2017

Contributions to Identity-Based Broadcast Encryption and Its Anonymity

Jianchang Lai
University of Wollongong

UNIVERSITY
OF WOLLONGONG
AUSTRALIA

Contributions to Identity-Based Broadcast Encryption and Its Anonymity

Jianchang Lai

Supervisor:
Prof. Yi Mu
Co-supervisor:
Dr. Fuchun Guo

This thesis is presented as required for the conferral of the degree:

Doctor of Philosophy

The University of Wollongong
School of Computing and Information Technology

May 2017

Declaration

I, Jianchang Lai, declare that this thesis submitted in fulfilment of the requirements for the conferral of the degree Doctor of Philosophy, from the University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. This document has not been submitted for qualifications at any other academic institution.

Jianchang Lai
May 20, 2018

Abstract

Broadcast encryption was introduced to improve the efficiency of encryption when a message should be sent to or shared with a group of users. Only the legitimate users chosen in the encryption phase are able to retrieve the message. The primary challenge in constructing a broadcast encryption scheme is to achieve collusion resistance such that the unchosen users learn nothing about the content of the encrypted message even they collude.

Revocation is an important issue of broadcast encryption. In the identity-based revocation system, the encryption algorithm takes the identities of revoked users as input, instead of the identities of selected users who are allowed to decrypt the ciphertext in the broadcast encryption so that the revoked users cannot obtain the message. This kind of revocation system can deal with the situation where some of the receivers' private keys are leaked or compromised in a broadcast encryption system, and should be revoked in the future broadcast. While a recipient revocable identity-based broadcast encryption scheme introduced by Susilo et al. allows a third party to revoke some receivers from the identity set stated in the original broadcast ciphertext without the knowledge of the encrypted message. This notion has been showed that it is still expressive enough for practical scenarios.

Anonymity in the broadcast encryption has been considered as an important property, since the receiver might be unwilling to expose its identity information in some applications. In this thesis, we further study the identity-based broadcast encryption (IBBE) and mainly focus on how to anonymously revoke the receivers from the ciphertext generated by a broadcast encryption scheme without knowing the encrypted message, and how to protect user privacy including the privacy of revoked users. Aiming to protect the user privacy, we use the technique of Lagrange polynomial interpolation to hide the users' identities when performing the message (data) encryption. In this research topic, we propose the first anonymous revocable identity-based broadcast encryption scheme, where the user revocation process does not require knowing the identity information of the receivers and the encrypted message. As our second result, we present a fully privacy-preserving revocable identity-based broadcast encryption scheme, where both the identity information of the receivers and the revoked users are protected. To improve the efficiency when

the number of the revoked users is large, we propose an authorization scheme as our third result, which also can achieve the anonymity of all receivers. This authorization scheme can be viewed as the re-allocation of decryption rights of receivers in the identity-based broadcast encryption.

In order to solve the all-or-noting affair in the broadcast encryption and meet the requirements of new applications, we consider a variant of identity-based broadcast encryption and introduce a new notion of identity-based broadcast encryption for inner products (IBBE-IP for short), where the message encryption is replaced by inner product encryption. The IBBE-IP can further protect the confidentiality of the encrypted data compared to the IBBE and allows the encryptor to control who are permitted to obtain the decryption result. It is useful in the context of descriptive statistics. More specifically, in the IBBE-IP, the user's private key is associated with a pair of a user identity and a vector $(ID, \vec{y})$. The user with a private key of $(ID, \vec{y})$ can decrypt the encrypted vector (message) $\vec{x}$ for an identity set S selected by the encryptor and learn the inner product $\langle \vec{x}, \vec{y} \rangle$ if and only if $ID \in S$ and nothing else. In this thesis, we present a construction of IBBE-IP by combining the technique of identity-based broadcast encryption and inner protect encryption. The proposed scheme achieves constant size private keys and supports unbounded private key queries issued by the adversary.

Acknowledgments

This research work presented in this thesis would never be possible without the support of many individuals.

First of all, I would like to express my sincere gratitude to my supervisor, Professor *Yi Mu*, for his instructive advice and useful suggestions on my thesis. Professor *Mu* is an extremely excellent supervisor who has offered me guidance and advice both in my academic study and daily life during my Ph.D. period. Many thanks to him for his support, for his encouragement, patience, motivation and immense knowledge. I would like to thank my co-supervisor, Dr *Fuchun Guo*, who had a tremendous influence on my research. I really appreciate his valuable training which helped me build up my knowledge in cryptography. I am deeply grateful for his help in the completion of this thesis.

I would like to give my special thanks to Professor *Willy Susilo* for his indispensable contribution to our joint research and his helpful advice. I also thank *Guoming Yang, Xinyi Huang, Man Ho Au, Futai Zhang, Jinguang Han, Mingwu Zhang, and Sha Ma* for their invaluable advice on my research. Discussions with them are always stimulating and rewarding.

I am fortunate to have many great colleagues and friends during my Ph.D. study in Wollongong. Many thanks for their support and kindness. The non-exhausted list includes *Peng Jiang, Rongmao Chen, Yizhen Wu, Hui Cui, Weiwei Liu, Min Xiao, Jiannan Wei, Jongkil Kim, Yinhao Jiang, Nan Li, Tran Viet Xuan Phuong, Fatemeh Rezaeibagha, Yangguang Tian, Shengmin Xu, Shiwei Zhang, Zhongyuan Yao, Ge Wu, Tong Wu, Xueqiao Liu, Zhen Zhao, Nan Yan, Yanwei Zhou, Bingrui Zhu, Xiaoguo Li, Ke Huang, Hongyun Zhang, Xiaoyu Yu, Jing Zhang, Zewei Ding,* etc. Even assuming that I did not forget people to whom that I want to express my appreciations, the list of people I am indebted to is far more extensive. It certainly includes all staff from School of Computing and Information Technology.

Finally, and most importantly, I want to thank my parents, my sister and my girlfriend *Xinyan Yang* for their patience, understanding, encouragement and love throughout my Ph.D. study. Without their support, all my achievements would never be possible.

Publications

During my PhD study in University of Wollongong, I wrote and published the following papers. This thesis is based on some of the following published papers and the accepted paper.

1. Jianchang Lai, Yi Mu, Fuchun Guo, Rongmao Chen, Sha Ma. Efficient k-out-of-n oblivious transfer scheme with the ideal communication cost. Theoretical Computer Science, 714: 15-26 (2018)

2. Jianchang Lai, Yi Mu, Fuchun Guo, and Rongmao Chen. Fully Privacy-Preserving ID-Based Broadcast Encryption with Authorization. The Computer Journal, 60(12): 1809-1821 (2017).

3. Jianchang Lai, Yi Mu, Fuchun Guo. Efficient Identity-Based Online/Offline Encryption and Signcryption with Short Ciphertext. International Journal of Information Security, 16(3):299-311, 2017.

4. Jianchang Lai, Yi Mu, Fuchun Guo, Willy Susilo, and Rongmao Chen. Fully Privacy-Preserving and Revocable ID-Based Broadcast Encryption for Data Access Control in Smart City. Personal and Ubiquitous Computing, 21(5):855-868, 2017.

5. Jianchang Lai, Yi Mu, Fuchun Guo, Willy Susilo, and Rongmao Chen. Anonymous Identity-Based Broadcast Encryption with Revocation for File Sharing. In Joseph K. Liu and Ron Steinfeld, editors, Information Security and Privacy - 21st Australasian Conference, ACISP 2016, volume 9723 of LNCS, pages 223-239. Springer, 2016.

6. Jianchang Lai, Yi Mu, Fuchun Guo, and Willy Susilo. Improved Identity-Based Online/Offline Encryption. In Ernest Foo and Douglas Stebila, editors, Information Security and Private-20th Australasian Conference, ACISP 2015, volume 9144 of LNCS, pages 160-173. Springer, 2015.

7. Jianchang Lai, Yi Mu, Fuchun Guo, Peng Jiang, and Sha Ma. Identity-Based Broadcast Encryption for Inner products. The computer Journal, acceptance data: 16 May 2018.

I am thankful to have opportunities to collaborate with others in other cryptographic areas. The contributions are listed below and they are beyond the scope of this thesis.

1. Peng Jiang, Fuchun Guo, Kaitai Liang, Jianchang Lai, Qiaoyan Wen. Searchain: Blockchain-based Private Keyword Search in Decentralized Storage. Future Generation Computer Systems, to appear, 2017. Acceptance Date: 18 August 2017.

2. Fuchun Guo, Rongmao Chen, Willy Susilo, Jianchang Lai, Guomin Yang, and Yi Mu. Optimal Security Reductions for Unique Signatures: Bypassing Impossibilities with A Counterexample. In Jonathan Katz and Hovav Shacham, editors, CRYPTO 2017, volume 10402 of LNCS, pages 517-547. Springer, 2017.

3. Fuchun Guo, Willy Susilo, Yi Mu, Rongmao Chen, Jianchang Lai, and Guomin Yang. Iterated random oracle: A Universal Approach for Finding Loss in Security Reduction. In Jung Hee Cheon and Tsuyoshi Takagi, editors, ASIACRYPT 2016, volume 10032 of LNCS, pages 745-776, 2016.

4. Peng Jiang, Xiaofen Wang, Jianchang Lai, Fuchun Guo, Rongmao Chen. Oblivious Keyword Search with Authorization. In Liqun Chen and Jinguang Han, editors, Provable Security - 10th International Conference, ProvSec 2016, volume 10005 of LNCS, pages 173-190, 2016.

5. Peng Jiang, Yi Mu, Fuchun Guo, Xiaofen Wang, and Jianchang Lai. Centralized Keyword Search on Encrypted Data for Cloud Applications. Security and Communication Networks, 9(18):5064-5084, 2016.

Contents

List of Tables

Chapter 1

Introduction

1.1 Background

In 1976, Diffie and Hellman [DH76] introduced the notion of public key cryptography, which has become one of the greatest revolutions in the history of cryptography. In the public key cryptography, the key used for encrypting the message is different from the one used in the decryption phase and we call them the public key and private key respectively. If a user named Bob wants to send a message to another user named Alice, Bob just encrypts the message using Alice's public key which is publicly known and generated from Alice's private key known by Alice only. After receiving the encrypted message, Alice uses her private key to decrypt it and retrieves the message. As the public key is publicly known, it arises a problem: how can Bob believe the received public key is really Alice's? Therefore, to securely transmit a message in the traditional public key encryption (PKE), it requires a trusted third party, called trusted authority (TA), to generate a certificate of public key for each user. The certificate can be used to verify the validity of the user's public key. As the TA needs to issue the certificate for each user, the certificate management is usually very complex and costly.

To address this issue appeared in the traditional public key cryptography, Shamir introduced the notion of identity-based encryption (IBE) [Sha84] in 1984. In the IBE system, the user's public key can be an arbitrary string binding its identity, such as an email address or a telephone number. The corresponding private key is generated by binding the identity with a system master key known by the private key generator (PKG). In such a system, there are four algorithms: (1) Setup run by the PKG takes a security parameter as input and generates the system master public key and the master secret key which is only known by itself, (2) KeyGen run by the PKG generates a private key for the user with a particular identity by using the master secret key, (3) Encrypt run by a sender allows it to encrypt a message to a specified identity, and (4) Decrypt run by the receiver allows it to decrypt the encrypted message by providing a private key for the corresponding identity.

IBE provides a simple way for certificate management. Considering the above

example, Bob can send an encrypted message to Alice at alice@iacr.org by using the string "alice@iacr.org". In this system, there is no requirement that Bob has to obtain Alice's public key and verify it first. Alice could decrypt the message using a private key corresponding to "alice@iacr.org" received from the PKG. Inevitably, in this solution, Alice needs to authenticate her identity to the PKG. Alice can do this authentication in the same way as in the traditional public key cryptography.

The IBE system provides a useful and efficient way to securely share a message with someone whose identity is known. When a sender wants to send a message to a group of users, it can trivially repeat the encryption scheme for each receiver independently. However, this trivial solution is too inefficient to be of practical use especially when the number of receivers is large. The ciphertext size and the computational cost are linear in the number of receivers. Therefore, this trivial solution is impractical and not suitable for the situation when a message should be sent to several users.

Aiming to improve the efficiency when a message to be sent (shared) to (with) a group of users, Fiat and Naor [FN94] introduced the notion of broadcast encryption (BE), which allows an encryptor to broadcast a message to a group of users via a public channel. In an identity-based broadcast encryption (IBBE) scheme, a message is encrypted under a set of identities selected by the encryptor in a way that only those users can decrypt the encrypted message and learn the content by providing their private keys. While the users who are not chosen in the encryption phase learn nothing about the message even they collude. IBBE (or BE) has been widely deployed in the real-life applications, such as in Pay TV, and has been extensively studied to capture more properties. However, as the encrypted message in the IBBE system can be decrypted by several users, in some scenarios, the receiver might not willing to expose its identity information to others. For instance, when a user subscribes some sensitive TV programs, the user might be unwilling any other users to know that he/she has subscribed the programs. The receiver privacy-preserving (anonymity) has received more and more attention in many practical cryptography fields, and has been extensively studied in the IBBE.

When some of receivers have left the system in the BE or their private keys have been leaked, we would like to revoke them from the future broadcasts. Simply encrypting the new message under the a new receiver set using the same broadcast encryption scheme might lead to a large computational cost in message encryption if the number of revoked users is small. To address this situation, Naor and Pinkas [NP00] introduced a novel technique for broadcast encryption, where the encryption is performed under the revoked users instead of the receivers such that any receiver can retrieve the encrypted message except the revoked users. This kind of broadcast encryption is also viewed as revocation system. The first revocation system with

small private keys in the identity-based setting was proposed by Lewko, Sahai and Waters [LSW10]. The revocation system is particularly useful for the situations where we would like to revoke some of receivers in the broadcast encryption from the future broadcasts.

In the IBBE, once the receivers' identities have been determined and used to encrypt the message, we cannot revoke some of them. This might restrict its deployment as in some cases, the decryption right of receivers might be relocated by other entities. To addresses this issue, Susilo et al. [SCG+16] extended IBBE and introduced a new notion of recipient revocable identity-based broadcast encryption (RR-IBBE). In the RR-IBBE system, it allows a third party to remove some of identities (users) from the identity set stated in the original IBBE ciphertext without knowing the encrypted message. The third party is unable to decrypt the ciphertext but it is permitted to revoke some of the receivers.

The aforementioned encryption notions, including the traditional public key encryption, identity-based encryption, identity-based broadcast encryption and recipient revocable identity-based broadcast encryption, provide data confidentiality by encrypting a message under the public key or identity of an intended receiver, who is the only person that is allowed to decrypt the ciphertext using the corresponding private key (or in the multi-user setting). In these systems, the decryption is an all-or-nothing affair, namely, a receiver is able to retrieve either the entire message or nothing. In order to satisfy some new application scenarios, functional encryption [BSW11] was introduced as a generalization of the PKE. In a functional encryption system, the amount of information which is revealed to the receiver from a given ciphertext is finely controlled. In a nutshell, given an encrypted message x and a private key sk_F associated with a value y over a function F, it allows the key holder to learn the value of $F(x, y)$ and nothing else. It perfectly overcomes the all-or-nothing affair appeared in the PKE system.

Inner product encryption (IPE) as a special functional encryption recently introduced by Abdalla et al. [ABCP15] considers the inner product functionality and aims to compute the actual value of inner product via decryption, which is entirely different from the predicate encryption [SSW09], which checks whether the inner product is zero or not, and retrieves the corresponding encrypted message if so. In an IPE scheme, each message is described as a vector. A ciphertext CT is created under a message vector $\vec{x}$, and a user with a private key of the vector $\vec{y}$ from the same space of the message is allowed to learn the value of $\langle \vec{x}, \vec{y} \rangle$ via decryption and nothing else about the message $\vec{x}$. IPE is useful in the context of descriptive statistics. For example, it can be used to compute the weighted mean of a collection of data without leaking the contents of data. As the encrypted message in the IPE can be decrypted by at most $n - 1$ private keys with the message length n, IPE can

also be viewed as a broadcast encryption with different decryption results.

1.2 Related Work

The work of Diffie and Hellman in [DH76] is a milestone of public key cryptography. The first public key scheme presented in [DH76] is for secure secret key exchange but not a general-purpose encryption algorithm. Rivest, Shamir and Adleman proposed the so-called RSA scheme [RSA78] in 1978. RSA scheme has become the most widely accepted and implemented general-purpose method to public key encryption. Its security is based on the hardness of prime factorization of the large number. Later, Rabin proposed the Rabin cryptosystem [Rab80], which is the first asymmetric cryptosystem where recovering the message from a ciphertext can be proved to be as hard as factoring. ElGamal, based on the Diffie-Hellman key exchange, proposed the ElGamal encryption scheme [Gam84] in 1984. Although the ElGamal encryption scheme has been shown insecure against malleable attacks, it has been widely used as a building block to construct many cryptosystems. Cramer and Shoup [CS98] extended the ElGamal system and proposed a scheme which can deal with the malleable attack appeared in the ElGamal encryption. The proposed scheme is the first efficient scheme proven to be secure against adaptive chosen-ciphertext attacks (CCA) under the standard cryptographic assumption.

1.2.1 Identity-Based Encryption

The notion of identity-based encryption (IBE) was introduced by Shamir [Sha84] in 1984, but the first two concrete IBE schemes were realized about twenty years later and proposed by Cocks [Coc01], Boneh and Franklin [BF01] in 2001 respectively. The Cocks' scheme uses the technique of quadratic residues and its security is based on the hardness of the integer factorization problem. While Boneh and Franklin [BF01] proposed a paring based IBE scheme (BF-IBE for short). BF-IBE scheme is the first IBE scheme with a security proof in a well-formulated model under the random oracle which has been regarded as a heuristic method. BF-IBE scheme has received much attention from researchers since the authors proposed it. In comparison, Cocks' system encrypts the message bit-by-bit and consequently outputs long ciphertexts, which is somewhat harder to use in practice than the BF-IBE scheme. Subsequently, significant research effort has been devoted to realizing efficient and secure IBE schemes.

Boneh and Boyen solved the open problem proposed in BF-IBE, and presented the first IBE scheme without random oracles and it was proved to be secure against chosen-plaintext attacks (CPA). This scheme is not very practical and mostly serves

as an existence proof. Later, the authors proposed two efficient schemes shown to be secure without using the random oracle, but in a weaker security model known as selective identity model (selective-ID, for short) [BB04]. The study of Waters in [Wat05] aimed to improve the efficiency and achieve higher security, and presented the first efficient IBE scheme which was fully secure (against adaptive identity attacks) without random oracles under the DBDH assumption. The first practical identity-based encryption scheme without random oracles is given by Gentry [Gen06] in 2006 (Gentry IBE, for short). The Gentry IBE scheme has several advantages over previous such systems, namely computational efficiency, shorter public parameters and a tighter security reduction. Further researches along this line mostly are based on the above IBE systems. Another research line of IBE focuses on the construction based on the learning-with-errors assumption [GPV08, CHKP10].

Horwitz and Lynn [HL02] suggested that the users are no longer identified by a single identity, but a tuple of identities which contain the identities of their ancestors in the hierarchy. The authors then introduced the notion of hierarchical IBE (HIBE). Based on the scheme in [BF01], Gentry and Silverberg [GS02] presented a full functional HIBE scheme with n-level hierarchy. Boneh and Boyen also gave a selective identity secure HIBE scheme without random oracles by extending their first scheme described in [BB04]. Boneh et al. [BBG05] presented a HIBE system with constant size ciphertext which is regardless of the hierarchy depth. Their scheme has been proved to be selective-ID secure in the standard model and fully secure in the random oracle model. Several following HIBE schemes are known based on the bilinear map [SW08, Wat09, GH09, LW10].

1.2.2 Broadcast Encryption

Broadcast encryption (BE) introduced by Fiat and Naor [FN94] aimed to efficiently send a message to a group of users. The primary challenge in the BE is to achieve collusion resistance, where the users who are not chosen in the encryption phase cannot retrieve the message even they collude. The proposed scheme in [FN94] is only secure against bounded collusions. Naor et al. [NNL01] proposed a fully collusion resistant broadcast encryption scheme for all but a small set of revoked users. The first full collusion resistant broadcast encryption scheme with constant size secret keys and ciphertexts is given by Boneh, Gentry and Waters [BGW05] in 2005. Subsequent work [DF03a, BGW05, DPP07, Del07, GW09, PPSS12, BWZ14] have proposed broadcast encryption systems with different properties. They mainly focused on reducing public key sizes, private key sizes, ciphertext sizes and computational costs for encryption and decryption. The first two broadcast encryption schemes in the identity-based setting have been realized in [SF07, Del07] in their

independent work, where the ciphertext is generated using receivers' identities instead of their public keys. The proposed schemes in both work achieve constant size ciphertext and private keys.

Anonymous Broadcast Encryption.

In the aforementioned broadcast encryption schemes, the receivers' identities (public key information) must be attached to the ciphertext and taken as input to perform the decryption algorithm. This definitely exposes the privacy of receivers. In other words, the public knows the identities of receivers from the ciphertext as the ciphertext is transmitted over a public channel. This might not be desirable for some applications. For example, in the TV-subscription system, when a customer subscribes some sensitive programs, he/she is usually unwilling other users to know his/her subscription. Anonymity is another important research area of broadcast encryption.

The first work addressing the anonymity in broadcast encryption appears in [BBW06]. The authors presented the notion of private broadcast encryption to protect the identities of receivers and gave a generic construction from any key indistinguishable against chosen-ciphertext attacks scheme, which achieves receiver anonymity and CCA security. Boneh, Sahai and Waters [BSW06] extended this notion to construct private linear broadcast encryption and proposed a fully collusion resistant traitor tracing scheme with sublinear size ciphertexts and constant size private keys.

Libert, Paterson and Quaglia [LPQ12] examined the security of the number-theoretic construction in [BBW06] and suggested the proof techniques without the random oracle. The authors then proposed an anonymous broadcast encryption scheme which achieves adaptive security without random oracles. The size of ciphertext in their schemes is linear in the number of receivers and the security depends on a one-time signature. Later, Fazio and Perera [FP12] formalized the notion of outsider-anonymous broadcast encryption. Their construction achieves sublinear size ciphertext but fails to obtain anonymity among receivers.

The work of Kiayias and Samari [KS13] aims to study the lower bounds for the ciphertext size of private broadcast encryption. They showed that an atomic private broadcast encryption scheme with fully anonymous must have a ciphertext size of $\Omega(n \cdot \lambda)$, where n is the number of identities selected in the encryption and λ is a security parameter. Fazio, Nicolosi and Perera [FNP14] studied the broadcast steganography and introduced a new construction called outsider-anonymous broadcast encryption with pseudorandom ciphertexts, which achieves sublinear size ciphertext and is secure without random oracles.

Multi-Receiver Encryption. Another approach to share a message among a

group of users is called multi-receiver encryption. The concept of multi-receiver public key encryption was formalized by Bellare, Boldyreva and Micali in [BBM00]. Their main result is that the security in the multi-receiver setting can be reduced to the security of public key encryption in the single-receiver setting. Later, based on ElGamal encryption, Kurosawa [Kur02] proposed an efficient multi-receiver public key encryption scheme by using the technique of "randomness re-use".

Baek, Naini and Susilo [BSS05] proposed the notion of multi-receiver identity-based encryption (MR-IBE) in PKC 2005 and gave the corresponding formal definition and security model. Comparing to simply re-encrypting a message n times for n receivers using BF-IBE [BF01], their scheme only needs one pairing computation to encrypt a single message. Fan, Huang and Ho [FHH10] proposed the first anonymous multi-receiver identity-based encryption scheme by using the technique of Lagrange interpolating polynomial mechanism. Unfortunately, it has been pointed out in [Chi12] that the scheme cannot protect the receiver privacy. The work in [PPT13] aims to deal with multiple messages simultaneously. The authors proposed multi-channel broadcast encryption schemes for pay-TV and used the dummy-help technique to prove the security. However, Phan et al.'s schemes [PPT13] suffer from the problems that the decryption has to take into account the public keys of all users in all sets, and cannot protect the receivers' identities from being exposed. It has been showed that any multi-receiver public key encryption scheme can be transferred into the corresponding broadcast encryption scheme.

1.2.3 Revocation

Revocation system is a variant of broadcast encryption, which takes a set of revoked users as input to the encryption algorithm in the way that the revoked users cannot decrypt the ciphertext anymore. We can view the revocation system as a negative analogue of broadcast encryption. When some of receivers' private keys are compromised in one broadcast encryption system, this kind of revocation can prevent these users from retrieving the future broadcasts. Similar to the broadcast encryption, the primary challenge in revocation system is to achieve full collusion resilience. Several elegant revocation constructions [NP00, NNL01, DF03b, GST04, BW06, LSW10, LKLP14] have been proposed.

Generally, there are three main techniques to construct revocation systems. The first technique called subset-cover framework was proposed by Naor, Naor and Lotspiech [NNL01]. Based on this framework, they proposed the first stateless tree-based revocation scheme which is secure against a collision of any number of users. Later, this method has been improved by Halevy and Shamir [HS02], and Goodrich, Sun and Tamassia [GST04] respectively, to achieve shorter ciphertext size

and private key size. The second type of techniques was introduced by Kurosawa and Desmedt [KD98] and Naor and Pinkas [NP00], which uses polynomial interpolation. However, the constructions based on this technique suffer from that both the secret key size and ciphertext size are either linear in the number of revoked users or linear in the maximal number of revoked users. The third technique to construct revocation schemes uses exponent-inversion technique introduced by Delerablée, Paillier and Pointcheval [DPP07], which can achieve either constant size secret keys or constant size ciphertexts.

Boneh and Waters [BW06] introduced a primitive called augmented broadcast encryption which is claimed to be sufficient for constructing trace and revoke schemes. The authors proposed a revocation scheme with sublinear size ciphertexts and private keys. The scheme is proved to be secure against the adaptive adversary. Lewko, Sahai and Waters [LSW10] proposed a revocation system in the identity-based setting (IBRS), which achieves constant size master public key and private keys using secret sharing and the "two equation" technique. The size of ciphertext in Lewko et al.'s scheme is linear in the number of revoked users. Subsequently, IBRS schemes with constant-size ciphertexts are proposed by Attrapadung and Libert in [AL10] and Attrapadung, Libert and Panafieu in [ALdP11] respectively. However, the size of both the master public key and private keys in their schemes is linear in the maximal number of revoked identities.

Lee et al. [LKLP14] presented a single revocation encryption (SRE) scheme, which allows a sender to broadcast a message to a group of selected users and one group user is revoked. Any group member can decrypt the ciphertext except the revoked user. The authors then proposed a public key trace and revoke scheme by combining the layered subset difference scheme and their SRE scheme. We note that among these schemes, the revocation list is determined by the encryptor.

Revocable encryption is a notion similar to the revocation system, which uses key update to revoke users, and the revocation list is maintained by the key authority (PKG) who issues the user's private keys. In a revocable identity-based encryption scheme [BGK08], a user with identity ID is given a long-term private key sk_{ID} from the key authority. For each time period T, the key authority broadcasts key update information ku_T using the revocation list, and the user with identity ID can generate a short-term decryption key $dk_{ID,T}$ by using sk_{ID} and ku_T if and only if ID is not a revoked identity. The user with the decryption key $dk_{ID,T}$ is able to decrypt the ciphertexts created in the time T. Significant research effort has been devoted to realizing revocable IBE schemes [AI09, LV09, SE13a, SE13b, PLL15]. Another notion called self-updatable encryption [LCL+13] uses ciphertext updating to revoke users. Each private key and ciphertext are associated with a time T' and T respectively. Only the private key with $T' \geq T$ can decrypt the ciphertext

successfully.

1.2.4 Inner product Encryption

Abdalla et at. [ABCP15] showed that the inner product functionality for very simple and efficient realizations can be constructed from the DDH assumption. They proposed the first IPE scheme with a selective security. Subsequently, Agrawal et al. in [ALS16] improved the work of Abdalla et al. and presented a construction which is provably secure against adaptive attacks without compromising the efficiency. Bishop et al. [BJK15] took the first step forward towards exploring the possibility of obtaining the IPE with function privacy using the efficient and well-studied primitives in the private-key setting[1]. They presented a function hiding IPE construction from asymmetric bilinear pairing groups. Their construction supports any polynomial number of private key queries and encryption queries in the full-hiding security model [BS15] and they derived its security from the SXDH assumption. Datta et al. [DDM16] improved Bishop et al.'s work by constructing a simple and efficient function private IPE scheme. Compared with the work in [BJK15], their construction achieves the strongest notion (indistinguishability-based) of function privacy in the private-key setting. Later, Kim et al. [KLM$^+$16] reduced the parameter sizes and the run-time complexity of the work in both [BJK15] and [DDM16].

Goldwasser et al. [GGG$^+$14] introduced the notion of multi-input functional encryption (MIFE), where decryption keys are associated with functions of several inputs and the decryption algorithm takes multiple ciphertexts as input. Lee and Lee [LL16] presented the first two-input IPE scheme from composite-order bilinear groups in the private-key setting, which achieves selective IND-security. In an independent work, Kim et al. [KLM$^+$16] showed how function-private IPE directly yields the single-key two-input function encryption for general functions over a small message space. The first multi-input IPE scheme was proposed by Abdalla et al. in [ARW16, AGRW17]. They showed how to realize n slots MIFE for the inner product for any polynomial number n under standard assumptions. The resulting scheme avoids the exponential security loss. Benhamouda et al. [BBL17] moved a step further to achieve security against chosen-ciphertext attacks (CCA). They gave a generic construction of IND-CCA IPE from projective hash functions with homomorphic properties and presented several instantiations based on different assumptions.

[1]In the private-key IPE, the encryption key is the master secret key which is used to generate the decryption key.

1.3 Problem Statements

In the IBBE, once the identities of receivers have been decided and used in the message encryption, we cannot revoke some of them before the next broadcast, which might be not suitable for some real-life applications. The above revocation system which is a negative analogue of IBBE by encrypting the message under the identities of revoked users can only prevent the revoked users from retrieving the future broadcast message, rather than the encrypted message stated in the IBBE. To address this issue and meet the requirements of new applications, Susilo et al. [SCG+16] combined the revocation together with IBBE, and introduced the notion of recipient revocable identity-based broadcast encryption (RR-IBBE). In the RR-IBBE, it allows a third party to remove some receivers from the original IBBE ciphertext without knowing the encrypted message or performing any decryption operation.

However, we observe that the work of Susilo et al. does not consider the receiver privacy. In order to revoke some receivers stated in the IBBE ciphertext, the original receiver set should be provided to the third party who performs the revocation algorithm. Both the identities of original receivers and the revoked users are attached in the final ciphertext. As stated in the IBBE, receiver anonymity is desirable when designing a scheme for some applications, even for the third party who performs user revocation. The pioneer work of Susilo et al. [SCG+16] cannot preserve the privacy of receivers. The identity information of receivers and the revoked users exposes not only to the third party, but also to the public. In this thesis, we will continue the study of recipient revocable identity-based broadcast encryption and focus on the user privacy-preserving in this system. Additionally, we also consider the negative analogue of RR-IBBE to meet the requirements when the receivers of one encrypted message are decided by different parties. The primary challenge is to achieve collusion resistant.

In the IPE, we observe that with any private key, one can decrypt any encrypted message and obtains the corresponding inner product. Which inner products associated with an encrypted message can be computed are determined by the private keys that have been generated. A wide range of practical applications, however, the encryptor not only wants to further protect the data confidentiality, but also decides who can learn the inner product associated with the encrypted message like in the IBBE. All the previous schemes in the literature have not considered this situation.

Additionally, noting that none of the known IPE constructions in public-key setting supports unbounded private key queries, which has been considered as something inherent to the functionality itself, as the functionality is linear. With n (the length of message) private key queries, the adversary can recover the encrypted

message. Although the works in [BJK15, DDM16] allow the adversary to query arbitrary private keys, both are in the private-key setting, where the encryption key is the master secret key which is used to generate the decryption key. In this thesis, we continue the study of IPE and focus on the possibility to design an IPE scheme in the public-key setting which allows the encryptor to control the decryption results and supports unbounded private key queries issued by the adversary.

1.4 Our Contributions

In this thesis, we further study the broadcast encryption in the identity-based setting and recipient revocable identity-based broadcast encryption. Roughly speaking, based on the above problem statements, we focus on how to achieve user (receiver) anonymity in the recipient revocable identity-based broadcast encryption against both the third party who performs user revocation and the public. Apart from protecting the receiver privacy, we also consider the privacy of the revoked users, which might be desired in some practical applications. If both the receivers' privacy and the revoked users' privacy are protected, we call it fully privacy-preserving in this thesis.

Additionally, we study a variant of recipient revocable identity-based broadcast encryption, namely, identity-based broadcast encryption with authorization. In the authorization system, only the users whose identities are in both the original broadcast identity set and the authorized identity set can retrieve the encrypted message. The authorization algorithm can be performed by several third parties with different authorized identity sets in a way that only the user with identity belonging to the intersection set of the selected identity set and all authorized identity sets can retrieve the encrypted message.

Finally, based on the problem statement in the IPE, we introduce a new notion of identity-based broadcast encryption for inner products (IBBE-IP), where the message encryption is replaced by the inner product encryption. In this system, each private key is associated with an identity and a vector. The encryptor can not only further protect the message confidentiality, but also control who are allowed to learn the inner product associated with the encrypted message like in the IBBE system. We then describe a construction of IBBE-IP which supports unbounded private key queries. In a nutshell, in this thesis, we study the following research objectives.

- The anonymity of receivers in the recipient revocable identity-based broadcast encryption.

- The anonymity of receivers and the revoked users in the recipient revocable

identity-based broadcast encryption. Here, we refer to it as fully anonymous.

– The re-allocation of decryption rights of receivers in the identity-based broadcast encryption or how to improve the efficiency when the number of revoked users is large.

– How to control the decryption rights of users in the inner product encryption system.

1.5 Structure of This Thesis

The thesis is composed of the following seven chapters.

In Chapter 1, we review some backgrounds of encryption and its development from one receiver setting to multi-receiver setting, which helps us understand our work better, including public key encryption, identity-based encryption, broadcast encryption, revocation and inner product encryption. We then review the corresponding related work and show the problem statements, and describe our contributions. In this chapter, we also give the structure of this thesis.

In Chapter 2, we give some mathematical tools towards constructing schemes, including the finite field and cyclic group. Bilinear pairing and some complexity assumptions which the security of schemes presented in this thesis based on are given in this chapter. We give several cryptographic foundations including hash function and random oracle, and the definitions of the public key encryption, identity-based encryption, identity-based broadcast encryption and inner product encryption. We then describe the corresponding security models in terms of each primitive.

In Chapter 3, we put forward an anonymous revocable identity-based broadcast encryption scheme. Our proposed scheme preserves the receiver privacy against the third party who performs the revocation. The first work of revocable identity-based broadcast encryption does not take the receiver privacy into consideration and derived its security under a q-type assumption. The security of our proposed scheme is based on the BDH assumption which is a standard assumption. One limitation is that to successfully perform decryption, the identities of revoked users should be attached to the ciphertext after revocation.

In Chapter 4, we further study the receiver anonymity in recipient revocable IBBE and present a fully privacy-preserving revocable IBBE scheme. The scheme presented in this chapter addresses the limitation of the scheme described in chapter 3. The fully privacy-preserving scheme not only protects the identity information of receivers, but also protects the identity information of the revoked users. The security of the proposed scheme is based on the hardness of the BDH problem in the random oracle model.

In Chapter 5, we consider a negative analogue of revocable identity-based broadcast encryption, namely, authorization. We present a fully privacy-preserving identity-based broadcast encryption with authorization scheme. Compared to the recipient revocable IBBE, it allows a third party to perform an authorization algorithm with an authorized identity set, in a way that only the user belonging to both the identity set stated in the original broadcast ciphertext and the authorized identity set can retrieve the encrypted message. Its security is based on the BDH assumption. The proposed authorization scheme also supports multiple authorizations. The authorization algorithm can be performed several times by using different authorized identity sets such that only the user in all authorized identity sets can retrieve the message by providing the corresponding private keys.

In Chapter 6, we study the inner product encryption proposed by Abdalla et al. [ABCP15] and introduce a notion of identity-based broadcast encryption for inner products (IBBE-IP). The notion of IBBE-IP features both the metrics of identity-based broadcast encryption and the metrics of inner product encryption. In the IBBE-IP, the user private key is associated with its identity and a vector. The decryption only gives the inner product of the encrypted message and the vector associated with the decryption key, which can further protect the encrypted message. Meanwhile, the encryptor can determine who are allowed to learn the inner products as in the broadcast encryption. In this chapter, we give a concrete IBBE-IP construction and derive its security from a q-type assumption in the generic group model.

In Chapter 7, we summarize the contributions of this thesis and give the future work.

Chapter 2

Preliminaries

In this chapter, we review some preliminaries which are used in this thesis, including mathematic tools and cryptographic notions. We begin with a description of some notations appeared in the definitions.

2.1 Notations

Table 2.1 presents some notations which are used throughout this thesis. Some special notations will be defined when they are first used.

If S is a finite set, then $|S|$ is its cardinality. S^n is a set of n-tuples of elements of S. $x \xleftarrow{\$} S$ denotes the assignment to x of an element picked uniformly from S. If A is a probability or stateful algorithm, then $y \leftarrow \mathsf{A}(x)$ denotes the assignment to y of the output of A on input x. Let $\mathbb{N}$ denote the set of natural numbers. A function $\epsilon : \mathbb{N} \rightarrow [0, 1]$ is said to be *negligible* if for every $d \in \mathbb{N}$, there exists a $\lambda_d \in \mathbb{N}$ such that $\epsilon(\lambda) \leq \lambda^{-d}$ for all $\lambda > \lambda_d$.

Table 2.1: Notations

Symbol	Description
λ	System security parameter.
$\vec{x}$	A vector x.
$\mathbb{Z}$	The set of integers.
$\mathbb{Z}_p$	The set consists of the integers modulo p.
$\mathbb{Z}_p^*$	The multiple group of integers modulo p.
$x \in A$	x is a member of set A.
$x \in_R A$	x is a random element chosen from set A.
$x \in A \cap B$	x is a member of both set A and set B.
$x \in A \cup B$	x is a member of either set A or set B.
$x \in A \backslash B$	x is a member of set A but not a member of set B.
$x \in A \triangle B$	x is a member of set A but not a member of set B, or x is a member of set B but not a member of set A.
$\epsilon(\lambda)$	A negligible function associated with λ.

2.2 Mathematical Foundations

In this section, we review some cryptographic mathematical foundations including finite fields, cyclic groups and the Lagrange polynomial interpolation.

