Cloud-assisted Multi-Channel Data Broadcasting

Dissertation submitted in partial fulfillment of the
requirements for the degree of

Master of Technology
in
Cryptology and Security

by

Debabrata Maji
Roll Number: CrS2307

July 2025

DECLARATION

I, **Debabrata Maji** (Roll No: **CrS2307**), hereby declare that this report entitled **"Cloud-assisted Multi-Channel Data Broadcasting with Optimal Decryption Costs and Storage for IoT"**, submitted to *Indian Statistical Institute, Kolkata* towards the fulfilment of the requirements for the degree of **Master of Technology** in *CSRU*, is an original work carried out by me under the supervision of **Captain Manish Khanna**, **Mriganka Mandal**, and **Lt. Cdr. Keval Krishan**, and has not formed the basis for the award of any degree or diploma in this or any other institution or university.

I have sincerely tried to uphold academic ethics and honesty. Whenever a piece of external information, statement, or result is used, it has been duly acknowledged and cited.

Kolkata – 700108

Debabrata Maji

July 23, 2025

Certificate

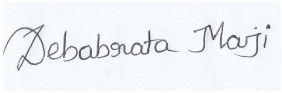
This is to certify that the work contained in this project report entitled **"Cloud-assisted Multi-Channel Data Broadcasting with Optimal Decryption Costs and Storage for IoT"** submitted by **Debabrata Maji (Roll No. CrS2307)** to the **Indian Statistical Institute, Kolkata** towards the fulfilment of the requirements for the degree of **Master of Technology in CSRU** has been carried out by him under my supervision and that it has not been submitted elsewhere for the award of any degree.

Dr. Mriganka Mandal
Assistant Professor
Cryptology and Security Research Unit
Indian Statistical Institute
Kolkata–700108, INDIA

ACKNOWLEDGEMENT

I thank everyone who has assisted me in seeing this project through to its completion. I would like to first express my profound gratitude and deepest regards to my Guides, **WESEE** and **ISI Kolkata**, and sincerely wish to acknowledge their vision, guidance, valuable feedback, and constant support throughout this project.

I am indebted to my friends for their steadfast encouragement and time. I am lastly grateful to the **Indian Statistical Institute, Kolkata** for providing the necessary resources and facilities to complete this project to the best of my ability.



WESEE,

New Delhi – 110066

Debabrata Maji

July 23, 2025

Abstract

The convergence of **I**nternet **o**f **T**hings (IoT) and cloud computing has transformed technology, impacting commerce, industrial production, data management, etc. **M**ulti-**C**hannel **B**roadcast **E**ncryption (MCBE), first introduced by Phan et al. (ASIACCS 2013), is a cryptographic encryption primitive used for both IoT and Cloud that permits a sender to efficiently and securely encrypt several messages for different groups of receivers. After thoroughly exploring the existing literature, we observe that none achieves the robust provable security within the standard model. This paper addresses this gap, aiming to achieve adaptive **IND**istinguishable under full-**ID**entity **C**hosen-**C**iphertext **A**ttack (IND-ID-CCA) security by constructing an efficient identity-based MCBE without the **R**andom **O**racle **M**odel (ROM). Our construction not only attains communication bandwidth of $O(\mu)$-size but also maintains constant-size overhead storage without any security vulnerabilities. Here, μ represents the number of messages. This is the first protocol proven to be adaptive IND-ID-CCA secure under the standard **D**ecisional **B**ilinear **D**iffie-**H**ellman Type-**3** (DBDH-3) assumption in public-key settings without any random oracles. Moreover, practical implementation data reveals an optimal decryption algorithm, taking a mere 0.0048 seconds on IoT devices, demonstrating real-world applicability. Furthermore, our proposed design is highly efficient as opposed to the other existing works, as shown by implemental and graphical data.

Keywords: Multi-Channel Broadcast Encryption, Identity-Based Encryption, Type-3 Bilinear Map, Adaptive IND-CCA Security

Contents

Chapter 1

Introduction

The rise of the Internet of Things (IoT) [1]–[7] has brought about a network of interconnected gadgets/devices, enabling seamless data exchange between the physical world and computer systems. Smart IoT-enabled devices have revolutionized daily life, presenting opportunities for innovation, safety, comfort, convenience, and efficiency like never before. The applications of IoT are pervasive, finding utility in various industries, including wearable technology, smart homes, healthcare, and more. Modern IoT applications have harnessed the power of Machine Learning (ML) and Artificial Intelligence (AI) to imbue gadgets with intelligence. The incorporation of ML and AI capabilities enhances the functionality and adaptability of IoT devices, empowering them to make intelligent decisions and respond effectively to dynamic situations.

On the other hand, cloud computing [8]–[15] is currently a trending technology in the field of cryptography. Due to the limited storage overhead and low computational capability of IoT-enabled multimedia devices, we prefer to store all kinds of information in cloud servers. The cloud data broadcasting protocol helps an encryptor efficiently and securely outsource encoded information, also known as ciphertext, to a group of receivers through a cloud computing infrastructure so that only authorized users can decrypt it to retrieve the original information. Therefore, a protocol that supports secure and efficient data sharing under cloud computing is needed.

The cryptographic primitive **B**roadcast **E**ncryption (BE) provides an efficient and secure method to share digital contents to a group of subscribers through a public broadcast medium in such a way that only legitimate users can recover the original broadcast contents. In contrast, the illegitimate users get nothing about the broadcast contents even if they collude. The concept of BE was first formally proposed by Fiat et al. [16]. Later, BE has been intensely and broadly studied [15], [17]–[26] as one of the leading primitives for various real-life application scenarios ranging from the pay-TV systems, video conference, distance learning, radio broadcast, distribution of copyrighted materials, etc., to the encrypted email system and many more. An **I**dentity-**B**ased **B**roadcast **E**ncryption (IBBE) [27] is an advanced variant of BE in which the **P**ublic-**K**ey **I**nfrastructure (PKI)

is not required [27]–[29]. The IBBE system can accommodate unbounded (or arbitrary) number of users, where the user's public key can be generated utilizing only its unique identity (e.g., user's IP address, employment ID, phone/email address, etc.).

Unfortunately, the previous BE schemes only deal with one single message and one single subscribed group at a time. If a broadcaster wants to transmit many different messages to different groups of users at a time, then the broadcaster needs to avail the underlying BE scheme repeatedly, which requires huge computation costs and communication bandwidth. To resolve the prior mentioned inefficiency, Phan et al. [30] have designed the first efficient variant of BE, named as **M**ulti-**C**hannel **B**roadcast **E**ncryption (MCBE), that transmits several encrypted messages to different groups at a time. More precisely, MCBE protocol is designed as a multi-message and multi-receiver data encryption system where a data sender generates a single ciphertext header for multiple subscriber groups. Each subscribed user in a particular group can retrieve their group key using their unique secret key associated with that specific group. In MCBE, each group has its unique group key, and all information is stored in a single ciphertext header in an encoded form. Using their secret key and the ciphertext header, a subscribed user can retrieve their group key and decrypt messages sent for their specific group. In contrast, all remaining users cannot access messages intended for different groups. Interestingly, an attacker can not get any information about the encryption data, regardless of any plan they perform. We can classify the MCBE schemes into two categories: Private-key MCBE [30], [31] and Public-key MCBE [32]–[34]. In the private-key MCBE, the broadcaster must have access to the master secret-key of the system, which may weaken the security. The broadcaster itself performs both setup and key extraction algorithms. In other words, the **P**rivate-**K**ey **G**eneration **C**enter (PKGC) and broadcaster play the same role in the private-key setting. On the other hand, the broadcaster and PKGC perform different roles in the public-key settings. In such a setting, PKGC executes both setup and key extraction algorithms, whereas the broadcaster only works as an encryptor. This setting reduces the burden of the broadcaster, which is desirable for any PKI framework.

Over the last several years, inadequate progress has been made in developing secure and efficient MCBE schemes [30]–[34], which are only able to achieve multiple trade-offs amidst performance, security, and underlying complexity assumptions. There exist only five MCBE schemes, of which the construction of [33] is in the adaptive security model, and the others [30]–[32], [34] are in the selective security model. However, we require further improvements of MCBE that simultaneously: (i) enjoys constant-size storage overhead (which are highly crucial for the IoT-enabled environments), (ii) effectuates adaptive **IND**istinguishable under full-**ID**entity **C**hosen-**C**iphertext **A**ttack (IND-ID-CCA)) security without the **R**andom **O**racle **M**odel (ROM), (iii) relies on the non-q-type security assumptions in the standard security model, (iv) accommodates an unbounded (or arbi-

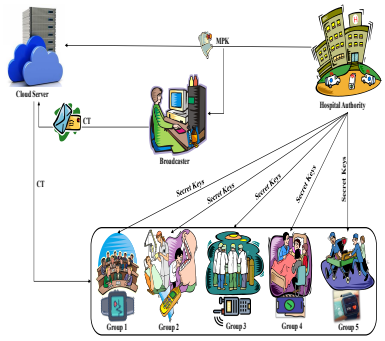
trary) number of users, (v) can be built over the advanced asymmetric prime-order Type-3 bilinear systems, and (vi) relies on public-key settings. In any MCBE framework, the prior mentioned six properties are highly desirable from both security and efficiency viewpoints and highly challenging to accomplish together. Furthermore, the properties (i), (ii), (iv), (v), and (vi) are required for many practical application scenarios, such as medical healthcare records systems, social data-sending applications, online social business frameworks, etc. Also, the properties (ii), (iii), and (iv) are now recommended cryptographic requirements about speed and performance, practicability, strongest security guarantees under the practical and the physical attack models, and least hardness assumptions [35]–[37]. There are now only three protocols on MCBE [32]–[34] that achieve only two vital properties (i) and (vi). However, we emphasize that, before our protocol, it was unexplained how to achieve all six properties together while accomplishing the identity-based framework; admittedly, this is comprehensively viewed as one of the major open problems in the MCBE framework.

1.1 Motivating application scenario

In this practical real-world scenario depicted in Fig 1.1, we have a healthcare system with various groups of specialized doctors, including orthopedics, gynecologists, dentists, general physicians, and oncologists. To maintain efficient access control and privacy, the hospital authority categorizes these doctors into distinct groups: Group 1 for general physicians (e.g., PH_1, PH_2, ...), Group 2 for dentists (e.g., DE_1, DE_2, ...), Group 3 for gynecologists (e.g., GY_1, GY_2, ...), Group 4 for orthopedics (e.g., OR_1, OR_2, ...), Group 5 for oncologists (e.g., ON_1, ON_2, ...), and so on. In our framework, the hospital authority acts as a PKGC (Private Key Generation Center) and provides secret keys to each doctor in their respective group. For example, the secret key for PH_1 is SK_{PH_1}, for DE_1 is SK_{DE_1}, and so forth. Since medical patient information is sensitive data, the hospital authority refrains from directly accessing data for all doctors. To facilitate secure communication and distribution of patient information, the broadcaster intends to send specific patient data to the relevant doctors within their respective groups. For instance, information about patients in Group 1 (i.e., M_1) will only be sent to legitimate doctors in Group 1 (e.g., PH_1, PH_2, PH_6). Similarly, information about patients in Group 2 (i.e., M_2) will exclusively reach legitimate doctors in Group 2 (e.g., DE_1, DE_2) and so on.

To achieve this, the broadcaster encrypts all messages ($M_1, M_2,$, etc.) for the authorized group of doctors, represented by $\mathcal{S} = \mathsf{S}_1 \cup \mathsf{S}_2 \cup \mathsf{S}_3$... and generates the ciphertext CT. This ciphertext is then uploaded to a cloud server for accessibility. Doctors in the hospital can access the ciphertext CT from the cloud server. However, only legitimate doctors equipped with IoT devices and the corresponding secret keys can decrypt the ci-

Figure 1.1: Application of IB-MCBE in e-healthcare system



phertext CT and obtain the appropriate patient information. In this scenario, employing our MCBE scheme proves highly advantageous. The MCBE scheme effectively reduces communication bandwidth, storage overhead, and computation costs. By centrally encrypting the data for each group, duplicate information storage is eliminated, and data distribution becomes more efficient. Furthermore, access control is enforced, ensuring that only authorized doctors within a specific group can decrypt and view relevant patient information, safeguarding patient privacy and confidentiality.

1.2 Our contributions

Motivated by the aforementioned practical application scenarios, we design a multi-message and multi-receiver data transmission system for both IoTs and cloud environments with the following interesting features.

