta_{\tilde{L}+1} c_{\tilde{L}+1} + \ldots + \delta_L c_L$, where $\delta_\ell = |U_k \cap V_\ell|$ for all $1 \le \ell \le L$.

3. Set $\tau_{u_1} = g_1^{u_1}$ and $\tau_{u_2} = g_1^{u_2}$. From ipk, select the AHO signatures $\tilde{\sigma}_{u_1} = (\tilde{\theta}_{u_1 1}, \ldots, \tilde{\theta}_{u_1 7})$ on the $g_1^{u_1}$ and $\tilde{\sigma}_{u_2} = (\tilde{\theta}_{u_2 1}, \ldots, \tilde{\theta}_{u_2 7})$ on the $g_1^{u_2}$.

4. Compute the GS commitments $com_{P_k}, com_{W_{\mathcal{V}}}, com_{\tau_{u_1}}, com_{\tau_{u_2}}$ to $P_k, W_{\mathcal{V}}, \tau_{u_1}, \tau_{u_2}$. Then, re-randomize the AHO signature σ_k to obtain $\sigma'_k = \{\theta'_1, \ldots, \theta'_7\}$, and compute GS commitments $\{com_{\theta'_i}\}_{i \in \{1,2,5\}}$ to $\{\theta'_i\}_{i \in \{1,2,5\}}$. Similarly, re-randomize the AHO signature $\tilde{\sigma}_{u_1}$ to obtain $\tilde{\sigma}'_{u_1} = \{\tilde{\theta'}_{u_1 1}, \ldots, \tilde{\theta'}_{u_1 7}\}$, and $\tilde{\sigma}_{u_2}$ to obtain $\tilde{\sigma}'_{u_2} = \{\tilde{\theta'}_{u_2 1}, \ldots, \tilde{\theta'}_{u_2 7}\}$,and compute GS commitments $\{com_{\tilde{\theta'}_{u_1 i}}\}_{i \in \{1,2,5\}}$ to $\{\tilde{\theta'}_{u_1 i}\}_{i \in \{1,2,5\}}$ and $\{com_{\tilde{\theta'}_{u_2 i}}\}_{i \in \{1,2,5\}}$ to $\{\tilde{\theta'}_{u_2 i}\}_{i \in \{1,2,5\}}$.

5. Generate the GS proofs $\{\pi_i\}_{i=1}^{9}$ s.t.

$$\begin{aligned}
1_{\mathcal{T}} = {} & e(P_k, acc_{\mathcal{V}}) \cdot e(g, W_{\mathcal{V}})^{-1} \\
& \qquad \cdot e(\tau_{u_1}, g_n)^{-1} \cdot e(\tau_{u_2}, g_n)^{-1}, && (5.2) \\
A^{(1)} & \cdot e(\theta'_3, \theta'_4)^{-1} \\
& \qquad = e(G_z^{(1)}, \theta'_1) \cdot e(G_r^{(1)}, \theta'_2) \cdot e(G^{(1)}, P_k), && (5.3) \\
B^{(1)} & \cdot e(\theta'_6, \theta'_7)^{-1} \\
& \qquad = e(H_z^{(1)}, \theta'_1) \cdot e(H_r^{(1)}, \theta'_5) \cdot e(H^{(1)}, P_k), && (5.4) \\
A^{(0)} & \cdot e(\tilde{\theta}'_{u_1 3}, \tilde{\theta}'_{u_1 4})^{-1} \\
& \qquad = e(G_z^{(0)}, \tilde{\theta}'_{u_1 1}) \cdot e(G_r^{(0)}, \tilde{\theta}'_{u_1 2}) \cdot e(G^{(0)}, \tau_{u_1}), && (5.5) \\
B^{(0)} & \cdot e(\tilde{\theta}'_{u_1 6}, \tilde{\theta}'_{u_1 7})^{-1} \\
& \qquad = e(H_z^{(0)}, \tilde{\theta}'_{u_1 1}) \cdot e(H_r^{(0)}, \tilde{\theta}'_{u_1 5}) \cdot e(H^{(0)}, \tau_{u_1}), && (5.6) \\
A^{(1)} & \cdot e(\tilde{\theta}'_{u_2 3}, \tilde{\theta}'_{u_2 4})^{-1} \\
& \qquad = e(G_z^{(0)}, \tilde{\theta}'_{u_2 1}) \cdot e(G_r^{(0)}, \tilde{\theta}'_{u_2 2}) \cdot e(G^{(0)}, \tau_{u_2}), && (5.7) \\
B^{(1)} & \cdot e(\tilde{\theta}'_{u_2 6}, \tilde{\theta}'_{u_2 7})^{-1} \\
& \qquad = e(H_z^{(0)}, \tilde{\theta}'_{u_2 1}) \cdot e(H_r^{(0)}, \tilde{\theta}'_{u_2 5}) \cdot e(H^{(0)}, \tau_{u_2}). && (5.8)
\end{aligned}$$