2.2.1 Finite Field

Definition 2.1 (Finite Field)**.** *A **finite field** denoted by $(\mathbb{F}, +, *)$ is a set $\mathbb{F}$ containing a finite number of elements together with two binary operations "$+$" (called addition), and "$*$" (called multiplication) defined as follows.*

1. *for all $u, v \in \mathbb{F}$, we have $u + v \in \mathbb{F}$ and $u * v \in \mathbb{F}$;*
2. *for all $u_1, u_2, u_3 \in \mathbb{F}$, we have $u_1 + (u_2 + u_3) = (u_1 + u_2) + u_3$ and $u_1 * (u_2 * u_3) = (u_1 * u_2) * u_3$;*
3. *for all $u, v \in \mathbb{F}$, we have $u + v = v + u$ and $u * v = v * u$;*
4. *there exist $0_{\mathbb{F}}, 1_{\mathbb{F}} \in \mathbb{F}$ (called the **identity** elements) such that for all $u \in \mathbb{F}$, we have $0_{\mathbb{F}} + u = u + 0_{\mathbb{F}} = u$ and $1_{\mathbb{F}} * u = u * 1_{\mathbb{F}} = u$;*
5. *for all $u \in \mathbb{F}$, there exists $-u \in \mathbb{F}$ (called the **additive inverse** of u) such that $u + (-u) = 0_{\mathbb{F}}$;*

6. *for all* $u \in \mathbb{F}$, *there exists* $u^{-1} \in \mathbb{F}$ *(called the* ***multiplicative inverse*** *of* u*) such that* $u * u^{-1} = 1_{\mathbb{F}}$;

7. *for all* $u_1, u_2, v \in \mathbb{F}$, *we have* $(u_1 + u_2) * v = u_1 * v + u_2 * v$.

The symbol $0_{\mathbb{F}}$ is the identity element under the addition operation while the symbol $1_{\mathbb{F}} \in \mathbb{F}$ is the identity element under the multiplication operation. We stress that the binary operations within the finite field definitions are different from the mathematical addition and the mathematical multiplication. Usually, we define $u - v = u + (-v)$ and call it subtraction operation, while define $\frac{u}{v} = u * v^{-1}$ and call it division operation.

In the design of group-based cryptography systems, we usually choose a prime field $\mathbb{F}_q$ with a large prime q. This is the field of residue classes modulo q, and there are q elements in this field which are $\{0, 1, 2, \cdots, q-1\}$. The operations in this field are the modular addition and the modular multiplication. Furthermore,

$$-u = q - u \text{ and } u^{-1} = u^{q-2} \mod q.$$

We use $\mathbb{Z}_q$ to denote the prime field instead of $\mathbb{F}_q$.

2.2.2 Cyclic Group

Definition 2.2 (Abelian Group)**.** *An* ***abelian group*** *denoted by* $(\mathbb{G}, *)$ *is a set* $\mathbb{G}$ *together with a binary operation "* $*$ *" defined as follows.*

(i) for all $u_1, u_2, u_3 \in \mathbb{G}$, *we have* $u_1 * (u_2 * u_3) = (u_1 * u_2) * u_3$ *(i.e. "* $*$ *" is associative);*

(ii) there exists $1_{\mathbb{G}} \in \mathbb{G}$ *(called the* ***identity*** *element) such that for all* $u \in \mathbb{G}$, *we have* $1_{\mathbb{G}} * u = u * 1_{\mathbb{G}} = u$;

(iii) for all $u \in \mathbb{G}$, *there exists* $u^{-1} \in \mathbb{G}$ *(called the* ***inverse*** *of* u*) such that* $u * u^{-1} = 1_{\mathbb{G}}$;

(iv) for all $u, v \in \mathbb{G}$, *we have* $u * v = v * u$ *(i.e. "* $*$ *" is commutative).*

If we drop the property (iv) from the definition 2.2, we get the definition of a more general notion of a **group**. Let $\mathbb{G}$ be an abelian group with binary operation $*$, we have that $\mathbb{G}$ contains only one identity element and every element of $\mathbb{G}$ has only one inverse.

Definition 2.3. *An abelian group is a* **cyclic group** *if there exists one element that can generate the whole group. If a cyclic group denoted by* $\mathbb{G} = \langle g \rangle$, *we say this cyclic group is generated by* g *and* g *is a generator of group* $\mathbb{G}$.

Let $|\mathbb{G}|$ denote the number of elements in $\mathbb{G}$ and we call it the order of group $\mathbb{G}$. For an element $g \in \mathbb{G}$, we call the minimum $a \in \mathbb{Z}$ such that $g^a = 1_{\mathbb{G}}$ as the order of g and denote it as $ord(g)$. Then we have following theorem.

Theorem 2.1. *If $\mathbb{G}$ is a cyclic group and $g \in \mathbb{G}$, then $|\mathbb{G}|$ is divisible by $ord(g)$.*

2.2.3 Lagrange Polynomial Interpolation

Here we review the Lagrange interpolation polynomial that will be used in the proposed scheme constructions. Given $k+1$ distinct points $(x_0, y_0), (x_1, y_1), \cdots, (x_k, y_k)$, let

$$f_i(x) = \prod_{j=0, j\neq i}^{k} \frac{x - x_j}{x_i - x_j} = \sum_{j=0}^{k} a_{i,j} x^j,$$

we have $f_i(x_j) = 1$, if $i = j$ and $f_i(x_j) = 0$, if $i \neq j$. Then there exists a unique interpolation polynomial $F(x)$ with order k such that $F(x_i) = y_i$, where

$$F(x) = \sum_{i=0}^{k} y_i f_i(x).$$

2.3 Bilinear Pairing

Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ be three cyclic groups of the same order p for some large prime p. Let g_1 be the generator of $\mathbb{G}_1$, g_2 be the generator of $\mathbb{G}_2$. A map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a bilinear map if it satisfies the following three properties.

1. Bilinear: for all $u \in \mathbb{G}_1$, $v \in \mathbb{G}_2$ and for all $a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. Non-Degenerate: $e(g_1, g_2)$ is a generator of $\mathbb{G}_T$.
3. Computable: for all $u \in \mathbb{G}_1$, $v \in \mathbb{G}_2$, there exists efficient algorithms to compute $e(u, v)$.

A bilinear group $\mathbb{BG} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, p, e)$ is composed of objects as described above. If $\mathbb{G}_1 = \mathbb{G}_2$, we call the pairing is a symmetric pairing. Otherwise, we call the pairing is a asymmetric pairing. In this thesis, we use the symmetric pairing in the scheme construction and denote the corresponding symmetric bilinear group as $\mathbb{BG} = (\mathbb{G}, \mathbb{G}_T, e, p)$.

2.4 Complexity Assumptions

In this section, we review some hard problems mentioned in this thesis which are believed to be intractable. The security of our proposed schemes are based on the

corresponding assumptions. Roughly speaking, the hard problems can be classified into *computational hard problems* and *decisional hard problems*. We list some hard problems which will be used in the analysis of our proposed schemes. In the following, $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of the large prime order p, and $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is a pairing map unless it is specified otherwise.

Computational Diffie-Hellman Problem (CDH) [DH76].Given $\left(g, g^a, g^b\right) \in \mathbb{G}$, to compute g^{ab}, where a, b are from $\mathbb{Z}_p$ and are unknown.

Definition 2.4 (CDH Assumption)**.** *Given $\left(g, g^a, g^b\right)$, we say the CDH assumption holds in $\mathbb{G}$ if no probabilistic polynomial time (PPT) adversary $\mathcal{A}$ can compute g^{ab} with the advantage*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{CDH}}(\lambda) = \Pr\left[\mathcal{A}\left(g, g^a, g^b\right) = g^{ab}\right] \geq \epsilon(\lambda)$$

where the probability is taken over the random choice of $a, b \in \mathbb{Z}_p$ and bits consumed by the adversary $\mathcal{A}$.

Decisional Diffie-Hellman Problem (DDH) [Bon98]. Given $\left(g, g^a, g^b, Z\right) \in \mathbb{G}$, to decide whether $Z = g^{ab}$, where a, b are from $\mathbb{Z}_p$ and are unknown.

Definition 2.5 (DDH Assumption)**.** *Given $\left(g, g^a, g^b, Z\right)$, we say the DDH assumption holds in $\mathbb{G}$ if no PPT adversary $\mathcal{A}$ can distinguish $\left(g, g^a, g^b, g^{ab}\right)$ from $\left(g, g^a, g^b, Z\right)$ with advantage*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{DDH}}(\lambda) = \left| \Pr\left[\mathcal{A}\left(g, g^a, g^b, g^{ab}\right) = 1\right] - \Pr\left[\mathcal{A}\left(g, g^a, g^b, Z\right) = 1\right] \right| \geq \epsilon(\lambda)$$

where the probability is taken over the random choice of $a, b, c \in \mathbb{Z}_p$ and bits consumed by the adversary $\mathcal{A}$.

Bilinear Diffie-Hellman Problem (BDH) [BF01]. Given $\left(g, g^a, g^b, g^c\right) \in \mathbb{G}$, to compute $e(g, g)^{abc}$, where a, b, c are from $\mathbb{Z}_p$ and are unknown.

Definition 2.6 (BDH Assumption)**.** *Given $\left(g, g^a, g^b, g^c\right)$, we say the BDH assumption holds in $\mathbb{G}$ if no PPT adversary $\mathcal{A}$ can compute $e(g, g)^{abc}$ with the advantage*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{BDH}}(\lambda) = \Pr\left[\mathcal{A}\left(g, g^a, g^b, g^c\right) = e(g, g)^{abc}\right] \geq \epsilon(\lambda)$$

where the probability is taken over the random choice of $a, b, c \in \mathbb{Z}_p$ and bits consumed by the adversary $\mathcal{A}$.

Decisional Bilinear Diffie-Hellman Problem (DBDH) [Wat05]. Given $\left(g, g^a, g^b, g^c, Z\right) \in \mathbb{G}$, to decide whether $Z = e(g, g)^{abc}$ or a random element from $\mathbb{G}_T$, where a, b, c are from $\mathbb{Z}_p$ and are unknown.

Definition 2.7 (DBDH Assumption)**.** *Given* $\left(g, g^a, g^b, g^c, Z\right)$*, we say the DBDH assumption holds in* $\mathbb{G}$ *if no PPT adversary* $\mathcal{A}$ *can distinguish* $\left(g, g^a, g^b, g^c, e(g,g)^{abc}\right)$ *from* $\left(g, g^a, g^b, g^c, Z\right)$ *with the advantage*

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{DBDH}}(\lambda) &= \left|\Pr\left[\mathcal{A}\left(g, g^a, g^b, g^c, e(g,g)^{abc}\right)=1\right] - \Pr\left[\mathcal{A}\left(g, g^a, g^b, g^c, Z\right)=1\right]\right| \\ &\geq \epsilon(\lambda),\end{aligned}$$

where the probability is taken over the random choice of $a, b, c \in \mathbb{Z}_p$ *and bits consumed by the adversary* $\mathcal{A}$*.*

2.4.1 General Diffie-Hellman Exponent Problem

Following [Del07], we describe the general Diffie-Hellman exponent problem. In [BBG05], Boneh, Boyen and Goh introduced a number of Diffie-Hellman-type complexity assumptions in the generic group model [Sho97]. They include the BDH assumption [BF01], the DH Inversion assumption (DHI)[BB04], the Linear DH assumption[BBS04], and the BDHE assumption[BGW05], and others.

We give an overview of the generalization of the Diffie-Hellman exponent assumptions in the symmetric case. Let $\mathbb{BG} = (\mathbb{G}, \mathbb{G}_T, e, p)$ be a bilinear map, g_0 be a generator of $\mathbb{G}$. Let s, n be positive integers and $P, Q \in \mathbb{F}_p[X_1, X_2, \cdots, X_n]^s$ be two s-tuples of n-variate polynomials over $\mathbb{F}_p$. Therefore, P and Q are just two lists containing s multi-variate polynomials each. We write $P = (p_1, p_2, \cdots, p_s)$ and $Q = (q_1, q_2, \cdots, q_s)$ and require that $p_1 = q_1 = 1$. For a set Ω, a function $h : \mathbb{F}_p \to \Omega$ and vector $(x_1, x_2, \cdots, x_n) \in \mathbb{F}_p^n$, we write

$$h\big(P(x_1, x_2, \cdots, x_n)\big) = \Big(h\big(p_1(x_1, x_2, \cdots, x_n)\big), \cdots, h\big(p_s(x_1, x_2, \cdots, x_n)\big)\Big) \in \Omega^s.$$

We use similar notion for the s-tuple Q.

We say that a polynomial $F \in \mathbb{F}_p[X_1, X_2, \cdots, X_n]$ depends on the sets (P, Q) which we denote by $F \in \langle P, Q\rangle$ if there exists $s^2 + s$ constants $\{a_{i,j}\}_{i,j=1}^s$, $\{b_i\}_{i=1}^s$ such that

$$F = \sum_{i,j=1}^{s} a_{i,j} p_i p_j + \sum_{i=1}^{s} b_i q_i.$$

We say that F is independent of (P, Q) which we denote by $F \notin \langle P, Q\rangle$ if F is not dependent on (P, Q). The (P, Q, F)- General Decision Diffie-Hellman Exponent Problem ((P,Q,F)-GDDHE) is defined as follow.

Definition 2.8 (**(P,Q,F)-GDDHE).** *Given the tuple*

$$H(x_1, x_2, \cdots, x_n) = \left(g_0^{P(x_1, x_2, \cdots, x_n)}, e(g_0, g_0)^{Q(x_1, x_2, \cdots, x_n)}\right) \in \mathbb{G}^s \times \mathbb{G}_T^s,$$

and $Z \in \mathbb{G}_T$*, to decide whether* $Z = e(g_0, g_0)^{F(x_1, x_2, \cdots, x_n)}$.

We say that an algorithm $\mathcal{D}$ that outputs a bit $\mu \in \{0, 1\}$ has advantage $\mathsf{Adv}^{\mathsf{gddhe}}(\mathcal{D})$ in solving the (P, Q, F)-GDDHE problem in $\mathbb{G}$ if

$$\left| \Pr\left[\mathcal{D}\left(H(x_1, \cdots, x_n), g_T^{F(x_1, \cdots, x_n)}\right) = 1\right] - \Pr\left[\mathcal{D}(H(x_1, \cdots, x_n), Z) = 1\right] \right| \geq \mathsf{Adv}^{\mathsf{gddhe}}(\mathcal{D}),$$

where the probability is over the random choice of generator $g_0 \in \mathbb{G}$, the random choice of $x_1, \cdots, x_n \in \mathbb{F}_p$, the random choice of $Z \in \mathbb{G}_T$, and the random bits consumed by $\mathcal{D}$. Then, we have the following result on the (P, Q, F)-GDDHE problem stated in [BBG05].

Theorem 2.2 ([BBG05])**.** *Let* $P, Q \in \mathbb{F}_p[X_1, \cdots, X_n]^s$ *be two* s*-tuples of* n *variate polynomials over* $\mathbb{F}_p$ *and let* $F \in \mathbb{F}_p[X_1, \cdots, X_n]$*. Let* d_P *(resp.* d_Q*,* d_F*) denote the maximal degree of elements of* P *(resp. of* Q*,* F*) and* $d = \mathsf{max}(2d_P, d_Q, d_F)$*. If* $F \notin \langle P, Q \rangle$ *then for any generic model distinguisher* $\mathcal{D}$ *that makes a total of at most* q *queries to the oracles computing the group operation in* $\mathbb{G}$*,* $\mathbb{G}_T$ *and the bilinear pairing* $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$*, we have*

$$\mathsf{Adv}^{\mathsf{gddhe}}(\mathcal{D}) \leq \frac{(q + 2s + 2)^2 \cdot d}{2p}.$$

2.5 Cryptographic Foundations

In this section, we describe some basic cryptographic primitives and useful cryptographic tools.

2.5.1 Hash Function

Hash function was introduced by Carter and Wegman [CW79]. A hash function $H : \{0, 1\}^* \to \{0, 1\}^l$ is a deterministic function which takes an arbitrary length string as input and returns a constant size string as output. A hash function features the following properties:

1. **One-wayness.** Given a value y, any PPT algorithm cannot find a value x such that $y = H(x)$ with non-negligible probability.

2. **Collusion Resistance.** No PPT algorithm can find $x \neq y$ such that $H(x) = H(y)$ with non-negligible probability.

Hash function has been widely used as a building block to scheme construction, including the encryption and digital signatures.

2.5.2 Random Oracle

Random oracle introduced by Bellare and Rogaway [BR93] provides a bridge between cryptographic theory and cryptographic practice. Random oracle has been regarded as a powerful tool to program the security reduction. A random oracle is typically used to represent an ideal hash function whose output is random and uniformly distributed in its output space. In the security reduction, if a hash function H is viewed as a random oracle, then we say this reduction is programmed in the random oracle model. In this model, when given an input x, we cannot compute the value of $H(x)$. The only way to obtain the value of $H(x)$ is to query the oracle. Before query x, $H(x)$ is unknown and uniformly distributed. While if H is a hash function, everyone can compute the value of $H(x)$ for the knowing input x.

When programming a security reduction in the random oracle model, random oracles are very helpful for the simulator, as the simulator can control the random oracles. To respond the oracle queries, the simulator selects any output that looks random from the corresponding output space. It helps the simulator complete the simulation. Usually, if a scheme is proved to be secure in the random oracle model, at least one of hash functions is regarded as random oracles. To obtain the value of $H(x)$, the adversary has to query x to the random oracle. As the value of $H(x)$ is determined by the simulator, it can help the simulator to solve the underlying hard problem. Therefore, the security proof in the random model are believed easier than that without random oracles.

2.5.3 Public Key Encryption

A public key encryption (PKE) scheme consists of the following four algorithms.

- **SysGen**(1^λ). Taking as input a security parameter λ, the system parameter generation algorithm returns the system parameter SP.
- **KeyGen**(SP). Taking as input the system parameter SP, the key generation algorithm returns a public/secret key pair (pk, sk).
- **Encrypt**(SP, pk, M). Taking as input the system parameter SP, the public key pk, and a message M from its message space, the encryption algorithm returns a ciphertext CT.
- **Decrypt**(SP, sk, CT). Taking as input the system parameter SP, the secret key sk, and a ciphertext CT, the decryption algorithm returns a message m or $\perp$ to denote failure.

Correctness. A PKE scheme should satisfy the following correctness requirement. For all $SP \leftarrow \mathsf{SysGen}(1^\lambda)$, $(pk, sk) \leftarrow \mathsf{KeyGen}(SP)$ and $CT \leftarrow \mathsf{Encrypt}(SP, pk, M)$, we have $M \leftarrow \mathsf{Decrypt}(SP, sk, CT)$.

Security models. The indistinguishability security of public key encryption is defined by a game played between a challenger and an adversary. The challenger first generates the public key to the adversary. Then the adversary outputs two messages M_0, M_1 from the same message space for challenge and the challenger generates a challenge ciphertext CT^* on a message randomly chosen from $\{M_0, M_1\}$. Finally, the adversary outputs its guess of the message in CT^*. During the game, the adversary is allowed to make queries with some restrictions to avoid trivial solutions. The security model of indistinguishability chosen-ciphertext attacks (IND-CCA) is defined by the following game.

- **Setup:** Let SP be the system parameter. The challenger runs the key generation algorithm to generate the public key pk and sends pk to the adversary.
- **Phase 1:** The adversary issues decryption queries on ciphertexts CT_i. To respond the query, the challenger runs the decryption algorithm and sends the decryption result to the adversary.
- **Challenge:** Once the adversary decides that the phase 1 is over, it outputs two distinct messages M_0, M_1 from the same message space for challenge. The challenger picks a random bit $\mu \in \{0, 1\}$ and generates a challenge ciphertext CT^*, then it sends CT^* to the adversary.
- **Phase 2:** The adversary continues to make decryption queries on ciphertext CT_i with the restriction that $CT_i \neq CT^*$. The challenger responds to the queries the same as in phase 1.
- **Guess:** The adversary outputs a guess μ' of μ and wins the game if $\mu' = \mu$.

We define the advantage of the adversary in winning this game as

$$\mathsf{Adv}_{\mathcal{PKE}}(\lambda) = \left| \Pr[\mu' = \mu] - \frac{1}{2} \right|.$$

Definition 2.9. *A public key encryption scheme is IND-CCA secure if there exists no probabilistic polynomial time adversary who can win the above game with a non-negligible advantage.*

If we require that the adversary is not allowed to make the decryption query in the IND-CCA security model, we get the security model of indistinguishability chosen-plaintext attacks (IND-CPA) for public key encryption.

2.5.4 Identity-Based Encryption

An identity-based encryption scheme (IBE) consists of the following four algorithms.

- **Setup**(1^λ). Taking as input a security parameter λ, the system setup algorithm returns a master public key mpk which is publicly known and a master secret key msk which is kept secretly.
- **KeyGen**(mpk, msk, ID). Taking as input the master key pair (mpk, msk) and a user identity ID, the key generation algorithm returns a user private key d_{ID}.
- **Encrypt**(mpk, ID, M). Taking as input the master public key mpk, an identity ID, and a message M from its message space, the encryption algorithm returns a ciphertext CT.
- **Decrypt**(mpk, CT, d_{ID}). Taking as input the master public key mpk, a ciphertext CT, an identity ID and the corresponding private key d_{ID}, the decryption algorithm returns a message M or $\perp$ to denote failure.

Correctness. An IBE scheme should satisfy the following correctness requirement. For all $(mpk, msk) \leftarrow \mathsf{Setup}(1^\lambda)$, $d_{ID} \leftarrow \mathsf{KeyGen}(mpk, msk, ID)$ and $CT \leftarrow \mathsf{Encrypt}(mpk, ID, M)$, we have $M \leftarrow \mathsf{Decrypt}(mpk, CT, d_{ID})$.

Security models. The security of IBE requires that without a private key, the adversary cannot decrypt the encrypted message. More precisely, the indistinguishability security of IBE is defined by a game playing between a challenger and an adversary. The challenger first generates the master public key to the adversary. Then the adversary outputs two message M_0, M_1 from the same message space for challenge and the challenger generates a challenge ciphertext CT^* on a message randomly chosen from $\{M_0, M_1\}$. Finally, the adversary outputs its guess of the message in CT^*. During the game, the adversary is allowed to make private key queries and decryption queries as needed with some restrictions. The security model of indistinguishability chosen-ciphertext attacks (IND-CCA) is defined by the following game.

- **Setup:** The challenger runs the setup algorithm to generate the master public key mpk and sends mpk to the adversary.
- **Phase 1:** In this phase, the adversary can issue private key queries and decryption queries as needed.

- *Private key query.* For the query on ID_i, the challenger runs the key generation algorithm to generate d_{ID_i} and sends d_{ID_i} to the adversary.
- *Decryption query.* For the query on (ID_i, CT_i), the challenger runs the key generation algorithm to generate the private key d_{ID_i} and runs the decryption algorithm to decrypt the ciphertext CT_i using d_{ID_i}.

– **Challenge:** Once the adversary decides that the phase 1 is over, it outputs an identity ID^* and two distinct messages M_0, M_1 from the same message space for challenge. We require that the private key of ID^* has not been queried in the phase 1 to avoid trivial solutions. The challenger picks a random bit $\mu \in \{0, 1\}$ and generates a challenge ciphertext CT^*, then it sends CT^* to the adversary.

– **Phase 2:** The adversary continues to make private key queries on $ID_i \neq ID^*$ and decryption queries on (ID_i, CT_i) with the restriction that $CT_i \neq CT^*$. The challenger responds to the queries the same as in phase 1.

– **Guess:** The adversary outputs a guess μ' of μ and wins the game if $\mu' = \mu$.

We define the advantage of the adversary in winning this game as

$$\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathcal{IBE}}(\lambda) = \left| \Pr[\mu' = \mu] - \frac{1}{2} \right|.$$

Definition 2.10. *An identity-based encryption scheme is IND-CCA secure if there exists no probability polynomial time adversary who can win the above game with a non-negligible advantage.*

If we require that the adversary is not allowed to make the decryption query in the IND-CCA security model, we get the security model of indistinguishability chosen-plaintext attacks (IND-CPA). We also get selective security if the adversary must commit the challenge identity ID^* before seeing the master public key.

2.5.5 Identity-Based Broadcast Encryption

An identity-based broadcast encryption (IBBE) scheme consists of the following four algorithms.

– **Setup**$(1^\lambda, N)$. Taking as input a security parameter λ and N the maximal size of the set of receivers for one encryption, the setup algorithm returns a master public key mpk and a master secret key msk. The master public key mpk is publicly known and the master secret key msk is kept secretly.

- **KeyGen**(mpk, msk, ID). Taking as input the master key pair (mpk, msk) and a user identity ID, the key generation returns a user private key d_{ID}.
- **Encrypt**(mpk, M, S). Taking as input the master public key mpk, a message M, and a set of identities S with $|S| \leq N$, the encryption algorithm returns a ciphertext CT.
- **Decrypt**(mpk, CT, ID, d_{ID}, S). Taking as input the master public key mpk, a ciphertext CT, a set S, an identity ID and the corresponding private key d_{ID}, the decryption algorithm returns M if $ID \in S$ or $\perp$ otherwise.

Correctness. An IBBE scheme should satisfy the following correctness requirement. For all $(mpk, msk) \leftarrow \mathsf{Setup}(1^\lambda, N)$, $d_{ID} \leftarrow \mathsf{KeyGen}(mpk, msk, ID)$ and $CT \leftarrow \mathsf{Encrypt}(mpk, M, S)$, if $ID \in S$, we have $M \leftarrow \mathsf{Decrypt}(mpk, CT, ID, d_{ID}, S)$.

Security models. The standard security notion of identity-based broadcast encryption scheme is indistinguishability security against chosen-ciphertext attacks (IND-CCA). It requires that given a set of identities S and two distinct messages M_0 and M_1 from the same message space, the adversary has a negligible advantage to tell apart which message has been encrypted in the challenge ciphertext. The adversary is permitted to access the private key query and the decryption query under some restrictions to avoid trivial solutions. Precisely, the IND-CCA secruity is defined via a security game played by a challenger and an adversary below. Both the adversary and the challenger are given as input N, the maximal size of the set of receivers S.

- **Setup:** The challenger runs the setup algorithm to generate the master public key mpk and sends mpk to the adversary.
- **Phase 1:** In this phase, the adversary can make private key queries and decryption queries as needed.
  - *Private key query.* For the query on ID_i, the challenger runs the key generation algorithm to generate d_{ID_i} and sends d_{ID_i} to the adversary.
  - *Decryption query.* For the query on (ID_i, S_i, CT_i) with $ID_i \in S_i$, the challenger runs the key generation algorithm to generate the private key d_{ID_i} and runs the decryption algorithm to decrypt the ciphertext CT_i using d_{ID_i}.

– **Challenge:** Once the adversary decides that the phase 1 is over, it outputs an identity set S^* with $|S^*| \leq N$ and two distinct messages M_0, M_1 from the same message space for challenge. We require that the private keys of $ID_i^* \in S^*$ have not been queried. The challenger picks a random bit $\mu \in \{0, 1\}$ and generates a challenge ciphertext CT^* under S^*, then it sends CT^* to the adversary.

– **Phase 2:** The adversary continues to make private key queries on $ID_i \notin S^*$ and decryption queries on (ID_i, S_i, CT_i) with the restriction that $CT_i \neq CT^*$. The challenger responds to the queries the same as in phase 1.

– **Guess:** The adversary outputs a guess μ' of μ and wins the game if $\mu' = \mu$.

We define the advantage of the adversary in winning this game as

$$\mathsf{Adv}_{\mathcal{IBBE}}^{\mathsf{IND\text{-}CCA}}(\lambda, N) = \left| \Pr[\mu' = \mu] - \frac{1}{2} \right|.$$

Definition 2.11. *An identity-based broadcast encryption scheme is IND-CCA secure if there exists no probabilistic polynomial time adversary who can win the above game with a non-negligible advantage.*

One weaker notion of IBBE is selective-ID security, where the adversary must choose the set of identities he wants to attack at the beginning of the game. We define IND-sID-CCA security of an IBBE system via the following game between an adversary and a challenger. Both the adversary and the challenger are given as input N, the maximal size of the set of receivers S.

– **Init:** The adversary first outputs a set $S^* = \{ID_1^*, ID_2^*, \cdots, ID_s^*\}$ of identities that he wants to attack with $s \leq N$.

– **Setup:** The challenger runs the setup algorithm to generate the master public key mpk and sends mpk to the adversary.

– **Phase 1:** In this phase, the adversary can make private key queries and decryption queries as needed.

  - *Private key query.* For the query on ID_i with the restriction that $ID_i \notin S^*$, the challenger runs the key generation algorithm to generate d_{ID_i} and sends d_{ID_i} to the adversary.

  - *Decryption query.* For the query on (ID_i, S_i, CT_i) with $ID_i \in S_i$ and $S_i \subseteq S^*$, the challenger runs the private key generation algorithm to generate the private key d_{ID_i} and runs the decryption algorithm to decrypt the ciphertext CT_i using d_{ID_i}.

– **Challenge:** Once the adversary decides that the phase 1 is over, it outputs two distinct messages M_0, M_1 from the same message space for challenge. The challenger picks a random bit $\mu \in \{0,1\}$ and generates a challenge ciphertext CT^* under S^*, then it sends CT^* to the adversary.

– **Phase 2:** The adversary continues to make private key queries and decryption queries as in the phase 1 with the restriction that $CT_i \neq CT^*$. The challenger responds to the queries the same as in phase 1.

– **Guess:** The adversary outputs a guess μ' of μ and wins the game if $\mu' = \mu$.

We define the advantage of the adversary in winning this game as

$$\mathsf{Adv}^{\mathsf{IND\text{-}sID\text{-}CCA}}_{\mathcal{IBBE}}(\lambda, N) = \left| \Pr[\mu' = \mu] - \frac{1}{2} \right|.$$

Definition 2.12. *An identity-based broadcast encryption scheme is IND-sID-CCA secure if there exists no probabilistic polynomial time adversary who can win the above game with a non-negligible advantage.*

If we require that the adversary is not allowed to access the decryption query in the above two security models, we get the corresponding indistinguishability security against chosen-plaintext attacks (IND-CPA).

2.5.6 Inner Product Encryption

The inner product encryption (IPE) is a special functional encryption [BSW11] for inner products. The output of the function is a real value of inner product. An IPE scheme can be specified by the following four algorithms.

– $\mathsf{Setup}(1^\lambda, n)$. Taking as input a security parameter λ and n the length of vectors, it outputs a master public key mpk and a master secret key msk. The master public key mpk is made public and the master secret key msk is kept secretly. The master public key contains the descriptions of a key space $\mathcal{K}$ and a message space $\mathcal{X}$.

– $\mathsf{KeyGen}(mpk, msk, \vec{y})$. Taking as input the master key pair (mpk, msk) and a vector $\vec{y} \in \mathcal{K}^n$, it outputs a private key $sk_{\vec{y}}$ of $\vec{y}$.

– $\mathsf{Encrypt}(mpk, \vec{x})$. Taking as input the master public key mpk and a message $\vec{x} \in \mathcal{X}^n$, it outputs a ciphertext CT.

– $\mathsf{Decrypt}(mpk, CT, sk_{\vec{y}})$. Taking as input the master public key mpk, a ciphertext CT, and a private key $sk_{\vec{y}}$ of $\vec{y}$, it outputs $\langle \vec{x}, \vec{y} \rangle$ or $\perp$.

We make the following correctness requirement: for any $(mpk, msk) \leftarrow \mathsf{Setup}(1^\lambda, n)$, all $\vec{y} \in \mathcal{K}^n$, $\vec{x} \in \mathcal{X}^n$, for $sk_{\vec{y}} \leftarrow \mathsf{KeyGen}(mpk, msk, \vec{y})$ and $CT \leftarrow \mathsf{Encrypt}(mpk, \vec{x})$, we have that $\mathsf{Decrypt}(mpk, CT, sk_{\vec{y}}) = \langle \vec{x}, \vec{y} \rangle$ whenever $\mathsf{Decrypt}(mpk, CT, sk_{\vec{y}}) = \perp$ except with a negligible probability.

IND-CPA Security. For an IPE scheme $\mathcal{IPE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ over $(\mathcal{K}, \mathcal{X})$, we define security against chosen-plaintext attacks (IND-CPA, for short) via a security game played by a challenger and an adversary. The security model of IND-CPA is defined by the following game. Both the adversary and the challenger are given as input n, the length of the message.

– **Setup:** The challenger runs the setup algorithm to generate the master public key mpk and sends mpk to the adversary. It then sets $V \leftarrow \emptyset$.

– **Phase 1:** In this phase, the adversary can issue private key queries as needed. For the query on $\vec{y}_i$, the challenger runs the key generation algorithm to generate $sk_{\vec{y}_i}$ and sends $sk_{\vec{y}_i}$ to the adversary. It then sets $V \leftarrow V \cup \{\vec{y}_i\}$.

– **Challenge:** Once the adversary decides that the phase 1 is over, it outputs two distinct vectors $\vec{x}_0, \vec{x}_1$ from the same space for challenge. We require that for all $\vec{y} \in V$, we have $\langle \vec{x}_0, \vec{y} \rangle = \langle \vec{x}_1, \vec{y} \rangle$. The challenger picks a random bit $\mu \in \{0, 1\}$ and generates a challenge ciphertext CT^*, then it sends CT^* to the adversary.

– **Phase 2:** The adversary continues to make private key queries on $\vec{y}_i$ with the restriction that $\langle \vec{x}_0, \vec{y}_i \rangle = \langle \vec{x}_1, \vec{y}_i \rangle$. The challenger responds to the queries the same as in phase 1.

– **Guess:** The adversary outputs a guess μ' of μ and wins the game if $\mu' = \mu$.

We define the advantage of the adversary in winning this game as

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{IPE}}(\lambda) = \left| \Pr[\mu' = \mu] - \frac{1}{2} \right|.$$

Definition 2.13. *An IPE scheme is IND-CPA secure if there exists no probability polynomial time adversary who can win the above game with non-negligible advantage.*

If we require $\mathcal{A}$ to commit to the challenge messages $\vec{x}_0, \vec{x}_1$ before seeing the master public key, we get selective security.

Chapter 3

Anonymous Revocable IBBE

This chapter describes a new construction of anonymous revocable identity-based broadcast encryption scheme. It is the first revocable IBBE scheme which considers the user privacy. The original scheme was presented at *ACISP 2016* [LMG+16].

3.1 Introduction

Since the pioneering work of Fiat and Naor in [FN94], broadcast encryption has been extensively studied and in many flavors to achieve more functionalities, higher efficiency and higher security [BGW05, DPP07, GW09, LPQ12]. Broadcast encryption in the identity-based setting (IBBE) [Del07] plays a significant role in the applications in terms of the metrics of identity-based cryptography. In the IBBE, a user is allowed to retrieve the encrypted message if and only if the corresponding identity is selected to perform the encryption. As some receivers might leave the system, such as the employees leave one company, or some receivers' private keys are exposed or compromised, we have to revoke these users such that they cannot retrieve the encrypted message anymore. Therefore, user revocation becomes an important research topic in the broadcast encryption system. Unfortunately, all the revocation systems in the broadcast encryption can only prevent the revoked users from decrypting the future broadcast message rather than the message stated in the broadcast encryption.

Aiming to revoke some users from the original receivers stated in the ciphertext generated in a broadcast encryption system, Susilo et al. [SCG+16] introduced a notion called recipient revocable identity-based broadcast encryption (RR-IBBE), which is an extension of the identity-based broadcast encryption. In the RR-IBBR, it allows a third party to remove some of receivers from the original ciphertext, but the third party cannot decrypt the ciphertext. That is, the receiver revocation operation does not require the knowledge of the message. The revoked users are unable to decrypt the encrypted message even they collude. Compared to the IBBE system, the RR-IBBE system is of one additional algorithm "revoke", which is used for the receiver revocation. In [SCG+16], the authors presented the first recipient

revocable broadcast encryption scheme in the identity-based setting. The ciphertext after revocation surprisingly achieves constant size. While this scheme does not take the receiver privacy into consideration. The decryption requires knowing the identity information of the receivers and the revoked users. However, the receiver privacy-preserving in the broadcast encryption is a very important issue when deploying a broadcast encryption system as we stated in chapter 1.

In this chapter, we will continue to study RR-IBBE and describe the first anonymous construction of RR-IBBE. We propose an anonymous revocable identity-based broadcast encryption scheme and derive its security based on the hardness of BDH problem (see Section 2.4) in the random oracle. In the proposed scheme, the receiver identity information is hidden to the third party who performs the revocation algorithm and to the public. The decryption does not need to know the receivers' identities.

Organization. The rest of this chapter is organized as follows. In the next Section, we give the definitions of anonymous revocable identity-based broadcast encryption and the corresponding security models. The concrete construction is described in Section 3.3. In Section 3.4, we provide the security analysis of the proposed scheme under the defined security models and conclude this chapter in Section 3.5.

3.2 Definitions and Security Models

This section will define the syntax and the security of anonymous revocable identity-based broadcast encryption (AR-IBBE). An AR-IBBE scheme consists of five algorithms defined as follows.

- **Setup**(1^λ). Taking as input a security parameter λ, the setup algorithm returns a master public key mpk and a master secret key msk. The mpk is publicly known while the msk is kept secretly.
- **KeyGen**(mpk, msk, ID). Taking as input the master key pair (msk, mpk) and a user identity ID, the key generation algorithm returns a user private key d_{ID}.
- **Encrypt**(mpk, M, S). Taking as input the master public key mpk, a message M and a set of identities $S = (ID_1, ID_2, ..., ID_n)$, the encryption algorithm returns a ciphertext CT.
- **Revoke**(mpk, R, CT). Taking as input the master public key mpk, a ciphertext CT and a set of revoked identities $R = (ID_1, ID_2, \cdots, ID_t)$ with

$t < n$, the revocation algorithm returns a new ciphertext CT' including the set R.

– **Decrypt**$(mpk, CT', ID, d_{ID}, R,)$. Taking as input the master public key mpk, a ciphertext CT' with a set R, an identity ID and the corresponding private key d_{ID}, the decryption algorithm returns M if $ID \in S \backslash R$, and $\perp$ otherwise to denote failure.

Correctness. Note that if $t = 0$, the AR-IBBE scheme is an anonymous identity-based broadcast encryption scheme. Thus, an AR-IBBE scheme should satisfy the following correctness requirements. For any $(mpk, msk) \leftarrow \mathsf{Setup}(1^\lambda)$, $d_{ID} \leftarrow \mathsf{KeyGen}(mpk, msk, ID)$, $CT \leftarrow \mathsf{Encrypt}(mpk, M, S)$, and $CT' \leftarrow \mathsf{Revoke}(mpk, R, CT)$, if $ID \in S \backslash R$, we have $\mathsf{Decrypt}\,(mpk, CT, ID, d_{ID}) = M$ and $\mathsf{Decrypt}(mpk, CT', ID, d_{ID}, R) = M$.

Remark. In the definition, there are two identity sets. One is the original receiver set S, another is the revoked identity set R. In the application, R is relatively small comparing to S. Therefore, the requirement $t < n$ is reasonable and for simplicity, we always assume that $t < n$ in the rest of this chapter.

Security Notions. The security of AR-IBBE requires that without a valid private key, both the encrypted message and the intended receivers are unknown to the adversary. Let CT be the original ciphertext for receivers S, R be the revoked users and CT' be the ciphertext after revocation. The indistinguishability security of AR-IBBE should satisfy the follows.

1. The message in the ciphertext CT cannot be distinguished without a valid private key associated with an identity $ID \in S$. The message in CT' cannot be distinguished without a valid private key associated with an identity $ID' \in S \backslash R$.