- As compared to all the existing MCBE protocols [31]–[34], our proposed design is the first to simultaneously obtain constant-size overhead storage, and $O(\mu)$-size communication bandwidth, which is so far a plausible achievement. Moreover, our scheme can adequately support an unbounded (or arbitrary) number of users without expanding the storage overhead and rupturing strong security guarantees (due to skillfully incorporating the identity-based encryption framework).
- All the existing MCBE schemes [31]–[34] are only able to achieve either selective or adaptive IND-ID-CPA security, which is comparatively a weaker notion of the security. In contrast, our proposed protocol is the *first* to achieve the adaptive IND-ID-CCA security, which is one of the major contributions of this paper. More positively, we are able to achieve the same level of security without relying on the non-standard q-type security assumption and without random oracles.
- In a private-key setting, the broadcaster, who has access to the master secret key of the system, if somehow gets compromised, then the whole system may collapse.

Therefore, the schemes in a public-key setting (ours and [32]–[34]) are more secure compared to the systems [31] that are in a private-key setting. More positively, as compared to the symmetric Type-1 bilinear maps, the asymmetric Type-3 bilinear maps are efficient and more secure [38].

- Existing MCBE schemes [31], [32], [34] have high encryption time as compared to ours. In particular, for the Type-1 map-based schemes [31]–[34], the encryption time significantly increases due to the pairing operation within Type-1 groups being nearly two times slower, while multiplication takes four times slower when compared to Type-3 groups [38]. The significantly low encryption time points to meaningful gains in computation costs and makes our protocol even more practical. We also note that our decryption time is comparatively fast due to the use of a constant number of pairings, multiplication, and inversion operations.
- To validate the practicality and feasibility of our proposed design, we conducted the implementation of the Setup, Encrypt, and Decrypt algorithms for our protocol, comparing it with existing approaches [32]–[34]. Leveraging the PBC library, we executed the implementation on both a Dell Laptop and an IoT device, considering different system user scenarios. The results of our implementation showcase highly promising outcomes, affirming the strength of our protocol. Notably, we observed constant and impressive figures for both decryption time and storage overhead. The decryption algorithm exhibited exceptional efficiency, taking a mere 0.0048 seconds, while the storage overhead remained consistently low at 0.1440 kB (up to four decimal places). These implementation data affirm the viability of integrating our proposed protocol seamlessly into the IoT framework.

1.3 Related works

In 2013, Phan et al. [30] first proposed the concept of MCBE, which was based on the construction of Boneh et al. [17]. Their construction was in the private-key settings and has achieved selective IND-ID-CCA security under the non-standard q-**D**ecisional **B**ilinear **D**iffie **H**ellman **E**xponent (q-DBDHE) assumption. Later, their work has been improved by Zhao et al. [31] that has achieved slightly improved parameter sizes and decryption costs. Moreover, their work has been employed to construct an efficient dynamically privileged BE system. In 2020, Le et al. [34] has constructed another MCBE protocol in the public-key setting that supports faster decryption. Their scheme is proven to be selective IND-CPA secure under the GDDHE assumptions in the random oracle model. In 2018, Acharya et al. [32] have presented two MCBE schemes. Their first construction has public-parameters size linear to the maximum number of users that the system can support, whereas the secret-key size is constant. The scheme has achieved semi-static

IND-CPA security under the nonstandard DBDHE-sum assumption without any random oracles. In the second construction, they have applied the complete subtree method of Naor et al. [39] to partition the authorized users and also shown an additional features in MCBE, called the *outsider-anonymity*. The work has a public parameter size linear to the maximum number of users, whereas the user's secret-key grows linearly with the height of its corresponding complete sub-tree. This work is proven to be selective IND-CPA secure under the m-sq-DBDHE assumption. Later, Acharya et al. [33] proposed another two secure and efficient MCBE schemes in the public-key setting. Their first construction is constructed using the work of Boneh et al. [17] and has achieved the selective IND-CPA security under the non-standard q-DBDHE without any random oracles. Their second scheme, which supports both dynamic and inclusive-exclusive properties, has achieved the adaptive IND-CPA security under the modified DBDHE (mDBDHE) assumption without any random oracles. Recently, Kim et al. [40] proposed a potential alternative of MCBE, Efficient Anonymous Multi-group Broadcast Encryption (AMGBE), that achieves IND-CPA security under non-standard complexity assumptions ($\mathcal{P}$-DBDH & XDH) in the standard model. However, we emphasize that all the aforementioned MCBEs are unable to achieve the robust notion of the provable security, i.e., the adaptive IND-ID-CCA security, without the non-standard q-type security assumption and the random oracles.

Organization. The paper is arranged as follows. In the next Section **??**, relevant cryptographic backgrounds are addressed, and also a concise definition and security framework of our IB-MCBE are shown. The proposed main construction and its security analysis are given in Section 3. The comparative analysis of our schemes with the existing MCBE schemes are exhibited in Section 5, and the implementation details are presented in Section 6. Lastly, the paper concludes in Section 7.

Chapter 2

Preliminaries

This chapter presents the fundamental concepts required to understand the proposed IB-MCBE scheme. We introduce Type-3 bilinear maps, the Decisional Bilinear Diffie-Hellman (DBDH-3) assumption, and the basics of Identity-Based Encryption (IBE). We also define the system model and security framework for the IB-MCBE protocol under the IND-ID-CCA security notion.

Symbol	**Description**
λ	Security parameter
$\perp$	Null string
ID_{x_i}	Unique identity for user x_i in group x
$\|M\|$	Length of message M
$[\mu]$	Set $\{1, 2, \ldots, \mu\}$
$x \leftarrow_R S$	x randomly chosen from set S
PPT	Probabilistic Polynomial Time
IND-ID-CCA	Indistinguishability under full-identity chosen-ciphertext attack

Table 2.1: Table-1: Notation used in This Thesis

2.1 Type-3 Bilinear Map

Definition 2.1.1 (Type-3 Bilinear Map)**.** *Consider three multiplicative cyclic groups* $\mathbb{G}_1$*,* $\mathbb{G}_2$ *and* $\mathbb{G}_T$ *having the properties:* $\mathbb{G}_1 \neq \mathbb{G}_2$ *with no efficiently computable isomorphism from* $\mathbb{G}_1$ *to* $\mathbb{G}_2$ *and all three groups have the same prime order* p*. Here,* p *represents a large prime such that its length is at least* 2^λ*. Let* g *represent a random element generating the first source group* $\mathbb{G}_1$*, and* $\widetilde{g}$ *be another random element generating the second source group* $\mathbb{G}_2$*. A function*

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$$

is called a Type-3 Bilinear Map (also designated as T3-map), if it satisfies the following three fundamental properties:

1. ***Bilinearity:*** *For all* $X \in_R \mathbb{G}_1$, $Y \in_R \mathbb{G}_2$ *and* $a, b \in_R \mathbb{Z}_p^*$, *it holds that*

$$e(X^a, Y^b) = e(X, Y)^{ab}.$$

2. ***Non-degeneracy:*** *There exist generators* $g \in_R \mathbb{G}_1$ *and* $\widetilde{g} \in_R \mathbb{G}_2$ *such that*

$$e(g, \widetilde{g}) \neq 1_{\mathbb{G}_T},$$

   i.e., the element $e(g, \widetilde{g})$ *generates the target group* $\mathbb{G}_T$.

3. ***Computability:*** *The pairing* $e(g, \widetilde{g})$ *can be computed efficiently.*

The tuple $\mathbb{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ *is called a prime order Type-3 bilinear map (or T3-map) system.*

2.2 Type-3 Bilinear Hard Problem

Our proposed protocol's security relies on the **D**ecisional **B**ilinear **D**iffie-**H**ellman Type-**3** (DBDH-3) hard problem, defined as follows:

Input:

$$Z = \left(\mathbb{G}, g, \widetilde{g}, \widetilde{g}^a, \widetilde{g}^b, g^b, g^c, \widetilde{g}^c\right), \quad a, b, c \in_R \mathbb{Z}_p^*, \quad \text{and} \quad U,$$

where either

$$U = e(g, \widetilde{g})^{abc} \quad \text{or} \quad U \in_R \mathbb{G}_T \text{ (random)}.$$

Output:

$$\begin{cases} 0 & \text{if } U = e(g, \widetilde{g})^{abc} \\ 1 & \text{otherwise} \end{cases}$$

Definition 2.2.1 (DBDH-3 Assumption)**.** *The DBDH-3 assumption holds if for any probabilistic polynomial-time (PPT) adversary* $\mathcal{A}$, *the advantage*

$$Adv_{\mathcal{A}}^{DBDH\text{-}3}(1^\lambda) = \left|\Pr\left[\mathcal{A}(Z, U = e(g, \widetilde{g})^{abc}) = 0\right] - \Pr\left[\mathcal{A}(Z, U = X) = 0\right]\right|$$

is negligible in the security parameter λ, *where* $X \in_R \mathbb{G}_T$ *is random.*

2.3 Identity-Based Encryption

Identity-Based Encryption (IBE) is a type of public-key cryptosystem in which any arbitrary string, such as an email address or name, can serve as a public key. This paradigm, introduced by Shamir in 1984, eliminates the need for digital certificates to bind public keys to user identities, thereby simplifying key management in distributed systems.

In a traditional Public Key Infrastructure (PKI), users must obtain and verify certificates issued by a trusted Certificate Authority (CA). However, in an IBE system, a trusted authority called the *Private Key Generator* (PKG) is responsible for generating private keys corresponding to user identities. This enables senders to encrypt messages using publicly known identifiers (e.g., `bob@company.com`), even if the recipient has not yet retrieved their private key.

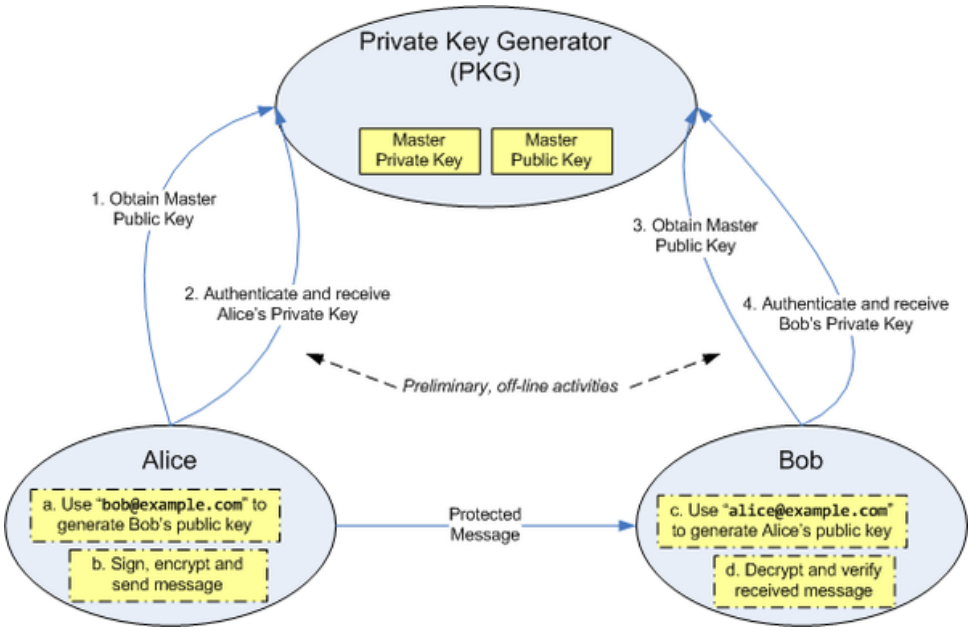


Figure 2.1: Illustration of Identity-Based Encryption (IBE) system.

The security of IBE systems is typically defined using notions such as IND-ID-CPA (indistinguishability under chosen-plaintext attack) and IND-ID-CCA (chosen-ciphertext attack) security. One of the most well-known and efficient IBE constructions is the Boneh-Franklin scheme, which is based on bilinear pairings over elliptic curves and assumes the hardness of a Bilinear Diffie-Hellman (BDH) problem.

2.3.1 Identity-Based Multi-Channel Broadcast Encryption

System Model. An **I**dentity-**B**ased **M**ulti-**C**hannel **B**roadcast **E**ncryption (IB-MCBE) scheme consists of three randomized algorithms — (Setup, Extract, Encrypt) and one deterministic Decryption algorithm . We can formally define the system model of IB-MCBE as follows.

