論文 / 著書情報
Article / Book Information

題目(和文)	公開鍵暗号系における匿名性
Title(English)	Anonymity on public-key cryptosystems
著者(和文)	林良太郎
Author(English)	RYOTARO HAYASHI
出典(和文)	学位:博士（理学）, 学位授与機関:東京工業大学, 報告番号:甲第6764号, 授与年月日:2007年3月26日, 学位の種別:課程博士, 審査員:
Citation(English)	Degree:Doctor of Science, Conferring organization: Tokyo Institute of Technology, Report number:甲第6764号, Conferred date:2007/3/26, Degree Type:Course doctor, Examiner:
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Anonymity on Public-Key Cryptosystems

Ryotaro Hayashi

PhD Thesis

Dept. of Mathematical and Computing Sciences
Graduate School of Information Science and Engineering
Tokyo Institute of Technology

Supervisor Keisuke Tanaka

November 21, 2006

Abstract

In this thesis, we study the security property for encryption and signature schemes, called "anonymity." Roughly speaking, it is said that an encryption scheme provides the anonymity when the eavesdropper, in possession of a ciphertext, cannot determine who is the receiver of the ciphertext. That is, the receiver is anonymous from the point of view of the eavesdropper. For signature schemes, it is said that a signature scheme provides the anonymity when it is infeasible to determine who produced the signature. Some signature schemes with special functionality, such as undeniable and confirmer signature schemes and ring signature schemes, require the anonymity property.

In the first half of this thesis, we study the techniques which can be used to obtain the RSA-based schemes with the anonymity property. In order to construct the schemes for encryption or signature with the anonymity property, it is necessary that the space of ciphertexts or signatures is common to each user. We propose two techniques for anonymity, and by using these techniques, the space of ciphertexts or signatures of RSA based schemes can be common to each user. We also construct the schemes for encryption, undeniable and confirmer signature, and ring signature, by applying our proposed techniques, and show the advantage and the disadvantage of the previous and our proposed schemes.

In the second half of this thesis, we carry on further research with respect to the anonymity property of encryption schemes. We first construct a family of Paillier's trap-door permutations with a common domain, and propose the schemes for public-key encryption with our proposed families of trap-door permutations. We next propose a new security notion for public-key encryption with anonymity, called "strong anonymity," and show the relationships between the data-privacy and the key-privacy for public-key encryption schemes. Furthermore, we propose a new security notion of plaintext awareness in the two-key setting, called PATK, and show that PATK implies IK-CCA, which is considered

as the required security level with respect to the anonymity property. We also propose the first generic conversion scheme for the anonymity from IK-CPA, which is a weaker security notion than IK-CCA, to IK-CCA. Finally, we formalize a special type of public-key encryption schemes called a universally anonymizable public-key encryption scheme. We then propose the universally anonymizable public-key encryption schemes based on the ElGamal encryption scheme, the Cramer-Shoup encryption scheme, and RSA-OAEP, and prove their security.

Acknowledgements

I would like to express my gratitude to all those who gave me the possibility to complete this thesis.

I would like to express my deep and sincere gratitude to my supervisor, Professor Keisuke Tanaka. This thesis would not have been begun, let alone finished, without his guidance. His wide knowledge has been of great value for me, and his understanding, encouraging, and personal guidance have provided a good basis for the present thesis. Besides of being an excellent supervisor, he always has a warm relationship with me and I have enjoyed an active and fulfilling student life in his laboratory. I am really glad that I have met him in my life and spent five worthwhile years with him.

I would also like to express my profound thanks to Dr. Tatsuaki Okamoto. At the beginning of the research, I got an opportunity to discuss with him. Thanks to his helpful advice, we wrote the paper and I made a presentation at Singapore. This was the first time that I talked at the international conference.

My sincere thanks are due to the official referees, Professor Ryo Kashima, Sadayoshi Kojima, Etsuya Shibayama, Ken Wakita, and Osamu Watanabe, for their detailed review and excellent advice during the preparation of this thesis. Especially, I wish to express my warm and sincere thanks to Professor Osamu Watanabe. I joined in his laboratory when I was a bachelor student, and his logical way of thinking is based on my research.

