Ph.D. Thesis

博士論文

A Study on Practical Information Retrieval Systems and Formalization of Their Security Models Considering Diverse Privacy Requirements

（多様なプライバシー要件を考慮した情報検索システムの効率化と
その安全性モデルに関する研究）

Junichiro Hayata

林田 淳一郎

Acknowledgement

Firstly, I would like to express my gratitude to my supervisor, Professor Kanta Matsuura, for his guidance during my graduate school days. I have learned a lot from him, including approaching research and preparing myself as a researcher. I would also like to thank the chief and deputy examiners, Professor Yoichi Sato, Shuichi Sakai, Shinya Sugiura, and Rie Yamaguchi, for their helpful comments and feedback on my research and this thesis.

I want to thank the many present and past Matsuura-lab members, including the secretaries and visitors of the meeting, for many valuable discussions. I was able to discuss and learn my research topic, cryptography, and various other aspects of information security.

I want to thank the researchers at Cyber-Physical Security Research Center Advanced Cryptography Research Team at AIST and many other study group participants (Shin-Akarui-Angou-Benkyou-Kai). Mainly, I would like to thank Fuyuki Kitagawa, Yusuke Sakai, Jacob Schuldt, and Goichiro Hanaoka for their attentive guidance, suggestions, and discussion.

Finally, I would like to thank my family for a great deal of support.

Abstract

With the spread of the Internet, there are more and more opportunities for people to search for information on databases. In addition, there are still many cases where information is leaked from databases due to the complexity of information retrieval systems. For the secure operation of such complex information retrieval systems, rigorous security notions that capture realistic attacks are required. In this thesis, we show two types of results on the security of information retrieval systems. Specifically, we deal with the security of information retrieval systems when the elements stored in the database are plaintexts and when they are encrypted.

First, we show the results of public-key encryption with keyword search (PEKS) in Chapters 3 and 4. PEKS allows us to perform a keyword search on encrypted data without decrypting ciphertexts. For the future practical use of PEKS, it is crucial to analyze the construction of PEKS schemes. In addition, it is also essential to add more functionalities to the PEKS schemes to support more applications. In doing so, it is necessary to formalize the security for more complex systems rigorously. In Chapter 3, we show the generic construction of an anonymous key-policy attribute-based encryption scheme from the PEKS scheme. In Chapter 4, we review the existing definitions of security against replayable chosen ciphertext attacks to provide more rigorous definitions of security and clarify the relationships among them.

Next, we show the results of private information retrieval (PIR) in Chapters 5 and 6. In these chapters, unlike the case of PEKS, we are considering a setting where the data is not encrypted. In such a setting, we may want to keep the contents of the client's query secret from the database server as a security requirement, and PIR allows us to do so. In Chapter 5, we introduce a new security notion called query indistinguishability for PIR schemes supporting basic range queries on one-dimensional databases and give constructions of the schemes satisfying query indistinguishability. In Chapter 6, we extend the result of Chapter 5 and construct the PIR scheme supporting multi-dimensional range queries. This concept of query indistinguishability can be applied to schemes that support more complex queries and is essential when considering the practical use of PIR.

Even though information retrieval systems have become more complex, the elements stored in the database are basically classified into plaintext or encrypted cases. Therefore, this research is expected to become a foundation of the security of information retrieval systems that will become more complex in the future.

Contents

Chapter 1

Introduction

1.1 Background and Motivation

In recent years, with the spread of the Internet, much information has been exchanged through the Internet. In addition, there are more and more opportunities for people to access remote databases. Some databases contain public data, such as stock price data, while others contain sensitive information, such as personal information. Therefore, a secure operation is strongly required depending on the type of database. However, there have been many incidents of information leakage from databases. For example, hundreds of millions of voter registration records [1], login credentials to classified systems [2], and sensitive medical records [3] have been leaked from databases. One of the reasons behind the constant leakage of information, as mentioned above, is presumably the complexity of information retrieval systems [4]. For example, taking the cloud services, various companies such as Amazon Web Services, Microsoft Azure, and Google Cloud Platform are currently providing cloud services, and we can use various functionalities. Since these services are used for more than simple information storage, we need to be more careful about security.

What helps us in our rigorous security discussion is the notion of provable security, cultivated in cryptography. Provable security is a framework for the rigorous formulation of the security of encryption schemes and cryptographic protocols. More specifically, it allows us to determine whether the scheme is secure or not within a defined model under the assumption that it is difficult to solve some specific mathematically intractable problems. While the absence of proof of security for a given cryptographic protocol does not directly imply that it is insecure, but the notion of provable security can be used to eliminate arguments about heuristic security. In order to formalize the security, we need a model of the requirement of the security (or security goal) we wish to show, an adversary model, and a formal definition of the mathematically intractable problem such as factoring problem, discrete log problem, and learning with errors problem. When we give proof of security, we need to show that no efficient algorithm breaks the security with non-negligible probability if the assumption of the hardness of the mathematical problem holds. In other words, if there exists a probabilistic polynomial-time algorithm that breaks the security of the scheme with non-negligible probability, we can show that the algorithm can be used to solve the mathematically intractable problem known in reality.

In this thesis, we formulate the security of complex information retrieval systems using the above notion of provable security. The possible settings can be classified into the following two categories, each of which requires different security considerations:

(i) Data is stored in encrypted form in the database: When a client outsources his data to the cloud, it is desirable to encrypt the data to prevent information leakage to the cloud servers or external attackers. However, simply encrypting the data is inconvenient because it makes it impossible to perform a keyword search.

(ii) Data is stored in plaintext form in the database: Data such as stock price can be accessed by anyone, and there is no need to encrypt. However, clients who want to retrieve data from the database may want to hide the contents of their queries. For example, it is conceivable that a stock price database server may obtain information about clients' investment strategies by checking the contents of their queries.

For setting (i) above, it is known that searches over encrypted data can be performed by using a technique called searchable encryption. More specifically, if we use searchable encryption, the information about the query conditions being searched is not leaked to the servers. In addition, we can hide the information of the keyword stored in the database by encrypting them. However, searchable encryption has not yet been put to practical use, and theoretical analysis and research on its practical use are still underway.

As for setting (ii), it is known that a technique called private information retrieval (PIR) can be used to perform secure searches. Using PIR, the client's access pattern, i.e., the information about which element in the database the client is accessing, can be kept secret from the server. Since the initial proposal of the PIR scheme, research on reducing the amount of communication between the client and server and the computation on the server-side has continued. However, it has not yet reached a level where it can be put to practical use. In addition, most of the research has focused on constructing the PIR schemes that support only simple queries, such as retrieving only one element of a database. Thus, there is still a lack of research on constructing the schemes that support flexible queries we use in our daily lives.

In this thesis, we aim at the practical application of the above-mentioned searchable encryption and PIR. When considering practical applications, it is necessary to consider increasingly complex systems, including clouds, such as those mentioned at the beginning of this section. Even if a scheme is proven to be secure in a simple system model, it does not necessarily mean that it is secure when used in a complex system. Since the security proof is only proof that the scheme is secure within a defined range, it is necessary to be more careful when considering complex systems. Therefore, the challenge is to formulate rigorous adversary models that capture realistic attacks on complex systems.

1.2 Outline and Summary of This Thesis

In this thesis, we show two types of results to address the above problems. In Chapters 3 and 4, we consider the case where elements stored in the databases are encrypted. In Chapters 5 and 6, we consider the case where data is stored in plaintext form in the databases.

- In Chapter 2, we explain the basic notations and cryptographic primitives which are used in this thesis.

- Public-key encryption with keyword search (PEKS) is the public-key type searchable encryption. It allows us to do keyword searches over encrypted data. In Chapter 3, we show our generic construction of key-policy attribute-based encryption (KP-ABE) scheme whose access structure is specified by monotone Boolean formula from the PEKS scheme whose search condition is specified by monotone Boolean formula. We also prove that the KP-ABE scheme constructed from our generic construction satisfies IND-CPA and IND-ANO-CPA if the underlying PEKS scheme satisfies IND-CKA. This result provides the guidelines for constructing the schemes in the future. Also, our result implies that such PEKS always requires high computational/communication costs and strong mathematical assumptions corresponding to ABE's.

- In Chapter 4, we consider proxy re-encryption with keyword search (PRES) that combines the features of PEKS and proxy re-encryption (PRE). With PRES, we can think of broader applications, such as more flexible mail routing services and keyword searches on encrypted data. While PRES is more valuable than PEKS, we need to be careful when formalizing the security because the system is becoming more complex. In fact, with the addition of PRE functionality, we need to think more rigorously about non-malleability. However, the problem is that the non-malleability against the adversary model, called replayable chosen ciphertext attack (RCCA), has not been formulated in a way that captures realistic attacks. Therefore, we give more rigorous definitions of non-malleability against RCCA. More specifically, we give simulation-based and game-based formulations as well as existing formulations of non-malleability. In addition, we give proofs of the relationship among security notions we define and the existing security notions.

- In Chapter 5, we consider the PIR schemes supporting basic range queries on the databases and give the definitions of security. Few studies have focused on constructing schemes supporting flexible queries such as range queries in existing PIR research. In addition, although there are studies that propose schemes supporting range queries, no rigorous security model has been given. We show that there are cases where existing schemes supporting range queries leak information. In order to prevent such information leakage, it is essential to formalize rigorous security models. Therefore, we formalize three security models for schemes supporting range queries. The notion of query indistinguishability defined in this chapter is a new security concept, and it is a security model that captures more realistic attacks than existing ones. We give two construction of the schemes supporting range queries that satisfy query indistinguishability.

- In Chapter 6, we extend the results of Chapter 5. In Chapter 5, we define security notions and construction of the PIR schemes supporting basic range queries. However, the databases we are considering in Chapter 5 are a simple one-dimensional array, but the databases we use in real life are much more complex. Thus, we need PIR schemes that can be used on more complex databases if we consider the practical use of PIR. As a first step towards practical use, we give security models for PIR schemes supporting multi-dimensional range queries on multi-dimensional databases. Also, we propose the construction of the PIR schemes supporting multi-dimensional range queries.

- In Chapter 7, we conclude the contributions of this thesis and describe the prospects that can be considered from this thesis.

Chapter 2

Preliminaries

In this chapter, we review the basic notation, definitions, and cryptographic primitives which are used in this thesis.

2.1 Basic Notations

We denote probabilistic polynomial time algorithm by PPTA, and for algorithm A, we denote the procedure that A is given input a and outputs b by $b \leftarrow A(a)$. We denote $\|$ as a concatenation operation, and for a set $\mathcal{S}$, we denote the cardinality of $\mathcal{S}$ by $\|\mathcal{S}\|$. In addition, we use the notation $\overrightarrow{x}$ for vectors and denote a vector with all elements being $\perp$ by $\overrightarrow{\perp}$.

Negligible Function: A function $\epsilon(k) : \mathbb{N} \to \mathbb{R}$ is negligible if for all positive polynomials p and all sufficiently large $k \in \mathbb{N}$, we have $f(k) < 1/p(k)$.

Difference Lemma: Let A, B and E be events. If $\Pr[A \wedge \neg E] = \Pr[B \wedge \neg E]$, then it holds that $|\Pr[A] - \Pr[B]| \leq \Pr[E]$.

Proof.

$$\begin{aligned}
&|\Pr[A] - \Pr[B]| \\
&= |\Pr[A \wedge E] + \Pr[A \wedge \neg E] - \Pr[B \wedge E] - \Pr[B \wedge \neg E]| \\
&= |\Pr[A \wedge E] - \Pr[B \wedge E]| \\
&\leq \Pr[E]
\end{aligned}$$

□

Game-hopping Proof: Game-hopping proof is a method for proving the security of the cryptographic protocols by transforming the games. The proof by game transformation is performed as follows: First, let game 0 be the security game defined we wish to prove. Then, we denote the advantage of the adversary in this game by $\text{Adv}_0(k)$. Next, consider game 1, which is a modification of game 0, and denote the adversary's advantage in this game by $\text{Adv}_1(k)$. In the same way, consider game $i + 1$, which is a modification of game i ($i = 0, 1, \ldots$), and denote the advantage of the adversary in this game by $\text{Adv}_{i+1}(k)$.

In this way, the game transformations are repeated one after another until the final game n is obtained. This final game is assumed to be such that its advantage can be easily shown that $\mathrm{Adv}_n(k)$ is zero.

The overview of the security proofs using this game-hopping is to prove $|\mathrm{Adv}_i(k) - \mathrm{Adv}_{i+1}(k)| < \epsilon(k)$ for all $i = 0, 1, \ldots, n-1$. If all these are shown, then it holds that $|\mathrm{Adv}_0(k) - \mathrm{Adv}_n(k)| \leq \sum_{i=0}^{n-1} |\mathrm{Adv}_i(k) - \mathrm{Adv}_{i+1}(k)| < \epsilon(k)$. In other words, we can show that this protocol is secure.

Definition 1. *For a set of variables $\boldsymbol{S}$, we call a Boolean formula that consists of logical disjunctions and logical conjunctions of each element in $\boldsymbol{S}$ as a monotone Boolean formula and denote it by $Q_{\boldsymbol{S}}$ (or simply Q). When given a monotone Boolean formula $Q_{\boldsymbol{S}}$ and a set of variables $\boldsymbol{W}$, we assign the value for each element in $Q_{\boldsymbol{S}}$ by the following rule: $x = 1$ if $x \in \boldsymbol{W}$ and $x = 0$ otherwise. We denote the output value of $Q_{\boldsymbol{S}}$ for $\boldsymbol{W}$ by $Q_{\boldsymbol{S}}(\boldsymbol{W}) \in \{0,1\}$. If $Q_{\boldsymbol{S}}(W) = 1$, then we say $\boldsymbol{W}$ satisfies $Q_{\boldsymbol{S}}$ (or $Q_{\boldsymbol{S}}$ is satisfied by $\boldsymbol{W}$).*

Definition 2. *For a set of variables $\boldsymbol{W}$, and a bit $b \in \{0,1\}$ we define $\boldsymbol{W}|_b$ as $\boldsymbol{W}|_b := \{w \| b \mid w \in \boldsymbol{W}\}$.*

Definition 3. *For a monotone Boolean formula $\mathbb{A}$, we denote by $\mathbb{A}|_b$ the Boolean formula that is constructed by replacing each variable w' in $\mathbb{A}$ with $w' \| b$.*

2.2 One-Way Function (OWF)

A One-way function (OWF) is one of the most fundamental primitives in cryptography. Informally, OWF is a function that is easy to compute on every input but hard to invert the image of a random input.

Let $f : \{0,1\}^n \rightarrow \{0,1\}^m$ be a polynomial-time computable function. f is one-way function if for any PPTA $\mathcal{A}$,

$$\Pr[x \leftarrow \{0,1\}^n; y = f(x) : \mathcal{A}(1^n, y) \in f^{-1}(f(x))]$$

is negligible.

2.3 Pseudorandom generator (PRG)

A pseudorandom generator (PRG) is a deterministic function that takes a short random input string and generates an output string longer than the input string that is indistinguishable from a truly random string.

Let $G : \{0,1\}^s \rightarrow \{0,1\}^\ell$ be a polynomial-time computable function, where $\ell > s$. We say G is PRG if for any PPTA $\mathcal{A}$,

$$|\Pr[r \leftarrow \{0,1\}^s : \mathcal{A}(G(r)) = 1] - \Pr[r' \leftarrow \{0,1\}^\ell : \mathcal{A}(r') = 1]|$$

is negligible.

2.4 Pseudorandom Function (PRF)

Informally speaking, the pseudorandom function (PRF) is a keyed function that it is indistinguishable from random function when a random key is chosen.

Let $\mathcal{K}$ be key space, $\{0,1\}^n$ be domain, and $\{0,1\}^m$ be range. A keyed function $F : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^m$ is PRF if for any PPTA $\mathcal{A}$,

$$|\Pr[k \leftarrow \mathcal{K} : \mathcal{A}^{F(k,\cdot)}(1^\lambda) = 1] - \Pr[f \leftarrow \mathsf{Funs}(\{0,1\}^n, \{0,1\}^m) : \mathcal{A}^{f(\cdot)}(1^\lambda) = 1]|$$

is negligible, where Funs is a set of all functions whose domain and range are $\{0,1\}^n$ and $\{0,1\}^m$, respectively.

2.5 Collision Resistant Hash Function (CRHF)

A collision resistant hash function (CRHF) is a hash function such that any PPTA cannot find a pair of input values where the output values are the same, but the input values are different.

A hash function $h : \{0,1\}^* \to \{0,1\}^m$ is collision resistant if for any PPTA $\mathcal{A}$,

$$\mathrm{Adv}^{\mathrm{CRHF}}_{\mathcal{A},h}(k) = \Pr[(x, x') \leftarrow \mathcal{A} : x \neq x' \text{ and } h(x) = h(x')]$$

is negligible.

2.6 Bilinear Map

Let $\mathbb{G}$ and $\mathbb{G}_T$ be groups of order q for some large prime q. We say that a map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is bilinear if e satisfies the following properties:

1. Efficient computability: There is an efficient algorithm to compute $e(g_1, g_2)$ for any $g_1, g_2 \in \mathbb{G}$.

2. Bilinearity: $e(g_1^a, g_2^b) = e(g, g)^{ab}$ for all $g_1, g_2 \in \mathbb{G}$ and all $a, b \in \mathbb{Z}$

3. Non-degeneracy: $e(g, g)$ is a generator of $\mathbb{G}_T$ iff $g \in \mathbb{G}$ is a generator of $\mathbb{G}$.

Let $\mathcal{G}$ be an algorithm that takes security parameter 1^λ as input, and outputs the description of a bilinear group $G = (q, \mathbb{G}, \mathbb{G}_T, g, e)$, where g is a generator of $\mathbb{G}$.

Computational Bilinear Diffie-Hellman (CBDH) Assumption: Let $\mathbb{G}$ and $\mathbb{G}_T$ be groups of order q for some large prime q, and e be a bilinear map. The Computational Bilinear Diffie-Hellman(CBDH) assumption states that for any PPTA $\mathcal{A}$,

$$\mathrm{Adv}^{\mathrm{CBDH}}_{\mathcal{A}}(\lambda) = \Pr[G = (q, \mathbb{G}, \mathbb{G}_T, g, e) \leftarrow \mathcal{G}; a, b, c \leftarrow \mathbb{Z}_q : e(g, g)^{abc} \leftarrow \mathcal{A}(1^\lambda, G, g^a, g^b, g^c)]$$

is negligible.

Bilinear maps have been used to construct a variety of cryptographic primitives such as IBE and PEKS, which will be introduced in Chapter 3.

2.7 Public-Key Encryption (PKE)

Diffie and Hellman proposed the concept of public-key encryption (PKE) [5]. PKE does not require prior key sharing and solves the disadvantage of symmetric key encryption, which requires two parties to communicate to share the same secret key in advance.

Then, we define public key encryption (PKE). In this thesis, we consider PKE schemes whose plaintext space is binary, that is, $\{0,1\}^\ell$, where ℓ is a polynomial of the security parameter.

Definition 4 (Public key encryption)**.** *A PKE scheme Σ is a tuple $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ of PPT algorithms. Below, let the message space of Σ be $\{0,1\}^\ell$, where ℓ is a polynomial of the security parameter.*

- *The key generation algorithm Gen, given a security parameter 1^λ, outputs a public key pk and a secret key sk.*
- *The encryption algorithm Enc, given a public key pk and message $m \in \{0,1\}^\ell$, outputs a ciphertext c.*
- *The decryption algorithm Dec, given a secret key sk and ciphertext c, outputs a plaintext $\tilde{m} \in \{\perp\} \cup \{0,1\}^\ell$.*

Correctness *We require $\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m$ for every $m \in \{0,1\}^\ell$ and $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$.*

2.7.1 Security

Secrecy and non-malleability are standard security requirements for PKE, and both have been formulated in simulation-based and game-based manners.

- **Game-based definition of security**: We formalize the security of a cryptographic protocol as a game between an adversary and a challenger. We define the protocol as secure if the advantage determined by the adversary's probability of winning the game is negligible. The game-based formalization is designed to make it easy to handle. Therefore, the security proof of a protocol is usually based on the game-based definition.
- **Simulation-based definition of security**: In the simulation-based definition of security, a protocol is defined as secure if any PPTA can not distinguish between a natural situation in which the protocol is used (the real world) and a simulation of the real world in a corresponding ideally secure situation (the ideal world). The simulation-based definition aims to clearly show the meaning of the definition by describing the ideal world. However, the security proof under the simulation-based definition is more complicated than the game-based one.

As for the adversary models, chosen plaintext attack (CPA), non-adaptive chosen ciphertext attack (CCA1), and adaptive chosen ciphertext attack (CCA2) are the basic models:

- **Chosen Plaintext Attack (CPA)**: Before and after receiving the target ciphertext c, the adversary can obtain ciphertexts for plaintexts of his choice.
- **Non-Adaptive Chosen Ciphertext Attack (CCA1)**: In addition to the attacks in CPA, the adversary can send ciphertexts of his choice to a decryption oracle that returns the decryption results before receiving the target ciphertext c.
- **Adaptive Chosen Ciphertext Attack (CCA2)**: In addition to the attacks in CPA, the adversary can send ciphertexts of his choice to a decryption oracle that returns the decryption results before and after receiving the target ciphertext c.

In the following of this thesis, we often refer to CCA2 as CCA.

Then, we give the definitions of secrecy for PKE. Secrecy guarantees that any PPTA can not obtain the information about the plaintext from the ciphertext.

Semantic Security: At first, we give the simulation-based definition of secrecy for PKE, which is called semantic security. We consider the following experiments SS-ATK-b ($b \in \{0, 1\}$, ATK$\in$ {CPA,CCA1,CCA2}):

```
$\underline{\mathrm{Exp}^{\text{SS-ATK-0}}_{\Sigma,\mathcal{A},h,f}(\lambda)}$
$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$;
$(\mathcal{M}, st_1) \leftarrow \mathcal{A}_1^{\mathcal{O}_1}(pk)$;
$m \leftarrow \mathcal{M}$;
$c^* \leftarrow \mathsf{Enc}(pk, m)$;
$v \leftarrow \mathcal{A}_2^{\mathcal{O}_2}(c^*, h(m), st_1)$;
if $v = f(m)$, then $\beta := 1$
else $\beta := 0$
output  $(\mathcal{M}, \beta)$
```

```
$\underline{\mathrm{Exp}^{\text{SS-ATK-1}}_{\Sigma,\mathcal{S},h,f}(\lambda)}$
$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$;
$(\mathcal{M}, st_1) \leftarrow \mathcal{S}_1(pk)$;
$m \leftarrow \mathcal{M}$;
$c^* \leftarrow \mathsf{Enc}(pk, m)$;
$v \leftarrow \mathcal{S}_2(h(m), st_1)$;
if $v = f(m)$, then $\beta := 1$
else $\beta := 0$
output  $(\mathcal{M}, \beta)$
```

where,

- ATK=CPA: $\mathcal{O}_1 = \mathcal{O}_2 = \phi$
- ATK=CCA1: $\mathcal{O}_1(c) = \mathsf{Dec}(sk, c), \mathcal{O}_2 = \phi$
- ATK-CCA2: $\mathcal{O}_1(c) = \mathsf{Dec}(sk, c), \mathcal{O}_2(c) = \mathsf{Dec}(sk, c)$

In the above two experiments, $\mathcal{M}$ is a distribution over the plaintext space. In addition to this, the function h in the above definition formalize the prior knowledge of the adversary. With this h, the definition states that even if the adversary has a priori knowledge about plaintext, the adversary cannot do more than attacks based on that a priori knowledge.

We define the advantage as

$$\begin{aligned}\mathsf{Adv}^{\text{SS-ATK}}_{\Sigma,\mathcal{A},\mathcal{S},\mathcal{D},h,f}(\lambda) :=| \Pr[\mathcal{D}(\mathsf{Exp}^{\text{SS-ATK-0}}_{\Sigma,\mathcal{A},h,f}(\lambda)) \rightarrow 1] \\ - \Pr[\mathcal{D}(\mathsf{Exp}^{\text{SS-ATK-1}}_{\Sigma,\mathcal{S},h,f}(\lambda)) \rightarrow 1]|.\end{aligned}$$

Definition 5 (SS-ATK security)**.** *We say that Σ is SS-ATK (ATK$\in$ { CPA,CCA1,CCA2 } secure if for any polynomial time computable function h and f, and for any pair of PPTAs $\mathcal{A}$, there exists a simulator $\mathcal{S}$ such that $\mathsf{Adv}^{\text{SS-ATK}}_{\Sigma,\mathcal{A},\mathcal{S},\mathcal{D},h,f}(\lambda)$ is negligible for any PPTA $\mathcal{D}$.*

Indistinguishability: Then, we give the game-based definition of secrecy for PKE, which is called indistinguishability. We consider the following experiments IND-ATK-b ($b \in \{0,1\}$, ATK$\in$ {CPA,CCA1,CCA2}):

```
$\underline{\mathrm{Exp}^{\text{IND-ATK-}b}_{\Sigma,\mathcal{A}}(\lambda)}$
$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$;
$(m_0, m_1, st_1) \leftarrow \mathcal{A}_1^{\mathcal{O}_1}(pk)$;
$c^* \leftarrow \mathsf{Enc}(pk, m_b)$;
$b' \leftarrow \mathcal{A}_2^{\mathcal{O}_2}(c^*, st_1)$;
output $b'$
```

where,

- ATK=CPA: $\mathcal{O}_1 = \mathcal{O}_2 = \phi$
- ATK=CCA1: $\mathcal{O}_1(c) = \mathsf{Dec}(sk, c), \mathcal{O}_2 = \phi$
- ATK-CCA2: $\mathcal{O}_1(c) = \mathsf{Dec}(sk, c), \mathcal{O}_2(c) = \mathsf{Dec}(sk, c)$

We define the advantage as

$$\begin{aligned}\mathsf{Adv}^{\text{IND-ATK}}_{\Sigma,\mathcal{A}}(\lambda) :=&|\Pr[\mathsf{Exp}^{\text{IND-ATK-0}}_{\Sigma,\mathcal{A}}(\lambda) \to 1] \\ &- \Pr[\mathsf{Exp}^{\text{IND-ATK-1}}_{\Sigma,\mathcal{A}}(\lambda) \to 1]|.\end{aligned}$$

Definition 6 (IND-ATK security)**.** *We say that Σ is IND-ATK (ATK$\in$ {CPA,CCA1,CCA2}) secure if $\mathsf{Adv}^{\textit{IND-RCCA}}_{\Sigma,\mathcal{A}}(\lambda)$ is negligible for any pair of PPTAs $\mathcal{A}$.*

Then, we give the definitions of non-malleability for PKE. Non-malleability guarantees that any PPTA can not make malicious modifications to the ciphertexts.

Simulation-based Non-Malleability: We give the the simulation-based definition of non-malleability for PKE. We consider the following experiments SNM-ATK-b ($b \in \{0,1\}$, ATK$\in$ {CPA,CCA1,CCA2}):

```
$\underline{\mathrm{Exp}^{\text{SNM-ATK-0}}_{\Sigma,\mathcal{A},h}(\lambda)}$
$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$;
$(\mathcal{M}, st_1) \leftarrow \mathcal{A}_1^{\mathcal{O}_1}(pk)$;
$m \leftarrow \mathcal{M}$;
$c^* \leftarrow \mathsf{Enc}(pk, m)$;
$(c_1, \dots, c_n, st_2) \leftarrow \mathcal{A}_2^{\mathcal{O}_2}(c^*, h(m), st_1)$;
for $i = 1$ to $n$
  $d_i := \mathsf{Dec}(sk, c_i)$
output $(\mathcal{M}, m, d_1, \dots, d_n, st_2)$
```

```
$\underline{\mathrm{Exp}^{\text{SNM-ATK-1}}_{\Sigma,\mathcal{S},h}(\lambda)}$
$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$;
$(\mathcal{M}, st_1) \leftarrow \mathcal{S}_1(pk)$;
$m \leftarrow \mathcal{M}$;

$(c_1, \dots, c_n, st_2) \leftarrow \mathcal{S}_2(h(m), st_1)$;
for $i = 1$ to $n$
  $d_i := \mathsf{Dec}(sk, c_i)$
output $(\mathcal{M}, m, d_1, \dots, d_n, st_2)$
```

where,

- ATK=CPA: $\mathcal{O}_1 = \mathcal{O}_2 = \phi$
- ATK=CCA1: $\mathcal{O}_1(c) = \mathsf{Dec}(sk, c), \mathcal{O}_2 = \phi$

- ATK-CCA2: $\mathcal{O}_1(c) = \mathsf{Dec}(sk, c), \mathcal{O}_2(c) = \mathsf{Dec}(sk, c)$

In the above two experiments, $\mathcal{M}$ is a distribution over the plaintext space. In addition to this, the function h in the above definition formalize the prior knowledge of the adversary. With this h, the definition states that even if the adversary has a priori knowledge about plaintext, the adversary cannot do more than attacks based on that a priori knowledge.

We define the advantage as

$$\begin{aligned}\mathsf{Adv}^{\text{SNM-ATK}}_{\Sigma,\mathcal{A},\mathcal{S},\mathcal{D},h}(\lambda) :=& |\Pr[\mathcal{D}(\mathsf{Exp}^{\text{SNM-ATK-0}}_{\Sigma,\mathcal{A},h}(\lambda)) \to 1] \\ & - \Pr[\mathcal{D}(\mathsf{Exp}^{\text{SNM-ATK-1}}_{\Sigma,\mathcal{S},h}(\lambda)) \to 1]|.\end{aligned}$$

Definition 7 (SNM-ATK security)**.** *We say that Σ is SNM-ATK ($ATK \in \{CPA, CCA1, CCA2\}$) secure if for any polynomial time computable function h and for any pair of PPTAs $\mathcal{A}$, there exists a pair of PPTAs $\mathcal{S}$ such that $\mathsf{Adv}^{SNM\text{-}ATK}_{\Sigma,\mathcal{A},\mathcal{S},\mathcal{D},h}(\lambda)$ is negligible for any PPTA $\mathcal{D}$.*

Indistinguishability-based Non-Malleability: Then we give the definition of indistinguishability-based non-malleability. We consider the following experiments INM-ATK-b ($b \in \{0, 1\}$, ATK$\in$ {CPA,CCA1,CCA2}):

```
$\underline{\mathrm{Exp}^{\text{INM-ATK-}b}_{\Sigma,\mathcal{A}}(\lambda)}$
$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$;
$(m_0, m_1, st_1) \leftarrow \mathcal{A}_1^{\mathcal{O}_1}(pk)$;
$c^* \leftarrow \mathsf{Enc}(pk, m_b)$;
$(c_1, \ldots, c_n, st_2) \leftarrow \mathcal{A}_2^{\mathcal{O}_2}(c^*, st_1)$;
for $i = 1$ to $n$
  $d_i := \mathcal{O}_2(c_i)$
$b' \leftarrow \mathcal{A}_3(d_1, \ldots, d_n, st_2)$;
output $b'$
```

where,

- ATK=CPA: $\mathcal{O}_1 = \mathcal{O}_2 = \phi$
- ATK=CCA1: $\mathcal{O}_1(c) = \mathsf{Dec}(sk, c), \mathcal{O}_2 = \phi$
- ATK-CCA2: $\mathcal{O}_1(c) = \mathsf{Dec}(sk, c), \mathcal{O}_2(c) = \mathsf{Dec}(sk, c)$

We define the advantage as

$$\begin{aligned}\mathsf{Adv}^{\text{INM-ATK}}_{\Sigma,\mathcal{A}}(\lambda) :=& |\Pr[\mathsf{Exp}^{\text{INM-ATK-0}}_{\Sigma,\mathcal{A}}(\lambda) \to 1] \\ & - \Pr[\mathsf{Exp}^{\text{INM-ATK-1}}_{\Sigma,\mathcal{A}}(\lambda) \to 1]|.\end{aligned}$$

Definition 8 (INM-ATK security)**.** *We say that Σ is INM-ATK ($ATK \in \{CPA, CCA1, CCA2\}$) secure if $\mathsf{Adv}^{INM\text{-}ATK}_{\Sigma,\mathcal{A}}(\lambda)$ is negligible for any triple of PPTAs $\mathcal{A}$.*

Although Diffie and Hellman proposed the concept of PKE, they did not give a concrete construction of the PKE scheme. Later, Rivest et al. proposed a concrete construction of the PKE scheme called RSA encryption scheme [6]. Various constructions such as Rabin encryption scheme [7] and Elgamal encryption scheme [8] have been proposed since then. However, the PKE schemes proposed in the early days, such as the RSA encryption scheme above, do not satisfy even the relatively weak security property called IND-CPA security. Therefore, the construction of PKE schemes that satisfy stronger security has been studied. Some PKE schemes have been proposed that satisfy the strong security notion IND-CCA2, such as RSA-OAEP [9, 10] and the Cramer-Shoup encryption scheme [11].

2.8 Key-Policy Attribute-Based Encryption (KP-ABE)

Attribute-based encryption (ABE) is an encryption scheme that allows flexible access control. (See also Chapter 3 for more information about ABE). There are two types of ABE, key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE). In this thesis, we focus on KP-ABE and consider the schemes that can specify a monotone Boolean formula as a policy. Regarding security, we consider indistinguishability against chosen plaintext attack (IND-CPA) and anonymity. IND-CPA guarantees that the information about the plaintext is kept secret from the ciphertext, and anonymity ensures that the information about the attributes used to encrypt the plaintext is kept secret from the ciphertext.

2.8.1 Model

A KP-ABE scheme consists of the following four polynomial time algorithms:

1. $\mathsf{Setup}(1^\lambda) \to (pp, msk)$: The setup algorithm takes as an input a security parameter 1^λ, and outputs the pair (pp, msk) of a public parameter and a master secret key.

2. $\mathsf{KeyGen}(pp, msk, \mathbb{A}) \to dk_{\mathbb{A}}$: The key generation algorithm takes as inputs a public parameter pp, a master secret key msk and a monotone Boolean formula $\mathbb{A}$, and outputs a secret key $dk_{\mathbb{A}}$ for $\mathbb{A}$.

3. $\mathsf{Enc}(pp, m, \boldsymbol{S}) \to C_{\boldsymbol{S}}$: The encryption algorithm takes as inputs a public parameter pp, a plaintext m and a set of attributes $\boldsymbol{S}$, and outputs a ciphertext $C_{\boldsymbol{S}}$.

4. $\mathsf{Dec}(pp, dk_{\mathbb{A}}, C_{\boldsymbol{S}}) \to m/\perp$: The decryption algorithm takes as inputs a public parameter pp, a secret key $dk_{\mathbb{A}}$ for a monotone Boolean formula $\mathbb{A}$ and a ciphertext $C_{\boldsymbol{S}}$, and outputs plaintext m or the rejection symbol $\perp$.

Correctness For correctness, we require the following: For any $\lambda \in \mathbb{N}$, any $(msk, pp) \leftarrow \mathsf{Setup}(1^\lambda)$, any monotone Boolean formula $\mathbb{A}$, any $dk_{\mathbb{A}} \leftarrow \mathrm{KeyGen}(pp, msk, \mathbb{A})$, any attribute set S which satisfies $\mathbb{A}$, any plaintext m, and any $C_{\boldsymbol{S}} \leftarrow \mathsf{Enc}(pp, m, \boldsymbol{S})$, it always holds that $\mathsf{Dec}(pp, dk_{\mathbb{A}}, C_{\boldsymbol{S}}) = 1$.

2.8.2 Security Definitions

Indistinguishability IND-CPA (indistinguishability against chosen plaintext attack) security ensures that the information about the plaintext is kept secret from the ciphertext when we consider PPTA. IND-CPA security with respect to a KP-ABE scheme $\Sigma_{KP\text{-}ABE}$ =(Setup,KeyGen,Enc,Dec) is defined by the following game between an adversary

$\mathcal{A}$ and a challenger $\mathcal{CH}$ who manages the list of pairs of each monotone Boolean formula queried to key generation oracle and the secret key generated for the monotone Boolean formula:

Setup: $\mathcal{CH}$ runs $\mathsf{Setup}(1^\lambda)$ and obtains (pp, msk). Then, $\mathcal{CH}$ sends pp to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ can adaptively use the following key generation oracle.

- **Key generation oracle:** $\mathcal{A}$ issues a monotone Boolean formula $\mathbb{A}$ to $\mathcal{CH}$. If $\mathcal{CH}$ has already generated a secret key $dk_{\mathbb{A}}$ for $\mathbb{A}$, $\mathcal{CH}$ sends the secret key $dk_{\mathbb{A}}$ to $\mathcal{A}$. Otherwise, $\mathcal{CH}$ runs $\mathsf{KeyGen}(pp, msk, \mathbb{A})$ and obtains the output $dk_{\mathbb{A}}$, then $\mathcal{CH}$ sends $dk_{\mathbb{A}}$ to $\mathcal{A}$.

Challenge: $\mathcal{A}$ chooses two plaintexts m_0 and m_1, where $|m_0| = |m_1|$, and a target attribute set $\boldsymbol{S}^*$, where $\boldsymbol{S}^*$ satisfies $\mathbb{A}_i(\boldsymbol{S}^*) = 0$ $(i = 1, 2 \ldots)$ for all $\mathbb{A}_i$ issued in the phase 1, and after that, sends them to $\mathcal{CH}$. Then, $\mathcal{CH}$ runs $C \leftarrow \mathsf{Enc}(pp, m_b, \boldsymbol{S}^*)$, where a bit $b \in \{0, 1\}$ is chosen randomly, and sends the output C to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ can issue a key generation oracle query for $\mathbb{A}$ adaptively like phase 1, where $\mathbb{A}$ satisfies $\mathbb{A}(\boldsymbol{S}^*) = 0$.

Guess: $\mathcal{A}$ outputs a guess bit b'.

We say $\mathcal{A}$ succeeds if $b' = b$, and denote the probability that $\mathcal{A}$ succeeds by $\Pr_{\mathcal{A},\Sigma_{KP\text{-}ABE}}[\text{Succ}]$. In addition, we define the advantage of $\mathcal{A}$ by

$$\mathsf{Adv}^{\text{ind-cpa}}_{\mathcal{A},\Sigma_{KP\text{-}ABE}}(\lambda) = |\Pr_{\mathcal{A},\Sigma_{KP\text{-}ABE}}[\text{Succ}] - 1/2|.$$

Definition 9. *We say $\Sigma_{KP\text{-}ABE}$ satisfies IND-CPA security if $\mathsf{Adv}^{\textit{ind-cpa}}_{\mathcal{A},\Sigma_{KP\text{-}ABE}}(\lambda)$ is negligible for any PPTA $\mathcal{A}$.*

Anonymity We define IND-ANO-CPA security as anonymity that ensures that the attributes used to encrypt the plaintext are kept secret from the ciphertext. We define anonymity using a game, as we did in the definition of IND-CPA.

Then, we introduce the security notion IND-ANO-CPA. IND-ANO-CPA security with respect to a KP-ABE scheme $\Sigma_{KP\text{-}ABE}$ =(Setup,KeyGen,Enc,Dec) is defined by the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{CH}$ who manages the list of pairs of a monotone Boolean formula and a trapdoor:

Setup: $\mathcal{CH}$ runs $\mathsf{Setup}(1^\lambda)$ and obtains (pp, msk). Then, $\mathcal{CH}$ sends pp to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ can adaptively use the following key generation oracle.

- **Key generation oracle:** $\mathcal{A}$ issues a monotone Boolean formula $\mathbb{A}$ to $\mathcal{CH}$. If $\mathcal{CH}$ already has generated a secret key $dk_{\mathbb{A}}$ for $\mathbb{A}$, $\mathcal{CH}$ sends the secret key $dk_{\mathbb{A}}$ to $\mathcal{A}$. Otherwise, $\mathcal{CH}$ runs $\mathsf{KeyGen}(pp, msk, \mathbb{A})$ and obtains the output $dk_{\mathbb{A}}$, then $\mathcal{CH}$ sends $dk_{\mathbb{A}}$ to $\mathcal{A}$.

Challenge: $\mathcal{A}$ chooses a plaintext m and two sets of attributes $\boldsymbol{S}_0^*$ and $\boldsymbol{S}_1^*$, such that $\mathbb{A}_i(\boldsymbol{S}_0^*) = \mathbb{A}_i(\boldsymbol{S}_1^*) = 0 \vee \mathbb{A}_i(\boldsymbol{S}_0^*) = \mathbb{A}_i(\boldsymbol{S}_1^*) = 1$ $(i = 1, 2, \dots)$ for all $\mathbb{A}_i$ issued in the phase 1, and after that, sends them to $\mathcal{CH}$. Then, $\mathcal{CH}$ runs $\mathsf{Enc}(pp, m, \boldsymbol{S}_b^*)$, where a bit $b \in \{0, 1\}$ is chosen randomly, and sends the output C to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ can issue a monotone Boolean formula $\mathbb{A}$ as query to the key generation oracle adaptively like Phase 1, under the restriction that $\mathbb{A}$ satisfies $\mathbb{A}(\boldsymbol{S}_0^*) = \mathbb{A}(\boldsymbol{S}_1^*)$.

Guess: $\mathcal{A}$ outputs a guess bit b'.

We say $\mathcal{A}$ succeeds if $b' = b$, and denote the probability that $\mathcal{A}$ that by $\Pr_{\mathcal{A},\Sigma_{KP\text{-}ABE}}[\text{Succ}]$. In addition, we define the advantage of $\mathcal{A}$ by

$$\mathsf{Adv}^{\text{ind-ano-cpa}}_{\mathcal{A},\Sigma_{KP\text{-}ABE}}(\lambda) = |\Pr_{\mathcal{A},\Sigma_{KP\text{-}ABE}}[\text{Succ}] - 1/2|.$$

Definition 10. *We say $\Sigma_{KP\text{-}ABE}$ satisfies IND-ANO-CPA security if $\mathsf{Adv}^{\textit{ind-ano-cpa}}_{\mathcal{A},\Sigma_{KP\text{-}ABE}}(\lambda)$ is negligible for any PPTA $\mathcal{A}$.*

We note that this security notion is sometimes referred to as ciphertext attribute hiding.

2.9 Function Secret Sharing (FSS)

Function secret sharing (FSS) was proposed by Boyle et al. [12], and a FSS scheme provides a means to split a function f into separate evaluation keys, where each party's key enables him to efficiently generate a standard secret share of the evaluation $f(x)$ for any input x, and yet each key individually does not reveal information about which function f has been shared.

In this section, we define syntax, correctness, and a security model for FSS schemes. A FSS scheme is defined for a function family $\mathcal{F}$, and we denote the domain of $f \in \mathcal{F}$ by D_f.

The syntax of a FSS scheme is as follows.

Definition 11. *For $p \in \mathbb{N}$, $T \subseteq [p]$, a p-party, T-secure function secret sharing (FSS) scheme $\mathcal{FSS}$ with respect to function class $\mathcal{F}$, is a pair of PPTA (Gen, Eval) with the following syntax:*

$(k_1, \dots, k_p) \leftarrow \mathsf{Gen}(1^\lambda, f)$**:** *Key Generation algorithm* Gen *takes as input the security parameter 1^λ and function description $f \in \mathcal{F}$, and outputs a p-tuple of keys $(k_1, \dots, k_p)$.*

$y_i \leftarrow \mathsf{Eval}(i, k_i, x)$**:** *Evaluation algorithm* Eval *takes as input a party index $i \in [p]$, key k_i and input string $x \in D_f$, and outputs a value y_i, corresponding to the party's share of $f(x)$.*

Correctness and secrecy requirements are as follows:

Correctness: *For all $f \in \mathcal{F}, x \in D_f$,*

$$\Pr\left[(k_1, \dots, k_p) \leftarrow \mathsf{Gen}(1^\lambda, f) : \sum_{i=1}^{p} \mathsf{Eval}(i, k_i, x) = f(x)\right] = 1.$$

Security: *Consider the following indistinguishability experiment for an adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *and corrupted parties* $T \subset [p]$:

1: The adversary outputs $(f^0, f^1, st) \leftarrow \mathcal{A}_1(1^\lambda)$*, where* $f^0, f^1 \in \mathcal{F}$ *with* $D_{f^0} = D_{f^1}$*.*

2: The challenger samples $b \leftarrow \{0,1\}$ *and computes* $(k_1, \dots, k_p) \leftarrow \mathsf{Gen}(1^\lambda, f^b)$*.*

3: Given the keys for corrupted parties T*, the adversary outputs a guess* $b' \leftarrow \mathcal{A}_2((k_i)_{i \in T}, st)$*.*

Denote by

$$\mathsf{Adv}_{\mathcal{FSS}}(1^\lambda, \mathcal{A}) := |\Pr[b = b'] - 1/2|$$

the advantage of $\mathcal{A}$ *in guessing* b *in the above experiment, where the probability is taken over the randomness of the challenger and* $\mathcal{A}$*. We say the scheme* $(\mathsf{Gen}, \mathsf{Eval})$ *is* T*-secure if there exists a negligible function* ϵ *such that for all PPTA* $\mathcal{A}$*, it holds that* $\mathsf{Adv}(1^\lambda, \mathcal{A}) \leq \epsilon(\lambda)$*.*

Although it is possible to construct FSS for arbitrary functions, practical FSS protocols only exist for some restricted function families. For example, Boyle et al. [12, 13] proposed FSS schemes for point function and interval function. These take the following forms:

- Point functions f_a are defined as $f_a(x) = 1$ if $x = a$ or 0 otherwise.
- Interval functions are defined as $f_{a,b}(x) = 1$ if $a < x < b$ or 0 otherwise.