- $\mathsf{Setup}(1^\lambda, L)$: This algorithm is executed by a trusted third party who is also known as the **P**rivate-**K**ey **G**eneration **C**enter (PKGC). The algorithm takes as inputs the length of the security parameter λ and maximal number of users L that the system can accommodate. It then outputs the master public-key MPK and the master secret-key MSK. It finally makes MPK publicly available and keeps MSK secret to itself. For convenience, we take μ groups of users Gr_1, Gr_2, ... Gr_μ with each group Gr_x (for all $x \in [\mu]$) having n number of system users.
- $\mathsf{Extract}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{ID}_{x_i})$: The PKGC takes as input the master public-key MPK, the master secret-key MSK and an identity ID_{x_i} corresponding to the user $x_i \in \mathsf{Gr}_x$. It outputs a secret-key SK_{x_i} for the user $x_i \in \mathsf{Gr}_x$, and finally sends it to the user x_i through a secure communication channel between them.
- $\mathsf{Encrypt}(\mathsf{MPK}, \mathcal{D}, \{M_x\}_{x=1}^{\mu})$: A broadcaster, who has the public information of all users, takes as input the master public-key MPK, a subscribed set $\mathcal{S} = \bigcup_{x=1}^{\mu} \mathsf{S}_x$ with the identity set $\mathcal{D}$, and messages $\{M_x\}_{x=1}^{\mu}$ from the message-space $\mathcal{M}^\mu$. Here, (for all $x \in [\mu]$) the subscribers set $\mathsf{S}_x \subseteq \mathsf{Gr}_x$. It finally outputs a ciphertext CT corresponding to the subscribers set $\mathcal{S}$.
- $\mathsf{Decrypt}(\mathsf{MPK}, \mathsf{SK}_{x_i}, \mathsf{CT})$: A decryptor takes as input the master public-key MPK, the secret-key SK_{x_i} of the user $x_i \in \mathsf{S}_x$ having identity $\mathsf{ID}_{x_i} \in \mathcal{D}$ and the ciphertext CT to either retrieve the correct message M_x or a null string $\perp$ that indicates the decryption failure.

Correctness. We say that the IB-MCBE protocol is correct if, for all security parameters λ, all message sets $\{M_x\}_{x=1}^{\mu} \in \mathcal{M}^\mu$, and all user identities $\mathsf{ID}_{x_i} \in \mathcal{D}$ with $x_i \in \mathsf{S}_x$, the following condition holds:

$$(M_x) \leftarrow \mathsf{Decrypt}\Big(\mathsf{MPK}, \mathsf{Extract}\big(\mathsf{MPK}, \mathsf{MSK}, \mathsf{ID}_{x_i}\big), \mathsf{Encrypt}\big(\mathsf{MPK}, \mathcal{D}, \{M_x\}_{x=1}^{\mu}\big)\Big),$$

where $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, L)$.

Security Framework. The security foundation of the aforementioned IB-MCBE scheme follows from the following security game.

Confidentiality. In the following game, we discuss the security definition for the confidentiality of IB-MCBE scheme under the standard adaptive *indistinguishability under full-identity chosen-ciphertext attack*, denoted by IND-ID-CCA, security model. This game is played between a challenger $\mathcal{C}$ and a PPT adversary $\mathcal{A}$. The game can be described as follows.

- **Setup:** The challenger $\mathcal{C}$ first generates the master public-secret key pair $(\mathsf{MPK}, \mathsf{MSK})$ by running the $\mathsf{Setup}(1^\lambda, L)$ algorithm. It then keeps MSK secret to itself, and sends

MPK to $\mathcal{A}$.

- **Phase-1:** The adversary $\mathcal{A}$ can make polynomial number of different adaptive queries to $\mathcal{C}$ by sending the following two oracle accesses.

  (a) Extract Oracle Query: The adversary $\mathcal{A}$ makes a Key Extract Query to $\mathcal{C}$ by sending the identity ID_{x_i} corresponding to the user $x_i \in \mathsf{Gr}_x$. The challenger $\mathcal{C}$ generates the secret-key SK_{x_i} by running the Extract algorithm, and returns SK_{x_i} to $\mathcal{A}$.

  (b) Decryption Oracle Query: The Adversary $\mathcal{A}$ makes a decryption query by sending the ciphertext $\mathsf{CT}^{'}$ and the identity $\mathsf{ID}^{'}_{x_j}$. In turn, $\mathcal{C}$ first calls the Extract Oracle Query for the identity $\mathsf{ID}^{'}_{x_j}$ to get the corresponding secret-key $\mathsf{SK}^{'}_{x_i}$. Then, it retrieves the correct message by running the main decryption algorithm Decrypt, and sends it to $\mathcal{A}$.

- **Challenge:** The adversary $\mathcal{A}$ submits two sets of messages $\{M_{x0}\}_{x=1}^{\mu}$, $\{M_{x1}\}_{x=1}^{\mu}$ with $|M_{x0}| = |M_{x1}|$ and a target $\mathcal{S}$, whose identity set is $\mathcal{D}^*$, with a limitation that all the identities from $\mathcal{D}^*$ should not appear in the Extract Oracle Query of **Phase-1**. Then, $\mathcal{C}$ randomly picks a bit $\delta \in_R \{0,1\}$ to produce the challenge ciphertext CT^* by running $\mathsf{Encrypt}(\mathsf{MPK}, \mathcal{D}^*, \{M_{x\delta}\}_{x=1}^{\mu})$ algorithm. Finally, $\mathcal{C}$ sends the challenge ciphertext CT^* to $\mathcal{A}$.

- **Phase-2:** The adversary $\mathcal{A}$ is also permitted to make queries (as that of in the **Phase-1**) except Extract Oracle Query for $\mathsf{ID}_{x_j} \in \mathcal{D}^*$ and Decryption Oracle Query for $(\mathsf{CT}^*, \mathsf{ID}_{x_j} \in \mathcal{D}^*)$.

- **Guess:** Finally, the adversary $\mathcal{A}$ produces a guess bit $\delta^{'} \in \{0,1\}$ for δ and wins the game if $\delta^{'} = \delta$.

The adversary $\mathcal{A}$'s advantage in the above security game can be defined as follows.

$$\mathsf{Adv}^{\mathsf{IND\text{-}ID\text{-}CCA}}_{\mathcal{A},\mathsf{IB\text{-}MCBE}}(1^{\lambda}) = \left|\Pr\left(\delta = \delta^{'}\right) - \frac{1}{2}\right|$$

Definition 2.3.1 (Security of Confidentiality)**.** *We say that the IB-MCBE scheme is IND-ID-CCA secure if all PPT adversaries have at most a negligible advantage in winning the above confidentiality game.*

In summary, this chapter provided the essential definitions, cryptographic assumptions, and system model necessary for analyzing our IB-MCBE scheme. These preliminaries form the basis for the design and security analysis discussed in the subsequent chapters.

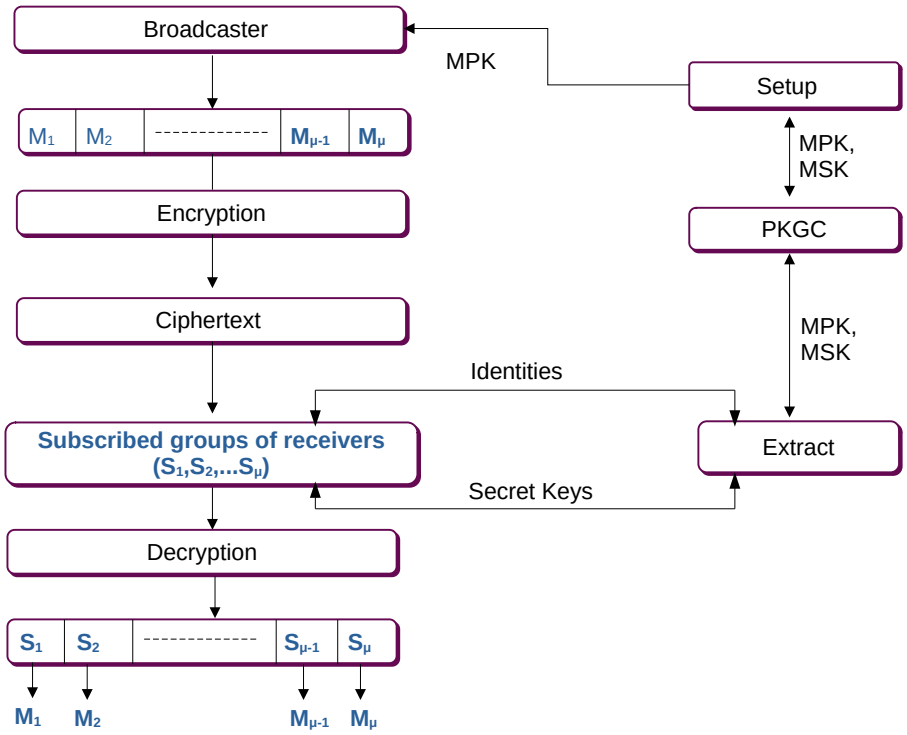


Figure 2.2: General system model of IB-MCBE.

Chapter 3

Proposed IB-MCBE Protocol

This section initiates by illustrating our IB-MCBE protocol and validating its correctness. Following this, we delve into explicating the security analysis concerning our proposed system.

3.0.1 Protocol Description

The communication model of our IB-MCBE scheme, which consists of four algorithms - (Setup, Extract, Encrypt, Decrypt), involves a PKGC, a broadcaster and several groups of subscribed users. These algorithms can be described as follows.

• $\mathsf{Setup}\big(1^\lambda, L\big)$: Let us assume that $\big\{\mathsf{Gr}_x : 1 \leq x \leq \mu\big\}$ be μ groups each having n number of system users. We set l (> 0) as the length of each user's identity. We assign the set of identities $\mathsf{D}_x = \{\mathsf{ID}_{x_i} : i \in \mathcal{I}\}$ corresponding to each group Gr_x of users, where $\mathcal{I}$ represents the index set of D_x with $|\mathcal{I}| = n$. The maximal number of users that the system can accommodate is $L = n \cdot \mu = 2^l$. Taking as input the length of the security parameter λ and the size L of the total number of users, PKGC performs the following steps.

S1. It first generates a prime order T3-map system $\mathbb{G} = \big(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e\big)$. Let g and $\widetilde{g}$ are the two random generators of the source groups $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively.

S2. It then picks random exponents α, β, ζ, and η from $\mathbb{Z}_p^*$ to compute the group elements: $\widetilde{g_1} = \widetilde{g}^\alpha$, $g_2 = g^\beta$, $\widetilde{g_2} = \widetilde{g}^\beta$, $\widetilde{X} = \widetilde{g}^\zeta$, $\widetilde{Y} = \widetilde{g}^\eta$.

S3. It also selects a collusion-resistant cryptographic hash function $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_p^*$ to set the master public-secret key pair as follows.

$$\mathsf{MPK} = \Big(\mathbb{G}, g, \widetilde{g}, \widetilde{g_1}, g_2, \widetilde{X}, \widetilde{Y}, \mathcal{H}\Big),$$
$$\mathsf{MSK} = \Big(\alpha, \beta, \widetilde{g_2}\Big)$$

S4. Finally, it makes MPK publicly available by uploading it to the cloud server and keeps MSK secret.

• $\mathsf{Extract}\big(\mathsf{MPK}, \mathsf{MSK}, \mathsf{ID}_{x_i}\big)$: To generate the secret-key for a user $x_i \in \mathsf{Gr}_x$, the PKGC takes as input master public-key MPK, master secret-key MSK and user's identity $\mathsf{ID}_{x_i} \in \{0,1\}^l$. It then outputs the secret-key SK_{x_i} corresponding to the user x_i by the following computations.

K1. For all the identities $\mathsf{ID}_{x_j} \in \{0,1\}^l$ corresponding to the users $x_j \in \big(\bigcup_{y=1}^{\mu} \mathsf{Gr}_y\big)$, it first computes the hash values $h_{x_j} = \mathcal{H}\big(\mathsf{ID}_{x_j}\big)$.

K2. It then sets the secret-key $\mathsf{SK}_{x_i} = \big(SK_{x_{i0}}, SK_{x_{i1}}, SK_{x_{i2}}\big)$, where the key components are given below.

$$SK_{x_{i0}} = \big(\widetilde{g}_2\big)^{h_{x_i}}$$
$$SK_{x_{i1}} = \big(\widetilde{g}_2\big)^{\left(\sum_{x_j \in \mathsf{Gr}_x \setminus \{x_i\}} h_{x_j}\right)}$$
$$SK_{x_{i2}} = \big(\widetilde{g}_2\big)^{\left(\sum_{x_j \in \left(\bigcup_{y=1}^{\mu} \mathsf{Gr}_y\right) \setminus \{x_i\}} h_{x_j}\right)}$$

K3. Finally, the PKGC sends SK_{x_i} to the user x_i through a secure communication channel.