This research is supported in part by NTT Information Sharing Platform Laboratories and Grant-in-Aid for Scientific Research, Ministry of Education, Culture, Sports, Science, and Technology. Their support is greatly appreciated.

Needless to say, I am grateful to all of my colleagues at Tanaka laboratory. They all gave me the feeling of being at home at work, and I have many precious and wonderful memories with them.

I would like to thank my all friends for their encouragement and their cooperative spirit, and for the nice times we spent together.

Last but not least, I would like to express my deepest appreciation to my father, mother, sister, and grandmother for their heartfelt encouragement and support, and would like to dedicate this thesis to them.

Again, I would like to thank everybody who was important to the successful realization of thesis, as well as expressing my apology that I could not mention personally one by one.

Contents

CHAPTER 1

Introduction

In this thesis, we study the security property for encryption and signature schemes, called "anonymity." Roughly speaking, it is said that an encryption scheme provides the anonymity when the eavesdropper, in possession of a ciphertext, cannot determine who is the receiver of the ciphertext. That is, the receiver is anonymous from the point of view of the eavesdropper. For signature schemes, it is said that a signature scheme provides the anonymity when it is infeasible to determine who produced the signature. Some signature schemes with special functionality, such as undeniable and confirmer signature schemes and ring signature schemes, require the anonymity property.

In Chapters 2 to 5, we study the techniques which can be used to obtain the RSA-based schemes with the anonymity property. We propose two techniques for anonymity. We also construct the schemes for public-key encryption, undeniable and confirmer signature, and ring signature, by applying our proposed techniques.

In Chapter 6, We construct a family of Paillier's trap-door permutations and that with a common domain. We also propose the schemes for public-key encryption with our proposed families of trap-door permutations.

In Chapter 7, we propose a new security notion for public-key encryption with anonymity, called "strong anonymity," and show the relationships between the data-privacy and the key-privacy for public-key encryption schemes.

In Chapter 8 we propose a new security notion of plaintext awareness in the two-key setting, called PATK, and show that PATK implies IK-CCA. We also propose the first

generic conversion scheme for the anonymity from IK-CPA to IK-CCA.

In Chapter 9, we formalize a special type of public-key encryption schemes called a universally anonymizable public-key encryption scheme. We then propose the universally anonymizable public-key encryption schemes based on the ElGamal encryption scheme, the Cramer-Shoup encryption scheme, and RSA-OAEP, and prove their security.

1.1 Techniques and Schemes for Public-Key Encryption and Signature with Anonymity

1.1.1 Background

We review public-key encryption, undeniable and confirmer signature, and ring signature, and the anonymity properties for them.

Public-Key Encryption In 1976, Diffie and Hellman published the idea of public-key cryptography in their famous paper [36]. They introduced a public-key method for key agreement which is in use to this day. In addition, they described how digital signatures would work, and proposed, as an open question, the search for such a function. The first public-key cryptosystem that could function as both key agreement and as digital signature was the RSA cryptosystem published in 1978 by Rivest, Shamir, and Adleman [75]. The RSA cryptosystem provides encryption and digital signatures and is the most popular and widely used public-key cryptosystem today.

Until up now, many public-key encryption schemes have been proposed, and the security notions for public-key encryption have also been proposed. A convenient way to define the security notions for public-key encryption is by considering separately the various possible goals and the various possible attack models, and then obtain each definition as a pairing of a particular goal and a particular attack model.

The classical security goal (requirement) of public-key encryption schemes is that it provides privacy of the encrypted data. Popular formalizations such as the indistinguishability of encryptions by Goldwasser and Micali [48], or the non-malleability by Dolev, Dwork, and Naor [37]. The indistinguishability (IND) formalizes that an adversary, given a ciphertext c, is not able to learn any information about the plaintext. The non-malleability (NM) formalizes that an adversary's inability, given a challenge ciphertext c, to output a different ciphertext c' such that the plaintexts m, m' underlying these two ciphertexts are meaningfully related. (For example, $m' = m + 1$.) It captures a sense in which ciphertexts can be tamper-proof. On the other hand, popular formalizations of the attack models are the chosen plaintext attack (CPA) and the adaptive chosen ciphertext attack (CCA). Under the CPA setting, the adversary can obtain ciphertexts of plaintexts of her