2. The identity set in the ciphertext CT cannot be distinguished without a valid private key associated with an identity $ID \in S$. The identity set in CT' cannot be distinguished without a valid private key associated with an identity $ID' \in S \backslash R$.

We define the IND-ID-CPA security and ANON-ID-CPA security for the AR-IBBE system in a similar way as anonymous IBBE system.

IND-ID-CPA Security (Confidentiality). The IND-ID-CPA security in the AR-IBBE allows the adversary to issue private key query to obtain the private key associated with any identity ID of its choice. The adversary is challenged on an

identity set S^*, two distinct messages M_0, M_1 from the same message space and a revoked identity set R^* adaptively. The adversary's goal is to distinguish whether the challenge ciphertext is generated under M_0 or M_1 for S^* with some restrictions to avoid trivial solutions. We say that the adversary breaks the scheme if it can guess the message correctly. Specifically, the security of IND-ID-CPA is defined under the following game between a challenger and an adversary.

- **Setup:** The challenger runs the setup algorithm to generate the master public key mpk and sends mpk to the adversary.
- **Phase 1:** In this phase, the adversary can issue private key queries as needed. Upon receiving a private key query on ID_i. The challenger runs the key generation algorithm to generate the private key d_{ID_i} and sends the result back to the adversary.
- **Challenge:** Once the adversary decides that the phase 1 is over, it outputs two distinct messages M_0, M_1 from the same message space, a challenge identity set $S^* = (ID_1, ID_2, \cdots, ID_n)$, and a revoked identity set $R^* = (ID'_1, ID'_2, \cdots, ID'_t)(t < n)$ with the restriction that the adversary has not queried the private key on ID_i in the phase 1, where $ID_i \in S^* \backslash R^*$. The challenger randomly picks a bit $\mu \in \{0, 1\}$ and generates the challenge ciphertext CT^* as follows:

$$CT = \mathsf{Encrypt}(mpk, M_b, S^*), \quad CT' = \mathsf{Revoke}(mpk, R^*, CT).$$

  If $R^* \neq \emptyset$, it sets $CT^* = CT'$ as the challenge ciphertext, otherwise sets $CT^* = CT$ as the challenge ciphertext, then it sends CT^* to the adversary.
- **Phase 2:** The adversary issues more private key queries on ID_i with the restriction established in the challenge phase.
- **Guess:** Finally, the adversary outputs its guess $\mu' \in \{0, 1\}$ of μ and wins the game if $\mu' = \mu$.

We refer to such an adversary as an IND-ID-CPA adversary and define the adversary's advantage in winning the above game as

$$\mathsf{Adv}^{\mathsf{IND\text{-}ID\text{-}CPA}}_{\mathcal{AR}-\mathcal{IBBE}}(\lambda) = \left| \Pr[\mu = \mu'] - \frac{1}{2} \right|.$$

The probability is over the random bits used by the challenger and the adversary.

Definition 3.1. *We say that an AR-IBBE scheme is IND-ID-CPA secure if for any probabilistic polynomial time IND-ID-CPA adversary,* $\mathsf{Adv}^{\mathsf{IND\text{-}ID\text{-}CPA}}_{\mathcal{AR}-\mathcal{IBBE}}(\lambda)$ *is negligible.*

ANON-ID-CPA Security (Anonymity). ANON-ID-CPA security in the AR-IBBE allows the adversary to issue the private key query to obtain the private key of any identity ID of its choice. Similarly, the adversary is challenged on a message M^*, two distinct identity sets S_0, S_1 and a revoked identity set R^* of its choice. Adversary's goal is to distinguish whether the challenge ciphertext is generated under S_0 or S_1 with some restrictions to avoid trivial solutions. We say that the adversary breaks the scheme if it can guess the identity set correctly. Specifically, the notion of ANON-ID-CPA is defined under the following game between a challenger and an adversary.

- **Setup:** The challenger runs the setup algorithm to generate the master public key mpk and sends mpk to the adversary.
- **Phase 1:** In this phase, the adversary can issues private key queries as needed. Upon receiving a private key query on ID_i. $\mathcal{C}$ runs the key generation algorithm to generate the private key d_{ID_i} and sends d_{ID_i} to the adversary.
- **Challenge:** When the adversary decides that the phase 1 is over, it outputs a message M^*, two distinct identity sets $S_0 = (ID_{0,1}, ID_{0,2}, ..., ID_{0,n})$, $S_1 = (ID_{1,1}, ID_{1,2}, ..., ID_{1,n})$ and a revoked identity set $R^* = (ID'_1, ID'_2, \cdots, ID'_t)$ $(t < n)$. We require that the adversary has not made the private key queries on ID_i in the phase 1, where $ID_i \in (S_0 \cup S_1) \backslash (S_0 \cap S_1)$. The challenger randomly picks a bit $\mu \in \{0, 1\}$ and generates the challenge ciphertext CT^* as follows:

  $$CT = \mathsf{Encrypt}(mpk, M^*, S_\mu), \quad CT' = \mathsf{Revoke}(mpk, R^*, CT).$$

  If $R^* \neq \emptyset$, set $CT^* = CT'$ as the challenge ciphertext, otherwise set $CT^* = CT$ as the challenge ciphertext, then send CT^* to the adversary.
- **Phase 2:** The adversary issues more private key queries as in the phase 1 with the restriction established in the challenge phase. The challenger responds the same as in phase 1.
- **Guess:** Finally, The adversary outputs its guess $\mu' \in \{0, 1\}$ of μ and wins the game if $\mu' = \mu$.

We refer to such an adversary as an ANON-ID-CPA adversary and define the adversary's advantage in winning the scheme as

$$\mathsf{Adv}^{\mathsf{ANON\text{-}ID\text{-}CPA}}_{\mathcal{AR}-\mathcal{IBBE}}(\lambda) = \left| \Pr[\mu' = \mu] - \frac{1}{2} \right|.$$

The probability is over the random bits used by the challenger and the adversary.

Definition 3.2. *We say that an AR-IBBE scheme is ANON-ID-CPA secure if for any probabilistic polynomial time ANON-ID-CPA adversary,* $\mathsf{Adv}^{\mathsf{ANON\text{-}ID\text{-}CPA}}_{\mathcal{AR}-\mathcal{IBBE}}(\lambda)$ *is negligible.*

3.3 The Proposed Scheme

In this section, we describe the proposed scheme.

3.3.1 Construction

Setup(1^λ). Given a security parameter λ, the setup algorithm randomly chooses a bilinear group $\mathbb{BG} = (\mathbb{G}, \mathbb{G}_T, e, p)$ with a generator $P \in \mathbb{G}$. It picks a random $s \in \mathbb{Z}_p$ and computes $P_{pub} = sP$. It then picks four cryptographic hash functions $H : \{0,1\}^* \to \mathbb{Z}_p$, $H_1 : \{0,1\}^* \to \mathbb{G}$, $H_2 : \mathbb{G}_T \times \{0,1\}^* \to \mathbb{G}$, and $H_3 : \mathbb{G}_T \times \{0,1\}^* \to \mathbb{G}$. The master public key and master secret key are

$$mpk = (\mathbb{BG}, P, P_{pub}, H, H_1, H_2, H_3), \quad msk = s.$$

KeyGen(mpk, msk, ID). Given the master key pair (mpk, msk) and an identity $ID \in \{0,1\}^*$, the key generation outputs the private key

$$d_{ID} = sH_1(ID).$$

Encrypt(msk, S, M). Given the master public key mpk, a set of identity $S = (ID_1, ID_2, \ldots, ID_n)$ with $n > 2$ and a message $M \in \mathbb{G}$, the encryption algorithm performs as follows.

1. Randomly choose $r_1, r_2 \in \mathbb{Z}_p$ and $v \in \mathbb{G}$.
2. For $i = 1, 2, \cdots, n$, compute $x_i = H(ID_i)$, and

$$f_i(x) = \prod_{j=1, j\neq i}^{n} \frac{x - x_j}{x_i - x_j} = \sum_{j=0}^{n-1} a_{i,j} x^j \mod p,$$

$$A_i = H_2\Big(e\big(H_1(ID_i), P_{pub}\big)^{r_1}, ID_i\Big),$$

$$B_i = v + H_3\Big(e\big(H_1(ID_i), P_{pub}\big)^{r_2}, ID_i\Big).$$

   We have $f_i(x_i) = 1$ and $f_i(x_j) = 0$ for $i \neq j$.

3. Create the ciphertext CT as $C_0 = v + M,\ C_1 = r_1 P,\ C_2 = r_2 P$, together with,

for each $i = 1, 2, \cdots, n$:

$$Q_i = \sum_{j=1}^{n} a_{j,i-1} A_j, \quad U_i = \sum_{j=1}^{n} a_{j,i-1} B_j.$$

Revoke(mpk, CT, R). Given a ciphertext CT which has been parsed as $(C_0, C_1, C_2, Q_i, U_i, i \in [1, n])$, the master public key mpk and a revoked identity set R, where $|R| = t < n$. The revocation algorithm performs as follows.

1. If $R = \emptyset$, set $CT' = CT$. Otherwise, perform as follows.
2. Randomly choose $u \in \mathbb{G}$ and compute $C_0' = u + C_0$.
3. For $ID_i \in R$, compute $x_i = H(ID_i)$, and

$$g(x) = \prod_{i=1}^{t} (x - x_i) = \sum_{i=0}^{t} b_i x^i \mod p,$$

and set $b_i = 0$ for $i = t+1, t+2, \cdots, n-1$

4. For each $i = 1, 2, \cdots, n$ compute

$$Q_i' = Q_i + b_{i-1} u,$$

and set $CT' = (R, C_0', C_1, C_2, Q_i', U_i, i \in [1, n])$.

Decrypt$(mpk, CT', ID_i, d_{ID_i})$. Given a ciphertext CT' after revocation which has been parsed as $(R, C_0', C_1, C_2, Q_i', U_i, i \in [1, n])$, the master public key mpk, and an identity ID_i and the corresponding private key d_{ID_i}, the decryption algorithm performs as follows.

1. Compute $x_i = H(ID_i)$ and

$$U = U_1 + x_i U_2 + x_i^2 U_3 + \cdots + x_i^{n-1} U_n,$$

$$Q = Q_1' + x_i Q_2' + x_i^2 Q_3' + \cdots + x_i^{n-1} Q_n'.$$

2. For each $ID_j \in R$, compute $x_j = H(ID_j)$ and reconstruct $g(x)$ as:

$$g(x) = \prod_{j=1}^{t} (x - x_j) = \sum_{j=0}^{t} b_j x^j \mod p.$$

3. Use the private key d_{ID_i} to compute

$$v' = U - H_3\big(e(C_2, d_{ID_i}), ID_i\big),$$

$$u' = g(x_i)^{-1}\left(Q - H_2\big(e(C_1, d_{ID_i}), ID_i\big)\right).$$

and recover the message by computing $M = C_0' - u' - v'$.

If $ID_i \in S \backslash R$, we have $u' = u$, $v' = v$, then it obtains the correct M after decryption.

Remark: For simplicity, we omit the modular operation and assume that the coefficients of all polynomials are from $\mathbb{Z}_p$ in this chapter unless otherwise stated explicitly.

3.3.2 Discussion and Correctness

In the encryption phase, we require that the size of the identity set S is at least 3. This requirement is resulted from using the technique of Lagrange base polynomial. This setting can also simplify our security proof and we do not consider the case where there are only two users in S. One may think that after revocation, the revoked identity set may be updated multiple times. Our scheme allows the third party (or server) to update the revoked identity set. For each update, the third party uses the original ciphertext and the new revoked identity set to perform the revocation algorithm. Thus, the third party needs to store the original ciphertext CT in our scheme. In our setting, there is no requirement of $R \subset S$. The revocation set R can be arbitrary users.

From our setting, only the users in S can decrypt the ciphertext CT. After revocation, the revoked users cannot decrypt the ciphertext CT'. We note that if $ID \in R$, $g(H(ID)) = 0$ and $g(H(ID))u = 0_{\mathbb{G}}$. The user with identity ID cannot retrieve one of the decryption keys u, even all users in R conclude. To obtain the decryption keys u and v, the user must belong to S and not belong to R. Thus our scheme ensures that even if all the revoked users collude, they still cannot access the file and learn the identities of receivers. However, the revoked identity set should be attached in the final ciphertext. Therefore, our scheme does not consider the privacy of the revoked users.

Next we show that our construction meets the requirements of correctness as we claimed in the definition. For any $ID_i \in S$ and $ID_i \notin R$, if $x_i = H(ID_i)$ is

computed correctly, we have $g(x_i) \neq 0$ and

$$\begin{aligned}
Q &= Q'_1 + x_i Q'_2 + x_i^2 Q'_3 + \cdots + x_i^{n-1} Q'_n \\
&= \left(Q_1 + x_i Q_2 + x_i^2 Q_3 + \cdots + x_i^{n-1} Q_n\right) + \left(b_0 + b_1 x_i + b_2 x_i^2 + \cdots + b_{n-1} x_i^{n-1}\right) u \\
&= (a_{1,0} A_1 + a_{2,0} A_2 + \cdots + a_{n,0} A_n) \\
&\quad + x_i \left(a_{1,1} A_1 + a_{2,1} A_2 + \cdots + a_{n,1} A_n\right) + \cdots \\
&\quad + x_i^{n-1} \left(a_{1,n-1} A_1 + a_{2,n-1} A_2 + \cdots + a_{n,n-1} A_n\right) + g(x_i) u \\
&= \left(a_{1,0} + a_{1,1} x_i + a_{1,2} x_i^2 + \cdots + a_{1,n-1} x_i^{n-1}\right) A_1 \\
&\quad + \left(a_{2,0} + a_{2,1} x_i + a_{2,2} x_i^2 + \cdots + a_{2,n-1} x_i^{n-1}\right) A_2 + \cdots \\
&\quad + \left(a_{n,0} + a_{n,1} x_i + a_{n,2} x_i^2 + \cdots + a_{n,n-1} x_i^{n-1}\right) A_n + g(x_i) u \\
&= f_1(x_i) A_1 + f_2(x_i) A_2 + \cdots + f_n(x_i) A_n + g(x_i) u \\
&= A_i + g(x_i) u,
\end{aligned}$$

$$\begin{aligned}
u' &= g(x_i)^{-1} \cdot \left(Q - H_2\big(e(C_1, d_{ID_i}), ID_i\big)\right) \\
&= g(x_i)^{-1} \cdot \left(A_i + g(x_i) u - H_2\big(e(C_1, d_{ID_i}), ID_i\big)\right) \\
&= g(x_i)^{-1} \cdot \Big(H_2\Big(e\big(H_1(ID_i), P_{pub}\big)^{r_1}, ID_i\Big) - H_2\Big(e\big(r_1 P, s H_1(ID_i)\big), ID_i\Big) + g(x_i) u\Big) \\
&= g(x_i)^{-1} \cdot \left(g(x_i) u\right) \\
&= u.
\end{aligned}$$

The user with identity ID_i uses its private key d_{ID_i} to remove A_i from Q_i via the above computation. As $g(x_i) \neq 0$, the user can obtain u. For another decryption key, it computes

$$\begin{aligned}
U &= U_1 + x_i U_2 + x_i^2 U_3 + \cdots + x_i^{n-1} U_n \\
&= (a_{1,0} B_1 + a_{2,0} B_2 + \cdots + a_{n,0} B_n) \\
&\quad + x_i (a_{1,1} B_1 + a_{2,1} B_2 + \cdots + a_{n,1} B_n) + \cdots \\
&\quad + x_i^{n-1} (a_{1,n-1} B_1 + a_{2,n-1} B_2 + \cdots + a_{n,n-1} B_n) \\
&= \left(a_{1,0} + a_{1,1} x_i + a_{1,2} x_i^2 + \cdots + a_{1,n-1} x_i^{n-1}\right) B_1 \\
&\quad + \left(a_{2,0} + a_{2,1} x_i + a_{2,2} x_i^2 + \cdots + a_{2,n-1} x_i^{n-1}\right) B_2 + \cdots \\
&\quad + \left(a_{n,0} + a_{n,1} x_i + a_{n,2} x_i^2 + \cdots + a_{n,n-1} x_i^{n-1}\right) B_n \\
&= f_1(x_i) B_1 + f_2(x_i) B_2 + \cdots + f_n(x_i) B_n \\
&= B_i, \\
v' &= U - H_3\big(e(C_2, d_{ID_i}), ID_i\big) \\
&= B_i - H_3\Big(e\big(r_2 P, s H_1(ID_i)\big), ID_i\Big) \\
&= v + H_3\Big(e\big(P, H_1(ID_i)\big)^{s r_2}, ID_i\Big) - H_3\Big(e\big(H_1(ID_i), P_{pub}\big)^{r_2}, ID_i\Big) \\
&= v.
\end{aligned}$$

After recovering u and v, we get the message as $C'_0 - u' - v' = M + v + u - u - v = M$.

3.4 Security Analysis

In this section, we show the security of the proposed scheme under the BDH assumption in the random oracle model.

Theorem 3.1. *Let H_1, H_2, H_3 be random oracles. If the BDH problem is hard, our proposed scheme is IND-ID-CPA secure. Specifically, suppose there is an IND-ID-CPA adversary who has advantage ϵ against our proposed scheme by making q_E private key queries and q_{H_1}, q_{H_2}, q_{H_3} queries to the functions H_1, H_2 and H_3 respectively. Then there is an algorithm $\mathcal{B}$ can solve the BDH problem with advantage*

$$\epsilon' \geq \frac{\epsilon}{n \cdot e \cdot q_E \cdot (q_{H_2} + q_{H_3})},$$

where n is the number of the identities stated in the ciphertext.

Proof. Suppose there exists an adversary $\mathcal{A}$ who can break the confidentiality of our scheme with advantage ϵ. We build a simulator $\mathcal{B}$ that can solve the BDH problem with advantage ϵ' by running $\mathcal{A}$. Let (P, aP, bP, cP) be a random instance of BDH problem taken as input by $\mathcal{B}$ and its goal is to compute $e(P,P)^{abc}$. In order to use $\mathcal{A}$ to solve the problem, $\mathcal{B}$ needs to simulate a challenger and respond all queries from $\mathcal{A}$. For ease of exposition, we assume that the H_2 and H_3 queries are after the H_1 query for the same identity. $\mathcal{B}$ works by interacting with $\mathcal{A}$ in an IND-ID-CPA game as follows.

Setup: $\mathcal{B}$ sets $P_{pub} = aP$ and creates the master public as $mpk = (p, P, P_{pub}, e, H)$. Then it sends mpk to $\mathcal{A}$. Here, the hash functions H_1, H_2, H_3 are viewed as random oracles controlled by the simulator.

H_1-queries: $\mathcal{A}$ makes H_1 queries. $\mathcal{B}$ responds a query on ID_i as follows. $\mathcal{B}$ maintains a list $\mathcal{L}_1$ of a tuple (ID_i, c_i, r_i, h_i). This list is initially empty. $\mathcal{B}$ first checks $\mathcal{L}_1$. If the query ID_i has already appeared in the $\mathcal{L}_1$ in a tuple (ID_i, c_i, r_i, h_i), it returns the corresponding h_i as the value of $H_1(ID_i)$. Otherwise, $\mathcal{B}$ performs as follows.

1. Select $c_i \in_R \{0,1\}$ with $\Pr[c_i = 0] = \delta$ for some δ (determine later).
2. Pick $r_i \in_R \mathbb{Z}_p$, if $c_i = 0$, compute $h_i = r_i bP$. If $c_i = 1$, compute $h_i = r_i P$.
3. Add the tuple (ID_i, c_i, r_i, h_i) to the $\mathcal{L}_1$ and respond with h_i to $\mathcal{A}$.

H_2-queries: $\mathcal{A}$ makes H_2 queries. $\mathcal{B}$ responds a query on (X_i, ID_i) as follows. $\mathcal{B}$ maintains a list $\mathcal{L}_2$ of a tuple (X_i, ID_i, λ_i). This list is initially empty. $\mathcal{B}$ first checks $\mathcal{L}_2$. If the query (X_i, ID_i) has already appeared in the $\mathcal{L}_2$ in a tuple (X_i, ID_i, λ_i), it returns the corresponding λ_i as the value of $H_2(X_i, ID_i)$. Otherwise, $\mathcal{B}$ randomly

picks a $\lambda_i \in \mathbb{G}$ as the value of $H_2(X_i, ID_i)$, adds the tuple (X_i, ID_i, λ_i) to the $\mathcal{L}_2$ and responds to $\mathcal{A}$ with λ_i.

H_3-queries: $\mathcal{A}$ makes H_3 queries. $\mathcal{B}$ responds a query on (Y_i, ID_i) as follows. $\mathcal{B}$ maintains a list $\mathcal{L}_3$ of a tuple (Y_i, ID_i, γ_i). This list is initially empty. $\mathcal{B}$ first checks $\mathcal{L}_3$. If the query (Y_i, ID_i) has already appeared in the $\mathcal{L}_3$ in a tuple (Y_i, ID_i, γ_i), it returns the corresponding γ_i as the value of $H_3(Y_i, ID_i)$. Otherwise, $\mathcal{B}$ randomly picks a $\gamma_i \in \mathbb{G}$ as the value of $H_3(Y_i, ID_i)$, adds the tuple (Y_i, ID_i, γ_i) to the $\mathcal{L}_3$ and responds to $\mathcal{A}$ with γ_i.

Phase 1: In this phase, $\mathcal{A}$ issues the private key queries on ID_i as needed. For each time, $\mathcal{B}$ first runs the H_1 query to get the corresponding c_i and r_i. If $c_i = 0$, $\mathcal{B}$ aborts. If $c_i = 1$, $\mathcal{B}$ computes $d_{ID_i} = sH_1(ID_i) = ar_iP = r_iP_{pub}$.

Challenge: When $\mathcal{A}$ decides phase 1 is over, it outputs two distinct messages M_0, M_1 from the same message space, a challenge identity set $S^* = (ID_1, ID_2, \cdots, ID_n)$ and a revoked identity set $R^* = (ID'_1, ID'_2, \cdots, ID'_t)$ with the restriction that $\mathcal{A}$ has not queried the private key on ID_i in the phase 1, where $ID_i \in S^* \backslash R^*$. $\mathcal{B}$ randomly picks a random bit $\mu \in \{0, 1\}$ and performs as follows.

Case 1: $R^* = \emptyset$. In this case, $\mathcal{B}$ randomly picks $r^* \in \mathbb{Z}_p$, $C_0^* \in \mathbb{G}$, and for each $ID_i \in S^*$, $i = 1, 2 \cdots, n$, randomly chooses A_i^*, $B_i^* \in \mathbb{G}$ and computes $x_i^* = H(ID_i)$,

$$f_i(x) = \prod_{j=1, j\neq i}^{n} \frac{x - x_j^*}{x_i^* - x_j^*} = \sum_{j=0}^{n-1} a_{i,j} x^j.$$

Then $\mathcal{B}$ generates the challenge ciphertext CT^* as $C_0, C_1^* = r^* cP, C_2^* = cP$, together with, for each $i = 1, 2, \cdots, n$:

$$Q_i^* = \sum_{j=1}^{n} a_{j,i-1} A_j^*, \quad U_i^* = \sum_{j=1}^{n} a_{j,i-1} B_j^*.$$

Case 2: $R^* \neq \emptyset$. In this case, $\mathcal{B}$ performs as follows.

1. Pick $r^* \in_R \mathbb{Z}_p$, $v^*, u^* \in_R \mathbb{G}$, compute $C_0'^* = v^* + u^* + M_b$, $C_1^* = r^* cP$, $C_2^* = cP$.

2. For each $ID_i \in S^* \backslash R^*$, $\mathcal{B}$ randomly chooses A_i^*, $B_i^* \in \mathbb{G}$. For each $ID_i \in S^* \cap R^*$, $\mathcal{B}$ gets r_i from the $\mathcal{L}_1$ (If ID_i is not in the $\mathcal{L}_1$, run H_1 queries to get r_i). Then it computes $X_i = e(aP, cP)^{r^* r_i}$ and checks whether the tuple (X_i, ID_i) in the $\mathcal{L}_2$. If yes, it obtains the corresponding λ_i and sets $A_i^* = \lambda_i$. Otherwise, it randomly choose $A_i^* \in \mathbb{G}$ and adds the new tuple (X_i, ID_i, A_i^*) to the $\mathcal{L}_2$. Then $\mathcal{B}$ computes $Y_i = e(aP, cP)^{r_i}$ and checks whether the tuple (Y_i, ID_i) in the $\mathcal{L}_3$. If yes, it obtains the corresponding γ_i and sets $w_i^* = \gamma_i$.

Otherwise, it randomly chooses $w_i^* \in \mathbb{G}$ and adds the new tuple (Y_i, ID_i, w_i^*) to the $\mathcal{L}_3$, and computes $B_i^* = w_i^* + v^*$.

3. For each $ID_i \in S^*$, $i = 1, 2 \cdots, n$, compute $x_i^* = H(ID_i)$,

$$f_i(x) = \prod_{j=1, j\neq i}^{n} \frac{x - x_j^*}{x_i^* - x_j^*} = \sum_{j=0}^{n-1} a_{i,j} x^j,$$

$$Q_i^* = \sum_{j=1}^{n} a_{j,i-1} A_j^*, \quad U_i^* = \sum_{j=1}^{n} a_{j,i-1} B_j^*.$$

4. Compute $x_i'^* = H(ID_i)$ for $ID_i \in R^*$ and

$$g(x) = \prod_{i=1}^{t} (x - x_i'^*) = \sum_{i=0}^{t} b_i x^i.$$

Then set $b_i = 0$ for $i = t+1, t+2, \cdots, n-1$.

5. For $1 \leq i \leq n$, compute

$$Q_i'^* = Q_i^* + b_{i-1} u^*,$$

and set $CT^* = (R^*, C_0'^*, C_1^*, C_2^*, Q_i'^*, U_i^*, i \in [1, n])$.

Phase 2: $\mathcal{A}$ continues to issue private key queries as needed with the restriction established in the challenge phase. $\mathcal{B}$ responds the same as in phase 1.

Guess: Finally, $\mathcal{A}$ outputs its guess $\mu' \in \{0, 1\}$ of μ.

Note that in the case $R^* = \emptyset$, we can view v^* as the encryption key to encrypt the challenge message. Let $W = (e(H_1(ID_i), P_{pub})^c, ID_i)$ where $ID_i \in S^*$. In the real scheme, $B_i^* = v^* + H_3(W)$, thus we also can regard $H_3(W)$ as the encryption key to encrypt v^*. Before querying the H_3 value on W, the result of $H_3(W)$ is unknown and random. From the view of the adversary, v^* is encrypted with a random key independent of W. Therefore, B_i^* is a one-time pad. In other words, the challenge ciphertext is a one-time pad. According to the assumption($\mathcal{A}$ can break our scheme with advantage ϵ), the adversary will query H_3 on W. In this case, the simulator decides the corresponding hard problem's solution is in the $\mathcal{L}_3$ and solves it with probability $\frac{\delta}{n}$ as the value of $H_1(ID_i)$ contains the b with probability δ.

When $R^* \neq \emptyset$, we can view v^* and u^* as the encryption keys to encrypt the challenge message. However, in this case, the adversary can retrieve v^* by querying the private key of $ID_i \in S^* \cap R^*$. That is, the message encryption key is only u^*. Let $\Omega = \left(e(H_1(ID_i), P_{pub})^{r^* c}, ID_i\right)$, where $ID_i \in S^* \backslash R^*$. Similarly, in real scheme $Q^* = A_i^* + g(x_i^*)u^* = H_2(\Omega) + g(x_i^*)u^*$, we can regard Ω as the encryption key to

encrypt u^*. Before querying the H_2 value on Ω, the result of $H_2(\Omega)$ is unknown and random. From the view of the adversary, u^* is encrypted with a random key independent of Ω. Therefore, Q^* is a one-time pad, that is, the challenge ciphertext is a one-time pad. According to the assumption($\mathcal{A}$ can break our scheme with advantage ϵ), the adversary will query H_2 on Ω. In this case, the simulator can decide the solution of the corresponding hard problem is in the $\mathcal{L}_2$ and solve it with probability $\frac{\delta}{n-l}$ where $l = |S^* \cap R^*|$. Here, we define the query which can solve the hard problem as *challenge query*.

If the challenge query is made, it means $c_j = 0$, $H_1(ID_j) = r_j bP$ and $d_{ID_j} = r_j abP$. From the decryption algorithm, we have $e(C_1^*, d_{ID_j}) = e(P,P)^{r^* r_j abc}$ and $e(C_2^*, d_{ID_j}) = e(P,P)^{r_j abc}$. Here $\mathcal{B}$ ignores the guess of $\mathcal{A}$ and picks a random tuple from the $\mathcal{L}_2$ or $\mathcal{L}_3$. It first obtains the corresponding r_j from the $\mathcal{L}_1$. If $\mathcal{B}$ picks the tuple (X_j, ID_j, λ_j) from the $\mathcal{L}_2$, it computes $X_j^{(r^* r_j)^{-1}}$ as the solution to the given instance of BDH problem. If $\mathcal{B}$ picks the tuple (Y_j, ID_j, γ_j) from the $\mathcal{L}_3$, it computes $X_j^{r_j^{-1}}$ as the solution to the given instance of BDH problem.

The above completes the description of the simulation. To complete the security proof, it remains to show that $\mathcal{B}$ correctly computes $e\,(P,P)^{abc}$ with advantage at least ϵ'. According to our above analysis, we first define the following events:

E_1: The simulation does not abort in private key query.

E_2: At least one of the H_1 values of challenge identities contains b.

E_3: The adversary chooses an identity where $c_i = 0$ to distinguish challenge message.

E_4: The simulator correctly chooses the solution from the $\mathcal{L}_2$ or $\mathcal{L}_3$.

The simulator can successfully solve the hard problem if and only if all events happen simultaneously. Next, we analyze the probability of all events. From the private key query, we know when each $c_i = 1$, simulation will not abort, thus

$$\Pr[E_1] = \Pr[c_i = 1, i = 1, 2, \cdots, q_E] = (1-\delta)^{q_E}\,.$$

All c_i are chosen by the simulator where $c_i = 0$ with probability δ, $c_i = 1$ with probability $1-\delta$. When $c_i = 0$, the value of H_1 contains b, thus $\Pr[E_2] = \delta$. Since all c_i are chosen by the simulator and they are secretly to $\mathcal{A}$, the adversary does not know which c_i of each identity is equal to 0 or 1. That is, from the point view of the adversary, it does not know the probabilities of $c_i = 0$ and $c_i = 1$. Therefore,

under event E_2, we have

$$\begin{aligned}\Pr[E_3] &= \Pr[E_3|c_i = 0]\Pr[c_i = 0] + \Pr[E_3|c_i = 1]\Pr[c_i = 1] \\ &= \tfrac{1}{n-l}\Pr[c_i = 0] + \tfrac{1}{n-l}\Pr[c_i = 1] \\ &= \tfrac{1}{n-l} \geq \tfrac{1}{n}.\end{aligned}$$

Note that the identity $ID_i \in S^* \cap R^*$ allows to query the corresponding private key. In our setting, these identities cannot provide any help for $\mathcal{A}$ to distinguish the encrypted message in the challenge ciphertext. Since $|S^* \cap R^*| = l$, the potential useful identity is $n - l$. Thus we have above result $\Pr[E_3] = \frac{1}{n-l} \geq \frac{1}{n}$.

Finally, from the point view of the simulator, if $\mathcal{A}$ has non-negligible advantage to guess the correct μ' and with the conditions that E_1, E_2, E_3 happen, it only knows that the solution of the hard problem is in the $\mathcal{L}_2$ or $\mathcal{L}_3$, but it does not know which one is. Thus $\Pr[E_4] \geq \frac{1}{q_{H_2}+q_{H_3}}$. It is clear that these four events are independent, therefore, we have

$$\begin{aligned}\epsilon' &\geq \Pr[E_1 \wedge E_2 \wedge E_3 \wedge E_4] \cdot \epsilon \\ &= \Pr[E_1] \cdot \Pr[E_2] \cdot \Pr[E_3] \cdot \Pr[E_4] \cdot \epsilon \\ &\geq (1-\delta)^{q_E} \cdot \delta \cdot \tfrac{1}{n} \cdot \tfrac{1}{q_{H_2}+q_{H_3}} \cdot \epsilon \\ &= (1-\delta)^{q_E} \cdot \delta \cdot \tfrac{\epsilon}{n(q_{H_2}+q_{H_3})}.\end{aligned}$$

The function $(1-\delta)^{q_E} \cdot \delta$ is maximized at $\delta = \frac{1}{q_E+1}$, we have

$$(1-\delta)^{q_E} \cdot \delta = \frac{1}{q_E+1} \cdot \left(1 - \frac{1}{q_E+1}\right)^{q_E} = \frac{1}{q_E} \cdot \left(1 - \frac{1}{q_E+1}\right)^{q_E+1}.$$

For a large q_E, $\left(1 - \frac{1}{q_E+1}\right)^{q_E+1} \approx \frac{1}{e}$, thus we have

$$\epsilon' \geq (1-\delta)^{q_E} \cdot \delta \cdot \frac{\epsilon}{n(q_{H_2}+q_{H_3})} \approx \frac{\epsilon}{n \cdot e \cdot q_E \cdot (q_{H_2}+q_{H_3})}.$$

This completes the proof. □

Discussion. When $R^* = \emptyset$, the challenge message is encrypted by v^*. If the adversary can distinguish the message, the simulator can decide it must have made the challenge query to the H_3, but the simulator does not know which input contains the solution of the hard problem. In this case, $\Pr[E_4] = \frac{1}{q_{H_3}} \geq \frac{1}{q_{H_2}+q_{H_3}}$. When $R^* \neq \emptyset$, even the inputs of H_3 contain the hard problem, the adversary can retrieve v^* by the identity $ID_i \in S^* \cap R^*$. Thus the challenge query is from H_2 and $\Pr[E_4] = \frac{1}{q_{H_2}} \geq \frac{1}{q_{H_2}+q_{H_3}}$.

Theorem 3.2. *Let H_1, H_2, H_3 be random oracles. The proposed scheme is ANON-ID-CPA secure under the BDH assumption. Specifically, suppose there is an ANON-ID-CPA adversary who has advantage ϵ against our proposed scheme by making q_E private key queries and q_{H_1}, q_{H_2}, q_{H_3} queries to the functions H_1, H_2 and H_3 respectively. Then there is an algorithm $\mathcal{B}$ to solve the BDH problem with advantage*

$$\epsilon' \geq \frac{\epsilon}{n \cdot e \cdot q_E \cdot (q_{H_2} + q_{H_3})},$$

where n is the number of the identities stated in the ciphertext.

Proof. The proof is similar to the proof of Theorem 3.1. Suppose there exists an adversary $\mathcal{A}$ who can break the anonymity of our scheme with advantage ϵ. We build a simulator $\mathcal{B}$ that can solve the BDH problem with advantage ϵ' by running $\mathcal{A}$. Given a random instance of BDH problem (P, aP, bP, cP), $\mathcal{B}$'s goal is to compute $e(P,P)^{abc}$. In order to use $\mathcal{A}$ to solve the problem, $\mathcal{B}$ needs to simulate a challenger and respond all queries from $\mathcal{A}$. For ease of exposition, we assume that the H_2 and H_3 queries are after the H_1 query for the same identity. $\mathcal{B}$ works by interacting with $\mathcal{A}$ in an ANON-ID-CPA game as follows.

Setup: $\mathcal{B}$ sets $P_{pub} = aP$ and creates the master public as $mpk = (p, P, P_{pub}, e, H)$. Then it sends mpk to $\mathcal{A}$. Here, the hash functions H_1, H_2, H_3 are viewed as random oracles controlled by the simulator.

H_1-queries: $\mathcal{A}$ makes H_1 queries. $\mathcal{B}$ responds a query on ID_i as follows. $\mathcal{B}$ maintains a list $\mathcal{L}_1$ of a tuple (ID_i, c_i, r_i, h_i). This list is initially empty. $\mathcal{B}$ first checks $\mathcal{L}_1$. If the query ID_i has already appeared in the $\mathcal{L}_1$ in a tuple (ID_i, c_i, r_i, h_i), it returns the corresponding h_i as the value of $H_1(ID_i)$. Otherwise, $\mathcal{B}$ performs as follows.

1. Select $c_i \in_R \{0,1\}$ with $\Pr[c_i = 0] = \delta$ for some δ (determine later).
2. Pick $r_i \in_R \mathbb{Z}_p$, if $c_i = 0$, compute $h_i = r_i bP$. If $c_i = 1$, compute $h_i = r_i P$.
3. Add the tuple (ID_i, c_i, r_i, h_i) to the $\mathcal{L}_1$ and respond with h_i to $\mathcal{A}$.

H_2-queries: $\mathcal{A}$ makes H_2 queries. $\mathcal{B}$ responds a query on (X_i, ID_i) as follows. $\mathcal{B}$ maintains a list $\mathcal{L}_2$ of a tuple (X_i, ID_i, λ_i). This list is initially empty. $\mathcal{B}$ first checks $\mathcal{L}_2$. If the query (X_i, ID_i) has already appeared in the $\mathcal{L}_2$ in a tuple (X_i, ID_i, λ_i), it returns the corresponding λ_i as the value of $H_2(X_i, ID_i)$. Otherwise, $\mathcal{B}$ randomly picks a $\lambda_i \in \mathbb{G}$ as the value of $H_2(X_i, ID_i)$, adds the tuple (X_i, ID_i, λ_i) to the $\mathcal{L}_2$ and responds to $\mathcal{A}$ with λ_i.

H_3-queries: $\mathcal{A}$ makes H_3 queries. $\mathcal{B}$ responds a query on (Y_i, ID_i) as follows. $\mathcal{B}$ maintains a list $\mathcal{L}_3$ of a tuple (Y_i, ID_i, γ_i). This list is initially empty. $\mathcal{B}$ first checks

$\mathcal{L}_3$. If the query (Y_i, ID_i) has already appeared in the $\mathcal{L}_3$ in a tuple (Y_i, ID_i, γ_i), it returns the corresponding γ_i as the value of $H_3(Y_i, ID_i)$. Otherwise, $\mathcal{B}$ randomly picks a $\gamma_i \in \mathbb{G}$ as the value of $H_3(Y_i, ID_i)$, adds the tuple (Y_i, ID_i, γ_i) to the $\mathcal{L}_3$ and responds to $\mathcal{A}$ with γ_i.

Phase 1: In this phase, $\mathcal{A}$ issues the private key queries on ID_i as needed. For each time, $\mathcal{B}$ first runs the H_1 query to get the corresponding c_i and r_i. If $c_i = 0$, $\mathcal{B}$ aborts. If $c_i = 1$, $\mathcal{B}$ computes $d_{ID_i} = sH_1(ID_i) = ar_iP = r_iP_{pub}$.

Challenge: When $\mathcal{A}$ decides that the phase 1 is over, it outputs a challenge message M^*, two distinct identity sets $S_0 = (ID_{0,1}, ID_{0,2}, \cdots, ID_{0,n})$, $S_1 = (ID_{1,1}, ID_{1,2}, \cdots, ID_{1,n})$ and a revoked identity set $R^* = (ID'_1, ID'_2, \cdots, ID'_t)$. We require that any identity $ID_i \in (S_0 \cup S_1) \backslash (S_0 \cap S_1)$ has not been queried the private key in the phase 1. $\mathcal{B}$ picks a random bit $\mu \in \{0, 1\}$ and performs as follows.