In addition to this, they showed a FSS construction for decision trees and it can be used for the FSS scheme for constant d-dimensional interval function: that is, functions $f(x_1, \dots, x_d)$ which evaluate to a selected nonzero value precisely when $a_i \leq x_i \leq b_i$ for some secret interval ranges $(a_i, b_i)_{i \in [d]}$. Regarding the size of the key of FSS scheme for such function, they proved following theorem.

Theorem 1 (Corollary 3.4 from [13]). *For* $d \in \mathbb{N}$ *there exists FSS for the class of* d*-dimensional intervals* $(a_i, b_i)_{i \in [d]}$ *with key size* $O(\lambda \cdot n^d)$*.*

Hereafter, for a function f, we denote by $(f_1, \dots, f_p) \leftarrow \mathsf{Gen}(1^\lambda, f)$ the function shares described by the keys $(k_1, \dots, k_\ell)$ generated by $\mathsf{Gen}(1^\lambda, f)$, and by $f_i(x)$ the output of $\mathsf{Eval}(i, k_i, \cdot)$.

Chapter 3

Generic Construction of Adaptively Secure Anonymous KP-ABE from Public-Key Encryption with Keyword Search

3.1 Introduction

3.1.1 Background and Motivation

With the development of cloud services, there have been more and more situations in which data is outsourced to external servers. From the viewpoint of security and privacy, the outsourced data should be encrypted. However, when the data stored in the server is encrypted, some functionalities are compromised. For example, it is difficult to determine which documents contain a particular keyword when retrieving documents from the server where the encrypted data is stored. One intuitive way to retrieve documents containing a specific keyword from encrypted data in an external server is to send the keyword to be retrieved and the decryption key to the server via secure channel. Although such a way can protect the data from third parties, the server can learn all the data that the client outsourced. Another intuitive way is that the client retrieves all encrypted data from the server, decrypts them, and searches the documents containing a specific keyword in the local environment. This approach can prevent information from being leaked to the server and third parties, but it requires a considerable amount of computation on the client-side.

Searchable encryption allows us to search encrypted data for particular keywords without decrypting ciphertexts. Therefore, the use of searchable encryption can guarantee security in situations such as the above. More specifically, secure search on encrypted data can be achieved by doing the following between the client and the server:

(i) Let $\mathbb{D} = \{D_1, \ldots, D_n\}$ be the data that the client wants to outsource to the server, and $KW_i = \{KW_{i,1}, \ldots, KW_{i,m}\}$ be the set of keywords contained in D_i. For each i, the client encrypts each of his data D_i using a standard encryption scheme, obtains a ciphertext C_i, and encrypts each keyword in KW_i using a searchable encryption scheme, and obtain $C_{KW_{i,1}}, \ldots, C_{KW_{i,m}}$. After that, the client sends the encrypted data $\mathbb{D}' = \{C'_1, \ldots, C'_n\}$ to the remote server, where $C'_i = (C_i, C_{KW_{i,1}}, \ldots, C_{KW_{i,m}})$.

(ii) When the client wants to retrieve the specific keywords data, he generates the trapdoor $Td_{KW'}$ for the keyword W'.

(iii) For each $i \in [n]$ and $j \in [m]$, the server tests whether the encrypted data contains the client's keywords to retrieve using the trapdoor $Td_{KW'}$ and searchable ciphertext $C_{KW_{i,j}}$.

Note that since the ciphertexts and the trapdoor hide the information about the keywords, the server can only learn whether each ciphertext is the data the client wants to retrieve or not.

The symmetric-key type, called symmetric-key searchable encryption (SSE), was firstly proposed by Song et al. [14]. The public-key type, called public-key encryption with keyword search (PEKS), was firstly proposed by Boneh et al. [15]. In addition to secure search on encrypted databases, PEKS is expected to be used for a wide range of applications such as flexible routing of e-mails [15]. SSE or PEKS schemes were also proposed in [16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26].

In recent years, there has been much research on cryptography with advanced functionality, which considers adding functionality to encryption schemes, such as searchable encryption. For example, identity-based encryption (IBE) [27, 28] is an extension of public-key encryption where any bit string can be used as a public key, and attribute-based encryption (ABE) [29] is an extension of IBE, which realizes flexible access control. In addition to this, various other types of cryptography with advanced functionality are studied, such as functional encryption [30], homomorphic encryption [31], and proxy re-encryption [32].

It has been proven that PEKS, capable of searching data by a single keyword, can be constructed from anonymous IBE [15]. Like the relationship between PEKS and anonymous IBE, PEKS that can specify a flexible search condition, such as logical disjunctions and logical conjunctions, can be constructed from anonymous ABE [33]. If such a generic construction is known, it can be used as a guideline for constructing encryption schemes. For example, since the generic construction from anonymous IBE to PEKS is known, if we want to consider the construction of the PEKS scheme, we can consider the construction of the anonymous IBE scheme. Therefore, clarifying the relationships among encryption schemes is one of the most critical problems. Although the generic construction from anonymous ABE scheme to PEKS scheme supporting logical disjunctions and logical conjunctions as search condition is known, it is believed that the opposite direction, that is, generic construction of such PEKS schemes from weaker cryptographic tools than ABE, is difficult. However, this intuition has not been rigorously justified.

3.1.2 Our Contribution

In this chapter, we rigorously justify this intuition by constructing Key-Policy ABE (KP-ABE) from PEKS for monotone Boolean formulas. Specifically, we prove that anonymous KP-ABE whose access structure is specified by a monotone Boolean formula can be generically constructed from PEKS whose search condition is specified by a monotone Boolean formula. We should note that our KP-ABE can deal with only plaintexts whose length is

1 bit. However, we show how to extend our result to deal with longer plaintext in Section 3.4. Refer to Figure 3.1.1 which describes some existing works which clarify a generic relationship between a PEKS whose search condition is specified by a single keyword or a monotone Boolean formula and an IBE or (KP-)ABE scheme whose access structure is specified by a monotone Boolean formula.

Existing Known Results
- IBE → PEKS for a single keyword [33] - KP-ABE for monotone Boolean formulas → PEKS for monotone Boolean formulas [34] - PEKS for a single keyword → IBE for 1 bit plaintexts [15]
Our Result
- PEKS for monotone Boolean formulas → KP-ABE for monotone Boolean formulas and 1 bit plaintexts

Figure 3.1.1: Existing and our results. A → B means that B can be generically constructed from A.

3.1.3 Main Difficulty

As we explained earlier, Boneh et al. [15] proposed a generic construction of IBE from PEKS. The main difficulty in their construction is that, while the PEKS encryption algorithm has only a single input, the keyword, the IBE encryption algorithm needs to take two inputs, the identity and the plaintext. To overcome this difficulty, Boneh et al. somehow encoded two objects, the identity ID and the plaintext $b \in \{0,1\}$, into a single object, the keyword $\boldsymbol{K}$. Fortunately, their encoding was very simple, and in fact was just to concatenate ID and b as $ID||b$ and to use it as a keyword. This works as the PEKS ciphertext hides the information on the keyword $ID||b$. Therefore, it also hides the information on the plaintext b. To decrypt the ciphertext, the receiver with identity ID will be issued with a pair of trapdoors for $ID||0$ and $ID||1$, which allow the receiver to decrypt the ciphertext by trying the two trapdoors one by one if the identities match.

However, it is not straightforward to extend their idea to the context of ABE and PEKS for Boolean formulas. Boneh et al.'s idea works because the identity ID and the plaintext b is non-structured objects, binary strings, and thus it was sufficient to simply concatenate them into a single binary string $ID||b$. In contrast, in PEKS for monotone Boolean formulas, for example, a set

$$\boldsymbol{S} = \{1100, 1010\}$$

will be associated with a ciphertext and a Boolean formula

$$\mathbb{A} = (1100 \vee 0101) \wedge 1010$$

will be associated with a trapdoor. We need to embed a plaintext $b \in \{0,1\}$ into the set $\boldsymbol{S}$ to obtain a new set $\boldsymbol{S}_b$ of keywords and also convert the Boolean formula $\mathbb{A}$ into Boolean formula (or a set of Boolean formulas) in such a way that (1) the receiver is able to extract

b if $\mathbb{S}$ satisfies $\mathbb{A}$ and (2) the receiver obtains no information on b if $\mathbb{S}$ does not satisfies $\mathbb{A}$. Since the formula $\mathbb{A}$ can be an arbitrary complex Boolean formula, this task is not very straightforward.

3.1.4 Related Work

Searchable encryption was firstly proposed by Song et al. [14], and they proposed a symmetric-key type scheme. After their research, many SSE schemes have been proposed. Curtmola et al. [35] defined the security for SSE and proposed the construction of an SSE scheme that is efficient in search time using the inverted index. Bost et al. [24] proposed an SSE scheme that supports index updates. Cash et al. [36] proposed a conjunctive keyword search scheme. Recently, Kamara et al. [37] proposed the SSE scheme supporting the Boolean search.

Public-key type searchable encryption was firstly proposed by Boneh et al. [15]. Zhang et al. [38] proposed the PEKS scheme supporting conjunctive keyword search. Boneh et al. [39] proposed the scheme supporting flexible queries such as subset queries. PEKS schemes were also proposed in [22, 23, 25, 26].

Shamir firstly proposed the notion of IBE, but he did not give a construction [27]. After that, Boneh et al. [28] firstly proposed the construction of the IBE scheme. ABE, an extension of IBE, was firstly proposed by Goyal [29], and they constructed a KP-ABE scheme. After their proposal, Bethencourt [40] proposed the construction of the CP-ABE scheme.

The relationship between encryption schemes is a common research theme. Regarding PEKS, relationships with several encryption primitives have also been investigated. Generic construction of PEKS with a single keyword as search condition from IBE was proposed by Abdalla et al. [33], and generic construction of IBE from PEKS with a single keyword as search condition was proposed by Boneh et al. [15]. In addition, the generic construction of PEKS, whose search condition is a monotone Boolean formula from ABE, was proposed by Han et al. [34]. However, it is not proven whether ABE can be constructed from PEKS. See Figure 3.1.1 for details.

3.1.5 Chapter Organization

The remainder of this chapter is organized as follows. In Section 3.2, we introduce the model of PEKS and its security definitions. In Section 3.3, we show our generic construction of KP-ABE from PEKS. In Section 3.4, we show the extension of our generic construction. In Section 3.5, we conclude this chapter.

3.2 Public-Key Encryption with Keyword Search (PEKS)

In this section, we review the model and the security definitions of PEKS. The concept of PEKS was introduced in [15], and that considers the single keyword search. In this chapter, we focus on multi-keyword PEKS whose search condition is specified by a monotone Boolean formula. Regarding security, we consider indistinguishability against chosen keyword attack (IND-CKA) and consistency. IND-CKA guarantees that no information is leaked from the ciphertext of keywords, and consistency ensures that unintended search results cannot be obtained when searching for keywords.

3.2.1 Model

In this thesis, we consider multi-keyword PEKS. The keyword encryption algorithm in this type of PEKS takes a *set* of keywords as input. Also, the trapdoor generation algorithm takes as input a monotone Boolean formula consisting of the logical disjunctions and logical conjunctions of keywords. We define a PEKS scheme to consist of the following four PPT algorithms:

1. $\mathsf{Gen}(1^\lambda) \to (pk, sk)$: The key generation algorithm takes as an input a security parameter 1^λ and outputs the pair (pk, sk) of a public key and a private key.
2. $\mathsf{PEKS}(pk, \boldsymbol{W}) \to C$: The keyword encryption algorithm takes as inputs a public key pk and a set of keywords $\boldsymbol{W}$, and outputs a ciphertext C.
3. $\mathsf{Td}(pk, sk, Q) \to td_Q$: The trapdoor generation algorithm takes as inputs a public key pk, a private key sk and a monotone Boolean formula Q, and outputs a trapdoor td_Q.
4. $\mathsf{Test}(pk, C, td_Q) \to 0/1$: The test algorithm takes as inputs a public key pk, a ciphertext C and a trapdoor td_Q, and outputs 0 or 1.

Correctness For correctness, we require the following: For any $\lambda \in \mathbb{N}$, any $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$, any set of keywords $\boldsymbol{W}$, any $C \leftarrow \mathsf{PEKS}(pk, \boldsymbol{W})$, any monotone Boolean formula Q that is satisfied by $\boldsymbol{W}$, and any $td_Q \leftarrow \mathsf{Td}(pk, sk, Q)$, it always holds that $\mathsf{Test}(pk, C, td_Q) = 1$.

3.2.2 Security Definitions

Consistency Consistency is a security notion defined in [33], that ensures that unintended search results cannot be obtained when searching for keywords. If we use a PEKS scheme that does not satisfy consistency in the mail service, mail routing errors will occur. This notion of consistency can be defined as follows.

Definition 12. *For a PEKS scheme $\Sigma_{PEKS} = ($Gen,PEKS,Td,Test$)$, if the following holds, then we say Σ_{PEKS} has perfect consistency: For any $\lambda \in \mathbb{N}$, any $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$, any set of keywords $\boldsymbol{W}$, any $C \leftarrow \mathsf{PEKS}(pk, \boldsymbol{W})$, any monotone Boolean formula Q' that is not satisfied by $\boldsymbol{W}$, any $td_{Q'} \leftarrow \mathsf{Td}(pk, sk, Q')$, it always holds that $\mathsf{Test}(pk, C, td_{Q'}) = 0$.*

Abdala et al. [33] defined three types of consistency, perfect, statistical, and computational consistency. The above defines perfect consistency, and it is the strongest notion of consistency. We say that a PEKS scheme has statistical consistency, if for any adversary $\mathcal{A}$, the probability that $\mathcal{A}$ outputs a set of keyword $\boldsymbol{W}$ and a monotone Boolean formula Q that is not satisfied by $\boldsymbol{W}$ such that $\mathsf{Test}(pk, C, td_Q) = 1$ is negligible, where $C \leftarrow \mathsf{PEKS}(pk, \boldsymbol{W})$ and $td_Q \leftarrow \mathsf{Td}(pk, sk, Q)$. We say that a PEKS scheme has computational consistency, if for any PPT adversary, the same condition as statistical one holds. According to [33], perfect consistency is difficult to achieve. We however require the underlying PEKS scheme to satisfy this notion for simplicity.

Indistinguishability IND-CKA (indistinguishability against chosen keyword attack) security ensures that any information about the keywords is not leaked from the ciphertexts. The IND-CKA security with respect to a PEKS scheme Σ_{PEKS} =(Gen,PEKS,Td, Test) is defined by using the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{CH}$ who manages the list of pairs of each monotone Boolean formula queried to trapdoor generation oracle and the trapdoor generated for the monotone Boolean formula:

Setup: $\mathcal{CH}$ runs $\mathsf{Gen}(1^\lambda)$ and obtains (pk, sk). Then, $\mathcal{CH}$ sends pk to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ can adaptively use the following trapdoor generation oracle.

- **Trapdoor generation oracle:** $\mathcal{A}$ issues a monotone Boolean formula Q to $\mathcal{CH}$. If $\mathcal{CH}$ has already generated a trapdoor td_Q for the monotone Boolean formula Q, then $\mathcal{CH}$ sends the trapdoor td_Q to $\mathcal{A}$. Otherwise, $\mathcal{CH}$ runs $\mathsf{Td}(pk, sk, Q)$ and obtains td_Q, then $\mathcal{CH}$ sends td_Q to $\mathcal{A}$.

Challenge: $\mathcal{A}$ chooses two sets of keywords $\boldsymbol{W}_0$ and $\boldsymbol{W}_1$ such that $|\boldsymbol{W}_0| = |\boldsymbol{W}_1|$ and for any Q_i $(i = 1, 2 \ldots)$ issued in phase 1, $Q_i(\boldsymbol{W}_0) = Q_i(\boldsymbol{W}_1)$ holds, and after that, sends them to $\mathcal{CH}$. Then, $\mathcal{CH}$ runs $\mathsf{PEKS}(pk, \boldsymbol{W}_b)$, where a bit $b \in \{0, 1\}$ is chosen randomly, and sends the output C to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ can adaptively use the trapdoor generation oracle in the same manner except that queried a monotone Boolean formula Q should satisfy $Q(\boldsymbol{W}_0) = Q(\boldsymbol{W}_1)$.

Guess: $\mathcal{A}$ outputs a guess bit b'.

We say $\mathcal{A}$ succeeds if $b' = b$, and denote the probability that $\mathcal{A}$ succeeds by $\Pr_{\mathcal{A},\Sigma_{PEKS}}[\text{Succ}]$. In addition, we define the advantage of $\mathcal{A}$ by

$$\mathsf{Adv}^{\text{ind-cka}}_{\mathcal{A},\Sigma_{PEKS}}(\lambda) = |\Pr_{\mathcal{A},\Sigma_{PEKS}}[\text{Succ}] - 1/2|.$$

Definition 13. *We say Σ_{PEKS} satisfies IND-CKA security if $\mathsf{Adv}^{\textit{ind-cka}}_{\mathcal{A},\Sigma_{PEKS}}(\lambda)$ is negligible for any PPTA $\mathcal{A}$.*

3.2.3 Construction of PEKS Scheme by Boneh et al.

To show that we can construct the PEKS scheme, we introduce the proposed PEKS scheme by Boneh et al. [15]. They use the bilinear map to construct the scheme.

Let $\mathbb{G}$ and $\mathbb{G}_T$ be groups of prime order p, and $H_1 : \{0,1\}^* \to \mathbb{G}$ and $H_2 : \mathbb{G}_T \to \{0,1\}^{\log p}$ be CRHFs, where $(p, \mathbb{G}, \mathbb{G}_T, g, e) \leftarrow \mathcal{G}(1^\lambda)$. Then, we show their construction in Figure 3.2.1.

The trapdoor generation algorithm takes secret key sk and a keyword W as input since their scheme only supports a single keyword search. Note that their construction does not satisfy perfect consistency.

3.3 Generic Construction of KP-ABE from PEKS

Our generic construction is based on the generic construction of IBE from PEKS by Boneh et al. [15]. In their generic construction, when we encrypt a bit b for an ID, the keyword

```
Gen($1^\lambda$):
  $\alpha \leftarrow \mathbb{Z}_p^*$
  return $(pk, sk) := ((g, h := g^\alpha), \alpha)$
PEKS($pk, W$)
  $r \leftarrow \mathbb{Z}_p^*$
  Compute $t = e(H_1(W), h^r)$
  return $C := (g^r, H_2(t))$
Td($sk, W$)
  return $T_W := H_1(W)^\alpha \in \mathbb{G}$
Test($T_{W'}, C'$) = 1,
  $C'$ is parsed as $(A, B)$
  if $H_2(e(T_{W'}, A)) = B$,
    then return 1
  else
    then return 0
```

Figure 3.2.1: Construction of PEKS Scheme by Boneh et al.

encryption algorithm is used as follows: At first, consider the string concatenated with ID and the bit b (i.e., $ID\|b$) as a keyword. Then encrypt it using keyword encryption algorithm PEKS in underlying PEKS scheme. We use the idea of concatenating strings before encrypting.

3.3.1 The Construction

In Figure 3.3.1, we present our generic construction of a KP-ABE scheme $\Pi_{KP\text{-}ABE}$. We use a PEKS scheme Π_{PEKS} =(Gen, PEKS, Td, Test) as a building block. For the meaning of notations such as $\boldsymbol{S}|_b$, $\mathbb{A}|_0$ and $\mathbb{A}|_1$, the reader should see Definition 2 and Definition 3. We are considering a PEKS scheme that encrypts a set of keywords collectively (e.g., [41]), and which supports a monotone Boolean formula as an access structure. Our construction is generic and only makes black-box use of the PEKS.

Let us explain the intuition behind our construction. In our construction, when we encrypt a single bit $b \in \{0, 1\}$ with a set of attributes $\boldsymbol{S}$, and let the encryption algorithm of the KP-ABE encrypt a set of keywords $\boldsymbol{S}|_b$ using the PEKS encryption algorithm. In addition, when we generate a secret key for a monotone Boolean formula $\mathbb{A}$ in the KP-ABE, we compute two trapdoors $td_{\mathbb{A}|_0}$ and $td_{\mathbb{A}|_1}$ for $\mathbb{A}|_0$ and $\mathbb{A}|_1$ respectively, with the PEKS trapdoor generation algorithm. Then, the secret key sk for $\mathbb{A}$ is $sk := (td_{\mathbb{A}|_0}, td_{\mathbb{A}|_1})$. We use $td_{\mathbb{A}|_b}$ as a decryption key for decrypting a ciphertext of b in KP-ABE. Finally, we explain the decryption algorithm. Let C be a ciphertext of a single bit b' with a set of attribute S'. When we decrypt C with a secret key $dk_{\mathbb{A}'}$, we execute the following decryption procedure; At first, we parse $dk_{\mathbb{A}'}$ as $(td_{\mathbb{A}'|_0}, td_{\mathbb{A}'|_1})$. Then, we compute $\mathsf{Test}(td_{\mathbb{A}'|_0}, C)$ and $\mathsf{Test}(td_{\mathbb{A}'|_1}, C)$. Since LSB of all attributes in $\mathbb{A}|_0$ is 0, if $\mathsf{Test}(td_{\mathbb{A}'|_0}, C) = 1 \wedge \mathsf{Test}(td_{\mathbb{A}'|_1}, C) = 0$, we can conclude $\mathbb{A}'|(\boldsymbol{S}') = 1$ and C is a ciphertext of 0. Likewise, if $\mathsf{Test}(td_{\mathbb{A}'|_0}, C) = 0 \wedge \mathsf{Test}(td_{\mathbb{A}'|_1}, C) = 1$, we can conclude $\mathbb{A}'|(\boldsymbol{S}') = 1$ and C is a ciphertext of 1. In this way, using the Test algorithm, we obtain information about

```
$\mathsf{Setup}(1^\lambda)$:
  $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$
  return $(pp, msk) := (pk, sk)$
$\mathsf{Enc}(pp, b \in \{0,1\}, \boldsymbol{S})$:
  $C \leftarrow \mathsf{PEKS}(pp, \boldsymbol{S}|_b)$
  return $C$
$\mathsf{KeyGen}(pp, msk, \mathbb{A})$:
  $td_{\mathbb{A}|_0} \leftarrow \mathsf{Td}(msk, \mathbb{A}|_0)$
  $td_{\mathbb{A}|_1} \leftarrow \mathsf{Td}(msk, \mathbb{A}|_1)$
  return $dk_{\mathbb{A}} := (td_{\mathbb{A}|_0}, td_{\mathbb{A}|_1})$
$\mathsf{Dec}(pp, dk_{\mathbb{A}'}, C')$:
  $dk_{\mathbb{A}'}$ is parsed as $(td_{\mathbb{A}'|_0}, td_{\mathbb{A}'|_1})$
  if $\mathsf{Test}(td_{\mathbb{A}'|_0}, C) = 1 \wedge \mathsf{Test}(td_{\mathbb{A}'|_1}, C) = 0$,
    then return $0$
  else if $\mathsf{Test}(td_{\mathbb{A}'|_0}, C) = 0 \wedge \mathsf{Test}(td_{\mathbb{A}'|_1}, C) = 1$,
    then return $1$
  else
    then return $\perp$
```

Figure 3.3.1: Construction of $\Pi_{KP\text{-}ABE}$

b' only when $\boldsymbol{S}'$ satisfies $\mathbb{A}'$.

3.3.2 Security

We show that the KP-ABE constructed above is IND-CPA, IND-ANO-CPA and correct if the underlying PEKS is IND-CKA and perfectly consistent. Theorem 2 (resp. Theorem 3) guarantees that our KP-ABE scheme $\Pi_{KP\text{-}ABE}$ is IND-CPA (resp. IND-ANO-CPA) if the PEKS scheme Π_{PEKS} is IND-CKA. Theorem 4 guarantees that our KP-ABE is correct if the PEKS is correct and perfectly consistent.

Theorem 2. *$\Pi_{KP\text{-}ABE}$ is IND-CPA if Π_{PEKS} is IND-CKA.*

Proof. By using an adversary $\mathcal{A}$ that breaks the IND-CPA with respect to the KP-ABE, we can construct an adversary $\mathcal{A}'$ that breaks the IND-CKA with respect to the PEKS as follows:

Setup: $\mathcal{A}'$ obtains pk. Then, $\mathcal{A}'$ sends $pp := pk$ to $\mathcal{A}$.

Phase 1: $\mathcal{A}'$ responds to a query of secret key for $\mathbb{A}$ from $\mathcal{A}$ as follows:

- $\mathcal{A}'$ issues the query $\mathbb{A}|_0$ to the trapdoor generation oracle, then obtains $td_{\mathbb{A}|_0}$. Likewise, $\mathcal{A}'$ issues the query $\mathbb{A}|_1$, then obtains $td_{\mathbb{A}|_1}$. After that, $\mathcal{A}'$ sends $dk_{\mathbb{A}} := (td_{\mathbb{A}|_0}, td_{\mathbb{A}|_1})$ to $\mathcal{A}$.

Challenge: $\mathcal{A}$ outputs two plaintexts $m_0 = 0$, $m_1 = 1$ without loss of generality. In addition, $\mathcal{A}$ outputs a set of target attributes, we denote this by $\boldsymbol{S}^*$. Then, $\mathcal{A}'$ outputs $(\boldsymbol{S}^*|_0, \boldsymbol{S}^*|_1)$ to $\mathcal{CH}$ as the challenge keywords sets. After that, $\mathbb{A}'$ receives the challenge ciphertext C from $\mathcal{CH}$, then $\mathcal{A}'$ sends C to $\mathcal{A}$.

Phase 2: $\mathcal{A}'$ responds to a query for secret key from $\mathcal{A}$ in the same way as Phase 1.

Guess: Bit b' is output from $\mathcal{A}$, then $\mathcal{A}'$ outputs b'.

In the following, we will explain that $\mathcal{A}'$ provides a perfect simulation of the IND-CPA game, and submits no forbidden queries.

In the challenge phase, $\mathcal{A}$ outputs a target attribute set $\boldsymbol{S}^*$. Here, $\boldsymbol{S}^*$ satisfies $\mathbb{A}_i(\boldsymbol{S}^*) = 0 \quad (i = 1, 2, \ldots)$ for all monotone Boolean formulas issued in the phase 1, so it holds $\mathbb{A}_i|_0(\boldsymbol{S}^*|_0) = \mathbb{A}_i|_0(\boldsymbol{S}^*|_1) = 0$. Likewise, it holds $\mathbb{A}_i|_1(\boldsymbol{S}^*|_0) = \mathbb{A}_i|_1(\boldsymbol{S}^*|_1) = 0$, so $(\boldsymbol{S}^*|_0, \boldsymbol{S}^*|_1)$ are valid challenge keywords sets.

In the phase 2, when the secret key oracle query for $\mathbb{A}'$ is submitted from $\mathcal{A}$, $\mathcal{A}'$ needs to obtain the trapdoors $td_{\mathbb{A}'|_0}$ for $\mathbb{A}'|_0$ and $td_{\mathbb{A}'|_1}$ for $\mathbb{A}'|_1$ from $\mathcal{CH}$. Here, $\mathcal{A}$ can only issue queries for secret keys for the monotone Boolean formula $\mathbb{A}'$ satisfying $\mathbb{A}'(\boldsymbol{S}^*) = 0$. Therefore, $\mathbb{A}'|_0(\boldsymbol{S}^*|_0) = 0$. In addition, $\mathbb{A}'|_0$ is constructed from a monotone Boolean formula $\mathbb{A}'$ by replacing all elements w' in $\mathbb{A}'$ with $w'\|0$, so $\mathbb{A}'|_0$ is a monotone Boolean formula. Here, the LSBs of all the variables in $\mathbb{A}'|_0$ is 0, and the LSBs of all elements constructing $\boldsymbol{S}^*|_1$ is 1, and it holds $\mathbb{A}'|_0(\boldsymbol{S}^*|_1) = 0$. Therefore, it holds $\mathbb{A}'|_0(\boldsymbol{S}^*|_0) = \mathbb{A}'|_0(\boldsymbol{S}^*|_1)$. Likewise, it also holds $\mathbb{A}'|_1(\boldsymbol{S}^*|_0) = \mathbb{A}'|_1(\boldsymbol{S}^*|_1)$, and $\mathbb{A}'|_1$ is a monotone Boolean formula. Hence, $\mathcal{A}'$ can responds correctly to the query from $\mathcal{A}$.

We can construct an adversary $\mathcal{A}'$ in the above way, and the probability that $\mathcal{A}'$ breaks the security of IND-CPA for the KP-ABE scheme is exactly the same as the probability that $\mathcal{A}$ breaks the security of IND-CKA for PEKS. Therefore, if PEKS satisfies IND-CKA security, then KP-ABE satisfies IND-CPA security. □

Theorem 3. *$\Pi_{KP\text{-}ABE}$ is IND-ANO-CPA if Π_{PEKS} is IND-CKA.*

Proof. By using an adversary $\mathcal{A}$ that breaks the IND-ANO-CPA with respect to the KP-ABE, we can construct an adversary $\mathcal{A}'$ that breaks the IND-CKA with respect to the PEKS as follows:

Setup: $\mathcal{A}'$ obtains pk. $\mathcal{A}'$ sends $pp := pk$ to $\mathcal{A}$.

Phase 1: $\mathcal{A}'$ responds to a key generation query for $\mathbb{A}$ from $\mathcal{A}$ as follows:

- $\mathcal{A}'$ issues the query $\mathbb{A}|_0$ to the trapdoor generation oracle, then obtains $td_{\mathbb{A}|_0}$. Likewise, $\mathcal{A}'$ issues the query $\mathbb{A}|_1$, then obtains $td_{\mathbb{A}|_1}$. After that, $\mathcal{A}'$ sends $dk_{\mathbb{A}} := (td_{\mathbb{A}|_0}, td_{\mathbb{A}|_1})$ to $\mathcal{A}$.

Challenge: $\mathcal{A}$ outputs a plaintext $m \in \{0, 1\}$ and two sets of target attributes $\boldsymbol{S}_0^*$ and $\boldsymbol{S}_1^*$. Then, $\mathcal{A}'$ outputs $(\boldsymbol{S}_0^*|_m, \boldsymbol{S}_1^*|_m)$ to $\mathcal{CH}$ as the challenge sets of keywords. After that, $\mathbb{A}'$ receives the challenge ciphertext C from $\mathcal{CH}$, then $\mathcal{A}'$ sends C to $\mathcal{A}$.

Phase 2: $\mathcal{A}'$ responds to a key generation query from $\mathcal{A}$ in the same way as Phase 1.

Guess: Bit b' is output from $\mathcal{A}$, then $\mathcal{A}'$ outputs b'.

We explain that $\mathcal{A}'$ perfectly simulates the IND-ANO-CPA game, and submits no forbidden queries.

In the challenge phase, $\mathcal{A}$ outputs two target attributes sets $\boldsymbol{S}_0^*$ and $\boldsymbol{S}_1^*$. Here, $\boldsymbol{S}_0^*$ and $\boldsymbol{S}_1^*$ satisfy $\mathbb{A}_i(\boldsymbol{S}_0^*) = \mathbb{A}_i(\boldsymbol{S}_1^*) = 0 \quad (i = 1, 2, \ldots)$ for all monotone Boolean formulas in phase 1, so it holds $\mathbb{A}_i|_0(\boldsymbol{S}_0^*|_0) = \mathbb{A}_i|_0(\boldsymbol{S}_1^*|_0) = 0$. In addition, the LSBs of all the variables in $\mathbb{A}_i|_1$ is 1, so it holds $\mathbb{A}_i|_1(\boldsymbol{S}_0^*|_0) = \mathbb{A}_i|_1(\boldsymbol{S}_1^*|_0) = 0$. Likewise, it holds $\mathbb{A}_i|_0(\boldsymbol{S}_0^*|_1) = \mathbb{A}_i|_0(\boldsymbol{S}_1^*|_1) = 0$, and $\mathbb{A}_i|_1(\boldsymbol{S}_0^*|_1) = \mathbb{A}_i|_1(\boldsymbol{S}_1^*|_1)$, so $(\boldsymbol{S}_0^*|_m, \boldsymbol{S}_1^*|_m)$ for $m \in \{0, 1\}$ is not prohibited as challenge sets of keywords.

In the phase 2, when the key generation query for $\mathbb{A}'$ is submitted from $\mathcal{A}$, $\mathcal{A}'$ needs to obtain the trapdoors $td_{\mathbb{A}'|_0}$ for $\mathbb{A}'|_0$ and $td_{\mathbb{A}'|_1}$ for $\mathbb{A}'|_1$ from $\mathcal{CH}$. Here, $\mathcal{A}$ can only issue queries for secret keys for a monotone Boolean formula $\mathbb{A}'$ satisfying $\mathbb{A}'(\boldsymbol{S}_0^*) = \mathbb{A}'(\boldsymbol{S}_1^*) = 0$. So, in the case $m = 0$, it holds $\mathbb{A}'|_0(\boldsymbol{S}_0^*|_0) = \mathbb{A}'|_0(\boldsymbol{S}_1^*|_0)$. In addition, the LSBs of all the variables in $\mathbb{A}'|_1$ is 1, and LSB of all elements constructing $\boldsymbol{S}_0^*|_0$ and $\boldsymbol{S}_1^*|_0$ is 0, so it holds $\mathbb{A}'|_1(\boldsymbol{S}_0^*|_0) = \mathbb{A}'|_1(\boldsymbol{S}_1^*|_0) = 0$. Likewise, in case $m = 1$, it holds $\mathbb{A}'|_0(\boldsymbol{S}_0^*|_1) = \mathbb{A}'|_0(\boldsymbol{S}_1^*|_1) = 0$ and $\mathbb{A}'|_1(\boldsymbol{S}_0^*|_1) = \mathbb{A}'|_1(\boldsymbol{S}_1^*|_1) = 0$. Therefore, $\mathcal{A}'$ can response correctly to the query from $\mathcal{A}$.

We can construct an adversary $\mathcal{A}'$ in the above way, and the probability that $\mathcal{A}'$ breaks security of IND-ANO-CPA for the KP-ABE scheme is exactly the same as the probability that $\mathcal{A}$ breaks security of IND-CKA for PEKS. Therefore, if PEKS satisfies IND-CKA security, then KP-ABE satisfies IND-ANO-CPA security. □

Theorem 4. $\Pi_{\mathit{KP\text{-}ABE}}$ *is correct if* Π_{PEKS} *is correct and perfectly consistent.*

Proof. We can prove this theorem by proving following lemma: For any security parameter 1^λ, any $(pk, sk) \leftarrow \mathrm{Gen}(1^\lambda)$, any attribute set $\boldsymbol{S}$, any $m \in \{0, 1\}$, any $C \leftarrow \mathrm{PEKS}(pk, \boldsymbol{S}|_m)$, any monotone Boolean formula $\mathbb{A}$ which is satisfied by $\boldsymbol{S}$, any $td_{\mathbb{A}|_0} \leftarrow \mathrm{Td}(pk, sk, \mathbb{A}|_0)$, and any $td_{\mathbb{A}|_1} \leftarrow \mathrm{Td}(pk, sk, \mathbb{A}|_1)$, it holds that $1 \leftarrow \mathrm{Test}(pk, C, td_{\mathbb{A}|_m})$, and $0 \leftarrow \mathrm{Test}(pk, C, td_{\mathbb{A}|_{\bar{m}}})$, where $\bar{m} = 1 - m$.

The above lemma holds by the correctness and perfect consistency of the PEKS scheme Π_{PEKS} and the following fact: For any keywords set $\boldsymbol{S}$, any monotone Boolean formula $\mathbb{A}$ which is satisfied by $\boldsymbol{S}$, and any $m \in \{0, 1\}$, $\mathbb{A}|_m$ is satisfied by $\boldsymbol{S}|_m$, and $\mathbb{A}|_{\bar{m}}$ is not satisfied by $\boldsymbol{S}|_m$, where $\bar{m} = 1 - m$. □

3.4 Extension

In this section, we discuss an extension of our generic construction. KP-ABE scheme constructed from our generic construction can deal with plaintexts whose length is 1 bit. However, we can extend our result to construct KP-ABE schemes that can deal with longer plaintexts. In Figure 3.4.1, we show the construction of the KP-ABE scheme $\Pi'_{\mathit{KP\text{-}ABE}}$ whose plaintext space is 2 bits from the PEKS scheme $\Pi'_{PEKS} = (\mathsf{Gen}, \mathsf{PEKS}, \mathsf{Td}, \mathsf{Test})$, where $\boldsymbol{S}|_{b_0 b_1} := \{s \| b_0 b_1 \mid s \in \boldsymbol{S}\}$ and $\mathbb{A}|_{b_0 b_1}$ is the Boolean formula constructed by replacing each variable w' in $\mathbb{A}$ with $w' \| b_0 b_1$ for $b_0 b_1 \in \{0, 1\}^2$.

We can prove $\Pi'_{\mathit{KP\text{-}ABE}}$ satisfies IND-CPA and IND-ANO-CPA if Π'_{PEKS} satisfies IND-CKA in the same way as Theorem 2 and 3. Also, it is obvious $\Pi'_{\mathit{KP\text{-}ABE}}$ is correct if Π'_{PEKS} is perfectly consistent.

The idea of the above extension is to achieve secret key generation and decryption in KP-ABE by generating a trapdoor for each plaintext in plaintext space. By applying

```
$\mathsf{Setup}(1^\lambda)$:
  $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$
  return $(pp, msk) := (pk, sk)$
$\mathsf{Enc}(pp, b_0b_1 \in \{0,1\}^2, \boldsymbol{S})$:
  $C \leftarrow \mathsf{PEKS}(pp, \boldsymbol{S}|_{b_0b_1})$
  return $C$
$\mathsf{KeyGen}(pp, msk, \mathbb{A})$:
  $td_{\mathbb{A}|_{00}} \leftarrow \mathsf{Td}(msk, \mathbb{A}|_{00})$
  $td_{\mathbb{A}|_{01}} \leftarrow \mathsf{Td}(msk, \mathbb{A}|_{01})$
  $td_{\mathbb{A}|_{10}} \leftarrow \mathsf{Td}(msk, \mathbb{A}|_{10})$
  $td_{\mathbb{A}|_{11}} \leftarrow \mathsf{Td}(msk, \mathbb{A}|_{11})$
  return $dk_{\mathbb{A}} := (td_{\mathbb{A}|_{00}}, td_{\mathbb{A}|_{01}}, td_{\mathbb{A}|_{10}}, td_{\mathbb{A}|_{11}})$
$\mathsf{Dec}(pp, dk_{\mathbb{A}'}, C')$:
  $dk_{\mathbb{A}'}$ is parsed as $(td_{\mathbb{A}'|_0}, td_{\mathbb{A}'|_1})$
  if $\mathsf{Test}(td_{\mathbb{A}'|_{00}}, C) = 1 \wedge \mathsf{Test}(td_{\mathbb{A}'|_{01}}, C) = \mathsf{Test}(td_{\mathbb{A}'|_{10}}, C) = \mathsf{Test}(td_{\mathbb{A}'|_{11}}, C) = 0$,
    then return 00
  else if $\mathsf{Test}(td_{\mathbb{A}'|_{01}}, C) = 1 \wedge \mathsf{Test}(td_{\mathbb{A}'|_{00}}, C) = \mathsf{Test}(td_{\mathbb{A}'|_{10}}, C) = \mathsf{Test}(td_{\mathbb{A}'|_{11}}, C) = 0$,
    then return 01
  else if $\mathsf{Test}(td_{\mathbb{A}'|_{10}}, C) = 1 \wedge \mathsf{Test}(td_{\mathbb{A}'|_{00}}, C) = \mathsf{Test}(td_{\mathbb{A}'|_{01}}, C) = \mathsf{Test}(td_{\mathbb{A}'|_{11}}, C) = 0$,
    then return 10
  else if $\mathsf{Test}(td_{\mathbb{A}'|_{11}}, C) = 1 \wedge \mathsf{Test}(td_{\mathbb{A}'|_{00}}, C) = \mathsf{Test}(td_{\mathbb{A}'|_{01}}, C) = \mathsf{Test}(td_{\mathbb{A}'|_{10}}, C) = 0$,
    then return 11
  else
    then return $\perp$
```

Figure 3.4.1: Construction of $\Pi'_{KP\text{-}ABE}$

this idea, it is possible to construct a KP-ABE scheme that can deal with longer plaintexts. More specifically, let the plaintext space be k bit. Then, the key generation is the same as Figure 3.4.1. To encrypt a plaintext $m \in \{0,1\}^k$ using attribute set $\boldsymbol{S}$ can be computed by encrypting $\boldsymbol{S}|_m$ using PEKS. In this way, the key generation and encryption algorithms can be extended very simply. On the other hand, the secret key generation and decryption algorithms are a bit more complicated. The secret key for access structure $\mathbb{A}$ is $(td_{\mathbb{A}|_{00\ldots00}}, td_{\mathbb{A}|_{00\ldots01}}, \ldots, td_{\mathbb{A}|_{11\ldots10}}, td_{\mathbb{A}|_{1\ldots1}})$ where $td_{\mathbb{A}|_M} \leftarrow \mathsf{Td}(msk, \mathbb{A}|_M)$ for each $M \in \{0,1\}^k$. Finally, the decryption algorithm takes a ciphertext C and a secret key $sk' = (td'_{00\ldots00}, td'_{00\ldots01}, \ldots, td'_{11\ldots10}, td'_{1\ldots1})$ as input, and computes $\mathsf{Test}(td_M, C)$ for each $M \in \{0,1\}^k$. If there is only one $td_{M'} \in \{0,1\}^k$ such that $\mathsf{Test}(td_{M'}, C) = 1$, and Test algorithm outputs 0 for all other trapdoors, then output M'. Otherwise, output $\perp$.

Note that since all algorithms have to be run in polynomial time, the length of plaintext (i.e., k) should be small. In addition to this, the length of the secret key for an access structure and computation time for decrypting ciphertext become longer if we deal with longer plaintexts. To solve these drawbacks is future work.

3.5 Conclusion

With the development of cloud technology, there are more and more opportunities to outsource data. Under these situations, secure data operations are required. Searchable encryption has been studied to realize secure search on encrypted databases. In particular, public-key schemes (called PEKS) have many possible applications, such as secure routing of e-mails, and various schemes have been proposed.

Most PEKS schemes that can use flexible search conditions are constructed using powerful cryptographic tools, for example, ABE. It seems hard to construct PEKS from weaker tools. However, it has not been rigorously verified whether this is true or not. In this chapter, we proved this intuition by giving a generic construction of KP-ABE, whose access structure is specified by a monotone Boolean formula from PKES, whose search condition is specified by a monotone Boolean formula. Specifically, we gave a generic construction of KP-ABE which satisfies IND-CPA, IND-ANO-CPA, and (perfect) correctness from a PEKS which satisfies IND-CKA and the perfect consistency.

It remains future work to formalize the computational correctness of KP-ABE and to prove that such a KP-ABE can be constructed from the computationally consistent PEKS. Also, to clarify whether the KP-ABE scheme constructed from our generic construction satisfies stronger security than IND-CPA or not is one of the future works.

Chapter 4

Security Notions Against Replayable CCA and the Relationship Among Them

4.1 Introduction

4.1.1 Background and Motivation

In Chapter 3, we introduced Public-key encryption with keyword search (PEKS). In addition to secure data retrieval over encrypted databases, PEKS can also be used to secure e-mail routing. Consider the situation where Bob wants to send an e-mail to Alice using Alice's public key. The receiver, Alice, wants to determine the device that will receive the e-mail based on the keywords contained in the e-mail. For example, if the e-mail contains the keyword "urgent," Alice wants to check the e-mail as soon as possible and route the e-mail to her cell phone for that purpose. In such a situation, if Alice gives her private key to the gateway, the gateway will route the mail correctly by decrypting the encrypted e-mail and checking the content. However, this solution reveals much information about e-mails to the gateway. So, it is expected that PEKS will be used to route e-mail. In the case of using PEKS, Alice only needs to give the gateway a trapdoor to the keyword "urgent" in advance. Then, the gateway can correctly determine the routing destination without knowing the contents of the mail by executing a test algorithm in PEKS using a trapdoor.

As mentioned above, PEKS is a useful tool, but other functionalities can be added to PEKS. For example, proxy re-encryption with keyword search (PRES) was proposed by Shao et al. [42]. PRES is an encryption scheme that combines the features of PEKS and proxy re-encryption (PRE). PRE [32] is an encryption scheme that allows a proxy to convert a ciphertext C encrypted with user A's public key into a ciphertext C' encrypted with user B's public key while preserving the plaintext. Note that the proxy performs this conversion without decrypting the ciphertext. Therefore the information about the plaintext is not revealed to the proxy. In PRES, the combination of PRE and PEKS allows for more flexible mail routing. Let consider the situation in which Carol wants to route e-mails containing specific keywords to her secretary, Dave. At first, Dave gives the trapdoor of a specific keyword to the gateway, and Carol gives the re-encryption key to the gateway. Then when the gateway receives the encrypted e-mail for Carol, he can check whether the e-mail contains the specific keyword. If so, the gateway can also convert that encrypted e-mail into encrypted mail for Dave.