6. Output $\sigma = (\{\theta'_i\}_{i=3,4,6,7}, \{\tilde{\theta}'_{u_1 i}\}_{i=3,4,6,7}, \{\tilde{\theta}'_{u_2 i}\}_{i=3,4,6,7},\ com_{P_k}, com_{W_{\mathcal{V}}}, com_{\tau_{u_1}}, com_{\tau_{u_1}}, \{com_{\theta'_i}\}_{i=1,2,5}, \{com_{\tilde{\theta}'_{u_1 i}}\}_{i=1,2,5}, \{com_{\tilde{\theta}'_{u_2 i}}\}_{i=1,2,5}, \{\pi_i\}_{i=1}^{9})$.

The equation (5.2) shows one of verification relations of accumulator:

$$\frac{e(\prod_{i \in U_k} g_i, acc_{\mathcal{V}})}{e(g, W_{\mathcal{V}})} = e(g_1^{u_1} . g_1^{u_2}, g_n) = z^{u_1 + u_2},$$

where $P_k = \prod_{i \in U_k} g_i$, $\tau_{u_1} = g_1^{u_1}$ and $\tau_{u_2} = g_1^{u_2}$. Equations (5.3), (5.4) show the knowledge of the AHO signature of P_k, i.e., the certificate $cert_k$. Equations (5.5), (5.6) show the knowledge of the AHO signature of τ_{u_1}. The equations (5.7), (5.8) show the knowledge of the AHO signature of τ_{u_2}. This ensures that $1 \leq \ \delta_\ell \leq \zeta$ where $u_1 = \delta_1 c_1 + \ldots + \delta_{\tilde{L}} c_{\tilde{L}}$ and $u_2 = \delta_{\tilde{L}+1} c_{\tilde{L}+1} + \ldots + \delta_L c_L$. Thus, together with equation (5.2), it ensures the verification of the accumulator. This is why the verifier is ensured that $U_k \cap V_\ell \neq \emptyset$, i.e, attributes in U_k satisfies the CNF formula Ψ.

Verify

The inputs are ipk, the proof σ, and the CNF formula Ψ.

1. Compute the accumulator $acc_{\mathcal{V}}$, as in **ProofGen**.

2. Accept σ, if the verifications of all GS proofs $\{\pi_i\}_{i=1}^{9}$ are successful.

5.3.2 Further Extension

As a further extension, more separations of the candidates of set Φ can be considered. If the elements of set Φ is separated into more than two, the public key size is supposed to be more smaller than the current. But the overhead can increase the computational cost by about 20% $\sim$ 30% than the current system. Still, the computational time will remain practical in the current general PCs.

Table 5.1: Environments of implementation and experiments.

CPU	Intel Core2 Duo (3GHz)
Memory	3.9 GBytes
OS	Ubuntu 12.04 (kernel Linux-3.2.0-59-generic-pae)
Compiler	GCC-4.5.2
Library	GMP-5.0.2 (Multiple Precision Arithmetic Library) ELiPS (Pairing Library)

Table 5.2: Experimental Results for equal values of ζ_ℓ.

	Previous System	Extended System
Public-Key Size	31 MB	0.3 MB
Proving Time	125 ms	151 ms
Verification Time	340 ms	479 ms
Proof Size	3.9 KB	5.6 KB

5.4 Implementation and Experiments

As in Chapter 4, we utilize the fast pairing library for the implementation.

5.4.1 Experimental results and evaluations

The environments of the implementation and experiments are shown in Table 5.1. In this scheme, our goal is to reduce the public key size. On the other hand, in the extended system, the attribute proof needs additional computational cost. Hence, to prove our goal and to show the practicality of our system, we measured the processing time of the attribute proof protocol and the data size in the implemented system.

We suppose the eID application as in Chapter 4. Similarly, we can consider the popular example of the following CNF formula:

$$gender = male \wedge birth\_year \in \{1900, \ldots, 1992\}$$

$$\wedge profession \in \{student, teacher, professor, \ldots\} \wedge \cdots .$$

By the OR relation of *birth_year*, the user proves that he is adult, but the concrete age is concealed. In the measurements, we evaluate the dependency on ζ_ℓ (the size of $|U \cap V_\ell|$) in case of the same value of ζ_ℓ and different values of ζ_ℓ. Thus, we performed two types of measurements: *Measurement 1*, and *Measurement 2*.