1. Pick $r^* \in_R \mathbb{Z}_p$, $v^* \in \mathbb{G}$, compute $C_0^* = v^* + M$, $C_1^* = r^*cP$, $C_2^* = cP$.

2. For each $ID_i \in S_\mu \backslash (S_0 \cap S_1)$, $\mathcal{B}$ randomly chooses $A_i^*, B_i^* \in \mathbb{G}$. For each $ID_i \in S_0 \cap S_1$, $\mathcal{B}$ first gets r_i from the $\mathcal{L}_1$ (If ID_i is not in the $\mathcal{L}_1$, run H_1 queries to get r_i). Then it computes $X_i = e(aP, cP)^{r^* r_i}$ and checks whether the tuple (X_i, ID_i) is in the $\mathcal{L}_2$. If yes, it obtains the corresponding λ_i and sets $A_i^* = \lambda_i$. Otherwise, it randomly chooses $A_i^* \in \mathbb{G}$ and adds the new tuple (X_i, ID_i, A_i^*) to the $\mathcal{L}_2$. Then $\mathcal{B}$ computes $Y_i = e(aP, cP)^{r_i}$ and checks whether the tuple (Y_i, ID_i) in the $\mathcal{L}_3$. If yes, it obtains the corresponding γ_i and sets $w_i^* = \gamma_i$. Otherwise, it randomly chooses $w_i^* \in \mathbb{G}$ and adds the new tuple (Y_i, ID_i, w_i^*) to the $\mathcal{L}_3$, and computes $B_i^* = w_i^* + v^*$.

3. For each $i = 1, 2 \cdots, n$, compute $x_i^* = H(ID_i)$,

$$f_i(x) = \prod_{j=1, j \neq i}^{n} \frac{x - x_j^*}{x_i^* - x_j^*} = \sum_{j=0}^{n-1} a_{i,j} x^j,$$

$$Q_i^* = \sum_{j=1}^{n} a_{j,i-1} A_j^*, \quad U_i^* = \sum_{j=1}^{n} a_{j,i-1} B_j^*,$$

and set $CT = (R^*, C_0^*, C_1^*, C_2^*, Q_i^*, U_i^*, i \in [1, n])$.

Case 1: $R^* = \emptyset$. $\mathcal{B}$ sets the challenge ciphertext $CT^* = CT$.

Case 2: $R^* \neq \emptyset$. $\mathcal{B}$ performs as follows.

1. Randomly choose $u^* \in \mathbb{G}$ and compute $C_0'^* = u^* + C_0^*$.

2. For each $ID_i \in R^*$, $\mathcal{B}$ computes $x_i'^* = H(ID_i)$ and computes

$$g(x) = \prod_{i=1}^{t}(x - x_i'^*) = \sum_{i=0}^{t} b_i x^i.$$

Then it sets $b_i = 0$ for $i = t+1, t+2, \cdots, n-1$.

3. For $i = 1, 2, \cdots, n$, $\mathcal{B}$ computes

$$Q_i'^* = Q_i^* + b_{i-1}u^*,$$

and sets $CT^* = (R^*, C_0'^*, C_1^*, C_2^*, Q_i'^*, U_i^*, i \in [1, n])$.

Phase 2: $\mathcal{A}$ continues to issue more private key queries with the restriction established in the challenge phase. $\mathcal{B}$ responds the same as in phase 1.

Guess: Finally, $\mathcal{A}$ outputs its guess $\mu' \in \{0, 1\}$ of μ.

From the scheme construction, we note that the adversary $\mathcal{A}$ can obtain the private keys for the identity ID_i where $ID_i \in S_0 \cap S_1$, then it computes v^* correctly through the decryption algorithm. If for the identities $ID_i, ID_j \in S_0 \cap S_1$, where $i \neq j$, $v_i \neq v_j$. $\mathcal{A}$ can distinguish the simulation from the real scheme and aborts immediately.

Similarly, for $ID_i \in S_0 \cap S_1 \cap R^*$, $\mathcal{A}$ can get A_i^* correctly through the challenge ciphertext. Meanwhile, $\mathcal{A}$ can use the private key to compute A_i^*. If both results are not equal for the same identity, $\mathcal{A}$ can distinguish the simulation from the real scheme and aborts immediately. Additionally, for $ID_i \in S_0 \cap S_1$, but $ID_i \notin R^*$, $\mathcal{A}$ can compute u^* correctly. For different identities ID_i, if $\mathcal{A}$ gets different u^*, it can distinguish the simulation from the real scheme and aborts immediately. Thus, in the security proof, we should take these issues into consideration. The settings in our proof can address these issues perfectly. As the same analysis in Theorem 3.1, the challenge ciphertext is a one-time pad unless the adversary has made the challenge query. According the assumption at the beginning of the proof, the adversary will make the challenge query to break the scheme.

When $c_j = 0$, we have $H_1(ID_j) = r_j bP$ and $d_{ID_j} = r_j abP$. From the decryption algorithm, we have $e(C_1^*, d_{ID_j}) = e(r^* cP, r_j abP) = e(P, P)^{r^* r_j abc}$ and $e(C_2^*, d_{ID_j}) = e(cP, r_j abP) = e(P, P)^{r_j abc}$. Here $\mathcal{B}$ ignores the guess of $\mathcal{A}$ and picks a random tuple from $\mathcal{L}_2$ or $\mathcal{L}_3$. It first obtains the corresponding r_j from the $\mathcal{L}_1$. If $\mathcal{B}$ picks the tuple (X_j, ID_j, λ_j) from $\mathcal{L}_2$, it computes $X_j^{(r^* r_j)^{-1}}$ as the solution to the given instance of BDH problem. If $\mathcal{B}$ picks the tuple (Y_j, ID_j, γ_j) from $\mathcal{L}_3$, it computes $Y_j^{r_j^{-1}}$ as the solution to the given instance of BDH problem.

The above completes the description of the simulation. To complete the security

proof, it remains to show that $\mathcal{B}$ correctly computes $e(P,P)^{abc}$ with advantage at least ϵ'. We first define the following events.

E_1: The simulation does not abort in private key query.

E_2: At least one of the H_1 values of challenge identities contains b.

E_3: The adversary chooses an identity where $c_i = 0$ to distinguish the challenge identity sets.

E_4: The simulator correctly chooses the solution from $\mathcal{L}_2$ or $\mathcal{L}_3$.

The simulator can successfully solve the hard problem if and only if all events happen simultaneously. Next, we analyze the probability of all events. From the private key queries, we know when each $c_i = 1$, simulation will not abort, thus

$$\Pr[E_1] = \Pr[c_i = 1, i = 1, 2, \cdots, q_E] = (1-\delta)^{q_E}.$$

All c_i are chosen by the simulator where $c_i = 0$ with probability δ, $c_i = 1$ with probability $1-\delta$. When $c_i = 0$, the value of H_1 contains b, thus $\Pr[E_2] = \delta$. Since all c_i are chosen by a certain probability which is decided by the simulator and they are secretly to the adversary and $\mathcal{A}$ does not know which identity's c_i is equal to 0 or 1. That is, from the point view of the adversary, it does not know the probabilities of $c_i = 0$ and $c_i = 1$. Therefore, under event E_2, we have

$$\begin{aligned}
&\Pr[E_3] \\
&= \Pr[E_3|c_i = 0]\Pr[c_i = 0] + \Pr[E_3|c_i = 1]\Pr[c_i = 1] \\
&\geq \tfrac{1}{n}\Pr[c_i = 0] + \tfrac{1}{n}\Pr[c_i = 1] \\
&= \tfrac{1}{n} \cdot \big(\Pr[c_i = 0] + \Pr[c_i = 1]\big) \\
&= \tfrac{1}{n}.
\end{aligned}$$

Note that the identity $ID_i \in S_0 \cap S_1$ allows to query the corresponding private keys, these identities cannot provide any help for $\mathcal{A}$ to distinguish the challenge identity sets in our setting. If $|S_0 \cap S_1| = k$, the potential useful identity is $n-k$. Thus we have above result $\Pr[E_3] = \frac{1}{n-k} \geq \frac{1}{n}$.

Finally, from the point view of the simulator, if the adversary has non-negligible advantage to guess the correct μ' and with the conditions that E_1, E_2, E_3 happen, it only knows that the solution of the hard problem is embed in the H_2 query or H_3 query, thus $\Pr[E_4] \geq \frac{1}{q_{H_2}+q_{H_3}}$. It is clear that these four events are independent,

therefore, we have

$$\begin{aligned}\epsilon' &\geq \Pr[E_1 \wedge E_2 \wedge E_3 \wedge E_4] \cdot \epsilon \\ &= \Pr[E_1] \cdot \Pr[E_2] \cdot \Pr[E_3] \cdot \Pr[E_4] \cdot \epsilon \\ &\geq (1-\delta)^{q_E} \cdot \delta \cdot \tfrac{1}{n} \cdot \tfrac{1}{q_{H_2}+q_{H_3}} \cdot \epsilon \\ &= (1-\delta)^{q_E} \cdot \delta \cdot \tfrac{\epsilon}{n(q_{H_2}+q_{H_3})}.\end{aligned}$$

The function $(1-\delta)^{q_E} \cdot \delta$ is maximized at $\delta = \frac{1}{q_E+1}$, we have

$$(1-\delta)^{q_E} \cdot \delta = \frac{1}{q_E+1} \cdot \left(1 - \frac{1}{q_E+1}\right)^{q_E} = \frac{1}{q_E} \cdot \left(1 - \frac{1}{q_E+1}\right)^{q_E+1}.$$

For a large q_E, $\left(1 - \frac{1}{q_E+1}\right)^{q_E+1} \approx \frac{1}{e}$, thus we have

$$\epsilon' \geq (1-\delta)^{q_E} \cdot \delta \cdot \frac{\epsilon}{n(q_{H_2}+q_{H_3})} \approx \frac{\epsilon}{e \cdot q_E \cdot n \cdot (q_{H_2}+q_{H_3})}.$$

This completes the proof. □

3.5 Conclusion

In this chapter, we considered the receiver privacy in the recipient revocable identity-based broadcast encryption for the first time and described a new construction of anonymous revocable identity-based broadcast encryption. The receivers' identity information in the proposed scheme can be protected well. Not only the message but also the receivers' identities are hidden to the third party. The security of the proposed scheme is based on the hardness of the Bilinear Diffie-Hellman (BDH) problem in the random oracle model.

Chapter 4

Fully Privacy-Preserving Revocable IBBE

This chapter describes a fully privacy-preserving revocable identity-based broadcast encryption scheme. The proposed scheme not only protects the receiver privacy, but also preserves the identity information of the revoked users. The original scheme was published in the *Personal and Ubiquitous Computing* [LMG+17].

4.1 Introduction

In chapter 3, we continued to study the recipient revocable identity-based broadcast encryption and proposed the first anonymous revocable identity-based broadcast encryption (AR-IBBE) scheme. In the proposed AR-IBBE, the receiver identity information can be protected well. One limitation is that the privacy of the revoked users has not been considered. To perform decryption successfully, the revoked user identity information should be attached as part of ciphertext and known publicly, which might not be desired in some applications. For example, if the original ciphertext is generated for a group of users who have featured some special attribute. When some of the receivers have been revoked, the special attribute held by the whole receivers will be exposed to the public from the revoked user if the identity information of them are not protected. As a consequence, the receivers' information are leaked. This motivates us to design a scheme with fully user privacy-preserving.

In this chapter, we present a revocable identity-based broadcast encryption scheme which can fully preserve the user privacy. Both the identity information of the receivers and the revoked users in the proposed scheme can be protected. The encrypted message can be securely protected and only the authorized user can retrieve it. The revocation process does not reveal any information about the message and the receivers' identities. The public learns nothing about the identities of receivers and the revoked users. The proposed scheme is still expressive enough for practical scenarios, such as in the smart city. The security of our proposed scheme is proved to be semantically secure in the random oracle model.

Organization. The rest of this chapter is organized as follows. In the next Section, we give the definitions of fully privacy-preserving revocable identity-based broadcast

encryption and the corresponding security models. The concrete construction is described in Section 4.3. In Section 4.4, we provide the security analysis of the proposed scheme under the defined security models, and concludes this chapter in Section 4.5.

4.2 Definition and Security Models

In this section, we will define the syntax and the security of fully privacy-preserving revocable identity-based broadcast encryption (FPPR-IBBE) following [LMG+16]. Roughly speaking, an FPPR-IBBE scheme should preserve the privacy of the receivers and the revoked users. It consists of the following five algorithms.

- **Setup**(1^λ). Taking as input a security parameter λ, the setup algorithm returns a master public key mpk and a master secret key msk. The mpk is publicly known while the msk is kept secretly.
- **KeyGen**(mpk, msk, ID). Taking as input the master key pair (msk, mpk) and a user identity ID, the key generation algorithm returns a user private key d_{ID}.
- **Encrypt**(mpk, M, S). Taking as input the master public key mpk, a message M and a set of identities $S = (ID_1, ID_2, ..., ID_n)$, the encryption algorithm returns a ciphertext CT.
- **Revoke**(mpk, R, CT). Taking the master public key mpk, a ciphertext CT and a revoked identity set $R = (ID_{l_1}, ID_{l_2}, \cdots, ID_{l_t})$ with $t < n$, the revocation algorithm returns a new ciphertext CT'.
- **Decrypt**(CT', ID, d_{ID}). Taking as input the master public key mpk, a ciphertext CT', an identity ID and the corresponding private key d_{ID}, the decryption returns a message M if $ID \in S \backslash R$ and $\perp$ otherwise.

Correctness. The same as Chapter 3, if $R = \emptyset$, the FPPR-IBBE scheme is an anonymous IBBE scheme and we set $CT' = CT$. Thus, for correctness, it requires that for any message M from its message space, if $(mpk, msk) \leftarrow \mathsf{Setup}(1^\lambda)$, $d_{ID} \leftarrow \mathsf{KeyGen}(mpk, msk, ID)$, $CT \leftarrow \mathsf{Encrypt}(mpk, M, S)$, $CT' \leftarrow \mathsf{Revoke}\ (mpk, R, CT)$, we have

$$\mathsf{Decrypt}(mpk, CT', ID, d_{ID}) = \begin{cases} M & \text{If } ID \in S \backslash R, \\ \perp & \text{otherwise.} \end{cases}$$

Security Notions. Now, we formalize the security models for an FPPR-IBBE scheme. In an FPPR-IBBE scheme, the encrypted data firstly will be sent to a third party (who performs the revocation procedure). Hence apart from the requirement that the ciphertext CT' preserves the message and the receiver privacy against the public, the message should also be unpredictable from CT and CT should preserve the receiver privacy against the third party. More specifically, the indistinguishability security of an FPPR-IBBE scheme requirements are as follows.

1. The message and the identity set in the ciphertext CT cannot be distinguished without a valid private key associated with an identity $ID \in S$.

2. The message in the CT' cannot be distinguished without a valid private key associated with an identity $ID \in S \backslash R$.

3. The revoked identity set in CT' cannot be distinguished without a valid non-trivial private key.

Note that the security of CT is similar to the security of anonymous identity-based broadcast encryption scheme [LPQ12] where the encryption of unpredictable message must be indistinguishable from a random string of the same length and the receivers identities must be indistinguishable from a random identity set with the same length. We follow [LPQ12] to define four security models to capture the security requirements of the FPPR-IBBE scheme, namely the IND-ID-CPA security, ANON-ID-CPA security, IND-rID-CPA security and selective ANON-rID-CPA. These four security models are defined under the following games between a challenger and an adversary in each model.

Game 1 (IND-ID-CPA Security). This security model claims that without a valid private key, the message in the CT is indistinguishable from a random string of the same length and it works as follows.

- **Setup:** The challenger runs the setup algorithm to generate the master public key mpk and gives mpk to the adversary.

- **Phase 1:** In this phase, the adversary can issue private key queries as needed. Upon receiving a private key query for ID_i, the challenger runs the key generation algorithm to generate the private key d_{ID_i} and responds by returning d_{ID_i}.

- **Challenge:** Once the adversary decides that the phase 1 is over, it outputs two distinct messages M_0, M_1 from the same message space and a challenge identity set $S^* = (ID_1, ID_2, \cdots, ID_n)$. We require that the adversary has not queried the private key for any $ID_i \in S^*$ in the phase 1. The challenger

randomly picks a bit $\mu \in \{0,1\}$ and generates the challenge ciphertext CT^* for message M_μ under S^*, then it returns CT^* to the adversary.

– **Phase 2:** The adversary continues to issue more private key queries with the restriction established in the challenge phase. The challenge responds the same as in phase 1.

– **Guess:** Finally, the adversary outputs its guess $\mu' \in \{0,1\}$ of μ and wins the game if $\mu = \mu'$.

We refer to such an adversary as an IND-ID-CPA adversary and define the adversary's advantage in winning the game as

$$\mathsf{Adv}^{\mathsf{IND\text{-}ID\text{-}CPA}}_{\mathsf{FPPR\text{-}IBBE}}(\lambda) = \left| \Pr[\mu = \mu'] - \frac{1}{2} \right|.$$

Definition 4.1. *We say that an FPPR-IBBE scheme is IND-ID-CPA secure if for any probabilistic polynomial time adversary, the advantage* $\mathsf{Adv}^{\mathsf{IND\text{-}ID\text{-}CPA}}_{\mathsf{FPPR\text{-}IBBE}}(\lambda)$ *in the* ***Game 1*** *is negligible.*

Game 2 (ANON-ID-CPA Security). This security model claims that the receiver set in CT is indistinguishable from a random identity set of the same length and it works as follows.

– **Setup:** The challenger runs the setup algorithm to generate the master public key mpk and gives mpk to the adversary.

– **Phase 1:** In this phase, the adversary can issue private key queries as needed. Upon receiving a private key query for ID_i, the challenger runs the key generation algorithm to generate the private key d_{ID_i} and responds by returning d_{ID_i}.

– **Challenge:** Once the adversary decides that the phase 1 is over, it outputs a message M^* and two distinct identity sets $S_0 = (ID_{0,1}, ID_{0,2}, ..., ID_{0,n})$, $S_1 = (ID_{1,1}, ID_{1,2}, ..., ID_{1,n})$. We require that the adversary has not issued the private key queries for any $ID_i \in S_0 \triangle S_1 = (S_0 \backslash S_1) \cup (S_1 \backslash S_0)$ in the phase 1. The challenger randomly picks a bit $\mu \in \{0,1\}$ and generates the challenge ciphertext CT^* for message M^* under S_μ.

– **Phase 2:** The adversary continues to issue more private key queries with the restriction established in the challenge phase. The challenger responds the same as in phase 1.

– **Guess:** Finally, the adversary outputs its guess $\mu' \in \{0,1\}$ of μ and wins the game if $\mu = \mu'$.

We refer to such an adversary as an ANON-ID-CPA adversary and define the adversary's advantage in winning the game as

$$\mathsf{Adv}^{\mathsf{ANON\text{-}ID\text{-}CPA}}_{\mathsf{FPPR\text{-}IBBE}}(\lambda) = \left| \Pr\left[\mu = \mu'\right] - \frac{1}{2} \right|.$$

Definition 4.2. *We say that an FPPR-IBBE scheme is ANON-ID-CPA secure if for any probabilistic polynomial time adversary, the advantage* $\mathsf{Adv}^{\mathsf{ANON\text{-}ID\text{-}CPA}}_{\mathsf{FPPR\text{-}IBBE}}(\lambda)$ *in the* ***Game 2*** *is negligible.*

Game 3 (IND-rID-CPA Security). This security model claims that without a valid private key, the message in CT' is indistinguishable from a random string of the same length and it works as follows.

- **Setup:** The challenger runs the setup algorithm to generate the master public key mpk and gives mpk to the adversary.
- **Phase 1:** In this phase, the adversary can issue private key queries as needed. Upon receiving a private key query for ID_i. The challenger runs the key generation algorithm to generate the private key d_{ID_i} and responds by returning d_{ID_i}.
- **Challenge:** Once the adversary decides that the phase 1 is over, it outputs two distinct messages M_0, M_1 from the same message space, a challenge identity set $S^* = (ID_1, ID_2, \cdots, ID_n)$ and a revoked identity set $R^* = (ID_{l_1}, ID_{l_2}, \cdots, ID_{l_t})\,(t < n)$ with the restriction that the adversary has not queried the private key for any $ID_i \in S^* \backslash R^*$ in the phase 1. The challenger randomly picks a bit $\mu \in \{0, 1\}$ and runs the algorithms of encryption and revocation to generate the challenge ciphertext CT'^* for the message M_μ under S^* and R^*, then it returns CT'^* to the adversary.
- **Phase 2:** The adversary can issue more private key queries as needed with the restriction established in the challenge phase. The challenger responds the same as in phase 1.
- **Guess:** Finally, the adversary outputs its guess $\mu' \in \{0, 1\}$ of μ and wins the game if $\mu = \mu'$.

We refer to such an adversary as an IND-rID-CPA adversary and define adversary's advantage in winning the game as

$$\mathsf{Adv}^{\mathsf{IND\text{-}rID\text{-}CPA}}_{\mathsf{FPPR\text{-}IBBE}}(\lambda) = \left| \Pr\left[\mu = \mu'\right] - \frac{1}{2} \right|.$$

Definition 4.3. *We say that an FPPR-IBBE scheme is IND-rID-CPA secure if for any probabilistic polynomial time adversary, the advantage* $\mathsf{Adv}^{\mathsf{IND\text{-}aID\text{-}CPA}}_{\mathsf{FPPR\text{-}IBBE}}(\lambda)$ *in the* ***Game 3*** *is negligible.*

Game 4 (Selective ANON-rID-CPA Security). This security model claims that given two equal-length distinct revoked identity sets, it is hard to distinguish that CT' is generated under which one without a valid non-trivial private key. It works as follows.

- **Init:** The adversary outputs two distinct revoked identity sets $R_0 = (ID_{0,1}, ID_{0,2}, ..., ID_{0,t})$, $R_1 = (ID_{1,1}, ID_{1,2}, ..., ID_{1,t})$ that it wants to attack.
- **Setup:** The challenger runs the setup algorithm to generate the master public key mpk and gives mpk to the adversary.
- **Phase 1:** In this phase, the adversary can issue private key queries with the restriction that $ID_i \notin R_0 \triangle R_1$. Upon receiving a private key query for ID_i. The challenger runs the key generation algorithm to generate the private key d_{ID_i} and responds by returning d_{ID_i}.
- **Challenge:** Once the adversary decides that the phase 1 is over, it outputs a message M^* and an identity set $S^* = (ID_1, ID_2, \cdots, ID_n)$ where $n > t$. The challenger randomly picks a bit $\mu \in \{0, 1\}$ and generates the challenge ciphertext CT'^* for message M^* under S^* and R_μ.
- **Phase 2:** The adversary can issue more private key queries for $ID_i \notin R_0 \triangle R_1$, $\mathcal{B}$ responds the same as in phase 1.
- **Guess:** Finally, the adversary outputs its guess $\mu' \in \{0, 1\}$ of μ and wins the game if $\mu = \mu'$.

We refer to such an adversary as an ANON-rID-CPA adversary and define the adversary's advantage in winning the game as

$$\mathsf{Adv}^{\mathsf{ANON\text{-}rID\text{-}CPA}}_{\mathsf{FPPR\text{-}IBBE}}(\lambda) = \left| \Pr[\mu = \mu'] - \frac{1}{2} \right|.$$

Definition 4.4. *We say that an FPPR-IBBE scheme is selective ANON-rID-CPA secure if for any probabilistic polynomial time ANON-rID-CPA adversary, the advantage* $\mathsf{Adv}^{\mathsf{ANON\text{-}rID\text{-}CPA}}_{\mathsf{FPPR\text{-}IBBE}}(\lambda)$ *in the* ***Game 4*** *is negligible.*

4.3 The Proposed Scheme

In this section, we present the construction of our proposed scheme.

4.3.1 Construction

Setup(1^λ). Given a security parameter λ, the setup algorithm randomly chooses a bilinear group $\mathbb{BG} = (\mathbb{G}, \mathbb{G}_T, e, p)$ with generator $P \in \mathbb{G}$. It picks $s \in \mathbb{Z}_p$ and sets $P_{pub} = sP$. Then it chooses four cryptographic hash functions $H : \{0,1\}^* \to \mathbb{G}$, $H_1 : \mathbb{G}_T \times \{0,1\}^* \to \mathbb{Z}_p$, $H_2 : \mathbb{G}_T \times \{0,1\}^* \to \mathbb{G}$, and $H_3 : \mathbb{G}_T \times \{0,1\}^* \to \mathbb{G}$. The master public key and the master secret key are

$$mpk = (\mathbb{BG}, P, P_{pub}, H, H_1, H_2, H_3)\,,\ msk = s.$$

KeyGen(mpk, msk, ID). Given the master key pair (mpk, msk) and a user identity $ID \in \{0,1\}^*$, the key generation algorithm returns a user private key as

$$d_{ID} = sH(ID).$$

Encrypt(mpk, M, S). Given the master public key mpk, a message $M \in \mathbb{G}$ and an identity set $S = (ID_1,\ ID_2, \ldots, ID_n)$, the encryption algorithm chooses a dummy user denoted as $ID_0 \notin S$ and performs as follows.

1. Randomly choose an encryption key $K_1 \in \mathbb{G}$ and random numbers $r_1, r_2, r_3 \in \mathbb{Z}_p$, compute

$$C_0 = K_1 + M,\ C_1 = r_1 P,\ C_2 = r_2 P,\ C_3 = r_3 P.$$

2. For each $i = [0, n]$, compute

$$x_i = H_1\left(e\big(H(ID_i), P_{pub}\big)^{r_1}, ID_i\right).$$

Then it constructs polynomial functions as follow

$$f_i(x) = \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j} = \sum_{j=0}^{n} a_{i,j} x^j \mod p,$$

and computes

$$A_i = H_2\Big(e\big(H(ID_i), P_{pub}\big)^{r_2}, ID_i\Big),$$

$$B_i = K_1 + H_3\Big(e\big(H(ID_i), P_{pub}\big)^{r_3}, ID_i\Big).$$

Finally, it computes

$$Q_i = \sum_{j=0}^{n} a_{j,i} A_j, \quad U_i = \sum_{j=0}^{n} a_{j,i} B_j.$$

The output ciphertext is $CT = (C_0, C_1, C_2, C_3, r_1, [Q_i, U_i]_{i=0}^n)$.

Revoke(mpk, CT, R). Given a ciphertext CT which is parsed as $CT = (C_0, C_1, C_2, r_1, [Q_i, U_i]_{i=0}^n)$, the master public key mpk and a revoked identity set $R = (ID_{l_1}, ID_{l_2}, \cdots, ID_{l_t})$ where $t < n$. If $R = \emptyset$, the revocation algorithm sets the new ciphertext $CT' = CT$. Otherwise, it does as follows.

1. Randomly choose $K_2 \in \mathbb{G}$ and computes $C_0' = K_2 + C_0$.
2. For each $ID_i \in R$, compute $x_i = H_1\Big(e\big(H(ID_i), P_{pub}\big)^{r_1}, ID_i\Big)$ and construct

$$g(x) = \prod_{i=1}^{t}(x - x_i) = \sum_{i=0}^{t} b_i x^i \mod p.$$

3. For $i = 0, 1, 2, \cdots, t$ compute

$$Q_i' = Q_i + b_i K_2.$$

and set the new ciphertext as $CT' = (C_0', C_1, C_2, C_3, b_0, b_1, \cdots, b_{t-1}, Q_0', Q_1', \cdots, Q_t', Q_{t+1}, \cdots, Q_n, U_0, U_1, \cdots, U_n)$.

Decrypt$(mpk, CT'ID_i, d_{ID_i})$. Given a ciphertext parsed CT' which is parsed as $(C_0', C_1, C_2, C_3, b_0, b_1, \cdots, b_{t-1}, Q_0', Q_1', \cdots, Q_t', Q_{t+1}, \cdots, Q_n, U_0, \cdots, U_n)$, the master public key mpk, an identity ID_i and the corresponding private key d_{ID_i}, the decryption algorithm performs as follows.

1. Compute

$$x_i = H_1\big(e(C_1, d_{ID_i}), ID_i\big),$$

$$g(x_i) = \sum_{j=0}^{t-1} b_j x_i^j + x_i^t \mod p.$$

2. If $g(x_i) = 0$, it aborts, otherwise, it computes

$$U = U_0 + x_i U_1 + x_i^2 U_2 + \cdots + x_i^n U_n,$$

$$Q = Q_0' + x_i Q_1' + x_i^2 Q_2' + \cdots + x_i^t Q_t' + x_i^{t+1} Q_{t+1} + \cdots + x_i^n Q_n.$$

3. Use the private key d_{ID_i} to recover the encryption keys by computing

$$K_1' = U - H_3\big(e(C_3, d_{ID_i}), ID_i\big),$$

$$K_2' = g(x_i)^{-1}\left(Q - H_2\big(e(C_2, d_{ID_i}), ID_i\big)\right),$$

and obtain the message $M' = C_0' - K_1' - K_2'$.

If $ID_i \in S \backslash R$, we have $K_1' = K_1$, $K_2' = K_2$ and can obtain the message M correctly.

4.3.2 Correctness and Discussion

Now, we give the correctness checking of the proposed scheme. For a user with identity $ID_i \in S$, we have

$$\begin{aligned} H_1\big(e(C_1, d_{ID_i}), ID_i\big) &= H_1\big(e(r_1 P, sH(ID_i)), ID_i\big) \\ &= H_1\Big(e\big(sP, H(ID_i)\big)^{r_1}, ID_i\Big) \\ &= H_1\Big(e\big(P_{pub}, H(ID_i)\big)^{r_1}, ID_i\Big) \\ &= x_i. \end{aligned}$$

After getting x_i by using its private key, the user computes

$$\begin{aligned} Q &= Q_0' + x_i Q_1' + x_i^2 Q_2' + \cdots + x_i^t Q_t' + x_i^{t+1} Q_{t+1} + \cdots + x_i^n Q_n \\ &= (Q_0 + x_i Q_1 + x_i^2 Q_2 + \cdots + x_i^n Q_n) + (b_0 + b_1 x_i + b_2 x_i^2 + \cdots + b_t x_i^t)\, u \\ &= (a_{0,0}A_0 + a_{1,0}A_1 + a_{2,0}A_2 + \cdots + a_{n,0}A_n) \\ &\quad + x_i\,(a_{0,1}A_0 + a_{1,1}A_1 + a_{2,1}A_2 + \cdots + a_{n,1}A_n) + \cdots \\ &\quad + x_i^n\,(a_{0,n}A_0 + a_{1,n}A_1 + a_{2,n}A_2 + \cdots + a_{n,n}A_n) + g(x_i)u \\ &= (a_{0,0} + a_{0,1}x_i + a_{0,2}x_i^2 + \cdots + a_{0,n}x_i^n)\, A_0 \\ &\quad + (a_{1,0} + a_{1,1}x_i + a_{1,2}x_i^2 + \cdots + a_{1,n}x_i^n)\, A_1 \\ &\quad + (a_{2,0} + a_{2,1}x_i + a_{2,2}x_i^2 + \cdots + a_{2,n}x_i^n)\, A_2 + \cdots \\ &\quad + (a_{n,0} + a_{n,1}x_i + a_{n,2}x_i^2 + \cdots + a_{n,n}x_i^n)\, A_n + g(x_i)u \\ &= f_0(x_i)A_0 + f_1(x_i)A_1 + f_2(x_i)A_2 + \cdots + f_n(x_i)A_n + g(x_i)u, \\ U &= U_0 + x_i U_1 + x_i^2 U_2 + \cdots + x_i^n U_n \\ &= (a_{0,0}B_0 + a_{1,0}B_1 + a_{2,0}B_2 + \cdots + a_{n,0}B_n) \\ &\quad + x_i(a_{0,1}B_0 + a_{1,1}B_1 + a_{2,1}B_2 + \cdots + a_{n,1}B_n) + \cdots \\ &\quad + x_i^n(a_{0,n}B_0 + a_{1,n}B_1 + a_{2,n}B_2 + \cdots + a_{n,n}B_n) \\ &= (a_{0,0} + a_{0,1}x_i + a_{0,2}x_i^2 + \cdots + a_{0,n}x_i^n)\, B_0 \\ &\quad + (a_{1,0} + a_{1,1}x_i + a_{1,2}x_i^2 + \cdots + a_{1,n}x_i^n)\, B_1 \\ &\quad + (a_{2,0} + a_{2,1}x_i + a_{2,2}x_i^2 + \cdots + a_{2,n}x_i^n)\, B_2 + \cdots \\ &\quad + (a_{n,0} + a_{n,1}x_i + a_{n,2}x_i^2 + \cdots + a_{n,n}x_i^n)\, B_n \\ &= f_0(x_i)B_0 + f_1(x_i)B_1 + f_2(x_i)B_2 + \cdots + f_n(x_i)B_n. \end{aligned}$$

Note that in our construction, $f_i(x_i) = 1$ and $f_i(x_j) = 0$ for $i \neq j$. Therefore, we have

$$\begin{aligned} Q &= f_0(x_i)A_0 + f_1(x_i)A_1 + f_2(x_i)A_2 + \cdots + f_n(x_i)A_n + g(x_i)u \\ &= A_i + g(x_i)u, \\ U &= f_0(x_i)B_0 + f_1(x_i)B_1 + f_2(x_i)B_2 + \cdots + f_n(x_i)B_n \\ &= B_i. \end{aligned}$$

Then K_1 can be obtained by computing

$$\begin{aligned} K_1' &= U - H_3\big(e(C_3, d_{ID_i}), ID_i\big) \\ &= B_i - H_3\Big(e\big(r_3 P, sH(ID_i)\big), ID_i\Big) \\ &= K_1 + H_3\Big(e\big(P, H(ID_i)\big)^{sr_3}, ID_i\Big) - H_3\Big(e\big(H(ID_i), P_{pub}\big)^{r_3}, ID_i\Big) \\ &= K_1. \end{aligned}$$

If $ID_i \in S \backslash R$, we have $g(x_i) \neq 0$ and K_2 can be obtained by computing

$$\begin{aligned} K_2' &= g(x_i)^{-1} \cdot \big(Q - H_2\big(e(C_2, d_{ID_i}), ID_i\big)\big) \\ &= g(x_i)^{-1} \cdot \big(A_i + g(x_i)u - H_2\big(e(C_2, d_{ID_i}), ID_i\big)\big) \\ &= g(x_i)^{-1} \cdot \Bigg(H_2\Big(e\big(H(ID_i), P_{pub}\big)^{r_2}, ID_i\Big) \\ &\qquad\qquad - H_2\Big(e\big(r_2 P, sH(ID_i)\big), ID_i\Big) + g(x_i)K_2\Bigg) \\ &= g(x_i)^{-1} \cdot \big(g(x_i)K_2\big) \\ &= K_2. \end{aligned}$$

After recovering K_1 and K_2, the user gets the message as

$$C_0' - K_1' - K_2' = M + K_1 + K_2 - K_1 - K_2 = M.$$

From the construction of our proposed scheme, one may observe that in the encryption algorithm, we choose a dummy user outside the identity set S. There is no such requirement in the work of Chapter 3. In the construction of Chapter 3, it is required that the size of broadcast identity set must be at least three, otherwise, the scheme is not secure in the case where there are only two user in S under the defined security models. In this scheme, we consider a more general situation and remove this restriction by using a dummy user which does not have any effect on the user decryption.

4.4 Security Analysis

In this section, we show that the proposed FPPR-IBBE scheme achieves the security requirements defined in the security models previously. The security of the proposed scheme is derived in the random oracle model under BDH assumption.

Theorem 4.1. *Let hash functions H, H_3 be random oracles. If the BDH assumption holds, the proposed scheme is IND-ID-CPA secure. Specifically, if there is an IDN-ID-CPA adversary $\mathcal{A}$ with advantage ϵ against our scheme, there is an algorithm $\mathcal{B}$*

that solves the BDH problem with advantage

$$\epsilon' \geq \frac{\epsilon}{e \cdot n \cdot q_E \cdot q_{H_3}},$$

where n is the number of the set S of identities, q_E and q_{H_3} are the number of private key queries and H_3 respectively. e is the natural logarithm.

Proof. Suppose there exists an IND-ID-CPA adversary $\mathcal{A}$ that break our scheme with non-negligible advantage ϵ. We build a simulator (algorithm) $\mathcal{B}$ that can solve the BDH problem with advantage ϵ' by running $\mathcal{A}$. Let (P, aP, bP, cP) be a random instance of BDH problem taken as input by $\mathcal{B}$ and its goal is to compute $e(P,P)^{abc}$. In order to use $\mathcal{A}$ to solve the problem, $\mathcal{B}$ needs to simulate a challenger and respond all queries from $\mathcal{A}$. $\mathcal{B}$ works by interacting with $\mathcal{A}$ in an IND-ID-CPA game (Game 1) as follows.

Setup: $\mathcal{B}$ sets $P_{pub} = aP$ and generates the master public key as $mpk = (\mathbb{BG}, P, P_{pub}, H_1, H_2)$. Here, H and H_3 are viewed as random oracles controlled by the simulator.

H-queries: For a query on ID_i, $\mathcal{B}$ responds as follows. $\mathcal{B}$ maintains a list $\mathcal{L}$ of tuples (ID_i, c_i, t_i, h_i). This list is initially empty. $\mathcal{B}$ first checks the $\mathcal{L}$. If the query ID_i has already appeared in $\mathcal{L}$ in a tuple (ID_i, c_i, t_i, h_i), it returns the corresponding h_i as the value of $H(ID_i)$. Otherwise, $\mathcal{B}$ randomly picks $t_i \in \mathbb{Z}_p$ and selects a random $c_i \in \{0,1\}$ with $Pr[c_i = 0] = \delta$ for some δ (determine later). If $c_i = 0$, it computes $h_i = t_i bP$, otherwise, it computes $h_i = t_i P$. Then it adds the tuple (ID_i, c_i, t_i, h_i) to the $\mathcal{L}$ and responds with h_i.

H_3-queries: For a query on (Y_i, ID_i), $\mathcal{B}$ responds as follows. $\mathcal{B}$ maintains a list $\mathcal{L}_3$ of tuples (Y_i, ID_i, γ_i). This list is initially empty. $\mathcal{B}$ first checks $\mathcal{L}_3$. If the query (Y_i, ID_i) has already appeared in $\mathcal{L}_3$ in a tuple (Y_i, ID_i, γ_i), it returns the corresponding γ_i as the value of $H_3(Y_i, ID_i)$. Otherwise, $\mathcal{B}$ randomly picks a $\gamma_i \in \mathbb{G}$ as the value of $H_3(Y_i, ID_i)$, then it adds (Y_i, ID_i, γ_i) to the $\mathcal{L}_3$ and responds with γ_i.

Phase 1: In this phase, the adversary can issue the private key queries on ID_i as needed. $\mathcal{B}$ firstly gets the corresponding c_i and t_i from $\mathcal{L}$. (If they do not exist, it runs the H query to get the corresponding c_i and t_i.) If $c_i = 0$, $\mathcal{B}$ aborts. If $c_i = 1$, $\mathcal{B}$ computes $d_{ID_i} = sH_1(ID_i) = at_iP = t_iP_{pub}$.