PRES is more useful than PEKS, but the additional processes of transforming the ciphertext require more careful thought about security and adversary models. Indeed, when we consider CCA adversaries in PRES or PRE, we need to be careful in formalizing security. For example, let consider a security game against CCA for PRE, a simple extension of the security game against CCA in a PKE setting. In this setting, we allow the adversary to access the decryption oracle. In addition, we also allow him to transform the ciphertext that is encrypted under one person's public key to the ciphertext that is encrypted under another person's public key. Then, the adversary can trivially win the game using the decryption oracle and the functionality of PRE as follows:

(i) In the challenge phase, the adversary chooses random plaintexts m_0 and m_1 and submit them to the challenger.

(ii) Upon receiving the challenge ciphertext c^* from the challenger, the adversary transforms that challenge ciphertext into the ciphertext c' for the other user.

(iii) The adversary submits c' to the decryption oracle and receives the decryption result m'.

(iv) If $m' = m_0$, the adversary outputs 0, and otherwise, the adversary outputs 1.

Note that the adversary cannot submit the challenge ciphertext to the decryption oracle in the security game. Although the plaintext of c' is the plaintext that the challenger chooses in the challenge phase, c' is not the challenge ciphertext, and the adversary can submit c' to the decryption oracle. So, trivially the adversary can learn the information about the plaintext of the challenge ciphertext. To prevent such attacks, the replayable chosen ciphertext attack (RCCA) adversary model is used. In the RCCA model, the adversary cannot learn anything from the ciphertext of plaintext m_0 and m_1, where m_0 and m_1 are the plaintexts that the adversary submits in the challenge phase. Therefore, the above attack will not work in the RCCA model. RCCA is an adversary model, but is closely related to the security requirement of non-malleability (NM).

NM [43] is one of the most fundamental security requirements for public key encryption (PKE). As mentioned in Chapter 2, NM guarantees that an adversary cannot modify the plaintext of a given ciphertext. For example, consider the electronic bidding using a PKE scheme played by companies A and B. Suppose that company A places its bid of \$1,000,000 by sending encryption c of \$1,000,000 generated by the PKE scheme over the internet. In this case, if the PKE scheme does not satisfy non-malleability, company B might be able to intercept c, make an encryption of \$1,000,001 by modifying c, and use it as its bid. In order to prevent such kind of malicious activities, the PKE scheme should satisfy non-malleability. There are both simulation-based and indistinguishability-based definitions of non-malleability for PKE. Bellare and Sahai [44, 45] showed that these two definitions are equivalent when considering each of chosen plaintext attack (CPA), non-adaptive chosen ciphertext attack (CCA1), and adaptive chosen ciphertext attack (CCA2). In this chapter, we study non-malleability against RCCA [46].

Canetti, Krawczyk, and Nielsen [46] introduced the notion of RCCA security in order to handle an encryption scheme that is "non-malleable except tampering which preserves the plaintext." RCCA security is a relaxation of CCA security and a useful security notion for many practical applications such as authentication and key exchange [46]. To formulate "non-malleability except tampering which preserves the plaintext", in the security experiment of RCCA security, the decryption oracle returns a symbol "Test" when an adversary queries encryption of m_0 and m_1, where m_0 and m_1 are challenge messages. Canetti et al. defined non-malleability against RCCA (NM-RCCA), indistinguishability against RCCA (IND-RCCA), and universal composability against RCCA (UC-RCCA). Moreover, they proved that these three security notions are equivalent when considering a PKE scheme whose plaintext space is super-polynomially large.

As noted above, RCCA security was introduced in order to handle an encryption scheme that is non-malleable except tampering which preserves the plaintext. To clarify whether a security notion against RCCA such as IND-RCCA captures non-malleability except tampering which preserves the plaintext, we should consider the equivalence between the security notion and NM-RCCA. Therefore, NM-RCCA seems to play the central role among security notions against RCCA.

However, the definition of NM-RCCA proposed by Canetti et al. is not a natural extension of original non-malleability. Therefore, it is not clear whether the definition plays the above required role. More specifically, in the security experiment of their NM-RCCA, an adversary is required to make an encryption of m_{1-b} given an encryption of m_b, where b is the challenge bit and (m_0, m_1) are challenge messages. It is not clear whether this definition captures the requirement of original non-malleability that given an encryption of some message m, an adversary cannot generate that of any message related to m. In fact, Canetti et al. claimed the validity of the definition of their NM-RCCA relying on the equivalence between NM-RCCA and UC-RCCA. However, it does not hold when considering an encryption scheme the size of whose plaintext space is polynomial. For this reason, we need to study non-malleability against RCCA more deeply and propose a definition of it that captures the requirement of original non-malleability.

Some readers may think that the UC-RCCA defined by Canetti et al. is a simulation-based formalization, and therefore we do not need more simulation-based non-malleability. However, UC and NM are originally different security notions, and it is not clear whether their formalization of UC captures NM or not. In addition to this, the ideal functionality of the UC-RCCA definition is defined by focusing on confidentiality. Therefore, it is important to clarify the relationship among security notions after formalizing a simulation-based non-malleability against RCCA that properly captures the intuition that an adversary cannot modify the plaintext of a given ciphertext.

4.1.2 Our Contribution

In this chapter, we propose simulation-based and indistinguishability-based definitions of non-malleability against RCCA (we call these SNM-RCCA and INM-RCCA, respectively). Note that while our real goal is a rigorous formulation of NM-RCCA for the PRES schemes, as a first step, we consider the public key encryption setting in which NM-RCCA was first defined. The proposed definitions are natural extensions of original non-malleability.

Thus they have the same spirit as original definitions capturing the intuition that given an encryption of some message m, an adversary cannot generate that of any message related to m. Moreover, we prove that these two security notions and IND-RCCA security proposed by Canetti et al. [46] are all equivalents regardless of the size of plaintext space.

While we can easily formalize indistinguishability-based non-malleability by naturally extending the definition of IND-RCCA proposed by Canetti et al., there is a problem when formalizing simulation-based non-malleability. The most non-trivial point is the decryption oracle we should allow an adversary to access when formalizing a simulation-based security against RCCA. This is because we have to carefully consider the effect of decryption oracles in RCCA environments on attackers by returning the special symbol "Test". At first glance, the decryption oracle in the RCCA environment seems to leak less information than the decryption oracle in the CCA environment. For example, let consider a rerandomizable encryption scheme. In the security game, CCA adversary can rerandomize the challenge ciphertext and know the plaintext of challenge ciphertext by querying the rerandomized challenge ciphertext. On the other hand, if the RCCA adversary makes the same attack, the decryption oracle returns a special symbol "Test", and thus the adversary cannot know the plaintext of challenge ciphertext. However, compared to the ordinary CCA, the decryption oracle seems to leak much more information about messages when considering RCCA due to the special symbol "Test" returned by the decryption oracle. Thus, when formalizing a simulation-based security under the RCCA environment, we need to formalize the intuition that an adversary cannot obtain any information about the plaintext encrypted in the ciphertext except information leaked from the decryption oracle. To capture the intuition, we use a predicate in the definition of simulation-based non-malleability against RCCA. See Section 4.4.1 for more details.

We can see the usefulness of using a predicate when formalizing RCCA security from the following fact. We can define semantic security [47] against RCCA using a predicate similar to the definition of simulation-based non-malleability against RCCA. Then, we can prove that the semantic security against RCCA is equivalent to IND-RCCA proposed by Canetti et al. In Section 4.7, we show the definition of semantic security against RCCA and its equivalence to IND-RCCA. We summarize our results in Figure 4.1.1. Especially, we showed the equivalence between the definitions of IND-RCCA and SNM-RCCA, where the latter is the most strict notion of non-malleability for the RCCA setting, and this implies that it is sufficient to prove IND-RCCA when giving a proof for the non-malleability against RCCA in the most strict sense.

4.1.3 Related Work

PRE and PRES Blaze et al. [32] firstly proposed a PRE scheme. Later, the notion and construction of PRES that combine the concept of PEKS and PRE was proposed by Shao et al. [42]. After their proposal, Yau et al. proposed the multi-hop bi-directional single keyword search scheme [48], and Yang and Ma proposed the PRES scheme supporting conjunctive keyword search [49]. Chen et al. [50] proposed the scheme that the proxy can re-encrypt only the ciphertext that contains some specific keywords.

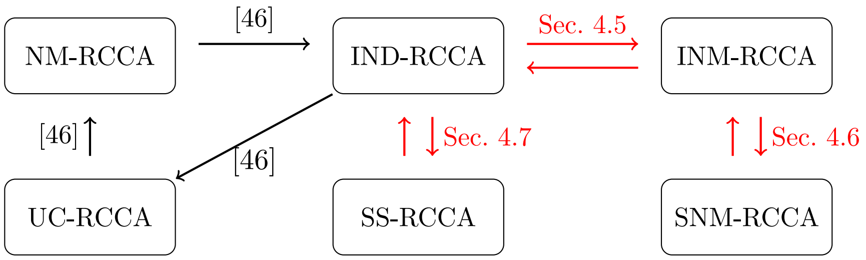

Figure 4.1.1: The summary of our results and previous results regarding security notions against RCCA. SNM-RCCA and INM-RCCA indicate proposed definitions of simulation-based and indistinguishability-based non-malleability, respectively. SS-RCCA indicates proposed definition of semantic security. Solid arrows indicate implications. Red arrows indicate our results. Note that the above results hold when the size of the plaintext space is super-polynomially large.

Security Notions for PKE When considering security for PKE, we need to define security requirements and adversary models. The security requirement that no information about the plaintext is leaked from the ciphertext was proposed by Goldwasser et al. [47]. They proposed simulation-based formalization called semantic security, and game-based formalization called indistinguishability. Regarding the adversary model, they proposed chosen plaintext attack (CPA) model. The notion of chosen ciphertext attack (CCA1) was formalized by Naor and Yung [51], and the adaptive chosen ciphertext attack (CCA2) was formalized by Rackoff et al. [52]. Other than the above adversary models, RCCA [46], WRCCA [53], ECCA [54], RECCA [55], and CCVA [56] have been proposed.

Relation among Notions for PKE Goldwasser and Micali [47] proved the equivalence between semantic security and indistinguishability against CPA. Watanabe et al. [57] proved the equivalence between semantic security and indistinguishability against CCA1 and CCA2. Bellare et al. [58] proved the equivalence between indistinguishability and non-malleability under the CCA2 environment. Bellare and Sahai [44, 45] proved the equivalence between simulation-based non-malleability and indistinguishability-based non-malleability. In addition to the above, Pass et al. [59] considered the situation that an adversary outputs a ciphertext which is decrypted to $\perp$ in the experiment and proved the relation of simulation-based non-malleability and indistinguishability based non-malleability under the condition they considered. Specifically, they proved that those two definitions are equivalent for a PKE scheme that allows efficient sampling of a ciphertext decrypted to $\perp$. On the other hand, they proved a separation scheme exists between the two definitions in the case that a PKE scheme does not allow efficient sampling of a ciphertext decrypted to $\perp$. Katz and Yung [60] proved relations among notions of security for symmetric-key encryption.

RCCA Security Several studies on the construction of RCCA secure PKE schemes have been done since RCCA security was proposed by Canetti et al. [46]. Also, some studies consider RCCA security for various cryptographic primitives such as proxy re-encryption [61, 62, 63], hybrid encryption [64, 65], signcryption [66], and steganography [67]. RCCA security is also used when dealing with rerandomizable encryption schemes. The construction of a rerandomizable encryption scheme that satisfies RCCA security was a non-trivial problem. Groth firstly solved this problem [53]. Their scheme satisfies a weaker variant of RCCA security but does not satisfy RCCA security. Subsequently, the construction of a rerandomizable PKE scheme satisfying RCCA security was proposed by Prabhakaran et al. [68]. However, their scheme uses non-standard cryptographic groups. After these researches, the construction of a rerandomizable PKE scheme satisfying RCCA security from a standard assumption was proposed by Chase et al. [69], and Libert et al. [70] improved the efficiency of their construction.

The idea of using predicates in the CCA security model has also been discussed by Abe et al. [67] and Hofheinz and Kiltz [71].

4.1.4 Chapter Organization

The remainder of this chapter is organized as follows. In Section 4.2, we review the definition of IND-RCCA and NM-RCCA defined by Canetti et al. [46]. In Section 4.3 we introduce the definition of SIM-NME' by Pass et al. [59]. In Section 4.4, we then give our simulation-based definition of non-malleability against RCCA (SNM-RCCA) and indistinguishability-based one (INM-RCCA). In Section 4.5, we prove the equivalence of IND-RCCA and INM-RCCA. In Section 4.6, we also prove the equivalence of INM-RCCA and SNM-RCCA. In Section 4.7, we give our definition of semantic security against RCCA, and prove that it is equivalent to IND-RCCA. In Section 4.8, we state the conclusion of this chapter.

4.2 RCCA Models Defined by Canetti et al.

We review the definition of IND-RCCA security introduced by Canetti et al. [46]. They formalized RCCA security by letting the decryption oracle in the second phase $\mathcal{O}_2$ return a special symbol "Test" when an adversary queries a ciphertext of m_0 or m_1, where m_0 and m_1 are the challenge messages. The formulation that $\mathcal{O}_2$ returns "Test" relaxes CCA security. By this relaxation, even if an adversary queries a ciphertext generated by rerandomizing the challenge ciphertext to $\mathcal{O}_2$, the adversary cannot obtain any information about the challenge bit in the experiments of RCCA security.

Then we give a formal definition of the IND-RCCA security. Let $\Sigma = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme, and $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a pair of PPTAs. We consider the following experiments IND-RCCA-b ($b \in \{0,1\}$):

```
$\underline{\mathrm{Exp}^{\text{IND-RCCA-}b}_{\Sigma,\mathcal{A}}(\lambda)}$
$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$;
$(m_0, m_1, st_1) \leftarrow \mathcal{A}_1^{\mathcal{O}_1}(pk)$;
$c^* \leftarrow \mathsf{Enc}(pk, m_b)$;
$b' \leftarrow \mathcal{A}_2^{\mathcal{O}_2}(c^*, st_1)$;
output $b'$
```

where,

$$\begin{aligned}
&\mathcal{O}_1(c) = \mathsf{Dec}(sk, c),\\
&\mathcal{O}_2(c) = \begin{cases} \mathsf{Test} & (\mathsf{Dec}(sk, c) \in \{m_0, m_1\}) \\ \mathsf{Dec}(sk, c) & (\text{otherwise}). \end{cases}
\end{aligned}$$

We define the advantage as

$$\begin{aligned}
\mathsf{Adv}^{\text{IND-RCCA}}_{\Sigma,\mathcal{A}}(\lambda) :=& |\Pr[\mathsf{Exp}^{\text{IND-RCCA-0}}_{\Sigma,\mathcal{A}}(\lambda) \to 1] \\
&- \Pr[\mathsf{Exp}^{\text{IND-RCCA-1}}_{\Sigma,\mathcal{A}}(\lambda) \to 1]|.
\end{aligned}$$

Definition 14 (IND-RCCA)**.** *We say that Σ is IND-RCCA secure if $\mathsf{Adv}^{\textit{IND-RCCA}}_{\Sigma,\mathcal{A}}(\lambda)$ is negligible for any pair of PPTAs $\mathcal{A}$.*

Then we also review the definition of NM-RCCA security introduced by Canetti et al. [46]. We consider the following experiments NM-RCCA-b ($b \in \{0, 1\}$):

$$\begin{array}{l}
\underline{\mathrm{Exp}^{\text{IND-RCCA-}b}_{\Sigma,\mathcal{A}}(\lambda)} \\
(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda); \\
(m_0, m_1, st_1) \leftarrow \mathcal{A}_1^{\mathcal{O}_1}(pk); \\
c^* \leftarrow \mathsf{Enc}(pk, m_b); \\
c' \leftarrow \mathcal{A}_2^{\mathcal{O}_2}(c^*, st_1); \\
m_{b'} \leftarrow \mathsf{Dec}(sk, c') \\
\text{output } m_{b'}
\end{array}$$

where,

$$\begin{aligned}
&\mathcal{O}_1(c) = \mathsf{Dec}(sk, c),\\
&\mathcal{O}_2(c) = \begin{cases} \mathsf{Test} & (\mathsf{Dec}(sk, c) \in \{m_0, m_1\}) \\ \mathsf{Dec}(sk, c) & (\text{otherwise}). \end{cases}
\end{aligned}$$

We define the advantage as

$$\begin{aligned}
\mathsf{Adv}^{\text{NM-RCCA}}_{\Sigma,\mathcal{A}}(\lambda) :=& |\Pr[\mathsf{Exp}^{\text{NM-RCCA-0}}_{\Sigma,\mathcal{A}}(\lambda) \to m_1] \\
&- \Pr[\mathsf{Exp}^{\text{NM-RCCA-1}}_{\Sigma,\mathcal{A}}(\lambda) \to m_1]|.
\end{aligned}$$

Definition 15 (NM-RCCA)**.** *We say that Σ is NM-RCCA secure if $\mathsf{Adv}^{\textit{NM-RCCA}}_{\Sigma,\mathcal{A}}(\lambda)$ is negligible for any pair of PPTAs $\mathcal{A}$.*

The above formulation of NM-RCCA is very similar to the formulation of IND-RCCA. The only differences between the two formulations are the adversary's output, and the condition of the adversary wins. Furthermore, when the NM-RCCA formulation is compared with the existing definition of NM like SNM-CCA2 or INM-CCA2 introduced in Chapter 2, the experiments are very different. More specifically, NM-RCCA is not a simulation-based formulation because there is no distinguisher and simulator. In addition, compared to INM-CCA2, there is no phase in which the adversary outputs multiple modified ciphertexts after receiving the challenge ciphertext, and the final output form of the adversary is very different.

Thus, the formulation of NM-RCCA is very similar to that of IND-RCCA and is also quite divergent from the existing formulation of NM. Therefore, it is not clear whether the formulation of NM-RCCA by Canetti et al. really captures the meaning of NM.

```
$\underline{\text{SIM-NME}'(\Pi, \mathcal{A}, \lambda, \ell, r)}$
$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$;
$(\mathcal{M}, st_1) \leftarrow \mathcal{A}_1^{\mathcal{O}_1}(pk)$;
$(m_1, \ldots, m_\ell) \leftarrow \mathcal{M}$;
for $i = 1$ to $\ell$
  $c_i := \mathsf{Enc}(pk, m_i)$
$(c'_1, \ldots, c'_r, st_2) \leftarrow \mathcal{A}_2^{\mathcal{O}_2}(c_1, \ldots, c_\ell,$
$h(m_1), \ldots, h(m_\ell), st_1)$;
for $i = 1$ to $r$
  $d_i := \begin{cases} \mathsf{Copy} & (c'_i \in \{c_1, \ldots, c_\ell\}) \\ \mathsf{Dec}(sk, c'_i) & (\text{otherwise}) \end{cases}$
output $(\mathcal{M}, m_1, \ldots, m_\ell, d_1, \ldots, d_r, st_2)$
```

```
$\underline{\overline{\text{SIM-NME}'}(\Pi, \mathcal{A}, \lambda, \ell, r)}$
$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$;
$(\mathcal{M}, st_1) \leftarrow \mathcal{S}_1(pk)$;
$(m_1, \ldots, m_\ell) \leftarrow \mathcal{M}$;
$(c'_1, \ldots, c'_t, st_2) \leftarrow \mathcal{S}_2(h(m_1), \ldots, h(m_\ell), st_1)$;
for $i = 1$ to $r$
  $d_i := \begin{cases} \mathsf{Copy} & (c'_i = \mathsf{Copy}) \\ \mathsf{Dec}(sk, c'_i) & (\text{otherwise}) \end{cases}$
output $(\mathcal{M}, m_1, \ldots, m_\ell, d_1, \ldots, d_r, st_2)$
```

Figure 4.3.1: Experiments SIM-NME$'$ and $\overline{\text{SIM-NME}'}$

4.3 Definition of SNM by Pass et al.

When formalizing SNM-RCCA, we refer to the formulation of Pass et al. [59]. Therefore, we introduce their formalization and explain their ideas. They formalize SNM under the name SIM-NME$'$ as shown in the Figure 4.3.1.

where, $\mathcal{M}$ is a Turing machine that samples a vector of $\ell(\lambda)$ messages from a distribution. We say that $\mathcal{M}$ is an (p, ℓ)-valid message-sampler if 1) the running time of $\mathcal{M}(1^\lambda)$ is bounded by $p(\lambda)$, and 2) there exists polynomials $l_1, l_2, \ldots, l_\ell$ such that $\mathcal{M}(1^\lambda)$ always outputs message sequences $(m_1, \ldots, m_{\ell(\lambda)})$ such that $|m_i| = l_i(1^\lambda)$ for all $1 \leq i \leq \ell(\lambda)$.

PKE scheme Π is SIM-NME$'$ secure if for polynomials $\ell(\lambda), r(\lambda)$ and $p(\lambda)$, every polynomial-time computable history function $h(\cdot)$, every PPTA $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ which runs in time $p(\lambda)$ and always outputs a (p, ℓ)-valid message sampler, there exists an PPTA $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ that always outputs a (p, ℓ)-valid message sampler, such that the following two distributions are computationally indistinguishable:

$$\left\{\text{SIM-NME}'(\Pi, \mathcal{A}, \lambda, \ell(\lambda), r(\lambda))\right\}_\lambda \approx \left\{\overline{\text{SIM-NME}'}(\Pi, \mathcal{S}, \lambda, \ell(\lambda), r(\lambda))\right\}_\lambda$$

They considered the case where an adversary outputs challenge ciphertext directly in the experiment of simulation-based and indistinguishability-based non-malleability under the CCA environment. They allowed a simulator to output the symbol "COPY" and formulated it. Similarly to the definitions of Pass et al., we allow a simulator to output special symbols "Test" in order to handle replays of ciphertexts. In addition to this, Pass et al. [59] showed that there is a separation between SNM-CCA and INM-CCA when the formalization of SNM-CCA does not allow the simulator to output $\perp$. Thus, in our definition of SNM-RCCA, we allow the simulator to output "Test" and $\perp$.

Regarding indistinguishability-based definition, they formalize INM under the name IND-NME$'$ that considers the case that the adversary outputs the challenge ciphertext in the experiment. However, when we consider CCA, they proved that the definition of IND-NME$'$ is equivalent to INM-CCA. This is because, unlike the case of SIN-NME$'$, we do not need to consider a simulator in the IND-NME$'$, so the adversary's behavior of outputting

```
$\mathrm{Exp}^{\text{SNM-RCCA-0}}_{\Sigma,\mathcal{A},h}(\lambda)$
$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$;
$(\mathcal{M}, \boldsymbol{P}(\cdot,\cdot), st_1) \leftarrow \mathcal{A}_1^{\mathcal{O}_1}(pk)$;
$m \leftarrow \mathcal{M}$;
$c^* \leftarrow \mathsf{Enc}(pk, m)$;
$(c_1, \ldots, c_n, st_2) \leftarrow \mathcal{A}_2^{\mathcal{O}_2}(c^*, h(m), st_1)$;
for $i = 1$ to $n$
    $d_i := \begin{cases} \mathsf{Test} & (\boldsymbol{P}(m, \mathsf{Dec}(sk, c_i)) = 1) \\ \mathsf{Dec}(sk, c_i) & (\text{otherwise}) \end{cases}$
output $(\mathcal{M}, m, \boldsymbol{P}(\cdot,\cdot), d_1, \ldots, d_n, st_2)$
```

```
$\mathrm{Exp}^{\text{SNM-RCCA-1}}_{\Sigma,\mathcal{S},h}(\lambda)$
$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$;
$(\mathcal{M}, \boldsymbol{P}(\cdot,\cdot), st_1) \leftarrow \mathcal{S}_1(pk)$;
$m \leftarrow \mathcal{M}$;

$(c_1, \ldots, c_n, st_2) \leftarrow \mathcal{S}_2^{\boldsymbol{P}(m,\cdot)}(h(m), st_1)$;
for $i = 1$ to $n$
    $d_i := \begin{cases} \mathsf{Test} & (\boldsymbol{P}(m, \mathsf{Dec}(sk, c_i) \\ & ) = 1 \lor c_i = \mathsf{Test}) \\ \perp & (c_i = \perp) \\ \mathsf{Dec}(sk, c_i) & (\text{otherwise}) \end{cases}$
output $(\mathcal{M}, m, \boldsymbol{P}(\cdot,\cdot), d_1, \ldots, d_n, st_2)$
```

Figure 4.4.1: Experiments SNM-RCCA-0 and SNM-RCCA-1

the challenge ciphertext during the experiment does not particularly affect it. Also, we do not need to consider the case where the adversary outputs the ciphertext decrypt to $\perp$ in the experiment. Therefore, when we consider indistinguishability-based NM, we can use the definition of INM that is easy to use. Likewise, when we formulate INM-RCCA by simply extending the definition of INM-CCA (i.e., the output of the decryption oracle is the same as in the case of IND-RCCA).

4.4 Definitions of SNM-RCCA and INM-RCCA

In this section, we introduce our definitions of simulation-based and indistinguishability-based non-malleability against RCCA.

4.4.1 Definition of SNM-RCCA

We give our definition of simulation-based non-malleability under the RCCA environment (SNM-RCCA) as follows.

Let $\Sigma = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme, $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ be pairs of PPTAs, and h be a polynomial time computable function. We consider the following experiments SNM-RCCA-0 and SNM-RCCA-1 as in Figure 4.4.1.

In the Figure 4.4.1, the predicate $\boldsymbol{P}$ should satisfy $\boldsymbol{P}(m, m) = 1$ for any m which is included in the support of $\mathcal{M}$, and

$$\mathcal{O}_1(c) = \mathsf{Dec}(sk, c),$$

$$\mathcal{O}_2(c) = \begin{cases} \mathsf{Test} & (\boldsymbol{P}(m, \mathsf{Dec}(sk, c)) = 1) \\ \mathsf{Dec}(sk, c) & (\text{otherwise}). \end{cases}$$

In the above two experiments, $\mathcal{M}$ is a distribution over the plaintext space. In addition to this, the function h in the above definition formalize the prior knowledge of the adversary.

With this h, the definition states that even if the adversary has a priori knowledge about plaintext, the adversary cannot do more than attacks based on that a priori knowledge.

We define the advantage as

$$\begin{aligned}\mathsf{Adv}^{\text{SNM-RCCA}}_{\Sigma,\mathcal{A},\mathcal{S},\mathcal{D},h}(\lambda) :=& |\Pr[\mathcal{D}(\mathsf{Exp}^{\text{SNM-RCCA-0}}_{\Sigma,\mathcal{A},h}(\lambda)) \to 1] \\ &- \Pr[\mathcal{D}(\mathsf{Exp}^{\text{SNM-RCCA-1}}_{\Sigma,\mathcal{S},h}(\lambda)) \to 1]|.\end{aligned}$$

Definition 16 (SNM-RCCA security). *We say that Σ is SNM-RCCA secure if for any polynomial time computable function h and for any pair of PPTAs $\mathcal{A}$, there exists a pair of PPTAs $\mathcal{S}$ such that $\mathsf{Adv}^{\text{SNM-RCCA}}_{\Sigma,\mathcal{A},\mathcal{S},\mathcal{D},h}(\lambda)$ is negligible for any PPTA $\mathcal{D}$.*

On the use of predicate. We use a predicate in the definition of SNM-RCCA above. The reason is as follows. When we formalize the simulation-based RCCA security, it is not trivial what decryption oracle we should allow an adversary to access. For example, suppose that we allow an adversary to access the decryption oracle which returns "Test" only when he queries a ciphertext of m, where m is the plaintext chosen in the experiment as the target of tampering. Then, this decryption oracle seems to leak m entirely to the adversary when the size of the plaintext space is polynomial. At first, the adversary queries a ciphertext of all plaintexts contained in the plaintext space, and then he can learn m by decryption oracle's response "Test".

In this way, in the RCCA environment, the decryption oracle leaks the information of the plaintext m chosen in the experiment. Thus, when we formalize simulation-based RCCA security, we need to formalize it by capturing the intuition that an adversary cannot obtain any information about the plaintext encrypted in the ciphertext except the information leaked from the decryption oracle. In our definition, To capture the intuition, we make an adversary output a predicate that determines whether a decryption result of a decryption query is "Test" or not. In other words, this predicate indicates which kind of tampering is considered to be a success. More importantly, we allow a simulator to access the predicate oracle in order to give him the same information leaked from the decryption oracle which the adversary accesses. We see that if such a simulator can simulate an adversary, the adversary does not obtain any information of the plaintext from the ciphertext except information leaked from the decryption oracle.

We can observe the usefulness of using a predicate when formalizing RCCA security from the following fact. We can define semantic security under the RCCA environment using a predicate in a similar way as the definition of SNM-RCCA. Then, we can prove that semantic security against RCCA is equivalent to IND-RCCA security proposed by Canetti et al. [46]. In Section 4.7, we show the definition of semantic security against RCCA and its equivalence to IND-RCCA.

4.4.2 Definition of INM-RCCA

We give our definition of indistinguishability-based non-malleability under the RCCA environment (INM-RCCA) as follows.

Let $\Sigma = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme, and $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ be a triple of PPTAs. We consider the following experiments INM-RCCA-b ($b \in \{0, 1\}$):

```
$\underline{\mathrm{Exp}^{\text{INM-RCCA-}b}_{\Sigma,\mathcal{A}}(\lambda)}$
$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$;
$(m_0, m_1, st_1) \leftarrow \mathcal{A}_1^{\mathcal{O}_1}(pk)$;
$c^* \leftarrow \mathsf{Enc}(pk, m_b)$;
$(c_1, \ldots, c_n, st_2) \leftarrow \mathcal{A}_2^{\mathcal{O}_2}(c^*, st_1)$;
for $i = 1$ to $n$
  $d_i := \mathcal{O}_2(c_i)$
$b' \leftarrow \mathcal{A}_3(d_1, \ldots, d_n, st_2)$;
output $b'$
```

where,

$$\mathcal{O}_1(c) = \mathsf{Dec}(sk, c),$$
$$\mathcal{O}_2(c) = \begin{cases} \mathsf{Test} & (\mathsf{Dec}(sk, c) \in \{m_0, m_1\}) \\ \mathsf{Dec}(sk, c) & (\text{otherwise}). \end{cases}$$

We define the advantage as

$$\begin{aligned}\mathsf{Adv}^{\text{INM-RCCA}}_{\Sigma,\mathcal{A}}(\lambda) :=& |\Pr[\mathsf{Exp}^{\text{INM-RCCA-0}}_{\Sigma,\mathcal{A}}(\lambda) \to 1] \\ &- \Pr[\mathsf{Exp}^{\text{INM-RCCA-1}}_{\Sigma,\mathcal{A}}(\lambda) \to 1]|.\end{aligned}$$

Definition 17 (INM-RCCA security)**.** *We say that Σ is INM-RCCA secure if $\mathsf{Adv}^{\text{INM-RCCA}}_{\Sigma,\mathcal{A}}$ (λ) is negligible for any triple of PPTAs $\mathcal{A}$.*

As mentioned in Section 4.3, our formalization of INM-RCCA is a natural extension of INM-CCA. Therefore, we do not need to use predicates in the experiment like in the case of SNM-RCCA.

4.5 Equivalence of IND-RCCA and INM-RCCA

We can prove the equivalence between IND-RCCA and INM-RCCA. Specifically, the following two theorems, Theorems 5 and 6 hold.

At first, we prove that INM-RCCA implies IND-RCCA.

Theorem 5. *If a PKE scheme $\Sigma = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is INM-RCCA secure, then Σ is IND-RCCA secure.*

Proof. We assume that for any INM-RCCA adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3)$, $\mathsf{Adv}^{\text{INM-RCCA}}_{\Sigma,\mathcal{B}}(\lambda)$ is negligible. Then, we show $\mathsf{Adv}^{\text{IND-RCCA}}_{\Sigma,\mathcal{A}}(\lambda)$ is negligible for any IND-RCCA adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$.

We construct an INM-RCCA adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3)$ who uses internally $\mathcal{A}$ as in Figure 4.5.1. When $\mathcal{A}_2$ queries a ciphertext c to $\mathcal{B}_2$, $\mathcal{B}_2$ sends c to the decryption oracle that he can access. Then $\mathcal{B}_2$ sends the response from the oracle to $\mathcal{A}_2$.

By the construction of $\mathcal{B}$ as in Figure 4.5.1, $\mathcal{B}$ simulates IND-RCCA-0 experiment for $\mathcal{A}$ when $\mathcal{B}$ receives an encryption of m_0. Moreover, $\mathcal{B}$ outputs 1 only when $\mathcal{A}$ outputs 1, and thus it holds that

$$\Pr[\mathsf{Exp}^{\text{INM-RCCA-0}}_{\Sigma,\mathcal{B}}(\lambda) \to 1] = \Pr[\mathsf{Exp}^{\text{IND-RCCA-0}}_{\Sigma,\mathcal{A}}(\lambda) \to 1].$$

```
$\mathcal{B}_1^{\mathcal{O}_1}(pk)$
$(m_0, m_1, st'_1) \leftarrow \mathcal{A}_1^{\mathcal{O}_1}(pk)$
$st_1 := (m_0, m_1, st'_1)$
output $(m_0, m_1, st_1)$
---
$\mathcal{B}_2^{\mathcal{O}_2}(c^*, st_1)$
$b' \leftarrow \mathcal{A}_2^{\mathcal{O}_2}(c^*, st'_1)$
$c$ : empty string
$st_2 := b'$
output $(c, st_2)$
---
$\mathcal{B}_3(d, st_2)$
output $b'$
```

Figure 4.5.1: The constructions of $\mathcal{B}$ used in Theorem 5

Likewise, it holds that

$$\Pr[\mathsf{Exp}_{\Sigma,\mathcal{B}}^{\text{INM-RCCA-1}}(\lambda) \to 1] = \Pr[\mathsf{Exp}_{\Sigma,\mathcal{A}}^{\text{IND-RCCA-1}}(\lambda) \to 1].$$

Therefore, we can derive

$$\begin{aligned}
&\mathsf{Adv}_{\Sigma,\mathcal{A}}^{\text{IND-RCCA}}(\lambda) \\
&= |\Pr[\mathsf{Exp}_{\Sigma,\mathcal{A}}^{\text{IND-RCCA-1}}(\lambda) \to 1] \\
&\qquad - \Pr[\mathsf{Exp}_{\Sigma,\mathcal{A}}^{\text{IND-RCCA-0}}(\lambda) \to 1]| \\
&= |\Pr[\mathsf{Exp}_{\Sigma,\mathcal{B}}^{\text{INM-RCCA-0}}(\lambda) \to 1] \\
&\qquad - \Pr[\mathsf{Exp}_{\Sigma,\mathcal{B}}^{\text{INM-RCCA-1}}(\lambda) \to 1]| \\
&= \mathsf{Adv}_{\Sigma,\mathcal{B}}^{\text{INM-RCCA}}(\lambda).
\end{aligned}$$

Since we assume Σ is INM-RCCA secure, it is negligible. □

Note that since $\mathcal{B}_2$ outputs empty string as c, the input d for $\mathcal{B}_3$ is also empty string in the above proof of theorem.

Then, we prove that IND-RCCA implies INM-RCCA.

Theorem 6. *If a PKE scheme $\Sigma = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is IND-RCCA secure, then Σ is INM-RCCA secure.*

Proof. We assume that for any IND-RCCA adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$, $\mathsf{Adv}_{\Sigma,\mathcal{B}}^{\text{IND-RCCA}}(\lambda)$ is negligible. Then, we show $\mathsf{Adv}_{\Sigma,\mathcal{A}}^{\text{INM-RCCA}}(\lambda)$ is negligible for any INM-RCCA adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$.

We construct an IND-RCCA adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ who uses internally $\mathcal{A}$ as in Figure 4.5.2. When $\mathcal{A}_2$ queries a ciphertext c to $\mathcal{B}_2$, $\mathcal{B}_2$ sends c to the decryption oracle that he can access. Then $\mathcal{B}_2$ sends the response from the oracle to $\mathcal{A}_2$.

By the construction of $\mathcal{B}$ as in Figure 4.5.2, $\mathcal{B}$ simulates INM-RCCA-0 experiment for $\mathcal{A}$ when $\mathcal{B}$ runs in IND-RCCA-0 experiment. Moreover, $\mathcal{B}$ outputs 1 only when $\mathcal{A}$ outputs 1. Therefore, we have

$$\Pr[\mathsf{Exp}_{\Sigma,\mathcal{B}}^{\text{IND-RCCA-0}}(\lambda) \to 1] = \Pr[\mathsf{Exp}_{\Sigma,\mathcal{A}}^{\text{INM-RCCA-0}}(\lambda) \to 1].$$

```
$\mathcal{B}_1^{\mathcal{O}_1}(pk)$
$(m_0, m_1, st'_1) \leftarrow \mathcal{A}_1^{\mathcal{O}_1}(pk)$
$st_1 := st'_1$
output $(m_0, m_1, st_1)$

$\mathcal{B}_2^{\mathcal{O}_2}(c^*, st_1)$
$(c_1, \ldots, c_n, st'_2) \leftarrow \mathcal{A}_2^{\mathcal{O}_2}(c^*, st'_1)$
for $i = 1$ to $n$
  $d_i := \mathcal{O}_2(c_i)$
$b' \leftarrow \mathcal{A}_3(d_1, \ldots, d_n, st'_2)$
```

Figure 4.5.2: The constructions of $\mathcal{B}$ used in Theorem 6

Similarly, it holds that

$$\Pr[\mathsf{Exp}_{\Sigma,\mathcal{B}}^{\text{IND-RCCA-1}}(\lambda) \to 1] = \Pr[\mathsf{Exp}_{\Sigma,\mathcal{A}}^{\text{INM-RCCA-1}}(\lambda) \to 1].$$

Therefore, we can derive

$$\begin{aligned}
&\mathsf{Adv}_{\Sigma,\mathcal{A}}^{\text{INM-RCCA}}(\lambda) \\
&= |\Pr[\mathsf{Exp}_{\Sigma,\mathcal{A}}^{\text{INM-RCCA-1}}(\lambda) \to 1] \\
&\qquad - \Pr[\mathsf{Exp}_{\Sigma,\mathcal{A}}^{\text{INM-RCCA-0}}(\lambda) \to 1]| \\
&= |\Pr[\mathsf{Exp}_{\Sigma,\mathcal{B}}^{\text{IND-RCCA-0}}(\lambda) \to 1] \\
&\qquad - \Pr[\mathsf{Exp}_{\Sigma,\mathcal{B}}^{\text{IND-RCCA-1}}(\lambda) \to 1]| \\
&= \mathsf{Adv}_{\Sigma,\mathcal{B}}^{\text{IND-RCCA}}(\lambda).
\end{aligned}$$

Since we assume Σ is IND-RCCA secure, it is negligible. □

4.6 Equivalence of SNM-RCCA and INM-RCCA

In this section, we prove the equivalence between SNM-RCCA and INM-RCCA proposed in this chapter. By defining SNM-RCCA using a predicate, we can prove the equivalence between them regardless of the size of plaintext space. In the proof of implication from INM-RCCA to SNM-RCCA, when the size of the plaintext space is polynomial, the simulator can submit all plaintexts in the plaintext space to predicate oracle, and he can know the outputs. By using that information, the simulator can simulate the behavior of the adversary. When the size of the plaintext space is super polynomially large, we can prove the equivalence like as the equivalence between SNM-CCA and INM-CCA by Bellare and Sahai [44, 45]. Specifically, we use the technique of switching the original key pair to different key pair such that the simulator can perform decryption.

4.6.1 INM-RCCA implies SNM-RCCA

We prove that INM-RCCA implies SNM-RCCA by a case analysis. We can consider two cases where the size of the plaintext space which a PKE scheme Σ supports is polynomial or not. We consider the case that the size of the plaintext space is polynomial at first, and give a proof. After that, we consider the other case, and give a proof.

Theorem 7. *If a PKE scheme* $\Sigma = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is INM-RCCA secure, and the size of the plaintext space of* Σ *is polynomial, then* Σ *is SNM-RCCA secure.*

In the proof of Theorem 7, we use a sequence of games (Game 0 to Game 3). We can show that Game 0 and Game 1 are equivalent games by using the key switching technique of [44, 45]. The reason for switching the keys is to allow a simulator who cannot access to the decryption oracle to answer the decryption queries from the adversary $\mathcal{A}$ running inside the simulator. Then, we can show from INM-RCCA that we assumed, the difference between Game 1 and Game 2 is negligible. Since the simulator does not receive the challenge ciphertext, he generates the ciphertext by himself and inputs it to $\mathcal{A}$ used inside the simulator. Game 2 is used to show the effects when the simulator itself creates the challenge ciphertext is negligible. Finally, by the construction of $\mathcal{S}$, we can show that Game 2 and Game 3 are equivalent games.

Proof. We assume $\mathsf{Adv}^{\text{INM-RCCA}}_{\Sigma,\mathcal{B}}(\lambda)$ is negligible for any INM-RCCA adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3)$. Then, we show for any SNM-RCCA adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and for any polynomial time computable function h, there exists a simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ such that $\mathsf{Adv}^{\text{SNM-RCCA}}_{\Sigma,\mathcal{A},\mathcal{S},\mathcal{D},h}(\lambda)$ is negligible for any distinguisher $\mathcal{D}$. To give a proof, we use a sequence of games (Game 0 to Game 3), and the construction of $\mathcal{S}$ is as in Figure 4.6.1.

We define Game 0 to Game 3 as follows:

Game 0: Game 0 is the same as SNM-RCCA-0 for $\mathcal{A}$ and $\mathcal{D}$. We denote the plaintext sampled in SNM-RCCA-0 as m_0.

Game 1: The difference from Game 0 is to create $(pk', sk') \leftarrow \mathsf{Gen}(1^\lambda)$ newly and change the game so that (pk', sk') are used throughout the game. The input to $\mathcal{A}_1$ is changed to pk', the challenge ciphertext is generated using pk', and the oracles $\mathcal{O}_1$ and $\mathcal{O}_2$ that $\mathcal{A}$ accesses are changed to the oracles that use sk'. In addition, the secret key used to decrypt ciphertexts c_i $(i = 1, \ldots, n)$ output by $\mathcal{A}_2$ is changed to sk'.

Game 2: The difference from Game 1 is that $m_1 \leftarrow \mathcal{P}_{m_0}$ is sampled in addition to m_0, where $\mathcal{P}_{m_0}$ is the uniform distribution over all plaintexts m' such that $\boldsymbol{P}(m_0, m') = 1$. In addition, the challenge ciphertext $c^* \leftarrow \mathsf{Enc}(pk', m_0)$ is changed to $c^* \leftarrow \mathsf{Enc}(pk', m_1)$.

Game 3: Game 3 is the same as SNM-RCCA-1 under PPTA $\mathcal{S}$ and pk.

We can sample efficiently from $\mathcal{P}_{m_0}$ which is used in Game 2 and $\mathcal{S}_2$. This is because, since the size of the plaintext space is polynomial, by inputing all plaintexts to $\boldsymbol{P}(m_0, \cdot)$, we can identify all plaintexts m' which satisfy $\boldsymbol{P}(m_0, m') = 1$. Note that there always exists at least one plaintext m' satisfying $\boldsymbol{P}(m_0, m') = 1$ because m_0 satisfies $\boldsymbol{P}(m_0, m_0)$.

Let T_i be the event that 1 is output by $\mathcal{D}$ in Game i.

Lemma 1. *It holds that* $\Pr[T_1] = \Pr[T_0]$.

Proof. Game 1 is SNM-RCCA-0 under (pk', sk'), and (pk, sk) is not used. Although the key pair (pk, sk) is generated as $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$, pk is not input to $\mathcal{A}$ in Game 1. Therefore, Game 0 and Game 1 are equivalent from $\mathcal{A}$'s view. Thus, it holds that $\Pr[T_1] = \Pr[T_0]$. □

```
$\underline{\mathcal{S}_1(pk)}$
$(pk', sk') \leftarrow \mathsf{Gen}(1^\lambda)$
$(\mathcal{M}, \boldsymbol{P}(\cdot,\cdot), st'_1) \leftarrow \mathcal{A}_1^{\mathcal{O}'_1}(pk')$
$st_1 := (pk, pk', sk', st'_1)$
output $(\mathcal{M}, \boldsymbol{P}(\cdot,\cdot), st_1)$
---------------------------------------------
$\underline{\mathcal{S}_2^{\boldsymbol{P}(m_0,\cdot)}(h(m_0), st_1)}$
$m_1 \leftarrow \mathcal{P}_{m_0}$
$c^* \leftarrow \mathsf{Enc}(pk', m_1)$
$(c'_1, \ldots, c'_n, st'_2) \leftarrow \mathcal{A}_2^{\mathcal{O}'_2}(c^*, h(m_0), st'_1)$
$st_2 := st'_2$
for $i = 1$ to $n$
    $d'_i := \begin{cases} \mathsf{Test} & (m'_i \leftarrow \mathsf{Dec}(sk', c'_i), \boldsymbol{P}(m_0, m'_i) = 1) \\ \mathsf{Dec}(sk', c'_i) & (\text{otherwise}) \end{cases}$
    $c_i := \begin{cases} \mathsf{Test} & (d'_i = \mathsf{Test}) \\ \bot & (d'_i = \bot) \\ \mathsf{Enc}(pk, d'_i) & (\text{otherwise}) \end{cases}$
output $(c_1, \ldots, c_n, st_2)$
```

Figure 4.6.1: The construction of $\mathcal{S}$ used in Theorem 7

Lemma 2. *There exists $\mathcal{B}$ as in Figure 4.6.2 such that* $|\Pr[T_2] - \Pr[T_1]|$=$\mathsf{Adv}_{\Sigma,\mathcal{B}}^{\textit{INM-RCCA}}(\lambda)$.