5.4.2 Measurement 1

In measurement 1, we measured the processing time and data size of **Attribute Proof** protocol for the user and verifier, in case of the same value of ζ_ℓ (the size of $|U \cap V_\ell|$). In this measurement, we set $L = 15$, $|V_\ell| = 10$, $\zeta_\ell = 2$ for all l, and $|U| = 16$.

Table 5.2 shows the experimental results of the public key size, the processing time of proving and verification in the attribute proof protocol, and the proof size in the measurement

Table 5.3: Experimental Results for unbalanced values of ζ_ℓ.

	Previous System	Extended System
Public-Key Size	14 MB	0.4 MB
Proving Time	149 ms	179 ms
Verification Time	335 ms	483 ms
Proof Size	3.9 KB	5.6 KB

1. From this table, we can confirm that the public key size is greatly reduced. The public key size of our previous system is 31 MB, the extended system needs only 0.3 MB. Compared to $|\Phi| = 2^{15} = 32,768$, $|\Phi_1| = 2^8 = 256$ and $|\Phi_2| = 2^7 = 128$. The extended system reduces the public-key size by $2\sqrt{N}$, where N is the number of signatures in previous system. From Table 5.2, we can find that the proving time, the verification time and the proof size are increased by about 20%, compared to those of the previous system, which is a trade off to reduce the public key size. This is because, in our extended system, we use two proofs of signatures for set Φ_1 and Φ_2. Previously, it was only one for the set Φ. Hence, the processing time and the proof size have increased by about 20% than the previous system.

However, still the proving and verification times are only 179 ms and 479 ms respectively and they are sufficiently practical in the current general PCs.

5.4.3 Measurement 2

In measurement 2, we measured the processing time and data size of **Attribute Proof** protocol in case of the unbalanced values of ζ_ℓ (the size of $|U \cap V_\ell|$). In this measurement, we set $L = 20$, $|V_\ell| = 10$, $\zeta_1 = 5$, $\zeta_2 = 10$, $\zeta_3 = 15$, $\zeta_4 = 20$, $\zeta_5 = 1, \ldots, \zeta_{20} = 1$, and $|U| = 21$. In this case, $\tilde{L} = 2$, and $|\Phi_1| = 50$ and $|\Phi_2| = 300$, compared to $|\Phi| = 15,000$.

Table 5.3 shows the experimental results of the measurement 2. From this table, we can confirm that the public key size is also greatly reduced in the unbalanced case. The public key size of our previous system is 14 MB, whereas the extended system needs only 0.4 MB. From Table 5.3, we can find that the proving time, the verification time and the proof size are increased by about 20%, compared to those of the previous system. However, still the processing time is sufficiently practical in the current general PCs.

5.5 Summary

In this chapter, we proposed an extension of an anonymous credential system with the constant-size proofs for CNF formulas to reduce the long public key. The experimental result shows that the public key size is reduced to $2\sqrt{N}$ for the original number of signatures of N. The computational costs are increased by about 20%, which we considered as a trade off in order to reduce the public key size.

Chapter 6

Offline/Online Technique for Efficiency Improvement

6.1 Introduction

In our proposed system in Chapter 3, a user can prove a CNF formula on the certified attributes anonymously. The eID application of the system can be used in mobile environments, and thus the efficiency of clients is important. In the previous chapters, we target the case with lots of OR relations. On the other hand, in this chapter, we consider the case with lots of AND relations. This is because, in some real application services, formulas with lots of AND relations are used. For instance, when utilizing user's address as the attribute, it consists of hierarchical names such as the country, the city, the prefecture, the street and so on, as shown in Fig. 6.1. In such a case, we need to consider formulas with lots of AND relations such as:

$$address = japan \wedge okayama \wedge okayama - shi \wedge tsushima \cdots$$

.

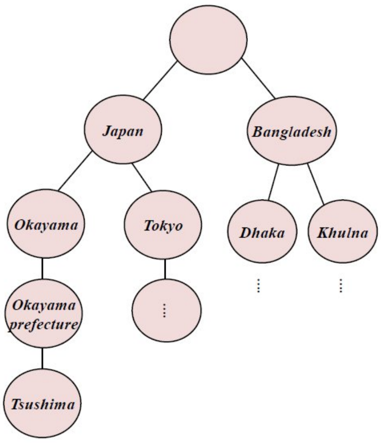

Figure 6.1: Hierarchy of address attributes.

In our accumulator, the number of exponentiations depends on the number of AND relations. In mobile applications, the exponentiation cost may cause a delay. Therefore, to reduce the computational overhead, we propose a system with a less online computation for the client in the case of lots of AND relations. Some parts of computations are executed offline in advance. The compensation is the increase of the storage to store the offline computed values.