choice. In the public-key setting, giving the adversary the public key suffices to capture this attack. Under the CCA setting [73], the adversary gets (in addition to the public key) access to an oracle for the decryption function. The only restriction is that the adversary cannot ask c to the decryption oracle. (The attack is called adaptive because queries to the decryption oracle can depend on the challenge c.) From the above argument, we can consider four security notions, IND-CPA, IND-CCA, NM-CPA, NM-CCA. Bellare, Desai, Pointcheval, and Rogaway [4] discussed the relationships between these security notions. The widely admitted appropriate security level for public-key encryption is the indistinguishability against the adaptive chosen ciphertext attack (IND-CCA). There are many public-key encryption schemes proved IND-CCA2, such as [26, 7], etc.

Bellare, Boldyreva, Desai, and Pointcheval [3] proposed a new security requirement of the encryption schemes called "key-privacy" or "anonymity." It asks that the encryption provide (in addition to privacy of the data being encrypted) privacy of the key under which the encryption was performed. That is, the receiver is anonymous from the point of view of the adversary.

Anonymous encryption schemes have many applications. For example, anonymous encryption schemes have arisen in the context of mobile communications. If a mobile user uses an anonymous encryption scheme, he can keep his identity private from an eavesdropping adversary. A particular case of this is anonymous authenticated key exchange protocol such as SKEME (Krawczyk [57]). The encryption scheme in SKEME must have the anonymity property. Anonymous credential system (Camenisch and Lysyanskaya [15]) enables users to control the dissemination of information about themselves. It is required to be infeasible to correlate transactions carried out by the same user. They use a verifiable circular encryption scheme that needs to have the anonymity property. Sako [77] proposed an auction protocol. They express a bid as an encryption of a fixed message, with the key to encrypt it corresponding to the value of the bid. In their scheme, if the encryption scheme has the anonymity property, the value of the bid is protected from the other bidders.

A simple observation that seems to be folklore is that standard RSA encryption, namely, a ciphertext is $x^e \bmod N$ where x is a plaintext and (N, e) is a public key, does not provide anonymity, even when all moduli in the system have the same length. Suppose an adversary knows that the ciphertext y is created under one of two keys (N_0, e_0) or (N_1, e_1), and suppose $N_0 \leq N_1$. If $y \geq N_0$ then the adversary bets it was created under (N_1, e_1), else the adversary bets it was created under (N_0, e_0). It is not hard to see that this attack has non-negligible advantage. To construct the schemes with anonymity, it is necessary that the space of ciphertexts is common to each user.

In [3], Bellare, Boldyreva, Desai, Pointcheval provided the key-privacy encryption

scheme, RSA-RAEP, which is a variant of RSA-OAEP (Bellare and Rogaway [7], Fujisaki, Okamoto, Pointcheval, and Stern [43]), and made the space of ciphertexts common to each user by repeating the evaluation of the RSA-OAEP permutation $f(x, r)$ with plaintext x and random r, each time using different r until the value is in the safe range (See Section 3.2.). For deriving a value in the safe range, the number of the repetition would be very large (the value of the security parameter). In fact, their algorithm can fail to give a desired output with some (small) probability.

Undeniable and Confirmer Signature Digital signature is an important tool for realizing security in open distributed systems and in electronic commerce as they guarantee the authenticity of data. In the common model, a digital signature can be verified by everyone (universal verifiability) and therefore its validity cannot be denied by the signer (non-repudiation). However, the universal verifiability property of digital signatures is not always a desirable property. For example, consider the situation that a signature contains some confidential agreement or private or personal information. In these case, limiting the ability of third parties to verify the validity of signatures is an important goal.