Proof. We construct a reduction $\mathcal{B}$ that breaks INM-RCCA security from $\mathcal{A}$. We construct INM-RCCA adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3)$ under (pk', sk') who uses internally $\mathcal{A}$ and $\mathcal{D}$ as in Figure 4.6.2. When $\mathcal{A}_2$ queries a ciphertext c to $\mathcal{B}_2$, $\mathcal{B}_2$ sends c to the decryption oracle that he can access. Then, $\mathcal{B}_2$ receives m or "Test" from the oracle. After that, if $\boldsymbol{P}(m_0, m) = 1$ or $\mathcal{B}_2$ receives "Test", then $\mathcal{B}_2$ sends "Test" to $\mathcal{A}_2$. Otherwise, $\mathcal{B}_2$ sends m to $\mathcal{A}_2$. Since the plaintext m_1 which is generated in $\mathcal{B}_1$ satisfies $\boldsymbol{P}(m_0, m_1) = 1$, $\mathcal{B}_2$ can simulate the decryption oracle correctly. Likewise, $\mathcal{B}_2$ can simulate the sequence of d_i which is input to $\mathcal{D}$ correctly.

When $\mathcal{B}$ runs in INM-RCCA-0, $\mathcal{B}$ simulates Game 1 for $\mathcal{A}$ and $\mathcal{D}$. Moreover, $\mathcal{B}$ outputs 1 only when $\mathcal{D}$ outputs 1. Therefore, we have

$$\Pr[\mathsf{Exp}_{\Sigma,\mathcal{B}}^{\text{INM-RCCA-0}}(\lambda) \to 1] = \Pr[T_1].$$

Similarly, we have

$$\Pr[\mathsf{Exp}_{\Sigma,\mathcal{B}}^{\text{INM-RCCA-1}}(\lambda) \to 1] = \Pr[T_2].$$

Therefore, we can derive

$$\begin{aligned} &|\Pr[T_2] - \Pr[T_1]| \\ &= |\Pr[\mathsf{Exp}_{\Sigma,\mathcal{B}}^{\text{INM-RCCA-1}}(\lambda) \to 1] \\ &\qquad - \Pr[\mathsf{Exp}_{\Sigma,\mathcal{B}}^{\text{INM-RCCA-0}}(\lambda) \to 1]| \\ &= \mathsf{Adv}_{\Sigma,\mathcal{B}}^{\text{INM-RCCA}}(\lambda). \end{aligned}$$

□

```
$\mathcal{B}_1^{\mathcal{O}'_1}(pk')$
$(\mathcal{M}, \boldsymbol{P}(\cdot,\cdot), st'_1) \leftarrow \mathcal{A}_1^{\mathcal{O}'_1}(pk')$
$m_0 \leftarrow \mathcal{M}, m_1 \leftarrow \mathcal{P}_{m_0}$
$st_1 := (m_0, m_1, \mathcal{M}, \boldsymbol{P}(\cdot,\cdot), st'_1)$
output $(m_0, m_1, st_1)$

$\mathcal{B}_2^{\mathcal{O}'_2}(c^*, st_1)$
$(c'_1, \ldots, c'_n, st'_2) \leftarrow \mathcal{A}_2^{\mathcal{O}'_2}(c^*, h(m_0), st'_1)$
$st_2 := (m_0, m_1, \mathcal{M}, \boldsymbol{P}(\cdot,\cdot), st'_2)$
output $(c'_1, \ldots, c'_n, st_2)$

$\mathcal{B}_3(d_1, \ldots, d_n, st_2)$
For each $d_i$, check the value of $\boldsymbol{P}(m_0, d_i)$ by using $m_0$ and $\boldsymbol{P}(\cdot,\cdot)$.
Set $d_i :=$"Test" that satisfies $\boldsymbol{P}(m_0, d_i) = 1$ by the above procedure.
$b' \leftarrow \mathcal{D}(\mathcal{M}, m_0, \boldsymbol{P}(\cdot,\cdot), d_1, \ldots, d_n, st'_2)$
output $b'$
```

Figure 4.6.2: The construction of $\mathcal{B}$ used in Lemma 2

In above proof, some readers might think that if an adversary $\mathcal{A}$ chooses a predicate $\boldsymbol{P}$ such that $\boldsymbol{P}(m_0, m') = 1$ only when $m_0 = m'$, it seems that we have to consider the INM-RCCA experiment with $m_1 = m_0$. However, the advantage of the adversary $\mathcal{B}$ against INM-RCCA is equal to 0 in that case. This implies the advantage of $\mathcal{A}$ against SNM-RCCA is also equal to 0. *Hence, it is trivially secure without assuming INM-RCCA*. Thus, we do not have to consider that case.

Lemma 3. *It holds that* $\Pr[T_3] = \Pr[T_2]$.

Proof. In Game 3, $\mathcal{S}$ uses $\mathcal{A}$ internally as in Figure 4.6.1. Since Game 3 is SNM-RCCA-1, $\mathcal{S}$ cannot access to the decryption oracle. However, $\mathcal{S}$ generates (pk', sk') internally, and he can access the predicate oracle $\boldsymbol{P}(m_0, \cdot)$. In addition, an input to $\mathcal{A}$ is pk' as in Game 2, and thus $\mathcal{S}_2$ can respond to decryption queries from $\mathcal{A}$ by using sk' and $\boldsymbol{P}(m_0, \cdot)$.

$\mathcal{S}_2$ inputs a ciphertext of m_1 which is generated internally in $\mathcal{S}_2$ to $\mathcal{A}_2$, and $\mathcal{A}_2$ outputs a sequence of ciphertexts. $\mathcal{S}_2$ decrypts them using sk'. After that, each d_i is encrypted by using pk, and the encrypted sequence is the output of $\mathcal{S}_2$. All ciphertexts are decrypted using sk after $\mathcal{S}_2$ outputs, and these sequence is input to $\mathcal{D}$. Here, the distributions of the inputs to $\mathcal{D}$ in Game 2 and Game 3 are identical. Thus, it holds that $\Pr[T_3] = \Pr[T_2]$. □

By using Lemma 1 to Lemma 3, we can derive

$$\begin{aligned}
&\mathsf{Adv}^{\text{SNM-RCCA}}_{\Sigma,\mathcal{A},\mathcal{S},\mathcal{D},h}(\lambda) \\
&= |\Pr[\mathcal{D}(\mathsf{Exp}^{\text{SNM-RCCA-0}}_{\Sigma,\mathcal{A},h}(\lambda)) \to 1] \\
&\quad - \Pr[\mathcal{D}(\mathsf{Exp}^{\text{SNM-RCCA-1}}_{\Sigma,\mathcal{A},h}(\lambda)) \to 1]| \\
&= |\Pr[T_0] - \Pr[T_3]| = |\Pr[T_1] - \Pr[T_2]| \\
&= \mathsf{Adv}^{\text{INM-RCCA}}_{\Sigma,\mathcal{B}}(\lambda).
\end{aligned}$$

Therefore, for any $\mathcal{A}$, there exists $\mathcal{S}$ as in Figure 4.6.1 such that $\mathsf{Adv}^{\text{SNM-RCCA}}_{\Sigma,\mathcal{A},\mathcal{S},\mathcal{D},h}(\lambda)$ is negligible for any $\mathcal{D}$. □

Then, we prove that INM-RCCA implies SNM-RCCA when the size of plaintext space is larger than polynomial.

Theorem 8. *If a PKE scheme* $\Sigma = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is INM-RCCA secure, and the size of the plaintext space of* Σ *is super polynomially large, then* Σ *is SNM-RCCA secure.*

In the proof of Theorem 8, we use a sequence of games (Game 0 to Game 5). We can show that Game 0 and Game 1 are equivalent games by using the key switching technique of [44, 45]. The reason for switching the keys is to allow a simulator who cannot access to the decryption oracle to answer the decryption queries from the adversary $\mathcal{A}$ running inside the simulator. Then, we can show that the difference between Game 1 and Game 2 is negligible by using the fact the size of plaintext space is super pollynomially large. Likewise, we can also show that the difference between Game 2 and Game 3 is negligible. After that, we can show from INM-RCCA that we assumed, the difference between Game 3 and Game 4 is negligible. Since the simulator does not receive the challenge ciphertext, he generates the ciphertext by himself and inputs it to $\mathcal{A}$ used inside the simulator. Game 2 to 4 are used to show the effects when the simulator itself creates the challenge ciphertext is negligible. Finally, by the construction of $\mathcal{S}$, we can show that Game 4 and Game 5 are equivalent games.

Proof. Let the plaintext space of Σ be $\{0,1\}^{\ell}$. We assume $\mathsf{Adv}^{\text{INM-RCCA}}_{\Sigma,\mathcal{B}}(\lambda)$ is negligible for any INM-RCCA adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3)$. Then, we show for any SNM-RCCA adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and for any polynomial time computable function h, there exists a simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ such that $\mathsf{Adv}^{\text{SNM-RCCA}}_{\Sigma,\mathcal{A},\mathcal{S},\mathcal{D},h}(\lambda)$ is negligible for any distinguisher $\mathcal{D}$. To give a proof, we use a sequence of games (Game 0 to Game 5), and the construction of $\mathcal{S}$ is as in Figure 4.6.3.

We define Game 0 to Game 5 as follows:

Game 0: Game 0 is the same as SNM-RCCA-0 for $\mathcal{A}$ and $\mathcal{D}$. We denote the plaintext sampled in SNM-RCCA-0 as m_0.

Game 1: The difference from Game 0 is to create $(pk', sk') \leftarrow \mathsf{Gen}(1^{\lambda})$ newly and change the game so that (pk', sk') are used throughout the game. The input to $\mathcal{A}_1$ is changed to pk', the challenge ciphertext is generated using pk', and the oracles $\mathcal{O}_1$ and $\mathcal{O}_2$ that $\mathcal{A}$ accesses are changed to oracles that use sk'. In addition, the secret key used to decrypt ciphertexts c_i $(i = 0, \ldots, n)$ output by $\mathcal{A}_2$ is changed to sk'.

Game 2: The difference from Game 1 is that $m_1 \leftarrow \{0,1\}^{\ell}$ is sampled in addition to m_0. In additon, when decrypting c'_i, if $\boldsymbol{P}(m_0, \mathsf{Dec}(sk', c'_i)) = 1 \vee m_1 = \mathsf{Dec}(sk', c'_i)$, then let d_i be "Test".

Game 3: The difference from Game 2 is that $\mathcal{O}'_2$ returns "Test" when a ciphertext of m_1 is queried or a ciphertext of m satisfying $\boldsymbol{P}(m_0, m) = 1$ is queried.

Game 4: The difference from Game 3 is that the challenge ciphertext $c^* \leftarrow \mathsf{Enc}(pk', m_0)$ is changed to $c^* \leftarrow \mathsf{Enc}(pk', m_1)$.

Game 5: Game 5 is the same as SNM-RCCA-1 under PPTA $\mathcal{S}$ and pk.

```
$\underline{\mathcal{S}_1(pk)}$
$(pk', sk') \leftarrow \mathsf{Gen}(1^\lambda)$
$(\mathcal{M}, \boldsymbol{P}(\cdot,\cdot), st_1') \leftarrow \mathcal{A}_1^{\mathcal{O}_1'}(pk')$
$st_1 := (pk, pk', sk', st_1')$
output $(\mathcal{M}, \boldsymbol{P}(\cdot,\cdot), st_1)$
---
$\underline{\mathcal{S}_2^{\boldsymbol{P}(m_0,\cdot)}(h(m_0), st_1)}$
$m_1 \leftarrow \{0,1\}^\ell$
$c^* \leftarrow \mathsf{Enc}(pk', m_1)$
$(c_1', \ldots, c_n', st_2') \leftarrow \mathcal{A}_2^{\mathcal{O}_2'}(c^*, h(m_0), st_1')$
$st_2 := st_2'$
for $i = 1$ to $n$
    $d_i' := \begin{cases} \mathsf{Test} & (m_i' \leftarrow \mathsf{Dec}(sk', c_i'), \boldsymbol{P}(m_0, m_i') = 1 \\ & \vee m_i' = m_1) \\ \mathsf{Dec}(sk', c_i') & (\text{otherwise}) \end{cases}$
    $c_i := \begin{cases} \mathsf{Test} & (d_i' = \mathsf{Test}) \\ \perp & (d_i' = \perp) \\ \mathsf{Enc}(pk, d_i') & (\text{otherwise}) \end{cases}$
output $(c_1, \ldots, c_n, st_2)$
```

Figure 4.6.3: The construction of $\mathcal{S}$ used in Theorem 8

Let T_i be the event that 1 is output by $\mathcal{D}$ in Game i.

Lemma 4. *It holds that* $\Pr[T_1] = \Pr[T_0]$.

Proof. Game 1 is SNM-RCCA-0 under (pk', sk'), and (pk, sk) is not used. Although the key pair (pk, sk) is generated as $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$, pk is not input to $\mathcal{A}$ in Game 1. Therefore, Game 0 and Game 1 are equivalent from $\mathcal{A}$'s view. Thus, it holds that $\Pr[T_1] = \Pr[T_0]$. □

Lemma 5. *It holds that* $|\Pr[T_2] - \Pr[T_1]| < \frac{poly(\lambda)}{2^\ell}$.

Proof. Game 1 and Game 2 are identical if $\mathcal{A}_2$ does not output a ciphertext of m_1. Here, m_1 is chosen at uniformly random from $\{0,1\}^\ell$. Since the number of ciphertexts that $\mathcal{A}_2$ outputs is polynomial, it holds that

$$|\Pr[T_2] - \Pr[T_1]| < \frac{poly(\lambda)}{2^\ell}$$

using the difference lemma [72] and the union bound. □

Lemma 6. *It holds that* $|\Pr[T_3] - \Pr[T_2]| < \frac{poly(\lambda)}{2^\ell}$.

Proof. Game 2 and Game 3 are identical if $\mathcal{A}_2$ does not query a ciphertext of m_1 as a decryption query. Here, m_1 is chosen at uniformly random from $\{0,1\}^\ell$. Since the number of ciphertexts that $\mathcal{A}_2$ queries is polynomial, it holds that

$$|\Pr[T_3] - \Pr[T_2]| < \frac{poly(\lambda)}{2^\ell}$$

using the difference lemma and the union bound. □

Lemma 7. *There exists $\mathcal{B}$ such that* $|\Pr[T_4] - \Pr[T_3]| = \mathsf{Adv}^{\textit{INM-RCCA}}_{\Sigma,\mathcal{B}}(\lambda)$.

Proof. We construct a reduction $\mathcal{B}$ that breaks INM-RCCA security from $\mathcal{A}$. We construct an INM-RCCA adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3)$ who uses internally $\mathcal{A}$ and $\mathcal{D}$ as in Figure 4.6.4. When $\mathcal{A}_2$ queries a ciphertext c to $\mathcal{B}_2$, $\mathcal{B}_2$ sends c to the decryption oracle that he can access. Then, $\mathcal{B}$ receives m or "Test" from the oracle. After that, if $\boldsymbol{P}(m_0, m) = 1 \vee m = m_1$ or $\mathcal{B}_2$ receives "Test", then $\mathcal{B}_2$ sends "Test" to $\mathcal{A}_2$. Otherwise, $\mathcal{B}_2$ sends m to $\mathcal{A}_2$. We see that $\mathcal{B}$ simulate the decryption oracles in Game 3 and Game 4 for $\mathcal{A}$. Likewise, $\mathcal{B}_3$ can simulate the sequence of d_i which is input to $\mathcal{D}$ correctly.

When $\mathcal{B}$ runs in INM-RCCA-0, $\mathcal{B}$ simulates Game 3 for $\mathcal{A}$ and $\mathcal{D}$. Moreover, $\mathcal{B}$ outpus 1 only when $\mathcal{D}$ outputs 1. Therefore, we have

$$\Pr[\mathsf{Exp}^{\text{INM-RCCA-0}}_{\Sigma,\mathcal{B}}(\lambda) \to 1] = \Pr[T_3].$$

Similarly, we have

$$\Pr[\mathsf{Exp}^{\text{INM-RCCA-1}}_{\Sigma,\mathcal{B}}(\lambda) \to 1] = \Pr[T_4].$$

Therefore, we can derive

$$\begin{aligned} &|\Pr[T_4] - \Pr[T_3]| \\ &= |\Pr[\mathsf{Exp}^{\text{INM-RCCA-1}}_{\Sigma,\mathcal{B}}(\lambda) \to 1] \\ &\qquad - \Pr[\mathsf{Exp}^{\text{INM-RCCA-0}}_{\Sigma,\mathcal{B}}(\lambda) \to 1]| \\ &= \mathsf{Adv}^{\text{INM-RCCA}}_{\Sigma,\mathcal{B}}(\lambda). \end{aligned}$$

□

Lemma 8. *It holds that* $\Pr[T_5] = \Pr[T_4]$.

Proof. In Game 5, $\mathcal{S}$ uses $\mathcal{A}$ internally as in Figure 4.6.3. Since Game 5 is SNM-RCCA-1, $\mathcal{S}$ cannot access to the decryption oracle. However, $\mathcal{S}$ generates (pk', sk') internally, and he can access predicate oracle $\boldsymbol{P}(m_0, \cdot)$. In addition, an input to $\mathcal{A}$ is pk' as in Game 4, and thus $\mathcal{S}_2$ can respond the decryption query from $\mathcal{A}$ by using sk' and $\boldsymbol{P}(m_0, \cdot)$.

$\mathcal{S}_2$ inputs a ciphertext of m_1 which is generated internally in $\mathcal{S}_1$ to $\mathcal{A}_2$, and $\mathcal{A}_2$ outputs a sequence of ciphertexts. $\mathcal{S}_2$ decrypts them by using sk'. After that, each d_i is encrypted by using pk, and the encrypted sequence is output of $\mathcal{S}_2$. All ciphertexts are decrypted by using sk after $\mathcal{S}_2$ outputs them, and these sequence is input to $\mathcal{D}$. Here, the distributions of the inputs to $\mathcal{D}$ in Game 4 and Game 5 are identical. Thus, it holds that $\Pr[T_5] = \Pr[T_4]$. □

```
$\underline{\mathcal{B}_1^{\mathcal{O}'_1}(pk')}$
$(\mathcal{M}, \boldsymbol{P}(\cdot,\cdot), st'_1) \leftarrow \mathcal{A}_1^{\mathcal{O}'_1}(pk')$
$m_0 \leftarrow \mathcal{M}, m_1 \leftarrow \{0,1\}^\ell$
$st_1 := (m_0, m_1, \mathcal{M}, \boldsymbol{P}(\cdot,\cdot), st'_1)$
output $(m_0, m_1, st_1)$

$\underline{\mathcal{B}_2^{\mathcal{O}'_2}(c^*, st_1)}$
$(c'_1, \ldots, c'_n, st'_2) \leftarrow \mathcal{A}_2^{\mathcal{O}'_2}(c^*, h(m_0), st'_1)$
$st_2 := (m_0, m_1, \mathcal{M}, \boldsymbol{P}(\cdot,\cdot), st'_2)$
output $(c'_1, \ldots, c'_n, st_2)$

$\underline{\mathcal{B}_3(d_1, \ldots, d_n, st_2)}$
By using $m_0$ and $\boldsymbol{P}(\cdot,\cdot)$, check the value of $\boldsymbol{P}(m_0, d_i)$ for each $d_i$
$d_i$ that satisfies $\boldsymbol{P}(m_0, d_i) = 1 \vee d_1 = m_1$ is set as $d_i :=$"$\mathsf{Test}$" by the above procedure
$b' \leftarrow \mathcal{D}(\mathcal{M}, m_0, \boldsymbol{P}(\cdot,\cdot), d_1, \ldots, d_n, st'_2)$
output $b'$
```

Figure 4.6.4: The construction of $\mathcal{B}$ used in Lemma 7

By using Lemma 4 to Lemma 8, we can derive

$$\begin{aligned}
&\mathsf{Adv}^{\text{SNM-RCCA}}_{\Sigma,\mathcal{A},\mathcal{S},\mathcal{D},h}(\lambda) \\
&= |\Pr[\mathcal{D}(\mathsf{Exp}^{\text{SNM-RCCA-0}}_{\Sigma,\mathcal{A},h}(\lambda)) \to 1] \\
&\quad - \Pr[\mathcal{D}(\mathsf{Exp}^{\text{SNM-RCCA-1}}_{\Sigma,\mathcal{A},h}(\lambda)) \to 1]| \\
&= |\Pr[T_0] - \Pr[T_5]| \\
&= |\Pr[T_1] - \Pr[T_4]| \\
&= |\Pr[T_1] - \Pr[T_2] + \Pr[T_2] - \Pr[T_3] + \Pr[T_3] - \Pr[T_4]| \\
&\leq |\Pr[T_1] - \Pr[T_2]| + |\Pr[T_2] - \Pr[T_3]| \\
&\qquad + |\Pr[T_3] - \Pr[T_4]| \\
&\leq \frac{poly(\lambda)}{2^\ell} + \frac{poly(\lambda)}{2^\ell} + \mathsf{Adv}^{\text{INM-RCCA}}_{\Sigma,\mathcal{B}}(\lambda).
\end{aligned}$$

Since we assume that 2^ℓ is super polynomially large and Σ is INM-RCCA secure, for any $\mathcal{A}$, there exists $\mathcal{S}$ as in Fig 4.6.3 such that $\mathsf{Adv}^{\text{SNM-RCCA}}_{\Sigma,\mathcal{A},\mathcal{S},\mathcal{D},h}(\lambda)$ is negligible for any $\mathcal{D}$. □

The following theorem holds from Theorem 7 and Theorem 8.

Theorem 9. *If a PKE scheme* $\Sigma = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is INM-RCCA secure, then* Σ *is SNM-RCCA secure.*

4.6.2 SNM-RCCA implies INM-RCCA

We prove that SNM-RCCA implies INM-RCCA. Unlike the case of the proof that INM-RCCA implies SNM-RCCA, the following theorem does not need to make a case analysis depending on the size of the plaintext space.

Theorem 10. *If a PKE scheme* $\Sigma = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is SNM-RCCA secure, then* Σ *is INM-RCCA secure.*

```
$\mathcal{B}_1^{\mathcal{O}_1}(pk)$
$(m_0, m_1, st_1') \leftarrow \mathcal{A}_1^{\mathcal{O}_1}(pk)$
$\mathcal{M} := [\{m_0, m_1\}, \Pr(m_0) = \Pr(m_1) = 1/2]$
$\boldsymbol{P}(m, m') := \begin{cases} 1 & (m' \in \{m_0, m_1\}) \\ 0 & (\text{otherwise}) \end{cases}$
$st_1 := (m_0, m_1, \boldsymbol{P}(\cdot,\cdot), st_1')$
output $(\mathcal{M}, \boldsymbol{P}(\cdot,\cdot), st_1)$

$\mathcal{B}_2^{\mathcal{O}_2}(c^*, st_1)$
$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$
$(c_1, \ldots, c_n, st_2') \leftarrow \mathcal{A}_2^{\mathcal{O}_2}(c^*, st_1')$
$st_2 := (m_0, m_1, \boldsymbol{P}(\cdot,\cdot), st_2')$
output $(c_1, \ldots, c_n, st_2)$

$\mathcal{D}(\mathcal{M}, m, \boldsymbol{P}(\cdot,\cdot), d_1, \ldots, d_n, st_2)$
if $\|\mathcal{M}\| \neq 2$, then output 0
else if $\boldsymbol{P}'(m, m_0) = 0 \lor \boldsymbol{P}'(m, m_1) = 0$, then output 0,
   where $[\{m_0, m_1\}, \Pr(m_0) = \Pr(m_1) = 1/2] = \mathcal{M}$
else if $b' \leftarrow \mathcal{A}_3(d_1, \ldots, d_n, st_2') \land m = m_{b'}$, then output 1
else then output 0
```

Figure 4.6.1: The constructions of $\mathcal{B}$ and $\mathcal{D}$ used in Theorem 10

Proof. We denote $\mathsf{Exp}_{\Sigma,\mathcal{A}}^{\text{INM-RCCA}}$ as the experiment that chooses the challenge bit b randomly and excute INM-RCCA-b. Without loss of generality, we can assume

$$\Pr[\mathsf{Exp}_{\Sigma,\mathcal{A}}^{\text{INM-RCCA}}(\lambda) \to b] \geq 1/2$$

for any INM-RCCA adversary $\mathcal{A}$. It is because if

$$\Pr[\mathsf{Exp}_{\Sigma,\mathcal{A}}^{\text{INM-RCCA}}(\lambda) \to b] < 1/2,$$

then we consider the adversary $\mathcal{A}'$ whose output is the reverse of $\mathcal{A}$'s output. Then, the advantage of $\mathcal{A}'$ is same as $\mathcal{A}$, and it holds

$$\Pr[\mathsf{Exp}_{\Sigma,\mathcal{A}'}^{\text{INM-RCCA-b}}(\lambda) \to b] \geq 1/2.$$

Thus, if we can bound the advantage of $\mathcal{A}'$, it means we can bound the advantage of $\mathcal{A}$ at the same time.

We assume that for any SNM-RCCA adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ and for any polynomial time computable function h, there exists a simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ such that $\mathsf{Adv}_{\Sigma,\mathcal{B},\mathcal{D},h}^{\text{SNM-RCCA}}(\lambda)$ is negligible for any distinguisher $\mathcal{D}$. Then, we show $\mathsf{Adv}_{\Sigma,\mathcal{A}}^{\text{INM-RCCA}}(\lambda)$ is negligible for any INM-RCCA adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$.

We consider the SNM-RCCA-0 experiment with $h : m \mapsto \epsilon$, where ϵ is the empty string, and we construct an SNM-RCCA adversary $\mathcal{B}$ and a distinguisher $\mathcal{D}$ who uses $\mathcal{A}$ internally as in Figure 4.6.1. When $\mathcal{A}_2$ submits a decryption query c to $\mathcal{B}_2$, $\mathcal{B}_2$ submits c to the decryption oracle that $\mathcal{B}_2$ can access. Then $\mathcal{B}_2$ sends the response from the oracle to $\mathcal{A}_2$.

By the construction of $\mathcal{B}$ and $\mathcal{D}$ above, $\mathcal{D}$ outputs 1 when $\mathcal{A}$ guesses bit b which is chosen in the experiment correctly. Thus, we can derive

$$\Pr[\mathcal{D}(\mathsf{Exp}^{\text{SNM-RCCA-0}}_{\Sigma,\mathcal{B},h}(\lambda)) \to 1] = \Pr[\mathsf{Exp}^{\text{INM-RCCA-b}}_{\Sigma,\mathcal{A}}(\lambda) \to b].$$

Let E be the event that $\mathcal{S}$ outputs $\mathcal{M}$ and $\mathcal{S}$ such that $\|\mathcal{M}\| = 2$ and $\boldsymbol{P}(m, m_0) = 1 \wedge \boldsymbol{P}(m, m_1) = 1$, and p be the probability that E occurs in SNM-RCCA-1, where $[\{m_0, m_1\}, \Pr(m_0) = \Pr(m_1) = 1/2] = \mathcal{M}$. When the event E occurs in SNM-RCCA-1, $\mathcal{S}$ does not receive the challenge ciphertext, and he cannot obtain any information about the choice of m_0 and m_1 even if he access the predicate oracle. Thus, since $\|\mathcal{M}\| = 2$ when E occurs, it holds that

$$\begin{aligned}
&\Pr\left[\mathcal{D}\left(\mathsf{Exp}^{\text{SNM-RCCA-1}}_{\Sigma,\mathcal{S},h}(\lambda)\right) \to 1\right] && (4.1)\\
&= \Pr\left[\mathcal{D}\left(\mathsf{Exp}^{\text{SNM-RCCA-1}}_{\Sigma,\mathcal{S},h}(\lambda)\right) \to 1 \middle| E\right] \cdot \Pr[E] \\
&= \frac{p}{2} \leq \frac{1}{2}. && (4.2)
\end{aligned}$$

Here, $\mathsf{Adv}^{\text{INM-RCCA}}_{\Sigma,\mathcal{A}}(\lambda)$ can be rewritten as follows.

$$\begin{aligned}
&\mathsf{Adv}^{\text{INM-RCCA}}_{\Sigma,\mathcal{A}}(\lambda)\\
&= |\Pr[\mathsf{Exp}^{\text{INM-RCCA-0}}_{\Sigma,\mathcal{A}}(\lambda) \to 1]\\
&\qquad - \Pr[\mathsf{Exp}^{\text{INM-RCCA-1}}_{\Sigma,\mathcal{A}}(\lambda) \to 1]|\\
&= |2 \cdot \Pr[\mathsf{Exp}^{\text{INM-RCCA}}_{\Sigma,\mathcal{A}}(\lambda) \to b] - 1|.
\end{aligned}$$

Then, it holds that

$$\begin{aligned}
&\mathsf{Adv}^{\text{INM-RCCA}}_{\Sigma,\mathcal{A}}(\lambda)\\
&= |2 \cdot \Pr[\mathsf{Exp}^{\text{INM-RCCA}}_{\Sigma,\mathcal{A}}(\lambda) \to b] - 1|\\
&= |2 \cdot \Pr[\mathcal{D}(\mathsf{Exp}^{\text{SNM-RCCA-0}}_{\Sigma,\mathcal{B},h}(\lambda)) \to 1] - 1|\\
&\leq |2 \cdot \Pr[\mathcal{D}(\mathsf{Exp}^{\text{SNM-RCCA-0}}_{\Sigma,\mathcal{B},h}(\lambda)) \to 1] - 2 \cdot p/2|\\
&= 2(|\Pr[\mathcal{D}(\mathsf{Exp}^{\text{SNM-RCCA-0}}_{\Sigma,\mathcal{B},h}(\lambda)) \to 1]\\
&\qquad - \Pr[\mathcal{D}(\mathsf{Exp}^{\text{SNM-RCCA-1}}_{\Sigma,\mathcal{S},h}(\lambda)) \to 1])\\
&= 2 \cdot \mathsf{Adv}^{\text{SNM-RCCA}}_{\Sigma,\mathcal{B},\mathcal{D},h}(\lambda).
\end{aligned}$$

The transformation of the third equality is derived from the fact that we can assume the advantage of $\mathcal{A}$ is greater than 1/2. Therefore, for any PPTA $\mathcal{A}$, $\mathsf{Adv}^{\text{INM-RCCA}}_{\Sigma,\mathcal{A}}(\lambda)$ is negligible. □

4.7 Definition of SS-RCCA and Its Equivalence with IND-RCCA

In this section, we propose the definition of semantic security under the RCCA environment (SS-RCCA) in a similar way as the definition of SNM-RCCA. Since we need to consider the simulator in SS-RCCA like the case of SNM-RCCA, we allow the simulator to access the predicate oracle.

We give our definition of SS-RCCA as follows.

Let $\Sigma = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme, $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ be pairs of PPTAs, and h and f be polynomial time computable functions. We consider the following experiments SS-RCCA-0 and SS-RCCA-1:

```
$\underline{\mathrm{Exp}^{\text{SS-RCCA-0}}_{\Sigma,\mathcal{A},h,f}(\lambda)}$
$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$;
$(\mathcal{M}, \boldsymbol{P}(\cdot,\cdot), st_1) \leftarrow \mathcal{A}_1^{\mathcal{O}_1}(pk)$;
$m \leftarrow \mathcal{M}$;
$c^* \leftarrow \mathsf{Enc}(pk, m)$;
$v \leftarrow \mathcal{A}_2^{\mathcal{O}_2}(c^*, h(m), st_1)$;
if $v = f(m)$, then $\beta := 1$
else $\beta := 0$
output $(\mathcal{M}, \boldsymbol{P}(\cdot,\cdot), \beta)$
```

```
$\underline{\mathrm{Exp}^{\text{SS-RCCA-1}}_{\Sigma,\mathcal{S},h,f}(\lambda)}$
$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$;
$(\mathcal{M}, \boldsymbol{P}(\cdot,\cdot), st_1) \leftarrow \mathcal{S}_1(pk)$;
$m \leftarrow \mathcal{M}$;
$c^* \leftarrow \mathsf{Enc}(pk, m)$;
$v \leftarrow \mathcal{S}_2^{\boldsymbol{P}(m,\cdot)}(h(m), st_1)$;
if $v = f(m)$, then $\beta := 1$
else $\beta := 0$
output $(\mathcal{M}, \boldsymbol{P}(\cdot,\cdot), \beta)$
```

where, a predicate $\boldsymbol{P}$ satisfies $\boldsymbol{P}(m, m) = 1$ for any m which is included in the support of $\mathcal{M}$, and

$$\mathcal{O}_1(c) = \mathsf{Dec}(sk, c),$$
$$\mathcal{O}_2(c) = \begin{cases} \mathsf{Test} & (\boldsymbol{P}(m, \mathsf{Dec}(sk, c)) = 1) \\ \mathsf{Dec}(sk, c) & (\text{otherwise}). \end{cases}$$

In the above two experiments, $\mathcal{M}$ is a distribution over the plaintext space.

We define the advantage as

$$\begin{aligned}\mathsf{Adv}^{\text{SS-RCCA}}_{\Sigma,\mathcal{A},\mathcal{S},\mathcal{D},h,f}(\lambda) :=& |\Pr[\mathcal{D}(\mathsf{Exp}^{\text{SS-RCCA-0}}_{\Sigma,\mathcal{A},h,f}(\lambda)) \to 1] \\ &- \Pr[\mathcal{D}(\mathsf{Exp}^{\text{SS-RCCA-1}}_{\Sigma,\mathcal{S},h,f}(\lambda)) \to 1]|.\end{aligned}$$

Definition 18 (SS-RCCA security)**.** *We say that Σ is SS-RCCA secure if for any polynomial time computable function h and f, and for any pair of PPTAs $\mathcal{A}$, there exists a simulator $\mathcal{S}$ such that $\mathsf{Adv}^{\text{SS-RCCA}}_{\Sigma,\mathcal{A},\mathcal{S},\mathcal{D},h,f}(\lambda)$ is negligible for any PPTA $\mathcal{D}$.*

4.7.1 IND-RCCA implies SS-RCCA

We prove that IND-RCCA implies SS-RCCA by a case analysis. Like the proofs of Theorem 7 and Theorem 8, we consider two cases that the size of the plaintext space which a PKE scheme Σ supports is polynomial or not.

Theorem 11. *If a PKE scheme $\Sigma = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is IND-RCCA secure, and the size of the plaintext space of Σ is polynomial, then Σ is SS-RCCA secure.*

In the proof of Theorem 11, we use a sequence of games (Game 0 to Game 3). We can prove can this theorem in the same way as proof of Theorem 7.

Proof. We assume $\mathsf{Adv}^{\text{IND-RCCA}}_{\Sigma,\mathcal{B}}(\lambda)$ is negligible for any IND-RCCA adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$. Then, we show for any SS-RCCA adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and for any polynomial time computable function h and f, there exists a simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ such that $\mathsf{Adv}^{\text{SS-RCCA}}_{\Sigma,\mathcal{A},\mathcal{S},\mathcal{D},h,f}$

```
$\mathcal{S}_1(pk)$
$(pk', sk') \leftarrow \mathsf{Gen}(1^\lambda)$
$(\mathcal{M}, \boldsymbol{P}(\cdot,\cdot), st_1') \leftarrow \mathcal{A}_1^{\mathcal{O}_1'}(pk')$
$st_1 := (pk, pk', sk', st_1')$
output $(\mathcal{M}, \boldsymbol{P}(\cdot,\cdot), st_1')$

$\mathcal{S}_2^{\boldsymbol{P}(m_0,\cdot)}(h(m_0), st_1)$
$m_1 \leftarrow \mathcal{P}_{m_0}$
$c^* \leftarrow \mathsf{Enc}(pk', m_1)$
$v \leftarrow \mathcal{A}_2^{\mathcal{O}_2'}(c^*, h(m_0), st_1')$
output $v$
```

Figure 4.7.1: The construction of $\mathcal{S}$ used in Theorem 11

(λ) is negligible for any distinguisher $\mathcal{D}$. To give a proof, we use a sequence of games (Game 0 to Game 3), and the construction of $\mathcal{S}$ is as in Figure 4.7.1

We define the Game 0 to Game 3 as follows:

Game 0: Game 0 is the same as SS-RCCA-0 for $\mathcal{A}$ and $\mathcal{D}$. We denote the plaintext sampled in SS-RCCA-0 as m_0.

Game 1: The difference from Game 0 is to create $(pk', sk') \leftarrow \mathsf{Gen}(1^\lambda)$ newly and change the game so that (pk', sk') are used throughout the game. The input to $\mathcal{A}_1$ is changed to pk', the challenge ciphertext is generated using pk', and the oracles $\mathcal{O}_1$ and $\mathcal{O}_2$ that $\mathcal{A}$ accesses are changed to the oracles that use sk'.

Game 2: The difference from Game 1 is that $m_1 \leftarrow \mathcal{P}_{m_0}$ is sampled in addition to m_0, where $\mathcal{P}_{m_0}$ is the uniform distribution over all plaintexts m' satisfying $\boldsymbol{P}(m_0, m') = 1$. In addition, the challenge ciphertext $c^* \leftarrow \mathsf{Enc}(pk', m_0)$ is changed to $c^* \leftarrow \mathsf{Enc}(pk', m_1)$.

Game 3: Game 3 is the same as SS-RCCA-1 under PPTA $\mathcal{S}$ and pk.

We can sample efficiently from $\mathcal{P}_{m_0}$, which is used in Game 2 and $\mathcal{S}_2$, in the same way as in the proof of Theorem 7.

Let T_i be the event that 1 is output by $\mathcal{D}$ in Game i.

Lemma 9. *It holds that* $\Pr[T_1] = \Pr[T_0]$.

Proof. Game 1 is SS-RCCA-0 under (pk', sk'), and (pk, sk) is not used. Although the key pair (pk, sk) is generated as $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$, pk is not input to $\mathcal{A}$ in Game 1. Therefore, Game 0 and Game 1 are equivalent from $\mathcal{A}$'s view. Thus, it holds that $\Pr[T_1] = \Pr[T_0]$. □

Lemma 10. *There exists* $\mathcal{B}$ *such that* $|\Pr[T_2] - \Pr[T_1]| = \mathsf{Adv}_{\Sigma,\mathcal{B}}^{\textit{IND-RCCA}}(\lambda)$.

Proof. We construct an IND-RCCA adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ under (pk', sk') who uses internally $\mathcal{A}$ and $\mathcal{D}$ as in Figure 4.7.2. When $\mathcal{A}_2$ queries a ciphertext c to $\mathcal{B}_2$, $\mathcal{B}_2$ sends c to the decryption oracle that he can access. Then, $\mathcal{B}$ receives m or "Test" from the

```
$\underline{\mathcal{B}_1^{\mathcal{O}'_1}(pk')}$
$(\mathcal{M}, \boldsymbol{P}(\cdot,\cdot), st'_1) \leftarrow \mathcal{A}_1^{\mathcal{O}'_1}(pk')$
$m_0 \leftarrow \mathcal{M}, m_1 \leftarrow \mathcal{P}_{m_0}$
$st_1 := (m_0, m_1, \boldsymbol{P}(\cdot,\cdot), \mathcal{M}, st'_1)$
output $(m_0, m_1, st_1)$
$\underline{\mathcal{B}_2^{\mathcal{O}'_2}(c^*, st_1)}$
$v \leftarrow \mathcal{A}_2^{\mathcal{O}'_2}(c^*, h(m_0), st'_1)$
if $v = f(m_0)$, then $\beta := 1$
else $\beta := 0$
$b' \leftarrow \mathcal{D}(\mathcal{M}, \boldsymbol{P}(\cdot,\cdot), \beta)$
output $b$
```

Figure 4.7.2: The construction of $\mathcal{B}$ used in Lemma 10

oracle. After that, if $\boldsymbol{P}(m_0, m) = 1$ or $\mathcal{B}_2$ receives "Test", then $\mathcal{B}_2$ sends "Test" to $\mathcal{A}_2$. Otherwise, $\mathcal{B}_2$ sends m to $\mathcal{A}_2$. Since the plaintext m_1 which is generated in $\mathcal{B}_1$ satisfies $\boldsymbol{P}(m_0, m_1) = 1$, $\mathcal{B}_2$ can simulate the decryption oracle correctly.

When $\mathcal{B}$ runs in IND-RCCA-0, $\mathcal{B}$ simulates Game 1 for $\mathcal{A}$ and $\mathcal{D}$. Moreover, $\mathcal{B}$ outpus 1 when $\mathcal{D}$ outputs 1. Therefore, we have

$$\Pr[\mathsf{Exp}_{\Sigma,\mathcal{B}}^{\text{IND-RCCA-0}}(\lambda) \to 1] = \Pr[T_1].$$

Similarly, we have

$$\Pr[\mathsf{Exp}_{\Sigma,\mathcal{B}}^{\text{IND-RCCA-1}}(\lambda) \to 1] = \Pr[T_2].$$

Therefore, we can derive

$$\begin{aligned}
&|\Pr[T_2] - \Pr[T_1]| \\
&= |\Pr[\mathsf{Exp}_{\Sigma,\mathcal{B}}^{\text{IND-RCCA-1}}(\lambda) \to 1] \\
&\qquad - \Pr[\mathsf{Exp}_{\Sigma,\mathcal{B}}^{\text{IND-RCCA-0}}(\lambda) \to 1]| \\
&= \mathsf{Adv}_{\Sigma,\mathcal{B}}^{\text{IND-RCCA}}(\lambda).
\end{aligned}$$

□

In above proof, we do not need to consider the case that an adversary $\mathcal{A}$ chooses a predicate $\boldsymbol{P}$ such that $\boldsymbol{P}(m_0, m') = 1$ only when $m_0 = m'$ like as the proof of Theorem 7. *It is trivially secure without assuming IND-RCCA.*

Lemma 11. *It holds that* $\Pr[T_3] = \Pr[T_2]$.

Proof. In Game 3, $\mathcal{S}$ uses $\mathcal{A}$ internally as in Figure 4.7.1. Since Game 3 is SS-RCCA-1, $\mathcal{S}$ cannot access to the decryption oracle. However, $\mathcal{S}$ generates (pk', sk') internally, and he can access predicate oracle $\boldsymbol{P}(m_0, \cdot)$. In addition, an input to $\mathcal{A}$ is pk' as in Game 2, and thus $\mathcal{S}_2$ can respond decryption queries from $\mathcal{A}$ by using sk' and $\boldsymbol{P}(m_0, \cdot)$.

$\mathcal{S}_2$ inputs a ciphertext of m_1 which is generated internally in $\mathcal{S}_2$ to $\mathcal{A}_2$, and $\mathcal{A}_2$ outputs v. $\mathcal{S}_2$ outputs this v. After that, v is input to $\mathcal{D}$. Here, the distributions of the inputs to $\mathcal{D}$ in Game 2 and Game 3 are identical. Thus, it holds that $\Pr[T_3] = \Pr[T_2]$. □

By using Lemma 9 to Lemma 11, we can derive

$$\begin{aligned}
&\mathsf{Adv}^{\text{SS-RCCA}}_{\Sigma,\mathcal{A},\mathcal{S},\mathcal{D},h,f}(\lambda)\\
&= |\Pr[\mathcal{D}(\mathsf{Exp}^{\text{SS-RCCA-0}}_{\Sigma,\mathcal{A},h,f}(\lambda)) \to 1]\\
&\qquad - \Pr[\mathcal{D}(\mathsf{Exp}^{\text{SS-RCCA-1}}_{\Sigma,\mathcal{A},h,f}(\lambda)) \to 1]|\\
&= |\Pr[T_0] - \Pr[T_3]|\\
&= |\Pr[T_1] - \Pr[T_2]|\\
&= \mathsf{Adv}^{\text{IND-RCCA}}_{\Sigma,\mathcal{B}}(\lambda).
\end{aligned}$$

Therefore, for any $\mathcal{A}$, there exists $\mathcal{S}$ as in Figure 4.7.1 such that $\mathsf{Adv}^{\text{SS-RCCA}}_{\Sigma,\mathcal{A},\mathcal{S},\mathcal{D},h,f}(\lambda)$. □

Then, we give a proof IND-RCCA implies SS-RCCA when the size of plaintext space is larger than polynomial.