Challenge: Once $\mathcal{A}$ decides that the phase 1 is over, it outputs two distinct messages M_0, M_1 from the same message space and an identity set $S^* = (ID_1, ID_2, \cdots, ID_n)$. We require that $\mathcal{A}$ has not queried the private key for any $ID_i \in S^*$ in the phase 1. $\mathcal{B}$ picks a random bit $\mu \in \{0,1\}$ and performs as follows.

1. Choose a random dummy identity $ID_0 \notin S^*$ and randomly choose $B_i^* \in \mathbb{G}$ for $i = [0, n]$.

2. Randomly choose $r_1^*, r_2^* \in \mathbb{Z}_p$, $C_0^* \in \mathbb{G}$ and compute $C_1^* = r_1^* P$, $C_2^* = r_2^* P$, $C_3^* = cP$.

3. For $i = [0, n]$, get the value of $H(ID_i)$ from $\mathcal{L}$ (If ID_i does not exist in $\mathcal{L}$, run the H query) and compute

$$x_i^* = H_1\left(e\big(H(ID_i), P_{pub}\big)^{r_1^*}, ID_i\right),$$

$$A_i^* = H_2\left(e\big(H(ID_i), P_{pub}\big)^{r_2^*}, ID_i\right),$$

and construct polynomial functions

$$f_i(x) = \prod_{j=0, j\neq i}^{n} \frac{x - x_j^*}{x_i^* - x_j^*} = \sum_{j=0}^{n} a_{i,j} x^j.$$

Then it computes

$$Q_i^* = \sum_{j=0}^{n} a_{j,i} A_j^*, \quad U_i^* = \sum_{j=0}^{n} a_{j,i} B_j^*.$$

and sets the challenge ciphertext as $CT^* = (C_0^*, C_1^*, C_2^*, C_3^*, r_1^*, [Q_i^*, U_i^*]_{i=0}^n)$.

Phase 2: $\mathcal{A}$ continues to issue private key queries as needed with the restriction established in the challenge phase. $\mathcal{B}$ responds the same as in phase 1.

Guess: Finally, $\mathcal{A}$ outputs its guess $\mu' \in \{0, 1\}$ of μ.

The above completes the description of the simulation. From above setting, we note that the correctness and randomness hold. $\mathcal{B}$ simulates a real attack environment for the adversary $\mathcal{A}$. When $c_j = 0$, we have $H(ID_j) = t_j bP$, $d_{ID_j} = t_j abP$. Thus $e\left(d_{ID_j}, C_3^*\right) = e(P, P)^{t_j abc}$. According to the breaking assumption that the adversary will break the scheme with non-negligible advantage. At this point, $\mathcal{B}$ ignores the guess of $\mathcal{A}$ and picks a random tuple (Y_j, ID_j, γ_j) from the list $\mathcal{L}_3$. It then obtains the corresponding t_j from $\mathcal{L}$ and outputs $Y_j^{t_j^{-1}}$ as the solution to the given instance of BDH. To complete the security proof, it remains to show that $\mathcal{B}$ outputs the correct solution with advantage at least ϵ'.

The success of proof bases on the adversary's query on H_3. Let $W_i = \Big(e\big(H(ID_i), P_{pub}\big)^c, ID_i\Big)$ where $ID_i \in S^*$. In the real scheme, $B_i^* = K + H_3(W_i)$. Before querying the H_3 value of W_i, the result of $H_3(W_i)$ is unknown and random. From the view of the adversary, K is encrypted with a random number independent of W_i. Therefore,

B_i^* is a one-time pad. In other words, the challenge ciphertext is a one-time pad. According to the assumption ($\mathcal{A}$ breaks our scheme with advantage ϵ), the adversary must at least query H_3 on one W_i with probability δ. In this case, $\mathcal{B}$ decides the solution is in the $\mathcal{L}_3$. According to the above analysis, we define the following events.

E_1: The simulation does not abort in the private key query.

E_2: At least one of the H values of challenge identities contains b.

E_3: The adversary chooses an identity where $c_i = 0$ to distinguish challenge message.

E_4: The simulator correctly chooses the solution from the $\mathcal{L}_3$.

The simulator can successfully solve the hard problem if and only if all events happen simultaneously. Next, we analyze the probability of all events. From the private key query, we know when each $c_i = 1$, the simulation will not abort, thus

$$\Pr[E_1] = \Pr[c_i = 1, i = 1, 2, \cdots, q_E] = (1-\delta)^{q_E}.$$

For the event E_2, it is easy to compute that $\Pr[E_2] = \delta$. As c_i are secretly chosen by the simulator, from the point view of the adversary, it does not know the probabilities of $c_i = 0$ and $c_i = 1$. Therefore, we have

$$\begin{aligned}\Pr[E_3] &= \Pr[E_3|c_i = 0]\Pr[c_i = 0] + \Pr[E_3|c_i = 1]\Pr[c_i = 1] \\ &= \tfrac{1}{n}\Pr[c_i = 0] + \tfrac{1}{n}\Pr[c_i = 1] \\ &= \tfrac{1}{n}.\end{aligned}$$

Finally, from the point view of $\mathcal{B}$, if the adversary can guess μ' correctly, $\mathcal{B}$ only knows that the correct solution of the hard problem is in the $\mathcal{L}_3$, but it does not know which one is, thus $\Pr[E_4] = \frac{1}{q_{H_3}}$. It is not hard to see that these four events are independent, hence, we have

$$\begin{aligned}\epsilon' &= \Pr[E_1 \wedge E_2 \wedge E_3 \wedge E_4] \cdot \epsilon \\ &= \Pr[E_1] \cdot \Pr[E_2] \cdot \Pr[E_3] \cdot \Pr[E_4] \cdot \epsilon \\ &\geq (1-\delta)^{q_E} \cdot \delta \cdot \tfrac{\epsilon}{n \cdot q_{H_3}}.\end{aligned}$$

The function $(1-\delta)^{q_E} \cdot \delta$ is maximized at $\delta_{opt} = \frac{1}{q_E+1}$. Using δ_{opt}, we have

$$\epsilon' \geq \frac{\epsilon}{e \cdot n \cdot q_E \cdot q_{H_3}}.$$

This completes the proof. □

Theorem 4.2. *Let hash functions H, H_2, H_3 be random oracles. If the BDH assumption holds, the proposed scheme is ANON-ID-CPA secure. Specifically, if there is an ANON-ID-CPA adversary $\mathcal{A}$ with advantage ϵ against our scheme, there is an algorithm $\mathcal{B}$ that solves the BDH problem with advantage*

$$\epsilon' \geq \frac{\epsilon}{e \cdot n \cdot q_E \cdot (q_{H_2} + q_{H_3})},$$

where n is the number of the set S of identities, q_E, q_{H_2} and q_{H_3} are the number of queries to private key, H_2 and H_3 respectively. e is the natural logarithm.

Proof. Suppose there exists an ANON-ID-CPA adversary $\mathcal{A}$ that breaks our scheme with non-negligible advantage ϵ. We build a simulator (algorithm) $\mathcal{B}$ that can solve the BDH problem with advantage ϵ' by running $\mathcal{A}$. Let (P, aP, bP, cP) be a random instance of BDH problem taken as input by $\mathcal{B}$ and its goal is to compute $e(P,P)^{abc}$. In order to use $\mathcal{A}$ to solve the problem, $\mathcal{B}$ needs to simulate a challenger and respond all queries from $\mathcal{A}$. $\mathcal{B}$ works by interacting with $\mathcal{A}$ in an ANON-ID-CPA game (Game 2) as follows.

Setup: $\mathcal{B}$ sets $P_{pub} = aP$ and generates the master public key as $mpk = (\mathbb{BG}, P, P_{pub}, H_1)$. Here, we viewed hash functions H, H_2, H_3 are random oracles controlled by the simulator.

***H*-queries:** For a query on ID_i, $\mathcal{B}$ responds as follows. $\mathcal{B}$ maintains a list $\mathcal{L}$ of tuples (ID_i, c_i, t_i, h_i). This list is initially empty. $\mathcal{B}$ first checks the $\mathcal{L}$. If the query ID_i has already appeared in $\mathcal{L}$ in a tuple (ID_i, c_i, t_i, h_i), it returns the corresponding h_i as the value of $H(ID_i)$. Otherwise, $\mathcal{B}$ randomly picks $t_i \in \mathbb{Z}_p$ and selects a random $c_i \in \{0,1\}$ with $Pr[c_i = 0] = \delta$ for some δ (determine later). If $c_i = 0$, it computes $h_i = t_i bP$, otherwise, it computes $h_i = t_i P$. Then it adds the tuple (ID_i, c_i, t_i, h_i) to the $\mathcal{L}$ and responds with h_i.

***H_2*-queries:** For a query on (X_i, ID_i), $\mathcal{B}$ responds as follows. $\mathcal{B}$ maintains a list $\mathcal{L}_2$ of tuples (X_i, ID_i, λ_i). This list is initially empty. $\mathcal{B}$ first checks the $\mathcal{L}_2$. If the query (X_i, ID_i) has already appeared in the $\mathcal{L}_2$ in a tuple (X_i, ID_i, λ_i), it returns the corresponding λ_i as the value of $H_2(X_i, ID_i)$. Otherwise, $\mathcal{B}$ randomly picks a $\lambda_i \in \mathbb{G}$ as the value of $H_2(X_i, ID_i)$, then it adds (X_i, ID_i, λ_i) to the $\mathcal{L}_2$ and responds with λ_i.

***H_3*-queries:** For a query on (Y_i, ID_i), $\mathcal{B}$ responds as follows. $\mathcal{B}$ maintains a list $\mathcal{L}_3$ of tuples (Y_i, ID_i, γ_i). This list is initially empty. $\mathcal{B}$ first checks $\mathcal{L}_3$. If the query (Y_i, ID_i) has already appeared in $\mathcal{L}_3$ in a tuple (Y_i, ID_i, γ_i), it returns the corresponding γ_i as the value of $H_3(Y_i, ID_i)$. Otherwise, $\mathcal{B}$ randomly picks a $\gamma_i \in \mathbb{G}$ as the value of $H_3(Y_i, ID_i)$, then it adds (Y_i, ID_i, γ_i) to the $\mathcal{L}_3$ and responds with γ_i.

Phase 1: In this phase, the adversary can issue the private key queries on ID_i as needed. $\mathcal{B}$ firstly gets the corresponding c_i and t_i from $\mathcal{L}$. (If they do not exist, it runs the H query to get the corresponding c_i and t_i.) If $c_i = 0$, $\mathcal{B}$ aborts. If $c_i = 1$, $\mathcal{B}$ computes $d_{ID_i} = sH_1(ID_i) = at_iP = t_iP_{pub}$.

Challenge: Once $\mathcal{A}$ decides that the phase 1 is over, it outputs a message M^* and two distinct identity sets $S_0 = (ID_{0,1}, ID_{0,2}, \cdots, ID_{0,n})$, $S_1 = (ID_{1,1}, ID_{1,2}, \cdots, ID_{1,n})$. We require that $\mathcal{A}$ has not queried the private key for any $ID_i \in S_0 \triangle S_1$ in the phase 1. $\mathcal{B}$ picks a random bit $\mu \in \{0, 1\}$ and performs as follows.

1. Randomly choose a dummy identity $ID_0 \notin S_0 \cup S_1$, $B_0^* \in \mathbb{G}$, $r_1^*, r_2^* \in \mathbb{Z}_p$ and $K^* \in \mathbb{G}$, and compute $C_0^* = K^* + M^*$, $C_1^* = r_1^*P,\ C_2 = r_2^*cP,\ C_3 = cP$.

2. For $i = [0, n]$, get the value of $H(ID_i)$ from $\mathcal{L}$ (If ID_i does not exist in $\mathcal{L}$, run the H query) and compute

$$x_i^* = H_1\left(e\big(H(ID_i), P_{pub}\big)^{r_1^*}, ID_i\right),$$

   and construct polynomial functions

$$f_i(x) = \prod_{j=0, j\neq i}^{n} \frac{x - x_j^*}{x_i^* - x_j^*} = \sum_{j=0}^{n} a_{i,j}x^j.$$

3. For each $ID_i \in S_\mu \backslash S_{1-\mu}$, randomly choose $A_i,\ B_i^* \in \mathbb{G}$. For each $ID_i \in S_0 \cap S_1$, $\mathcal{B}$ first gets $c_i,\ t_i$ from the $\mathcal{L}$. If $c_i = 0$, it randomly chooses $A_i,\ B_i^* \in \mathbb{G}$. If $c_i = 1$, it computes $X_i = e(aP, cP)^{r_2^* t_i}$ and checks whether the tuple (X_i, ID_i) is in the $\mathcal{L}_2$. If yes, it obtains the corresponding λ_i and sets $A_i^* = \lambda_i$. Otherwise, it randomly chooses $A_i^* \in \mathbb{G}$ and adds the new tuple (X_i, ID_i, A_i^*) to the $\mathcal{L}_2$. Then it computes $Y_i = e(aP, cP)^{t_i}$ and checks whether the tuple (Y_i, ID_i) in the $\mathcal{L}_3$. If yes, it obtains the corresponding γ_i and sets $w_i^* = \gamma_i$. Otherwise, it randomly chooses $w_i^* \in \mathbb{G}$ and adds the new tuple (Y_i, ID_i, w_i^*) to the $\mathcal{L}_3$. Then it computes $B_i^* = K^* + w_i^*$.

4. For $i = [0, n]$, compute

$$Q_i^* = \sum_{j=0}^{n} a_{j,i}A_j^*, \quad U_i^* = \sum_{j=0}^{n} a_{j,i}B_j^*.$$

   and set the challenge ciphertext as $CT^* = \left(C_0^*, C_1^*, C_2, C_3, r_1^*, [Q_i^*, U_i^*]_{i=0}^n\right)$.

Phase 2: $\mathcal{A}$ continues to issue private key queries as needed with the restriction established in the challenge phase. $\mathcal{B}$ responds the same as in phase 1.

Guess: Finally, $\mathcal{A}$ outputs its guess $\mu' \in \{0,1\}$ of μ.

The above completes the description of the simulation. From the setting, we note that the correctness and randomness hold. $\mathcal{B}$ simulates a real attack environment for the adversary $\mathcal{A}$. At this point, $\mathcal{B}$ ignores the guess of $\mathcal{A}$ and picks a random tuple (X_j, ID_j, λ_j) from the list $\mathcal{L}_2$ or (Y_j, ID_j, γ_j) from the list $\mathcal{L}_3$. If its choice is $\mathcal{L}_2$, it outputs $X_j^{(r_2^* t_j)^{-1}}$ as the solution to the given instance of BDH. If its choice is $\mathcal{L}_3$, it outputs $Y_j^{t_j^{-1}}$ as the solution to the given instance of BDH. Similar to the analysis in Theorem 4.1, we have $\epsilon' \geq \frac{\epsilon}{e \cdot n \cdot q_E \cdot (q_{H_2} + q_{H_3})}$.

This completes the proof. □

Theorem 4.3. *Let hash functions H, H_2 be random oracles. If the BDH assumption holds, the proposed scheme is IND-rID-CPA secure. Specifically, if there is an IND-rID-CPA adversary $\mathcal{A}$ with advantage ϵ against our scheme, there is an algorithm $\mathcal{B}$ that solves the BDH problem with advantage*

$$\epsilon' \geq \frac{\epsilon}{e \cdot n \cdot q_E \cdot q_{H_2}},$$

where n is the number of the set S of identities, q_E and q_{H_2} are the number of queries to private key and H_2 respectively. e is the natural logarithm.

Proof. Suppose there exists an IND-rID-CPA adversary $\mathcal{A}$ that breaks our scheme with non-negligible advantage ϵ. We build a simulator (algorithm) $\mathcal{B}$ that can solve the BDH problem with advantage ϵ' by running $\mathcal{A}$. Let (P, aP, bP, cP) be a random instance of BDH problem taken as input by $\mathcal{B}$ and its goal is to compute $e(P,P)^{abc}$. In order to use $\mathcal{A}$ to solve the problem, $\mathcal{B}$ needs to simulate a challenger and respond all queries from $\mathcal{A}$. $\mathcal{B}$ works by interacting with $\mathcal{A}$ in an IND-rID-CPA game (Game 3) as follows.

Setup: $\mathcal{B}$ sets $P_{pub} = aP$ and generates the master public key as $mpk = (\mathbb{BG}, P, P_{pub}, H_1, H_3)$. Here, we view H, H_2 as random oracles controlled by the simulator.

***H*-queries:** For a query on ID_i, $\mathcal{B}$ responds as follows. $\mathcal{B}$ maintains a list $\mathcal{L}$ of tuples (ID_i, c_i, t_i, h_i). This list is initially empty. $\mathcal{B}$ first checks the $\mathcal{L}$. If the query ID_i has already appeared in $\mathcal{L}$ in a tuple (ID_i, c_i, t_i, h_i), it returns the corresponding h_i as the value of $H(ID_i)$. Otherwise, $\mathcal{B}$ randomly picks $t_i \in \mathbb{Z}_p$ and selects a random $c_i \in \{0,1\}$ with $Pr[c_i = 0] = \delta$ for some δ (determine later). If $c_i = 0$, it computes $h_i = t_i bP$, otherwise, it computes $h_i = t_i P$. Then it adds the tuple (ID_i, c_i, t_i, h_i) to the $\mathcal{L}$ and responds with h_i.

***H_2*-queries:** For a query on (X_i, ID_i), $\mathcal{B}$ responds as follows. $\mathcal{B}$ maintains a list $\mathcal{L}_2$ of tuples (X_i, ID_i, λ_i). This list is initially empty. $\mathcal{B}$ first checks the $\mathcal{L}_2$. If the

query (X_i, ID_i) has already appeared in the $\mathcal{L}_2$ in a tuple (X_i, ID_i, λ_i), it returns the corresponding λ_i as the value of $H_2(X_i, ID_i)$. Otherwise, $\mathcal{B}$ randomly picks a $\lambda_i \in \mathbb{G}$ as the value of $H_2(X_i, ID_i)$, then it adds (X_i, ID_i, λ_i) to the $\mathcal{L}_2$ and responds with λ_i.

Phase 1: In this phase, the adversary can issue the private key queries on ID_i as needed. $\mathcal{B}$ firstly gets the corresponding c_i and t_i from $\mathcal{L}$. (If they do not exist, it runs the H query to get the corresponding c_i and t_i.) If $c_i = 0$, $\mathcal{B}$ aborts. If $c_i = 1$, $\mathcal{B}$ computes $d_{ID_i} = sH_1(ID_i) = at_iP = t_iP_{pub}$.

Challenge: Once $\mathcal{A}$ decides that the phase 1 is over, it outputs two distinct messages $M_0,\ M_1$ from the same message space, an identity set $S^* = (ID_1, ID_2, \cdots, ID_n)$ and a non-empty revoked identity set $R^* = (ID_{l_1}, ID_{l_2}, \cdots, ID_{l_t})$. We require that $\mathcal{A}$ has not issued the private key queries for any $ID_i \in S^* \backslash R^*$ in the phase 1. $\mathcal{B}$ picks a random bit $\mu \in \{0, 1\}$ and performs as follows.

1. Choose a random dummy identity $ID_0 \notin S^* \cup R^*$, randomly pick $A_0^*,\ K_1^*,\ K_2^* \in \mathbb{G}$, $r_1^*,\ r_3^* \in \mathbb{Z}_p$ and compute $C_0'^* = K_1^* + K_2^* + M_\mu$, $C_1^* = r_1^*P,\ C_2^* = cP,\ C_3^* = r_3^*P$.

2. For $ID_i \in S^* \cap R^*$, it obtains the tuple (c_i, t_i, h_i) from $\mathcal{L}$ (If ID_i does not exist in $\mathcal{L}$, run the H query). If $c_i = 0$, it aborts, otherwise, it computes $X_i = e(aP, cP)^{t_i}$ and checks whether the tuple (X_i, ID_i) has appeared in $\mathcal{L}_2$. Return λ_i if it exists in $\mathcal{L}_2$, otherwise $\mathcal{B}$ picks a random $\lambda_i \in \mathbb{G}$. Then it sets $A_i^* = \lambda_i$ and adds the new tuple (X_i, ID_i, λ_i) to the $\mathcal{L}_2$. For each i where $ID_i \in S^* \backslash R^*$, it picks a random $A_i^* \in \mathbb{G}$.

3. For $i = [0, n]$, compute

$$x_i^* = H_1\left(e\big(H(ID_i), P_{pub}\big)^{r_1^*}, ID_i\right),$$

$$f_i(x) = \prod_{j=0, j\neq i}^{n} \frac{x - x_j^*}{x_i^* - x_j^*} = \sum_{j=0}^{n} a_{i,j}x^j,$$

$$B_i^* = K_1^* + H_3\left(e\big(H(ID_i), P_{pub}\big)^{r_3^*}, ID_i\right).$$

   Then it computes

$$Q_i^* = \sum_{j=0}^{n} a_{j,i}A_j^*, \quad U_i^* = \sum_{j=0}^{n} a_{j,i}B_j^*.$$

4. For each $ID_i \in R^*$, compute

$$x_i^* = H_1\left(e\big(H(ID_i), P_{pub}\big)^{r_1^*}, ID_i\right),$$

and construct

$$g(x) = \prod_{i=1}^{t}(x - x_i^*) = \sum_{i=0}^{t} b_i x^i \mod p.$$

5. For $i = 0, 1, 2, \cdots, t$, compute

$$Q_i'^* = Q_i^* + b_i K_2^*,$$

and set the challenge ciphertext as $CT'^* = \big(C_0'^*, C_1^*, C_2^*, C_3^*, b_0, b_1, \cdots, b_{t-1}, Q_0'^*, Q_1'^*, \cdots, Q_t'^*, Q_{t+1}^*, \cdots, Q_n^*, U_0^*, U_1^*, \cdots, U_n^*\big)$.

Phase 2: $\mathcal{A}$ continues to issue private key queries as needed with the restriction established in the challenge phase. $\mathcal{B}$ responds the same as in phase 1.

Guess: Finally, $\mathcal{A}$ outputs its guess $\mu' \in \{0, 1\}$ of μ.

The above completes the description of the simulation. From the setting, we note that the correctness and randomness hold. $\mathcal{B}$ simulates a real attack environment for the adversary $\mathcal{A}$. $\mathcal{B}$ ignores the guess of $\mathcal{A}$ and picks a random tuple (X_j, ID_j, λ_j) from the list $\mathcal{L}_2$. It then obtains the corresponding t_j from $\mathcal{L}$ and outputs $X_j^{t_j^{-1}}$ as the solution to the given instance of BDH. $\mathcal{A}$ can obtain the private keys for $ID_i \in S^* \cap R^*$. But this will not help the adversary to distinguish the message in our security reduction. According to the breaking assumption that the adversary will break the scheme with non-negligible advantage, it must query H_2 with the input containing the solution of the hard problem if we only consider the case that $\mathcal{A}$ chooses the ID_i where $H(ID_i) = t_i bP$ to break our scheme. As the analysis in Theorem 4.1, we have $\epsilon' \geq \frac{\epsilon}{e \cdot n \cdot q_E \cdot q_{H_2}}$.

This completes the proof. □

Theorem 4.4. *Let hash functions H, H_1 be random oracles. If the BDH assumption holds, the proposed scheme is selective ANON-rID-CPA secure. Specifically, if there is a selective ANON-rID-CPA adversary $\mathcal{A}$ with advantage ϵ against our scheme, there is an algorithm $\mathcal{B}$ that solves the BDH problem with advantage*

$$\epsilon' \geq \frac{\epsilon}{t \cdot q_{H_1}},$$

where t is the number of revoked identities and q_{H_1} is the number of H_1 queries.

Proof. Suppose there exists an AIND-rID-CPA adversary $\mathcal{A}$ that breaks our scheme with non-negligible advantage ϵ. We build a simulator (algorithm) $\mathcal{B}$ that can solve the BDH problem with advantage ϵ' by running $\mathcal{A}$. Let (P, aP, bP, cP) be a random instance of BDH problem taken as input by $\mathcal{B}$ and its goal is to compute $e(P, P)^{abc}$.

In order to use $\mathcal{A}$ to solve the problem, $\mathcal{B}$ needs to simulate a challenger and respond all queries from $\mathcal{A}$. $\mathcal{B}$ works by interacting with $\mathcal{A}$ in an ANON-rID-CPA game (Game 4) as follows.

Init: The adversary outputs two distinct target revoked identity sets $R_0 = (ID_{0,1}, ID_{0,2}, ..., ID_{0,t})$ and $R_1 = (ID_{1,1}, ID_{1,2}, ..., ID_{1,t})$ that he wants to attack.

Setup: $\mathcal{B}$ sets $P_{pub} = aP$ and generates the master public key as $mpk = (\mathbb{BG}, P, P_{pub}, H_2, H_3)$. Here, we view H, H_1 as random oracles controlled by the simulator. Then $\mathcal{B}$ randomly picks a bit $\mu \in \{0, 1\}$ and an identity $ID^* \in R_\mu \backslash R_{1-\mu}$.

H-queries: For a query on ID_i, $\mathcal{B}$ responds as follows. $\mathcal{B}$ maintains a list $\mathcal{L}$ of a tuple (ID_i, k_i, h_i). This list is initially empty. $\mathcal{B}$ first checks $\mathcal{L}$. If the query ID_i has already appeared in the $\mathcal{L}$ in a tuple (ID_i, k_i, h_i), it returns the corresponding h_i as the value of $H(ID_i)$, otherwise, it chooses a random $k_i \in \mathbb{Z}_p$. If $ID_i = ID^*$, it sets $h_i = k_i bP$ and sets $h_i = k_i P$ if $ID_i \neq ID^*$.

H_1-queries: For a query on (T_i, ID_i), $\mathcal{B}$ responds as follows. $\mathcal{B}$ maintains a list $\mathcal{L}_1$ of a tuple (T_i, ID_i, η_i). This list is initially empty. $\mathcal{B}$ first checks the $\mathcal{L}_1$. If the query (T_i, ID_i) has already appeared in the $\mathcal{L}_1$ in a tuple (T_i, ID_i, η_i), it returns the corresponding η_i as the value of $H_1(T_i, ID_i)$. Otherwise, $\mathcal{B}$ randomly picks a $\eta_i \in \mathbb{G}$ as the value of $H_1(T_i, ID_i)$, adds the tuple (T_i, ID_i, η_i) to $\mathcal{L}_1$ and responds with η_i.

Phase 1: In this phase, $\mathcal{A}$ can issue the private key queries for $ID_i \notin R_0 \triangle R_1$ as needed. $\mathcal{B}$ gets the corresponding k_i from $\mathcal{L}$ (If they do not exist, it runs the H query and gets the corresponding k_i.) and computes $d_{ID_i} = sH_1(ID_i) = ak_iP = k_iP_{pub}$.

Challenge: Once $\mathcal{A}$ decides that the phase 1 is over, it outputs a message M^* and an identity set $S^* = (ID_1, ID_2, \cdots, ID_n)$. $\mathcal{B}$ performs as follows.

1. Choose a dummy identity $ID_0 \notin S^* \cup R_0 \cup R_1$, randomly pick $K_1^*,\ K_2^* \in \mathbb{G}$, $r_2^*,\ r_3^* \in \mathbb{Z}_p$ and compute

$$C_0'^* = K_1^* + K_2^* + M_b,\ C_1^* = c^*P,\ C_2^* = r_2^*P,\ C_3^* = r_3^*P.$$

2. For each $ID_i \in S^*$ and ID_0, if $ID_i = ID^*$, randomly choose $x^* \in \mathbb{Z}_p$ and set $x_i^* = x^*$. Otherwise, obtain the tuple (k_i, h_i) from $\mathcal{L}$, compute $T_i = e(aP, cP)^{k_i}$ and check whether the tuple (T_i, ID_i) appeared in the $\mathcal{L}_1$. Return η_i if it exists in $\mathcal{L}_1$, otherwise pick a random $\eta_i \in \mathbb{Z}_p$. Then $\mathcal{B}$ sets $x_i^* = \eta_i$ and adds the new tuple (T_i, ID_i, η_i) to the $\mathcal{L}_1$.

3. For $i = [0, n]$, compute

$$f_i(x) = \prod_{j=0, j\neq i}^{n} \frac{x - x_j^*}{x_i^* - x_j^*} = \sum_{j=0}^{n} a_{i,j} x^j,$$

$$A_i^* = H_2\left(e\big(H(ID_i), P_{pub}\big)^{r_2^*}, ID_i\right),$$

$$B_i^* = K_1^* + H_3\left(e\big(H(ID_i), P_{pub}\big)^{r_3^*}, ID_i\right).$$

Then it computes

$$Q_i^* = \sum_{j=0}^{n} a_{j,i} A_j^*, \quad U_i^* = \sum_{j=0}^{n} a_{j,i} B_j^*.$$

4. For each $ID_i \in R_0 \cap R_1$, obtain the tuple (k_i, h_i) from $\mathcal{L}$, compute $T_i = e(aP, cP)^{k_i}$ and check whether the tuple (T_i, ID_i) has appeared in the $\mathcal{L}_1$. Return η_i if it exists in the $\mathcal{L}_1$, otherwise pick a random $\eta_i \in \mathbb{Z}_p$. Then it sets $x_i^* = \eta_i$ and adds the new tuple (T_i, ID_i, η_i) to the $\mathcal{L}_1$. For $ID_i \in R_\mu \backslash R_{1-\mu}$, randomly choose $x_i^* \in \mathbb{Z}_p$. Then it computes

$$g(x) = \prod_{i=1}^{t} (x - x_i^*) = \sum_{i=0}^{t} b_i x^i \mod p.$$

5. For $i = 0, 1, 2, \cdots, t$, compute

$$Q_i'^* = Q_i^* + b_i K_2^*,$$

and set the challenge ciphertext as $CT'^* = (C_0'^*, C_1^*, C_2^*, C_3^*, b_0, b_1, \cdots, b_{t-1}, Q_0'^*, Q_1'^*, \cdots, Q_t'^*, Q_{t+1}^*, \cdots, Q_n^*, U_0^*, U_1^*, \cdots, U_n^*)$.

Phase 2: $\mathcal{A}$ continues to issue private key queries on $ID_i \notin R_0 \triangle R_1$. $\mathcal{B}$ responds the same as in phase 1.

Guess: Finally, $\mathcal{A}$ outputs its guess $\mu' \in \{0, 1\}$ of μ.

The above completes the description of the simulation. From the setting, we note that the correctness and randomness hold. $\mathcal{B}$ simulates a real attack environment for the adversary $\mathcal{A}$. There is no abortion in our simulation. If the adversary chooses ID^* to distinguish the revoked identity sets, $\mathcal{B}$ can successfully solve the BDH problem by computing $T^{*\frac{1}{k^*}}$. It is not hard to compute that the probability of $\mathcal{A}$ chooses ID^* to break our scheme is $\frac{1}{t-k} \geq \frac{1}{t}$ where $k = |R_0 \cap R_1|$, we have $\epsilon' \geq \frac{\epsilon}{t \cdot q_{H_1}}$ and yield Theorem 4.4.

This completes the proof. □

Note that the reduction in Theorem 4.4 is slightly different from the defined security model (game 4). In the security model, the random bit μ is chosen in the challenge phase. But in our simulation, μ is chosen in the setup phase. We claim that it does not have security issues. As μ is chosen by $\mathcal{B}$ secretly, from the point

view of the adversary, it cannot distinguish whether μ is chosen in the setup phase or the challenge phase. Thus our proof of Theorem 4.4 is reasonable and correct.

4.5 Conclusion

In this chapter, we further studied the receiver privacy-preserving in the revocable identity-based broadcast encryption system. We proposed a framework of revocable identity-based broadcast encryption, which can fully preserve the user privacy including the receivers stated in the original ciphertext and the revoked users. The proposed scheme is the first work that can achieve fully user anonymity in the revocable identity-based broadcast encryption. Finally, we gave the concrete security analysis of the proposed scheme under the hardness of BDH problem in the random oracle model.

Chapter 5

Fully Privacy-Preserving IBBE with Authorization

This chapter describes a fully privacy-preserving identity-based broadcast encryption with authorization scheme. The proposed scheme considers a negative analogue of revocable identity-based broadcast encryption. The original scheme was published in *The Computer Journal* [LMGC17].

5.1 Introduction

Broadcast encryption (BE) introduced by Fiat and Naor [FN94] provides an efficient approach for sharing a message with a group of users. It allows an encryptor to encrypt a message under a broadcast receiver set S with whom the encryptor wants to share the message via a public broadcast channel. Only those users whose identities belong to S are able to obtain the message. The basic security requirement of broadcast encryption is collusion resistance. A broadcast encryption scheme is said to be full collusion resistant if the outsiders (the users who are not in S) are unable to learn any information about the broadcast message even if they collude.

For flexible applications, a new cryptographic primitive called recipient revocable identity-based broadcast encryption (RR-IBBE) was introduced in [SCG+16]. It attempts to securely share an externally stored data with a group of users while allowing the encryptor to delegate revocation computations on the outsourced encrypted data to revoke some users' data retrieval rights. In the RR-IBBE, a third party cannot decrypt the ciphertext, but it can remove some of users from the receiver set stated in the original ciphertext generated via a broadcast encryption. Unfortunately, the work in [SCG+16] does not consider the receiver privacy. It is required to send the receiver identity set to the third party who performs the revocation operation, which exposes the receiver privacy. Aiming to protect the receiver privacy, Lai et al. [LMG+16] subsequently put forward the first anonymous RR-IBBE scheme. The revocation process does not require any information of the receiver identity. However, the work in [LMG+16] cannot protect the revoked users' identity information. The identities of revoked users must be attached in the final

ciphertext, which somehow exposes the user privacy, especially the user who has been revoked. Later, a fully privacy-preserving RR-IBBE scheme has been achieved in [LMG+17], where both the identities of the receivers and the revoked users can be protected. The size of ciphertext is linear in the size of the original broadcast identity set.

In this chapter, we focus on the authorization in identity-based broadcast encryption. Authorization is a negative analogue of revocation. If we consider the revocation as "black list", then the authorization can be viewed as "white list". In an authorization system, the final ciphertext is generated by an authorized user set in a way that users who are not in the authorized set are unable to decrypt the ciphertext even they are in the broadcast identity set. In contrast, a revocation system uses the revoked user set to generate the final ciphertext such that the revoked user cannot retrieve the message anymore. We observe that if the revoked sets are large, such as larger than half of the universal user set, the authorization system might be more efficient than the revocation system. Authorization is also suitable for the scenarios where the receivers are decided by several entities. Unfortunately, this has never been captured in the literature.

At the first glance, one might think the authorization in broadcast encryption can be achieved by using a re-encryption mechanism. Although the authorization looks like re-encryption in some extent, it is entirely different from re-encryption and requires a higher security level in comparison with re-encryption. In particular, since it is in a multi-user setting, a traditional re-encryption approach cannot resist collusion attacks, which are explained as follows. Suppose a user with ID_1 is within the broadcast identity set but not belongs to the authorized set, and a user with ID_2 is not in the broadcast identity set but in the authorized set. These two users might try to decrypt the ciphertext and let the user with ID_2 get the original broadcast ciphertext and the user with ID_1 decrypts it using its secret key. For the re-encryption in the broadcast encryption, Chu et al. [CWC+09] presented a conditional proxy broadcast re-encryption scheme which allows a proxy to transform a ciphertext intended for a receiver set to another ciphertext intended for another receiver set. Xu et al. [XJW+16] presented a conditional identity-based broadcast proxy re-encryption scheme with constant size ciphertext based on [Del07]. Nevertheless, the receivers identities are required to send to the proxy in both systems.

In this chapter, we propose a new notion of fully privacy-preserving identity-based broadcast encryption with authorization (FPP-IBBEA), which allows a third party to authorize the decryption rights for a set of users. Only the user whose identity belongs to both the identity set sated in the original broadcast ciphertext and the authorized identity set is able to retrieve the encrypted message. This notion is especially suitable for the applications where the user privacy should be considered

and the users ciphertext decryption rights are able to be reallocated. We present an FPP-IBBEA construction where the final ciphertext size is linear in the size of the authorized identity set, instead of the size of the original broadcast identity set. The security of the proposed scheme is derived from the hardness of BDH problem in the random oracle model. The proposed scheme supports multiple authorization and all properties are maintained. Namely, the authorization operation can be performed dependently by multiple third parties under each selected authorized identity set. Only the users belonging to all authorized identity sets are able to obtain the message successfully.

Organization. The rest of this chapter is organized as follows. In the next Section, we give the definitions of fully privacy-preserving identity-based broadcast encryption with authorization and the corresponding security models to capture its security. The proposed scheme is presented in Section 5.3. We also give a discussion in this section. In Section 5.4, we provide the security analysis of the proposed scheme under the defined security models and conclude this chapter in Section 5.5.

5.2 Definitions and Security Models

This section will define the syntax and the security of the fully privacy-preserving identity-based broadcast encryption with authorization (FPP-IBBEA). We can view the authorization is a negative analogue of revocation where only the authorized users can obtain the encrypted message. An FPP-IBBEA scheme is made up of five algorithms below.

- **Setup**(1^λ). Taking as input a security parameter λ, the setup algorithm returns a master key pair (mpk, msk), where the master public key mpk is publicly known while the master secret key msk is kept secretly.
- **KeyGen**(mpk, msk, ID). Taking as input the master public/secret key pair (mpk, msk) and a user identity ID, the key generation returns a user private key d_{ID}.
- **Encrypt**(mpk, M, S). Taking as input the master public key mpk, a message M and an identity set $S = (ID_1, ID_2, \cdots, ID_n)$, the encryption algorithm returns a ciphertext CT.
- **Authorize**(mpk, L, CT). Taking as input the master public key mpk, a set of authorized identities $L = (ID_{l_1}, ID_{l_2}, \cdots, ID_{l_k})$, and a ciphertext

CT, the authorization algorithm returns a new ciphertext CT' after authorization.

– **Decrypt** (mpk, CT', ID, d_{ID}). Taking as input the master public key mpk, a ciphertext CT', an identity ID and the corresponding private key d_{ID}, the decryption algorithm returns the message M if $ID \in S \cap L$ or $\perp$ otherwise to denote failure.

Correctness. We require the following correctness requirement. Suppose that (mpk, msk) are the result of calling $\mathsf{Setup}(1^\lambda)$, the d_{ID} and CT are then the result of calling $\mathsf{KeyGen}(mpk, msk, ID)$ and $\mathsf{Encrypt}(mpk, M, S)$ respectively, CT' is the result of calling $\mathsf{Authorize}(mpk, L, CT)$. We then have

$$\mathsf{Decrypt}(mpk, CT', ID, d_{ID}) = \begin{cases} M & \text{If } ID \in S \cap L, \\ \perp & \text{otherwise.} \end{cases}$$

Without loss of generality, we always assume that L is a non-empty set in this chapter.

Security Notions. In this section, we formalize the indistinguishability security models for FPP-IBBEA. The final ciphertext CT' should preserve the message confidentiality and the receiver privacy against the adversary (the public). As the encrypted message CT firstly will be distributed to a third party. The initial ciphertext CT should also be secure against the third party. More specifically, the security requirements are as follows.