Undeniable signature proposed by Chaum and Antwerpen [22, 20] is a solution to this problem. Undeniable signature scheme is non-self-authenticating signature scheme, that is, the signatures can only be verified by conducting a confirmation protocol with the signer, assuming the signer participates. However, if a signature is only verifiable with the aid of a signer, a dishonest signer may refuse to authenticate a genuine document. Undeniable signature solves this problem by adding a new component called the denial (disavowal) protocol in addition to the normal components of signature and verification. Chaum also provided confirmer signature [21] which is undeniable signature where signatures may also be verified by interacting with an entity called the confirmer who has been designated by the signer, and many undeniable and confirmer signature schemes have been proposed [47, 63, 16, 45]. The standard security requirements for undeniable and confirmer signature are the unforgeability of signatures, and the correctness and soundness of the confirmation and denial protocols.

In 2003, Galbraith and Mao proposed a new security notion of undeniable and confirmer signature named "anonymity" in [44]. Informally, this security property is as follows. Imagine a system with n users and suppose an adversary is given a valid message-signature pair and is asked to determine which user generated the signature. By running signature confirmation or denial protocols with a given user (or their designated confirmer) one can determine whether or not the user generated the signature. An undeniable or confirmer signature scheme has the anonymity property if it is infeasible to determine whether a user is or is not the signer of the message without interacting with the user or with the

$n-1$ other users, with a given message-signature pair. In [44], Galbraith and Mao pointed out that the RSA-based undeniable and confirmer signature scheme proposed by Gennaro, Krawczyk and Rabin [47] does not satisfy the anonymity property, and provided a new undeniable and confirmer signature scheme with anonymity. Since their scheme is based on the RSA function, it is necessary that the space of signatures is common to each user, similar to the case of public-key encryption schemes with anonymity. They made the space of signatures common to each user by encoding the message to an N-ary representation and applying the standard RSA permutation to the low-order digits where N is a public key for each user (See Section 4.2.). This technique was originally proposed by Desmedt [35].

Ring Signature The general notion of group signature was introduced by Chaum and van Heyst [23]. In such a scheme, a trusted group manager predefines certain groups of users and distributes specially designed keys to their members. Individual members can then use these keys to anonymously sign messages on behalf of their group. That is, the receiver of the signature can verify that it is a valid signature of the group, but cannot find which member of the group produced the signature (anonymity). Though the signatures produced by different group members look indistinguishable to the verifiers, not to the group manager who can revoke the anonymity of misbehaving signers.

In 2001, Rivest, Shamir, and Tauman [76] proposed the notion of ring signature schemes. These are simplified group signature schemes which have only users and no managers. Unlike group signature, ring signature has no group managers, no setup procedures, no revocation procedures, and no coordination. The signer does not need the knowledge, consent, or assistance of the other ring members to put them in the ring. All the signer needs is knowledge of their regular public keys. To produce a ring signature, the actual signer declares an arbitrary set of possible signers that includes himself, and computes the signature entirely by himself using only his secret key and the other's public keys. Then, the receiver of the signature can verify that it is a valid signature of the ad hoc group, but cannot find which member of the group produced the signature. Here, since the group manager does not exist, no one revoke the anonymity of the actual signer (unless he decides to expose himself).

They also proposed the efficient schemes based on RSA and Rabin. In their RSA-based scheme, the trap-door RSA permutations of the various ring members will have domains of different sizes. This makes it awkward to combine the individual signatures, so one should construct some trap-door one-way permutation which has a common domain for each user. Intuitively, in the ring signature scheme, Rivest, Shamir, and Tauman solved this problem by encoding the message to an N_i-ary representation and applying an standard RSA permutation f to the low-order digits where N_i is a public key for each user (See

Section 5.2.). This is the same kind of the technique employed in the undeniable and confirmer signature by Galbraith and Mao. As mentioned in [76], for deriving a secure permutation g with a common domain, the domain of g would be 160 bits larger than that of f.

1.1.2 Our Contribution on Techniques

From the previous results mentioned above, we can find two techniques, repeating, expanding for anonymity of cryptosystems based on RSA.

Repeating Repeating the evaluation of the encryption (respectively the signing) with plaintext x (resp. message m), random r, and the RSA function, each time using different r until the value is smaller than any public key N of each user.
In [3], Bellare, Boldyreva, Desai, and Pointcheval used this technique for the encryption scheme.