• $\mathsf{Encrypt}\big(\mathsf{MPK}, \mathcal{D}, \{M_x\}_{x=1}^{\mu}\big)$: Let us assume that $\mathcal{S} = \bigcup_{x=1}^{\mu} \mathsf{S}_x$ be the subscribers set. Here, each set S_x is a subset of the group of users Gr_x for all $x \in \big[\mu\big]$. The set $\mathcal{D}$ of all identities corresponding to the subscribers set $\mathcal{S}$ is given by $\mathcal{D} = \bigcup_{x=1}^{\mu} \{\mathsf{ID}_{x_i} : i \in \mathcal{I}_x\}$, where $0 \leq |\mathcal{I}_x| \leq n$. Taking as inputs the master public-key MPK, the set $\mathcal{D}$ of all identities and messages $M_x \in \mathbb{G}_T$ corresponding to the subscribed groups S_x (for all $x \in \big[\mu\big]$), the broadcaster generates the ciphertext by executing the following steps.

E1. For all the identities $\mathsf{ID}_{y_j} \in \{0,1\}^l$ corresponding to the users $y_j \in \big(\bigcup_{y=1}^{\mu} \mathsf{Gr}_y\big)$, it first computes the hash values $h_{y_j} = \mathcal{H}\big(\mathsf{ID}_{y_j}\big)$.

E2. Define a bilinear pairing function

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T.$$

Now, define a function $f : \mathbb{G}_T \to \mathbb{Z}_p^*$. For each subscriber set S_x $(1 \leq x \leq \mu)$, choose a random value r, $r_x \in \mathbb{Z}_p^*$, and compute $\tilde{g}^r \in \mathbb{G}_2$. Then define

$$f\left(e\left((g_2)^{h_{x_i}}, \tilde{g}^r\right)\right) = r_x, \quad \text{for all } x_i \in \mathsf{S}_x.$$

If $x_1, x_2, \ldots, x_n \in \mathsf{S}_x$, then

$$\begin{aligned} f\left(e\left((g_2)^{h_{x_1}}, \tilde{g}^r\right)\right) &= f\left(e\left((g_2)^{h_{x_2}}, \tilde{g}^r\right)\right) \\ &= \cdots \\ &= f\left(e\left((g_2)^{h_{x_n}}, \tilde{g}^r\right)\right) = r_x. \end{aligned}$$

That is,

$$\begin{aligned} &\text{for all } x \in \mathsf{S}_1,\ f\left(e\left((g_2)^{h_x}, \tilde{g}^r\right)\right) = r_1, \\ &\text{for all } x \in \mathsf{S}_2,\ f\left(e\left((g_2)^{h_x}, \tilde{g}^r\right)\right) = r_2, \\ &\qquad\qquad \vdots \\ &\text{for all } x \in \mathsf{S}_\mu,\ f\left(e\left((g_2)^{h_x}, \tilde{g}^r\right)\right) = r_\mu. \end{aligned}$$

For the rest of the users $x_i \in \bigcup_{x=1}^{\mu}(\mathsf{Gr}_x \setminus \mathsf{S}_x)$, choose a distinct random value $r_{x_i} \in \mathbb{Z}_p^* \setminus \{r_1, r_2, \ldots, r_\mu\}$, and compute $\tilde{g}^r \in \mathbb{G}_2$. Then define

$$f\left(e\left((g_2)^{h_{x_i}}, \tilde{g}^r\right)\right) = r_{x_i}, \quad \text{for all such } x_i,$$

or equivalently, assuming the **Discrete Logarithm (DL)** problem is hard,

$$f\left(e\left((g_2)^{h_x}, \tilde{g}^r\right)\right) = g^{r_x}, \quad \text{for all } x \in \mathsf{Gr}_x \setminus \mathsf{S}_x.$$

E3. It then picks a random exponent s from $\mathbb{Z}_p^*$ to compute the following ciphertext components corresponding to each group Gr_x.

$$\begin{aligned} C_{1,x} &= e\Big(g_2, \Big(\frac{\widetilde{g_1}}{\widetilde{g}}\Big)^{\left(\sum_{x_j \in \mathcal{S} \setminus \mathsf{S}_x} h_{x_j}\right)}\Big)^s \\ &\quad \times \frac{1}{e\Big(g_2, (\widetilde{g})^{\left(\sum_{y_j \in \bigcup_{y=1, y\neq x}^{\mu} \left(\mathsf{Gr}_y \setminus \mathsf{S}_y\right)} h_{y_j}\right)}\Big)^s}, \\ C_{2,x} &= M_x \cdot e\Big(g_2, (\widetilde{g_1})^{\sum_{x_j \in \mathcal{S} \setminus \mathsf{S}_x} h_{x_j}}\Big)^s, \\ C_{0,x} &= g^{s \cdot r_x}, \quad C = g^r, \quad \theta_x = \mathcal{H}\big(g^s, C_{1,x}, C_{2,x}\big), \\ \widetilde{\Gamma_x} &= \Big(\widetilde{X} \cdot \widetilde{Y}^{\theta_x}\Big)^s \end{aligned}$$

E4. Finally, the broadcaster publishes $\mathsf{CT} = \big(C, \{C_{0,x}\}_{x=1}^{\mu}, \{C_{1,x}\}_{x=1}^{\mu}, \{C_{2,x}\}_{x=1}^{\mu}, \{\widetilde{\Gamma_x}\}_{x=1}^{\mu}, f\big)$ as the ciphertext and uploads it to the cloud server.

• $\mathsf{Decrypt}\big(\mathsf{MPK}, \mathsf{SK}_{x_i}, \mathsf{CT}\big)$: A decryptor $x_i \in \mathsf{S}_x$ ($\subseteq \mathsf{Gr}_x$), with the identity $\mathsf{ID}_{x_i} \in \{0,1\}^l$, takes as inputs the master public-key MPK, its secret-key $\mathsf{SK}_{x_i} = \big(SK_{x_{i0}}, SK_{x_{i1}}, SK_{x_{i2}}\big)$ and the ciphertext $\mathsf{CT} = \big(C, \{C_{0,x}\}_{x=1}^{\mu}, \{C_{1,x}\}_{x=1}^{\mu}, \{C_{2,x}\}_{x=1}^{\mu}, \{\widetilde{\Gamma_x}\}_{x=1}^{\mu}, f\big)$ from the cloud

server to either recover the correct message M_x corresponding to the group Gr_x or get a designated symbol $\perp$ indicating the decryption failure.

D1. For each decryptor $x_i \in \mathsf{S}_x \subseteq \mathsf{Gr}_x$ with identity ID_{x_i}, the decryptor performs the following steps:

- Computes the pairing between the ciphertext component $C = g^r$ and the secret key $\mathrm{SK}_{x_i} = \widetilde{g}^{\beta^{h_{x_i}}}$:

$$e(C, \mathrm{SK}_{x_i}) = e(g^r, \widetilde{g}^{\beta^{h_{x_i}}}) = e(g^{\beta^{h_{x_i}}}, \widetilde{g}^r)$$

- Computes r_x using a function $f : \mathbb{G}_T \to \mathbb{Z}_p^*$ defined during encryption:

$$r_x = f\left(e(g^{\beta^{h_{x_i}}}, \widetilde{g}^r)\right)$$

- Recovers the masked message component:

$$g^s = C_{0,x}{}^{\frac{1}{r_x}}$$

D2. It first recovers the components by executing the below computations.

$$M_x^{'} = \frac{C_{2,x}}{C_{1,x}} \cdot \frac{e\big(g^s, SK_{x_{i0}} \cdot SK_{x_{i1}}\big)}{e\big(g^s, SK_{x_{i0}} \cdot SK_{x_{i2}}\big)},$$
$$\theta_x^{'} = \mathcal{H}\big(g^s, C_{1,x}, C_{2,x}\big)$$

D3 . It then retrieves the original message as

$$M_x = \begin{cases} M_x^{'} & \text{if } e\big(C_0, \widetilde{X} \cdot \widetilde{Y}^{\theta_x^{'}}\big) = e\big(g, \widetilde{\Gamma_x}\big) \\ \perp & \text{otherwise.} \end{cases}$$

Note. The inputs of hash function $\mathcal{H}$ are C_0, $C_{1,x}$, and $C_{2,x}$ for $x \in [\mu]$. Since C_0, $C_{1,x}$, and $C_{2,x}$ are all either from source groups $\mathbb{G}_1$ and $\mathbb{G}_2$ or from target group $\mathbb{G}_T$, we use the binary representation of these elements and concatenate them to handle any ambiguity.

Correctness. Assume that a user $x_i \in \mathsf{Gr}_x$, with its identity $\mathsf{ID}_{x_i} \in \mathcal{D}$ and the ciphertext $\mathsf{CT} = \big(C_0, \{C_{1,x}\}_{x=1}^{\mu}, \{C_{2,x}\}_{x=1}^{\mu}, \{\widetilde{\Gamma_x}\}_{x=1}^{\mu}\big)$, executes the decryption algorithm. If the user x_i is not a subscribed user, then the following computation will output as the decryption failure symbol $\perp$. However, for a subscribed user x_i, the correctness of our decryption

algorithm Decrypt immediately follows from the following computations.

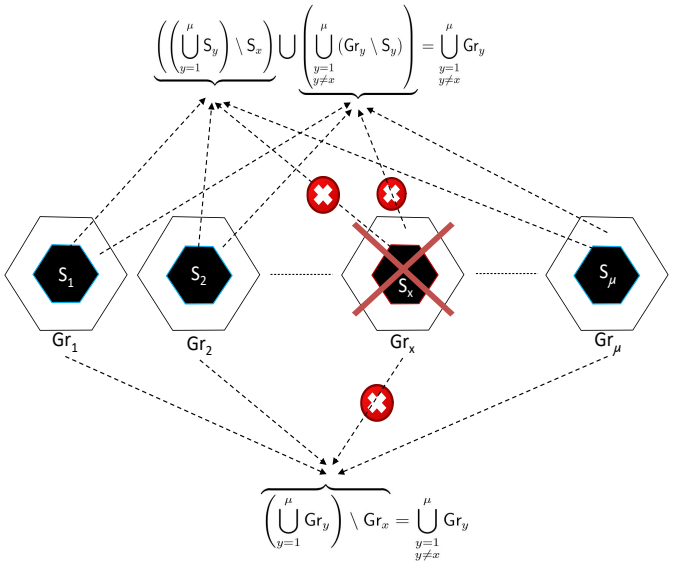
$$
\begin{aligned}
M_x^{'} &= \frac{C_{2,x}}{C_{1,x}} \cdot \frac{e\big(g^s, SK_{x_{i0}} \cdot SK_{x_{i1}}\big)}{e\big(g^s, SK_{x_{i0}} \cdot SK_{x_{i2}}\big)} \\
&= M_x \times \; e\Big(g^{\beta}, \widetilde{g}^{\Big(\sum_{y_j \in \bigcup_{y=1, y\neq x}^{\mu} (\mathsf{Gr}_y \setminus \mathsf{S}_y)} h_{y_j}\Big)}\Big)^s \\
&\qquad \times \frac{e\Big(g^{\beta}, \widetilde{g}^{\Big(\alpha \cdot \sum_{x_j \in \mathcal{S} \setminus \mathsf{S}_x} h_{x_j}\Big)}\Big)^s}{e\Big(g^{\beta}, (\widetilde{g})^{(\alpha-1)\cdot(\sum_{x_j \in \mathcal{S} \setminus \mathsf{S}_x} h_{x_j})}\Big)^s} \\
&\qquad \times \frac{e\Big(g^s, \big(\widetilde{g}^{\beta}\big)^{(\sum_{x_j \in \mathsf{Gr}_x} h_{x_j})}\Big)}{e\Big(g^s, \big(\widetilde{g}^{\beta}\big)^{(\sum_{x_j \in (\bigcup_{y=1}^{\mu} \mathsf{Gr}_y)} h_{x_j})}\Big)} \\
&= M_x \times \; e\Big(g, \widetilde{g}\Big)^{\Big(\beta s \sum_{y_j \in \bigcup_{y=1, y\neq x}^{\mu} (\mathsf{Gr}_y \setminus \mathsf{S}_y)} h_{y_j}\Big)} \\
&\qquad \times \frac{e\Big(g, \widetilde{g}\Big)^{\Big(\alpha\beta s \sum_{x_j \in \mathcal{S} \setminus \mathsf{S}_x} h_{x_j}\Big)}}{e\Big(g, \widetilde{g}\Big)^{(\alpha-1)\beta s(\sum_{x_j \in \mathcal{S} \setminus \mathsf{S}_x} h_{x_j})}} \\
&\qquad \times \frac{e\Big(g, \widetilde{g}\Big)^{\Big(\beta s \sum_{x_j \in \mathsf{Gr}_x} h_{x_j}\Big)}}{e\Big(g, \widetilde{g}\Big)^{\Big(\beta s \sum_{x_j \in (\bigcup_{y=1}^{\mu} \mathsf{Gr}_y)} h_{x_j}\Big)}} \\
&= M_x \times \boxed{e\Big(g, \widetilde{g}\Big)^{\Big(\beta s \sum_{y_j \in \bigcup_{y=1, y\neq x}^{\mu} (\mathsf{Gr}_y \setminus \mathsf{S}_y)} h_{y_j}\Big)}} \\
&\qquad \times \boxed{\frac{e\Big(g, \widetilde{g}\Big)^{\Big(\beta s \sum_{x_j \in \mathcal{S} \setminus \mathsf{S}_x} h_{x_j}\Big)}}{e\Big(g, \widetilde{g}\Big)^{\Big(\beta s \sum_{y_j \in (\bigcup_{y=1}^{\mu} \mathsf{Gr}_y) \setminus \mathsf{Gr}_x} h_{y_j}\Big)}}} \\
&= M_x \times \boxed{\frac{e\Big(g, \widetilde{g}\Big)^{\Big(\beta s \sum_{y_j \in \bigcup_{y=1, y\neq x}^{\mu} \mathsf{Gr}_y} h_{y_j}\Big)}}{e\Big(g, \widetilde{g}\Big)^{\Big(\beta s \sum_{y_j \in \bigcup_{y=1, y\neq x}^{\mu} \mathsf{Gr}_y} h_{y_j}\Big)}}} \\
&= M_x
\end{aligned}
\tag{3.1}
$$

$$
\theta_x^{'} = \mathcal{H}(C_0, C_{1,x}, C_{2,x}) = \theta_x \tag{3.2}
$$

$$
\begin{aligned}
e\big(C_0, \widetilde{X} \cdot \widetilde{Y}^{\theta_x^{'}}\big) &= e\big(g^s, \widetilde{X} \cdot \widetilde{Y}^{\theta_x}\big) = e\big(g, \widetilde{X} \cdot \widetilde{Y}^{\theta_x}\big)^s \\
&= e\big(g, (\widetilde{X} \cdot \widetilde{Y}^{\theta_x})^s\big) = e\big(g, \widetilde{\Gamma_x}\big)
\end{aligned}
\tag{3.3}
$$