1. The message and the identity set in the ciphertext CT cannot be distinguished without a valid private key associated with an identity $ID \in S$.

2. The message and the authorized identity set in CT' cannot be distinguished without a valid private key associated with an identity $ID \in S \cap L$.

Similar to the security of anonymous identity-based broadcast encryption scheme where the encryption of unpredictable message must be indistinguishable from a random string of the same length and the receiver identity set must be indistinguishable from a random identity set with the same size. Next, we define four indistinguishability security models to capture the security requirements of the proposed scheme, namely the IND-ID-CPA security, ANON-ID-CPA security, IND-aID-CPA security and ANON-aID-CPA security. These four security models are defined with the following four games played between a challenger and an adversary.

IND-ID-CPA Security. This security model claims that the message in the CT is indistinguishable from a random string of the same length and it works as follows.

- **Setup:** The challenger runs the setup algorithm to generate the master public key mpk. It then sends mpk to the adversary.
- **Phase 1:** In this phase, the adversary can issue the private key queries as needed. Upon receiving a private key query for ID_i, the challenger runs the key generation algorithm to generate the private key d_{ID_i} and responds by returning d_{ID_i}.
- **Challenge:** Once the adversary decides that the phase 1 is over, it outputs two distinct messages M_0, M_1 from the same message space and a challenge identity set $S^* = (ID_1, ID_2, \cdots, ID_n)$. We require that the adversary has not queried the private key on $ID_i \in S^*$ in the phase 1. The challenger picks a random bit $\mu \in \{0, 1\}$ and generates the challenge ciphertext CT^* on message M_μ under S^*, then it returns CT^* to the adversary.
- **Phase 2:** The adversary continues to issue more private key queries with the restriction established in the challenge phase. The simulator responds the same as in phase 1.
- **Guess:** Finally, the adversary outputs its guess $\mu' \in \{0, 1\}$ of μ and wins the game if $\mu = \mu'$.

We refer to such an adversary as an IND-ID-CPA adversary and define the advantage of the adversary in winning the game as

$$\mathsf{Adv}^{\mathsf{IND\text{-}ID\text{-}CPA}}_{\mathsf{FPP-IBBEA}}(\lambda) = \left| \Pr[\mu = \mu'] - \frac{1}{2} \right|.$$

The probability is over the random coins of the adversary, the challenger and all probabilistic algorithms run by the challenger.

Definition 5.1. *We say that a scheme is IND-ID-CPA secure if for any probabilistic polynomial time IND-ID-CPA adversary, the advantage* $\mathsf{Adv}^{\mathsf{IND\text{-}ID\text{-}CPA}}_{\mathsf{FPP-IBBEA}}(\lambda)$ *is negligible.*

ANON-ID-CPA Security. This security model claims that the identity set in the CT is indistinguishable from a random identity set of the same length and it works as follows.

- **Setup:** The challenger runs the setup algorithm to generate the master public key mpk and sends mpk to the adversary.

- **Phase 1:** In the phase, the adversary can issue private key queries as needed. Upon receiving a private key query for ID_i, The challenger runs the key generation algorithm to generate the private key d_{ID_i} and responds by returning d_{ID_i}.

- **Challenge:** Once the adversary decides that the phase 1 is over, it outputs a message M^* and two distinct identity sets $S_0 = (ID_{0,1}, ID_{0,2}, ..., ID_{0,n})$, $S_1 = (ID_{1,1}, ID_{1,2}, ..., ID_{1,n})$. We require that the adversary has not issued the private key queries on any ID_i in the phase 1, where $ID_i \in S_0 \triangle S_1 = (S_0 \backslash S_1) \cup (S_1 \backslash S_0)$. The simulator picks a random bit $\mu \in \{0, 1\}$ and generates the challenge ciphertext CT^* on message M^* under S_μ.

- **Phase 2:** The adversary continues to issue more private key queries with the restriction established in the challenge phase. The challenger responds the same as in phase 1.

- **Guess:** Finally, the adversary outputs its guess $\mu' \in \{0, 1\}$ of μ and wins the game if $\mu = \mu'$.

We refer to such an adversary as an ANON-ID-CPA adversary and define the advantage of the adversary in winning the game as

$$\mathsf{Adv}^{\mathsf{ANON\text{-}ID\text{-}CPA}}_{\mathsf{FPP-IBBEA}}(\lambda) = \left| \Pr[\mu = \mu'] - \frac{1}{2} \right|.$$

The probability is over the random coins of the adversary, the challenger and all probabilistic algorithms run by the challenger.

Definition 5.2. *We say that a scheme is ANON-ID-CPA secure if for any probabilistic polynomial time ANON-ID-CPA adversary, the advantage* $\mathsf{Adv}^{\mathsf{ANON\text{-}ID\text{-}CPA}}_{\mathsf{FPP-IBBEA}}(\lambda)$ *is negligible.*

IND-aID-CPA Security. This security model claims that the message in the ciphertext CT' after authorization is indistinguishable from a random string of the same length and it works as follows.

- **Setup:** The challenger runs the setup algorithm to generate the master public key mpk and sends mpk to the adversary.

- **Phase 1:** In this phase, the adversary can issue private key queries as needed. Upon receiving a private key query for ID_i. The challenger runs the key generation algorithm to generate the private key d_{ID_i} and responds by returning d_{ID_i}.

- **Challenge:** Once the adversary decides that the phase 1 is over, it outputs two distinct messages M_0, M_1 from the same message space, a challenge identity set $S^* = (ID_1, ID_2, \cdots, ID_n)$ and an authorized identity set $L^* = \{ID_{l_1}, ID_{l_2}, \cdots, ID_{l_k}\}$. We require that the adversary has not queried the private key on ID_i in the phase 1, where $ID_i \in S^* \cap L^*$. The challenger picks a random bit $\mu \in \{0, 1\}$ and generates the challenged ciphertext CT'^* on message M_μ under S^* and L^*, then it returns CT'^* to the adversary.

- **Phase 2:** The adversary continues to issue more private key queries with the restriction established in the challenge phase. The challenger responds the same as phase 1.

- **Guess:** Finally, the adversary outputs its guess $\mu' \in \{0, 1\}$ of μ and wins the game if $\mu = \mu'$.

We refer to such an adversary as an IND-aID-CPA adversary and define the advantage of the adversary in winning the game as

$$\mathsf{Adv}^{\mathsf{IND\text{-}aID\text{-}CPA}}_{\mathsf{FPP-IBBEA}}(\lambda) = \left| \Pr[\mu = \mu'] - \frac{1}{2} \right|.$$

The probability is over the random coins of the adversary, the challenger and all probabilistic algorithms run by the challenger.

Definition 5.3. *We say that a scheme is IND-aID-CPA secure if for any probabilistic polynomial time IND-aID-CPA adversary, the advantage* $\mathsf{Adv}^{\mathsf{IND\text{-}aID\text{-}CPA}}_{\mathsf{FPP-IBBEA}}(\lambda)$ *is negligible.*

ANON-aID-CPA Security. This security model claims that the authorized identity set in CT' is indistinguishable from a random identity set of the same length without a valid private key, and it works as follows.

- **Setup:** The challenger runs the setup algorithm to generate the master public key mpk and sends mpk to the adversary.

- **Phase 1:** In this phase, the adversary can issue private key queries as needed. Upon receiving a private key query for ID_i. The challenger runs the key generation algorithm to generate the private key d_{ID_i} and sends d_{ID_i} to the adversary.

- **Challenge:** Once the adversary decides that the phase 1 is over, it outputs a message M^*, an identity set $S^* = (ID_1, ID_2, \cdots, ID_n)$ and two distinct authorized identity sets $L_0 = (ID_{0,1}, ID_{0,2}, ..., ID_{0,k})$, $L_1 = (ID_{1,1}, ID_{1,2}, ..., ID_{1,k})$. We require that the adversary has not issued the private key queries on any ID_i

in the phase 1, where $ID_i \in S^* \cap (L_0 \triangle L_1)$, here $L_0 \triangle L_1 = (L_0 \backslash L_1) \cup (L_1 \backslash L_0)$. The challenger randomly picks a bit $\mu \in \{0,1\}$ and generates the challenge ciphertext CT'^* on message M^* under S^* and L_μ.

– **Phase 2:** The adversary continues to issue more private key queries with the restriction established in the challenge phase.

– **Guess:** Finally, the adversary outputs its guess $\mu' \in \{0,1\}$ of μ and wins the game if $\mu = \mu'$.

We refer to such an adversary as an ANON-aID-CPA adversary and define the advantage of the adversary in winning the game as

$$\mathsf{Adv}^{\mathsf{ANON\text{-}aID\text{-}CPA}}_{\mathsf{FPP-IBBEA}}(\lambda) = \left| \Pr[\mu = \mu'] - \frac{1}{2} \right|.$$

The probability is over the random coins of the adversary, the challenger and all probabilistic algorithm run by the challenger.

Definition 5.4. *We say that a scheme is ANON-aID-CPA secure if for any probabilistic polynomial time ANON-aID-CPA adversary, the advantage* $\mathsf{Adv}^{\mathsf{ANON\text{-}aID\text{-}CPA}}_{\mathsf{FPP-IBBEA}}(\lambda)$ *is negligible.*

5.3 The Proposed Scheme

In this section, we give the concrete construction of our proposed scheme.

5.3.1 Construction

We now present our construction based on the symmetric bilinear group defined in the Chapter 2. We choose an identity $ID_0 \notin S$ as a dummy user. To do this, we can make a simplifying assumption. We assume that the identity in the system cannot be all zero or all one. No one allows to query such kind of identities. This assumption is reasonable and has been used in many papers, such as [BB11]. Another method about how to choose dummy identities can be found in [HLR10]. Under this assumption, hence we can just simply set the dummy identity ID_0 as the one composed of all zero or all one. The usage of dummy identity also enables us to prove the privacy of the identity set S even there are two identities in S, namely the indistinguishability security of S for CT.

Setup(1^λ). Given a security parameter λ, the setup algorithm randomly chooses a bilinear group $\mathbb{BG} = (\mathbb{G}, \mathbb{G}_T, e, p)$ with generator $P \in \mathbb{G}$. Then it chooses a random $s \in \mathbb{Z}_p$, three cryptographic hash functions $H : \{0,1\}^* \to \mathbb{Z}_p$, $H_1 : \{0,1\}^* \to \mathbb{G}$,

$H_2 : \mathbb{G}_T \times \{0,1\}^* \to \mathbb{G}$ and sets $P_{pub} = sP$. The master public key and the master secret key are

$$mpk = (\mathbb{BG}, P, P_{pub}, H, H_1, H_2)\,,\ msk = s.$$

KeyGen(mpk, msk, ID). Given the master key pair (mpk, msk) and a user identity $ID \in \{0,1\}^*$, the key generation algorithm outputs the user's private key d_{ID} by computing

$$d_{ID} = sH_1(ID).$$

Encrypt(mpk, M, S). Given the master public key mpk, a message $M \in \mathbb{G}$ and an identity set $S = \{ID_1, ID_2, \ldots, ID_n\}$, the encryption algorithm chooses a random $ID \notin S$ as the dummy user denoted by ID_0 and performs as follows.

1. Pick $K \in \mathbb{G}$ as an encryption key and $r \in \mathbb{Z}_p$, and compute

$$C = K + M,\quad C' = rP.$$

2. For each $i \in [0, n]$, compute

$$A_i = K + H_2\big(e(H_1(ID_i), P_{pub})^r, ID_i\big),$$

$$x_i = H(ID_i),$$

$$f_i(x) = \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j} = \sum_{j=0}^{n} a_{i,j}x^j \mod p.$$

After computing all $f_i(x)$, it uses these functions' coefficients to compute

$$C_i = \sum_{j=0}^{n} a_{j,i}A_j.$$

The output ciphertext is $CT = (C, C', \{C_i\}_{i=0}^n)$.

From the setting of $f_i(x)$, we have $f_i(x_i) = 1, f_i(x_j) = 0$ where $i \neq j$.

Authorize(mpk, CT, L). Given the master public key mpk, a ciphertext $CT = (C, C', \{C_i\}_{i=0}^n)$ and an authorized identity set $L = \{ID_{l_1}, ID_{l_2}, \cdots, ID_{l_k}\}$. For simplicity, we denote L as the identity index set. The authorization algorithm, for each $i \in L$, computes

$$x_i = H(ID_i),$$

$$A_i' = C_0 + x_iC_1 + x_i^2C_2 + \cdots + x_i^nC_n,$$

$$g_i(x) = \prod_{j\in L, j\neq i} \frac{x - x_j}{x_i - x_j} = \sum_{j=0}^{k-1} b_{i,j}x^j \mod p,$$

Then for $i \in [1, k]$, it computes

$$C'_i = \sum_{j=1}^{k} b_{j,i-1} A'_j.$$

From the setting of $g_i(x)$, we also have $g_i(x_i) = 1$, $g_i(x_j) = 0$ where $i \neq j$. The output ciphertext after authorization is $CT' = \left(C, C', \{C_i\}_{i=1}^{k}\right)$.

Decrypt$(mpk, CT', ID_i, d_{ID_i})$. Given the master public key mpk, a ciphertext $CT' = \left(C, C', \{C_i\}_{i=1}^{k}\right)$, an identity ID_i and the corresponding private key d_{ID_i}, the decryption algorithm decrypts the ciphertext by computing

$$x_i = H(ID_i),$$

$$W = C'_1 + x_i C'_2 + x_i^2 C'_3 + \cdots + x_i^{k-1} C'_k,$$

$$K' = W - H_2\big(e\left(d_{ID_i}, C'\right), ID_i\big).$$

After retrieving the encryption key K', it is easy to obtain the message by computing

$$M = C - K'.$$

5.3.2 Correctness

Below we show that our construction meets the correctness requirement defined previously. We observe that for a ciphertext CT' formed by calling $\mathsf{Authorize}(mpk, CT, L)$, where CT is formed by calling $\mathsf{Encrypt}(mpk, M, S)$ and a key formed by calling $\mathsf{KeyGen}(mpk, msk, ID)$, we have that if $ID_i \in S$ and $x_i = H(ID_i)$,

$$\begin{aligned}
A'_i &= C_0 + x_i C_1 + x_i^2 C_2 + \cdots + x_i^n C_n \\
&= (a_{0,0} A_0 + a_{1,0} A_1 + \cdots + a_{n,0} A_n) \\
&+ (a_{0,1} A_0 + a_{1,1} A_1 + \cdots + a_{n,1} A_n)\, x_i \\
&+ (a_{0,2} A_0 + a_{1,2} A_1 + \cdots + a_{n,2} A_n)\, x_i^2 + \cdots \\
&+ (a_{0,n} A_0 + a_{1,n} A_1 + \cdots + a_{n,n} A_n)\, x_i^n \\
&= (a_{0,0} + a_{0,1} x_i + a_{0,2} x_i^2 + \cdots + a_{0,n} x_i^n)\, A_0 \\
&+ (a_{1,0} + a_{1,1} x_i + a_{1,2} x_i^2 + \cdots + a_{1,n} x_i^n)\, A_1 + \cdots \\
&+ (a_{n,0} + a_{n,1} x_i + a_{n,2} x_i^2 + \cdots + a_{n,n} x_i^n)\, A_n \\
&= f_0(x_i) A_0 + f_1(x_i) A_1 + \cdots + f_n(x_i) A_n \\
&= A_i,
\end{aligned}$$

Table 5.1: Comparison of Performances.

Scheme	Mpk	Size of CT	Size of CT'	CT Anon.	R./A. Anon.	Multiple	Security Model
[SCG+16]	$O(N)$	$O(m)$	$O(1)$	×	×	×	Selective
[LMG+16]	$O(1)$	$O(n)$	$O(n)$	✓	×	×	Adaptive
[AD17]	$O(N)$	$O(m)$	$O(1)$	×	×	×	Adaptive
Ours	$O(1)$	$O(n)$	$O(k)$	✓	✓	✓	Adaptive

Anon. is short for anonymity. R. and A. denote revocation and authorization respectively.

If $ID_i \in L$ and $x_i = H(ID_i)$, we have

$$\begin{aligned} W &= C'_1 + x_i C'_2 + x_i^2 C'_3 + \cdots + x_i^{k-1} C'_k \\ &= (b_{1,0}A'_1 + b_{2,0}A'_2 + \cdots + b_{k,0}A'_k) \\ &+ (b_{1,1}A'_1 + b_{2,1}A'_2 + \cdots + b_{k,1}A'_k)\, x_i \\ &+ (b_{1,2}A'_1 + b_{2,2}A'_2 + \cdots + b_{k,2}A'_k)\, x_i^2 + \cdots \\ &+ (b_{1,k-1}A'_1 + b_{2,k-1}A'_2 + \cdots b_{k,k-1}A'_k)\, x_i^{k-1} \\ &= \left(b_{1,0} + b_{1,1}x_i + b_{1,2}x_i^2 + \cdots + b_{1,k-1}x_i^{k-1}\right) A'_1 \\ &+ \left(b_{2,0} + b_{2,1}x_i + b_{2,2}x_i^2 + \cdots + b_{2,k-1}x_i^{k-1}\right) A'_2 \\ &+ \cdots \\ &+ \left(b_{k,0} + b_{k,1}x_i + b_{k,2}x_i^2 + \cdots + b_{k,k-1}x_i^{k-1}\right) A'_k \\ &= g_1(x_i)A'_1 + g_2(x_i)A'_2 + \cdots + g_n(x_i)A'_k \\ &= A'_i. \end{aligned}$$

Therefore, if $ID_i \in S \cap L$, $W = A_i$, otherwise, A'_i is a random number in $\mathbb{G}$. After obtaining A_i, it uses its private key d_{ID_i} and computes

$$\begin{aligned} K' &= W - H_2\left(e\left(d_{ID_i}, C'\right), ID_i\right) \\ &= A_i - H_2\left(e\left(d_{ID_i}, C'\right), ID_i\right) \\ &= K + H_2\left(e(H_1(ID_i), P_{pub})^r, ID_i\right) - H_2\left(e\left(sH_1(ID_i), rP\right), ID_i\right) \\ &= K + H_2\left(e(H_1(ID_i), P_{pub})^r, ID_i\right) - H_2\left(e\left(H_1(ID_i), sP\right)^r, ID_i\right) \\ &= K. \end{aligned}$$

$$M = C_0 - K' = K + M - K = M.$$

That is, if $ID_i \in S \cap L$, using a valid private key to the decryption on the ciphertext produces the original message M.

5.3.3 Comparison and Discussion

In this section, we compare our scheme with the other three works [SCG+16, LMG+16, AD17]. In our comparison, we use CT to denote the ciphertext generated by the Encrypt algorithm. For the ciphertext generated by the Authorize algorithm in our scheme and the ciphertext generated by the Revoke algorithm in [SCG+16, LMG+16,

AD17], we use the same notation CT' to denote them. n is the number of identity set S for one encryption in the encryption phase. k is the number of the authorized identity set L in the authorization phase. N is the maximum size of the set of receivers for one encryption, that is, $n \leq N$. m is the maximum revocation number, where $m \leq n$.

From the Table 5.1, we observe that in [SCG+16] and [AD17], the size of public key is linear in the maximum size of the set of the revoked receivers R for one encryption in the encryption phase and have to specify the maximum receiver number in the setup phase. These restrictions seem unavoidable for doing their security reduction successfully. In contrast, there is no such restrictions in [LMG+16] and our proposed scheme. Although the final ciphertext CT' in [SCG+16, AD17] is a constant size, the ciphertext CT and CT' in both papers cannot preserve the privacy of the receivers and the revoked users identities. In other words, in their scheme, the ciphertext CT and CT' should attach the identity sets S and R, respectively. Both attachments seem rather unavoidable for decrypting the ciphertext correctly.

In contrast, both the CT and CT' in Lai et al. [LMG+16] are linear in the number of broadcast identity set for one encryption. Their scheme protects the identity in S, but, the revoked ciphertext CT' cannot hide the revoked users identities and R should be attached as part of ciphertext.

In our construction, the length of CT and the length of CT' have no any relationship. The length of each ciphertext only depends on the size of the input identity set of the corresponding algorithm. This property achieves that the ciphertext size without subjecting to the size of each identity set and optimizes the ciphertext size in each stage. Moreover, our scheme achieves full receiver privacy-preserving, that is, the ciphertext generated in each phase will not leak the user identity information, which has not been achieved in [SCG+16, LMG+16, AD17]. The work in [LMG+16, AD17] and ours achieve adaptive secure, while the scheme in [SCG+16] is only selective secure.

Furthermore, the Authorize algorithm in our proposed scheme can be performed several times by different executors to achieve multiple authorizations such that only the users belonged to all authorized identity set can obtain the message finally. Also in this case, our proposed scheme resists the collusion attack both for the message confidentiality and the identity privacy.

5.4 Security Analysis

In this section, we show that the proposed scheme achieves the security requirements defined in the security models previously. We use the proof technique of Boneh and Franklin [BF01] to prove the security of our scheme under the BDH assumption in

the random oracle model.

Theorem 5.1. *Let hash functions H_1, H_2 be random oracles. If the BDH assumption holds, the proposed scheme is IND-ID-CPA secure. Specifically, if there is an IND-ID-CPA adversary $\mathcal{A}$ with advantage ϵ against our scheme, there is an algorithm $\mathcal{B}$ that solves the BDH problem with advantage*

$$\epsilon' \geq \frac{\epsilon}{e \cdot n \cdot q_E \cdot q_{H_2}},$$

where n is the number of the identities in the encryption algorithm, q_E and q_{H_2} are the number of queries to private key and H_2 respectively by the adversary. e is the natural logarithm.

Proof. Suppose there exists an IND-ID-CPA adversary $\mathcal{A}$ that attacks our scheme with non-negligible advantage ϵ. We build a simulator (algorithm) $\mathcal{B}$ that can solve the BDH problem with advantage ϵ'. Let (P, aP, bP, cP) be a random instance of BDH problem taken as input by $\mathcal{B}$ and its goal is to compute $e(P,P)^{abc}$. In order to use $\mathcal{A}$ to solve the problem, $\mathcal{B}$ needs to simulate a challenger and respond all queries from $\mathcal{A}$. $\mathcal{B}$ works by interacting with $\mathcal{A}$ in an IND-ID-CPA game as follows.

Setup: $\mathcal{B}$ sets $P_{pub} = aP$ and creates the master public key as $mpk = (\mathbb{BG}, P, P_{pub}, H)$. Here H_1 and H_2 are viewed as random oracles controlled by $\mathcal{B}$.

H_1-queries: For an H_1 query on ID_i, $\mathcal{B}$ responds as follows. $\mathcal{B}$ maintains a list $\mathcal{L}_1$ of a tuple (ID_i, η_i, r_i, h_i). This list is initially empty. $\mathcal{B}$ first checks the $\mathcal{L}_1$.

1. If the query ID_i has already appeared in the $\mathcal{L}_1$ in a tuple (ID_i, η_i, r_i, h_i), it responses with h_i.

2. Otherwise, it randomly picks $r_i \in \mathbb{Z}_p$ and a random $\eta_i \in \{0,1\}$ with $\Pr[\eta_i = 0] = \delta$ for some δ (determine later). If $\eta_i = 0$, it computes $h_i = r_i bP$ otherwise, it computes $h_i = r_i P$. Then it adds the tuple (ID_i, η_i, r_i, h_i) to the $\mathcal{L}_1$ and returns h_i .

H_2-queries: For an H_2 query on (X_i, ID_i), $\mathcal{B}$ responds as follows. $\mathcal{B}$ maintains a list $\mathcal{L}_2$ of a tuple (X_i, ID_i, γ_i). This list is initially empty. $\mathcal{B}$ first checks the $\mathcal{L}_2$.

1. If the query (X_i, ID_i) has already appeared in the $\mathcal{L}_2$ in a tuple (X_i, ID_i, γ_i), it returns the corresponding γ_i.

2. Otherwise, $\mathcal{B}$ randomly picks a $\gamma_i \in \mathbb{G}$ as the value of $H_2(X_i, ID_i)$, and then adds the tuple (X_i, ID_i, γ_i) to the $\mathcal{L}_2$ and responds to $\mathcal{A}$ with γ_i.

Phase 1: In this phase, $\mathcal{A}$ can issue the private key queries on ID_i as needed. For each time, $\mathcal{B}$ first checks whether ID_i in the $\mathcal{L}_1$, otherwise it runs the H_1 query

and gets the corresponding η_i and r_i. If $\eta_i = 0$, $\mathcal{B}$ aborts. If $\eta_i = 1$, $\mathcal{B}$ computes $d_{ID_i} = sH_1(ID_i) = ar_iP = r_iP_{pub}$ and responds with d_{ID_i}.

Challenge: Once $\mathcal{A}$ decides the phase 1 is over, it outputs two distinct messages M_0, M_1 from the same message space and a challenge identity set $S^* = (ID_1, ID_2, \cdots, ID_n)$. We require that $\mathcal{A}$ has not queried the private key on $ID_i \in S^*$ in the phase 1. $\mathcal{B}$ performs as follows.

1. Randomly choose $C^* \in \mathbb{G}$ and set $C'^* = cP$.
2. For each $i \in S^*$, randomly choose $A_i^* \in \mathbb{G}$ and compute $x_i^* = H(ID_i)$.
3. Randomly choose $x_0^* \in \mathbb{Z}_p \backslash \{x_i\}$, $A_0^* \in \mathbb{G} \backslash \{A_i\}$, where $i \in S^*$. For $i = [0, n]$, compute

$$f_i(x) = \prod_{j=0, j\neq i}^{n} \frac{x - x_j^*}{x_i^* - x_j^*} = \sum_{j=0}^{n} a_{i,j} x^j,$$

$$C_i^* = \sum_{j=0}^{n} a_{j,i} A_j^*,$$

and set the challenge ciphertext as $CT^* = \left(C^*, C'^*, \{C_i^*\}_{i=0}^n\right)$.

Phase 2: $\mathcal{A}$ continues to issue more private key queries on $ID_i \notin S^*$. $\mathcal{B}$ responds the same as phase 1.

Guess: Finally, $\mathcal{A}$ outputs its guess $\mu' \in \{0, 1\}$ of μ.

The above completes the description of the simulation. It is not hard to verify that the simulation is indistinguishable from the scheme as the correctness and randomness hold.

If $\eta_j = 0$, we observe that $H_1(ID_j) = r_jbP$ and $d_{ID_j} = r_jabP$. We then have $e\left(d_{ID_j}, C_1^*\right) = e(P, P)^{r_jabc}$. At this point, $\mathcal{B}$ ignores the guess of $\mathcal{A}$ and picks a random tuple (X_j, ID_j, γ_j) from the list $\mathcal{L}_2$. It then obtains the corresponding r_j from $\mathcal{L}_1$, and outputs $X_j^{r_j^{-1}}$ as the solution to the given instance of BDH. To complete the security proof, it remains to show that $\mathcal{B}$ outputs the correct answer with advantage at least ϵ'.

The success of proof bases on the adversary's query on H_2. Let $\omega_i = \left(e(H_1(ID_i), P_{pub})^c, ID_i\right)$ where $ID_i \in S^*$. In the real scheme, $A_i^* = K + H_2(\omega_i)$. Before querying the H_2 value of ω_i, the result of $H_2(\omega_i)$ is unknown and random. From the view of the adversary, K is encrypted with a random number independent of ω_i. Therefore, A_i^* is a one-time pad. That is, the challenge ciphertext is a one-time pad. According to the assumption ($\mathcal{A}$ breaks our scheme with advantage ϵ), $\mathcal{A}$ must at least query H_2 on one ω_i with probability δ.

For ease exposition, we define following events: E_1, the simulation does not abort in private key query. E_2, at least one of the H_1 values of challenge identities contains

b. E_3, $\mathcal{A}$ chooses an identity where $\eta_i = 0$ to distinguish challenge message. E_4, $\mathcal{B}$ chooses the correct solution from the $\mathcal{L}_2$. $\mathcal{B}$ successfully solves the hard problem if and only if all events happen simultaneously. From the private key query, we know when each $\eta_i = 1$, simulation will not abort, thus

$$\Pr[E_1] = \Pr[\eta_i = 1, i = 1, 2, \cdots, q_E] = (1-\delta)^{q_E}.$$

It is easy to compute that $\Pr[E_2] = \delta$. Since η_i are secretly chosen by $\mathcal{B}$, from the point view of the adversary, η_i is randomly chosen. Therefore, we have $\Pr[E_3] = \frac{1}{n}$.

Finally, from the point view of the simulator, if the adversary can guess the correct μ' under the conditions that E_1, E_2, E_3 happen, it only knows that the correct solution of the hard problem is in the $\mathcal{L}_2$, but it cannot decide which one is, thus $\Pr[E_4] = \frac{1}{q_{H_2}}$. Since these four events are independent, we have

$$\begin{aligned}\epsilon' &= \Pr[E_1 \wedge E_2 \wedge E_3 \wedge E_4] \cdot \epsilon \\ &= \Pr[E_1] \cdot \Pr[E_2] \cdot \Pr[E_3] \cdot \Pr[E_4] \cdot \epsilon \\ &\geq (1-\delta)^{q_E} \cdot \delta \cdot \tfrac{\epsilon}{n \cdot q_{H_2}}.\end{aligned}$$

The function $(1-\delta)^{q_E} \cdot \delta$ is maximized at $\delta_{opt} = \frac{1}{q_E+1}$. Using δ_{opt}, we have

$$\epsilon' \geq \frac{\epsilon}{e \cdot n \cdot q_E \cdot q_{H_2}}.$$

This completes the proof. □

Theorem 5.2. *Let H_1, H_2 be random oracles. If the BDH assumption holds, the proposed scheme is ANON-ID-CPA secure. Specifically, suppose there is an ANON-ID-CPA adversary $\mathcal{A}$ that has advantage ϵ against our scheme, there is an algorithm $\mathcal{B}$ to solve the BDH problem with advantage*

$$\epsilon' \geq \frac{\epsilon}{e \cdot n \cdot q_E \cdot q_{H_2}},$$

where n is the number of the identities in the encryption algorithm, q_E and q_{H_2} are the number of queries to private key and H_2 respectively by the adversary. e is the natural logarithm.

Proof. Suppose there exists an ANON-ID-CPA adversary $\mathcal{A}$ that attacks our scheme with non-negligible advantage ϵ. We build a simulator (algorithm) $\mathcal{B}$ that can solve the BDH problem with advantage ϵ'. Let (P, aP, bP, cP) be a random instance of BDH problem taken as input by $\mathcal{B}$ and its goal is to compute $e(P,P)^{abc}$. In order to use $\mathcal{A}$ to solve the problem, $\mathcal{B}$ needs to simulate a challenger and respond all queries from $\mathcal{A}$. $\mathcal{B}$ works by interacting with $\mathcal{A}$ in an ANON-ID-CPA game as follows.

Setup: $\mathcal{B}$ sets $P_{pub} = aP$ and creates $mpk = (\mathbb{BG}, P, P_{pub}, H)$. Here H_1 and H_2 are viewed as random oracles controlled by $\mathcal{B}$.

H_1-queries: For an H_1 query on ID_i, $\mathcal{B}$ responds as follows. $\mathcal{B}$ maintains a list $\mathcal{L}_1$ of a tuple (ID_i, η_i, r_i, h_i). This list is initially empty. $\mathcal{B}$ first checks the $\mathcal{L}_1$.

1. If the query ID_i has already appeared in the $\mathcal{L}_1$ in a tuple (ID_i, η_i, r_i, h_i), it responses with h_i.

2. Otherwise, it randomly picks $r_i \in \mathbb{Z}_p$ and a random $\eta_i \in \{0, 1\}$ with $\Pr[\eta_i = 0] = \delta$ for some δ (determine later). If $\eta_i = 0$, it computes $h_i = r_i bP$ otherwise, it computes $h_i = r_i P$. Then it adds the tuple (ID_i, η_i, r_i, h_i) to the $\mathcal{L}_1$ and returns h_i .

H_2-queries: For an H_2 query on (X_i, ID_i), $\mathcal{B}$ responds as follows. $\mathcal{B}$ maintains a list $\mathcal{L}_2$ of a tuple (X_i, ID_i, γ_i). This list is initially empty. $\mathcal{B}$ first checks the $\mathcal{L}_2$.

1. If the query (X_i, ID_i) has already appeared in the $\mathcal{L}_2$ in a tuple (X_i, ID_i, γ_i), it returns the corresponding γ_i.

2. Otherwise, $\mathcal{B}$ randomly picks a $\gamma_i \in \mathbb{G}$ as the value of $H_2(X_i, ID_i)$, and then adds the tuple (X_i, ID_i, γ_i) to the $\mathcal{L}_2$ and responds to $\mathcal{A}$ with γ_i.

Phase 1: In this phase, $\mathcal{A}$ can issue the private key queries on ID_i as needed. For each time, $\mathcal{B}$ first checks whether ID_i in the $\mathcal{L}_1$, otherwise it runs the H_1 query and gets the corresponding η_i and r_i. If $\eta_i = 0$, $\mathcal{B}$ aborts. If $\eta_i = 1$, $\mathcal{B}$ computes $d_{ID_i} = sH_1(ID_i) = ar_i P = r_i P_{pub}$ and responds with d_{ID_i}.

Challenge: Once $\mathcal{A}$ decides that the phase 1 is over, it outputs a challenge message M^* and two distinct identity sets $S_0 = \{ID_{0,1}, ID_{0,2}, \cdots, ID_{0,n}\}, S_1 = \{ID_{1,1}, ID_{1,2}, \cdots, ID_{1,n}\}$. We require that $\mathcal{A}$ has not queried the private key on $ID_i \in S_0 \triangle S_1$ in the phase 1. $\mathcal{B}$ picks a random bit $\mu \in \{0, 1\}$ and performs as follows.

1. Randomly choose $K^* \in \mathbb{G}$ and compute $C^* = K^* + M^*$, $C'^* = cP$. For each $ID_i \in S_\mu$, compute $x_i^* = H(ID_i)$.

2. For each $ID_i \in S_\mu \backslash S_{1-\mu}$, randomly choose $A_i^* \in \mathbb{G}$. For each $ID_i \in S_0 \cap S_1$, it firstly gets r_i and c_i from $\mathcal{L}_1$ (If ID_i is not in the list $\mathcal{L}_1$, it runs H_1 queries to get r_i and η_i). If $\eta_i = 0$, it randomly chooses $A_i^* \in \mathbb{G}$. If $\eta_i = 1$, it computes $X_i = e(aP, cP)^{r_i}$ and checks whether the tuple (X_i, ID_i) is in one of the tuples (X_i, ID_i, γ_i) in the $\mathcal{L}_2$. If yes, set $\tau_i^* = \gamma_i$. Otherwise, it randomly picks $\tau_i^* \in \mathbb{G}$ and adds the new tuple (X_i, ID_i, τ_i^*) to the list $\mathcal{L}_2$. It then computes $A_i^* = K^* + \tau_i^*$.

3. Randomly choose $x_0^* \in \mathbb{Z}_p$, $A_0^* \in \mathbb{G}$. For $i = [0, n]$, compute the polynomial functions

$$f_i(x) = \prod_{j=0, j\neq i}^{n} \frac{x - x_j^*}{x_i^* - x_j^*} = \sum_{j=0}^{n} a_{i,j} x^j.$$

Finally, it computes

$$C_i^* = \sum_{j=0}^{n} a_{j,i} A_j^*,$$

and sets $CT^* = (C^*,\ C'^*,\ \{C_i^*\}_{i=0}^n)$.

Phase 2: $\mathcal{A}$ continues to issue private key queries as needed with the restriction established in the challenge phase. $\mathcal{B}$ responds the same as in phase 1.

Guess: Finally, $\mathcal{A}$ outputs its guess $\mu' \in \{0, 1\}$.

The above completes the description of the simulation. It is not hard to verify that the simulation is indistinguishable from the scheme as the correctness and randomness hold. At this point, $\mathcal{B}$ ignores the guess of $\mathcal{A}$ and picks a random tuple (X_j, ID_j, γ_j) from the list $\mathcal{L}_2$. It then obtains the corresponding r_j from $\mathcal{L}_1$ and outputs $X_j^{r_j^{-1}}$ as the solution to the given instance of BDH. We have noted that $\mathcal{A}$ can obtain the private keys for $ID_i \in S_0 \cap S_1$. But this will not help the adversary to distinguish the identity set from our setting. Similar to the analysis in Theorem 5.1, we have $\epsilon' \geq \frac{\epsilon}{e \cdot n \cdot q_E \cdot q_{H_2}}$. The details of analysis are omitted here.

This completes the proof. □

Theorem 5.3. *Let H_1, H_2 be random oracles. The proposed scheme is IND-aID-CPA secure if the BDH assumption holds.*

Proof. In this proof, we allow the adversary $\mathcal{A}$ to query the private key on ID_i where $ID_i \in \big((S^* \backslash L^*) \cup (L^* \backslash S^*)\big)$. But we claim that these information cannot provide $\mathcal{A}$ any help to distinguish the challenge messages. From the decryption algorithm, if $ID_i \in S^* \backslash L^*$, it will get a random $A_i'^* \neq A_i^*$. If $ID_i \in L^* \backslash S^*$, it will get a correct $A_i'^*$ generated by the authorization algorithm, but this $A_i'^*$ also a random element in $\mathbb{G}$ and does not equal to A_i^* generated by the encryption algorithm. Therefore, even with the private key d_{ID_i} where $ID_i \in \big((S^* \backslash L^*) \cup (L^* \backslash S^*)\big)$, the adversary still cannot obtain the correct A_i^* which hides the encryption key. Thus, if the scheme is IND-ID-CPA secure, it is IND-aID-CPA secure, namely the security against IND-ID-CPA attacks implies the security against IND-aID-CPA attacks. We yield Theorem 5.1. □

Theorem 5.4. *Let hash functions H_1, H_2 be random oracles. If the BDH assumption holds, the proposed scheme is ANON-aID-CPA secure. Specifically, if there is*

an ANON-aID-CPA adversary $\mathcal{A}$ with advantage ϵ against our scheme, there is an algorithm $\mathcal{B}$ that solves the BDH problem with advantage

$$\epsilon' \geq \frac{\epsilon}{e \cdot n \cdot q_E \cdot q_{H_2}},$$

where n is the number of the identities in the encryption algorithm, q_E and q_{H_2} are the number of queries to private key and H_2 respectively by the adversary. e is the natural logarithm.

Proof. Suppose there exists an ANON-aID-CPA adversary $\mathcal{A}$ that attacks our scheme with non-negligible advantage ϵ. We build a simulator (algorithm) $\mathcal{B}$ that can solve the BDH problem with advantage ϵ' by running $\mathcal{A}$. Let (P, aP, bP, cP) be a random instance of BDH problem taken as input by $\mathcal{B}$ and its goal is to compute $e(P,P)^{abc}$. In order to use $\mathcal{A}$ to solve the problem, $\mathcal{B}$ needs to simulate a challenger and respond all queries from $\mathcal{A}$. $\mathcal{B}$ works by interacting with $\mathcal{A}$ in an ANON-aID-CPA game as follows.

Setup: The simulator $\mathcal{B}$ sets $P_{pub} = aP$ and creates the master public key as $mpk = (\mathbb{BG}, P, P_{pub}, H)$. Here H_1 and H_2 are viewed as random oracles controlled by $\mathcal{B}$.