6.2 Proposed Construction

We introduce an online/offline technique using the offline precomputation of some public parameters to reduce the amount of the online computation. In the precomputation, the exponentiations of all patterns used for the accumulator, $acc_{\mathcal{V}}$ and witness, $W_{\mathcal{V}}$ computations are offline executed in advance.

In the accumulator and witness computations, the computations of $(g_{n+1-j})^{c_\ell}$ are needed. In this improved system, a separated offline algorithm computes all the parameters of $g_j^{c_\ell}$ for $1 \leq j \leq 2n$ except $j = n+1$ and for $1 \leq l \leq L$ in advance. Then, the computed $g_j^{c_\ell} s$ are stored in a file. In **ProofGen**, the signer can access the file and can use the parameter $g_j^{c_\ell}$ to generate the $acc_{\mathcal{V}}$ and $W_{\mathcal{V}}$ values without an exponentiation cost. In the basic system of Chapter 3, the computations of $acc_{\mathcal{V}}$ and $W_{\mathcal{V}}$ require L exponentiations of c_l, which brings a processing cost. By the precomputation, the computations of $acc_{\mathcal{V}}$ and $W_{\mathcal{V}}$ need only multiplications depending on the size of CNF clauses, $|V_l|$ and the number of OR clauses in proved CNF formula, L. To compute $g_j^{c_\ell}$, we do not need any secret parameters, and thus this offline precomputation can be executed by any party. When the precomputations are executed in the issuer, the precomputed values $g_j^{c_\ell}$ are distributed to the user as public parameters.

The construction of our proposed online/offline system is similar to our basic system in Chapter 3, except the offline public parameter computation. Hence, we omit the detailed construction of the same protocol here. This construction is given bellow.

6.2.1 IssuerKeyGen

The construction of this protocol is the same as that in our basic system in Chapter 3. n, L, η_ℓ, and ζ_ℓ for all $1 \leq \ell \leq L$ are given. This algorithm executes **AccSetup** to obtain the public parameters of the extended accumulator, generates key pairs of AHO signatures, generates CRS for GS NIWI proof, and prepares AHO signatures on ${g_1}^u$ for all $u \in \{\sum_{\ell=1}^{L} \delta_\ell c_\ell | 1 \leq \delta_\ell \leq \zeta_\ell \text{ for all } 1 \leq \ell \leq L\}$.

6.2.2 CertObtain

This is an interactive protocol between **CertObtain-$\mathcal{U}_k$** (user) and **CertObtain-$\mathcal{I}$** (issuer). The common inputs of this protocol consist of ipk and U_k that are the indexes of attribute values of the user. The input of **CertObtain-$\mathcal{I}$** is isk. In this protocol, to the user, the issuer sends a certificate $cert_k$ including the AHO signature σ_k on $P_k = \prod_{i \in U_k} g_i$. The construction and mechanishm of this protocol is also the same as our basic system in Chapter 3.

6.2.3 Offline Precomputation

This algorithm, given ipk, computes all the parameters with exponentiations for the accumulator and witness. The inputs of this algorithm are n, L, g_j, h_j and c_ℓ. This algorithm generates $g_j^{c_\ell}$ and $h_j^{c_\ell}$ for all $1 \leq j \leq 2n$ except $j = n+1$ and for all $1 \leq l \leq L$ and outputs $\Omega_1 = g_j^{c_\ell}$ and $\Omega_2 = h_j^{c_\ell}$. In **ProofGen**, this precomputed parameters Ω_1 and Ω_2 are used to compute the accumulator and witness value.

6.2.4 ProofGen

In this algorithm, the inputs are ipk, U_k, $cert_k$, the CNF formula Ψ, and the precomputed Ω_1, Ω_2. For a given formula $\Psi = (a_{11} \vee a_{12} \vee \cdots) \wedge (a_{21} \vee a_{22} \vee \cdots) \wedge \cdots (a_{L'1} \vee a_{L'2} \vee \cdots)$ with $a_{11}, a_{12}, \ldots, a_{21}, a_{22}, \ldots \in \{1, \ldots, n\}$, define $V_1 = \{a_{11}, a_{12}, \ldots\}, V_2 = \{a_{21}, a_{22}, \ldots\}, \ldots, V_l = \{a_{l1}, a_{l2}, \ldots\}$. If $L' < L$, define $V_{L'+1} = \cdots = V_L = \{a_{\mathrm{SP}}\}$. This algorithm generates the GS NIWI proof proving that P_k satisfies the accumulator verification for the accumulator $acc_{\mathcal{V}}$ indicating the proved predicate Ψ, and proving P_k is signed as a AHO signature σ_k by the issuer's public key. In the accumulator verification, the GS proof for an AHO signature on g_1^u is also utilized.