Expanding Doing the evaluation of the encryption (respectively the signing) with plaintext x (resp. message m), random r, and the RSA function, and expanding it to the common domain.
This technique was proposed by Desmedt [35]. In [44], Galbraith and Mao used this technique for the undeniable signature scheme. In [76], Rivest, Shamir, and Tauman also used this technique for the ring signature scheme.

In this thesis, we propose two new techniques for obtaining the anonymity property of RSA-based cryptosystems.

An RSA Family of Trap-Door Permutation with a Common Domain We first consider an underlying primitive element common to the key-privacy encryption and the ring signature schemes, that is, families of trap-door permutations with a common domain. As we have seen before, for a standard RSA family of trap-door permutations denoted by RSA, even if all of the functions in a family use RSA moduli of the same size (the same number of bits), it will have domains with different sizes. In this thesis, we construct an RSA family of trap-door permutations with a common domain denoted by RSACD. The domain and range of RSACD are common to each user when each user has an RSA modulus of the same size. We also prove the properties of RSACD, that is, we show that the θ-partial one-wayness (Roughly speaking, given a function f and an element $y = f(x)$, it is hard to compute a θ fraction of the most significant bits of x.) of RSACD is equivalent to the one-wayness of RSACD for $\theta > 0.5$, and that the one-wayness of RSACD is equivalent to the one-wayness of RSA. Fujisaki, Okamoto, Pointcheval, and Stern [43] showed that the θ-partial one-wayness of RSA is equivalent to the one-wayness of RSA for $\theta > 0.5$. Thus,

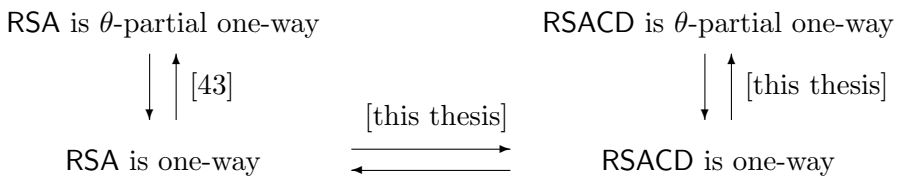

Figure 1.1: Relationships between RSA and RSACD for $\theta > 0.5$.

the relations in Figure 1.1 are satisfied for $\theta > 0.5$. From these relations, we have that the θ-partial one-wayness of RSACD is equivalent to the one-wayness of RSA. This property is useful to construct the public-key encryption scheme with anonymity.

By using the RSACD function, we propose a new technique for obtaining the anonymity property.

RSACD Doing the evaluation of the encryption (respectively the signing) with plaintext x (resp. message m), random r, and the RSACD function.

The Sampling Twice Technique We next propose a new technique for obtaining the anonymity property of RSA-based cryptosystems, called "sampling twice." We propose an algorithm `ChooseAndShift` as follows. It takes two numbers $x_1, x_2 \in \mathbb{Z}_N$ as input and returns a value $y \in [0, 2^k)$ where $|N| = k$, and if x_1 and x_2 are independently and uniformly chosen from $\mathbb{Z}_N$ then y is uniformly distributed over $[0, 2^k)$.

$$
\begin{array}{l}
\texttt{Algorithm ChooseAndShift}_{N,k}(x_1, x_2) \\
\quad \texttt{if } (0 \le x_1, x_2 < 2^k - N) \\
\qquad \texttt{return} \begin{cases} x_1 & \text{with probability } \frac{1}{2} \\ x_1 + N & \text{with probability } \frac{1}{2} \end{cases} \\
\quad \texttt{elseif } (2^k - N \le x_1, x_2 < N) \\
\qquad \texttt{return } x_1 \\
\quad \texttt{else} \\
\qquad y_1 \leftarrow \min\{x_1, x_2\}; \quad y_2 \leftarrow \max\{x_1, x_2\} \\
\qquad \%\%\% \textit{ Note that } 0 \le y_1 < 2^k - N \textit{ and } 2^k - N \le y_2 < N. \ \%\%\% \\
\qquad \texttt{return} \begin{cases} y_1 & \text{with probability } (\frac{1}{2} + \frac{N}{2^{k+1}}) \times \frac{1}{2} \\ y_1 + N & \text{with probability } (\frac{1}{2} + \frac{N}{2^{k+1}}) \times \frac{1}{2} \\ y_2 & \text{with probability } \frac{1}{2} - \frac{N}{2^{k+1}} \end{cases}
\end{array}
$$

By using the algorithm `ChooseAndShift`, we propose the sampling twice technique.