H_1-queries: For an H_1 query on ID_i, $\mathcal{B}$ responds as follows. $\mathcal{B}$ maintains a list $\mathcal{L}_1$ of a tuple (ID_i, η_i, r_i, h_i). This list is initially empty. $\mathcal{B}$ first checks the $\mathcal{L}_1$.

1. If the query ID_i has already appeared in the $\mathcal{L}_1$ in a tuple (ID_i, η_i, r_i, h_i), it responses with h_i.

2. Otherwise, it randomly picks $r_i \in \mathbb{Z}_p$ and a random $\eta_i \in \{0,1\}$ with $\Pr[\eta_i = 0] = \delta$ for some δ (determine later). If $\eta_i = 0$, it computes $h_i = r_i bP$ otherwise, it computes $h_i = r_i P$. Then it adds the tuple (ID_i, η_i, r_i, h_i) to the $\mathcal{L}_1$ and returns h_i .

H_2-queries: For an H_2 query on (X_i, ID_i), $\mathcal{B}$ responds as follows. $\mathcal{B}$ maintains a list $\mathcal{L}_2$ of a tuple (X_i, ID_i, γ_i). This list is initially empty. $\mathcal{B}$ first checks the $\mathcal{L}_2$.

1. If the query (X_i, ID_i) has already appeared in the $\mathcal{L}_2$ in a tuple (X_i, ID_i, γ_i), it returns the corresponding γ_i.

2. Otherwise, $\mathcal{B}$ randomly picks a $\gamma_i \in \mathbb{G}$ as the value of $H_2(X_i, ID_i)$, and then adds the tuple (X_i, ID_i, γ_i) to the $\mathcal{L}_2$ and responds to $\mathcal{A}$ with γ_i.

Phase 1: In this phase, $\mathcal{A}$ can issue the private key queries on ID_i as needed. For each time, $\mathcal{B}$ first checks whether ID_i in the $\mathcal{L}_1$, otherwise it runs the H_1 query and gets the corresponding η_i and r_i. If $\eta_i = 0$, $\mathcal{B}$ aborts. If $\eta_i = 1$, $\mathcal{B}$ computes $d_{ID_i} = sH_1(ID_i) = ar_iP = r_iP_{pub}$ and responds with d_{ID_i}.

Challenge: Once $\mathcal{A}$ decides that the phase 1 is over, it outputs a challenge message M^*, an identity set $S^* = \{ID_1, ID_2, \cdots, ID_n\}$ and two distinct authorized sets $L_0 = \{ID_{0,1}, ID_{0,2}, \cdots, ID_{0,k}\}$, $L_1 = \{ID_{1,1}, ID_{1,2}, \cdots, ID_{1,k}\}$. We require that $\mathcal{A}$ has not queried the private key on ID_i in the phase 1, where $ID_i \in S^* \cap (L_0 \triangle L_1)$. $\mathcal{B}$ picks a random bit $\mu \in \{0, 1\}$ and performs as follows.

1. Randomly choose $K^* \in \mathbb{G}$, $x_0^* \in \mathbb{Z}_p$, $A_0^* \in \mathbb{G}$ and compute $C^* = K^* + M^*$, $C'^* = cP$. For each $ID_i \in S^*$, compute $x_i^* = H(ID_i)$.

2. For each $ID_i \in (S^* \cap L_0 \cap L_1)$, it first gets r_i and η_i from $\mathcal{L}_1$. If $\eta_i = 0$, it randomly chooses $A_i^* \in \mathbb{G}$. If $\eta_i = 1$, it computes $X_i = e(aP, cP)^{r_i}$ and checks whether the tuple (X_i, ID_i) is in one of the tuples (X_i, ID_i, γ_i) in the $\mathcal{L}_2$. If yes, it sets $\tau_i^* = \gamma_i$. Otherwise, it randomly picks $\tau_i^* \in \mathbb{G}$ and adds the new tuple (X_i, ID_i, τ_i^*) to the list $\mathcal{L}_2$. Then it computes $A_i^* = K^* + \tau_i^*$. For each $ID_i \in S^* \backslash (S^* \cap L_0 \cap L_1)$, randomly choose $A_i^* \in \mathbb{G}$.

3. For $i = [0, n]$, compute

$$f_i(x) = \prod_{j=0, j\neq i}^{n} \frac{x - x_j^*}{x_i^* - x_j^*} = \sum_{j=0}^{n} a_{i,j} x^j.$$

   After obtaining all $f_i(x)$, it computes

$$C_i^* = \sum_{j=0}^{n} a_{j,i} A_j^*, \ i = 0, 1, 2, \cdots, n.$$

4. For each $i \in L_\mu$, compute $x_i = H(ID_i)$. After computing all x_i, it then computes for $i \in L_\mu$

$$g_i(x) = \prod_{j=0, j\neq i}^{k} \frac{x - x_j}{x_i - x_j} = \sum_{j=0}^{k-1} b_{i,j} x^j \mod p,$$

$$A_i'^* = C_0^* + x_i C_1^* + x_i^2 C_2^* + \cdots + x_i^n C_n^*.$$

   Finally, it computes

$$C_i'^* = \sum_{j=1}^{k} b_{j,i-1} A_j'^*.$$

   and sets $CT'^* = \left(C^*, C'^*, \{C_i'^*\}_{i=0}^k\right)$.

Phase 2: $\mathcal{A}$ continues to issue private key queries as needed with the restriction established in the challenge phase. $\mathcal{B}$ responds the same as in phase 1.

Guess: Finally, $\mathcal{A}$ outputs its guess $\mu' \in \{0,1\}$ of μ.

The above completes the description of the simulation. It is not hard to see that the simulation is indistinguishable from the scheme as the correctness and randomness hold. At this point, $\mathcal{B}$ ignores the guess of $\mathcal{A}$ and picks a random tuple (X_j, ID_j, γ_j) from the list $\mathcal{L}_2$. It then obtains the corresponding r_i in $\mathcal{L}_1$ and outputs $X_j^{r_j^{-1}}$ as the solution to the given instance of BDH. The only way for the adversary to distinguish L_0 and L_1 is by querying the values of H_2 on $ID_i \in S^* \cap (L_\mu \backslash L_{1-\mu})$. Similar to the analysis in the Theorem 5.1, we have $\epsilon' \geq \frac{\epsilon}{e \cdot n \cdot q_E \cdot q_{H_2}}$. We omit the details of the analysis here and yield the Theorem 5.4.

This completes the proof. □

5.5 Conclusion

In this chapter, we presented a new notion of fully privacy-preserving identity-based broadcast encryption with authorization. It allows any third party to perform the authorization computations without knowing the message and the receiver identity. Only the user whose identity belongs to the intersection set of the broadcast identity set and the authorized identity set is able to obtain the message. The final ciphertext reveals nothing about the encrypted message and the identity information of the receivers and the authorized users. A concrete construction has been presented and the security is proved in the random oracle model. Furthermore, the proposed scheme supports multiple authorizations in a way that only the user whose identity belongs to all authorized set can decrypt the ciphertext successfully.

Chapter 6

IBBE for Inner Products

This chapter introduces a new notion of identity-based broadcast encryption for inner products, which is a variant of identity-based broadcast encryption. The proposed scheme features the metrics of both the identity-based broadcast encryption and the inner product encryption introduced by Abdalla et al. [ABCP15]. The original scheme was submitted to *The Computer Journal* and is under review.

6.1 Introduction

Broadcast Encryption. The concept of broadcast encryption (BE) was introduced by Fiat and Naor in [FN94] as a generalization of public key encryption aiming to efficiently share a message among a group of users over a public broadcast channel. In a BE scheme, a message is encrypted under a set of users selected by the encryptor in a way that only those users can decrypt the encrypted message by providing their private keys. While the users who are not chosen in the encryption learn nothing about the message even they collude. BE has been widely deployed in the real-life applications, such as Pay TV. Identity-based broadcast encryption (IBBE) [Del07, SF07] is a variant of BE where a user's public key is replaced by an arbitrary string which can uniquely identify the user, such as an email address or a phone number. IBBE has shown its merit in key management and has been studied extensively. A number of remarkable IBBE schemes have been considered in [GW09, LPQ12, BWZ14].

Inner Product Encryption. In 2015, Abdalla et al. [ABCP15] considered the inner product encryption (IPE) as a special functional encryption which has expressive practical applications. In the IPE system, a message is described as a vector $\vec{x}$ from an inner product space and private keys are generated for vectors $\vec{y}_i$ from the same space. A user with a private key $sk_{\vec{y}}$ of vector $\vec{y}$ enables to decrypt an encrypted message $\vec{x}$ and learns the inner product $\langle \vec{x}, \vec{y} \rangle$ without knowing $\vec{x}$. Differing from inner product predicate schemes [SSW09, OT12, OT10, AAB$^+$15], IPE computes the real value of inner products via decryption while the predicate encryption retrieves the message if and only if the inner product is zero. IPE has been studied extensively

since it introduced as it is extremely useful in the context of descriptive statistics. For example, it can be used to compute the weighted mean of a collection of information without leaking the content of the message. More applications of IPE can be found in [ABCP15]. Since the work of Abdalla et al., several schemes have been presented. IPE with function privacy[1] are studied in [BJK15, DDM16]. Multi-input IPE has been considered in [AGRW17, LL16, KLM$^+$16].

In the IBBE (or broadcast encryption), as we aforementioned, users are permitted to either get the message or learn nothing. While in the IPE, users can only obtain the inner product $\langle \vec{x}, \vec{y} \rangle$ via decryption and learn nothing about the message $\vec{x}$ (so-called further protect the message), where $\vec{y}$ is associated with the decryption key. If we apply an IBBE scheme to encrypt $\vec{x}$, we are unable to protect $\vec{x}$ like in the IPE. On the other hand, none of the existing IPE schemes allows the encryptor to decide who is permitted to decrypt the encrypted message like in the IBBE, as we observe that any private key in the IPE can be used to decrypt the encrypted message and learn the corresponding inner product. In a nutshell, the comparison of IBBE and IPE can be summarized as follows.

- IBBE can determine who is able to obtain the message, but it cannot further protect the message.
- IPE cannot determine who is able to learn the inner products, but it can further protect the message.

Our motivation is to introduce a system which captures the merits of both IBBE and IPE. More precisely, a system in which the decryption only gives the inner product associated with the encrypted message, and the encryptor can determine who are allowed to learn the inner product by providing private keys. This new system can be applied in those scenarios where IPE is desired and the encryptor also wants to control who are allowed to obtain the inner products. One might think that we could first use an IPE to encrypt the message and then perform an IBBE to encrypt the IPE ciphertext for a group of specified users. Unfortunately, this trivial solution gives rise to a security threat. That is, once the decryption result of IBBE is made public (e.g. one of the specified users exposes the decryption result of IBBE), any private key generated in the IPE system can obtain the corresponding inner product.

In this chapter, we further explore the notion of IBBE and introduce an extension of IBBE called *identity-based broadcast encryption for inner product* (IBBE-IP). In the IBBE-IP, each user is associated with an identity ID and a vector $\vec{y}$, which

[1]Function privacy requires that functional keys reveal no necessary information about the functionality. An IPE scheme is function private if the key for vector $\vec{y}$ reveals nothing about $\vec{y}$. See, e.g., [BRS13, BS15] for more details about function privacy.

is chosen by the user or the private key generator (PKG) depending on the application during the key generation. A message $\vec{x}$ is encrypted with a set of identities $\mathbb{S}$ chosen by the encryptor without knowing their vectors. A user with a private key of $(ID, \vec{y})$ can decrypt the encrypted message $\vec{x}$ and learn the inner product $\langle \vec{x}, \vec{y} \rangle$ without the knowledge of $\vec{x}$ if and only if $ID \in \mathbb{S}$. Therefore, the notion of IBBE-IP can further protect the message like in the IPE. The confidentiality of message in the IBBE-IP can be protected as long as the number of selected identities is less than the length of vectors. Better than the existing IPE schemes, which only allow to query bounded number of private keys, the proposed IBBE-IP notion supports unbounded private key queries[2].

We give an instantiation of IBBE-IP. The private key size in the proposed scheme is constant and the size of ciphertext is linear in the length of vectors. The security of the proposed scheme relies on the intractability of one specific q-type problem defined by Boneh, Boyen and Goh in [BBG05] and the scheme is proved to be IND-sIDV-CPA secure (defined in Section 6.2.2) in the random oracle model.

Organization. The rest of this chapter is organized as follows. In Section 6.2, we define the identity-based broadcast encryption for inner product and the corresponding security notions. A new complexity assumption which the security of the proposed scheme based on is showed in Section 6.3. We describe the proposed scheme in Section 6.4 and formally analyze its security in Section 6.5. Finally, we conclude this chapter in Section 6.6.

6.2 Identity-Based Broadcast Encryption for Inner Product

In this section, we formally define the syntax of identity-based broadcast encryption for inner product (IBBE-IP) and its security models.

6.2.1 Definitions

An IBBE-IP scheme is defined by the following algorithms.

- $\mathsf{Setup}(1^\lambda, n)$: Taking as input a security parameter λ and n the length of vectors, it outputs a master public key mpk and a master secret key msk.

[2]Here, the IPE schemes refer to those constructed in the public key setting. In this chapter, if we say IPE, it means the public-key IPE unless otherwise stated explicitly. In the related work, we will mention that there are some IPE works in the private key setting can achieve unbounded private keys queries in the specified security models.

The master public key mpk is publicly known, while the master secret key msk is kept secretly. The master public key contains the descriptions of a key space $\mathcal{K}$ and a message space $\mathcal{X}$.

- $\mathsf{KeyGen}(mpk, msk, ID, \vec{y})$: Taking as input the master key pair (mpk, msk), an identity ID and a vector $\vec{y} \in \mathcal{K}^n$, it outputs a private key $sk_{ID,\vec{y}}$ of $(ID, \vec{y})$.
- $\mathsf{Encrypt}(mpk, \mathbb{S}, \vec{x})$: Taking as input the master public key mpk, a set of identities $\mathbb{S}$ with $|\mathbb{S}| < n$ and a message $\vec{x} \in \mathcal{X}^n$, it outputs a ciphertext CT.
- $\mathsf{Decrypt}(mpk, CT, \mathbb{S}, sk_{ID,\vec{y}}, \vec{y})$: Taking as input the master public key mpk, a ciphertext CT together with an identity set $\mathbb{S}$ and a private key $sk_{ID,\vec{y}}$ of $(ID, \vec{y})$, it outputs either a value $\langle \vec{x}, \vec{y} \rangle$ or $\perp$.

For correctness, it should satisfy the following requirements. For simplicity, we omit the input of the master public key mpk in each algorithm. We suppose that (mpk, msk) are the result of calling $\mathsf{Setup}(1^\lambda, n)$, and $sk_{ID,\vec{y}}, CT$ are the result of calling $\mathsf{KeyGen}(msk, ID, \vec{y})$ and $\mathsf{Encrypt}(\mathbb{S}, \vec{x})$ respectively. We then require that if $ID \in \mathbb{S}$, we have

$$\Pr\left[\mathsf{Decrypt}(CT, \mathbb{S}, sk_{ID,\vec{y}}, \vec{y}) = \langle \vec{x}, \vec{y} \rangle : \begin{array}{l} (mpk, msk) \leftarrow \mathsf{Setup}(1^\lambda, n); \\ sk_{ID,\vec{y}} \leftarrow \mathsf{KeyGen}(msk, ID, \vec{y}) \\ CT \leftarrow \mathsf{Encrypt}(\mathbb{S}, \vec{x}); \end{array}\right] = 1.$$

In this chapter, we focus on the inner product functionality over $\mathbb{Z}_p$ where the key space $\mathcal{K}^n$ and message space $\mathcal{X}^n$ both consisting of vectors in $\mathbb{Z}_p$ of length n. We require that it is $\langle \vec{x}, \vec{y} \rangle$ when $\langle \vec{x}, \vec{y} \rangle$ is from a fit polynomial range of values inside $\mathbb{Z}_p$ as stated in [ABCP15, ALS16], since this will allow a decryption algorithm to compute it as a discrete logarithm in a group where discrete logarithm is generally hard. A summary of IBBE, IPE and IBBE-IP is shown in table.6.1, where Dec. is the short for decryption.

Remark. In the definition of IBBE-IP, when $n = 1$ and the vector of each user equals to "1", the IBBE-IP scheme is actually the IBBE scheme as $\left\langle \vec{x}, \vec{1} \right\rangle = \vec{x}$. This is why we say that the IBBE-IP is an extension of IBBE. In the encryption algorithm, we require the number of selected identities must be less than n for one encryption. It implies the number of inner products obtained from the decryption is less than n (in our setting) which is determined by the functionality and appears

	Number of Keys	Encrypt	Decrypt	Dec. Result	Dec. Condition
IBBE	unbounded	$(M, \mathbb{S})$	$(CT, \mathbb{S}, sk_{ID_j}, ID_j)$	M	$ID_j \in \mathbb{S}$
IPE	bounded	$\vec{x}$	$\left(CT, sk_{\vec{y}_j}, \vec{y}_j\right)$	$\langle \vec{x}, \vec{y}_j \rangle$	None
IBBE-IP	unbounded	$(\vec{x}, \mathbb{S})$	$\left(CT, \mathbb{S}, sk_{ID_j, \vec{y}_j}, \vec{y}_j\right)$	$\langle \vec{x}, \vec{y}_j \rangle$	$ID_j \in \mathbb{S}$

Table 6.1: Comparison of IBBE, IPE and IBBE-IP.

in all IPE schemes. With n inner products, the adversary can retrieve the message. From table 6.1, who can learn the inner product in the IBBE-IP is finely controlled by the encryptor and it can issue unbounded private keys.

6.2.2 Security Notions

Now, we describe the security of IBBE-IP. From the definition of IBBE-IP, we note that the output of decryption is the inner product over $\mathbb{Z}_p$. Users can easily obtain new inner products via collusion without performing any additional decryption. This issue happens to all IPE schemes in the literature and has been considered as something inherent to the functionality itself. As pointed out by Agrawal et al. [ALS16], collusion is permitted in the IPE. Therefore, we do not consider such security issue in our defined security models.

Based on the above statement, the security of an IBBE-IP scheme requires that for an encrypted message $\vec{x}$, only the user with identity $ID \in \mathbb{S}$ can compute $\langle \vec{x}, \vec{y} \rangle$ via decryption using the private key associated with $(ID, \vec{y})$. Attackers whose identities are not in $\mathbb{S}$ are unable to compute the inner products even they can access users' vectors and the corresponding private keys. We define security against chosen-plaintext attacks (IND-CPA security, for short) for IBBE-IP via the security game played by a challenger and an adversary below. Both the challenger and the adversary are given the length n of vectors as input.

- **Setup:** The challenger chooses a security parameter λ and runs algorithm $\mathsf{Setup}(1^\lambda, n)$ to obtain a master public key mpk, then it sends mpk to the adversary.

- **Phase 1:** The adversary issues private key queries on $(ID_i, \vec{y}_i)$ as needed. The challenger runs the KeyGen algorithm to generate the private key $sk_{ID_i, \vec{y}_i}$ and forwards $sk_{ID_i, \vec{y}_i}$ to the adversary.

- **Challenge:** Once the adversary decides that the phase 1 is over, it outputs an identity set $\mathbb{S}^* = \{ID_1^*, ID_2^*, \cdots, ID_{s^*}^*\}$ with $s^* < n$ and two distinct messages $\vec{x}_0$ and $\vec{x}_1$ from the same space for challenge. We require that if $(ID_i^*, \vec{y}_i^*)$ has queried its private key, then $\langle \vec{x}_0, \vec{y}_i^* \rangle = \langle \vec{x}_1, \vec{y}_i^* \rangle$. The challenger picks a random

bit $\mu \in \{0,1\}$ and generates a challenge ciphertext CT^* for $\vec{x}_\mu$ under $\mathbb{S}^*$ and returns CT^* to the adversary.

– **Phase 2:** The adversary issues more private key queries with the restriction established in the challenge phase. The challenger responds as in phase 1.

– **Guess:** Finally, the adversary outputs a guess $\mu' \in \{0,1\}$ and wins the game if $\mu = \mu'$.

We refer to such an adversary as an IND-CPA adversary and define the advantage of the adversary in winning the game as

$$\mathsf{adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{IBBE\text{-}IP}}(\lambda) = \left| \Pr[\mu = \mu'] - \frac{1}{2} \right|.$$

Definition 6.1. *We say that an IBBE-IP scheme is IND-CPA if for any probabilistic polynomial time IND-CPA adversary, the advantage* $\mathsf{adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{IBBE\text{-}IP}}(\lambda)$ *is negligible.*

Selective Security. The security analysis of the proposed IBBE-IP scheme makes use of a weaker notion of security called *selective security*. In the selective security model, the adversary should output the challenge identity set $\mathbb{S}^*$ and the challenge messages $\vec{x}_0, \vec{x}_1$ before seeing the system master public key, and then tells apart the challenge ciphertext is generated under which message. We denote this model by IND-sIDV-CPA. Similarly, IND-sIDV-CPA security is defined via a security game played by a challenger and an adversary . Both the challenger and the adversary are given the length n of vectors as input.

– **Init:** The adversary outputs a target identity set $\mathbb{S}^* = \{ID_1^*, ID_2^*, \cdots, ID_{s^*}^*\}$ with $s^* < n$ and two distinct messages $\vec{x}_0, \vec{x}_1$ that it wants to challenge.

– **Setup:** The challenger chooses a security parameter λ and runs algorithm $\mathsf{Setup}(1^\lambda, n)$ to obtain a master public key mpk, then it sends mpk to the adversary.

– **Phase 1:** The adversary issues private key queries on $(ID_i, \vec{y}_i)$ as needed. If $ID_i \in \mathbb{S}^*$, we require $\langle \vec{x}_0, \vec{y}_i \rangle = \langle \vec{x}_1, \vec{y}_i \rangle$. The challenger runs the KeyGen algorithm to generate the private key $sk_{ID_i,\vec{y}_i}$ and forwards $sk_{ID_i,\vec{y}_i}$ to the adversary.

– **Challenge:** Once the adversary decides that the phase 1 is over, the challenger picks a random bit $\mu \in \{0,1\}$ and generates a challenge ciphertext CT^* for $\vec{x}_\mu$ under $\mathbb{S}^*$ and returns CT^* to the adversary.

– **Phase 2:** The adversary issues more private key queries. The challenger responds as in phase 1.

– **Guess:** Finally, the adversary outputs a guess $\mu' \in \{0,1\}$ and wins the game if $\mu = \mu'$.

We refer to such an adversary as an IND-sIDV-CPA adversary and define the advantage of the adversary in winning the game as

$$\mathsf{adv}_{\mathsf{IBBE\text{-}IP}}^{\mathsf{IND\text{-}sIDV\text{-}CPA}}(\lambda) = \left| \Pr[\mu = \mu'] - \frac{1}{2} \right|.$$

Definition 6.2. *We say that an IBBE-IP scheme is IND-sIDV-CPA secure if for any PPT IND-sIDV-CPA adversary, the advantage* $\mathsf{adv}_{\mathsf{IBBE\text{-}IP}}^{\mathsf{IND\text{-}sIDV\text{-}CPA}}(\lambda)$ *is negligible.*

6.3 New Complexity Assumption

In this section, we define one specific q-type problem that we call the *augmented general decisional Diffie-Hellman exponent problem* denoted by AGDDHE. We have proved that the (f, g)-AGDDHE problem has generic security as it fits the general Diffie-Hellman exponent problem framework of [BBG05]. The security of our IBBE-IP scheme relies on the hardness of (f, φ)-AGDDHE problem.

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic groups of prime order p. A map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is called bilinear map if it satisfies: (1) for all $g_1, g_2 \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, we have $e\left(g_1^a, g_2^b\right) = e\left(g_1, g_2\right)^{ab}$. (2) $e(g_1, g_2) \neq 1$ and there is an efficient algorithm to compute $e(g_1, g_2)$ for all $g_1, g_2 \in \mathbb{G}$. A bilinear group $\mathbb{BG} = (\mathbb{G}, \mathbb{G}_T, e, p)$ is composed of the above defined objects. The (f, φ)-AGDDHE problem is depicted as follow: Let g_0, h_0 be the random distinct generators of $\mathbb{G}$, q, n be distinct integers.

Input: A random polynomial $f(x)$ of degree q over $\mathbb{Z}_p^*$ with $f(a) \neq 0$, a set $\Lambda = \{z_{q+1}, z_{q+2}, \cdots, z_{q+n}\}$ whose components are pairwise distinct elements chosen uniformly at random from $\mathbb{Z}_p^*$, which define the polynomial

$$\varphi(z) = \prod_{i=q+1}^{q+n} (z - z_i).$$

The values

$$h_0,\ h_0^a,\ h_0^{a^2}, \cdots,\ h_0^{a^{q+n}}, \tag{6.1}$$

$$h_0^{bc}, h_0^{bca}, h_0^{bca^2}, h_0^{bca^3}, \cdots, h_0^{bca^{2n}}, \tag{6.2}$$

$$g_0,\ g_0^{f(a)},\ g_0^{af(a)},\ g_0^{a^2 f(a)}, \cdots,\ g_0^{a^{q-1} f(a)}, \tag{6.3}$$

$$h_0^{\gamma bc\varphi(a)}, g_0^{bcf(a)}, g_0^{af^2(a)},\ g_0^{\gamma af(a)},\ g_0^{\gamma af^2(a)}, \tag{6.4}$$

and an element Z from $\mathbb{G}_T$, where a, b, c and γ are unknown random exponents of $\mathbb{Z}_p^*$.

Output: a bit μ.

We say that the problem is correctly solved if the output is

$$\mu = \begin{cases} 1 & Z = e(g_0, h_0)^{\gamma bcf(a)}, \\ 0 & Z \neq e(g_0, h_0)^{\gamma bcf(a)}. \end{cases}$$

In other words, the goal of the (f, φ)-AGDDHE problem is to distinguish whether Z is equal to $e(g_0, h_0)^{\gamma bcf(a)}$ or to a random value of $\mathbb{G}_T$.

Let us denote by $\mathcal{I}\left(f(x), \Lambda, a, b, c, \gamma, Z\right)$ the input of the instance, true the event that Z is equal to $e(g_0, h_0)^{\gamma bcf(a)}$ and by false the event that Z is not equal to $e(g_0, h_0)^{\gamma bcf(a)}$. We then define the advantage of an algorithm $\mathcal{D}$ in solving the (f, φ)-AGDDHE problem in $\mathbb{BG}$ as

$$\begin{aligned} \mathsf{Adv}_{\mathcal{D}}^{(\mathsf{f},\varphi)\text{-}\mathsf{AGDDHE}}(\lambda) \;=\; & \Big| \Pr\left[\mathcal{D}(\mathcal{I}\left(f(x), \Lambda, a, b, c, \gamma, Z\right)) = 1 \Big| \mathsf{true}\right] \\ & - \Pr\left[\mathcal{D}(\mathcal{I}\left(f(x), \Lambda, a, b, c, \gamma, Z\right)) = 1 \Big| \mathsf{false}\right] \Big|, \end{aligned}$$

where the probability is taken over all random choices and over the random coins of $\mathcal{D}$.

Theorem 6.1. *The defined (f, φ)-AGDDHE problem is one of the GDDHE problems fulfilling the hardness conditions defined in [BBG05].*

Proof. Let $h_0 = g_0^t$. For simplicity, the (f, φ)-AGDDHE problem can be reformulated as

$$\begin{aligned} P &= \left(\begin{array}{l} t,\ ta,\ ta^2,\ ta^3, \cdots,\ ta^{q+n}, \\ tbc,\ tbca,\ tbca^2,\ tbca^3, \cdots,\ tbca^{2n}, \\ 1,\ f(a),\ af(a),\ a^2 f(a), \cdots,\ a^{q-1} f(a), \\ t\gamma bc\varphi(a),\ bcf(a),\ af^2(a),\ \gamma af(a),\ \gamma af^2(a), \end{array} \right), \\ Q &= 1, \\ F &= t\gamma bcf(a). \end{aligned}$$

We need to show that F is independent of (P, Q), that is, no not-all-zero coefficients $\{x_{i,j}\}$ and y_1 exist such that

$$t\gamma bcf(a) = F = \sum x_{i,j} d_i d_j + y_1,$$

where $d_i, d_j \in P$. For more details, the reader is referred to [BBG05, Del07]. All

possible multiplications of any two elements from P must contain $t\gamma bc$ in order to satisfy the above equation. By making all possible multiplications listed in P', we want to prove that no linear combination among the elements from the P' below leads to P.

$$P' = \begin{pmatrix} t\gamma bc \cdot \varphi(a),\ t\gamma bc \cdot \varphi(a)f(a),\ t\gamma bc \cdot a\varphi(a)f(a),\ t\gamma bc \cdot a^2\varphi(a)f(a), \cdots, \\ t\gamma bc \cdot a^{q-1}\varphi(a)f(a),\ t\gamma bc \cdot a\varphi(a)f^2(a) \\ t\gamma bc \cdot af(a),\ t\gamma bc \cdot a^2f(a),\ t\gamma bc \cdot a^3f(a), \cdots,\ t\gamma bc \cdot a^{2n+1}f(a), \\ t\gamma bc \cdot af^2(a),\ t\gamma bc \cdot a^2f^2(a),\ t\gamma bc \cdot a^3f^2(a), \cdots,\ t\gamma bc \cdot a^{2n+1}f^2(a). \end{pmatrix}.$$

As $t\gamma bc \cdot a\varphi(a)f^2(a)$ can be represented by the last line, P' can be simplified by removing $t\gamma bc$ as below

$$\begin{pmatrix} \varphi(a),\ \varphi(a)f(a),\ a\varphi(a)f(a),\ a^2\varphi(a)f(a), \cdots, a^{q-1}\varphi(a)f(a), \\ af(a),\ a^2f(a),\ a^3f(a), \cdots,\ a^{2n+1}f(a), \\ af^2(a),\ a^2f^2(a),\ a^3f^2(a), \cdots,\ a^{2n+1}f^2(a), \end{pmatrix}.$$

Any such linear combination associated with $f(a)$ can be written as

$$f(a) = \varphi(a) + A(a)\varphi(a)f(a) + B(a)af(a) + B(a)af^2(a),$$

where $A(a), B(a)$ are polynomials with $0 \leq deg(A(a)) \leq q-1$ and $0 \leq deg(B(a)) \leq 2n$. To satisfy the above equation, we have $A(a) = 0$, $B(a) = 0$ and $\varphi(a) = f(a)$, which contradicts that $deg(f(a))$ and $deg(\varphi(a))$ are distinct. Therefore, the defined (f, φ)-AGDDHE problem is one of the intractable GDDHE problems.

6.4 The Proposed Scheme

In this section, we describe the proposed scheme.

6.4.1 Construction

Setup$(1^\lambda, n)$. Given a security parameter λ and n the length of vectors, a bilinear group $\mathbb{BG} = (p, \mathbb{G}, \mathbb{G}_T, e)$ is constructed. The size of the group is determined by the security parameter λ. Two generators $g, h \in \mathbb{G}$ are randomly selected as well as a secret value $\alpha \in \mathbb{Z}_p$. As $\langle \vec{0}, \vec{x} \rangle = 0$ for any vector $\vec{x} \in \mathbb{Z}_p^n$, if $\vec{x} \in \mathbb{Z}_p^n$, we mean that $\vec{x}$ is chosen from $\mathbb{Z}_p^n \backslash \vec{0}$ in our construction. It chooses a cryptographic hash function $H : \{0,1\}^* \to \mathbb{Z}_p^*$. Then it samples $\vec{\beta} = (\beta_1, \beta_2, \cdots, \beta_n) \in \mathbb{Z}_p^n$, chooses random exponents $\gamma_1, \gamma_2, \eta_1, \eta_2 \in \mathbb{Z}_p$ such that $\gamma_1\eta_1 = \gamma_2\eta_2$ and sets the master public key mpk as

$$mpk = \left(\mathbb{BG}, n, H, h, g, g^{\eta_1\alpha}, g^{\gamma_2\eta_2} \left\{ g^{\eta_1\alpha\beta_i}, h^{\gamma_1\alpha^i}, h^{\beta_i} \right\}_{i=1}^{n} \right).$$

The master secret key msk consists of $(\alpha, \gamma_1, \gamma_2, \eta_1, \eta_2, \vec{\beta})$.

KeyGen$(mpk, msk, ID, \vec{y})$. Given the master key pair (mpk, msk), an identity $ID \in \{0,1\}^*$ and vector $\vec{y} = (y_1, y_2, \cdots, y_n) \in \mathbb{Z}_p^n$, this algorithm randomly chooses $k \in \mathbb{Z}_p$ and computes a secret key

$$sk_{ID,\vec{y}} = (K_1, K_2) = \left(g^{\eta_1 \cdot \frac{\langle \vec{\beta}, \vec{y} \rangle - k}{\alpha - H(ID)}},\ k \right).$$

Encrypt$(mpk, \mathbb{S}, \vec{x})$. Assume for notational simplicity that $\mathbb{S} = \{ID_j\}_{j=1}^s$. On input the master public key mpk and message $\vec{x} = (x_1, x_2, \cdots, x_n) \in \mathbb{Z}_p^n$, it chooses a random $r \in \mathbb{Z}_p$ and creates the ciphertext as:

$$C_0 = h^{r \cdot \gamma_1 \cdot \prod_{i=1}^s (\alpha - H(ID_i))},\quad C_1 = e(g^{\gamma_2 \eta_2}, h)^r,\quad C_2 = (g^{\eta_1 \alpha})^r,$$

together with, for each $i = 1, 2, \cdots, n$:

$$C_{i,1} = \left(g^{\eta_1 \alpha \beta_i}\right)^{-r},\quad C_{i,2} = \left(e(g^{\gamma_2 \eta_2}, h)^{\beta_i}\right)^r \cdot e(g,h)^{x_i}.$$

The output ciphertext is $CT = \left(C_0,\ C_1,\ C_2,\ \{C_{i,1},\ C_{i,2}\}_{i=1}^n\right)$.

Decrypt$(mpk, CT, \mathbb{S}, sk_{ID_j,\vec{y}_j}, \vec{y}_j)$. On input the master public key mpk, the ciphertext CT, the identity set $\mathbb{S}$, an identity ID_j and the corresponding private key $sk_{ID_j,\vec{y}_j}$ of vector $\vec{y}_j = (y_{j,1}, y_{j,2}, \cdots, y_{j,n})$. Let $ID_j \in \mathbb{S}$. The user with identity ID_j first defines

$$p_{j,\mathbb{S}}(\alpha) = \frac{\gamma_1}{\alpha} \cdot \left(\prod_{i=1, i \neq j}^s (\alpha - H(ID_i)) + (-1)^s \prod_{i=1, i \neq j}^s H(ID_i) \right).$$

After that, the user computes

$$\begin{aligned}
A &= e\left(K_{1,ID_j,\vec{y}_j}, C_0\right) \cdot e\left(\prod_{i=1}^n C_{i,1}^{y_{j,i}}, h^{p_{j,\mathbb{S}}(\alpha)} \right) \\
&= e(g,h)^{-k \cdot r \cdot \gamma_1 \eta_1 \cdot \prod_{i=1, i \neq j}^s (\alpha - H(ID_i))} \cdot e(g,h)^{(-1)^{s+1} r \cdot \gamma_1 \eta_1 \cdot \langle \vec{\beta}, \vec{y}_j \rangle \prod_{i=1, i \neq j}^s H(ID_i)}, \\
B &= e\left((C_2)^{-K_{2,ID_j,\vec{y}_j}}, h^{p_{j,\mathbb{S}}(\alpha)} \right) \cdot C_1^{(-1)^{s+1} K_{2,ID_j,\vec{y}_j} \cdot \prod_{i=1, i \neq j}^s H(ID_i)} \\
&= e(g,h)^{k \cdot r \cdot \gamma_1 \eta_1 \cdot \prod_{i=1, i \neq j}^s (\alpha - H(ID_i))} \cdot e(g,h)^{(-1)^s k \cdot r \cdot \gamma_1 \eta_1 \cdot \prod_{i=1, i \neq j}^s H(ID_i)} \\
&\quad \cdot \left(e\left(g^{\gamma_2 \eta_2}, h\right)^r \right)^{(-1)^{s+1} k \cdot \prod_{i=1, i \neq j}^s H(ID_i)} \\
&= e\left(g,h\right)^{k \cdot r \cdot \gamma_1 \eta_1 \cdot \prod_{i=1, i \neq j}^s (\alpha - H(ID_i))}.
\end{aligned}$$

It hen obtains

$$D = (A \cdot B)^{\frac{(-1)^{s+1}}{\prod_{i=1, i\neq j}^{s} H(ID_i)}} = e(g,h)^{r\cdot\gamma_1\eta_1\cdot\langle\vec{\beta},\vec{y}_j\rangle}.$$

Finally, the user computes

$$\prod_{i=1}^{n} C_{i,2}^{y_{j,i}} = e(g,h)^{r\cdot\gamma_1\eta_1\cdot\langle\vec{\beta},\vec{y}_j\rangle} \cdot e(g,h)^{\langle\vec{x},\vec{y}_j\rangle},$$

and learns the inner product by computing

$$\prod_{i=1}^{n} C_{i,2}^{y_{j,i}}/D = e(g,h)^{\langle\vec{x},\vec{y}_j\rangle}.$$

6.4.2 Discussion

The decryption algorithm requires to compute a discrete logarithm that cannot be avoided. We assume that the inner products will be contained in an interval $[0, L]$ with a polynomial upper bound L (The discrete logarithm can be computed in time $O\left(\sqrt{L}\right)$ by using the Pollard's kangaroo method [Pol00].). This assumption as pointed out in [ABCP15, ALS16] is reasonable for statistical applications since the results will be in a small space. There is no bound for private key issuing in our construction. For each message encryption, the only restriction is that the number of identities in $\mathbb{S}$ is less than n.

6.5 Security Analysis

In this section, we analysis the security of the proposed scheme under the defined security model. Now, we prove that the proposed IBBE-IP scheme is IND-sIDV-CPA secure assuming that the (f, φ)-AGDDHE problem is hard. Our proof strategy draws inspiration from [Gen06] and [Del07].

Theorem 6.2. *Let H be a random oracle and q be the total number of private key queries and oracle queries issued by the adversary. For any adversary $\mathcal{A}$ against the IND-sIDV-CPA security of our proposed identity-based broadcast inner product encryption scheme with $\mathsf{adv}^{\mathsf{IND\text{-}sIDV\text{-}CPA}}_{\mathcal{A},\mathsf{IBBE\text{-}IP}}(\lambda)$, the (f, φ)-AGDDHE problem can be solved by an algorithm $\mathcal{B}$ with advantage*

$$\mathsf{Adv}^{(\mathsf{f},\varphi)\text{-}\mathsf{AGDDHE}}_{\mathcal{B}}(\lambda) \geq \mathsf{adv}^{\mathsf{IND\text{-}sIDV\text{-}CPA}}_{\mathcal{A},\mathsf{IBBE\text{-}IP}}(\lambda).$$

Proof. Suppose that there exists an adversary $\mathcal{A}$ who has non-negligible advantage $\mathsf{adv}^{\mathsf{IND\text{-}sIDV\text{-}CPA}}_{\mathcal{A},\mathsf{IBBE\text{-}IP}}(\lambda)$ in breaking our proposed scheme. We can construct a simulator (algorithm) $\mathcal{B}$ that uses $\mathcal{A}$ to solve the (f, φ)-AGDDHE problem. Assuming that

both the adversary and the challenger are given as input an integer n and q the total number of private key queries and random oracle queries. Let $\mathcal{I}\left(f(x), \Lambda, a, b, c, \gamma, Z\right)$ be an (f, φ)-AGDDHE instance as the input of algorithm $\mathcal{B}$. We have $f(x)$ and φ are polynomials of respective degrees q and n. $\mathcal{B}$ works by interacting with $\mathcal{A}$ in the IND-sIDV-CPA game as follows.