Theorem 12. *If a PKE scheme* $\Sigma = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is IND-RCCA secure, and the size of the plaintext space of* Σ *is super polynomially large, then* Σ *is SS-RCCA secure.*

In the proof of Theorem 12, we use a sequence of games (Game 0 to Game 4). We can prove can this theorem in the same way as proof of Theorem 8.

Proof. Let the plaintext space of Σ be $\{0,1\}^\ell$. We assume $\mathsf{Adv}^{\text{IND-RCCA}}_{\Sigma,\mathcal{B}}(\lambda)$ is negligible for any IND-RCCA adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$. Then, we show for any SS-RCCA adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and for any polynomial time computable function h and f, there exists a simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ such that $\mathsf{Adv}^{\text{SS-RCCA}}_{\Sigma,\mathcal{A},\mathcal{S},\mathcal{D},h,f}(\lambda)$ is negligible for any distinguisher $\mathcal{D}$. To give a proof, we use a sequence of games (Game 0 to Game 4), and the construction of $\mathcal{S}$ is as in Figure 4.7.3.

We define Game 0 to Game 4 as follows:

Game 0: Game 0 is the same as SS-RCCA-0 for $\mathcal{A}$ and $\mathcal{D}$. We denote the plaintext sampled in SS-RCCA-0 as m_0.

Game 1: The difference from Game 0 is to create $(pk', sk') \leftarrow \mathsf{Gen}(1^\lambda)$ newly and change the game so that (pk', sk') are used throughout the game. The input to $\mathcal{A}_1$ is changed to pk', and the challenge ciphertext is generated using pk'. In addition, the oracles $\mathcal{O}_1$ and $\mathcal{O}_2$ that $\mathcal{A}$ accesses are changed to the oracles that use sk'.

Game 2: The difference from Game 2 is that $m_1 \leftarrow \{0,1\}^\ell$ is sampled in addition to m_0. In addition, $\mathcal{O}'_2$ returns "Test" when a ciphertext of m_1 is queried or a ciphertext of m satisfying $\boldsymbol{P}(m_0, m) = 1$ is queried.

Game 3: The difference from Game 2 is that the challenge ciphertext $c^* \leftarrow \mathsf{Enc}(pk', m_0)$ is changed to $c^* \leftarrow \mathsf{Enc}(pk', m_1)$.

Game 4: Game 4 is the same as SS-RCCA-1 under PPTA $\mathcal{S}$ and pk.

Let T_i be the event that 1 is output by $\mathcal{D}$ in Game i.

Lemma 12. *It holds that* $\Pr[T_1] = \Pr[T_0]$.

```
$\mathcal{S}_1(pk)$
$(pk', sk') \leftarrow \mathsf{Gen}(1^\lambda)$
$(\mathcal{M}, \boldsymbol{P}(\cdot,\cdot), st'_1) \leftarrow \mathcal{A}_1^{\mathcal{O}'_1}(pk')$
$st_1 := (pk, pk', sk', st'_1)$
output $(\mathcal{M}, \boldsymbol{P}(\cdot,\cdot), st'_1)$
$\mathcal{S}_2^{\boldsymbol{P}(m_0,\cdot)}(h(m_0), st_1)$
$m_1 \leftarrow \{0,1\}^\ell$
$c^* \leftarrow \mathsf{Enc}(pk', m_1)$
$v \leftarrow \mathcal{A}_2^{\mathcal{O}'_2}(c^*, h(m_0), st'_1)$
output $v$
```

Figure 4.7.3: The construction of $\mathcal{S}$ used in Theorem 12

Proof. Game 1 is SS-RCCA-0 under (pk', sk'), and (pk, sk) is not used. Although the key pair (pk, sk) is generated as $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$, pk is not input to $\mathcal{A}$ in Game 1. Therefore, Game 0 and Game 1 are equivalent from $\mathcal{A}$'s view. Thus, it holds that $\Pr[T_1] = \Pr[T_0]$. □

Lemma 13. *It holds that* $|\Pr[T_2] - \Pr[T_1]| < \frac{poly(\lambda)}{2^\ell}$.

Proof. Game 2 and Game 3 are identical if $\mathcal{A}_2$ does not query a ciphertext of m_1. Here, m_1 is chosen at uniformly random from $\{0,1\}^\ell$. Since the number of ciphertexts that $\mathcal{A}_2$ queries is polynomial, it holds that

$$|\Pr[T_2] - \Pr[T_1]| < \frac{poly(\lambda)}{2^\ell}$$

using the difference lemma and the union bound. □

Lemma 14. *There exists $\mathcal{B}$ such that* $|\Pr[T_3] - \Pr[T_2]| = \mathsf{Adv}_{\Sigma,\mathcal{B}}^{IND\text{-}RCCA}(\lambda)$.

Proof. We construct an IND-RCCA adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ who uses internally $\mathcal{A}$ and $\mathcal{D}$ as in Figure 4.7.4. When $\mathcal{A}_2$ queries a ciphertext c to $\mathcal{B}_2$, $\mathcal{B}_2$ sends c to the decryption oracle that he can access. Then, $\mathcal{B}$ receives m or "Test" from the oracle. After that, if $\boldsymbol{P}(m_0, m) = 1 \vee m = m_1$ or $\mathcal{B}_2$ receives "Test", then $\mathcal{B}_2$ sends "Test" to $\mathcal{A}_2$. Otherwise, $\mathcal{B}_2$ sends m to $\mathcal{A}_2$. We see that $\mathcal{B}$ simulates the decryption oracles in Game 2 and Game 3 for $\mathcal{A}$.

When $\mathcal{B}$ runs in IND-RCCA-0, $\mathcal{B}$ simulates Game 2 for $\mathcal{A}$ and $\mathcal{D}$. Moreover, $\mathcal{B}$ outpus 1 when $\mathcal{D}$ outputs 1. Therefore, we have

$$\Pr[\mathsf{Exp}_{\Sigma,\mathcal{B}}^{\text{IND-RCCA-0}}(\lambda) \to 1] = \Pr[T_2].$$

Similarly, we have

$$\Pr[\mathsf{Exp}_{\Sigma,\mathcal{B}}^{\text{IND-RCCA-1}}(\lambda) \to 1] = \Pr[T_3].$$

Therefore, we can derive

$$\begin{aligned}
&|\Pr[T_3] - \Pr[T_2]| \\
&= |\Pr[\mathsf{Exp}_{\Sigma,\mathcal{B}}^{\text{IND-RCCA-1}}(\lambda) \to 1] \\
&\qquad - \Pr[\mathsf{Exp}_{\Sigma,\mathcal{B}}^{\text{IND-RCCA-0}}(\lambda) \to 1]| \\
&= \mathsf{Adv}_{\Sigma,\mathcal{B}}^{\text{IND-RCCA}}(\lambda).
\end{aligned}$$

```
$\mathcal{B}_1^{\mathcal{O}'_1}(pk')$
$(\mathcal{M}, \boldsymbol{P}(\cdot,\cdot), st'_1) \leftarrow \mathcal{A}_1^{\mathcal{O}'_1}(pk')$
$m_0 \leftarrow \mathcal{M}, m_1 \leftarrow \{0,1\}^\ell$
$st_1 := (m_0, m_1, \mathcal{M}, \boldsymbol{P}(\cdot,\cdot), st'_1)$
output $(m_0, m_1, st_1)$

$\mathcal{B}_2^{\mathcal{O}'_2}(c^*, st_1)$
$v \leftarrow \mathcal{A}_2^{\mathcal{O}'_2}(c^*, h(m_0), st'_1)$
if $v = f(m_0)$, then $\beta := 1$
else $\beta := 0$
$b' \leftarrow \mathcal{D}(\mathcal{M}, \beta)$
output $b'$
```

Figure 4.7.4: The construction of $\mathcal{B}$ used in Lemma 14

□

Lemma 15. *It holds that* $\Pr[T_4] = \Pr[T_3]$.

Proof. In Game 4, $\mathcal{S}$ uses $\mathcal{A}$ internally as in Figure 4.7.3. Since Game 4 is SS-RCCA-1, $\mathcal{S}$ cannot access to the decryption oracle. However, $\mathcal{S}$ generates (pk', sk') internally, and he can access the predicate oracle $\boldsymbol{P}(m_0, \cdot)$. In addition, the input to $\mathcal{A}$ is pk' as in Game 3, and thus $\mathcal{S}_2$ can respond decryption queries from $\mathcal{A}$ using sk' and $\boldsymbol{P}(m_0, \cdot)$.

$\mathcal{S}_2$ inputs a ciphertext of m_1 which is generated internally in $\mathcal{S}_2$ to $\mathcal{A}_2$, and $\mathcal{A}_2$ outputs v. $\mathcal{S}_2$ outputs this v. After that, v is input to $\mathcal{D}$. Here, the distributions of the inputs to $\mathcal{D}$ in Game 3 and Game 4 are identical. Thus, it holds that $\Pr[T_4] = \Pr[T_3]$. □

By using Lemma 12 to Lemma 15, we can derive

$$\begin{aligned}
&\mathsf{Adv}^{\text{SS-RCCA}}_{\Sigma,\mathcal{A},\mathcal{S},\mathcal{D},h,f}(\lambda)\\
&= |\Pr[\mathcal{D}(\mathsf{Exp}^{\text{SS-RCCA-0}}_{\Sigma,\mathcal{A},h,f}(\lambda)) \to 1]\\
&\qquad - \Pr[\mathcal{D}(\mathsf{Exp}^{\text{SS-RCCA-1}}_{\Sigma,\mathcal{A},h,f}(\lambda)) \to 1]|\\
&= |\Pr[T_0] - \Pr[T_4]|\\
&= |\Pr[T_1] - \Pr[T_3]|\\
&= |\Pr[T_1] - \Pr[T_2] + \Pr[T_2] - \Pr[T_3]\\
&\le |\Pr[T_1] - \Pr[T_2]| + |\Pr[T_2] - \Pr[T_3]|\\
&\le \frac{poly(\lambda)}{2^\ell} + \mathsf{Adv}^{\text{IND-RCCA}}_{\Sigma,\mathcal{B}}(\lambda).
\end{aligned}$$

Since we assume that 2^ℓ is super-polynomially large and Σ is IND-RCCA secure, for any $\mathcal{A}$, there exists $\mathcal{S}$ as in Figure 4.7.1 such that $\mathsf{Adv}^{\text{SS-RCCA}}_{\Sigma,\mathcal{A},\mathcal{S},\mathcal{D},h,f}(\lambda)$. □

The following theorem holds from Theorem 11 and Theorem 12.

Theorem 13. *If a PKE scheme* $\Sigma = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is IND-RCCA secure, then* Σ *is SS-RCCA secure.*

4.7.2 SS-RCCA implies IND-RCCA

We prove that SS-RCCA implies IND-RCCA. Unlike the case of the proof that IND-RCCA implies SS-RCCA, the following theorem does not need to make a case analysis depending on the size of the plaintext space.

Theorem 14. *If a PKE scheme* $\Sigma = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is SS-RCCA secure, then* Σ *is IND-RCCA secure.*

Proof. We denote $\mathsf{Exp}^{\text{IND-RCCA}}_{\Sigma,\mathcal{A}}$ as the experiment that chooses the challenge bit b randomly and excute IND-RCCA-b. Without loss of generality, we can assume

$$\Pr[\mathsf{Exp}^{\text{IND-RCCA}}_{\Sigma,\mathcal{A}}(\lambda) \to b] \geq 1/2$$

for any IND-RCCA adversary. It is because if

$$\Pr[\mathsf{Exp}^{\text{IND-RCCA}}_{\Sigma,\mathcal{A}}(\lambda) \to b] < 1/2,$$

then we consider the adversary $\mathcal{A}'$ whose output is the reverse of $\mathcal{A}$'s output. Then, the advantage of $\mathcal{A}'$ is same as $\mathcal{A}$, and it holds

$$\Pr[\mathsf{Exp}^{\text{IND-RCCA-b}}_{\Sigma,\mathcal{A}'}(\lambda) \to b] \geq 1/2.$$

Thus, if we can bound the advantage of $\mathcal{A}'$, it means we can bound the advantage of $\mathcal{A}$ at the same time.

We assume that for any SS-RCCA adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ and for any polynomial time computable function h and f, there exists a simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ such that $\mathsf{Adv}^{\text{SS-RCCA}}_{\Sigma,\mathcal{B},\mathcal{S},\mathcal{D},h,f}(\lambda)$ is negligible for any distinguisher $\mathcal{D}$. Then, we show $\mathsf{Adv}^{\text{IND-RCCA}}_{\Sigma,\mathcal{A}}(\lambda)$ is negligible for any IND-RCCA adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$.

We consider the SS-RCCA-0 experiment with $h : m \mapsto \epsilon$, where ϵ is the empty string, and we construct an SS-RCCA adversary $\mathcal{B}$ and a distinguisher $\mathcal{D}$ who uses $\mathcal{A}$ internally as in Figure 4.7.1. When $\mathcal{A}_2$ submits a decryption query c to $\mathcal{B}_2$, $\mathcal{B}_2$ submits c to the decryption oracle that $\mathcal{B}_2$ can access. Then $\mathcal{B}_2$ sends the response from the oracle to $\mathcal{A}_2$.

By the construction of $\mathcal{B}$ and $\mathcal{D}$ above, $\mathcal{D}$ outputs 1 when $\mathcal{A}$ guesses the bit b which is chosen in the experiment correctly. Thus, we can derive

$$\Pr[\mathcal{D}(\mathsf{Exp}^{\text{SS-RCCA-0}}_{\Sigma,\mathcal{B},h,f}(\lambda)) \to 1] = \Pr[\mathsf{Exp}^{\text{IND-RCCA-}b}_{\Sigma,\mathcal{A}}(\lambda) \to b].$$

Let E be the event that $\mathcal{S}$ outputs $\mathcal{M}$ such that $\|\mathcal{M}\| = 2$ and $\boldsymbol{P}(m, m_0) = 1 \wedge \boldsymbol{P}(m, m_1) = 1$, and p be the probability that E occurs in SS-RCCA-1. When the event E occurs in SS-RCCA-1, $\mathcal{S}$ does not receive the challenge ciphertext, and he cannot obtain any information about the choice of m_0 and m_1 even if he access the predicate oracle. Thus, since $\|\mathcal{M}\| = 2$ when E occurs, it holds that

$$\begin{aligned} &\Pr\left[\mathcal{D}\left(\mathsf{Exp}^{\text{SS-RCCA-1}}_{\Sigma,\mathcal{S},h,f}(\lambda)\right) \to 1\right] \\ &= \Pr\left[\mathcal{D}\left(\mathsf{Exp}^{\text{SS-RCCA-1}}_{\Sigma,\mathcal{S},h,f}(\lambda)\right) \to 1 \middle| E\right] \cdot \Pr[E] \\ &= \frac{p}{2} \leq \frac{1}{2}. \end{aligned} \tag{4.3}$$

```
$\underline{\mathcal{B}_1^{\mathcal{O}_1}(pk)}$
$(m_0, m_1, st_1') \leftarrow \mathcal{A}_1^{\mathcal{O}_1}(pk)$
$\mathcal{M} := [\{m_0, m_1\}, \Pr(m_0) = \Pr(m_1) = 1/2]$
$\boldsymbol{P}'(m, m') := \begin{cases} 1 & (\boldsymbol{P}(m') = 1) \\ 0 & (\text{otherwise}) \end{cases}$
$st_1 := (m_0, m_1, \boldsymbol{P}(\cdot), \boldsymbol{P}'(\cdot,\cdot), st_1')$
output $(\mathcal{M}, \boldsymbol{P}'(\cdot,\cdot), st_1)$

$\underline{\mathcal{B}_2^{\mathcal{O}_2}(c^*, st_1)}$
$b' \leftarrow \mathcal{A}_2^{\mathcal{O}_2}(c^*, st_1')$
$v := m_{b'}$
output $v$

$\underline{\mathcal{D}(\mathcal{M}, \boldsymbol{P}'(\cdot,\cdot), \beta)}$
if $\|\mathcal{M}\| \neq 2$, then output 0
else if $\boldsymbol{P}'(m_1, m_0) = 0 \vee \boldsymbol{P}'(m_0, m_1) = 0$, then output 0,
  where $[\{m_0, m_1\}, \Pr(m_0) = \Pr(m_1) = 1/2] = \mathcal{M}$
else then output $\beta$
```

Figure 4.7.1: The constructions of $\mathcal{B}$ and $\mathcal{D}$ used in Theorem 14

Here, $\mathsf{Adv}_{\Sigma,\mathcal{A}}^{\text{IND-RCCA}}(\lambda)$ can be rewritten as follows.

$$\begin{aligned}
&\mathsf{Adv}_{\Sigma,\mathcal{A}}^{\text{IND-RCCA}}(\lambda) \\
&= |\Pr[\mathsf{Exp}_{\Sigma,\mathcal{A}}^{\text{IND-RCCA-0}}(\lambda) \to 1] \\
&\qquad - \Pr[\mathsf{Exp}_{\Sigma,\mathcal{A}}^{\text{IND-RCCA-1}}(\lambda) \to 1]| \\
&= |2 \cdot \Pr[\mathsf{Exp}_{\Sigma,\mathcal{A}}^{\text{IND-RCCA}}(\lambda) \to b] - 1|.
\end{aligned}$$

Then, it holds that

$$\begin{aligned}
&\mathsf{Adv}_{\Sigma,\mathcal{A}}^{\text{IND-RCCA}}(\lambda) \\
&= |2 \cdot \Pr[\mathsf{Exp}_{\Sigma,\mathcal{A}}^{\text{IND-RCCA}}(\lambda) \to b] - 1| \\
&= |2 \cdot \Pr[\mathcal{D}(\mathsf{Exp}_{\Sigma,\mathcal{B},h,f}^{\text{SS-RCCA-0}}(\lambda)) \to 1] - 1| \\
&\leq |2 \cdot \Pr[\mathcal{D}(\mathsf{Exp}_{\Sigma,\mathcal{B},h,f}^{\text{SS-RCCA-0}}(\lambda)) \to 1] - 2 \cdot p/2| \\
&= 2(|\Pr[\mathcal{D}(\mathsf{Exp}_{\Sigma,\mathcal{B},h,f}^{\text{SS-RCCA-0}}(\lambda)) \to 1] \\
&\qquad - \Pr[\mathcal{D}(\mathsf{Exp}_{\Sigma,\mathcal{S},h,f}^{\text{SS-RCCA-1}}(\lambda)) \to 1]) \\
&= 2 \cdot \mathsf{Adv}_{\Sigma,\mathcal{B},\mathcal{D},h,f}^{\text{SS-RCCA}}(\lambda).
\end{aligned}$$

The transformation of the third equality is derived from the fact that we can assume the advantage of $\mathcal{A}$ is greater than 1/2. Therefore, for any PPTA $\mathcal{A}$, $\mathsf{Adv}_{\Sigma,\mathcal{A}}^{\text{IND-RCCA}}(\lambda)$ is negligible. □

4.8 Conclusion

PEKS allows us to perform a keyword search on encrypted databases. When considering the practical application of PEKS, it would be beneficial to add more functionalities to

PEKS. For example, PRES is the encryption scheme that combines the features of PEKS and PRE. It is known that we can construct a more flexible mail routing service using PRES. PRES is more useful than PEKS, but the additional processes of transforming the ciphertext require more careful thought about security and adversary models. When we consider the security game that is a natural extension of the IND-CCA game in the PKE setting, the adversary can trivially win the game. To prevent trivial attacks, we usually use RCCA as an adversary model when we consider the security for PRE or PRES.

RCCA security was introduced in order to handle the security of encryption schemes that are non-malleable except tampering which preserves the plaintext. Therefore, the rigorous definition of non-malleability is crucial in the context of RCCA security. However, at first glance, NM-RCCA proposed by Canetti et al. [46] seems not to properly capture requirements for the non-malleability. In this chapter, we formulated simulation-based non-malleability and indistinguishability-based non-malleability under the RCCA environment by extending the standard definitions of non-malleability, and we proved that these two proposed security notions and IND-RCCA proposed by Canetti et al. are all equivalents (regardless of the size of plaintext space). Especially, by showing the equivalence between the definitions of IND-RCCA and SNM-RCCA, where the latter is the most strict notion of non-malleability for the RCCA setting, it becomes clear that it is sufficient to prove IND-RCCA when giving a proof for the non-malleability against RCCA in the most strict sense.

Since all the results presented in this chapter are for the PKE setting, simulation-based and game-based formulations of NM-RCCA for the PRES setting are future work. Also, clarifying the relationship among the security notion for PRES is one of the future works.

Chapter 5

New Security Notion for Private Information Retrieval Supporting Range Queries

5.1 Introduction

5.1.1 Background and Motivation

We dealt with information retrieval on encrypted data in Chapters 3 and 4. However, there are also databases whose data is not encrypted. While the data itself might not be private, the information regarding a client's queries might be. For example, investors searching for information regarding companies and stock prices, might involuntarily leak their investment interests and intentions through their queries. It is conceivable that a malicious data manager collects statistical data from the client's queries and attempt to exploit this information.

To prevent such attacks, private information retrieval (PIR) was proposed [73]. Using PIR, a client can retrieve data from a database without the database server learning what is being retrieved. A trivial way to achieve PIR would be for the client to download all data from the database. However, since this trivial approach would incur communication cost $O(n)$ for the client, assuming the size of the database is n, this solution quickly becomes unreasonable when we consider larger databases. Hence, a PIR scheme is required to have communication cost lower than $O(n)$.

The first PIR scheme was proposed by Chor et al. [73]. Their construction assumed that many servers hold a replicated database, and that the servers do not communicate with each other. A PIR scheme constructed under these assumptions is called a multi-server PIR scheme. A PIR scheme relying on just a single server, which is technically more difficult to construct, was first achieved by Kushilevitz and Ostrovsky [74]. After that, several works on constructing single-server PIR and multi-server PIR schemes have been introduced, gradually improving the communication cost of PIR [75, 76, 77]. However, while PIR provides strong security guarantees, the standard definition of PIR only considers queries that retrieve a single element, and do not consider other often used queries types, such as basic range queries.

In contrast, in the somewhat related area of encrypted databases, most schemes aim at providing functionality approaching standard SQL, including range queries [78, 79]. However, note that even setting aside the problem of how data would be encrypted and decrypted, an encrypted database would not address the privacy concerns considered in a PIR scheme, as the aim is only to protect the confidentiality of the data against a malicious server, and no attempts are done in these scheme to hide the access pattern by clients. Furthermore, several attacks reconstructing the underlying data or partial information about this, based on the functionality of encrypted database schemes, have been discovered, e.g., attacks based on information leakage in searchable encryption [80], and volume attacks based on observing only the volume of answers to the range queries [81, 82, 83].

An interesting recent scheme that provides a SQL-like functionality, but still aims at preserving the privacy of client queries, is the private query scheme by Wang et al. [84]. While the scheme does not directly support unbounded range queries, by combining the supported TOPK and COUNT queries of their scheme, we can implement the range query functionality we consider in this paper. The scheme is based on a two-server setup, and uses function secret sharing [12] to generate and respond to queries. Essentially, each server will only receive a share of a function that extracts the relevant information the client is interested in obtaining, and evaluate that share over his own copy of the database. By the security property of function secret sharing, the server will not learn what function is being evaluated, but correctness allows the client to combine the evaluation results from the two servers, to obtain the output of the function. While Wang et al. [84] do not provide any formal security models or security proofs for their construction, it is plausible that their construction will satisfy our simple extension of PIR security to range queries, defined in Section 5.2. However, it is relatively easy to see that the structure of their scheme leaks the kind of queries a client is making, and for range queries, the number of elements returned by the server. As discussed below, this can potentially be problematic with respect to maintaining query privacy.

5.1.2 Our Contribution

In this chapter, we focus on PIR schemes that simultaneously provide strong security and functionality beyond simple standard PIR. Specifically, we consider schemes supporting range queries, which is one of the most frequently used queries for online data analytics [85, 86]. Only very few works seem to have a similar focus (Wang et al. [84] being an exception). In addition to this, to the best of our knowledge, all query privacy preserving schemes that do support some kind of range queries, are not formally shown secure.

Firstly, we formalize PIR schemes supporting both standard PIR queries as well as range queries, and introduce corresponding security models. More specifically, we define three security notions. The first notion captures ordinary PIR security i.e., when a client request just a single element at a given position in the database (which we denote an index query), the server(s) does not learn what element is being retrieved. The second notion, which is a simple extension of the first notion to range queries, captures that when a client request all database entries x satisfying $a \leq x \leq b$ for chosen bounds (a, b), the server(s) does not learn what elements are being retrieved i.e., the server cannot distinguish this query from any other range query containing the same number of elements. However, we note that this notion might not be sufficient to protect query privacy in some scenarios. For example, consider a simple database consisting of five elements; three distinct elements (x, y, z) as well as two additional elements (z', z'') identical to the third element i.e., $z = z' = z''$. For this particular database, any range query resulting in a three elements response, can only have been for ranges including z, but not x or y; any query resulting in two elements must have been for a range including x and y; and any query resulting in a single element, must have been for ranges include either x or y, but not both. In other words, the privacy of the range queries is almost completely lost, if the fact that a range query is made, and the number of elements in the response, leaks to the server. This is the case for the scheme by Wang et al. [84]. Furthermore, if additional information regarding the distribution of queries a client is likely to make, is available to the server, deriving what queries the client makes becomes even easier.

While this type of information leakage might seem inherent to range queries, we define a third notion aimed at addressing this. This notion, which we call query indistinguishability, captures that the server(s) cannot distinguish between range queries and an appropriate number of independent index queries. This adds an additional layer of security, in particular if multiple queries (ideally from multiple indistinguishable clients) are done simultaneously. In other words, this notion ensures that the server(s) cannot detect range queries (or the boundaries between different range queries), and that server(s) can only obtain an overall estimate of the size of the data transferred in all queries by the client(s). This can greatly reduce the ability of the server(s) to infer information about client queries. We note that the definition of query indistinguishability addresses the *structural* information leakage with respect to range queries, but, like most other cryptographic security definitions, does not address *temporal* information leakage i.e. what servers might infer from the timing of the queries made by the clients. In Section 5.3, we discuss ways to address this.

Having defined the above three notions of security, we show that query indistinguishability implies the other two, i.e., schemes shown to satisfy query indistinguishability will also satisfy ordinary PIR security as well as the simple extension to range queries. We then show a simple generic construction of a PIR scheme satisfying query indistinguishability from a standard PIR scheme. This scheme has a range query round complexity with a multiplicative overhead of $O(k + \log n)$, where n is the size of the database and k is the number of elements retrieved in the range query, compared to the underlying PIR scheme. Lastly, we give a direct construction of a multi-server PIR scheme supporting range queries based on function secret sharing. Our construction takes a similar approach to the private query scheme of Wang et al. [84], but whereas the scheme from [84] is not formally shown secure, and can potentially only achieve the simple extension of PIR security to range queries, our construction is shown to satisfy query indistinguishability. In contrast to the generic constructions, the round complexity of the direct construction is $2 + k$. We additionally implemented the client and server components of our scheme, and performed various performance measurements. These show that the time required to process a range query containing 50 elements from a database containing 7.5 million elements, is about 200 seconds. The details of this are discussed in Section 5.7.

5.1.3 Related Works

Multi-server PIR Supporting Index Queries: Chor et al. [73] firstly proposed the construction of information theoretically secure multi-server schemes. More specifically, they proposed a 2-server scheme and a k-server scheme. Subsequent research has proposed the ways to improve the amount of communication between the client and the servers [87, 88, 89, 90].

Chor and Gilboa [91] firstly proposed a computationally secure multi-server PIR scheme from the quadratic residue problem. After their proposal, many computationally secure multi-server PIR schemes are proposed from various assumptions [91, 92].

Table 5.1 below summarizes the results of existing studies on multi-server PIR schemes.

Table 5.1: Known results and efficiency for multi-server PIR schemes

Ref	Servers	Communication Cost	Security
GKG95 [73]	2	$O(n^{1/3})$	Information Theoretical
BIK05 [87]	2	$O(n^{1/3})$	Information Theoretical
DG15 [88]	2	$n^{O(\sqrt{\log\log n/\log n})}$	Information Theoretical
GKG95 [73]	k	$O(n^{1/k})$	Information Theoretical
Amb97 [89]	k	$O(n^{\frac{1}{2k-1}})$	Information Theoretical
BIKR02 [90]	k	$n^{O(\frac{\log\log k}{k\log k})}$	Information Theoretical
CG97 [91]	2	$O(n^{\epsilon})$ $(\epsilon > 0)$	Computational
GI14 [92]	2	$polylog(n)$	Computational

Single-server PIR Supporting Index Queries: Chor et al. [73] proved that the information theoretical secure single-server PIR scheme always requires at least $O(n)$ communication cost between the client and the server. Therefore, the trivial way that the client downloads all data from the server is optimal when we consider the information theoretical secure single-server PIR scheme. Kushilevitz and Ostrovsky [74] firstly proposed the computationally secure single-server scheme based on the quadratic residuosity assumption. After that, many single-server PIR schemes are proposed from various assumptions [76, 93, 94, 77].

Table 5.2 below summarizes the results of existing studies on single-server PIR schemes.

Table 5.2: Known results and efficiency for single-server PIR schemes

Ref	Communication Cost	Security Assumption
KO97 [74]	$O(n^{\epsilon})$ $(\epsilon > 0)$	QR Problem
CMS99 [76]	$polylog(n)$	Φ Hiding
GR05 [93]	$\log^2 n$	Φ Hiding
DSH14 [94]	$\log n$	Security of NTRU
DC14 [77]	$\log\log n$	Security of BGV FHE

PIR Supporting Flexible Queries: While most PIR schemes support only index queries, there are a few exceptions in the literature. Chor et al. [95] proposed a PIR scheme supporting keyword search queries. Tillem et al. [96] proposed a PIR scheme supporting range queries, and Wang et al. [84], highlighted above, proposed PIR schemes providing functionality approaching standard SQL, including range queries. We note that the latter two works do not formally define security and provide security proofs. Furthermore, neither of these schemes satisfy query indistinguishability, and it is unclear whether the scheme by Tillem et al. even satisfies our simple security notion for range queries.

The concept of multi-query PIR proposed by Groth et al. [97] allows multiple elements to be retrieved simultaneously. Groth et al. [97] gave an information-theoretic lower bound on the communication of any multi-query PIR scheme, as well as a construction matching this bound. We note, however, that in multi-query PIR, it is assumed that the client knows the (possibly independent) indices of the elements to be retrieved, whereas, in a range query, no such assumption is made. Hence, multi-query PIR schemes and PIR schemes supporting range queries are not directly comparable.

Reducing Server-side Computation: The bottleneck in the execution of the PIR protocol is the amount of computation on the server-side. It is known that when the client and the server execute the PIR protocol, the server needs to perform $O(n)$ computation, where n is the size of the database. If the amount of computation performed by the server is less than n, it means that the there is at least 1 bit that the server did not use to compute the reply to the client. Then, the server can learn that the client is not interested in those bits. Therefore, to maintain the privacy of the client's queries, the server always performs $O(n)$ computation.

Beimel et al. [98] proposed a multi-server PIR with preprocessing scheme to reduce server-side computation cost. Regarding the single-server scheme, Canetti et al. [99] firstly proposed the preprocessing scheme. In addition to these studies, research on constructing the preprocessing scheme to reduce server-side computation is continuing [100, 101].

5.1.4 Chapter Organization

The structure of this chapter is as follows. In Section 5.2, we define the syntax and security models for multi-server PIR schemes supporting range queries, and in Section 5.3, we discuss information leakage due to the timing of queries and how to address this. In Section 5.4, we prove relations among the introduced security notions. In Section 5.5, we show a generic construction of a PIR scheme supporting range queries from a standard PIR scheme. In Section 5.6, we show our efficient constructions of a PIR scheme supporting range queries using function secret sharing. In Section 5.7, we show experimental results regarding the efficiency of our scheme. In Section 5.8, we conclude this chapter.

5.2 PIR Schemes Supporting Range Queries

Parameters. The following is a list of parameters we will use in this chapter:

- ℓ: number of servers.
- n: size of the database (number of elements).
- V: size of each element.

ID	Value
1	10
2	23
$\vdots$	$\vdots$
$n-1$	110
n	120

Figure 5.2.1: An example of database of size n: The elements in the database is sorted in ascending order.

We denote probabilistic polynomial time algorithm by PPTA, denote PIR supporting range queries by RQ-PIR. In this section, we define syntax and security models for PIR schemes supporting range queries. In the following, we will treat a database as consisting of an n-entry vector $\overrightarrow{x} = (x_1, \dots, x_n)$, and each entry as a V-bit integer. Furthermore, we will assume that the entries in the database are sorted in ascending order i.e. it holds that $x_1 \leq x_2 \leq \cdots \leq x_n$. We attach an implicit ID to each item, and we set the ID of i-th as i $(i = 1, \dots, n)$.

5.2.1 Model

Our notion of a RQ-PIR scheme supports two types of queries: index queries and range queries. In an index query, the client specifies an index $i \in [n]$, and obtains the ith entry in the database i.e. x_i. However, in a range query, the client specifies a range by values $a, b \in \mathbb{N}$ $(a < b)$, and obtains all entries x_j in the database satisfying $a \leq x_j \leq b$. Note that the client might be unaware of the indices j of the elements retrieved in a range query.

To capture interactive schemes, we define a RQ-PIR scheme via stateful algorithms. Note, however, that we only require the client to maintain state. Specifically, only the algorithms *Index* and *Range* defined below, which the client will run to make an index or range query, respectively, will be stateful, whereas the algorithm *Res* run by the servers to respond to the clients request, will be stateless.

$(\vec{q}, st') \leftarrow Index(1^\lambda, \overrightarrow{ans}, st)$: The *Index* algorithm is a stateful interactive algorithm run by the client to execute an index query. The algorithm takes as input the security parameter 1^λ, potential previous answers $\overrightarrow{ans} := (ans_1, ans_2, \dots, ans_\ell)$ from servers, where ans_j is the answer from server j, and state st. The algorithm outputs server queries $\overrightarrow{q} := (q_1, q_2, \dots, q_\ell)$, and new state st'. The client sets the initial state to $st := \{i\}$, where i is the index of the entry in the database he would retrieve, and sets the initial $\overrightarrow{ans} \leftarrow \overrightarrow{\perp}$. When *Index* outputs $\vec{q} = \overrightarrow{\perp}$, it indicates termination of the index query, and the client outputs st as the final output y.

$(\vec{q}, st') \leftarrow Range(1^\lambda, \overrightarrow{ans}, st)$: The *Range* algorithm is a stateful interactive algorithm run by the client to execute a range query. The algorithm takes as input the security parameter 1^λ, potential previous answers $\overrightarrow{ans} := (ans_1, ans_2, \dots, ans_\ell)$ from servers, where ans_j is the answer from server j, and state st. The algorithm outputs queries $\overrightarrow{q} := (q_1, q_2, \dots, q_\ell)$, and new state st'. The client sets the initial state to $st := \{a, b\}$, where $a, b \in \mathbb{N}, a \leq b$ and $[a, b]$ is the range that client wants to retrieve from database, and sets the initial $\overrightarrow{ans} \leftarrow \overrightarrow{\perp}$. When *Range* outputs $\vec{q} = \overrightarrow{\perp}$, it indicates termination of the range query, and client outputs st as a final output $\vec{y}$.

$ans_j \leftarrow Res(1^\lambda, \vec{x}, j, q)$**:** The *Res* algorithm is stateless and run by each server to respond to the clients queries. The algorithm takes as input the security parameter 1^λ, database $\vec{x}$, the identifier of the server $j \in [\ell]$, and query q from the client, and outputs answer ans_j.

To simplify notation, we will often omit the security parameter 1^λ from the input of the above defined algorithms. In addition to this, we will use $Res_j(\vec{x}, q)$ to denote $Res(1^\lambda, \vec{x}, j, q)$.

Based on the above algorithms, we obtain protocols for index and range queries by respectively combining *Index* and *Res*, and *Range* and *Res*. We will use the following notation regarding these:

$(y, \overrightarrow{\perp}) \leftarrow \langle Index, Res_1, \dots, Res_\ell \rangle (i, \vec{x}, \dots, \vec{x})$**:** This denotes the client running the *Index* algorithm with input the initial state $st_0 := \{i\}$, and server j replying to each query q_j from client by running $Res_j(\vec{x}, q_j)$, for all $j \in [\ell]$, until the *Index* algorithm halts. After completing the protocol, the client outputs y and each server receives no output (i.e. $\perp$).

$(\vec{y}, \overrightarrow{\perp}) \leftarrow \langle Range, Res_1, \dots, Res_\ell \rangle ((a, b), \vec{x}, \dots, \vec{x})$**:** This denotes the client running the *Range* algorithm with input initial state $st_0 := \{a, b\}$, and server j replying to queries q_j from client by running $Res_j(\vec{x}, q_j)$, for all $j \in [\ell]$ until the *Range* algorithm halts. After completing the protocol, the client outputs $\vec{y}$ and each server outputs $\perp$.

For the above protocols, we will consider the transcript of an interaction between a client and a server: for any $j \in [\ell]$ and for any $i \in [n]$, we denote by $\mathsf{Trans}_j(\langle Index, Res_1, \dots, Res_\ell \rangle (i, \vec{x}, \dots, \vec{x}))$ the messages sent between the client and server j when executing the above defined protocol for retrieving the database entry for index i (held by the client). Likewise, for any $j \in [\ell]$ and for any $a, b \in \mathbb{N}$, we denote by $\mathsf{Trans}_j(\langle Range, Res_1, \dots, Res_\ell \rangle ((a, b), \vec{x}, \dots, \vec{x}))$ the messages sent between client and server j during the execution of the above defined protocol for retrieving values in the range $[a, b]$.

5.2.2 Security Definitions

We will now define three security models for PIR schemes that support index queries and range queries. In all of these models, we assume that serves do not collude, and there is a secure channel between the client and each server.

At first, we define security of index queries for RQ-PIR schemes. This security notion is equivalent to the computational notion normally considered for standard (multi-server) PIR, and captures that the servers do not learn anything regarding the element being retrieved in the index query.

Definition 19. *A RQ-PIR scheme provides secure index queries if for all $\lambda \in \mathbb{N}$, for any database $\vec{x} = (x_1, \dots, x_n)$ of size n, for any server $m \in [\ell]$, for any indices $i, j \in [n]$, for any PPTA distinguisher $\mathcal{D}$,*

$$|\Pr[\mathcal{D}(\mathsf{View}_i^m) = 1] - \Pr[\mathcal{D}(\mathsf{View}_j^m) = 1]|$$

is negligible, where

$$\mathsf{View}_k^m \leftarrow \{\mathsf{Trans}_m(\langle Index, Res_1, \ldots, Res_\ell \rangle(k, \vec{x}, \ldots, \vec{x})), \vec{x}, r\}$$

for $k \in \{i, j\}$, and where r is the randomnesses used by server m during the execution of the protocol.

We now define a simple extension of the above security notion aimed at capturing security of range queries. This security notion captures that the servers do not learn the bounds (a, b) of the range query, and what elements are being retrieved via an indistinguishability requirement: any server should be unable to distinguish two different range queries as long as the number of elements in the query is the same.

Definition 20. *A RQ-PIR scheme provides secure range queries if for all $\lambda \in \mathbb{N}$, for any database $\vec{x} = (x_1, \ldots, x_n)$ of size n,for any server $m \in [\ell]$, for any bounds $a, b, c, d \in [n]$ such that $|\{x_i \mid a \leq x_i \leq b\}| = |\{x_i \mid c \leq x_i \leq d\}| = k$, for any PPTA distinguisher $\mathcal{D}$,*

$$|\Pr[\mathcal{D}(\mathsf{View}_{a,b}^m) = 1] - \Pr[\mathcal{D}(\mathsf{View}_{c,d}^m) = 1]|$$

is negligible, where

$$\mathsf{View}_{h_0,h_1}^m \leftarrow \{\mathsf{Trans}_m(\langle Range, Res_1, \ldots, Res_\ell \rangle((h_0, h_1), \vec{x}, \ldots, \vec{x})), \vec{x}, r\}$$

for $(h_0, h_1) \in \{(a, b), (c, d)\}$, and where r is the randomnesses of server m during the execution of the protocol.

While Definition 20 above intuitively guarantees the security of range queries, this security notion does not guarantee that the type of query being made, or the number of elements in a range query, are hidden. As discussed in the introduction, that might lead to the privacy of range queries being compromised. Thus, a stronger security notion is desirable.

Hence, we define query indistinguishability of a RQ-PIR scheme aimed at addressing this. This security notion captures that the servers cannot learn whether the client is performing a range query or a number of independent index queries for a set of arbitrary unrelated set of entries in the database.

Definition 21. *A RQ-PIR scheme provides query indistinguishability if there exists a polynomial time computable function $f : \mathbb{N} \to \mathbb{N}$ such that for all $\lambda \in \mathbb{N}$, for any database $\vec{x} = (x_1, \ldots, x_n)$ of size n, for any server $m \in [\ell]$, for any bounds $a, b \in \mathbb{N}$, for any st of indices $i_1, \ldots i_{f(k)} \in [n]$ where $k = |\{x_i \mid a \leq x_i \leq b\}|$, for any PPTA distinguisher $\mathcal{D}$,*

$$|\Pr[\mathcal{D}(\mathsf{View}_{range}^m) = 1] - \Pr[\mathcal{D}(\mathsf{View}_{index}^m) = 1]|$$

is negligible, where

$$\mathsf{View}_{range}^m \leftarrow \{\mathsf{Trans}_m(\langle Range, Res_1, \ldots, Res_\ell \rangle((a, b), \vec{x}, \ldots, \vec{x})), \vec{x}, r\},$$

and

$$\mathsf{View}_{index}^m \leftarrow \mathsf{View}_{i_1} || \cdots || \mathsf{View}_{i_{f(k)}},$$

where $\mathsf{View}_s \leftarrow \{\mathsf{Trans}_m(\langle Index, Res_1, \ldots, Res_\ell \rangle(k, \vec{x}, \ldots, \vec{x})), \vec{x}, r\}$, *and* $s \in \{i_1, \ldots, i_{f(k)}\}$ *and r is the randomnesses of server m during the execution of the protocol.*

Note that while Definition 21 guarantees that database server(s) cannot detect range queries (or boundaries between these) from the queries alone, the definition by itself does not address information derived from the timing of queries. In Section 5.3, we discuss ways to address this.

5.3 Discussion

In both the introduction and in Section 5.2, it was highlighted that the definition of query indistinguishability addresses the structural part of the problem of hiding range queries, but does not, by itself, address information leakage due to the timing of queries. Furthermore, it was left open how to take advantage of multiple clients accessing the same server. In this section, we will provide an informal discussion of this.

Query indistinguishability guarantees that the server(s) cannot tell from the queries alone, whether a client makes a set of index queries, a single or multiple range queries, or a combination of these. However, under the assumption that a client will always wait a certain amount of time between each query, and that the individual steps that range queries are comprised of, are executed immediately, the server(s) will be able to infer from the timing of the queries whether a range query is being made or not, and potentially the amount of data transfer in the range query.

To address this, clients might adopt a number of different countermeasures. The perhaps simplest of these, is for the client to adopt a constant query rate, in which all index queries and each step of range queries, are executed at a constant rate. Additionally, adding dummy queries to maintain the query rate in between real queries will eliminate information leakage due to the timing of queries. However, the drawback of this approach is that if the query rate is high, dummy queries might cause a significant overhead for servers, as these would have to be processed like ordinary queries, and if the query rate is low, a delay with respect to the completion of range queries will be introduced, which might be significant if large amounts of data are retrieved.

A different approach is to group queries from different clients via a mechanism that will hide from the server which queries belong to which clients. This will leave the server(s) unable to analyze the query pattern of individual clients, and given a sufficient number of clients generating various queries, this can prevent the server from inferring what type of queries clients are making.

An easily conceivable but naive approach to this, is to use a proxy server for queries. The clients submit their queries to the proxy server, which will group queries for a given time interval, and then forward these to the database server. When the database server responds, the proxy server would forward the appropriate responses to the appropriate clients. Note, however, this merely moves the problem of protecting query privacy from the database server itself to the proxy server. While a proxy server without access to the database itself might be able to infer less about the queries made by clients, it would still be able to detect whether or not range queries are made, and estimate the amount of data being retrieved.

A potential way to resolve this issue, is to use an approach similar to mix networks (e.g., the Tor network [102]). In a mix network, the origin of a message is disguised by routing it through various mixing servers, and each intermediate server will not be able to determine the source of the message. Note, however, that to avoid the same issue that arose when using a proxy server, the client must distribute their index queries and range query steps across different entry nodes in the mix network. In order to maximize the number of mixing servers and entry points, it is conceivable that each client would act as a mixing server, and randomly distribute his own queries among all participating clients and servers, who would then route the queries through the mix network to the database server. This is very similar to the approach taken in user-private information retrieval (UPIR) [103]. In UPIR, multiple clients form a P2P network with a shared memory, and the clients forward each other's queries to the database, thereby preventing the database from learning the identity of the user who sent a particular query. A full analysis of this type of construction is outside the scope of this paper, and is left as future work.