Here, step 1 and step 2 are modified. The rests from step 3 are the same as in Chapter 3. Using precomputed Ω_1 and Ω_2, step 1 and step 2 are as follows:

1. Compute the accumulator:

$$acc_{\mathcal{V}} = \prod_{1 \leq \ell \leq L} \big(\prod_{j \in V_\ell} g_{l,n+1-j}\big)$$

2. Compute the witness

$$W_{\mathcal{V}} = \prod_{i \in U_k} \prod_{1 \leq \ell \leq L} \big(\prod_{j \in V_\ell}^{j \neq i} g_{l,n+1-j+i}\big)$$

   that U_k satisfies $\mathcal{V}$ for $acc_{\mathcal{V}}$, and sets $u = \delta_1 c_1 + \ldots + \delta_L c_L$, where $\delta_\ell = |U_k \cap V_\ell|$ for all $1 \leq \ell \leq L$.

This technique completely reduces the exponentiation costs of the **ProofGen** protocol in online scheme, since only the multiplication cost is needed. Hence, this scheme is much more efficient.

6.2.5 Verify

The construction of this protocol is the same as that in our basic system in Chapter 3.

6.3 Implementation and Experimental Evaluations

To evaluate the effectiveness of our proposed improvement, we implemented our scheme. The environments of the implementation and experiments are shown in Table 6.1.

Our target is to reduce the computational cost of the online computations in **ProofGen** protocol. On the other hand, the extended system needs some overhead storage. Hence, to show the practicality of our system, we measured the processing times of the **ProofGen** and **Verify** and the data sizes of the precomputed Ω_1 and Ω_2 in the implemented system.

Table 6.1: Environments of implementation and experiments.

CPU	Intel Core2 Duo (3GHz)
Memory	3.9 GBytes
OS	Ubuntu 12.04 (kernel Linux-3.2.0-59-generic-pae)
Compiler	GCC-4.5.2
Library	GMP-5.0.2 (Multiple Precision Arithmetic Library) ELiPS (Pairing Library)

In this chapter, we suppose the same eID application in Chapter 1. In this application, we can consider an example of the following CNF formula:

$$gender = male \wedge birth\_year \in \{1900, \ldots, 1992\}$$

$$\wedge profession \in \{student, teacher, professor, \ldots\}$$

$$\wedge address \in \{japan \wedge okayama \wedge okayama - shi \wedge tsushima \ldots\} \wedge \ldots$$

This example means that a user's proved CNF formula may consist of many AND relations. In the experiment, we evaluate the dependency on the size of the proved CNF formula L, $10 \leq L \leq 50$. For the other parameters we set $|V_\ell| = 10$ for the half of $V_\ell' s$ and $|V_\ell| = 1$ for the others. We set $\zeta_\ell = 1$ for all l and consider $n = 231$ as the total number of attributes value. In this measurement, for simplicity, we set $|U| \geq L$ which is a popular setting for the eID application, and concretely set $|U| = L + 1$.

6.3.1 Processing Time

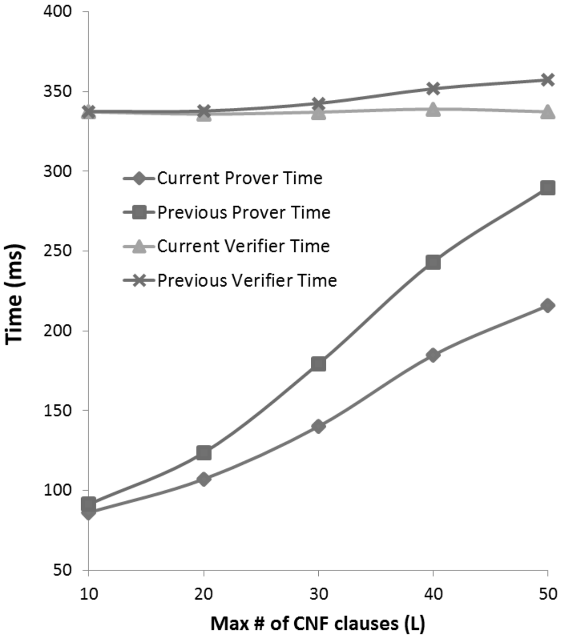

Figure 6.2: Processing times of **ProofGen** and **Verify**.

In our experiment, we measured the computational time of **ProofGen** and **Verify**. After measuring the processing time, we compare it with our previous system in Chapter 3. The comparisons are shown in Fig. 6.2. This figure shows that the prover/user time in **ProofGen** is significantly reduced than that of our previous system in Chapter 3. On the other hand, the verification time also decreases slightly. This result of the greatly reduced **ProofGen** time is very meaningful in the mobile environment.