Sampling Twice Doing the evaluation of the encryption (respectively the signing) twice with plaintext x (resp. message m), random r_1 and r_2, and the RSA function, and

	Repeating	Expanding	RSACD	Sampling Twice
Encryption	Bellare et al.	**this thesis**	**this thesis**	**this thesis**
Undeniable and Confirmer Signature	**this thesis**	Galbraith et al.	-	**this thesis**
Ring Signature	**this thesis**	Rivest et al.	**this thesis**	**this thesis**

Figure 1.2: The previous and our proposed schemes.

applying our proposed algorithm `ChooseAndShift` for the two resulting values.

1.1.3 Our Contribution on Schemes

We then propose the schemes for encryption, undeniable and confirmer signature, and ring signature, by applying our proposed techniques. More precisely, we propose the schemes for encryption and ring signature with the RSACD function, and those for encryption, undeniable and confirmer signature, and ring signature with the sampling twice technique. We also prove the anonymity property and other required security of the schemes. Unfortunately, we have not succeeded to construct the undeniable and confirmer signature scheme with anonymity by applying the RSACD function.

Furthermore, we present the previously unproposed schemes with the anonymity property by applying the repeating and expanding techniques. We also prove the anonymity property and other required security of the schemes (See Figure 1.2.).

We summarize the (dis)advantage of the schemes with four techniques.

The scheme with repeating is efficient with respect to the sizes of ciphertexts and signatures, the computational costs to encrypt messages and to sign messages in the average case, and those to decrypt ciphertexts and to verify signatures. However, it is inefficient with respect to the computational costs to encrypt messages and to sign messages in the worst case. In order to obtain the anonymity property, it is necessary for each user to choose a public key with almost the same size.

The scheme with expanding provides anonymity even if each user uses the public key of different length. It is efficient with respect to the computational costs to encrypt messages, to sign messages, to decrypt ciphertexts, and to verify signatures. However, the sizes of ciphertexts and signatures are larger than those of the other schemes.

The scheme with RSACD is efficient with respect to the sizes of ciphertexts and signatures, and the computational costs to encrypt messages and to sign messages. However, it is inefficient with respect to the computational costs to decrypt a ciphertext and to verify a signature. In order to obtain the anonymity property, it is necessary for each user to choose a public key with exact the same size.

The scheme with sampling twice is efficient with respect to the sizes of ciphertexts and

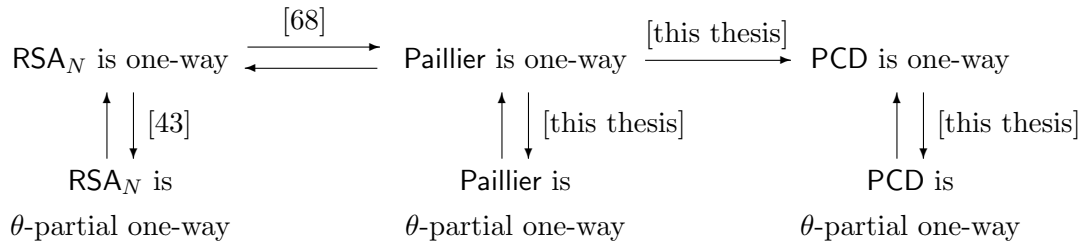

Figure 1.3: Relationships between RSA$_N$, Paillier, and PCD for $\theta > 0.5$.

signatures, the computational costs to decrypt ciphertexts and to verify signatures, and the computational costs to encrypt messages and to sign messages in the worst case. However, the number of exponentiations for encryption or signing is two, while that of the other schemes is one or 1.5 in the average case. Similar to the scheme with RSACD, in order to obtain the anonymity property, it is necessary for each user to choose a public key with exact the same size.