Init: At the beginning of the game, $\mathcal{A}$ outputs a set of identities $\mathbb{S}^* = \{ID_1^*, ID_2^*, \cdots, ID_{s^*}^*\}$ and two distinct message vectors $\vec{x}_0 = (x_{0,1}, x_{0,2}, \cdots, x_{0,n})$ and $\vec{x}_1 = (x_{1,1}, x_{1,2}, \cdots, x_{1,n})$ that it wants to attack, where $s^* < n$.

Setup: $\mathcal{B}$ firstly picks $\theta_1, \theta_2, \cdots, \theta_n$ uniformly at random in $\mathbb{Z}_p$ and implicitly sets $\alpha = a, \gamma_1 = bc, \gamma_2 = c, \eta_1 = f(a), \eta_2 = bf(a)$ and

$$\vec{\beta} = (\beta_1, \beta_2, \cdots, \beta_n) = \big(\theta_1 f(a), \theta_2 f(a), \cdots, \theta_n f(a)\big).$$

As $f(x)$ is a random polynomial from the instance and $f(a) \neq 0$, we have β in this setting is random in $\mathbb{Z}_p$. We define

$$F_{\vec{v}}(x) = \theta_1 f(x) v_1 + \theta_2 f(x) v_2 + \cdots + \theta_n f(x) v_n,$$

where $\vec{v} = (v_1, v_2, \cdots, v_n) \in \mathbb{Z}_p^n$. Then $F_{\vec{v}}(a) = \left\langle \vec{\beta}, \vec{v} \right\rangle$ and $F_{\vec{v}}(x)$ is a function with an order at most q. Finally, $\mathcal{B}$ sets the master public key mpk as

$$\begin{aligned}
g &= g_0, \\
h &= h_0^{\prod_{i=q+s^*+1}^{q+n}(a-z_i)}, \text{ can be computed from line (1)}, \\
g^{\eta_1 \alpha} &= g_0^{af(a)}, \text{ from line (3)} \\
g^{\gamma_2 \eta_2} &= g_0^{bcf(a)}, \text{ from line (4)} \\
g^{\eta_1 \alpha \beta_j} &= g_0^{\theta_j a \cdot f^2(a)}, \quad j \in [1, n] \text{ can be computed from the line of (4)}, \\
h^{\beta_j} &= h_0^{\theta_j f(a) \cdot \prod_{i=q+s^*+1}^{q+n}(a-z_i)}, \quad j \in [1, n] \text{ can be computed from line (1)}, \\
h^{\gamma_1 \alpha^j} &= h_0^{bca^j \cdot \prod_{i=q+s^*+1}^{q+n}(a-z_i)}, \quad j \in [1, n] \text{ can be computed from line (2)}.
\end{aligned}$$

Here, we view H as a random oracle controlled by $\mathcal{B}$. Before the hash query, $\mathcal{B}$ picks $z_1, z_2, \cdots, z_q$ uniformly at random in $\mathbb{Z}_p$ until they are distinct from the elements in Λ.

Hash-Queries: At any time $\mathcal{A}$ can query the random oracle on any identity ID_i (at most $q - q_E$ times, with q_E the number of private key queries). To respond, $\mathcal{B}$ maintains a list $\mathcal{L}$ of tuples $(ID_i, \vec{y}_i, z_i, sk_{ID_i, \vec{y}_i})$ that contains at the beginning:

$$\{*, *, z_i, *\}_{i=1}^{q}, \quad \{ID_i^*, *, z_i^*, *\}_{i=1}^{s^*} = \{ID_i, *, z_i, *\}_{i=q+1}^{q+s^*},$$

where $*$ denotes an empty entry in $\mathcal{L}$. Upon receiving the query on ID_i, $\mathcal{B}$ will first check whether ID_i already appears in the list $\mathcal{L}$ and respond with the corresponding z_i if so. Otherwise, $\mathcal{B}$ sets $H(ID_i) = z_i$ and completes the list with $(ID_i, *, z_i, *)$.

Phase 1: In this phase, $\mathcal{A}$ can issue private key queries on $(ID_i, \vec{y}_i)$, $i \in [1, m]$, where m is decided by the adversary. If $ID_i \in \mathbb{S}^*$, we require $\langle \vec{y}_i, \vec{x}_0 \rangle = \langle \vec{y}_i, \vec{x}_1 \rangle$. To generate the private keys, $\mathcal{B}$ performs as follows:

1. If $\mathcal{A}$ has already issued a private key query on $(ID_i, \vec{y}_i)$, $\mathcal{B}$ will respond with the corresponding $sk_{ID_i,\vec{y}_i}$ in the list $\mathcal{L}$.

2. Else, if $\mathcal{A}$ has already issued a hash query on ID_i, then $\mathcal{B}$ will use the corresponding z_i to compute

$$sk_{ID_i,\vec{y}_i} = (K_{1,ID_i,\vec{y}_i}, K_{2,ID_i,\vec{y}_i}) = \left(g_0^{f(a)\frac{\left(F_{\vec{y}_i}(a) - F_{\vec{y}_i}(z_i)\right)}{a - z_i}}, F_{\vec{y}_i}(z_i) \right)$$

   and complete the list $\mathcal{L}$. $K_{1,ID_i,\vec{y}_i}$ is computable from the line (3). $\mathcal{B}$ responds with the corresponding $sk_{ID_i,\vec{y}_i}$ and updates the list $\mathcal{L}$.

3. Otherwise, $\mathcal{B}$ runs the **Hash-Queries** to get z_i, computes the corresponding $sk_{ID_i,\vec{y}_i}$ as step 2 and updates the list $\mathcal{L}$.

Challenge: Once $\mathcal{A}$ decides the phase 1 is over, $\mathcal{B}$ randomly picks a random bit $\mu \in \{0, 1\}$ and generates the challenge ciphertext CT^* for $\vec{x}_\mu$ as follows.

It first obtains the private keys for all identities in $\mathbb{S}^*$ from the list $\mathcal{L}$. If the private key for ID_i^* does not exist, it randomly picks a vector $\vec{y}_i^* \in \mathbb{Z}_p^n$ such that $\vec{y}_i^*$ is different from all vectors that have been queried private keys and runs the private key query in the phase 1 to obtain the corresponding private keys. Let the private key of $(ID_i^*, \vec{y}_i^*)$ is

$$sk_{ID_i^*,\vec{y}_i^*} = (K_{1,ID_i^*,\vec{y}_i^*}, K_{2,ID_i^*,\vec{y}_i^*}) = \left(g_0^{f(a)\frac{\left(F_{\vec{y}_i^*}(a) - F_{\vec{y}_i^*}(z_i^*)\right)}{a - z_i^*}}, F_{\vec{y}_i^*}(z_i^*) \right), \ i \in [1, s^*].$$

It then sets

$$C_0^* = h_0^{\gamma bc\varphi(a)}, \ C_2^* = g_0^{\gamma af(a)},$$

and computes

$$C_1^* = Z^{(-1)^{n-s^*-1} \prod_{i=q+s^*+1}^{q+n} z_i} \cdot e\left(g_0^{\gamma af(a)}, h_0^{q(a)}\right),$$

where

$$q(a) = \frac{bc}{a} \cdot \left(\prod\nolimits_{i=q+s^*+1}^{q+n} (a - z_i) + (-1)^{n-s^*} \prod\nolimits_{i=q+s^*+1}^{q+n} z_i \right).$$

For $i = 1, 2, \cdots, n$, it computes

$$C^*_{i,1} = g_0^{-\theta_i \gamma a f^2(a)}, \text{ which are computable from the line (4).}$$

Let $\vec{y}^*_j = \{y_{j,1}, y_{j,2}, \cdots, y_{j,n}\}$. In order to compute the ciphertext component $C^*_{i,2}$, $\mathcal{B}$ firstly uses each private key of the identity ID^*_j in $\mathbb{S}$ and the corresponding vector $\vec{y}^*_j$ to compute

$$\begin{aligned} A_j &= e\left(K_{1,ID^*_j,\vec{y}^*_j}, C^*_0\right) \cdot e\left(\prod_{i=1}^{n} {C^*_{i,1}}^{y_{j,i}}, h^{p_{j,\mathbb{S}}(a)}\right), \\ B_j &= e\left(\left(C^*_2\right)^{-K_{2,ID^*_j,\vec{y}^*_j}}, h^{p_{j,\mathbb{S}}(a)}\right) \cdot {C^*_1}^{(-1)^{s^*} K_{2,ID^*_j,\vec{y}^*_j} \cdot \prod_{i=1,i\neq j}^{s^*} H(ID^*_i)}, \end{aligned}$$

where

$$p_{j,\mathbb{S}}(a) = \frac{bc}{a} \cdot \left(\prod_{i=1,i\neq j}^{s^*} (a - H(ID^*_i)) + (-1)^{s^*} \prod_{i=1,i\neq j}^{s^*} H(ID^*_i)\right).$$

$h^{p_{j,\mathbb{S}}(a)}$ is computable from the line (2). Then it computes

$$D_j = (A_j \cdot B_j)^{\frac{(-1)^{s^*}}{\prod_{i=1,i\neq j}^{s^*} H(ID^*_i)}}, \quad e(g,h)^{\left\langle \vec{x}_\mu, \vec{y}^*_j \right\rangle}, \ j \in [1, s^*].$$

Finally, it can get the following equations

$$\begin{aligned} \prod_{i=1}^{n} {C^*_{i,2}}^{y_{1,i}} &= D_1 \cdot e(g,h)^{\left\langle \vec{x}_\mu, \vec{y}^*_1 \right\rangle}, \\ \prod_{i=1}^{n} {C^*_{i,2}}^{y_{2,i}} &= D_2 \cdot e(g,h)^{\left\langle \vec{x}_\mu, \vec{y}^*_2 \right\rangle}, \\ &\cdots \\ \prod_{i=1}^{n} {C^*_{i,2}}^{y_{s^*,i}} &= D_{s^*} \cdot e(g,h)^{\left\langle \vec{x}_\mu, \vec{y}^*_{s^*} \right\rangle}. \end{aligned}$$

We observe that it has n unknown elements $C^*_{i,2}$, $i \in [1, n]$, but only has $s^* < n$ equations. It randomly pick $(n - s^*)$ elements from $\mathbb{G}_T$ and sets them as $C^*_{s^*+1}, C^*_{s^*+2}, \cdots, C^*_n$ such that the remaining s^* equations have a solution for $C^*_{i,2}$, $i \in [1, s^*]$ via the method of elimination. Finally, $\mathcal{B}$ responds with the challenge ciphertext CT^* as

$$CT^* = \left(C^*_0,\ C^*_1,\ C^*_2,\ \left\{C^*_{i,1},\ C^*_{i,2}\right\}_{i=1}^{n}\right).$$

If $Z = e(g_0, h_0)^{\gamma bcf(a)}$, let $r = \gamma$, we have

$$
\begin{aligned}
C_0^* &= h^{r\cdot\gamma_1\cdot\prod_{i=1}^{s}(\alpha-H(ID_i^*))} \\
&= h_0^{\prod_{i=q+s^*+1}^{q+n}(a-z_i)\cdot r\cdot bc\cdot\prod_{i=q+1}^{q+s^*}(\alpha-z_i)} \\
&= h_0^{r\cdot bc\cdot\prod_{i=q+1}^{q+n}(\alpha-z_i)} \\
&= h_0^{\gamma bc\varphi(a)}, \\
C_1^* &= Z^{(-1)^{n-s^*-1}\prod_{i=q+s^*+1}^{q+n} z_i}\cdot e\left(g_0^{\gamma af(a)}, h_0^{q(a)}\right) \\
&= e(g_0,h_0)^{\gamma bcf(a)(-1)^{n-s^*-1}\prod_{i=q+s^*+1}^{q+n} z_i}\cdot e\left(g_0^{\gamma af(a)}, h_0^{q(a)}\right) \\
&= e(g_0,h_0)^{\gamma bcf(a)\prod_{i=q+s^*+1}^{q+n}(a-z_i)} \\
&= e\left(g_0^{bcf(a)},\ h_0^{\prod_{i=q+s^*+1}^{q+n}(a-z_i)}\right)^{\gamma} \\
&= e(g^{\gamma_2\eta_2},h)^r, \\
C_2^* &= g^{\eta_1\alpha r} = g_0^{\gamma af(a)}, \\
C_{i,1}^* &= g^{-r\eta_1\alpha\beta_i} = g_0^{-\theta_i\gamma af^2(a)}.
\end{aligned}
$$

Then, from the setting,

$$D_j = e(g,h)^{r\cdot\gamma_1\eta_1\left\langle\vec{\beta},\vec{y}_j^*\right\rangle}, j\in[1,s^*],$$

which is identical with the real scheme.

Phase 2: $\mathcal{A}$ continues to issue private key queries on $(ID_j,\vec{y}_j)$, $j\in[m+1,q_E]$ with the restriction that if $ID_j\in\mathbb{S}^*$, it require that $\langle\vec{y}_j,\vec{x}_0\rangle=\langle\vec{y}_j,\vec{x}_1\rangle$. $\mathcal{C}$ responds as in phase 1.

Guess: Finally, $\mathcal{A}$ outputs a guess $\mu'\in\{0,1\}$. If $\mu=\mu'$, $\mathcal{B}$ answers 1 as the solution to the given instance of the (f,φ)-AGDDHE problem, meaning that $Z=e(g_0,h_0)^{\gamma bcf(a)}$. Otherwise, $\mathcal{B}$ answers 0 which indicates that $Z\neq e(g_0,h_0)^{\gamma bcf(a)}$.

Next, we analyze the advantage of $\mathcal{B}$ to solve the hard problem. For notation simplicity, we use $\mathcal{I}$ to represent $\mathcal{I}(f(x),\Lambda,a,b,c,\gamma,Z)$.

$$
\begin{aligned}
\mathsf{Adv}_{\mathcal{B}}^{(\mathsf{f},\varphi)\text{-}\mathsf{AGDDHE}}(\lambda) &= \left|\Pr\left[\mathcal{B}(I)=1\middle|\mathsf{true}\right]-\Pr\left[\mathcal{B}(I)=1\middle|\mathsf{false}\right]\right| \\
&= \left|\Pr\left[\mu=\mu'\middle|\mathsf{true}\right]-\Pr\left[\mu=\mu'\middle|\mathsf{false}\right]\right|.
\end{aligned}
$$

From the above simulation, we have that when the event true occurs, from the point view of the adversary, the simulation is indistinguishable from the real scheme. Thus, $\Pr[\mu=\mu'|\mathsf{true}]=1/2+\mathsf{adv}_{\mathcal{A},\mathsf{IBBE\text{-}IP}}^{\mathsf{IND\text{-}sIDV\text{-}CPA}}(\lambda)$. If it falls in the false event, the view of $\mathcal{A}$ is independent of the bit μ. In this case, the probability $\Pr[\mu=\mu'|\mathsf{false}]=1/2$.

There is no abortion in our simulation. Finally, we obtain

$$\begin{aligned}\mathsf{Adv}_{\mathcal{B}}^{(\mathsf{f},\varphi)\text{-}\mathsf{AGDDHE}}(\lambda) &= \left| \Pr\left[\mu = \mu' \middle| \mathsf{true}\right] - \Pr\left[\mu = \mu' \middle| \mathsf{false}\right] \right| \\ &= 1/2 + \mathsf{Adv}_{\mathcal{A}}^{\mathsf{IND\text{-}sIDV\text{-}CPA}}(\lambda) - 1/2 \\ &= \mathsf{adv}_{\mathcal{A},\mathsf{IBBE\text{-}IP}}^{\mathsf{IND\text{-}sIDV\text{-}CPA}}(\lambda).\end{aligned}$$

□

Remark. In the simulation of the challenge ciphertext, we draw inspiration from Gentry IBE [Gen06]. We use the private keys of the identities in $\mathbb{S}^*$ to generate one component of the challenge ciphertext. To generate the challenge ciphertext, we randomly pick a vector (e.g. $\vec{y}_i^*$) for the challenge identity (e.g. ID_i^*) which has not been queried its private key and generate the corresponding private key. One might think that in phase 2, the adversary may query the private key of $(ID_j, \vec{y}_j)$ where $ID_j = ID_i^*$ but $\vec{y}_j \neq \vec{y}_i^*$. We stress that even this case happens, our simulation is still indistinguishable from the real scheme. In our construction, the decryption condition is that the user's identity must be in $\mathbb{S}$ and there is no any restriction about the vectors. The vector held by the user is decided by the user or the PKG. Therefore, the adversary cannot distinguish the simulation from the challenge ciphertext.

6.6 Conclusion

In this chapter, we introduced a notion of identity-based broadcast encryption for inner product (IBBE-IP) and presented a concrete construction. The IBBE-IP captures the merits of both IBBE and IPE. In the IBBE-IP, each user is associated with an identity and a vector which is selected by the user or the PKG depending on the application. During an encryption, the encryptor can determine who are permitted to learn the inner products of the encrypted message and the vector associated with the decryption keys without leaking the message vector. The proposed IBBE-IP scheme has constant-size private keys and supports unbounded private key queries. The security of our proposed scheme is based on the hardness of one specific q-type problem and the scheme is proved secure in the IND-sIDV-CPA security model with random oracles.

Chapter 7

Conclusion and Future Work

In this chapter, we summarize the work presented in this thesis and put forward several directions for further research.

7.1 Conclusion

Broadcast encryption plays a signification role in modern cryptography. It not only efficiently protects the data (message) confidentiality but also allows the encryptor to decide who can decrypt the encrypted data. These merits make broadcast encryption popular in the real-life applications, such as in Pay-TV. Identity-based broadcast encryption (IBBE) has shown its advantage in key management, where only those users whose identities are selected in the computation of the ciphertext can decrypt the encrypted message. IBBE has been studied extensively.

User revocation is another important research area in the broadcast encryption system. If any of these users who can obtain the message is compromised, we should revoke them such that they cannot decrypt the encrypted message. Most of the revocation schemes in the literature focus on preventing the users from retrieving the future broadcast message. The first work revokes the users from the IBBE is studied in [SCG+16], which allows a third party to remove some of the receivers from the identity set stated in the original broadcast ciphertext without knowing the encrypted message. Recipient revocable IBBE has shown its expressive for practical applications. However, the work in [SCG+16] does not consider the receiver privacy which is indispensable in some scenarios. Based on this observation, we proposed several anonymous revocable IBBE schemes from bilinear groups to fill this gap.

In Chapter 3, we formalized the definition of anonymous revocable IBBE to capture the receiver privacy. We put forward the first revocable IBBE scheme with the receiver anonymity. The third party who performs the revocation learns nothing about the receiver identity. The security of the proposed scheme is proved in the random oracle under the hardness of Bilinear Diffie-Hellman problem. However the privacy of the revoked users still cannot be preserved, which might be a bottleneck for some applications. The identities of the revoked users might expose some in-

formation about other non-revoked receivers if some of the revoked users are the original receivers.

In Chapter 4, we addressed the limitation of the proposed scheme in Chapter 3 to achieve fully privacy-preserving. We described a fully privacy-preserving revocable IBBE scheme, where both the identity information of the receivers and the revoked users are hidden. The ciphertext size is linear in the number of the receivers stated in the original ciphertext. We derived the security of the proposed scheme under the hardness of Bilinear Diffie-Hellman problem in the random oracle.

In Chapter 5, we considered a variant of revocable IBBE, namely, authorization. IBBE with authorization is capable for the situation where the receivers are decided by more than one parties. We presented a novel construction of fully privacy-preserving IBBE with authorization. The size of the final ciphertext depends on the size of the authorized identity set instead of the number of receivers stated in the original ciphertext. Its security is reduced to the hardness of Bilinear Diffie-Hellman problem in the random oracle. The authorization algorithm can be performed by different third parties with different authorized identity sets. Only the user belongs to all the authorized identity sets enable to retrieve the encrypted message.

In Chapter 6, we reviewed the inner product encryption (IPE) [ABCP15] and gave a comparison between IBBE and IPE. We noted that IPE is able to further protect the message as the decryption only reveals the inner product of the encrypted message and the vector associated with the decryption key, instead of the whole message. While IPE cannot control who are able to learn the inner product via decryption like in the IBBE. Based on this observation, we introduced a notion of identity-based broadcast encryption for inner products (IBBE-IP), which captures both the merits of IPE and IBBE. We then gave a concrete construction of IBBE-IP. Our scheme supports unbounded private key queries. The security of the proposed scheme is based on a q-type Diffie-Hellman exponent assumption in the generic group model.

7.2 Future Work

The revocable identity-based broadcast encryption schemes presented in this thesis can protect the user privacy well, but the security analysis is provided in the random oracle model. Although the random oracle model has been widely used in the research of cryptography, it is desirable if the cryptographic schemes could be proved secure without relying on random oracles. Thus, our future work mainly focuses on how to construct an anonymous revocable identity-based broadcast encryption scheme whose security can be derived without using the random oracle. Secondly, our proposed schemes only achieve selective security, we ask whether it is possible

to design a revocable IBBE scheme with anonymity which can achieve adaptive security.

In terms of the identity-based broadcast encryption for inner products, the proposed IBBE-IP scheme captures both the advantages of IBBE and IPE, but its security is based on one q-type assumption in the generic group model. The open question is how to construct an IBBE-IP scheme which can derive its security based on the standard assumptions.

Bibliography

[AAB+15] Shashank Agrawal, Shweta Agrawal, Saikrishna Badrinarayanan, Abishek Kumarasubramanian, Manoj Prabhakaran, and Amit Sahai. On the practical security of inner product functional encryption. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 777–798. Springer, 2015.

[ABCP15] Michel Abdalla, Florian Bourse, Angelo De Caro, and David Pointcheval. Simple functional encryption schemes for inner products. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 733–751. Springer, 2015.

[AD17] Kamalesh Acharya and Ratna Dutta. Adaptively secure recipient revocable broadcast encryption with constant size ciphertext. *IACR Cryptology ePrint Archive*, 2017:59, 2017.

[AGRW17] Michel Abdalla, Romain Gay, Mariana Raykova, and Hoeteck Wee. Multi-input inner-product functional encryption from pairings. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017*, volume 10210 of *LNCS*, pages 601–626, 2017.

[AI09] Nuttapong Attrapadung and Hideki Imai. Attribute-based encryption supporting direct/indirect revocation modes. In Matthew G. Parker, editor, *Cryptography and Coding 2009*, volume 5921 of *LNCS*, pages 278–300. Springer, 2009.

[AL10] Nuttapong Attrapadung and Benoît Libert. Functional encryption for inner product: Achieving constant-size ciphertexts with adaptive security or support for negation. In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010*, volume 6056 of *LNCS*, pages 384–402. Springer, 2010.

[ALdP11] Nuttapong Attrapadung, Benoît Libert, and Elie de Panafieu. Expressive key-policy attribute-based encryption with constant-size ciphertexts. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Anto-

nio Nicolosi, editors, *PKC 2011*, volume 6571 of *LNCS*, pages 90–108. Springer, 2011.

[ALS16] Shweta Agrawal, Benoît Libert, and Damien Stehlé. Fully secure functional encryption for inner products, from standard assumptions. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016*, volume 9816 of *LNCS*, pages 333–362. Springer, 2016.

[ARW16] Michel Abdalla, Mariana Raykova, and Hoeteck Wee. Multi-input inner-product functional encryption from pairings. *IACR Cryptology ePrint Archive*, 2016:425, 2016.

[BB04] Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238. Springer, 2004.

[BB11] Dan Boneh and Xavier Boyen. Efficient selective identity-based encryption without random oracles. *J. Cryptology*, 24(4):659–693, 2011.

[BBG05] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 440–456. Springer, 2005.

[BBL17] Fabrice Benhamouda, Florian Bourse, and Helger Lipmaa. CCA-secure inner-product functional encryption from projective hash functions. In Serge Fehr, editor, *PKC 2017*, volume 10175 of *LNCS*, pages 36–66. Springer, 2017.

[BBM00] Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 259–274. Springer, 2000.

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew K. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, 2004.

[BBW06] Adam Barth, Dan Boneh, and Brent Waters. Privacy in encrypted content distribution using private broadcast encryption. In Giovanni Di Crescenzo and Aviel D. Rubin, editors, *FC 2006*, volume 4107 of *LNCS*, pages 52–64. Springer, 2006.

[BF01] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, 2001.

[BGK08] Alexandra Boldyreva, Vipul Goyal, and Virendra Kumar. Identity-based encryption with efficient revocation. In Peng Ning, Paul F. Syverson, and Somesh Jha, editors, *Proceedings of the 2008 ACM Conference on Computer and Communications Security, CCS 2008*, pages 417–426. ACM, 2008.

[BGW05] Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 258–275. Springer, 2005.

[BJK15] Allison Bishop, Abhishek Jain, and Lucas Kowalczyk. Function-hiding inner product encryption. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015*, volume 9452 of *LNCS*, pages 470–491. Springer, 2015.

[Bon98] Dan Boneh. The decision Diffie-Hellman problem. In Joe Buhler, editor, *Algorithmic Number Theory, ANT 1998*, volume 1423 of *LNCS*, pages 48–63. Springer, 1998.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *CCS 1993*, pages 62–73. ACM, 1993.

[BRS13] Dan Boneh, Ananth Raghunathan, and Gil Segev. Function-private identity-based encryption: Hiding the function in functional encryption. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013*, volume 8043 of *LNCS*, pages 461–478. Springer, 2013.

[BS15] Zvika Brakerski and Gil Segev. Function-private functional encryption in the private-key setting. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015*, volume 9015 of *LNCS*, pages 306–324. Springer, 2015.

[BSS05] Joonsang Baek, Reihaneh Safavi-Naini, and Willy Susilo. Efficient multi-receiver identity-based encryption and its application to broadcast encryption. In Serge Vaudenay, editor, *PKC 2005*, volume 3386 of *LNCS*, pages 380–397. Springer, 2005.

[BSW06] Dan Boneh, Amit Sahai, and Brent Waters. Fully collusion resistant traitor tracing with short ciphertexts and private keys. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 573–592. Springer, 2006.

[BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 253–273. Springer, 2011.

[BW06] Dan Boneh and Brent Waters. A fully collusion resistant broadcast, trace, and revoke system. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 211–220. ACM, 2006.

[BWZ14] Dan Boneh, Brent Waters, and Mark Zhandry. Low overhead broadcast encryption from multilinear maps. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014*, volume 8616 of *LNCS*, pages 206–223. Springer, 2014.

[Chi12] Hung-Yu Chien. Improved anonymous multi-receiver identity-based encryption. *Comput. J.*, 55(4):439–446, 2012.

[CHKP10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 523–552. Springer, 2010.

[Coc01] Clifford Cocks. An identity-based encryption scheme based on quadratic residues. In Bahram Honary, editor, *Cryptography and Coding*, volume 2260 of *LNCS*, pages 360–363. Springer, 2001.

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *CRYPTO 1998*, volume 1462 of *LNCS*, pages 13–25. Springer, 1998.

[CW79] Larry Carter and Mark N. Wegman. Universal classes of hash functions. *J. Comput. Syst. Sci.*, 18(2):143–154, 1979.

[CWC+09] Cheng-Kang Chu, Jian Weng, Sherman S. M. Chow, Jianying Zhou, and Robert H. Deng. Conditional proxy broadcast re-encryption. In Colin Boyd and Juan Manuel González Nieto, editors, *ACISP 2009*, volume 5594 of *LNCS*, pages 327–342. Springer, 2009.

[DDM16] Pratish Datta, Ratna Dutta, and Sourav Mukhopadhyay. Functional encryption for inner product with full function privacy. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *PKC 2016*, volume 9614 of *LNCS*, pages 164–195. Springer, 2016.

[Del07] Cécile Delerablée. Identity-based broadcast encryption with constant size ciphertexts and private keys. In Kaoru Kurosawa, editor, *ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 200–215. Springer, 2007.

[DF03a] Yevgeniy Dodis and Nelly Fazio. Public key broadcast encryption for stateless receivers. In Joan Feigenbaum, editor, *Security and Privacy in Digital Rights Management Workshop*, volume 2696 of *LNCS*, pages 61–80. Springer, 2003.

[DF03b] Yevgeniy Dodis and Nelly Fazio. Public key trace and revoke scheme secure against adaptive chosen ciphertext attack. In Yvo Desmedt, editor, *PKC 2003*, volume 2567 of *LNCS*, pages 100–115. Springer, 2003.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Trans. Information Theory*, 22(6):644–654, 1976.

[DPP07] Cécile Delerablée, Pascal Paillier, and David Pointcheval. Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys. In Tsuyoshi Takagi, Tatsuaki Okamoto, Eiji Okamoto, and Takeshi Okamoto, editors, *Pairing 2007*, volume 4575 of *LNCS*, pages 39–59. Springer, 2007.

[FHH10] Chun-I Fan, Ling-Ying Huang, and Pei-Hsiu Ho. Anonymous multireceiver identity-based encryption. *IEEE Trans. Computers*, 59(9):1239–1249, 2010.

[FN94] Amos Fiat and Moni Naor. Broadcast encryption. In Douglas R. Stinson, editor, *CRYPTO 1993*, volume 773 of *LNCS*, pages 480–491. Springer, 1994.

[FNP14] Nelly Fazio, Antonio Nicolosi, and Irippuge Milinda Perera. Broadcast steganography. In Josh Benaloh, editor, *CT-RSA 2014*, volume 8366 of *LNCS*, pages 64–84. Springer, 2014.

[FP12] Nelly Fazio and Irippuge Milinda Perera. Outsider-anonymous broadcast encryption with sublinear ciphertexts. In Marc Fischlin, Johannes A. Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 225–242. Springer, 2012.

[Gam84] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and David Chaum, editors, *CRYPTO 1984*, volume 196 of *LNCS*, pages 10–18. Springer, 1984.

[Gen06] Craig Gentry. Practical identity-based encryption without random oracles. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 445–464. Springer, 2006.

[GGG+14] Shafi Goldwasser, S. Dov Gordon, Vipul Goyal, Abhishek Jain, Jonathan Katz, Feng-Hao Liu, Amit Sahai, Elaine Shi, and Hong-Sheng Zhou. Multi-input functional encryption. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 578–602. Springer, 2014.

[GH09] Craig Gentry and Shai Halevi. Hierarchical identity based encryption with polynomially many levels. In Omer Reingold, editor, *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009*, volume 5444 of *LNCS*, pages 437–456. Springer, 2009.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Cynthia Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 197–206. ACM, 2008.

[GS02] Craig Gentry and Alice Silverberg. Hierarchical ID-based cryptography. In Yuliang Zheng, editor, *ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 548–566. Springer, 2002.

[GST04] Michael T. Goodrich, Jonathan Z. Sun, and Roberto Tamassia. Efficient tree-based revocation in groups of low-state devices. In Matthew K. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 511–527. Springer, 2004.

[GW09] Craig Gentry and Brent Waters. Adaptive security in broadcast encryption systems (with short ciphertexts). In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 171–188. Springer, 2009.

[HL02] Jeremy Horwitz and Ben Lynn. Toward hierarchical identity-based encryption. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 466–481. Springer, 2002.

[HLR10] Javier Herranz, Fabien Laguillaumie, and Carla Ràfols. Constant size ciphertexts in threshold attribute-based encryption. In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010*, volume 6056 of *LNCS*, pages 19–34. Springer, 2010.

[HS02] Dani Halevy and Adi Shamir. The LSD broadcast encryption scheme. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 47–60. Springer, 2002.

[KD98] Kaoru Kurosawa and Yvo Desmedt. Optimum traitor tracing and asymmetric schemes. In Kaisa Nyberg, editor, *EUROCRYPT 1998*, volume 1403 of *LNCS*, pages 145–157. Springer, 1998.

[KLM+16] Sam Kim, Kevin Lewi, Avradip Mandal, Hart William Montgomery, Arnab Roy, and David J. Wu. Function-hiding inner product encryption is practical. *IACR Cryptology ePrint Archive*, 2016:440, 2016.

[KS13] Aggelos Kiayias and Katerina Samari. Lower bounds for private broadcast encryption. In Matthias Kirchner and Dipak Ghosal, editors, *IH 2012*, volume 7692 of *LNCS*, pages 176–190. Springer, 2013.

[Kur02] Kaoru Kurosawa. Multi-recipient public-key encryption with shortened ciphertext. In David Naccache and Pascal Paillier, editors, *PKC 2002*, volume 2274 of *Lecture Notes in Computer Science*, pages 48–63. Springer, 2002.

[LCL+13] Kwangsu Lee, Seung Geol Choi, Dong Hoon Lee, Jong Hwan Park, and Moti Yung. Self-updatable encryption: Time constrained access control with hidden attributes and better efficiency. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013*, volume 8269 of *LNCS*, pages 235–254. Springer, 2013.

[LKLP14] Kwangsu Lee, Woo Kwon Koo, Dong Hoon Lee, and Jong Hwan Park. Public-key revocation and tracing schemes with subset difference methods revisited. In Miroslaw Kutylowski and Jaideep Vaidya, editors, *ESORICS 2014*, volume 8713 of *LNCS*, pages 1–18. Springer, 2014.

[LL16] Kwangsu Lee and Dong Hoon Lee. Two-input functional encryption for inner products from bilinear maps. *IACR Cryptology ePrint Archive*, 2016:432, 2016.

[LMG+16] Jianchang Lai, Yi Mu, Fuchun Guo, Willy Susilo, and Rongmao Chen. Anonymous identity-based broadcast encryption with revocation for file

sharing. In Joseph K. Liu and Ron Steinfeld, editors, *ACISP2016*, volume 9723 of *LNCS*, pages 223–239. Springer, 2016.

[LMG+17] Jianchang Lai, Yi Mu, Fuchun Guo, Willy Susilo, and Rongmao Chen. Fully privacy-preserving and revocable ID-based broadcast encryption for data access control in smart city. *Personal and Ubiquitous Computing*, 21(5):855–868, 2017.

[LMGC17] Jianchang Lai, Yi Mu, Fuchun Guo, and Rongmao Chen. Fully privacy-preserving ID-based broadcast encryption with authorization. *Comput. J.*, 60(12):1809–1821, 2017.

[LPQ12] Benoît Libert, Kenneth G. Paterson, and Elizabeth A. Quaglia. Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model. In Marc Fischlin, Johannes A. Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 206–224. Springer, 2012.

[LSW10] Allison B. Lewko, Amit Sahai, and Brent Waters. Revocation systems with very small private keys. In *S&P 2010*, pages 273–285. IEEE Computer Society, 2010.

[LV09] Benoît Libert and Damien Vergnaud. Adaptive-ID secure revocable identity-based encryption. In Marc Fischlin, editor, *CT-RSA 2009*, volume 5473 of *LNCS*, pages 1–15. Springer, 2009.

[LW10] Allison B. Lewko and Brent Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In Daniele Micciancio, editor, *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010*, volume 5978 of *LNCS*, pages 455–479. Springer, 2010.

[NNL01] Dalit Naor, Moni Naor, and Jeffery Lotspiech. Revocation and tracing schemes for stateless receivers. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 41–62. Springer, 2001.

[NP00] Moni Naor and Benny Pinkas. Efficient trace and revoke schemes. In Yair Frankel, editor, *Financial Cryptography, 4th International Conference, FC 2000*, volume 1962 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2000.

[OT10] Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure functional encryption with general relations from the decisional linear assumption.

In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 191–208. Springer, 2010.

[OT12] Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure unbounded inner-product and attribute-based encryption. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 349–366. Springer, 2012.

[PLL15] Seunghwan Park, Kwangsu Lee, and Dong Hoon Lee. New constructions of revocable identity-based encryption from multilinear maps. *IEEE Trans. Information Forensics and Security*, 10(8):1564–1577, 2015.

[Pol00] John M. Pollard. Kangaroos, Monopoly and Discrete Logarithms. *J. Cryptology*, 13(4):437–447, 2000.

[PPSS12] Duong Hieu Phan, David Pointcheval, Siamak Fayyaz Shahandashti, and Mario Strefler. Adaptive CCA broadcast encryption with constant-size secret keys and ciphertexts. In Willy Susilo, Yi Mu, and Jennifer Seberry, editors, *ACISP 2012*, volume 7372 of *LNCS*, pages 308–321. Springer, 2012.

[PPT13] Duong Hieu Phan, David Pointcheval, and Viet Cuong Trinh. Multi-channel broadcast encryption. In Kefei Chen, Qi Xie, Weidong Qiu, Ninghui Li, and Wen-Guey Tzeng, editors, *ASIA CCS 2013*, pages 277–286. ACM, 2013.

[Rab80] Michael O. Rabin. Probabilistic algorithms in finite fields. *SIAM J. Comput.*, 9(2):273–280, 1980.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[SCG+16] Willy Susilo, Rongmao Chen, Fuchun Guo, Guomin Yang, Yi Mu, and Yang-Wai Chow. Recipient revocable identity-based broadcast encryption: How to revoke some recipients in IBBE without knowledge of the plaintext. In Xiaofeng Chen, XiaoFeng Wang, and Xinyi Huang, editors, *AsiaCCS 2016*, pages 201–210. ACM, 2016.

[SE13a] Jae Hong Seo and Keita Emura. Efficient delegation of key generation and revocation functionalities in identity-based encryption. In Ed Dawson, editor, *CT-RSA 2013*, volume 7779 of *LNCS*, pages 343–358. Springer, 2013.

[SE13b] Jae Hong Seo and Keita Emura. Revocable identity-based encryption revisited: Security model and construction. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*, pages 216–234. Springer, 2013.

[SF07] Ryuichi Sakai and Jun Furukawa. Identity-based broadcast encryption. *IACR Cryptology ePrint Archive*, 2007:217, 2007.

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *CRYPTO 1984*, volume 196 of *LNCS*, pages 47–53. Springer, 1984.

[Sho97] Victor Shoup. Lower bounds for Discrete Logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT 1997*, volume 1233 of *LNCS*, pages 256–266. Springer, 1997.

[SSW09] Emily Shen, Elaine Shi, and Brent Waters. Predicate privacy in encryption systems. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 457–473. Springer, 2009.

[SW08] Elaine Shi and Brent Waters. Delegating capabilities in predicate encryption systems. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008*, volume 5126 of *LNCS*, pages 560–578. Springer, 2008.

[Wat05] Brent Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127. Springer, 2005.

[Wat09] Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 619–636. Springer, 2009.

[XJW+16] Peng Xu, Tengfei Jiao, Qianhong Wu, Wei Wang, and Hai Jin. Conditional identity-based broadcast proxy re-encryption and its application to cloud email. *IEEE Trans. Computers*, 65(1):66–79, 2016.