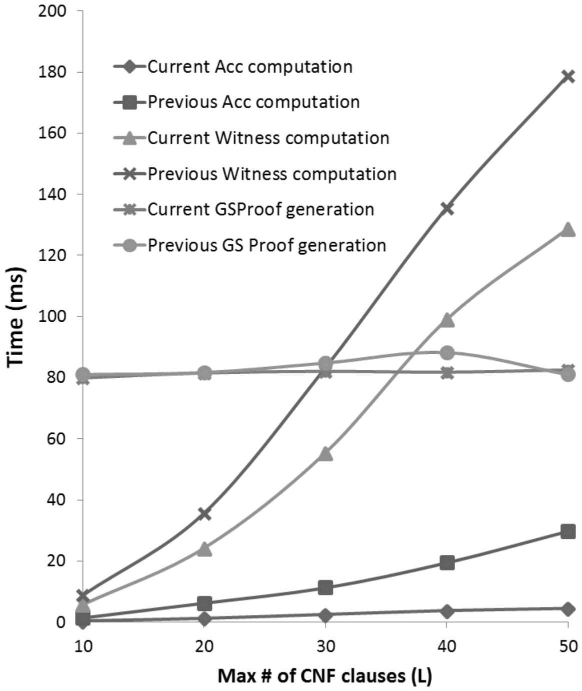

Figure 6.3: Comparison of **detailed processing times** in **Prover**.

To examine the effectiveness of the online/offline technique, we measured the detailed processing time of **ProofGen** as shown in Fig. 6.3. We can confirm the accumulator computation and witness computation are greatly reduced from the previous system in Chapter 3. The GS proof generation time is the same of our basic system.

6.3.2 Data Size

Table 6.2: Size of precomputed values.

L	$\Omega_1 + \Omega_2$
10	224 KB
20	888 KB
30	2008 KB
40	3584 KB
50	4710 KB

We measured the size of the precomputed values Ω_1 and Ω_2 as shown in Table 6.2. In case of $L = 50$, the size is more than 4 MB. However, in the current smartphones, the storage cost of more than 4 MB is not so serious. Thus, in such a mobile environment, our improvemnet is sufficiently practical.

6.4 Summary

In this chapter, we proposed and implemented the online/offline technique to reduce the computational costs of **ProofGen**. By precomputing all candidates of $g_{n+1-j}^{c_\ell}$, exponentiations in the online accumulator and witness computations are excluded. The experimental result shows that the computational costs are greatly reduced than our previous system in Chapter 3. The demerit is the storage cost. However the current small mobile devices have huge storage space. Hence, the demerit of storage is not so serious in the current mobile situation.

Chapter 7

Conclusion and Future Works

In this dissertation, we studied the privacy-enhancing attribute authentication systems with faster authentication times.

Firstly, we proposed an anonymous credential system with the constant-size proofs of CNF formulas. Using our extended pairing-based accumulator, the proof generation cost is more efficient than the system in [6], since only multiplications depending on the number of literals are needed. The compensation is the increase of public parameters. We demonstrated that, for CNF formulas that can be often used in eID applications, this increase is not so huge.

Secondly, to show the practicality of our system, using the fast pairing library, we implemented the system and measured the computational process times. The experimental result shows that the proof generation time and verification time depend on the size of CNF formula. Even for the size of $|V_\ell| = 100{,}000$, the proof generation time is only 228 ms and the verification time is only 371 ms. This indicates that our system is sufficiently practical.

Thirdly, we proposed an extension of an anonymous credential system with the constant-size proofs for CNF formulas to reduce the public key size. The experimental result shows that the public key size is reduced to $2\sqrt{N}$ for the original size N, although the computational costs are increased by about 20%, which we consider as a trade-off to reduce the public key size.

Finally, we proposed and implemented the online/offline precomputation technique to reduce the online computational costs of the proof generation in case of lots of AND relations in the proved CNF formulas. The experimental result shows that the computational costs are greatly reduced than our system in Chapter 3. The demerit is the storage cost for lots of precomputed values. However, since the current small mobile devices have sufficiently large storage, the requirement of storage is not so serious.

One future work is to propose a system allowing proofs beyond CNF formulas. Although our proposed system focuses only on the CNF formulas, in some real applications, we may need some other logical relations beyond CNF formulas, such as monotone relations or even negations. Another future work is the implementation of our system on smart phones such as Android devices.

Bibliography

[1] L. Guo, C. Zhang, J. Sun, and Y. Fang, "PAAS: A privacy-preserving attribute-based authentication system for ehealth networks," Proc. Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on, pp. 224-233. IEEE, 2012.

[2] J. Camenisch and A. Lysyanskaya, "Dynamic accumulators and application to efficient revocation of anonymous credentials," Advances in Cryptology — CRYPTO 2002, LNCS 2442, pp.61–76, Springer–Verlag, 2002.