Observe that the correctness of the transition from the previous step to the next step of Equation 3.1, which is rounded by a dotted-box, can be readily realized from the **Fig. 3.1**.

Figure 3.1: The pictorial view of correctness



Remark 3.0.1. *The system initializes the total number of users during the Setup algorithm. The PKGC then generates the secret key for each user and distributes it to them through a secure channel. In our system, the information of all system users (i.e., the identities of the system users) is required for computing the secret key components and the ciphertext. As a result, if a user joins or leaves the system, then the previous Setup, Extract, and Encrypt algorithms do not work for the new system; the algorithms need to run again for the new system. Therefore, our system does not satisfy the dynamic property. For a dynamic system, the secret key generation for a new user or a defector does not depend on the information of others. Therefore, if we want to employ the algorithms Join and Revocation in the system, the system must be designed in such a way that the Setup and Extract algorithms do not require the information of all system users. Employing the Join and Revocation algorithms in our system is an interesting direction for future research.*

Chapter 4

Security Analysis

In this chapter, we show the security of confidentiality of our IB-MCBE construction under the standard adaptive IND-ID-CCA security model without the ROM. assuming the hardness of the Decisional Bilinear Diffie-Hellman Type-3 (DBDH-3) problem. To demonstrate this, we construct a reduction from any adversary capable of breaking the IB-MCBE scheme to a simulator that can solve the DBDH-3 problem, thereby proving the confidentiality of the system.

Theorem 4.0.1 (Security of Confidentiality)**.** *We show that our IB-MCBE protocol (cf. Sect. 3.0.1) achieves adaptive IND-ID-CCA security (cf. Def. 2.3.1) under the standard asymmetric DBDH-3 assumption (cf. Def. 2.2.1).*

Proof. We assume that there exists a PPT adversary $\mathcal{A}$ that wins the confidentiality game of our proposed IB-MCBE scheme with a non-negligible advantage. The adversary $\mathcal{A}$ is allowed to make at most polynomial number, say $m = \mathsf{poly}(\lambda)$, of user's secret-key queries or encryption and decryption queries against our system. Here, $\mathsf{poly}(\lambda)$ denotes a polynomial in the security parameter λ. We now show that we can construct a simulator $\mathcal{B}$ that can solve the asymmetric DBDH-3 assumption using $\mathcal{A}$ as a subroutine. The simulator $\mathcal{B}$ plays the role of the challenger in our confidentiality security game. At the beginning of the game, $\mathcal{B}$ obtains the asymmetric DBDH-3 challenge instance $\langle Z = \big(\mathbb{G}, g, \widetilde{g}, \widetilde{g}^a, \widetilde{g}^b, g^b, g^c, \widetilde{g}^c\big), U\rangle$ to determine whether U is either $e\big(g, \widetilde{g}\big)^{abc}$ or a random element X from the target group $\mathbb{G}_T$. Here, a, b and c are randomly chosen from $\mathbb{Z}_p^*$. Now, the simulator $\mathcal{B}$ and the adversary $\mathcal{A}$ plays the game of confidentiality as follows.

• **Setup:** The simulator $\mathcal{B}$ executes the following steps.

(i) Initially, $\mathcal{B}$ picks $\zeta, \eta \in_R \mathbb{Z}_p^*$ and sets the following elements utilizing the asymmetric DBDH-3 challenge instance.

$$\widetilde{g}_1 = \widetilde{g}^a,\ g_2 = g^b,\ \widetilde{X} = \widetilde{g}^{\zeta},\ \widetilde{Y} = \widetilde{g}^{\eta}.$$

(ii) It then picks a collusion-resistant cryptographic hash functions $\mathcal{H} : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$.

(iii) Finally, it sends the simulated version of the master public-key $\mathsf{MPK} = \big(\mathbb{G},\, g,\, \widetilde{g},\, \widetilde{g}_1,\, g_2,\, \widetilde{X},\, \widetilde{Y},\, \mathcal{H}\big)$ to $\mathcal{A}$.

• **Phase-1:** The adversary $\mathcal{A}$ is allowed to adaptively issue polynomially many queries, say $m = \mathsf{poly}(\lambda)$, in the following two oracles.

(a) Extract Oracle Query: The adversary $\mathcal{A}$ sends a query to $\mathcal{B}$ for an identity $\mathsf{ID}_{x_i} \in \{0,1\}^l$. To return a valid secret-key, $\mathcal{B}$ computes $h_{x_j} = \mathcal{H}(\mathsf{ID}_{x_j}) \in \mathbb{Z}_p^*$ for all identity ID_{x_j}. It then sets the secret-key $\mathsf{SK}_{x_i} = \big(SK_{x_{i0}},\, SK_{x_{i1}},\, SK_{x_{i2}}\big)$, where

$$SK_{x_{i0}} = \big(\widetilde{g}^b\big)^{h_{x_i}},$$
$$SK_{x_{i1}} = \big(\widetilde{g}^b\big)^{\left(\sum_{x_j \in \mathsf{Gr}_x \setminus \{x_i\}} h_{x_j}\right)},$$
$$SK_{x_{i2}} = \big(\widetilde{g}^b\big)^{\left(\sum_{x_j \in (\bigcup_{y=1}^{\mu} \mathsf{Gr}_y) \setminus \{x_i\}} h_{x_j}\right)}$$

Finally, the simulator $\mathcal{B}$ returns the simulated secret-key SK_i to the adversary $\mathcal{A}$.

(c) Decryption Oracle Query: The adversary $\mathcal{A}$ may issue decryption queries for any ciphertext $\mathsf{CT}^{'} = \big(c', \big\{C'_{0,x}\big\}_{x=1}^{\mu}, \big\{C'_{1,x}\big\}_{x=1}^{\mu}, \big\{C'_{2,x}\big\}_{x=1}^{\mu}, \big\{\widetilde{\Gamma}'_x\big\}_{x=1}^{\mu}, f\big)$ and identity $\mathsf{ID}^{'}_{x_i} \in \{0,1\}^l$.

For valid queries, $\mathcal{B}$ first invokes the Extract Oracle Query for the identity $\mathsf{ID}^{'}_{x_i}$ to obtain the corresponding secret key $\mathsf{SK}^{'}_{x_i}$. Then, it responds to $\mathcal{A}$ by executing the decryption algorithm:

$$\mathsf{Decrypt}\big(\mathsf{MPK}, \mathsf{SK}^{'}_{x_i}, \mathsf{CT}^{'}\big).$$

• **Challenge:** The adversary $\mathcal{A}$ submits two set of messages $\{M_{x0}\}_{x=1}^{\mu}$, $\{M_{x1}\}_{x=1}^{\mu}$ with $|M_{x0}| = |M_{x1}|$ for all $x \in \big[\mu\big]$. It also submits a target subscribers set $\mathcal{S}^* = \bigcup_{x=1}^{\mu} \mathsf{S}_x^*$, whose corresponding set of identities $\mathcal{D}^* = \bigcup_{x=1}^{\mu} \{\mathsf{ID}_{x_i}^* : i \in \mathcal{I}_x^*\}$ with a restriction that any identity from $\mathcal{D}^*$ should not appear in the Phase-1. Here, $\mathcal{I}_x^*$ is the index set corresponding to the subscribers set S_x^*. Then, $\mathcal{B}$ picks a random bit $\delta \in_R \{0,1\}$ and computes the components of the challenge ciphertext corresponding to the set of identities $\mathcal{D}^*$ as follows.

i Define a bilinear pairing function

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T.$$

Now, define a function $f : \mathbb{G}_T \rightarrow \mathbb{Z}_p^*$. choose a random value $r^* \in \mathbb{Z}_p^*$. For each subscriber set S_x $(1 \leq x \leq \mu)$, choose a random value $r_x^* \in \mathbb{Z}_p^*$, and compute $\tilde{g}^{r^*} \in \mathbb{G}_2$. Then define

$$f\left(e\left((g_2)^{h_{x_i}}, \tilde{g}^{r^*}\right)\right) = r_x^*, \quad \text{for all } x_i \in \mathsf{S}_x.$$

If $x_1, x_2, \ldots, x_n \in \mathsf{S}_x$, then

$$\begin{aligned} f\left(e\left((g_2)^{h_{x_1}}, \tilde{g}^{r^*}\right)\right) &= f\left(e\left((g_2)^{h_{x_2}}, \tilde{g}^{r^*}\right)\right) \\ &= \cdots \\ &= f\left(e\left((g_2)^{h_{x_n}}, \tilde{g}^{r^*}\right)\right) = r^*_x. \end{aligned}$$

For the rest of the users $x_i \in \bigcup_{x=1}^{\mu}(\mathsf{Gr}_x \setminus \mathsf{S}_x)$, choose a distinct random value $r^*_{x_i} \in \mathbb{Z}^*_p \setminus \{r^*_1, r^*_2, \ldots, r^*_\mu\}$, and compute $\tilde{g}^{r^*} \in \mathbb{G}_2$. Then define

$$f\left(e\left((g_2)^{h_{x_i}}, \tilde{g}^{r^*}\right)\right) = r^*_{x_i}, \quad \text{for all such } x_i,$$

or equivalently, assuming the **Discrete Logarithm (DL)** problem is hard,

$$f\left(e\left((g_2)^{h_x}, \tilde{g}^{r^*}\right)\right) = g^{r^*_x}, \quad \text{for all } x \in \mathsf{Gr}_x \setminus \mathsf{S}_x.$$

(ii) For all the identities $\mathsf{ID}^*_{y_j} \in \{0,1\}^l$ corresponding to the users $y^*_j \in \left(\bigcup_{y=1}^{\mu} \mathsf{Gr}_y\right)$, it first computes the hash values $h^*_{y_j} = \mathcal{H}\left(\mathsf{ID}_{y^*_j}\right)$. It then computes the following ciphertext components corresponding to the group Gr_x of users.

$$\begin{aligned} C_{0,x} &= g^{s \cdot r_x}, \quad C = g^r, \\ C_{2,x} &= M_{x\delta} \cdot U^{\left(\sum_{x_i \in \mathcal{S}^* \setminus \mathsf{S}^*_x} h^*_{x_i}\right)}, \\ C_{1,x} &= \left(\frac{U}{e(g^b, \widetilde{g}^c)}\right)^{\left(\sum_{x_i \in \mathcal{S}^* \setminus \mathsf{S}^*_x} h^*_{x_i}\right)} \\ &\qquad \times \left(\frac{1}{e\left(g^b, \widetilde{g}^c\right)}\right)^{\left(\sum_{y_j \in \bigcup_{y=1, y\neq x}^{\mu} (\mathsf{Gr}_y \setminus \mathsf{S}^*_y)} h^*_{y_j}\right)}, \\ \theta_x &= \widetilde{\mathcal{H}}(C_0, C_{1,x}, C_{2,x}),\ \widetilde{\Gamma_x} = \left(\widetilde{g}^c\right)^{\hat{x} + \theta_x \cdot \hat{y}} \end{aligned}$$

(iii) Finally, $\mathcal{B}$ sends to $\mathcal{A}$ the challenge ciphertext $\mathsf{CT}^* = \left(C, \left\{C_{0,x}\right\}_{x=1}^{\mu}, \left\{C_{1,x}\right\}_{x=1}^{\mu}, \left\{C_{2,x}\right\}_{x=1}^{\mu}, \left\{\widetilde{\Gamma_x}\right\}_{x=1}^{\mu}, f\right)$.