Lastly, note that query rate limitation and mixing of client queries can easily be combined.

5.4 Relation Among Security Notions

In this section, we prove implications among the security notions introduced above. Specifically, we prove that query indistinguishability implies secure index queries, as well as secure range queries.

Theorem 15. *If RQ-PIR scheme* Π *provides query indistinguishability, then* Π *provides secure range queries.*

Proof. For all λ, for any database $\vec{x} = (x_1, \dots, x_n)$ of size n, for any $m \in [\ell]$, for any $a, b, c, d \in [n]$ such that $|\{x_i \mid a \leq x_i \leq b\}| = |\{x_i \mid c \leq d \leq b\}| = k$, we consider a PPTA $\mathcal{D}$ against the security for range queries in RQ-PIR scheme Π. The advantage of $\mathcal{D}$ is defined by

$$\mathsf{Adv}^{range}_{\mathcal{D},\Pi} = |\Pr[\mathcal{D}(\mathsf{View}^m_{a,b}) = 1] - \Pr[\mathcal{D}(\mathsf{View}^m_{c,d}) = 1]|.$$

To obtain a proof, we use a sequence of games (**Game** 0 to **Game** 2).

Game 0**:** This game corresponds to the client and servers running $\langle Range, Res_1, \dots, Res_\ell \rangle ((a, b), \vec{x}, \dots, \vec{x})$.

Game 1**:** This game corresponds to the client and servers running $\langle Index, Res_1, \dots, Res_\ell \rangle (i_1, \vec{x}, \dots, \vec{x}), \dots, \langle Index, Res_1, \dots, Res_\ell \rangle (i_{f(k)}, \vec{x}, \dots, \vec{x})$, where $k := |\{x_i \mid a \leq x_i \leq b\}|$ and $i_1, \dots, i_{f(k)}$ are chosen randomly from $[n]$.

Game 2**:** This game corresponds to the client and servers running $\langle Range, Res_1, \dots, Res_\ell \rangle ((c, d), \vec{x}, \dots, \vec{x})$.

For all r, we denote by View^m_r the view for server m generated in **Game** r.

Lemma 16. *If RQ-PIR scheme* Π *provides query indistinguishability, then for any* $r \in \{1, 2\}$ *and for any PPTA* $\mathcal{B}$*, it holds that the difference between the probability that* $\mathcal{B}$ *outputs 1 in* **Game** $r - 1$ *and* r *is negligible.*

Proof. We consider a PPTA $\mathcal{B}$ who distinguishes View^m_{r-1} from View^m_r, and denote the advantage of $\mathcal{B}$ by

$$\mathsf{Adv}^r_{\mathcal{B},\Pi} = |\Pr[\mathcal{B}(\mathsf{View}^m_r) = 1] - \Pr[\mathcal{B}(\mathsf{View}^m_{r-1}) = 1]|.$$

Then, since View^m_0 and View^m_2 are same as View^m_{range} in the definition of query indistinguishability, and View^m_1 is same as View^m_{index}, we can see $\mathcal{B}$ as an adversary against query indistinguishability for Π. Thus, we can conclude

$$\mathsf{Adv}^r_{\mathcal{B},\Pi} = |\Pr[\mathcal{B}(\mathsf{View}^m_r) = 1] - \Pr[\mathcal{B}(\mathsf{View}^m_{r-1}) = 1]|$$

is negligible. □

By using Lemma 19, we can derive

$$\begin{aligned}
&|\Pr[\mathcal{D}(\mathsf{View}^m_{a,b}) = 1] - \Pr[\mathcal{D}(\mathsf{View}^m_{c,d}) = 1]| \\
&\leq \sum_{r=1}^{2} |\Pr[\mathcal{D}(\mathsf{View}^m_{r-1}) = 1] - \Pr[\mathcal{D}(\mathsf{View}^m_r) = 1]| \\
&\leq 2 \cdot negl.
\end{aligned}$$

□

Then, we prove that query indistinguishability implies secure index queries. To show this implication, we introduce the security notion of secure index queries for sets. At first, we prove that query indistinguishability implies secure index queries for sets, and then show that index queries for sets implies secure index queries.

Definition 22. *A RQ-PIR scheme provides secure index queries for sets if for all $\lambda \in \mathbb{N}$, for any database $\vec{x} = (x_1, \ldots, x_n)$ of size n, for any server $m \in [\ell]$, for any set of indices $\vec{i_1} = (i_{1_1}, \ldots, i_{1_k}), \vec{i_2} = (i_{2_1}, \ldots, i_{2_k}) \in [n]^k$, for any PPTA distinguisher $\mathcal{D}$,*

$$|\Pr[\mathcal{D}(\mathsf{View}^m_{\vec{i_1}}) = 1] - \Pr[\mathcal{D}(\mathsf{View}^m_{\vec{i_2}}) = 1]|$$

is negligible, where

$$\begin{aligned}
\mathsf{View}^m_{k} \leftarrow \{&\mathsf{Trans}_m(\langle Index, Res_1, \ldots, Res_\ell\rangle(i_{t_1}, \vec{x}, \ldots, \vec{x})), \vec{x}, r_1\}|| \cdots \\
&||\{\mathsf{Trans}_m(\langle Index, Res_1, \ldots, Res_\ell\rangle(i_{t_k}, \vec{x}, \ldots, \vec{x})), \vec{x}, r_k\}
\end{aligned}$$

for $t \in \{1, 2\}$, and where $r_1, \ldots, r_k$ is the randomnesses used by server m during the execution of the protocol.

Theorem 16. *If RQ-PIR scheme Π provides query indistinguishability, then Π provides secure index queries for sets.*

Proof. To obtain a proof, we use a sequence of games (**Game** 0 to **Game** 2).

Game 0**:** This game corresponds to the client and servers running $\langle Index, Res_1, \ldots, Res_\ell\rangle$ $(i_{1_1}, \vec{x}, \ldots, \vec{x})|| \cdots ||\langle Index, Res_1, \ldots, Res_\ell\rangle(i_{1_k}, \vec{x}, \ldots, \vec{x})$.

Game 1**:** This game corresponds to the client and servers running $\langle Range, Res_1, \ldots, Res_\ell\rangle$ $((a, b), \vec{x}, \ldots, \vec{x})$ such that $|\{x_i \mid a \leq x_i \leq b\}| = k$.

Game 2**:** This game corresponds to the client and servers running $\langle Index, Res_1, \ldots, Res_\ell\rangle(i_{2_1}, \vec{x}, \ldots, \vec{x})|| \cdots ||\langle Index, Res_1, \ldots, Res_\ell\rangle(i_{2_k}, \vec{x}, \ldots, \vec{x})$.

For all r, we denote by View_r^m the a view for server m generated in **Game** r.

Lemma 17. *If RQ-PIR scheme* Π *provides query indistinguishability, then for any* $r \in \{1, 2\}$ *and for any PPTA* $\mathcal{B}$*, it holds that the difference between the probability that* $\mathcal{B}$ *outputs 1 in* **Game** $r - 1$ *and* r *is negligible.*

Proof. We consider a PPTA $\mathcal{B}$ who distinguishes View_{r-1}^m from View_r^m, and denote the advantage of $\mathcal{B}$ by

$$\mathsf{Adv}_{\mathcal{B},\Pi}^r = |\Pr[\mathcal{B}(\mathsf{View}_r^m) = 1] - \Pr[\mathcal{B}(\mathsf{View}_{r-1}^m) = 1]|.$$

Then, since View_0^m and View_2^m are same as View_{index}^m in the definition of query indistinguishability, and View_1^m is same as View_{range}^m, we can see $\mathcal{B}$ as an adversary against query indistinguishability for Π. Thus, we can conclude

$$\mathsf{Adv}_{\mathcal{B},\Pi}^r = |\Pr[\mathcal{B}(\mathsf{View}_r^m) = 1] - \Pr[\mathcal{B}(\mathsf{View}_{r-1}^m) = 1]|$$

is negligible. □

By using Lemma 20, we can derive

$$\begin{aligned}
&|\Pr[\mathcal{D}(\mathsf{View}_1^m) = 1] - \Pr[\mathcal{D}(\mathsf{View}_2^m) = 1]| \\
&\leq \sum_{r=1}^{2} |\Pr[\mathcal{D}(\mathsf{View}_{r-1}^m) = 1] - \Pr[\mathcal{D}(\mathsf{View}_r^m) = 1]| \\
&\leq 2 \cdot negl.
\end{aligned}$$

□

Theorem 17. *If RQ-PIR scheme* Π *provides secure index queries for sets, then* Π *provides secure index queries.*

Proof. For all λ, for any database $\vec{x} = (x_1, \ldots, x_n)$ of size n, for any $m \in [\ell]$, for any $i, j \in [n]$, we consider a PPTA $\mathcal{D}$ against the security for index queries in RQ-PIR scheme Π. The advantage of $\mathcal{D}$ is defined by

$$\mathsf{Adv}_{\mathcal{D},\Pi}^{index} = |\Pr[\mathcal{D}(\mathsf{View}_i^m) = 1] - \Pr[\mathcal{D}(\mathsf{View}_j^m) = 1]|.$$

To obtain a proof, we use a sequence of games (**Game** 0 to **Game** 1).

Game 0**:** This game corresponds to the client and servers running $\langle Index, Res_1, \ldots, Res_\ell\rangle(i_1, \vec{x}, \ldots, \vec{x})||\langle Index, Res_1, \ldots, Res_\ell\rangle(i_2, \vec{x}, \ldots, \vec{x})|| \cdots ||\langle Index, Res_1, \ldots, Res_\ell\rangle(i_k, \vec{x}, \ldots, \vec{x})$.

Game 1**:** The difference from **Game** 1 is that client and server runs $\langle Index, Res_1, \ldots, Res_\ell\rangle(i_1', \vec{x}, \ldots, \vec{x})$ at first instead of runnning $\langle Index, Res_1, \ldots, Res_\ell\rangle(i_1, \vec{x}, \ldots, \vec{x})$.

```
$\underline{\mathcal{B}(\mathsf{View}^m)}$
Let $\mathsf{View}^m := \{\mathsf{Trans}_m(\langle Index, Res_1, \ldots, Res_\ell\rangle(j_1, \vec{x}, \ldots, \vec{x})), \vec{x}, r_1\} || \cdots$
$|| \{\mathsf{Trans}_m(\langle Index, Res_1, \ldots, Res_\ell\rangle(j_k, \vec{x}, \ldots, \vec{x})), \vec{x}, r_k\}$
$b \leftarrow \mathcal{D}(\{\mathsf{Trans}_m(\langle Index, Res_1, \ldots, Res_\ell\rangle(j_1, \vec{x}, \ldots, \vec{x})), \vec{x}, r_1\})$
output $b$
```

Figure 5.4.1: Construction of $\mathcal{B}$ in Theorem 17

For all r, we denote by View_r^m the a view for server m generated in **Game** r.

We construct $\mathcal{B}$ who distinguishes View_0^m from View_1^m who internally uses $\mathcal{D}$ as in Figure 5.4.1. We denote the advantage of $\mathcal{B}$ as

$$\mathsf{Adv}_{\mathcal{B},\Pi}^{set-index} = |\Pr[\mathcal{B}(\mathsf{View}_0^m) = 1] - \Pr[\mathcal{B}(\mathsf{View}_1^m) = 1]|.$$

By construction, $\mathcal{B}$ simulates View_i^m for $\mathcal{D}$ for a server m when $\mathcal{B}$ receives View_0^m. Moreover, $\mathcal{B}$ outputs 1 only when $\mathcal{D}$ outputs 1. Thus, the probability that $\mathcal{B}$ output 1 in the **Game** 0 is equal to $\Pr[\mathcal{D}(\mathsf{View}_i^m) = 1]$. Likewise, the probability that $\mathcal{B}$ output 1 in the **Game** 0 is equal to $\Pr[\mathcal{D}(\mathsf{View}_j^m) = 1]$. Therefore, we can obtain that

$$\begin{aligned}\mathsf{Adv}_{\mathcal{D},\Pi}^{index} &= |\Pr[\mathcal{D}(\mathsf{View}_i^m) = 1] - \Pr[\mathcal{D}(\mathsf{View}_j^m) = 1]| \\ &= \mathsf{Adv}_{\mathcal{B},\Pi}^{set-index}.\end{aligned}$$

Since we assume Π provides secure index queries for sets, we can conclude $\mathsf{Adv}_{\mathcal{D},\Pi}^{index}$ is negligible. □

From Theorem 16 and 17, we can derive following theorem.

Theorem 18. *If RQ-PIR scheme* Π *provides query indistinguishability, then* Π *provides secure index queries.*

Since we proved that query indistinguishability implies both secure index queries and secure range queries, it is enough to prove the query indistinguishability when we give the proof of security for RQ-PIR schemes. However, if the query indistinguishability is so strong notion that it is unattainable, then the above discussion becomes meaningless. In Sections 5.5 and 5.6, we give the constructions of RQ-PIR schemes that actually satisfy query indistinguishability.

5.5 Generic Construction of RQ-PIR from PIR

In this section, we give a generic construction of the RQ-PIR scheme from a PIR scheme that supports only index queries. Let $\Sigma = (Index, Res_1, \ldots, Res_\ell)$ be a PIR scheme that provides secure index queries. We give a simple generic construction of a RQ-PIR scheme Π from Σ.

Π simply uses the index query algorithm provided by the underlying Σ for index queries, and we omit the description of this. Likewise, Π uses the response algorithms from Σ, and we omit the description of these as well. However, how range queries are implemented does not follow immediately, and care must be taken to avoid these leaking information. We firstly discuss the intuition of our construction, and then provide the full details.

Note that when a client sends a range query, he does not know the indices corresponding to the elements he would like to retrieve. Thus, to retrieve these elements using Σ, he needs to obtain the relevant indices first. To do this, we use binary search. However, if binary search is used naively, it might terminate in less than $\log n$ rounds, which will leak information regarding the search, and prevent us from showing query indistinguishability, as the query size of range queries for a given number of elements is required to be constant. To prevent this, we adjust the communication rounds by sending dummy queries.

After the client has run binary searches for the bounds defining the range query, he obtains the corresponding indices. However, when the database contains elements with the same value, it is not guaranteed that the indices the client obtained cover all elements in the range. To address this, the client will query additional elements on either side of the obtained indices, until elements outside the desired range is obtained.

In the following, $[a, b]$ denotes a range query specified by the client.

$Range(\overrightarrow{ans}, st)$:

- Using binary search, the client searches for an index i such that $x_i = a$, by appropriately setting the queries $\overrightarrow{q}$, processing the corresponding $\overrightarrow{ans}$, and updating st. Each query in the search is done using *Index* from Σ. The client additionally maintains a counter c during this execution, which represents the total number of queries made. If the index i is found, but $c < \log n$, the client chooses random index $i' \in [n]$ and runs additional *Index* queries for i' until $c = \log n$.
- Then, if $x_i \geq a$, the client runs *Index* queries for $i - k$ for $k = 1, 2, ...$ until he retrieves an element such that $x_{i-k'} < a$ (i.e. $i - k' + 1$ is the smallest index of the elements with value a).
- The client then searches an index j such that $x_j = b$, using binary search as above.
- Then, if $x_j \leq b$, the client runs *Index* queries for $j + t$ for $t = 1, 2, ...$ until he retrieves an element such that $x_{j+t'} > b$ (i.e. $j - t' - 1$ is the largest index of the elements whose value is b).
- Finally, the client generates index queries for $i + 1$ to $j - 1$. Since the elements in the range from $i - k' + 1$ to i and j to $j + t' - 1$ have already been retrieved, the client obtains all elements within the range $[a, b]$.

Regardless of the type of queries executed by the client, only *Index* queries from Σ is used when communicating with the server. In addition to this, when the client submits a range query, the number of elements retrieved is always $k + 2 + 2 \log n$, where k is the number of elements within $[a, b]$. Hence, the following theorem easily follows.

Theorem 19. *The RQ-PIR scheme* Π *above provides query indistinguishability.*

5.6 Construction of RQ-PIR Scheme from FSS

In this section, we give a construction of a two-server RQ-PIR scheme using function secret sharing. After that, we give a security proof for our construction. Our construction is more efficient than the scheme described in Section 5.5 in terms of communication complexity and the number of communication rounds.

5.6.1 Construction of Two-server RQ-PIR Scheme

We construct a two-server RQ-PIR scheme $\Pi = (Index, Range, Res)$. Our construction is based on function secret sharing and we take a similar approach to the private query construction of Wang et al. [84]. While the scheme from [84] can be used for range queries as highlighted in the introduction, the scheme allows the server to distinguish whether the query from the client is an index query or range query. Our construction avoids this issue by computing a server response that the client can simultaneously used for both index and range queries, which leads to the server being unable to distinguish which query it receives. Our constructions of $Index, Range, Res_j$ $(j \in \{1,2\})$ are as follows:

$Index(\overrightarrow{ans}, st)$: This algorithm is stateless besides the initial state $st = i$ indicating the index i to retrieve, and allows a one-round index query protocol.

- On input $st = i$, the client computes $(f_1, f_2) \leftarrow \mathsf{Gen}(1^\lambda, f)$, where
$$f(x) = \begin{cases} 1 & i-1 < x < i+1 \\ 0 & \text{otherwise.} \end{cases}$$
The client then output $(\vec{q}, st') = ((f_1, f_2), \bot)$, implying that f_1 is sent to server 1 and f_2 to server 2 in the protocol.
- On input answers $\overrightarrow{ans} = ((a_{1,1}, a_{1,2}), (a_{2,1}, a_{2,2}))$ from server 1 and server 2 (and state $st = \bot$), the client computes $y \leftarrow a_{1,2} + a_{2,2}$, and outputs $(\vec{q}, st') = ((\bot, \bot), y)$ indicating termination.

$Range(\overrightarrow{ans}, st)$: During the range query protocol, the client maintains state information $st := \{st_1, st_2, st_3, st_4\}$, and initial state is $st := \{\{a, b\}, \{\}, \bot, \{\}\}$ where a and b are the bounds in the range query.

- On input $st_1 = \{a, b\}$, the client computes $(f_1, f_2) \leftarrow \mathsf{Gen}(1^\lambda, f)$, where
$$f(x) = \begin{cases} 1 & 0 < x < a \\ 0 & \text{otherwise.} \end{cases}$$
The client then outputs $(\vec{q}, st') := ((f_1, f_2), \{\{b\}, \{\}, \bot, \{\}\})$, indicating that f_1 is sent to server 1 and f_2 to server 2 in the protocol.
- On input $st_1 = \{b\}$ and $\overrightarrow{ans} = ((a_{1,1}, a_{1,2}), (a_{2,1}, a_{2,1}))$ from server 1 and server 2, the client computes $y \leftarrow a_{1,1} + a_{2,1}$ and start index $s \leftarrow y + 1$, and updates the state $st = \{\{\}, \{s\}, \bot, \{\}\}$. After that, the client computes $(g_1, g_2) \leftarrow \mathsf{Gen}(1^\lambda, g)$, where
$$g(x) = \begin{cases} 1 & 0 < x < b+1 \\ 0 & \text{otherwise.} \end{cases}$$
Finally, the client outputs $(\vec{q}, st') := ((g_1, g_2), st)$, indicating that g_1 is sent to server 1 and g_2 to server 2.

- On input $st_1 = \{\}$, $st_2 = \{s\}$, and $\overrightarrow{ans} = ((a_{1,1}, a_{1,2}), (a_{2,1}, a_{2,2}))$ from server 1 and server 2, the client computes $y' \leftarrow a_{1,1} + a_{2,1}$, sets the end index $t \leftarrow y'$, computes the number of elements in the range $k \leftarrow t - s$, and updates the state $st = \{\{\}, \{s, t\}, k\}$. After that, client computes $(h_1, h_2) \leftarrow \mathsf{Gen}(1^\lambda, h)$, where

  $$h(x) = \begin{cases} 1 & t - 1 < x < t + 1 \\ 0 & \text{otherwise.} \end{cases}$$

  The client outputs $(\vec{q}, st') := ((h_1, h_2), st)$, indicating that h_1 is sent to server 1 and h_2 to server 2.
- On input $st_1 = \{\}$, $st_2 = \{s, t\}$, $st_3 \neq 0$, and $\overrightarrow{ans} = ((a_{1,1}, a_{1,2}), (a_{2,1}, a_{2,2}))$ from server 1 and server 2, the client does the following:
  The client computes $y \leftarrow a_{1,2} + a_{2,2}$, $st_3 = st_3 - 1$, $st_4 = st_4 || y$. After that, the client computes $(h_1', h_2') \leftarrow \mathsf{Gen}(1^\lambda, h')$, where

  $$h'(x) = \begin{cases} 1 & st_3 - 1 < x < st_3 + 1 \\ 0 & \text{otherwise.} \end{cases}$$

  The client outputs $(\vec{q}, st') := ((h_1', h_2'), st)$, indicating that h_1' is sent to server 1 and h_2' to server 2.
- On input $st_1 = \{\}$, $st_2 = \{s, t\}$, $st_3 = 0$, and $\overrightarrow{ans} = ((a_{1,1}', a_{1,2}'), (a_{2,1}', a_{2,2}'))$ from server 1 and server 2, the client computes $y \leftarrow a_{1,2}' + a_{2,2}'$,$st_4 = st_4 || y$, and outputs $(\vec{q}, \vec{y}) := ((\perp, \perp), st_4)$, indicating termination.

$Res_j(\vec{x}, f_j)$**:** In the above algorithms, the client sends a share of a function f_j to server j. Upon receiving this, the server computes $a_{j,1} = \sum_{i=1}^{n} f_j(x_i)$, $a_{j,2} = \sum_{i=1}^{n} f_j(i) \cdot x_i$, and sends these to the client. Note that the server response is the same whether the query from the client is a range query or an index query. Note also that the *Res* algorithm is deterministic.

In the following, we prove security of our RQ-PIR scheme Π. Since query indistinguishability implies secure index queries and secure range queries (as shown in Section 5.4), we only prove query indistinguishability. The main idea of the security proof is to use the security of the FSS scheme to gradually change the function shares sent from the client to the servers, transforming a range query into an appropriate number of index queries.

Theorem 20. *If FSS scheme $\mathcal{FSS} = (\mathsf{Gen}, \mathsf{Eval})$ is secure, then RQ-PIR scheme Π provides query indistinguishability.*

Proof. For the function $f(x) := x + 2$,for all λ, for any database $\vec{x} = (x_1, \ldots, x_n)$ of size n, for any $m \in [2]$, for any $a, b \in [n]$, for any $i_1, \ldots i_{f(k)} \in [n]$ where $k = |\{x_i \mid a \leq x_i \leq b\}|$, we consider a PPTA $\mathcal{D}$ against query indistinguishability in RQ-PIR scheme Π. The advantage of $\mathcal{D}$ is defined by

$$\mathsf{Adv}^{ind}_{\mathcal{D},\Pi} = |\Pr[\mathcal{D}(\mathsf{View}^m_{range}) = 1] - \Pr[\mathcal{D}(\mathsf{View}^m_{index}) = 1]|.$$

Since the *Res* algorithm in our RQ-PIR scheme Π is deterministic, $\mathsf{View}^m_{range_{a,b}}$ can be written as

$$\{\mathsf{Trans}_m(\langle Range, Res_1, Res_2\rangle((a, b), \vec{x}, \vec{x}), \vec{x}\} = \{(\vec{q}, \vec{a}), \vec{x}\},$$

where $|\{x_i \mid a \leq x_i \leq b\}| = k$, $\vec{q} = (q_1, \ldots, q_{k+2})$ and the i-th element in $\vec{q}$ is a query for server m generated from some function f_i by the client, and $\vec{a} = ((a_{1,1}, a_{1,2}), \ldots, (a_{k+2,1}, a_{k+2,2}))$ and $(a_{i,1}, a_{i,2})$ is the reply from server m for query q_i.

To obtain a proof, we use a sequence of games (**Game** 0 to **Game** $k+2$).

Game 0**:** This game corresponds to the client and servers running $\langle Range, Res_1, Res_2 \rangle ((a, b), \vec{x}, \vec{x})$.

Game r $(1 \leq r \leq k+1)$**:** The difference from **Game** $r-1$ is that q_{r-1} is replaced with q'_{r-1} where q'_{r-1} is a function share for server m generated from a function $f_{r-1}(x) = \begin{cases} 1 & i_{r-1} - 1 < x < i_{r-1} + 1 \\ 0 & \text{otherwise} \end{cases}$.

Game $k+2$**:** This game corresponds to the client and servers running $\langle Index, Res_1, Res_2 \rangle (i_r, \vec{x}, \vec{x})$ for $r = 1, \ldots, f(k)$.

For all r, we denote by View_r^m the view for server m generated by the experiment **Game** r.

Lemma 18. *If FSS scheme* $\mathcal{FSS} = (\mathsf{Gen}, \mathsf{Eval})$ *is secure, then for any* $1 \leq s \leq k+2$ *and for any PPTA* $\mathcal{B}$*, it holds that the difference between the probability that* $\mathcal{B}$ *outputs 1 in* **Game** $s-1$ *and* s *is negligible.*

Proof. We consider a PPTA $\mathcal{B}$ who distinguishes View_{r-1}^m from View_r^m, and denote the advantage of $\mathcal{B}$ by

$$\mathsf{Adv}_{\mathcal{B},\Pi}^r = |\Pr[\mathcal{B}(\mathsf{View}_r^m) = 1] - \Pr[\mathcal{B}(\mathsf{View}_{r-1}^m) = 1]|.$$

Then, we construct an adversary $\mathcal{A}$ against $\mathcal{FSS}$ who uses $\mathcal{B}$ internally as shown in Figure 5.6.1.

By the construction of $\mathcal{A}$, $\mathcal{A}$ simulates View_{r-1}^m for $\mathcal{B}$ when $\mathcal{A}$ receives a function share of f^0 in the FSS security experiment. Moreover, $\mathcal{A}$ outputs 1 only when $\mathcal{B}$ outputs 1. Thus the probability that $\mathcal{A}$ outputs 1 in the experiment that $\mathcal{A}$ receives a function share of f^0 is equal to $\Pr[\mathcal{B}(\mathsf{View}_{r-1}^m) = 1]$. Likewise, the probability that $\mathcal{A}$ outputs 1 in the experiment that $\mathcal{A}$ receives a function share of f^1 is equal to $\Pr[\mathcal{B}(\mathsf{View}_r^m) = 1]$. Therefore, we obtain

$$\mathsf{Adv}_{\mathcal{B},\Pi}^r \;=\; |\Pr[\mathcal{B}(\mathsf{View}_r^m) = 1] - \Pr[\mathcal{B}(\mathsf{View}_{r-1}^m) = 1]| = \mathsf{Adv}_{\mathcal{FSS}}(1^\lambda, \mathcal{A}).$$

Since we assume $\mathcal{FSS}$ is secure i.e. that $\mathsf{Adv}_{\mathcal{FSS}}(1^\lambda, \mathcal{A})$ is negligible for all PTTA $\mathcal{A}$, we can conclude $\mathsf{Adv}_{\mathcal{B},\Pi}^r$ is negligible. □

By using Lemma 18, we can derive

$$\begin{aligned} &|\Pr[\mathcal{D}(\mathsf{View}_{range}^m) = 1] - \Pr[\mathcal{D}(\mathsf{View}_{index}^m) = 1]| \\ &\leq \sum_{r=1}^{k+2} |\Pr[\mathcal{D}(\mathsf{View}_{r-1}^m) = 1] - \Pr[\mathcal{D}(\mathsf{View}_r^m) = 1]| \\ &\leq (k+2) \cdot negl. \end{aligned}$$

□

```
$\underline{\mathcal{A}_1(1^\lambda)}$
Send $f^0 := f_{r-1}$ and $f^1 := f'_{r-1}$ to challenger,
where $f_{r-1}$ is the $r-1$-th function used in the range protocol for $[a,b]$,
and $f^1(x) = \begin{cases} 1 & i_{r-1} - 1 < x < i_{r-1} + 1 \\ 0 & \text{otherwise.} \end{cases}$
Output $st := (\{(a,b),(i_1,\dots,i_{f(k)})\},\vec{x})$
$\underline{\mathcal{A}_2(f_m^b, st)}$
Let $st := \{(a',b'),(i'_1,\dots,i'_{f(k)}),\vec{x'}\}$
Compute $s = \sum_{i=1}^n f_m^b(x'_i)$, $s' = \sum_{i=1}^n f_m^b(i) \cdot x'_i$
Run $(y_1||y_2||\dots||y_{k+2}) \leftarrow \mathsf{Trans}_m(\langle Range, Res_1, Res_2\rangle((a',b'),\vec{x'},\vec{x'}))$,
where $y_t := q_t||a_{t,1}||a_{t,2}$ for $t = 1,\dots,k+2$
For $u = 1$ to $r-2$
    $(q'_u, a'_{u,1}, a'_{u,2}) \leftarrow \mathsf{Trans}_m(\langle Index, Res_1, Res_2\rangle(i_u,\vec{x'},\vec{x'}))$
$\mathsf{View} \leftarrow \{(y'_1||y'_2||\dots||y'_{r-2}||q'_{r-2}||f_m^b||s||s'||y_r||\dots||y_{k+2}),\vec{x}\}$,
where $y'_w := q'_w||a'_{w,1}||a'_{w,2}$ for $w = 1,\dots,r-2$
$b' \leftarrow \mathcal{B}(\mathsf{View})$
Output $b'$
```

Figure 5.6.1: Construction of $\mathcal{A}$ in Lemma 18

Efficiency We summarize the efficiency of the FSS-based RQ-PIR scheme in Figure 5.6.2, and compare this to the generic constructions from Section 5.5. The RQ-PIR scheme requires a FSS scheme for interval functions, and the most efficient among these was proposed by Boyle et al [13]. Specifically, let $G : \{0,1\}^\lambda \to \{0.1\}^{2\lambda+2}$ be a PRG, and $f_{a,b} : \mathbb{G}^{\text{in}} \to \mathbb{G}^{\text{out}}$ be an interval function. Then, in their construction, the key size (i.e. the size of the query sent in one round from the client to the servers in the RQ-PIR scheme) is $8m \cdot (\lambda+1) + 2m\ell + 2\lambda$, and the size of the evaluation (i.e. the size of the response from servers to the client) is ℓ, where $m = \lceil \log_2 |\mathbb{G}^{\text{in}}| \rceil$ and $\ell = \lceil \log_2 |\mathbb{G}^{\text{out}}| \rceil$.

We note that for range queries in particular, the generic construction is less efficient, both in terms of communication cost and the number of communication rounds, compared to the FSS-based construction.

5.7 Experimental Results

We implemented the client query generation and server response computation of our RQ-PIR scheme in C++ using FSS library [104]. In our evaluation, we used a server with a 10-core Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz and 130 GB of RAM. As client, we used a 1.3 GHz Intel Core i5 machine with 8GB of RAM.

5.7.1 Evaluation

Our evaluation was done using a databases with elements consisting of 24-bits integers, and a total database size of 5, 7.5 and 10 million elements. We measured the overall time of generating client queries and server responses in our scheme for range queries retrieving

	Generic Construction	FSS based Construction
Com. Cost *Index*	cc_{pir}	cc_{fss}
Rounds *Index*	cr_{pir}	1
Com. Cost *Range*	$O(cc_{pir} \cdot (k + \log n))$	$(k + 2) \cdot cc_{fss}$
Rounds *Range*	$O(cr_{pir} \cdot (k + \log n))$	$(k + 2)$

Figure 5.6.2: The figure above shows the efficiency of the RQ-PIR schemes from Section 5.5 and 5.6 in terms of communication cost and round complexity of the underlying PIR and FSS schemes, respectively. We denote communication cost by *Com. Cost*, and the communication cost of the PIR and FSS schemes cc_{pir} and cc_{fss}, respectively. The communication costs for the client and the servers are obtained simply by using the corresponding values of cc_{pir} and cc_{fss} (note that cc_{fss} corresponds to a FSS evaluation key when considering the communication cost of the client, and a FSS evaluation result when considering the communication cost of the servers). We denote the round complexity by *Rounds*, and the round complexity of the PIR scheme by cr_{pir}. The parameter k represents the number of data entries in the database hit by the range query.

10, 50 and 100 elements from the database. In addition to the above, we noted the size of the query that the client generates and the size of the response from servers.

In our implementation, in each communication round between client and servers, the client generates a query of size 144 bytes, whereas the size of the response from each server is 8 bytes. Since our RQ-PIR scheme requires $k + 2$ rounds of communication when the client makes a range query, the total communication cost for the client is $144 \cdot (k + 2)$ bytes per server, and each server needs to generate and send responses with a total size of $8 \cdot (k+2)$ bytes, where k is the number retrieved from the database by the range query. In our particular network setup, in which the client connected to the server via a Wifi router, the total round trip time for the client to send 144 bytes of data to the server, and the server to respond with a 8 byte response, was 24 ms.

The client side computations in each round consists of generating the keys for the function secret sharing, and retrieving and combining the values from the servers' responses. Our measurements show that those computations take less than 1 ms per communication round for the client to perform. In contrast, our measurements showed that the server side computation took several orders of magnitude longer to execute, even for our smallest test case. Hence, almost the entire execution time is occupied by server side computations, which is the most important factor when considering practicality.

Figure 5.7.1 shows the experimental results of the server side computation time during the execution of the protocol. Note that our scheme was implemented to perform parallel processing of the database on the server side to accelerate the protocol execution. In particular, the server used 20 threads when running the experiments.

5.8 Conclusion

An increasing number of applications and services rely on remotely stored data. While the data itself might not be private, the information regarding a client's queries might be. Using PIR, a client can retrieve data from a database without the database server learning

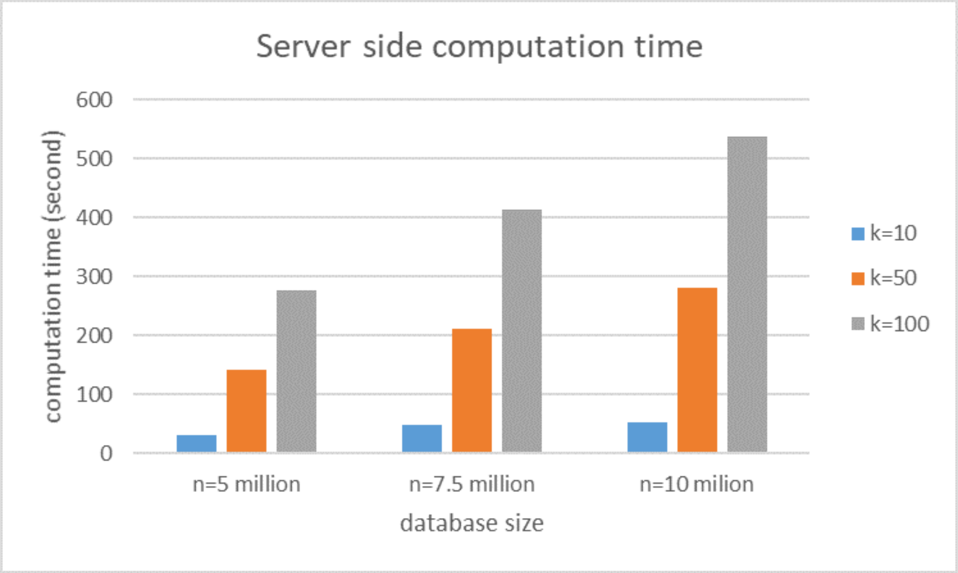

Figure 5.7.1: Experimental result of server side computation time: n denotes the database size, and k denotes the number of element that client retrieves from database by range query.

what is being retrieved.

Although many PIR schemes have been proposed in the literature, almost all of these focus on the retrieval of a single database element, and do not consider more flexible retrieval queries such as basic range queries. In addition to this, to the best of our knowledge, all PIR schemes that do support range queries, are not formally shown secure. Indeed, we pointed out that the existing PIR scheme supporting range queries leaks the information of clients' queries in some cases. In this chapter, we formalized a security model for PIR schemes that supports range queries and gave the construction of a secure multi-server scheme based on function secret sharing.

To show the usefulness of the proposed scheme in practical applications, we gave an implementation of the proposed scheme and measured its performance. This result showed that our proposed scheme is comparable to existing schemes.

Chapter 6

PIR Supporting Multi-dimensional Range Queries

6.1 Introduction

6.1.1 Background and Motivation

In Chapter 5, we gave new security notions and constructions of the PIR scheme supporting basic range queries on one-dimensional databases. However, those results are not sufficient when considering the practical use of PIR. This is because the databases used in the real world are not one-dimension but often more complex structures. While simple range queries is a step in the right direction, we need to consider more complex databases and more complex queries to support real-world applications. For example, multi-dimensional range queries are an extremely common part of many workloads [105], and the construction of PIR schemes supporting multi-dimensional range queries is an important step to bridge the gap between the PIR as an academic research topic and real-world applications. Some efforts towards this goal have been made. For example, Splinter, a private query system for public datasets by Wang et al. [84], supports a subset of typical SQL-like queries, including multi-dimensional range queries. Also, Boyle et al. [13] stated that multi-dimensional range queries are possible by using the function secret sharing scheme they proposed, although but they did not give a concrete construction and a formal security analysis.

However, a common problem with all of the above works is that the structure of the proposed schemes leaks the kind of queries a client is making, and for range queries, the number of elements returned by the server. We emphasize that the number of elements matched by a range query can leak much information regarding the query a client is making, and it is not hard to see that in some cases, this will reveal exactly what elements the client retrieves. Furthermore, none of the above mentioned works formalize security for range queries or provide a formal analysis for this type of queries, which leaves it unclear what kind of security the proposed schemes provide.

It might seem that information regarding the number of elements retrieved in a range query must inherently leak unless the trivial approach of downloading the entire database is employed. However, we proposed security notions for PIR schemes supporting range queries as well as concrete constructions, which aim at addressing this in Chapter 5. More specifically, we introduced the notion of query indistinguishability which informally requires that the PIR servers cannot distinguish between the different types of queries made by the client. Furthermore, this notion implies that, within a stream of queries, the servers cannot determine the boundaries between queries, and when combined with appropriate mixing of queries from different users, this holds the promise of hiding even the number of elements retrieved by each individual user. However, the notions and constructions in Chapter 5 are restricted to one-dimensional range queries.

6.1.2 Our Contribution

In this chapter, we extended the results in Chapter 5. Most of the existing PIR schemes consider searching simple one-dimensional databases and the supported query types are often limited to index queries only, which retrieve a single element from the databases. However, most real-world applications require more complex databases and query types.

We build upon the notion of query indistinguishability in Chapter 5, and formalize query indistinguishability for multi-dimensional range queries. More specifically, we provide four security notions for the PIR scheme supporting multi-dimensional range queries. Furthermore, we clarify the relationships among these security notions. We then give a construction of a secure multi-server scheme based on function secret sharing. This is the first instantiation of a PIR scheme supporting multi-dimensional range queries while being capable of hiding the type of query being made and, in the case of multi-dimensional range queries, the number of elements retrieved in each query, when considering a stream of queries.

6.1.3 Related Works

While most PIR schemes support only index queries, there are a few exceptions in the literature, as stated in Chapter 5 (See Subsection 5.1.3). Wang et al. [84], highlighted above, proposed PIR schemes providing functionality approaching standard SQL, including multi-dimensional range queries. It seems that their scheme supports multi-dimensional range queries. However, their scheme leaks the information of clients' queries in some cases, and they did not formally define security and provide security proofs.

6.1.4 Chapter Organization

The structure of this paper is as follows. In Section 6.2, we define the syntax and security models for multi-server PIR schemes supporting multi-dimensional range queries, and in Section 6.3, we prove relations among the introduced security notions. In Section 6.4, we show our constructions of a PIR scheme supporting multi-dimensional range queries using function secret sharing. In Section 6.5, we conclude the paper.

6.2 PIR Schemes Supporting Multi-dimensional Range Queries

Parameters. The following is a list of parameters we will use in this chapter:

- ℓ: number of servers.
- n: size of the database (number of records).

0	1	$\cdots$	$s-1$	s
ID	Age	$\cdots$	Weight	Height
1	10	$\cdots$	40	140
2	19	$\cdots$	60	170
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$n-1$	12	$\cdots$	50	150
n	25	$\cdots$	70	170

Figure 6.2.1: An example of s dimensional database of size n

- s: dimension of the database.
- V_i: size of each field in i-th dimension ($i = 1, \ldots, s$)

We denote PIR supporting multi-dimensional range queries on multi-dimensional databases by MDRQ-PIR. In this section, we define syntax and security models for MDRQ-PIR schemes. In the following, we will treat a s-dimensional database with n records as an n-entry vector $\overrightarrow{x} = (rec_1, \ldots, rec_n)$, where each rec_i consists of s values i.e. we view each entry rec_i as $(rec_{i,1}, \ldots, rec_{i,s})$, and let V_j ($j = 1, \ldots, s$) be the bit-length of the jth entry i.e. $rec_{i,j} \in \{0,1\}^{V_j}$ for all $i \in [n]$. We attach an implicit ID to each record corresponding to its row number by setting $rec_{i,0} = i$ ($i = 1, \ldots, n$). An example database is shown in Figure 6.2.1.

Note that the syntax and security models given in this section are easily applicable to PIR schemes supporting basic one-dimensional range queries (i.e. this corresponds to the case $s = 1$). However, unlike the case of Chapter 5, we do not assume the elements in the databases are sorted. Therefore, the model of PIR scheme supporting basic range queries in Chapter 5 is a special case of the model defined in Subsection 6.2.1.

6.2.1 Model

Our notion of an MDRQ-PIR scheme supports two types of queries: index queries and multi-dimensional range queries. In an index query, the client specifies an index $i \in [n]$, and obtains the i-th entry in the database i.e. rec_i. However, in a multi-dimensional range query, the client specifies a range by values $(a_i, b_i)_{i \in [s]} \in \mathbb{N}^{2s}$ ($a_i < b_i$), and obtains all entries rec_j in the database satisfying $a_i \leq rec_{j,i} \leq b_i$ for all $i = 1, \ldots, s$. Note that the client might be unaware of the indices j of the elements retrieved in a multi-dimensional range query.

Formally, we define MDRQ-PIR as interactive algorithms `Index` and `MDRange` to be executed by the client and `Res` to be executed by the server. Note that the algorithm executed by the server is `Res` only, regardless of the algorithm executed by the client.

$y/\bot \leftarrow$ `Index`$(1^\lambda, i)$**:** The `Index` algorithm is an interactive and stateful algorithm that is run by the client and takes as input the security parameter 1^λ and the index i of the entry in the database that will be retrieved. The `Index` algorithm interacts with the servers by sending queries $\vec{q} = (q_1, \ldots, q_\ell)$, where q_j is query for server j ($j = 1, \ldots, \ell$). Finally, the `Index` algorithm outputs a database entry or $\bot$.

$\vec{y}/\perp \leftarrow \texttt{MDRange}(1^\lambda, (a_i, b_i)_{i\in[s]})$**:** The MDRange algorithm is an interactive and stateful algorithm that is run by the client and takes as input the security parameter 1^λ and the range $(a_i, b_i)_{i\in[s]}$ that client wants to retrieve from database. The MDRange algorithm interacts with the servers by forwarding queries $\vec{q} = (q_1, \ldots, q_\ell)$, where q_j is the query for server j ($j = 1, \ldots, \ell$). Finally, the MDRange algorithm outputs database entries $\vec{y}$ or $\perp$.

$ans_j \leftarrow \texttt{Res}(1^\lambda, \vec{x}, j, q_j)$**:** The Res algorithm is run by each server to respond to the clients queries. The algorithm takes as input the security parameter 1^λ, database $\vec{x}$, the identifier of the server $j \in [\ell]$, and query q_j from the client, and outputs answer ans_j.

To simplify notation, we will often omit the security parameter 1^λ from the input of the above defined algorithms. In addition to this, we will use $\texttt{Res}_j(\vec{x}, q)$ to denote $\texttt{Res}(1^\lambda, \vec{x}, j, q)$.

Based on the above algorithms, we obtain protocols for index and multi-dimensional range queries by respectively combining Index and Res, and MDRange and Res. We will use the following notation regarding these:

$(y, \overrightarrow{\perp})\langle\texttt{Index}, \texttt{Res}_1, \ldots, \texttt{Res}_\ell\rangle(i, \vec{x}, \ldots, \vec{x})$**:** This denotes the client running the Index algorithm with input i, and server j replying to each query q_j from client by running $\texttt{Res}_j(\vec{x}, q_j)$, for all $j \in [\ell]$, until the Index algorithm halts. After completing the protocol, the client outputs y and each server receives no output (i.e. $\perp$).

$(\vec{y}, \overrightarrow{\perp}) \leftarrow \langle\texttt{MDRange}, \texttt{Res}_1, \ldots, \texttt{Res}_\ell\rangle((a_i, b_i)_{i\in[s]}, \vec{x}, \ldots\ , \vec{x})$: This denotes the client running the MDRange algorithm with input $(a_i, b_i)_{i\in[s]}$, and server j replying to queries q_j from client by running $\texttt{Res}_j(\vec{x}, q_j)$, for all $j \in [\ell]$ until the MDRange algorithm halts. After completing the protocol, the client outputs $\vec{y}$ and each server outputs $\perp$.