[3] J. Camenisch and T. Groß, "Efficient attributes for anonymous credentials," Proc. ACM Conference on Computer and Communications Security 2008 (ACM-CCS'08), pp.345–356, 2008.

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

[5] A. Sudarsono, T. Nakanishi, and N. Funabiki, "Efficient proofs of attributes in pairing-based anonymous credential system," Proc. 11th Privacy Enhancing Technologies Symposium (PETS 2011), LNCS 6794, pp.246–263, Springer–Verlag, 2011.

[6] M. Izabachène, B. Libert, and D. Vergnaud, "Block-wise p-signatures and non-interactive anonymous credentials with efficient attributes," Proc. 13th IMA Conference on Cryptography and Coding (IMACC 2011), LNCS 7089, pp.431–450, Springer–Verlag, 2011.

[7] P. Bichsel, J. Camenisch, T. Groß, and V. Shoup, "Anonymous credentials on a standard java card," Proc. ACM Conference on Computer and Communications Security 2009 (ACM-CCS'09), pp.600–610, 2009.

[8] S. Galbraith, K. Paterson, and N. Smart, "Pairings for cryptographers," Technical report. 2006/165, p.165, IACR ePrint archive, 2006.

[9] E. Lee, H. Lee, and C. Park, "Efficient and generalized pairing computation on abelian varieties," IEEE Trans. Information Theory, Vol. 55, Issue 4, pp. 1793–1803, 2009.

[10] F. Vercauteren, "Optimal Pairings," Cryptology ePrint archive, 2008/096.pdf.

[11] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," Advances in Cryptology - EUROCRYPT 2008, LNCS 4965, pp.146–162, Springer–Verlag, 2008.

[12] N. Begum, T. Nakanishi, and N. Funabiki, "Efficient proofs for cnf formulas on attributes in pairing-based anonymous credential system," Proc. 15th Annual International Conference on Information Security and Cryptology, 2012 (ICISC 2012), 2012.

[13] N. Begum, T. Nakanishi, and N. Funabiki, "Implementation and evaluation of an pairing-based anonymous credential system with constant-size proofs and efficient proof generations," Proc. The Third International Workshop on Advance in Networking and Computing, 2012 (WANC 2012), pp.264–268, 2012.

[14] N. Begum, T. Nakanishi and N. Funabiki, "Efficient proofs for CNF formulas on attributes in pairing-based anonymous credential system," IEICE Trans. Fundamentals, vol.E96-A, no.12, pp.2422–2433, 2013.

[15] A. Sudarsono, T. Nakanishi, and N. Funabiki, "Efficient proofs of attributes in anonymous credential systems using a pairing-based accumulator," Proc. Computer Security Symposium (CSS2010), pp. 801–806, 2010.

[16] M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, and M. Ohkubo, "Structure-preserving signatures and commitments to group elements," Advances in Cryptology - CRYPTO 2010, LNCS 6223, pp.209–236, Springer–Verlag, 2010.

[17] B. Libert, T. Peters, and M. Yung, "Scalable group signatures with revocation," Advances in Cryptology - EUROCRYPT 2012, LNCS 7323, pp.609–627, Springer–Verlag, 2012.

[18] J. Groth and A. Sahai, "Efficient non-interactive proof systems for bilinear groups," Advances in Cryptology - EUROCRYPT 2008, LNCS 4965, pp.415–432, Springer–Verlag, 2008.

[19] M. Akane, Y. Nogami, and Y. Morikawa, "Fast Ate pairing computation of embedding degree 12 using subfield-twisted eliptic curve," IEICE Trans. Fundamentals, vol.E92-A, no.2, pp.508–516, 2009.

[20] M. Akane, H. Kato, T. Okimoto, Y. Nogami, and Y. Morikawa, "Efficient parameters for ate pairing computation with Barreto-Naehrig curve," Proc. Computer Security Symposium (CSS 2007), pp. 495–500, 2007.

[21] H. Cohen and G. Frey, "Handbook of elliptic and hyperelliptic curve cryptography," Chapman & Hall/CRC, 2005.

[22] S. Galbraith, K. Paterson and N. Smart, "Pairing for cryptographers," Discrete Applied Mathematics, 6, pp. 3113–3121, 2008.

[23] S. Chatterjee, D. Hankerson, E. Knapp, and A. Menezes, "Comparing two pairing-based aggregate signature schemes," preprint. Available at http://eprint.iacr.org/2009/060.

[24] GNU multiple precision arithmetic library (GMP), http://gmplib.org/.

[25] P. S. L. M. Barreto and M. Naehrig, "Pairing-friendly elliptic curves of prime order," Proc. Selected areas in cryptography (SAC 2005), Springer Berlin Heidelberg, LNCS 3897, pp. 319–331, 2006.