• **Phase-2:** The adversary $\mathcal{A}$ is permitted to adaptively make polynomially many Query (same as in the Phase-1) except the Extract Oracle Query for $\mathsf{ID}_{x_i} \in D^*$ and Decryption Oracle Query for $\left(\mathsf{CT}^*, \mathsf{ID}_{x_i} \in D^*\right)$.

• **Guess:** Finally, $\mathcal{A}$ sends a guess bit $\delta' \in \{0,1\}$ for δ and wins the game if $\delta' = \delta$.

Perfect Simulation. Since the simulator $\mathcal{B}$ has full control over the master secrect-key MSK, the components of the secret-key $\mathsf{SK}_{x_i} = \left(SK_{x_{i0}}, SK_{x_{i1}}, SK_{x_{i2}}\right)$ are valid secret-key components as that of in the original protocol. In the following, we can show that each component of SK_{x_i} is valid.

$$SK_{x_{i0}} = \left(\widetilde{g}^b\right)^{h_{x_i}} = \left(\widetilde{g}_2\right)^{h_{x_i}},$$

$$
\begin{aligned}
SK_{x_{i1}} &= \big(\widetilde{g}^{b}\big)^{\left(\sum_{x_j \in \mathsf{Gr}_x \setminus \{x_i\}} h_{x_j}\right)} \\
&= \big(\widetilde{g}_2\big)^{\left(\sum_{x_j \in \mathsf{Gr}_x \setminus \{x_i\}} h_{x_j}\right)}, \\
SK_{x_{i2}} &= \big(\widetilde{g}^{b}\big)^{\left(\sum_{x_j \in (\bigcup_{y=1}^{\mu} \mathsf{Gr}_y) \setminus \{x_i\}} h_{x_j}\right)} \\
&= \big(\widetilde{g}_2\big)^{\left(\sum_{x_j \in (\bigcup_{y=1}^{\mu} \mathsf{Gr}_y) \setminus \{x_i\}} h_{x_j}\right)}
\end{aligned}
$$

To compute the user's secret-key, we have explicitly set $\beta = b$ in the above computations.

For the challenge ciphertext CT^*, we again have explicitly set $c = s$. If $U = e\big(g, \widetilde{g}\big)^{abc}$, then we have the ciphertext components as follows.

$$
\begin{aligned}
C_{0,x} &= g^{c \cdot r_x}, \\
C_{1,x} &= \left(\frac{U}{e(g^b, \widetilde{g}^c)}\right)^{\left(\sum_{x_i \in \mathcal{S}^* \setminus \mathsf{S}_x^*} h^*_{x_i}\right)} \\
&\times \left(\frac{1}{e\left(g^b, \widetilde{g}^c\right)}\right)^{\left(\sum_{y_j \in \bigcup_{y=1, y \neq x}^{\mu} (\mathsf{Gr}_y \setminus \mathsf{S}_y^*)} h^*_{y_j}\right)} \\
&= e\big(g^b, \widetilde{g}^{(a-1)}\big)^{\left(c \cdot \sum_{x_i \in \mathcal{S}^* \setminus \mathsf{S}_x^*} h^*_{x_i}\right)} \\
&\times \left(\frac{1}{e\left(g^b, \widetilde{g}\right)}\right)^{\left(c \cdot \sum_{y_j \in \bigcup_{y=1, y \neq x}^{\mu} (\mathsf{Gr}_y \setminus \mathsf{S}_y^*)} h^*_{y_j}\right)} \\
&= e\left(g_2, \left(\frac{\widetilde{g}_1}{\widetilde{g}}\right)^{\sum_{x_j \in \mathcal{S}^* \setminus \mathsf{S}_x^*} h^*_{x_j}}\right)^c \\
&\times \frac{1}{e\left(g_2, \left(\widetilde{g}\right)^{\left(\sum_{y_j \in \bigcup_{y=1, y \neq x}^{\mu} (\mathsf{Gr}_y \setminus \mathsf{S}_y^*)} h^*_{y_j}\right)}\right)^c}, \\
C_{2,x} &= M_{x\delta} \cdot U^{\left(\sum_{x_i \in \mathcal{S}^* \setminus \mathsf{S}_x^*} h^*_{x_i}\right)} \\
&= M_{x\delta} \cdot e\left(g^b, \widetilde{g}^a\right)^{\left(c \cdot \sum_{x_j \in \mathcal{S}^* \setminus \mathsf{S}_x^*} h^*_{x_i}\right)} \\
&= M_{x\delta} \cdot e\left(g_2, \left(\widetilde{g}_1\right)^{\left(\sum_{x_i \in \mathcal{S}^* \setminus \mathsf{S}_x^*} h^*_{x_i}\right)}\right)^c, \\
\widetilde{\Gamma_x} &= \big(\widetilde{g}^c\big)^{\hat{x} + \theta_x \cdot \hat{y}} = \big(\widetilde{g}^{\hat{x}} \cdot \big(\widetilde{g}^{\hat{y}}\big)^{\theta_x}\big)^c = \big(\widetilde{X} \cdot \widetilde{Y}^{\theta_x}\big)^c
\end{aligned}
$$

It is readily follows from the above calculations that CT^* is well-defined and perfectly simulated. When U is random, the distribution of δ is fully independent from $\mathcal{A}$'s view point. Thus, $\mathcal{A}$'s advantage in the above confidentiality game must be 0.

Probability Analysis. If $\delta' = \delta$, then $\mathcal{B}$ outputs 0 that indicates $U = e\big(g, \widetilde{g}\big)^{abc}$; otherwise, it outputs 1 that indicates $U = X$, a random element of the target group $\mathbb{G}_T$. From the above computations, the simulation of $\mathcal{B}$ is perfect when $U = e\big(g, \widetilde{g}\big)^{abc}$. Therefore, we have

$$
\Pr\Big[\mathcal{B}\Big(Z, U = e\big(g, \widetilde{g}\big)^{abc}\Big) = 0\Big] = \frac{1}{2} + \mathsf{Adv}^{\mathsf{IND\text{-}ID\text{-}CCA}}_{\mathcal{A}, \mathsf{IB\text{-}MCBE}}(1^\lambda),
$$

where $\mathsf{Adv}^{\mathsf{IND\text{-}ID\text{-}CCA}}_{\mathcal{A}, \mathsf{IB\text{-}MCBE}}(1^\lambda)$ is the advantage of $\mathcal{A}$ in the above confidentiality game. However, when $U = X$, a random element of $\mathbb{G}_T$, then the message is completely unknown for $\mathcal{A}$.

Therefore, we have the following probability.

$$\Pr\Big[\mathcal{B}\big(Z, U = X\big) = 0\Big] = \frac{1}{2}$$

Hence, the advantage of $\mathcal{B}$ in breaking the asymmetric DBDH-3 challenge can be given as follows.

$$\begin{aligned}
\mathsf{Adv}_{\mathcal{B}}^{\text{DBDH-3}}(1^\lambda) =& \Big|\Pr\Big[\mathcal{B}\big(Z, U = e(g, \widetilde{g})^{abc}\big) = 0\Big] \\
& \qquad - \Pr\Big[\mathcal{B}\big(Z, U = X\big) = 0\Big]\Big| \\
=& \Big|\frac{1}{2} + \mathsf{Adv}_{\mathcal{A},\text{IB-MCBE}}^{\text{IND-ID-CCA}}(1^\lambda) - \frac{1}{2}\Big| \\
=& \mathsf{Adv}_{\mathcal{A},\text{IB-MCBE}}^{\text{IND-ID-CCA}}(1^\lambda)
\end{aligned}$$

Therefore, if $\mathcal{A}$ has non-negligible advantage to get correct $\delta^{'} \in \{0, 1\}$, then $\mathcal{B}$ correctly predicts either $U = e\big(g, \widetilde{g}\big)^{abc}$ or $U = X$, a random element of $\mathbb{G}_T$. This implies that $\mathcal{B}$ breaks the asymmetric DBDH-3 challenge with a non-negligible advantage.

Hence, the proof follows.

In this chapter, we have formally proven the IND-ID-CCA security of our IB-MCBE scheme. By constructing a simulator that uses an adversary to distinguish the DBDH-3 challenge, we demonstrated that any non-negligible advantage against our scheme would imply a break of the underlying hard problem. This reduction confirms the confidentiality and robustness of our protocol in the standard model.

In the next chapter, we briefly discuss the comparison of decryption time with other related models and present the implementation results to demonstrate the practical efficiency of our scheme. □

Chapter 5

Comparative Study

In this section, we present comparative studies among our IB-MCBE protocol and the existing recent MCBEs [31]–[34] in **Tab. 5.1**, **5.2** and **5.3**. It is worth noting that all hash values in our IB-MCBE are pre-computed so that we can ignore them in our study. More precisely, we can discuss the efficiency of our scheme as follows.

- Our IB-MCBE has achieved $O(\mu)$-size communication bandwidth that is similar to the works of [31]–[34] (cf. **Tab. 5.1**). On the other hand, the size of the user's secret keys is constant, which is similar to the constructions of [31]–[34]. However, we emphasize that, in our IB-MCBE, we have significantly reduced the size of public parameters (i.e., from linear to constant), which is a plausible improvement as opposed to the seminal works [31]–[34]. Due to low storage overhead and less communication bandwidth, our scheme can be employed on IoT-enabled small devices.
- The comparative studies of security and other functionalities among our IB-MCBE and existing MCBEs [31]–[34] are shown in **Tab. 5.2**. Ours IB-MCBE has achieved the most robust adaptive IND-ID-CCA security in the standard model, whereas the works of [31]–[34] are able to accomplish only the weaker selective security. However, we emphasize that we have used the standard DBDH-3 assumption to achieve the adaptive IND-ID-CCA security, whereas the designs of [31]–[34] are used the nonstandard q-type complexity assumptions. We note that the constructions [31] are designed over the private-key setting. On the other hand, the schemes of [32]–[34], including ours IB-MCBE, support the public-key setting, where anyone can play the role of the broadcaster. Moreover, our most secure T3-map based construction can accommodate an unbounded (or arbitrary) number of system users (due to the use of IBE framework), whereas the comparatively less secure T1-map based designs of [31]–[33] can only support a polynomial number of users.
- **Tab. 5.3** shows the comparative studies of computation costs, such as public-key size, encryption time, and decryption time. Our main contribution is that

Table 5.1: Comparative studies of communication bandwidth and storage overhead

Scheme	Communication	Storage	
	Ciphertext	Public-parameter	Secret-key
Zhao et al. [31]	$\mu\|\mathsf{B}_t\| + 2\|\mathsf{B}_s\|$	$(2L+\mu-1)\|\mathsf{B}_s\|$	$\|\mathsf{B}_s\|$
Acharya et al. [32]	$\mu\|\mathsf{B}_t\| + 2\|\mathsf{B}_s\|$	$(3L+\mu)\|\mathsf{B}_s\|$	$\|\mathsf{B}_s\| + \|\mathbb{Z}_p^*\|$
Le et al. [34]	$\mu\|\mathsf{B}_t\| + 2\|\mathsf{B}_s\|$	$(\mu L + L + 1)\|\mathsf{B}_s\| + \mu\|\mathsf{B}_t\|$	$\|\mathsf{B}_s\|$
Acharya et al. [33]-I	$\mu\|\mathsf{B}_t\| + 2\|\mathsf{B}_s\|$	$2L\|\mathsf{B}_s\|$	$2\|\mathsf{B}_s\|$
Acharya et al. [33]-II	$\mu\|\mathsf{B}_t\| + 2\|\mathsf{B}_s\|$	$(3L+\mu+3)\|\mathsf{B}_s\|$	$2\|\mathsf{B}_s\|$
Ours	$3\mu\|\mathsf{A}_t\| + (\mu+2)\|\mathsf{A}_s\|$	$6\|\mathsf{A}_s\|$	$3\|\mathsf{A}_s\|$

L = total number of users within the system, μ = total number of messages, $|\mathsf{B}_s|$ = bit size of an element belonging to the prime order symmetric bilinear source group, $|\mathsf{B}_t|$ =bit size of an element within the prime order symmetric bilinear target group, $|\mathsf{A}_s|$ = bit size of an element in either $\mathbb{G}_1$ or $\mathbb{G}_2$, $|\mathsf{A}_t|$ = bit size of an element within $\mathbb{G}_T$.