For the above protocols, we will consider the transcript of an interaction between a client and a server: for any $j \in [\ell]$ and for any $i \in [n]$, we denote by $\mathsf{Trans}_j(\langle\texttt{Index}, \texttt{Res}_1, \ldots, \texttt{Res}_\ell\rangle(i, \vec{x}, \ldots, \vec{x}))$ the messages sent between the client and server j when executing the above defined protocol for retrieving the database entry for index i (held by the client). Likewise, for any $j \in [\ell]$ and for any $(a_k, b_k)_{k\in[s]} \in \mathbb{N}^{2s}$, we denote by $\mathsf{Trans}_j(\langle\texttt{MDRange}, \texttt{Res}_1, \ldots, \texttt{Res}_\ell\rangle((a_k, b_k)_{k\in[s]}, \vec{x}, \ldots, \vec{x}))$ the messages sent between client and server j during the execution of the above defined protocol for retrieving values in the range $(a_k, b_k)_{k\in[s]}$.

6.2.2 Security Definitions

We will now define index and range query security for MDRQ-PIR schemes. In the security models, we assume that serves do not collude, and there is a secure channel between the client and each server.

At first, we define the security for MDRQ-PIR schemes called secure index queries on multi-dimensional databases. This security notion captures that the servers do not learn anything regarding the element being retrieved in the index query.

Definition 23. *An MDRQ-PIR scheme provides secure index queries on multi-dimensional databases if for all $\lambda \in \mathbb{N}$, for any dimension $s \in \mathbb{N}$, for any $V_u \in \mathbb{N}$ ($u = 1, \ldots, s$), for* *any s dimensional database $\vec{x} = (rec_1, \ldots, rec_n)$ of size n where $rec_t = (rec_{t,1}, \ldots, rec_{t,s})$ and $|rec_{t,u}| = V_u$ $(t = 1, \ldots, n)$, for any server $m \in [\ell]$, for any indices $i, j \in [n]$, for any PPTA distinguisher $\mathcal{D}$,*

$$|\Pr[\mathcal{D}(\mathsf{View}_i^m) = 1] - \Pr[\mathcal{D}(\mathsf{View}_j^m) = 1]|$$

is negligible, where

$$\mathsf{View}_k^m \leftarrow \{\mathsf{Trans}_m(\langle \texttt{Index}, \mathsf{Res}_1, \ldots, \mathsf{Res}_\ell \rangle (k, \vec{x}, \ldots, \vec{x})), \vec{x}, r\}$$

for $k \in \{i, j\}$, and where r is the randomnesses used by server m during the execution of the protocol.

We now define a simple extension of the above security notion aimed at capturing security of multi-dimensional range queries. This security notion captures that the servers do not learn the bounds $(a_i, b_i)_{i \in [s]}$ of the range query, and what elements are being retrieved via an indistinguishability requirement: any server should be unable to distinguish two different range queries as long as the number of elements in the query is the same. While the prior works considering range queries [84, 12] do not formally define security, the following security notion appears to capture the kind of security these works aim at.

Definition 24. *An MDRQ-PIR scheme provides secure multi-dimensional range queries on multi-dimensional databases if for all $\lambda \in \mathbb{N}$, for any dimension $s \in \mathbb{N}$, for any $V_u \in \mathbb{N}$ $(u = 1, \ldots, s)$, for any s dimensional database $\vec{x} = (rec_1, \ldots, rec_n)$ of size n where $rec_t = (rec_{t,1}, \ldots, rec_{t,s})$ and $|rec_{t,u}| = V_u$ $(t = 1, \ldots, n)$, for any server $m \in [\ell]$, for any bounds $a_i, b_i, c_i, d_i \in [n]$ $(2 \leq i \leq s)$ such that $|\{rec_j \mid a_i \leq rec_{j,i} \leq b_i \text{ for } i = 2, \ldots, s\}| = |\{rec_j \mid c_i \leq rec_{j,i} \leq d_i \text{ for } i = 2, \ldots, s\}| = k$, for any PPTA distinguisher $\mathcal{D}$,*

$$|\Pr[\mathcal{D}(\mathsf{View}^m_{(a_i,b_i)_{i\in[s]}}) = 1] - \Pr[\mathcal{D}(\mathsf{View}^m_{(c_i,d_i)_{i\in[s]}}) = 1]|$$

is negligible, where

$$\mathsf{View}^m_{(h_w,h'_w)_{w\in[s]}} \leftarrow \mathsf{Trans}_m(\langle \texttt{MDRange}, \mathsf{Res}_1, \ldots, \mathsf{Res}_\ell \rangle ((h_w, h'_w)_{w\in[s]}, \vec{x}, \ldots, \vec{x}))$$

for $(h_w, h'_w)_{w\in[s]} \in \{(a_i, b_i)_{i\in[s]}, (c_i, d_i)_{i\in[s]}\}$, and where r is the randomnesses of server m during the execution of the protocol.

While Definition 24 above intuitively guarantees some form of security for multi-dimensional range queries, this security notion does not guarantee that the type of query being made, or the number of elements in a multi-dimensional range query, are hidden. As discussed in the introduction, that might lead to the privacy of range queries being compromised. Thus, a stronger security notion is desirable.

Hence, we define query indistinguishability on multi-dimensional databases of an MDRQ-PIR scheme aimed at addressing this. This security notion captures that the servers cannot learn whether the client is performing a range query or a number of independent index queries for a set of arbitrary unrelated set of entries in the database. This security implies that the servers cannot distinguish individual queries within a stream of queries, and that servers can only try to estimate the overall size of the data retrieved by a client. Furthermore, this security notion allows clients to add noise to any estimate by the adversary by performing dummy queries, and if queries from multiple clients are mixed, as we discuss later, extracting reliable information regarding any individual client query becomes even harder for the adversary.

We formalize two very similar variants of query indistinguishability on multi-dimensional databases: weak query indistinguishability on multi-dimensional databases and strong query indistinguishability on multi-dimensional databases. The former captures the intuition described above, that servers are oblivious to the type of query being done and cannot detect individual queries within a stream of queries, but does not enforce that multi-dimensional range queries matching the same number of elements will always result in the same round complexity and the total amount of data transferred. In contrast, this is guaranteed by strong query indistinguishability on multi-dimensional databases, or in other words, strong query indistinguishability on multi-dimensional databases implies secure multi-dimensional range queries on multi-dimensional databases as defined above. Strong query indistinguishability on multi-dimensional databases provides an extra layer of security in case a client is not making multiple queries, is unwilling or unable to make dummy queries, and mixing queries from different clients is not possible. However, strong query indistinguishability on multi-dimensional databases inherently results in more communication between client and servers, and hence yields less efficient schemes. Therefore, in environments where a certain amount of client query activity or client query mixing can be guaranteed, weak query indistinguishability on multi-dimensional databases might be a better choice of security. We emphasize that in addition, we show how to generically convert a scheme satisfying weak query indistinguishability on multi-dimensional databases to a scheme satisfying strong query indistinguishability on multi-dimensional databases. The conversion is simple and can be applied dynamically i.e. a client can decide on a query-by-query basis whether ordinary or strong query indistinguishability on multi-dimensional databases should be achieved.

Lastly, note that while Definition 25 and 26 guarantee that database server(s) cannot detect multi-dimensional range queries (or boundaries between these) from the queries alone, the definition by itself does not address information derived from the timing of queries. In Subsection 6.2.3, we discuss ways to address this.

We formalize weak query indistinguishability on multi-dimensional databases as follows.

Definition 25. *An MDRQ-PIR scheme provides weak query indistinguishability on multi-dimensional databases if for all $\lambda \in \mathbb{N}$, for any dimension s, for any $V_u \in \mathbb{N}$ $(u = 1, \dots, s)$, for any s dimensional database $\vec{x} = (rec_1, \dots, rec_n)$ of size n where $rec_t = (rec_{t,1}, \dots, rec_{t,s})$ and $|rec_{t,u}| = V_u$ $(t = 1, \dots, n)$, for any server $m \in [\ell]$, for any bounds $(a_i, b_i)_{i\in[s]}$, there exists a constant $C \in \mathbb{N}$ such that for any set of indices $i_1, \dots i_C \in [n]$, for any PPTA distinguisher $\mathcal{D}$,*

$$|\Pr[\mathcal{D}(\mathsf{View}^m_{mdrange}) = 1] - \Pr[\mathcal{D}(\mathsf{View}^m_{index}) = 1]|$$

is negligible, where

$$\mathsf{View}^m_{mdrange} \leftarrow \{\mathsf{Trans}_m(\langle \mathit{MDRange}, \mathit{Res}_1, \dots, \mathit{Res}_\ell\rangle((a_i, b_i)_{i\in[s]}, \vec{x}, \dots, \vec{x})), \vec{x}, r\},$$

and

$$\mathsf{View}^m_{index} \leftarrow \mathsf{View}_{i_1}||\cdots||\mathsf{View}_{i_C},$$

where

$$\mathsf{View}_w \leftarrow \{\mathsf{Trans}_m(\langle \mathtt{Index}, \mathtt{Res}_1, \ldots, \mathtt{Res}_\ell\rangle(w, \vec{x}, \ldots, \vec{x})), \vec{x}, r\},$$

and $w \in \{i_1, \ldots, i_C\}$ and r is the randomnesses of server m during the execution of the protocol.

Then, we give the definition of strong query indistinguishability on multi-dimensional databases.

Definition 26. *An MDRQ-PIR scheme provides strong query indistinguishability on multi-dimensional databases if there exists a polynomial time computable function $f : \mathbb{N} \to \mathbb{N}$ such that query indistinguishability holds for $C = f(k)$, where k is the number of element in the database that satisfy the condition specified by $(a_i, b_i)_{i\in[s]}$, i.e.,*

$$|\Pr[\mathcal{D}(\mathsf{View}^m_{mdrange}) = 1] - \Pr[\mathcal{D}(\mathsf{View}^m_{index}) = 1]|$$

is negligible, where

$$\mathsf{View}^m_{mdrange} \leftarrow \{\mathsf{Trans}_m(\langle \mathit{MDRange}, \mathit{Res}_1, \ldots, \mathit{Res}_\ell\rangle((a_i, b_i)_{i\in[s]}, \vec{x}, \ldots, \vec{x})), \vec{x}, r\},$$

and

$$\mathsf{View}^m_{index} \leftarrow \mathsf{View}_{i_1} || \cdots || \mathsf{View}_{i_{f(k)}},$$

where

$$\mathsf{View}_w \leftarrow \{\mathsf{Trans}_m(\langle \mathtt{Index}, \mathtt{Res}_1, \ldots, \mathtt{Res}_\ell\rangle(w, \vec{x}, \ldots, \vec{x})), \vec{x}, r\},$$

and $w \in \{i_1, \ldots, i_{f(k)}\}$ and r is the randomnesses of server m during the execution of the protocol.

6.2.3 Discussion

In this subsection, we discuss our definition of weak query indistinguishability on multi-dimensional databases. In Chapter 5, we proposed three security notions: indistinguishability of index queries, indistinguishability of range queries retrieving the same number of elements, and query indistinguishability for 1-dimensional databases setting. Furthermore, we showed that the notion of query indistinguishability implies indistinguishability for range queries. However, this is not the case for our notion of weak query indistinguishability on multi-dimensional databases. Specifically, our notion of weak query indistinguishability on multi-dimensional databases does not imply that multi-dimensional range queries retrieving the same number of elements from the database, will use the same number of communication rounds to do so. This will theoretically allow a server to distinguish between two range queries retrieving the same number of elements, if these queries are considered in isolation. However, the implicit security goal of query indistinguishability in Chapter 5 is not to achieve security for range queries in isolation as defined, as this provides no guarantee with respect to range queries retrieving different numbers of elements. Rather, the goal is to obtain the property that individual queries cannot be distinguished within a stream of queries. For this, our definition suffices. Furthermore, our definition allows more efficient constructions.

It should be noted that weak query indistinguishability on multi-dimensional databases only provides meaningful security for a stream of queries. The simplest one is for the client to adopt a constant query rate, in which all index queries and each step of range queries, are executed at a constant rate. Additionally, adding dummy queries to maintain the query rate in between real queries will eliminate information leakage due to the timing of queries. The other way is to use private information retrieval (UPIR) [103], as we mentioned in Chapter 5. In UPIR, multiple clients form a P2P network with a shared memory, and the clients forward each other's queries to the database, thereby preventing the database from learning the identity of the user who sent a particular query. Swanson et al. [106] extend the results of [103] and analyze the privacy properties of the scheme with respect to not only the database but also coalitions of honest-but-curious users by providing a formal analysis of the probabilistic advantage coalitions of users have in guessing the source of queries.

We emphasize that in the above scenarios, our formalization of weak query indistinguishability on multi-dimensional databases will suffice, and introducing additional requirements regarding the communication round complexity of individual queries will not enhance security.

6.3 Relation Among Security Notions

In this section, we prove implications among the security notions introduced above. Specifically, we prove that weak query indistinguishability on multi-dimensional databases implies secure index queries on multi-dimensional databases, and that strong query indistinguishability on multi-dimensional databases implies secure multi-dimensional range queries on multi-dimensional databases. Lastly, we show how a scheme can be strengthened from weak to strong query indistinguishability on multi-dimensional databases via a generic conversion.

At first, we prove that strong query indistinguishability on multi-dimensional databases implies secure multi-dimensional range queries for multi-dimensional databases.

Theorem 21. *If MDRQ-PIR scheme* Π *provides strong query indistinguishability on multi-dimensional databases, then* Π *provides secure multi-dimensional range queries on multi-dimensional databases.*

Proof. For all $\lambda \in \mathbb{N}$, for any dimension $s \in \mathbb{N}$, for any $V_u \in \mathbb{N}$ $(u = 1, \ldots, s)$, for any s dimensional database $\vec{x} = (rec_1, \ldots, rec_n)$ of size n where $rec_t = (rec_{t,1}, \ldots, rec_{t,s})$ and $|rec_{t,u}| = V_u$ $(t = 1, \ldots, n)$, for any server $m \in [\ell]$, for any bounds $a_i, b_i, c_i, d_i \in [n]$ $(i \in [s])$ such that $|\{rec_j \mid a_i \leq rec_{j,i} \leq b_i \text{ for } i = 1, \ldots, s\}| = |\{rec_j \mid c_i \leq rec_{j,i} \leq d_i \text{ for } i = 1, \ldots, s\}| = k$, we consider a PPTA $\mathcal{D}$ against secure multi-dimensional range queries on multi-dimensional databases in MDRQ-PIR scheme Π. The advantage of $\mathcal{D}$ is defined by

$$\mathsf{Adv}^{range}_{\mathcal{D},\Pi} = |\Pr[\mathcal{D}(\mathsf{View}^m_{(a_i,b_i)_{i\in[s]}}) = 1] - \Pr[\mathcal{D}(\mathsf{View}^m_{(c_i,d_i)_{i\in[s]}}) = 1]|.$$

To obtain a proof, we use a sequence of games (**Game** 0 to **Game** 2).

Game 0**:** This game corresponds to the client and servers running $\langle \texttt{Range}, \texttt{Res}_1, \ldots, \texttt{Res}_\ell \rangle$ $((a_i, b_i)_{i\in[s]}, \vec{x}, \ldots, \vec{x})$.

Game 1**:** This game corresponds to the client and servers running $\langle \texttt{Index}, \texttt{Res}_1, \ldots, \texttt{Res}_\ell \rangle$ $(i_1, \vec{x}, \ldots, \vec{x}), \ldots,$
$\langle \texttt{Index}, \texttt{Res}_1, \ldots, \texttt{Res}_\ell \rangle(i_{f(k)}, \vec{x}, \ldots, \vec{x})$, where $i_1, \ldots, i_{f(k)}$ are chosen randomly from $[n]$.

Game 2**:** This game corresponds to the client and servers running $\langle \texttt{Range}, \texttt{Res}_1, \ldots, \texttt{Res}_\ell \rangle((c_i, d_i)_{i \in [s]}, \vec{x}, \ldots, \vec{x})$.

For all r, we denote by View_r^m the view for server m generated in **Game** r.

Lemma 19. *If RQ-PIR scheme* Π *provides strong query indistinguishability on multi-dimensional databases, then for any* $r \in \{1, 2\}$ *and for any PPTA* $\mathcal{B}$*, it holds that the difference between the probability that* $\mathcal{B}$ *outputs 1 in* **Game** $r-1$ *and* r *is negligible.*

Proof. We consider a PPTA $\mathcal{B}$ who distinguishes View_{r-1}^m from View_r^m, and denote the advantage of $\mathcal{B}$ by

$$\mathsf{Adv}_{\mathcal{B},\Pi}^r = |\Pr[\mathcal{B}(\mathsf{View}_r^m) = 1] - \Pr[\mathcal{B}(\mathsf{View}_{r-1}^m) = 1]|.$$

Then, since View_0^m and View_2^m are same as View_{range}^m in the definition of strong query indistinguishability on multi-dimensional databases, and View_1^m is same as View_{index}^m, we can see $\mathcal{B}$ as an adversary against strong query indistinguishability on multi-dimensional databases for Π. Thus, we can conclude

$$\mathsf{Adv}_{\mathcal{B},\Pi}^r = |\Pr[\mathcal{B}(\mathsf{View}_r^m) = 1] - \Pr[\mathcal{B}(\mathsf{View}_{r-1}^m) = 1]|$$

is negligible. □

By using Lemma 19, we can derive

$$\begin{aligned} &|\Pr[\mathcal{D}(\mathsf{View}_{(a_i,b_i)_{i\in[s]}}^m) = 1] \\ &\quad - \Pr[\mathcal{D}(\mathsf{View}_{(c_i,d_i)_{i\in[s]}}^m) = 1]| \\ &\leq \sum_{r=1}^{2} |\Pr[\mathcal{D}(\mathsf{View}_{r-1}^m) = 1] - \Pr[\mathcal{D}(\mathsf{View}_r^m) = 1]| \\ &\leq 2 \cdot negl. \end{aligned}$$

□

We now prove that weak query indistinguishability on multi-dimensional databases implies secure index queries on multi-dimensional databases. To show this implication, we introduce an intermediate security notion in which we denote index queries for sets on multi-dimensional databases. At first, we prove that weak query indistinguishability on multi-dimensional databases implies secure index queries for sets on multi-dimensional databases. Then we show that index queries for sets on multi-dimensional databases implies secure index queries on multi-dimensional databases. Note that this definition is only introduced to prove Theorem 22, and is not meant to be useful by itself.

Definition 27. *An MDRQ-PIR scheme provides secure index queries for sets on multi-dimensional databases if for all $\lambda \in \mathbb{N}$, for any dimension s, for any $V_u \in \mathbb{N}$ ($u = 1, \dots, s$), for any s dimensional database $\vec{x} = (rec_1, \dots, rec_n)$ of size n where $rec_t = (rec_{t,1}, \dots, rec_{t,s})$, and $|rec_{t,u}| = V_u$ ($t = 1, \dots, n$), there exists $C \in \mathbb{N}$ such that for any server $m \in [\ell]$, any set of indices $\vec{i_1} = (i_{1,1}, \dots, i_{1,C}), \vec{i_2} = (i_{2,1}, \dots, i_{2,C}) \in [n]^C$, for any PPTA distinguisher $\mathcal{D}$,*

$$|\Pr[\mathcal{D}(\mathsf{View}^m_{\vec{i_1}}) = 1] - \Pr[\mathcal{D}(\mathsf{View}^m_{\vec{i_2}}) = 1]|$$

is negligible, where

$$\begin{aligned}\mathsf{View}^m_k \leftarrow \mathsf{Trans}_m(\langle \mathtt{Index}, \mathtt{Res}_1, \dots, \mathtt{Res}_\ell \rangle (i_{t,1}, \vec{x}, \dots, \vec{x}))|| \\ \cdots ||\mathsf{Trans}_m(\langle \mathtt{Index}, \mathtt{Res}_1, \dots, \mathtt{Res}_\ell \rangle (i_{t,C}, \vec{x}, \dots, \vec{x})),\end{aligned}$$

for $t \in \{1, 2\}$, and where $r_1, \dots, r_k$ is the randomnesses used by server m during the execution of the protocol.

Theorem 22. *If MDRQ-PIR scheme Π provides weak query indistinguishability on multi-dimensional databases, then Π provides secure index queries for sets on multi-dimensional databases.*

Proof. Let $(a_i, b_i)_{i \in [s]}$ be any multi-dimensional range query. Since Π provides weak query indistinguishability on multi-dimensional databases, we know that there exists a $C \in \mathbb{N}$ such that this multi-dimensional range query is indistinguishable from any set of C index queries. We use this property to obtain a proof. Specifically, we use the following sequence of games (**Game** 0 to **Game** 2).

Game 0**:** This game corresponds to the client and servers running $\langle \mathtt{Index}, \mathtt{Res}_1, \dots, \mathtt{Res}_\ell \rangle$ $(i_{1,1}, \vec{x}, \dots, \vec{x})|| \cdots ||\langle \mathtt{Index}, \mathtt{Res}_1, \dots, \mathtt{Res}_\ell \rangle (i_{1,C}, \vec{x}, \dots, \vec{x})$.

Game 1**:** This game corresponds to the client and servers running $\langle \mathtt{MDRange}, \mathtt{Res}_1, \dots, \mathtt{Res}_\ell \rangle$ $((a_i, b_i)_{i \in [s]}, \vec{x}, \dots, \vec{x})$.

Game 2**:** This game corresponds to the client and servers running $\langle \mathtt{Index}, \mathtt{Res}_1, \dots, \mathtt{Res}_\ell \rangle$ $(i_{2,1}, \vec{x}, \dots, \vec{x})|| \cdots ||\langle \mathtt{Index}, \mathtt{Res}_1, \dots, \mathtt{Res}_\ell \rangle (i_{2,C}, \vec{x}, \dots, \vec{x})$.

For all r, we denote by View^m_r the a view for server m generated in **Game** r. Note that the advantage of any distinguisher $\mathcal{D}$ for sets $\vec{i_1} = (i_{1,1}, \dots, i_{1,C})$ and $\vec{i_2} = (i_{2,1}, \dots, i_{2,C})$ is given by

$$\mathsf{Adv}_{\mathcal{D},\Pi} = |\Pr[\mathcal{D}(\mathsf{View}^m_0) = 1] - \Pr[\mathcal{D}(\mathsf{View}^m_2) = 1]|.$$

Lemma 20. *If MDRQ-PIR scheme Π provides weak query indistinguishability on multi-dimensional databases, then for any $r \in \{1, 2\}$ and for any PPTA $\mathcal{B}$, it holds that the difference between the probability that $\mathcal{B}$ outputs 1 in* **Game** $r - 1$ *and r is negligible.*

Proof. We consider a PPTA $\mathcal{B}$ who distinguishes View^m_{r-1} from View^m_r, and denote the advantage of $\mathcal{B}$ by

$$\mathsf{Adv}^r_{\mathcal{B},\Pi} = |\Pr[\mathcal{B}(\mathsf{View}^m_r) = 1] - \Pr[\mathcal{B}(\mathsf{View}^m_{r-1}) = 1]|.$$

Then, since View_0^m and View_2^m are same as View_{index}^m in the definition of weak query indistinguishability on multi-dimensional databases, and View_1^m is same as View_{range}^m, we can see $\mathcal{B}$ as an adversary against weak query indistinguishability on multi-dimensional databases for Π. Thus, we can conclude

$$\mathsf{Adv}_{\mathcal{B},\Pi}^{r} = |\Pr[\mathcal{B}(\mathsf{View}_r^m) = 1] - \Pr[\mathcal{B}(\mathsf{View}_{r-1}^m) = 1]|$$

is negligible. □

By using Lemma 20, we can derive

$$\begin{aligned} &= |\Pr[\mathcal{D}(\mathsf{View}_0^m) = 1] - \Pr[\mathcal{D}(\mathsf{View}_2^m) = 1]| \\ &\leq \sum_{r=1}^{2} |\Pr[\mathcal{D}(\mathsf{View}_{r-1}^m) = 1] - \Pr[\mathcal{D}(\mathsf{View}_r^m) = 1]| \\ &\leq 2 \cdot negl. \end{aligned}$$

□

Theorem 23. *If MDRQ-PIR scheme* Π *provides secure index queries for sets on multi-dimensional databases, then* Π *provides secure index queries on multi-dimensional databases.*

Proof. Let $\lambda \in \mathbb{N}$ be security parameter and $\vec{x} = (rec_1, \ldots, rec_n)$ be s-dimensional database of size n where $rec_t = (rec_{t,1}, \ldots, rec_{t,s})$ and $|rec_{t,u}| = V_u$ $(t = 1, \ldots, n)$. Then for any server $m \in [\ell]$, for any $i, j \in [n]$, we consider a PPTA $\mathcal{D}$ against the secure index queries on multi-dimensional databases in MDRQ-PIR scheme Π. The advantage of $\mathcal{D}$ is defined by

$$\mathsf{Adv}_{\mathcal{D},\Pi}^{index} = |\Pr[\mathcal{D}(\mathsf{View}_i^m) = 1] - \Pr[\mathcal{D}(\mathsf{View}_j^m) = 1]|.$$

To obtain a proof, we consider the adversary $\mathcal{B}$ against secure index queries for sets on multi-dimensional databases who distinguishes whether the client and servers execute the index queries for $\vec{i_1} = (i, i_2, \ldots, i_C)$ or $\vec{i_2} = (j, i_2, \ldots, i_C)$ where C is the number of queries corresponds to the definition of secure index queries for sets on multi-dimensional databases.

We construct $\mathcal{B}$ who internally uses $\mathcal{D}$ as in Figure 6.3.1. We denote the advantage of $\mathcal{B}$ as

$$\mathsf{Adv}_{\mathcal{B},\Pi}^{set-index} = |\Pr[\mathcal{B}(\mathsf{View}_0^m) = 1] - \Pr[\mathcal{B}(\mathsf{View}_1^m) = 1]|.$$

By construction, $\mathcal{B}$ simulates View_i^m for $\mathcal{D}$ for a server m when $\mathcal{B}$ receives View_0^m. Moreover, $\mathcal{B}$ outputs 1 only when $\mathcal{D}$ outputs 1. Therefore, we can obtain that

$$\begin{aligned} &\mathsf{Adv}_{\mathcal{D},\Pi}^{index} = |\Pr[\mathcal{D}(\mathsf{View}_i^m) = 1] - \Pr[\mathcal{D}(\mathsf{View}_j^m) = 1]| \\ &= \mathsf{Adv}_{\mathcal{B},\Pi}^{set-index}. \end{aligned}$$

Since we assume Π provides secure index queries for sets on multi-dimensional databases, we can conclude $\mathsf{Adv}_{\mathcal{D},\Pi}^{index}$ is negligible. □

```
$\underline{\mathcal{B}(\mathsf{View}^m)}$
Let $\mathsf{View}^m := \{\mathsf{Trans}_m(\langle Index, Res_1, \ldots, Res_\ell \rangle(j_1, \vec{x}, \ldots, \vec{x})), \vec{x}, r_1\} || \cdots$
$|| \{\mathsf{Trans}_m(\langle Index, Res_1, \ldots, Res_\ell \rangle(j_k, \vec{x}, \ldots, \vec{x})), \vec{x}, r_k\}$
$b \leftarrow \mathcal{D}(\{\mathsf{Trans}_m(\langle Index, Res_1, \ldots, Res_\ell \rangle(j_1, \vec{x}, \ldots, \vec{x})), \vec{x}, r_1\})$
output $b$
```

Figure 6.3.1: Construction of $\mathcal{B}$ in Theorem 23

From Theorem 22 and 23, we can derive following theorem.

Theorem 24. *If MDRQ-PIR scheme* Π *provides weak query indistinguishability on multi-dimensional databases, then* Π *provides secure index queries on multi-dimensional databases.*

Also, following theorem trivially holds.

Theorem 25. *If MDRQ-PIR scheme* Π *provides strong query indistinguishability on multi-dimensional databases, then* Π *provides weak query indistinguishability on multi-dimensional databases.*

Therefore, like the case of query indistinguishability in Chapter 5, strong query indistinguishability on multi-dimensional databases implies both secure index queries on multi-dimensional databases and secure multi-dimensional range queries on multi-dimensional databases.

6.3.1 Generic Conversion from Weak to Strong Query Indistinguishability

In this subsection, we show our generic conversion of a PIR scheme satisfying strong query indistinguishability on multi-dimensional databases from a PIR scheme satisfying weak query indistinguishability on multi-dimensional databases.

Let $\Sigma = (\mathtt{Index}, \mathtt{MDRange}, \mathtt{Res}_1, \ldots, \mathtt{Res}_\ell)$ be an MDRQ-PIR scheme that provides weak query indistinguishability on multi-dimensional databases. We give a simple generic conversion of an MDRQ-PIR scheme Π from Σ.

Π simply uses the index query algorithm provided by the underlying Σ for index queries, and we omit the description of this. Likewise, Π uses the response algorithms from Σ, and we omit the description of these as well. Hence, it remains to show the construction of $\mathtt{MDRange}'$ in Π. We firstly discuss the intuition of our construction, and then provide the full details.

In Σ, when the client executes multi-dimensional range queries, two queries with the same number of data hits are not guaranteed to have the same communication complexity and could potentially be distinguished because of this. A simple solution is to add dummy traffic that ensures all queries hitting the same number of elements will have the same communication complexity. Note, however, that this traffic must appear to be part of the range queries to any potential distinguisher. We use index queries for this purpose, and since we assume Σ satisfies weak query indistinguishability on multi-dimensional databases, no distinguisher will be able to determine when the original multi-dimensional range query ends and the dummy index queries begins.

Protocol 1 $\langle \texttt{MDRange}', \texttt{Res}_1, \ldots, \texttt{Res}_\ell \rangle((a_i, b_i)_{i \in [s]}, \vec{x}, \ldots, \vec{x})$

```
$(\vec{y}, \perp, \ldots, \perp) \leftarrow \langle \texttt{MDRange}, \texttt{Res}_1, \ldots, \texttt{Res}_\ell \rangle(a, \vec{x}, \ldots, \vec{x})$
for $j = C + 1$ to $g(k)$ do
    $dum \leftarrow [n]$
    $(z, \overrightarrow{\perp}) \leftarrow \langle \texttt{Index}, \texttt{Res}_1, \ldots, \texttt{Res}_\ell \rangle(dum, \vec{x}, \ldots, \vec{x})$
end for
return $\vec{y}$
```

In the following, $(a_i, b_i)_{i \in [s]}$ denotes a multi-dimensional range query specified by the client. Let k be the number of records hit by the multi-dimensional range query $(a_i, b_i)_{i \in [s]}$, C be the equivalent number of index queries guaranteed by query indistinguishability of Σ, and let $g(k)$ be an upper bound on the number of index queries equivalent to any multi-dimensional range queries that hit k records in Σ. The construction of $\texttt{MDRange}'$ is shown in Protocol 1, in protocol form (i.e. via the interaction with $\texttt{Res}$).

Then, we prove the security of Π.

Theorem 26. *If Σ provides weak query indistinguishability on multi-dimensional databases, then Π above provides strong query indistinguishability on multi-dimensional databases.*

Proof. For the function $g(x)$, for all λ, for any dimension s, for any $V_u \in \mathbb{N}$ ($u = 1, \ldots, s$), for any s dimensional database $\vec{x} = (rec_1, \ldots, rec_n)$ of size n where $rec_t = (rec_{t,1}, \ldots, rec_{t,s})$ and $|rec_{t,u}| = V_u$ ($t = 1, \ldots, n$), for any server $m \in [\ell]$, for any $(a_i, b_i)_{i \in [s]}$, for any $i_1, \ldots i_{g(k)} \in [n]$ where k is the number of element in the database that satisfy the condition specified by $(a_i, b_i)_{i \in [s]}$, we consider a PPTA $\mathcal{D}$ against strong query indistinguishability on multi-dimensional databases in MDRQ-PIR scheme Π. The advantage of $\mathcal{D}$ is defined by

$$\mathsf{Adv}^{ind}_{\mathcal{D},\Pi} = |\Pr[\mathcal{D}(\mathsf{View}^m_{mdrange}) = 1] - \Pr[\mathcal{D}(\mathsf{View}^m_{index}) = 1]|.$$

From our construction of Π, $\mathsf{View}^m_{mdrange}$ can be written as

$$\begin{aligned}
&\{\mathsf{Trans}_m(\langle \texttt{MDRange}', \texttt{Res}_1, \ldots, \texttt{Res}_\ell \rangle((a_i, b_i)_{i \in [s]}, \vec{x}, \ldots, \vec{x}), \vec{x}\} \\
&= \{\mathsf{Trans}_m(\langle \texttt{MDRange}, \texttt{Res}_1, \ldots, \texttt{Res}_\ell \rangle((a_i, b_i)_{i \in [s]}, \vec{x}, \ldots, \vec{x}), \vec{x}\} \\
&|| \{\mathsf{Trans}_m(\langle \texttt{Index}, \texttt{Res}_1, \ldots, \texttt{Res}_\ell \rangle(j_{C+1}, \ldots \vec{x}, \vec{x}), \vec{x}\} || \\
&\cdots || \{\mathsf{Trans}_m(\langle \texttt{Index}, \texttt{Res}_1, \ldots, \texttt{Res}_\ell \rangle(j_{g(k)}, \vec{x}, \ldots, \vec{x}), \vec{x}\},
\end{aligned}$$

where $j_{C+1}, \ldots, j_{g(k)}$ corresponds to the randomly chosen indices for dummy queries.

To obtain a proof, we use a sequence of games (**Game** 0 to **Game** 2).

Game 0**:** This game corresponds to the client and servers running $\langle \texttt{MDRange}', \texttt{Res}_1, \ldots, \texttt{Res}_\ell \rangle((a_i, b_i)_{i \in [s]}, \vec{x}, \ldots, \vec{x})$.

Game 1**:** This game corresponds to the client and servers running $\langle \texttt{Index}, \texttt{Res}_1, \ldots, \texttt{Res}_\ell \rangle(i_1, \ldots \vec{x}, \vec{x}), \ldots, \langle \texttt{Index}, \texttt{Res}_1, \ldots, \texttt{Res}_\ell \rangle(i_C, \vec{x}, \ldots, \vec{x}), \langle \texttt{Index}, \texttt{Res}_1, \ldots, \texttt{Res}_\ell \rangle(j_{C+1}, \ldots \vec{x}, \vec{x}), \ldots, \langle \texttt{Index}, \texttt{Res}_1, \ldots, \texttt{Res}_\ell \rangle(j_{g(k)}, \vec{x}, \ldots, \vec{x})$.

Game 2**:** This game corresponds to the client and servers running $\langle \texttt{Index}, \texttt{Res}_1, \ldots, \texttt{Res}_\ell \rangle(i_1, \vec{x}, \ldots, \vec{x}), \ldots, \langle \texttt{Index}, \texttt{Res}_1, \ldots, \texttt{Res}_\ell \rangle(i_{g(k)}, \vec{x}, \ldots, \vec{x})$.

For all r, we denote by View_r^m the view for server m generated by the experiment **Game** r.

Lemma 21. *If Σ provides weak query indistinguishability on multi-dimensional databases, then for any PPTA $\mathcal{B}$, it holds that the difference between the probability that $\mathcal{B}$ outputs 1 in* **Game** 0 *and* 1 *is negligible.*

Proof. We consider a PPTA $\mathcal{B}$ who distinguishes View_0^m from View_1^m, and denote the advantage of $\mathcal{B}$ by

$$\mathsf{Adv}_{\mathcal{B},\Pi} = |\Pr[\mathcal{B}(\mathsf{View}_1^m) = 1] - \Pr[\mathcal{B}(\mathsf{View}_0^m) = 1]|.$$

Then, since the transcript $\{\mathsf{Trans}_m(\langle \mathtt{Index}, \mathtt{Res}_1, \ldots, \mathtt{Res}_\ell \rangle (j_{C+1}, \ldots \vec{x}, \vec{x}), \vec{x}\} || \cdots || \{\mathsf{Trans}_m(\langle \mathtt{Index}, \mathtt{Res}_1, \ldots, \mathtt{Res}_\ell \rangle (j_{g(k)}, \vec{x}, \ldots, \vec{x}), \vec{x}\}$ is the same in Game 0 and 1, View_0^m and View_1^m is the same as $\mathsf{View}_{mdrange}^m$ and View_{index}^m, respectively, in the definition of weak query indistinguishability on multi-dimensional databases. Thus, we conclude

$$\mathsf{Adv}_{\mathcal{B},\Pi} = |\Pr[\mathcal{B}(\mathsf{View}_1^m) = 1] - \Pr[\mathcal{B}(\mathsf{View}_0^m) = 1]|$$

is negligible. □

Lemma 22. *If Σ provides weak query indistinguishability on multi-dimensional databases, then for any PPTA $\mathcal{B}$, it holds that the difference between the probability that $\mathcal{B}$ outputs 1 in* **Game** 1 *and* 2 *is negligible.*

Proof. This lemma immediately follows from Theorem 22.

□

By using Lemma 21 and 22, we can derive

$$\begin{aligned}&|\Pr[\mathcal{D}(\mathsf{View}_{mdrange}^m) = 1] - \Pr[\mathcal{D}(\mathsf{View}_{index}^m) = 1]| \\ &\leq 2 \cdot negl.\end{aligned}$$

□

Since we have a generic conversion from the MDRQ-PIR scheme satisfying strong query indistinguishability on multi-dimensional databases from the scheme satisfying weak query indistinguishability on multi-dimensional databases, when considering the security of the MDRQ-PIR schemes in this paper, we will prove only weak query indistinguishability on multi-dimensional databases.

6.4 Construction of an MDRQ-PIR Scheme

In this section, we give the construction of an MDRQ-PIR scheme. Unlike basic range queries on a one-dimensional database, multi-dimensional databases cannot necessarily be sorted with respect to all columns. As a consequence, the basic technique used in the construction of RQ-PIR schemes cannot be used, and the construction of an MDRQ-PIR scheme requires a new approach. Especially, it seems difficult to extend the result of the generic construction of the RQ-PIR scheme from the PIR scheme supporting only index queries in Chapter 5 to multi-dimensional databases setting.

6.4.1 Construction of a Two-server MDRQ-PIR Scheme

We construct a two-server MDRQ-PIR scheme $\Pi = (\texttt{Index}, \texttt{MDRange}, \texttt{Res})$. Note that while not all field values can be assumed to be sorted in MDRQ-PIR, as in Chapter 5, the implicitly assigned IDs for each row will be sorted by definition. Our construction will make use of these IDs.

The basic tool we use to implement our scheme is function secret sharing for decision trees, which allows multiple conditions to be checked simultaneously. Specifically, in our construction we will use `FSSQuery` shown in Protocol 2 as a sub-protocol.

The input to `FSSQuery` protocol from the client is $(i_1, i_2, (a_i, b_i)_{i\in[s]})$ where $i_1, i_2 \in [n]$ are boundaries for the implicitly defined row IDs and $(a_i, b_i)_{i\in[s]} \in \mathbb{N}^{2s}$ is the condition on the row values. This structure allows the client to restrict which part of the database the `FSSQuery` is applied to.

The server response $\texttt{Res}_j$ $(j \in \{1, 2\})$ is defined as follows (note that in the multi-dimensional setting, each record $rec_i = (rec_{i,0}, \dots, rec_{i,s})$ is a vector of size $s+1$ corresponding to the input of the function secret shares defined above):

$\texttt{Res}_j(\vec{x}, f_j)$: Upon receiving a share of a function f_j from the client, the server j computes $a_{j,1} = \sum_{i=1}^{n} f_j(rec_i)$, $a_{j,2} = \sum_{i=1}^{n} f_j(rec_i) \cdot rec_i$, and sends these to the client.

The main difference between our FSS-based RQ-PIR scheme in Chapter 5 and our MDRQ-PIR scheme lies in the construction of `MDRange`. In the following, we give the intuition for our construction of the `MDRange` algorithm: At first, the client identifies how many elements in the database that match the given range by using `FSSQuery`. Then, the client and servers split the database into two parts and process the parts recursively. For each part, the following cases are considered:

- There are no matching elements.
- There is exactly one matching element.
- There is more than one matching element, but not all elements match.
- All elements are matching elements.

In the first case, the client finds the matching elements that he wants to retrieve are all stored in the other part, so he continues the search in the other part. In the second case, the client can retrieve the element by executing `FSSQuery` with the servers. In the third case, the client continues the search by further splitting the database into smaller parts. In the fourth case, all the elements stored in the partitioned database can be retrieved one by one using `FSSQuery`.

If the client wants to retrieve the element in range $(a_i, b_i)_{i\in[s]}$ from the database, he can retrieve all elements that satisfy the condition by executing $\langle \texttt{MDRange}, \texttt{Res}_1, \texttt{Res}_2 \rangle ((1, n, \perp, (a_i, b_i)_{i\in[s]}), \vec{x}, \vec{x})$.

Protocol 2 $\langle \texttt{FSSQuery}, \texttt{Res}_1, \texttt{Res}_2 \rangle (i_1, i_2, (a_i, b_i)_{i \in [s]}, \vec{x}, \vec{x})$

```
The client computes $(f_1, f_2) \leftarrow \mathsf{Gen}(1^\lambda, f)$, where
    $f(x_0, x_1, \ldots, x_s) = \begin{cases} 1 & i_1 \leq x_0 \leq i_2 \wedge (a_i < x_i < b_i)_{i \in [s]} \\ 0 & \text{otherwise.} \end{cases}$
The client sends a share of a function $f_j$ to server $j$.
Upon receiving a share of a function $f_j$ from the client, the server $j$ computes $(a_{j,1}, a_{j,2}) \leftarrow \texttt{Res}_j(\vec{x}, f_j)$ and sends $(a_{j,1}, a_{j,2})$ to the client.
Upon receiving $(a_{j,1}, a_{j,2})$ from server $j$, the client computes $a_0 \leftarrow a_{1,1} + a_{2,1}$ and $a_1 \leftarrow a_{1,2} + a_{2,2}$
The client outputs $(a_1, a_2)$.
```

The full construction of MDRange is shown in Protocol 4 in protocol form (i.e. via the interaction with Res). Note that the communication complexity of our $\langle \texttt{MDRange}, \texttt{Res}_1, \texttt{Res}_2 \rangle ((1, n, \perp, (a_i, b_i)_{i \in [s]}), \vec{x}, \vec{x})$ is $O(k \log n)$ in worst case.

Lastly, we show the construction of Index in Protocol 3 with the only difference being that the additional condition $(a_i, b_i)_{i \in [s]}$ required for the function secret sharing is set to match all possible values in the database. Note that the server response is the same whether the query from the client is a range query or an index query, and the Res algorithm is deterministic and stateless.

In the following, we prove the security of our MDRQ-PIR scheme Π. Specifically, we prove weak query indistinguishability on multi-dimensional databases for Π. In our security proof, we use a sequence of games. In adjacent games, a function share sent from client to server is replaced with another function share, and any adversary who distinguish adjacent game can be reduce to the security of the FSS scheme.

Theorem 27. *If FSS scheme $\mathcal{FSS} = (\mathsf{Gen}, \mathsf{Eval})$ is secure, then MDRQ-PIR scheme Π provides weak query indistinguishability on multi-dimensional databases.*

Proof. For all $\lambda \in \mathbb{N}$, for any dimension s, for any $V_u \in \mathbb{N}$ $(u = 1, \ldots, s)$, for any s dimensional database $\vec{x} = (rec_1, \ldots, rec_n)$ of size n, where $rec_t = (rec_{t,1}, \ldots, rec_{t,s})$ and $|rec_{t,u}| = V_u$ $(t = 1, \ldots, n)$, for any server $m \in [2]$, for any bounds $(a_w, b_w)_{w \in [s]}$, let $C \in \mathbb{N}$ be the number of FSSQuery queries required in the execution of $\langle \texttt{MDRange}, \texttt{Res}_1, \texttt{Res}_2 \rangle ((a_w, b_w)_{w \in [s]}, \vec{x}, \vec{x})$ and let $i_1, \ldots i_C \in [n]$ be any set of C indices. We consider a PPTA distinguisher $\mathcal{D}$ against weak query indistinguishability on multi-dimensional databases of Π for the multi-dimensional range query with bounds $(a_w, b_w)_{w \in [s]}$ and index queries for $i_1, \ldots i_C$. The advantage of $\mathcal{D}$ is defined by

$$\mathsf{Adv}^{ind}_{\mathcal{D},\Pi} = |\Pr[\mathcal{D}(\mathsf{View}^m_{mdrange}) = 1] - \Pr[\mathcal{D}(\mathsf{View}^m_{index}) = 1]|.$$

The transcript $\mathsf{View}^m_{mdrange}$ can be written as

$$\{\mathsf{Trans}_m(\langle \texttt{MDRange}, \texttt{Res}_1, \texttt{Res}_2 \rangle ((a_w, b_w)_{w \in [s]}, \vec{x}, \vec{x})\} = \{(\vec{q}, \vec{a}), \vec{x}\},$$

where $\vec{q} = (q_1, \ldots, q_C)$ and the i-th element in $\vec{q}$ is a query for server m generated from a function f_i by the client, and $\vec{a} = ((a_{1,1}, a_{1,2}), \ldots, (a_{C,1}, a_{C,2}))$ and $(a_{i,1}, a_{i,2})$ is the reply from server m for query q_i.