[26] R. Granger and N.P. Smart, “On computing products of pairings,” Cryptology ePrint Archieve: Report 2006/172.

[27] Y. Nogami, Y. Sakemi, T. Okimoto, K. Nekado, M. Akane, and Y. Morikawa, “Scalar multiplication using frobenius expansion over twisted elliptic curve for ate pairing based cryptography,” IEICE Trans. Fundamentals of electronics, communications and computer sciences, Vol. E92-A, no.1, pp. 182–189, 2009.

[28] Y. Nogami, M. Akane, Y. Sakemi, H. Kato, and Y. Morikawa, “Integer variable χ-based ate pairing,” Proc. 2nd International Conference on Pairing-based Cryptography 2008, LNCS 5209, pp. 178–191, 2008.

[29] Y. Sakemi, Y. Nogami, K. Okeya, H. Kato, and Y. Morikawa, “Skew frobenius map and efficient scalar multiplication for pairing-based cryptography” Proc. 7th International Conference Cryptology and Network Security, CANS 2008, LNCS 5339, pp. 226–239, 2008.

[30] D. Boneh, X. Boyen, and H. Shacham, “Short group signatures,” Advances in Cryptology — CRYPTO 2004, LNCS 3152, pp.41–55, Springer–Verlag, 2004.

[31] M. Abe, K. Haralambiev, and M. Ohkubo, “Signing on elements in bilinear groups for modular protocol design.” Cryptology ePrint Archive, Report 2010/133, 2010. http://eprint.iacr.org/.

[32] D. Boneh and H. Shacham, “Group signatures with verifier-local revocation,” Proc. ACM-CCS 2004, pp. 167–177, 2004.

[33] D. Chaum and E. Van Heijst, “Group signatures,” Proc. EUROCRYPT 1991, LNCS 547, pp. 241–246, Springer-Verlag, 1991.

[34] G. Tsudik and S. Xu, “Accumulating composites and improved group signing,” Proc. ASIACRYPT 2003, LNCS 2894, pp. 269–286, Springer-Verlag, 2003.

[35] J. Camenisch and E.V. Herreweghen, “Design and implementation of the idemix anonymous credential system,” Proc. 9th ACM Conference on Computer and Communications Security (ACM-CCS 2002), pp. 21–30, 2002.

[36] J. Camenisch and J. Groth, “Group signature: better efficiency and new theoretical aspects,” Proc. SCN 2004, LNCS 3352, pp.120–133, Springer Verlag, 2004.

[37] J. Camenisch and M. Michels, “A group signature scheme based on an RSA-variant,” Proc. ASIACRYPT 1998, LNCS 1514, Springer Verlag. pp. 160–174, 1998.

[38] J. Furukawa and H. Imai, “An efficient group signature scheme from bilinear maps,” Proc. ACIPS 2005, LNCS 3574, pp. 455–467, Springer-Verlag, 2005.

[39] T. Isshiki, K. Mori, K. Sako, I. Teranishi, and S. Yonezawa, “Using group signatures for identity management and its implementation,” Proc. ACM workshop on Digital identity management (ACM-DIM 2006), pp. 73–78, 2006.

[40] T. Nakanishi and N. Funabiki, “Verifier-local revocation group signature scheme with backward unlinkability from bilinear maps,” Proc. Advances in Cryptology-Asiacrypt (ASIACRYPT 2005), LNCS 3788, pp. 533–548, Springer Verlag, 2005.

[41] T. Nakanishi and N. Funabiki, “A short verifier-local revocation group signature scheme with backward unlinkability,” Proc. 1st International Workshop on Security (IWSEC 2006), LNCS 4266, pp. 17–32, Springer Verlag, 2006.

[42] T. Nakanishi and N. Funabiki, “Short verifier-local revocation group signature scheme with backward unlinkability,” IEICE Trans. Fundamentals of electronics, communications and computer sciences, vol. E90-A, no. 9, pp. 1793–1802, 2007.

[43] T. Nakanishi, N. Hamada, T. Nakayama, and N. Funabiki, “Group signature schemes with efficient membership revocation using small primes,” Proc. 8th International Workshop on Information Security Applications (WISA 2007), pp. 411–426, 2007.

[44] D. Boneh and X. Boyen, “Short group signatures without random oracles,” Advances in Cryptography - EUROCRYPT 2004, LNCS 3027, pp. 56–73, 2004.

[45] J. Camenisch, R. Chaabouni, and A. Shelat, “Efficient protocols for set membership and range proofs,” Advances in Cryptology — ASIACRYPT 2008, LNCS 5350, pp.234–252, Springer–Verlag, 2008.