1.2 A Family of Paillier's Trap-Door Permutations and its Applications to Public-Key Encryption with Anonymity

Background. In [68], Paillier provided a public-key encryption scheme based on the problem of computing high-degree residuosity classes modulo N^2 where N is a typical RSA modulus. His encryption scheme is based on the permutation $(m, r) \mapsto g^m r^N \bmod N^2$. Paillier proved that his encryption scheme is IND-CPA if and only if the Decisional Composite Residuosity Problem (given $w \in \mathbb{Z}^*_{N^2}$, to decide whether w is an N-th residue modulo N^2 or not) is hard, and that the Decisional Composite Residuosity Problem is hard if the RSA problem is hard. Paillier also provided a trap-door one-way bijective function, and proved that the function is one-way if and only if the problem of extracting N-th roots modulo N is hard.

Our Contribution. In this thesis, we focus on the four techniques described above in the case using the Paillier's bijective function instead of the RSA function. We slightly modify his function and construct a family of Paillier's trap-door permutations denoted by Paillier. We also construct a family of Paillier's trap-door permutations with a common domain denoted by PCD, and prove the relations in Figure 1.3 for $\theta > 0.5$. Here, RSA$_N$ denotes an RSA family of trap-door permutations with the fixed exponent N.

We prove that the one-wayness of Paillier is reduced to that of PCD. The proof is similar to that for RSA and RSACD. On the other hand, we cannot prove the partial one-wayness of Paillier by directly applying a similar argument for that of RSA in [43]. Furthermore, although the construction of PCD is similar to that of RSACD, we cannot prove the partial

one-wayness of PCD by directly applying a similar argument for that of RSACD.

We also apply Paillier and PCD to encryption, and obtain Paillier-OAEP (OAEP with Paillier's trap-door permutation) with repeating, that with expanding, that with sampling twice, and PCD-OAEP (OAEP with Paillier's trap-door permutation with a common domain). We prove that the anonymity and the indistinguishability of Paillier-OAEP with repeating, that with expanding, and that with sampling twice can be reduced directly to the θ-partial one-wayness of Paillier. We also prove that the anonymity and the indistinguishability of PCD-OAEP is reduced directly to the θ-partial one-wayness of PCD. From the relations in Figure 1.3, our proposed schemes provide the anonymity and the indistinguishability assuming that RSA_N is one-way.

1.3 Relationships between Data-Privacy and Key-Privacy

Background. We have considered two kinds of security notions, data-privacy and key-privacy. Popular formalizations for data-privacy, such as indistinguishability (IND) under either the chosen plaintext attack (CPA) or the adaptive chosen ciphertext attack (CCA), is directed at capturing various data-privacy requirements. On the other hand, the security notions for key-privacy, such as indistinguishability of keys (IK) under either the chosen plaintext attack or the adaptive chosen ciphertext attack, asks that an encryption scheme provides privacy of the key under which the encryption was performed.

On the data-privacy and key-privacy, Halevi [49] provided a simple sufficient condition for an public-key encryption scheme which meets IND to meet IK. It is, roughly speaking, for any two public-key pk_0, pk_1, the distribution of ciphertexts of random messages under the key pk_0 and that under the key pk_1 are statistically close. In [2], Abdalla et. al. extended the Halevi's condition to identity-based encryption. They weakened the statistical (i.e. information-theoretic) requirement of [49] to a computational one. We call the computational version of the Halevi's condition for public-key encryption schemes the anonymity with random messages (IKR).

Our Contribution. We revisit the definition of key-privacy by Bellare, Boldyreva, Desai, and Pointcheval [3]. In the experiment of the definition by [3], the adversary chooses only one message $m \in \mathtt{MSPC}(pk_0) \cap \mathtt{MSPC}(pk_1)$ and receives a ciphertext of m encrypted with one of two keys pk_0 and pk_1. Then the adversary tries to determine under which key the encryption was performed. Therefore, their definition guarantees the anonymity property only when the message is chosen from the set $\mathtt{MSPC}(pk_0) \cap \mathtt{MSPC}(pk_1)$.