To obtain a proof, we use a sequence of games (**Game** 0 to **Game** C).

Protocol 3 $\langle \texttt{Index}, \texttt{Res}_1, \texttt{Res}_2\rangle(i, \vec{x}, \vec{x})$

```
The client and servers execute $((a_1, a_2), \perp, \perp) \leftarrow \texttt{FSSQuery}((i, i, (0, 2^{V_j} - 1)_{2 \leq j \leq s}), \texttt{Res}_1, \texttt{Res}_2)$
Then the client outputs $a_2$
```

Protocol 4 $\langle \texttt{MDRange}, \texttt{Res}_1, \texttt{Res}_2\rangle((i_1, i_2, k, (a_i, b_i)_{i \in [s]}), \vec{x}, \vec{x})$

```
if $k = \perp$ then
    Then execute $((d_1, d_2), \perp, \perp) \leftarrow \langle \texttt{FSSQuery}, \texttt{Res}_1, \texttt{Res}_2\rangle((i_1, i_2, (a_i, b_i)_{i \in [s]}), \vec{x}, \vec{x})$
    Client runs $\langle \texttt{MDRange}, \texttt{Res}_1, \texttt{Res}_2\rangle((i_1, i_2, d_1, (a_i, b_i)_{i \in [s]}), \vec{x}, \vec{x})$
else
    Then execute $((d'_1, d'_2), \perp, \perp) \leftarrow \langle \texttt{FSSQuery}, \texttt{Res}_1, \texttt{Res}_2\rangle((i_1, (i_1 + i_2)/2, (a_i, b_i)_{i \in [s]}), \vec{x}, \vec{x})$
    if $d'_1 = 0$ then
        return $\langle \texttt{MDRange}, \texttt{Res}_1, \texttt{Res}_2\rangle((i_1 + i_2)/2 + 1, i_2, k, (a_i, b_i)_{i \in [s]}), \vec{x}, \vec{x})$
    else if $d'_1 = 1$ then
        return $d'_2 || \langle \texttt{MDRange}, \texttt{Res}_1, \texttt{Res}_2\rangle((i_1 + i_2)/2 + 1, i_2, k - 1, (a_i, b_i)_{i \in [s]}), \vec{x}, \vec{x})$
    else if $d'_1 = k$ then
        $ans = \{\}$
        for $\ell = 0$ to $k - 1$ do
            $((d_{\ell_1}, d_{\ell_2}), \perp, \perp) \leftarrow \langle \texttt{FSSQuery}, \texttt{Res}_1, \texttt{Res}_2\rangle((i_1 + \ell, i_1 + \ell, (a_i, b_i)_{i \in [s]}), \vec{x}, \vec{x})$
            $ans \leftarrow ans || d_{\ell_2}$
            $\ell = \ell + 1$
        end for
        return $ans$
    else
        return $\langle \texttt{MDRange}, \texttt{Res}_1, \texttt{Res}_2\rangle((i_1, (i_1 + i_2)/2, d'_1, (a_i, b_i)_{i \in [s]}), \vec{x}, \vec{x}) ||$
        $\langle \texttt{MDRange}, \texttt{Res}_1, \texttt{Res}_2\rangle((1 + (i_1 + i_2)/2, i_2, k - d'_1, (a_i, b_i)_{i \in [s]}), \vec{x}, \vec{x})$
    end if
end if
```

Game 0**:** This game corresponds to the client and servers running $\langle \texttt{MDRange}, \texttt{Res}_1, \texttt{Res}_2\rangle$ $((a_w, b_w)_{w \in [s]}, \vec{x}, \vec{x})$.

Game r $(1 \leq r \leq C - 1)$**:** The difference from **Game** $r - 1$ is that q_{r-1} is replaced with q'_{r-1} where q'_{r-1} is a function share for server m generated from the function

$$f_{r-1}(x_1, \ldots, x_s) = \begin{cases} 1 & i_{r-1} - 1 < x_0 < i_{r-1} + 1 \\ & \wedge (0 < x_u < 2^{V_u} - 1)_{u \in [s]} \\ 0 & \text{otherwise.} \end{cases}$$

Note that f_{r-1} above corresponds to the function used in an index query.

Game C**:** This game corresponds to the client and servers running $\langle Index, Res_1, Res_2\rangle(i_r, \vec{x}, \vec{x})$ for $r = 1, \ldots, C$.

For all r, we denote by View_r^m the view for server m generated by the experiment **Game** r.

```
$\underline{\mathcal{A}_1(1^\lambda)}$
Send $f^0 := f_{r-1}$ and $f_1$ to challenger,
where $f_{r-1}$ is the $r-1$-th function used in the multi-dimensional range protocol for $(a_w, b_w)_{2w\in[s]}$,
and $f^1(x_1, \ldots, x_s) = \begin{cases} 1 & i_{r-1} - 1 < x_0 < i_{r-1} + 1 \land (0 < x_u < 2^{V_u} - 1)_{u\in[s]} \\ 0 & \text{otherwise.} \end{cases}$
output $st := (\{(a_w, b_w)_{w\in[s]}, (i_1, \ldots, i_{f(k)})\}, \vec{x})$
$\underline{\mathcal{A}_2(f_m^b, st)}$
Let $st := \{(a'_w, b'_w)_{w\in[s]}, (i'_1, \ldots, i'_{f(k)}), \vec{x'}(= (rec'_1, \ldots, rec'_n))\}$
Compute $z = \sum_{i=1}^n f_m^b(rec'_i), z' = \sum_{i=1}^n f_m^b(rec'_i) \cdot rec'_i$
Run $(q_1, a_{1,1}, a_{1,2}, \ldots, q_k, a_{k,1}, a_{k,2}) \leftarrow \mathsf{Trans}_m(\langle \mathtt{MDRange}, \mathtt{Res}_1, \ldots, \mathtt{Res}_2 \rangle((a'_w, b'_w)_{w\in[s]}, \vec{x'}, \vec{x'}))$
For $u = 1$ to $r - 2$
    $(q'_u, a'_{u,1}, a'_{u,2}) \leftarrow \mathsf{Trans}_m(\langle \mathtt{Index}, \mathtt{Res}_1, \mathtt{Res}_2 \rangle(i_u, \vec{x'}, \vec{x'}))$
$\mathsf{View} \leftarrow \{(q'_1, a'_{1,1}, a'_{1,2}, \ldots, q'_{r-2}, a'_{r-2,1}, a'_{r-2,2}, f_m^b, z, z', q_r, a_{r,1}, a_{r,2} \ldots, q_k, a_{k,1}, a_{k,2}), \vec{x}\}$
$b' \leftarrow \mathcal{B}(\mathsf{View})$
output $b'$
```

Figure 6.4.1: Construction of $\mathcal{A}$ in Lemma 23

Lemma 23. *If FSS scheme $\mathcal{FSS} = (\mathsf{Gen}, \mathsf{Eval})$ is secure, then for any $1 \leq r \leq C$ and for any PPTA $\mathcal{B}$, it holds that the difference between the probability that $\mathcal{B}$ outputs 1 in* **Game** *$r - 1$ and s is negligible.*

Proof. We consider a PPTA $\mathcal{B}$ who distinguishes View_{r-1}^m from $\mathsf{View}_{\mathbf{Game}\ r}^m$, and denote the advantage of $\mathcal{B}$ by

$$\mathsf{Adv}_{\mathcal{B},\Pi}^r = |\Pr[\mathcal{B}(\mathsf{View}_r^m) = 1] - \Pr[\mathcal{B}(\mathsf{View}_{r-1}^m) = 1]|.$$

Then, we construct an adversary $\mathcal{A}$ against $\mathcal{FSS}$ who uses $\mathcal{B}$ internally as shown in Figure 6.4.1.

By the construction of $\mathcal{A}$, $\mathcal{A}$ simulates View_{r-1}^m for $\mathcal{B}$ when $\mathcal{A}$ receives a function share of f^0 in the FSS security experiment. Moreover, $\mathcal{A}$ outputs 1 only when $\mathcal{B}$ outputs 1. Thus the probability that $\mathcal{A}$ outputs 1 in the experiment that $\mathcal{A}$ receives a function share of f^0 is equal to $\Pr[\mathcal{B}(\mathsf{View}_{r-1}^m) = 1]$. Likewise, the probability that $\mathcal{A}$ outputs 1 in the experiment that $\mathcal{A}$ receives a function share of f^1 is equal to $\Pr[\mathcal{B}(\mathsf{View}_r^m) = 1]$. Therefore, we obtain

$$\begin{aligned} \mathsf{Adv}_{\mathcal{B},\Pi}^r &= |\Pr[\mathcal{B}(\mathsf{View}_r^m) = 1] - \Pr[\mathcal{B}(\mathsf{View}_{r-1}^m) = 1]| \\ &= \mathsf{Adv}_{\mathcal{FSS}}(1^\lambda, \mathcal{A}). \end{aligned}$$

Since we assume $\mathcal{FSS}$ is secure i.e. that $\mathsf{Adv}_{\mathcal{FSS}}(1^\lambda, \mathcal{A})$ is negligible for all PTTA $\mathcal{A}$, we can conclude $\mathsf{Adv}_{\mathcal{B},\Pi}^r$ is negligible. □

By using Lemma 23, we can derive

$$
\begin{aligned}
&|\Pr[\mathcal{D}(\mathsf{View}^m_{mdrange}) = 1] \\
&\qquad\qquad - \Pr[\mathcal{D}(\mathsf{View}^m_{index}) = 1]| \\
&\leq \sum_{r=1}^{C} |\Pr[\mathcal{D}(\mathsf{View}^m_{r-1}) = 1] \\
&\qquad\qquad - \Pr[\mathcal{D}(\mathsf{View}^m_{r}) = 1]| \\
&\leq C \cdot negl.
\end{aligned}
$$

□

6.5 Conclusion

Most PIR schemes support only simple queries such as index queries and basic queries on simple databases. While the PIR scheme supporting basic range queries is a step in the right direction, we need to consider more complex databases and more complex queries to support real-world applications. In this chapter, we proposed the model of the PIR scheme supporting multi-dimensional range queries on multi-dimensional databases as complex queries and complex databases. We gave four security notions for the PIR scheme supporting multi-dimensional range queries: secure index queries on multi-dimensional databases, secure multi-dimensional range queries on multi-dimensional databases, weak query indistinguishability on multi-dimensional databases, and strong query indistinguishability on multi-dimensional databases. We also give proofs of the relationship among security notions we proposed in this chapter. More specifically, we proved that strong query indistinguishability on multi-dimensional databases implies all three other security notions. Then we gave the generic conversion from the scheme satisfying weak query indistinguishability on multi-dimensional databases to the scheme satisfying strong one. In addition, we gave the construction of the PIR scheme supporting multi-dimensional range queries satisfying weak query indistinguishability on multi-dimensional databases based on function secret sharing.

Considering PIR schemes on the databases more complex than multi-dimensional databases and constructing PIR schemes supporting more flexible queries such as SQL are future works. In addition, for the practical use of PIR, it is also important to combine the results of this research with the results of research on reducing server-side computation costs, which is a bottleneck of PIR protocol, and to implement and measure the performance of the proposed scheme.

Chapter 7

Conclusion

In this thesis, we showed the results of public-key encryption with keyword search and private information retrieval. In the following, we summarize the contributions of this thesis, and conclude this thesis with prospects.

7.1 Summary of Contributions

Public-Key Encryption with Keyword Search: In Chapter 3, we proposed the generic construction of a KP-ABE scheme whose access structure is specified by monotone Boolean formula from PEKS scheme whose search condition is specified by monotone Boolean formula. We also proved that if PEKS scheme Π satisfies perfect consistency and IND-CKA, then the KP-ABE scheme constructed from our generic construction satisfies correctness, IND-CPA, and IND-ANO-CPA.

In Chapter 4, we pointed out the existing definition of NM-RCCA was formalized arbitrarily. To give more rigorous formulations, we used simulation-based formalization, called SNM-RCCA. In addition, we also gave a game-based definition of NM-RCCA, called INM-RCCA. Then, we proved that the equivalence between our SNM-RCCA and INM-RCCA. Regarding secrecy against RCCA, we gave the definition of semantic security, called SS-RCCA. Like the case of NM, we gave the proof of equivalence between our SS-RCCA and IND-RCCA. Also, we proved that our INM-RCCA is equivalent to IND-RCCA proposed by Canetti et al. As a result, when we give the security against RCCA, we only need to give the proof of IND-RCCA.

Private Information Retrieval: In Chapter 5, we consider the PIR scheme supporting basic range queries. Although some PIR schemes support basic range queries, no rigorous proof had been given for those schemes. Therefore, it was not clear existing schemes supporting basic queries are secure or not. In fact, we pointed out that there are cases where existing schemes leak the information about the queries the client submits. To prevent such attacks, we rigorously formalize the security for PIR schemes supporting range queries. More specifically, we gave the definitions of secure index queries, secure range queries, and query indistinguishability. Then, we give the proofs that query indistinguishability implies other two security notions. In addition, we gave two constructions of the PIR scheme supporting range queries satisfying query indistinguishability. To show the usefulness of the proposed method in practical applications, we gave an implementation of the proposed scheme and measured its performance. The result showed that our proposed scheme is comparable to existing schemes.

In Chapter 6, we extended the result of Chapter 5. We consider the PIR scheme supporting multi-dimensional range queries on multi-dimensional databases. To take into account more realistic situations, we gave four security notions, secure index queries on multi-dimensional databases, secure range queries on multi-dimensional databases, weak query indistinguishability on multi-dimensional databases, and strong query indistinguishability on multi-dimensional databases. We proved that strong query indistinguishability on multi-dimensional databases implies other three security notions. In addition to this, we give the generic conversion from the scheme satisfying weak query indistinguishability on multi-dimensional databases to the scheme satisfying strong query indistinguishability on multi-dimensional databases. Then, we gave the construction of the PIR scheme supporting multi-dimensional range queries satisfying weak query indistinguishability on multi-dimensional databases based on function secret sharing. This is the first instantiation of a PIR scheme supporting multi-dimensional range queries while being capable of hiding the type of query being made and, in the case of multi-dimensional range queries, the number of elements retrieved in each query, when considering a stream of queries.

There are still many cases where information is leaked from databases due to the complexity of information retrieval systems. Furthermore, the structure of information retrieval systems is expected to become more complex in the future. Therefore, it is essential to rigorously formulate the security notions that can be used universally for increasingly complex and changing systems. Even though information retrieval systems have become more complex, the elements stored in the database are basically classified into plaintext or encrypted cases. Therefore, this research is expected to become a foundation of the security of information retrieval systems that will become more complex in the future. Furthermore, the security definitions we gave in each chapter capture more realistic attacks than existing ones. In particular, the formulation of query indistinguishability not only captures attacks that have not been captured by existing security definitions, but are also applicable to other systems. We will discuss this in more detail in the next section.

7.2 Future Prospects

Formulation of Security Models Capturing Realistic Attacks: In this thesis, we treated security for information retrieval systems. We believe that the ideas for the formulations of SNM-RCCA and query indistinguishability are useful for capturing more realistic attacks when considering security definitions for protocols. For example, in Chapter 4, we pointed out that it is not clear whether the existing game-based formulation of NM-RCCA captures the requirement of non-malleability. We gave a more rigorous formulation of security (i.e., simulation-based security) to solve this problem. This result suggests that simulation-based formulations are essential to capture the meaning of certain security notions. Currently, game-based security definitions, which are easy-to-use formulations, are widely used in many cryptography with advanced functionalities. Thus, the importance of verifying whether those formulations really capture the meaning of security will also be increased by this thesis.

In addition to the above, the idea of query indistinguishability can be extended to other protocols. For example, when considering a protocol that supports two types of functionalities, the security of not being able to distinguish externally whether or not either functionality is being performed can be formulated by simply applying the concept of query indistinguishability. Also, even if the number of functionalities the protocol supports increases, this concept can be easily applied. Therefore, the concept of query indistinguishability is useful to formulate the security for protocols.

Construction of Efficient Protocols for Complex Systems: In Chapter 6, we showed that even if we consider a security with a weakened definition of query indistinguishability in Chapter 5, it is not a problem depending on the system in which the protocol is used. In other words, we suggested that it is possible to increase security through systems other than cryptography. We believe that this idea will be useful for efficient protocol construction. For example, universal composability is one of the most important concepts for composing complex protocols, but protocols that satisfy such a strong security requirements are often inefficient. Therefore, it is necessary to think of security with an eye to the practical application of cryptographic protocols. Our approach is not to guarantee the security of the entire protocol with cryptography alone, but to consider the combination with existing systems. We believe that efficient construction is possible by considering the appropriate security notions for a particular protocol.

Practical Application of Secure Information Retrieval Systems: Since the first schemes were proposed, PIR and PEKS have not yet been implemented in society. This may be due to the efficiency problems and the flexibility of the queries supported by the schemes, but there are other factors. For example, since PIR and PEKS require the server-side to follow the protocol, companies such as search engine service providers need to implement PIR or PEKS to provide secure information retrieval services. Since those companies use information obtained from client queries to display appropriate advertisements, this information will no longer be available if PIR and PEKS are introduced. Therefore, there is no motivation for search engine service companies to introduce PIR and PEKS. However, it is also true that GDPR [107] and other moves to protect personal information occur worldwide. Therefore, it is only a matter of time before search engine service providers are forced to adopt technologies such as PIR and PEKS. Furthermore, research on the construction of secure information retrieval systems such as PIR and PEKS is expected to be actively conducted in the future, and their practical application is expected to progress as the times change. At that time, the required security and privacy requirements may also change. In this research, we have developed a theory that serves as a foundation for secure information retrieval systems. Therefore, the theory of security proposed in this thesis will be extended in the future and is expected to play an important role in considering the security requirements of that future era.

Bibliography

[1] UpGuard Team. The RNC Files: Inside the Largest US Voter Data Leak. 2017. https://www.upguard.com/breaches/the-rnc-files

[2] UpGuard Team. Black Box, Red Disk: How Top Secret NSA and Army Data Leaked Online. https://www.upguard.com/breaches/cloud-leak-inscom

[3] GIZMODO. Data Breach Exposed Medical Records, Including Blood Test Results, of Over 100 Thousand Patients. 2017 https://gizmodo.com/data-breach-exposedmedical-records-including-blood-te-1819322884

[4] Jack Cable, Drew Gregory, Liz Izhikevich, Zakir Durumeric. Stratosphere: Finding Vulnerable Cloud Storage Buckets. *24th International Symposium on Research in Attacks, Intrusions and Defenses (RAID '21)*, pp.399–411, 2021.

[5] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, vol.22, no.6, pp.644–654, 1976.

[6] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, vol.21, no.2, pp.120–126, 1978.

[7] Michael O. Rabin. Digitalized Signatures and Public-key Functions as Intractable as Factorization. *In Massachusetts Institute of Technology*, 1979.

[8] Taher El Gamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *Advances in Cryptology, Proceedings of CRYPTO '84*, Lecture Notes in Computer Science, vol.196, pp.10–18, 1984.

[9] Eiichiro Fujisaki, Tatsuaki Okamoto, David Pointcheval, and Jacques Stern. RSA-OAEP Is Secure under the RSA Assumption. *Advances in Cryptology, Proceedings of CRYPTO '01*, Lecture Notes in Computer Science, vol.2139, pp.260-274, 2001.

[10] Phillip Rogaway and Mihir Bellare. Optimal Asymmetric Encryption. *Advances in Cryptology - EUROCRYPT '94*, Lecture Notes in Computer Science, vol.950, pp.92–111, 1994.

[11] Ronald Cramer and Victor Shoup. A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack. *Advances in Cryptology, Proceedings of CRYPTO '98*, Lecture Notes in Computer Science, vol.1462, pp.13-25, 1998.

[12] Elette Boyle, Niv Gilboa, and Yuval Ishai. Function Secret Sharing. *Advances in Cryptology - EUROCRYPT 2015*, Lecture Notes in Computer Science, vol.9057, pp.337–367, 2015

[13] Elette Boyle, Niv Gilboa, and Yuval Ishai. Function Secret Sharing: Improvements and Extensions. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp.1292-1303, 2016.

[14] Dawn Xiaodong Song, David A. Wagner, and Adrian Perrig. Practical Techniques for Searches on Encrypted Data. *Proceeding 2000 IEEE Symposium on Security and Privacy. (S&P 2000)*, pp.44–55, 2000.

[15] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public Key Encryption with Keyword Search. *Advances in Cryptology - EUROCRYPT 2004*, Lecture Notes in Computer Science, vol.3027, pp.506–522, 2004.

[16] Salam Md Iftekhar, Yau Wei-Chuen Chin, Ji-Jian, Heng Swee-Huay, Ling Huo-Chong, Phan Raphael C-W, Poh Geong Sen, Tan Syh-Yuan, and Yap Wun-She. Implementation of searchable symmetric encryption for privacy-preserving keyword search on cloud storage. *Human-centric Computing and Information Sciences*, vol.5, no.19, 2015.

[17] Eu-Jin Goh. Secure Indexes. *Cryptology ePrint Archive*, Report 2003/216, 2003.

[18] Mariana Raykova, Binh Vo, Steven M. Bellovin, and Tal Malkin. Secure anonymous database search. *Proceedings of the first ACM Cloud Computing Security Workshop, CCSW 2009*, pp.115–126, 2009.

[19] Takanori Suga, Takashi Nishide, and Kouichi Sakurai. Secure Keyword Search Using Bloom Filter with Specified Character Positions. *Provable Security - 6th International Conference, ProvSec 2012*, Lecture Notes in Computer Science, vol.7496, pp.235–252, 2012.

[20] Seny Kamara and Charalampos Papamanthou. Parallel and Dynamic Searchable Symmetric Encryption. *Financial Cryptography and Data Security - 17th International Conference, FC 2013*, Lecture Notes in Computer Science, vol.7859, pp.258–274, 2013.

[21] Ke Yuan, Zheli Liu, Chunfu Jia, and Jun Yang, Shuwang Lv. Multi-user Public Key Timed-Release Searchable Encryption. *2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies*, pp.363–370, 2013.

[22] Changjiang Hou, Fei Liu, Hongtao Bai, and Lanfang Ren. Public-key searchable encryption from lattices. *International Journal of High Performance Systems Architecture*, vol.5, no.1, pp.25–32, 2014.

[23] Bing Wang, Wei Song, Wenjing Lou, and Y. Thomas Hou. Inverted index based multi-keyword public-key searchable encryption with strong privacy guarantee. *2015 IEEE Conference on Computer Communications (INFOCOM)*, pp.2092–2100, 2015.

[24] Raphael Bost. $\Sigma o\varphi o\varsigma$: Forward Secure Searchable Encryption. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp.1143–1154, 2016.

[25] Min-Shiang Hwang, Shih-Ting Hsu, and Cheng-Chi Lee. A New Public Key Encryption with Conjunctive Field Keyword Search Scheme. *Information Technology and Control*, vol.43, no.3, pp.277–288, 2014.

[26] Yu Zhang, Yin Li, and Yifan Wang. Conjunctive and Disjunctive Keyword Search over Encrypted Mobile Cloud Data in Public Key System. *Mobile Information Systems* vol. 2018, article ID: 3839254, pp.1–11, 2018.

[27] Adi Shamir. Identity-Based Cryptosystems and Signature Schemes. *Advances in Cryptology, Proceedings of CRYPTO '84*, Lecture Notes in Computer Science, vol.196, pp.47–53, 1984.

[28] Dan Boneh and Matthew K. Franklin. Identity-Based Encryption from the Weil Pairing. *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference*, Lecture Notes in Computer Science, vol.2139, pp.213–229, 2001.

[29] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006*, pp.89–98, 2006.

[30] Dan Boneh, Amit Sahai, and Brent Waters. Functional Encryption: Definitions and Challenges. *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011*, Lecture Notes in Computer Science, vol.6597, pp.253-273, 2011.

[31] Pascal Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques*, Lecture Notes in Computer Science, vol.1592, pp.223-238, 1999.

[32] Matt Blaze, Gerrit Bleumer, and Martin Strauss. Divertible Protocols and Atomic Proxy Cryptography. *Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques*, Lecture Notes in Computer Science, vol.1403, pp.127-144, 1998.

[33] Michel Abdalla, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kohno, Tanja Lange, John Malone-Lee, Gregory Neven, Pascal Paillier, and Haixia Shi. Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions. *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference*, Lecture Notes in Computer Science, vol.3621, pp.205–222.

[34] Fei Han, Jing Qin, Huawei Zhao, and Jiankun Hu. A general transformation from KP-ABE to searchable encryption. *Future Generation Computer Systems*, vol.30, pp.107–115, 2014.

[35] Reza Curtmola, Juan A. Garay, Seny Kamara, and Rafail Ostrovsky. Searchable symmetric encryption: Improved definitions and efficient constructions. *Journal of Computer Security*, vol.19, no.5, pp.895–934, 2011.

[36] David Cash, Stanislaw Jarecki, Charanjit S. Jutla, Hugo Krawczyk, Marcel-Catalin Rosu, and Michael Steiner. Highly-Scalable Searchable Symmetric Encryption with Support for Boolean Queries. *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference*, Lecture Notes in Computer Science, vol.8042, pp.353-373, 2013.

[37] Seny Kamara and Tarik Moataz. Boolean Searchable Symmetric Encryption with Worst-Case Sub-linear Complexity. *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Lecture Notes in Computer Science, vol.10212, pp.94–124, 2017.

[38] Bo Zhang and Fangguo Zhang. An efficient public key encryption with conjunctive-subset keywords search. *Journal of Network and Computer Applications*, vol.34, no.1, pp.262–267, 2011.

[39] Dan Boneh and Brent Waters. Conjunctive, Subset, and Range Queries on Encrypted Data. *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007*, Lecture Notes in Computer Science, vol.4392, pp.535–554.

[40] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-Policy Attribute-Based Encryption. *2007 IEEE Symposium on Security and Privacy (S&P 2007)*, pp.321–334, 2007.

[41] Dong Jin Park, Kihyun Kim, and Pil Joong Lee. Public Key Encryption with Conjunctive Field Keyword Search. *Information Security Applications, 5th International Workshop, WISA 2004*, pp.73-86, 2004.

[42] Jun Shao, Zhenfu Cao, Xiaohui Liang, and Huang Lin. Proxy re-encryption with keyword search. *Information Sciences*, vol.180, issue.13, pp.2576–2587, 2010.

[43] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-Malleable Cryptography (Extended Abstract). *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, pp.542–552, 1991.

[44] Mihir Bellare and Amit Sahai. Non-Malleable Encryption: Equivalence between Two Notions, and an Indistinguishability-based Characterization. *Cryptology ePrint Archive*, Report 2006/228, 2006.

[45] Mihir Bellare and Amit Sahai. Non-malleable Encryption: Equivalence between Two Notions, and an Indistinguishability-Based Characterization. *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference*, Lecture Notes in Computer Science, vol.1666, pp.519–536, 1999.

[46] Ran Canetti, Hugo Krawczyk, and Jesper Buus Nielsen. Relaxing Chosen-Ciphertext Security. *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference*, Lecture Notes in Computer Science, vol.2729, pp.565–582, 2003.

[47] Shafi Goldwasser and Silvio Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, vol.28, no.2, pp.270–299, 1984

[48] Wei Chuen Yau , Raphael W. Phan, Swee Huay Heng, and Bok Min Goi. Proxy Re encryption with Keyword Search: New Definitions and Algorithms. *Security Technology, Disaster Recovery and Business Continuity - International Conferences, SecTech and DRBC 2010, Held as Part of the Future Generation Information Technology Conference, FGIT 2010*, pp.149–160, 2010.

[49] Yang Yang and Maode Ma. Conjunctive Keyword Search With Designated Tester and Timing Enabled Proxy Re Encryption Function for E Health Clouds. *IEEE Transactions on Information Forensics and Security*, vol.11, no.4, pp.746–759, 2016.

[50] Zhenhua Chen, Shundong Li, Qiong Huang, Yilei Wang, and Sufang Zhou. A restricted proxy re encryption with keyword search for fine grained data access control in cloud storage. *Concurrency and Computation Practice and Experience*, vol.28, no.10, pp.2858–2876, 2016.

[51] Moni Naor and Moti Yung. Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, pp.427-437, 1990.

[52] Charles Rackoff and Daniel R. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. *Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference*, Lecture Notes in Computer Science, vol.576, pp.433–444, 1991.

[53] Jens Groth. Rerandomizable and Replayable Adaptive Chosen Ciphertext Attack Secure Cryptosystems. *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004*, Lecture Notes in Computer Science, vol.2951, pp.152–170, 2004.

[54] Dana Dachman-Soled, Georg Fuchsbauer, Payman Mohassel, and Adam O'Neill. Enhanced Chosen-Ciphertext Security and Applications. *Public-Key Cryptography - PKC 2014 - 17th International Conference on Practice and Theory in Public-Key Cryptography*, Lecture Notes in Computer Science, vol.8383, pp.329–344, 2014.

[55] Honglong Dai, Jinying Chang, Zhenduo Hou, and Maozhi Xu. Relaxing Enhanced Chosen-Ciphertext Security. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol.101-A, no.12, pp.2454–2463, 2018.

[56] Sumit Kumar Pandey, Santanu Sarkar, and Mahabir Prasad Jhanwar. Relaxing IND-CCA: Indistinguishability against Chosen Ciphertext Verification Attack. *Security, Privacy, and Applied Cryptography Engineering - Second International Conference, SPACE 2012, pp.63–76, 2012.*

[57] Yodai Watanabe, Junji Shikata, and Hideki Imai. Equivalence between Semantic Security and Indistinguishability against Chosen Ciphertext Attacks. *Public Key Cryptography - PKC 2003, 6th International Workshop on Theory and Practice in Public Key Cryptography*, Lecture Notes in Computer Science, vol.2567, pp.71–84, 2003.

[58] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations Among Notions of Security for Public-Key Encryption Schemes. *Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference*, Lecture Notes in Computer Science, vol.1462, pp.26–45, 1998.

[59] Rafael Pass, abhi shelat, and Vinod Vaikuntanathan. Relations Among Notions of Non-malleability for Encryption. *Advances in Cryptology - ASIACRYPT 2007, 13th International Conference on the Theory and Application of Cryptology and Information Security*, Lecture Notes in Computer Science, vol.4833, pp.519–535, 2007.

[60] Jonathan Katz and Moti Yung. Characterization of Security Notions for Probabilistic Private-Key Encryption. *Journal of Cryptology*, vol.19, no.1, pp.67–95, 2006.

[61] Benoît Libert, and Damien Vergnaud. Unidirectional Chosen-Ciphertext Secure Proxy Re-encryption. *Public Key Cryptography - PKC 2008, 11th International Workshop on Practice and Theory in Public-Key Cryptography*, Lecture Notes in Computer Science, vol.4939, pp.360–379, 2008.

[62] Keying Li, Jianfeng Wang, Yinghui Zhang, and Hua Ma. Key Policy Attribute-based Proxy Re-encryption and RCCA Secure Scheme. *Journal of Internet Services and Information Security*, vol.4, no.2, pp.70–82, 2014.

[63] Rongxing Lu, Xiaodong Lin, Jun Shao, and Kaitai Liang. RCCA-Secure Multi-use Bidirectional Proxy Re-encryption with Master Secret Security. *Provable Security - 8th International Conference, ProvSec 2014*, Lecture Notes in Computer Science, vol.8782, pp.194–205, 2014.

[64] Yuan Chen and Qingkuan Dong. RCCA security for KEM+DEM style hybrid encryptions and a general hybrid paradigm from RCCA-secure KEMs to CCA-secure encryptions. *Security and Communication Networks*, vol.7, no.8, pp.1219–1231, 2014.

[65] Masayuki Abe, Rosario Gennaro, and Kaoru Kurosawa. Tag-KEM/DEM: A New Framework for Hybrid Encryption. *Journal of Cryptology*, vol.21, no.1, pp.97–130, 2008.

[66] Honglong Dai, Ding Wang, Jinyong Chang, and Maozhi Xu. On the RCCA Security of Hybrid Signcryption for Internet of Things. *Wireless Communications and Mobile Computing*, vol.2018, article ID.8646973, pp.1–11, 2018.

[67] Michael Backes and Christian Cachin. Public-Key Steganography with Active Attacks. *Theory of Cryptography, Second Theory of Cryptography Conference, TCC 2005*, Lecture Notes in Computer Science, vol.3378, pp.210-226, 2005.

[68] Manoj Prabhakaran and Mike Rosulek. Rerandomizable RCCA Encryption. *Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference*, Lecture Notes in Computer Science, vol.4622, pp.517–534, 2007.

[69] Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Sarah Meiklejohn. Malleable Proof Systems and Applications. *Advances in Cryptology - EUROCRYPT 2012*

- 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lecture Notes in Computer Science, vol.7237, pp.281–300, 2012.

[70] Benoît Libert, Thomas Peters, and Chen Qian. Structure-Preserving Chosen-Ciphertext Security with Shorter Verifiable Ciphertexts. *Public-Key Cryptography - PKC 2017 - 20th IACR International Conference on Practice and Theory in Public-Key Cryptography*, Lecture Notes in Computer Science, vol.10174, pp.247–276, 2017.

[71] Dennis Hofheinz and Eike Kiltz. Secure Hybrid Encryption from Weakened Key Encapsulation. *Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference*, Lecture Notes in Computer Science, vol.4622, pp.553–571, 2007.

[72] Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. *Cryptology ePrint Archive*, Report 2004/332, 2004.

[73] Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private Information Retrieval. *36th Annual Symposium on Foundations of Computer Science*, pp.41–50, 1995.

[74] Eyal Kushilevitz and Rafail Ostrovsky. Replication is NOT Needed: SINGLE Database, Computationally-Private Information Retrieval. *38th Annual Symposium on Foundations of Computer Science, FOCS '97*, pp.364–373, 1997.

[75] Zeev Dvir and Sivakanth Gopi. 2-Server PIR with Sub-Polynomial Communication. *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015*, pp.577–584, 2015.

[76] Christian Cachin, Silvio Micali, and Markus Stadler. Computationally Private Information Retrieval with Polylogarithmic Communication. *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques*, Lecture Notes in Computer Science, vol.1592, pp.402–414, 1999.

[77] Changyu Dong and Liqun Chen. A Fast Single Server Private Information Retrieval Protocol with Low Communication Cost. *Computer Security - ESORICS 2014 - 19th European Symposium on Research in Computer Security*, Lecture Notes in Computer Science, vol.8712, pp.380–399, 2014.

[78] Bijit Hore, Sharad Mehrotra, and Gene Tsudik. A Privacy-Preserving Index for Range Queries. *Proceedings of the Thirtieth International Conference on Very Large Data Bases, VLDB 2004*, pp.720–731, 2004.

[79] Seny Kamara and Tarik Moataz. SQL on Structurally-Encrypted Databases. *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security*, Lecture Notes in Computer Science, vol.11272, pp.149–180, 2014.

[80] David Cash, Paul Grubbs, Jason Perry, and Thomas Ristenpart. Leakage-Abuse Attacks Against Searchable Encryption. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp.668–679, 2015.

[81] Georgios Kellaris, George Kollios, Kobbi Nissim, and Adam O'Neill. Generic Attacks on Secure Outsourced Databases. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp.1329–1340, 2016.

[82] Paul Grubbs, Marie-Sarah Lacharité, Brice Minaud, and Kenneth G. Paterson. Pump up the Volume: Practical Database Reconstruction from Volume Leakage on Range Queries. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018*, pp.315–331, 2018.

[83] Zichen Gui, Oliver Johnson, and Bogdan Warinschi. Encrypted Databases: New Volume Attacks against Range Queries. *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019*, pp.361–378, 2019.

[84] Frank Wang, Catherine Yun, Shafi Goldwasser, Vinod Vaikuntanathan, and Matei Zaharia. Splinter: Practical Private Queries on Public Data. *14th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2017*, pp.299–313, 2017.

[85] Jun Li and Edward Omiecinski. Efficiency and Security Trade-Off in Supporting Range Queries on Encrypted Databases. *Data and Applications Security XIX, 19th Annual IFIP WG 11.3 Working Conference on Data and Applications Security*, Lecture Notes in Computer Science, vol.3654, pp.69–83, 2005.

[86] Keke Chen, Ramakanth Kavuluru, and Shumin Guo. RASP: efficient multidimensional range query on attack-resilient encrypted databases. *First ACM Conference on Data and Application Security and Privacy, CODASPY 2011*, pp.249–260, 2011.

[87] Amos Beimel, Yuval Ishai, and Eyal Kushilevitz. General constructions for information-theoretic private information retrieval. *Journal of Computer and System Sciences*, vol.71, no.2, pp.213–247, 2005.

[88] Zeev Dvir and Sivakanth Gopi. 2-Server PIR with Sub-Polynomial Communication. *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015*, pp.577–584, 2015.

[89] Andris Ambainis. Upper Bound on Communication Complexity of Private Information Retrieval. *Automata, Languages and Programming, 24th International Colloquium ICALP'97*, Lecture Notes in Computer Science, vol.1256, pp.401–407, 1997.

[90] Amos Beimel, Yuval Ishai, Eyal Kushilevitz, and Jean-François Raymond. Breaking the $O(n^{1/(2k-1)})$ Barrier for Information-Theoretic Private Information Retrieval. *43rd Symposium on Foundations of Computer Science (FOCS 2002)*, pp.261-270, 2002.

[91] Benny Chor and Niv Gilboa. Computationally Private Information Retrieval (Extended Abstract). *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing*, pp.304–313, 1997.

[92] Niv Gilboa and Yuval Ishai. Distributed Point Functions and Their Applications *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Lecture Notes in Computer Science, vol.8441, pp.640–658, 2014.

[93] Craig Gentry and Zul

kar Ramzan. Single-Database Private Information Retrieval with Constant Communication Rate. *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005*, Lecture Notes in Computer Science, vol.3580, pp.803–815, 2005.

[94] Yarkin Doröz, Berk Sunar, and Ghaith Hammouri. Bandwidth Efficient PIR from NTRU *Financial Cryptography and Data Security - FC 2014 Workshops, BITCOIN and WAHC 2014*, Lecture Notes in Computer Science, vol.8438, pp.195–207, 2014.

[95] Benny Chor, Niv Gilboa, and Moni Naor. Private Information Retrieval by Keywords. *Cryptology ePrint Archive*, Report 1998/3, 1998.

[96] Gamze Tillem, Ömer Mert Candan, Erkay Savas, and Kamer Kaya. Hiding Access Patterns in Range Queries Using Private Information Retrieval and ORAM. *Financial Cryptography and Data Security - FC 2016 International Workshops, BITCOIN, VOTING, and WAHC*, Lecture Notes in Computer Science, vol.9604, pp.253–270, 2016.

[97] Jens Groth, Aggelos Kiayias, and Helger Lipmaa. Multi-query Computationally-Private Information Retrieval with Constant Communication Rate. *Public Key Cryptography - PKC 2010, 13th International Conference on Practice and Theory in Public Key Cryptography*, Lecture Notes in Computer Science, vol.6056, pp.107–123, 2010.

[98] Amos Beimel, Yuval Ishai, and Tal Malkin. Reducing the Servers Computation in Private Information Retrieval: PIR with Preprocessing. *Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference*, Lecture Notes in Computer Science, vol.1880, pp.55–73, 2000.

[99] Ran Canetti, Justin Holmgren, and Silas Richelson. Towards Doubly Efficient Private Information Retrieval. *Theory of Cryptography - 15th International Conference, TCC 2017*, Lecture Notes in Computer Science, vol.10678, pp.694–726, 2017.

[100] Henry Corrigan-Gibbs and Dmitry Kogan. Private Information Retrieval with Sublinear Online Time. *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Lecture Notes in Computer Science, vol.12105, pp.44–75, 2020.

[101] Elaine Shi, Waqar Aqeel, Balakrishnan Chandrasekaran, and Bruce M. Maggs. Puncturable Pseudorandom Sets and Private Information Retrieval with Near-Optimal Online Bandwidth and Time. *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021*, Lecture Notes in Computer Science, vol.12828, pp.641–669, 2021.

[102] Roger Dingledine, Nick Mathewson, and Paul F. Syverson. Tor: The Second-Generation Onion Router. *Proceedings of the 13th USENIX Security Symposium*, pp.303–320, 2004.

[103] Josep Domingo-Ferrer, Maria Bras-Amorós, Qianhong Wu, and Jesús A. Manjón. User-private information retrieval based on a peer-to-peer community. *Data & Knowledge Engineering*, vol.68, no.11, pp.1237–1252, 2009.

[104] LibFSS. https://github.com/frankw2/libfss

[105] Stefan Sprenger, Patrick Schäfer, and Ulf Leser Multidimensional range queries on modern hardware. *Proceedings of the 30th International Conference on Scientific and Statistical Database Management, SSDBM 2018*, pp.4:1–4:12, 2018.

[106] Colleen M. Swanson and Douglas R. Stinson. Extended results on privacy against coalitions of users in user-private information retrieval protocols. *Cryptography and Communications*, vol.7, no.4, pp.415–437, 2015.

[107] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). https://eur-lex.europa.eu/eli/reg/2016/679/oj

Appendix A

List of Publications

Refereed Papers

1. Junichiro Hayata, Masahito Ishizaka, Yusuke Sakai, Goichiro Hanaoka, Kanta Matsuura. Generic Construction of Adaptively Secure Anonymous Key-Policy Attribute-Based Encryption from Public-Key Searchable Encryption. *2018 International Symposium on Information Theory and Its Applications (ISITA2018)*, IEEE, pp.707–711, 2018.

2. Junichiro Hayata, Fuyuki Kitagawa, Yusuke Sakai, Goichiro Hanaoka, Kanta Matsuura. Equivalence Between Non-malleability Against Replayable CCA and Other RCCA-Security Notions. *Advances in Information and Computer Security - 14th International Workshop on Security (IWSEC2019)*, Lecture Notes in Computer Science, vol. 11689, Springer, pp.253–272, 2019.

3. Junichiro Hayata, Jacob C. N. Schuldt, Goichiro Hanaoka, Kanta Matsuura. On Private Information Retrieval Supporting Range Queries. *25th European Symposium on Research in Computer Security (ESORICS2020)*, Lecture Notes in Computer Science, vol. 12309, Springer, pp.674–694, 2020.

4. Junichiro Hayata, Masahito Ishizaka, Yusuke Sakai, Goichiro Hanaoka, Kanta Matsuura. Generic Construction of Adaptively Secure Anonymous Key-Policy Attribute-Based Encryption from Public-Key Searchable Encryption. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 2020*, IEICE, vol. 103-A, Issue 1, pp.107–113, 2020.

5. Junichiro Hayata, Fuyuki Kitagawa, Yusuke Sakai, Goichiro Hanaoka, Kanta Matsuura. Equivalence Between Non-malleability Against Replayable CCA and Other RCCA-Security Notions. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 2021*, IEICE, vol. 104-A, Issue 1, pp.89–103, 2021.

Non-refereed Papers

1. 林田淳一郎, 石坂理人, 坂井祐介, 花岡悟一郎, 松浦幹太. 公開鍵型検索可能暗号を用いた適応的安全な匿名鍵ポリシー型属性ベース暗号の一般的構成. *2018年 暗号と情報セキュリティシンポジウム (SCIS2018)*, 2018.

2. 林田淳一郎, 北川冬航, 坂井祐介, 花岡悟一郎, 松浦幹太. 公開鍵暗号の Replayable CCA 環境下での安全性概念間の等価性について. *2019年 暗号と情報セキュリティシンポジウム (SCIS2019)*, 2019.

3. 林田淳一郎, Jacob C. N. Schuldt, 花岡悟一郎, 松浦幹太. A Private Information Retrieval Scheme Supporting Range Queries. *2020年 暗号と情報セキュリティシンポジウム (SCIS2020)*, 2020.

4. Junichiro Hayata, Jacob C. N. Schuldt, Goichiro Hayata, Kanta Matsuura. On Private Information Retrieval Supporting Multi-dimensional Range Queries. *2021年 暗号と情報セキュリティシンポジウム (SCIS2021)*, 2021.

5. 林リウヤ, 浅野泰輝, 林田淳一郎, 松田隆宏, 山田翔太, 勝又秀一, 坂井祐介, 照屋唯紀, シュルツ ヤコブ, アッタラパドゥン ナッタポン, 花岡悟一郎, 松浦幹太, 松本勉. モノの電子署名：物体に署名するための一検討. *コンピュータセキュリティシンポジウム 2021 (CSS2021)*, 2021.

Awards

1. コンピュータセキュリティシンポジウム 2018 最優秀デモンストレーション賞. 秘匿依頼計算アプリ開発のための汎用ライブラリ. 受賞者: 森遼太, 光成滋生, 照屋唯紀, 浅井潔, 岡田大弥, 北井宏昌, 小松みさき, 橋本侑知, 林田淳一郎, 花岡悟一郎. *コンピュータセキュリティシンポジウム 2018 (CSS2018)*, 2018.