Table 5.2: Comparative studies of security and other functionalities

Scheme	Group type	ROM	Pub-Key	IBE	Security	
					Model	Assumption
Zhao et al. [31]	Prime Order, T1	✗	No	✗	SEL-CPA	q-DBDHE
Acharya et al. [32]	Prime Order, T1	✗	Yes	✗	SEMI-CPA	DBDHE-sum
Le et al. [34]	Prime Order, T3	✓	Yes	✗	SEL-CPA	GDDHE
Acharya et al. [33]-I	Prime Order, T1	✗	Yes	✗	SEL-CPA	q-DBDHE
Acharya et al. [33]-II	Prime Order, T1	✗	Yes	✗	ADAP-CPA	mDBDHE
Ours	Prime Order, T3	✗	Yes	✓	ADAP-CCA	DBDH-3

IBE = identity-based encryption, Pub-Key = public-key setting, ROM = random oracle model, SEL-CPA = Selective chosen-plaintext attack, SEL-CCA = Selective chosen-ciphertext attack, SEMI-CPA = Semi-static chosen-plaintext attack, ADAP-CPA = adaptive chosen-plaintext attack, ADAP-CCA = adaptive chosen-ciphertext attack, T1 = Type-1 map, T3 = Type-3 map, q-DBDHE = q-type decisional bilinear Diffie-Hellman exponent, GDDHE = gap decisional Diffie-Hellman exponent, mDBDHE = modified decisional bilinear Diffie-Hellman exponent, DBDH-3 = decisional bilinear Diffie-Hellman Type-3, DBDHE-sum = modified decisional bilinear Diffie-Hellman exponent sum.

we have significantly reduced the cost for public-key as opposed to the works of [31]–[34]. Our proposed IB-MCBE requires only four exponentiation to construct the public-keys, whereas the schemes [31]–[34] require $O(L)$ exponentiation and the design of [34] needs additional $O(\mu)$ pairings to generate the public-keys. Since less exponentiation and multiplication are required in our encryption algorithm, we can readily explicate that our encryption cost is significantly less compared to the systems of [31]–[33]. However, ours and [34] have similar encryption costs, which have two exponentiations in bilinear source groups, $O(\mu)$ exponentiation in bilinear target group, $O(\mu)$ multiplication, and $O(\mu)$ pairing operations. Moreover, the decryption costs of our work is significantly less as opposed to the works [31]–[34].

Table 5.3: Comparative studies of computation cost

Scheme	Public-parameter		Secret-key	Encryption Time				Decryption Time			
	$\#\mathsf{E}_1$	$\#\mathsf{P}$	$\#\mathsf{E}_1$	$\#\mathsf{E}_1$	$\#\mathsf{E}_2$	$\#\mathsf{M}$	$\#\mathsf{P}$	$\#\mathsf{E}_1$	$\#\mathsf{M}$	$\#\mathsf{P}$	$\#\mathsf{I}$
Zhao et al. [31]	$(2L+\mu)$	0	1	$(\mu+1)$	μ	$(L+\mu)$	μ	0	$L+2\mu+n+1$	$\mu+1$	1
Acharya et al. [32]	$(2L+\mu+4)$	0	1	$2L+\mu+3$	1	$(2L+\mu)$	1	$L+1$	$2L+1$	2	$L+1$
Le et al. [34]	$(\mu+1)(L+1)$	μ	1	2	μ	μ	μ	2	$3L+1$	2	2
Acharya et al. [33]-I	$2L$	0	2	2	2μ	$L+\mu$	2μ	1	$L+1$	4	1
Acharya et al. [33]-II	$(6L+2)$	0	2	3	2	$2L+\mu$	2μ	2	$L+1$	4	1
Ours	5	0	3	2	3μ	3μ	3μ	0	4	3	1

L = total number of users of the system, μ = total number of messages, $\#\mathsf{E}_1$ = number of exponentiation in $\mathbb{G}_1$ or $\mathbb{G}_2$, $\#\mathsf{E}_2$ = number of exponentiation in $\mathbb{G}_T$, $\#\mathsf{P}$ = number of pairing operations, $\#\mathsf{M}$ = number of multiplication in $\mathbb{G}_1$ or $\mathbb{G}_2$ or $\mathbb{G}_T$, $\#\mathsf{I}$ = number of inversion in $\mathbb{G}_T$.

Chapter 6

Implementation Details

This section analyzes and implements our proposed IB-MCBE scheme and the existing public-key MCBE protocols [32]–[34]. The experimental setup included a Dell Laptop functioning as the PKGC and broadcaster, equipped with an AMD A9-9400 Radeon processor, 12GB memory, and running on a 64-bit Ubuntu 20.04.3LTS Linux operating system with a 3.36.8 GNOME version. We utilized a Raspberry Pi 4 Model B Rev 1.4 with 8GB memory, and 1.5GHz 64-bit quad-core ARM Cortex-A72 processor and a POCO M4 Pro Android smartphone featuring 6GB memory, and 2.05GHz octa-core Mediatek Helio G96 processor for the decryptor component as IoT smart devices. To implement the algorithms for all protocols (including ours), we employed a type A supersingular elliptic curve $y = x^3+x$ with field size of 512 bits using the standard Pairing-Based Cryptography (PBC) library (version 0.5.14) [41]. In the case of executing the Decrypt algorithm in the POCO M4 Pro Android smartphone, we utilized an application called Termux. To gain a better understanding of the algorithm's effectiveness in real-life scenarios, we separately implemented the decryption algorithm Decrypt on both the Raspberry Pi 4 Model B Rev 1.4 and POCO M4 Pro Android smartphone. This approach allowed us to evaluate the algorithm's performance and efficiency across different IoT devices.

We discuss the implementation details of each public-key MCBEs constructions (including ours) [32]–[34] in **Tab. 6.2**, **6.3**, **6.4** and graphical representation is shown in **Fig. 6.1a**, **6.1b**, **6.1c**, **6.1d**. We have implemented the protocols as mentioned earlier using the total number of users $L = 1024, 2048, 4096, 8192, 16384, 32768$ with the following number of subscribed users $N = 256$, 512, 1024, 2048, 4096, 8192, respectively. Additionally, during our implementation of Encrypt algorithm, we have assigned $\mu = 4$ to represent the number of messages.

We also implemented the communication bandwidth and the storage overhead of our proposed scheme using different values of L and N that are shown in **Tab.** 6.1. We emphasize that the system requires 96 bytes (=0.096 kB) and 48 bytes (=0.048 kB) respectively to store the master public-key and the user's secret-key. Consequently,

Table 6.1: Implementation details for overhead storage and communication details of our proposal (in bytes)

Total number of users	Number of subscribed users	|MPK|	$\|SK_{x_i}\|$	|CT|
1024 ($\mu = 4, n = 256$)	256 ($\mu = 4, n = 64$)	96	48	208
2048 ($\mu = 8, n = 256$)	512 ($\mu = 8, n = 64$)	96	48	400
4096 ($\mu = 8, n = 512$)	1024 ($\mu = 8, n = 128$)	96	48	400
8192 ($\mu = 16, n = 512$)	2048 ($\mu = 16, n = 128$)	96	48	784
16384 ($\mu = 16, n = 1024$)	4096 ($\mu = 16, n = 256$)	96	48	784
32768 ($\mu = 32, n = 1024$)	8192 ($\mu = 32, n = 256$)	96	48	1552

Table 6.2: The execution time in seconds for Setup & Extract algorithms

Total number of users	Acharya et al. [32]	Li et al. [34]	Acharya et al.- I [33]	Acharya et al.- II [33]	Our Scheme
1024	5.932068	23.554679	5.901974	8.891522	4.28500
2048	11.800568	49.862647	11.92252	17.644394	8.536211
4096	23.96875	99.972114	23.855458	36.992378	17.926811
8192	47.368596	203.289588	47.771504	75.866365	33.483667
16384	94.007581	419.761978	95.438624	153.626914	67.049074
32768	196.405467	843.224368	191.688744	296.092054	133.639953

Table 6.3: The execution time in seconds for Encrypt algorithm

Total number of users	Number of subscribed users	Acharya et al. [32]	Li et al. [34]	Acharya et al.- I [33]	Acharya et al.- II [33]	Our Scheme
1024	256	0.030238	0.003502	0.027933	0.034182	0.034479
2048	512	0.031047	0.003767	0.029721	0.035342	0.035259
4096	1024	0.033230	0.003801	0.33434	0.042302	0.036142
8192	2048	0.037449	0.004498	0.039798	0.049507	0.037141
16384	4096	0.045469	0.004880	0.053879	0.079202	0.036519
32768	8192	0.069079	0.005746	0.082489	0.093964	0.036995

Table 6.4: The execution time in seconds for Decrypt algorithm

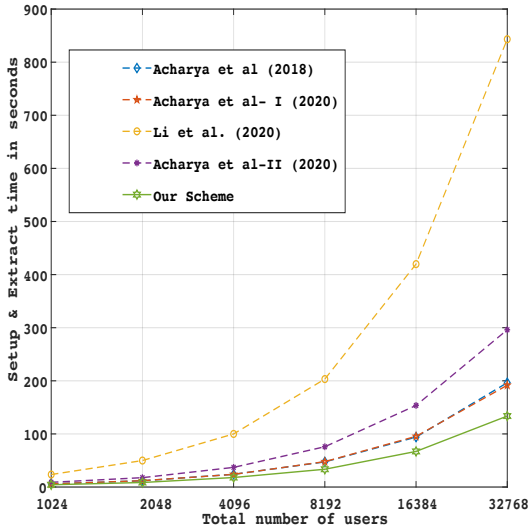
IoT device	Total number of users	Number of subscribed users	Acharya et al. [32]	Li et al. [34]	Acharya et al.- I [33]	Acharya et al.- II [33]	Our Scheme
Raspberry Pi 4	1024	256	0.016102	0.010151	0.010421	0.183333	0.006766
	2048	512	0.016062	0.010600	0.010633	0.193970	0.006801
	4096	1024	0.016079	0.011027	0.011258	0.221363	0.006780
	8192	2048	0.016066	0.12576	0.11896	0.244836	0.006783
	16384	4096	0.016076	0.014170	0.13635	3.961496	0.006789
	32768	8192	0.016085	0.016483	0.018920	7.730063	0.006793
POCO M4 Pro	1024	256	0.011387	0.007179	0.00737	0.129653	0.004785
	2048	512	0.011359	0.007496	0.00752	0.137176	0.00481
	4096	1024	0.011371	0.007798	0.007962	0.156548	0.004795
	8192	2048	0.011362	0.008894	0.008413	0.173148	0.004797
	16384	4096	0.011369	0.010021	0.009643	2.801575	0.004801
	32768	8192	0.011375	0.011657	0.01338	5.46671	0.004804

the storage overhead (i.e., master public-key size + user's secret key size) is constant (i.e., 0.144 kB) for any number of system users L. From **Tab.** 6.1, we realize that the communication bandwidth grows linearly with the number of channels (or messages) in the system.

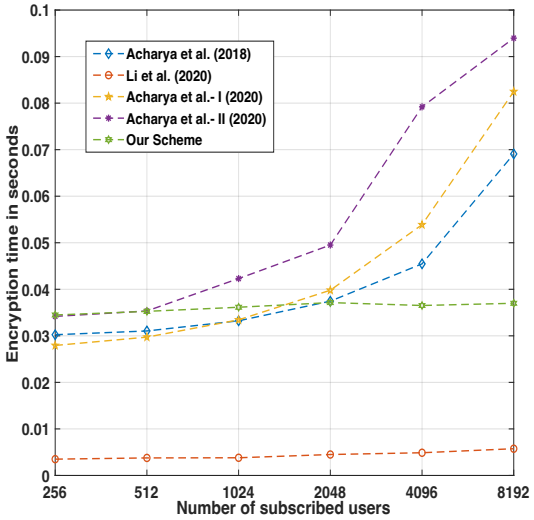
We have shown the execution time of Setup & Extract algorithms in **Tab.** **6.2** and analyzed it in **Fig.** **6.1a**. The execution time for Setup & Extract algorithms of our scheme is exceptionally less as opposed to the works of [32]–[34]. The graph presented in **Fig.** **6.1a** for our scheme is almost linear. According to the data from **Tab.** **6.2**, the slope of the line-graph of the works [32]–[34] in **Fig.** **6.1a** is significantly high compared to ours.

In **Tab.** **6.3**, we have illustrated the analysis of the execution time of Encrypt algorithm and given a pictorial presentation in **Fig.** **6.1b**. We emphasize that the execution of Encrypt algorithm for the works of [32], [33] are gradually increasing with respect to the number of subscribed users. The execution time for Encrypt algorithm of Li et al. [34] is in between 0.003 and 0.005 seconds, whereas the same for our scheme is in between 0.03 and 0.04 seconds.

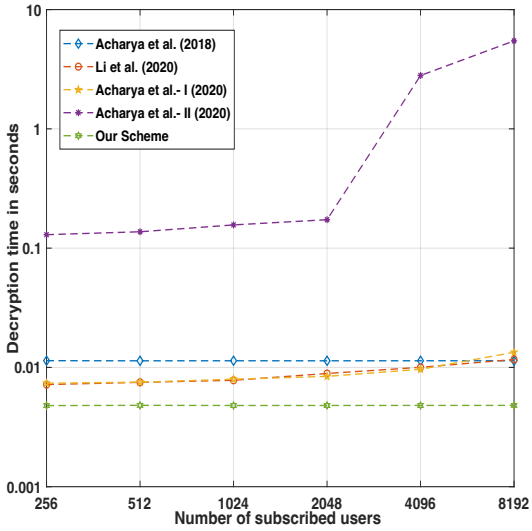
In our evaluation of the Decrypt algorithm, we implemented it on two distinct IoT devices and compared the results with the works of [32]–[34], as presented in **Tab.** **6.4** and **Fig.** 6.1c, and 6.1d. According to the **Tab.** **6.4**, we observed that the decryption time implemented on POCO M4 Pro for [32]–[34] are increasing with respect to the number of users in the subscribed set. For the Raspberry Pi 4 Model B Rev 1.4, which has limited computing capabilities, our system consistently achieved a decryption cost of 0.0068 seconds (rounded to four decimal places), which is better than [32]–[34]. Similarly, on the POCO M4 Pro Android smartphone, our system demonstrated an even faster decryption time of 0.0048 seconds (rounded to four decimal places), again comparable to the per-



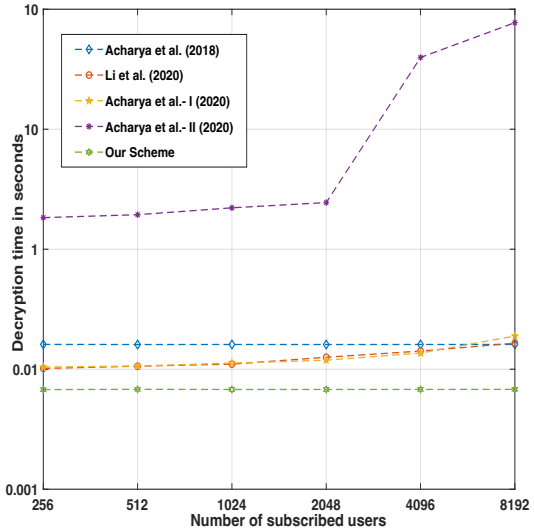
(a) The performance of Setup & Extract algorithm



(b) The performance of Encrypt algorithm



(c) The performance of Decrypt algorithm in Raspberry Pi 4



(d) The performance of Decrypt algorithm in POCO M4 Pro

Figure 6.1: Performance analysis of Setup, Extract, Encrypt, and Decrypt algorithms

formance of [32]–[34]. The results shown in **Tab.** 6.4 and **Fig.** 6.1c, and 6.1d indicate that our decryption algorithm is highly efficient, regardless of the resource constraints of the IoT devices used. Moreover, **Fig.** 6.1c, and 6.1d imply that the decryption time is almost constant (upto four decimal places) and independent of the number of subscribed users. These findings affirm the efficiency and effectiveness of our proposed algorithms within the IoT framework, showcasing their ability to handle a large number of users while maintaining a consistently low computational overhead.

Chapter 7

Conclusion and Future Work

We have proposed an adaptive secure identity-based MCBE protocol, which has achieved security under the standard asymmetric DBDH-3 assumption. Our design has a constant-size public parameter that is a plausible achievement as opposed to similar existing works. As a significant contribution, we have achieved the *first* adaptive IND-ID-CCA secure MCBE without any random oracles by eliminating the non-standard q-type security assumption. As far as our concern, none of the existing MCBE schemes has these achievements.

One limitation of our proposed protocol is that our scheme can't have anonymity properties. It is essential to maintain the users' privacy in the identity-based framework. On the other hand, the ciphertext size of our design is linear to the number of users; reducing the communication bandwidth (i.e., ciphertext size) is still an open problem in the MCBE framework.

Future Work

While our proposed IB-MCBE protocol offers strong theoretical guarantees and practical efficiency, several directions remain open for future exploration:

- **Post-Quantum Security:** The emergence of quantum computing necessitates the development of post-quantum secure MCBE schemes based on lattice or code-based assumptions.
- **Lightweight Cryptography for Edge Devices:** Although our scheme is optimized for IoT environments, further improvements can be made to reduce computational and storage overhead on ultra-resource-constrained devices.

Bibliography

[1] S. Fugkeaw, "Secure data sharing with efficient key update for industrial cloud-based access control," *IEEE Transactions on Services Computing*, 2023.

[2] Q. Huang, Y. Yang, and J. Fu, "Secure data group sharing and dissemination with attribute and time conditions in public cloud," *IEEE Transactions on Services Computing*, vol. 14, no. 4, pp. 1013–1025, 2018.

[3] X. Zhang, C. Xu, H. Wang, Y. Zhang, and S. Wang, "Fs-peks: Lattice-based forward secure public-key encryption with keyword search for cloud-assisted industrial internet of things," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1019–1032, 2019.

[4] M. Ma, D. He, N. Kumar, K. Choo, and J. Chen, "Certificateless searchable public key encryption scheme for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 759–767, 2017.

[5] H. Zhong, S. Zhang, J. Cui, L. Wei, and L. Liu, "Broadcast encryption scheme for v2i communication in vanets," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 3, pp. 2749–2760, 2021.

[6] W. Lee and S. Hwang, "High throughput implementation of post-quantum key encapsulation and decapsulation on gpu for internet of things applications," *IEEE Transactions on Services Computing*, vol. 15, no. 6, pp. 3275–3288, 2022.

[7] L. Sha, F. Xiao, H. Huang, Y. Chen, and R. Wang, "Catching escapers: A detection method for advanced persistent escapers in industry internet of things based on identity-based broadcast encryption (ibbe)," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 18, no. 3, pp. 1–25, 2019.

[8] J. Lai, F. Guo, W. Susilo, X. Huang, P. Jiang, and F. Zhang, "Data access control in cloud computing: Flexible and receiver extendable," *IEEE Transactions on Services Computing*, vol. 15, no. 5, pp. 2658–2670, 2021.

[9] C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Toward secure and dependable storage services in cloud computing," *IEEE Transactions on Services Computing*, vol. 5, no. 2, pp. 220–232, 2011.

[10] F. Rezaeibagha, Y. Mu, K. Huang, L. Chen, and L. Zhang, "Authenticable additive homomorphic scheme and its application for mec-based iot," *IEEE Transactions on Services Computing*, 2022.

[11] C. Ge, Z. Liu, J. Xia, and L. Fang, "Revocable identity-based broadcast proxy re-encryption for data sharing in clouds," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1214–1226, 2019.

[12] X. Yi, "Security of bertino-shang-wagstaff time-bound hierarchical key management scheme for secure broadcasting," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 2, pp. 303–304, 2010.

[13] X. Liu, G. Yang, Y. Mu, and R. Deng, "Multi-user verifiable searchable symmetric encryption for cloud storage," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 6, pp. 1322–1332, 2018.

[14] L. Liu, Y. Zhang, and X. Li, "Keyd: Secure key-deduplication with identity-based broadcast encryption," *IEEE Transactions on Cloud Computing*, vol. 9, no. 2, pp. 670–681, 2018.

[15] J. Shen, T. Miao, J. Lai, X. Chen, J. Li, and S. Yu, "Ims: An identity-based many-to-many subscription scheme with efficient key management for wireless broadcast systems," *IEEE Transactions on Services Computing*, vol. 15, no. 3, pp. 1707–1719, 2020.

[16] A. Fiat and M. Naor, "Broadcast encryption," in *Annual International Cryptology Conference*, Springer, 1993, pp. 480–491.

[17] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in *Annual International Cryptology Conference*, Springer, 2005, pp. 258–275.

[18] Y. Dodis and N. Fazio, "Public key broadcast encryption for stateless receivers," in *ACM Workshop on Digital Rights Management*, Springer, 2002, pp. 61–80.

[19] M. Mandal, "Privacy-preserving fully anonymous ciphertext policy attribute-based broadcast encryption with constant-size secret keys and fast decryption," *Journal of Information Security and Applications*, vol. 55, p. 102 666, 2020.

[20] Q. Wu, B. Qin, L. Zhang, J. Domingo-Ferrer, O. Farràs, and J. Manjón, "Contributory broadcast encryption with efficient encryption and short ciphertexts," *IEEE Transactions on Computers*, vol. 65, no. 2, pp. 466–479, 2016.

[21] L. Chen, J. Li, and Y. Zhang, "Anonymous certificate-based broadcast encryption with personalized messages," *IEEE Transactions on Broadcasting*, vol. 66, no. 4, pp. 867–881, 2020.

[22] D. Phan, D. Pointcheval, S. Shahandashti, and M. Strefler, "Adaptive cca broadcast encryption with constant-size secret keys and ciphertexts," *International Journal of Information Security*, vol. 12, pp. 251–265, 2013.

[23] C. Gritti, W. Susilo, T. Plantard, K. Liang, and D. Wong, "Broadcast encryption with dealership," *International Journal of Information Security*, vol. 15, pp. 271–283, 2016.

[24] W. Liu, J. Liu, Q. Wu, B. Qin, and Y. Li, "Practical chosen-ciphertext secure hierarchical identity-based broadcast encryption," *International Journal of Information Security*, vol. 15, pp. 35–50, 2016.

[25] P. Darco and A. Perez del Pozo, "Toward tracing and revoking schemes secure against collusion and any form of secret information leakage," *International Journal of Information Security*, vol. 12, pp. 1–17, 2013.

[26] M. Naor and B. Pinkas, "Efficient trace and revoke schemes," *International Journal of Information Security*, vol. 9, pp. 411–424, 2010.

[27] C. Delerablée, "Identity-based broadcast encryption with constant size ciphertexts and private keys," in *International Conference on the Theory and Application of Cryptology and Information Security*, Springer, 2007, pp. 200–215.

[28] R. Sakai and J. Furukawa, "Identity-based broadcast encryption," *IACR Cryptol. ePrint Arch.*, vol. 2007, p. 217, 2007.

[29] J. Kim, W. Susilo, M. Au, and J. Seberry, "Adaptively secure identity-based broadcast encryption with a constant-sized ciphertext," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 679–693, 2015.

[30] D. Phan, D. Pointcheval, and V. Trinh, "Multi-channel broadcast encryption," in *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, 2013, pp. 277–286.

[31] X. Zhao and H. Li, "Improvement on a multi-channel broadcast encryption scheme," in *Applied Mechanics and Materials*, vol. 427, Trans Tech Publ, 2013, pp. 2163–2169.

[32] K. Acharya and R. Dutta, "Constructions of secure multi-channel broadcast encryption schemes in public key framework," in *Cryptology and Network Security (CANS 2018)*, Springer, 2018, pp. 495–515.

[33] K. Acharya, "Secure and efficient public key multi-channel broadcast encryption schemes," *Journal of Information Security and Applications*, vol. 51, p. 102 436, 2020.

[34] M. Le, V. Tran, V. Trinh, *et al.*, "Compacting ciphertext in multi-channel broadcast encryption and attribute-based encryption," *Theoretical Computer Science*, vol. 804, pp. 219–235, 2020.

[35] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. CRC Press, 2020.

[36] M. Chase and S. Meiklejohn, "Déjà q: Using dual systems to revisit q-type assumptions," in *EUROCRYPT 2014*, Springer, 2014, pp. 622–639.

[37] B. Waters, "Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions," in *Annual International Cryptology Conference*, Springer, 2009, pp. 619–636.

[38] S. Chatterjee, D. Hankerson, and A. Menezes, "On the efficiency and security of pairing-based protocols in the type 1 and type 4 settings," in *International Workshop on the Arithmetic of Finite Fields*, Springer, 2010, pp. 114–134.

[39] D. Naor, M. Naor, and J. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Annual International Cryptology Conference*, Springer, 2001, pp. 41–62.

[40] I. Kim, S. Hwang, W. Susilo, J. Baek, and J. Kim, "Efficient anonymous multi-group broadcast encryption," in *International Conference on Applied Cryptography and Network Security*, Springer, 2020, pp. 251–270.

[41] B. Lynn, *Pbc library: The pairing-based cryptography library (version 0.5.14)*, Available at `https://crypto.stanford.edu/pbc/`, 2013.