However, in some public-key encryption schemes, the ciphertext space may be common even if the message spaces for each public-key are different, and such schemes may provide the anonymity property.

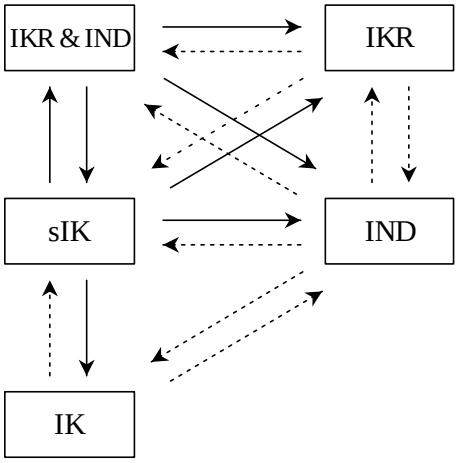

Figure 1.4: Relationships between data-privacy and key-privacy.

In this thesis, to consider this situation, we propose a new security notion for public-key encryption, called "strong anonymity." In the experiment of our definition, the adversary chooses two messages m_0 and m_1 where m_0 and m_1 are in the message spaces for pk_0 and pk_1, respectively, and receives either a ciphertext of m_0 encrypted with pk_0 or a ciphertext of m_1 encrypted with pk_1. Thus, our security notion captures the situation described above.

We then show the relationships between data-privacy and key-privacy. We consider the indistinguishability (IND) as the security notion for the data-privacy, and the anonymity (IK), the anonymity with random messages (IKR), and the strong anonymity (sIK) as those for the key-privacy.

We show the relationships between data-privacy and key-privacy in Figure 1.4. These relations hold under the chosen message attack and the adaptive chosen ciphertext attack. In this figure, for notions of security A and B,

- "$A \longrightarrow B$" means that A implies B, that is, for any public-key encryption scheme which is secure in the sense of A is also secure in the sense of B (We denote it as $A \Rightarrow B$.), and
- "$A \dashrightarrow B$" means that A does *not* imply B, that is, there exists a public-key encryption scheme which is secure in the sense of A and *not* secure in the sense of B (We denote it as $A \not\Rightarrow B$.).

In this thesis, we prove the relations in Figure 1.5. In this figure, the number on the arrow refers to the section of this thesis. By using the relations in Figure 1.5 and trivial relations (IKR-atk $\wedge$ IND-atk $\Rightarrow$ IKR-atk, IKR-atk $\wedge$ IND-atk $\Rightarrow$ IND-atk), the relations which are in Figure 1.4 and not in Figure 1.5 are determined automatically.

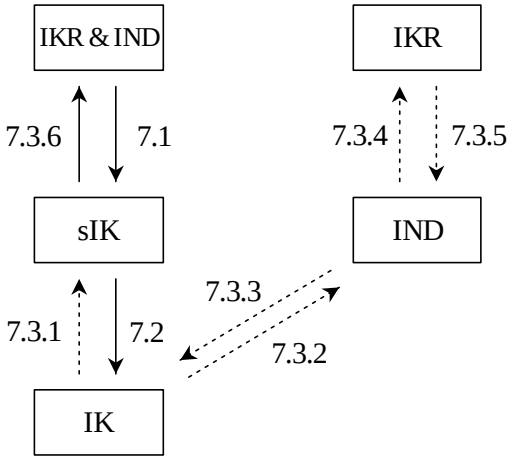

Figure 1.5: Relationships proved in this thesis.

From this figure, we can see that sIK is very strong security notion. For example, IK does not imply IND, but sIK implies IND. Furthermore, we can also see that sIK implies IKR. Therefore, sIK is equivalent to IKR $\wedge$ IND, while IK is weaker than IKR $\wedge$ IND.

1.4 Plaintext Awareness in the Two-Key Setting and a Generic Conversion for Encryption with Anonymity

Background. As mentioned before, the classical security requirement of public-key encryption schemes is that it provides privacy of the encrypted data. The widely admitted appropriate security level for public-key encryption is the indistinguishability against the adaptive chosen ciphertext attack (IND-CCA). A promising way to construct such a public-key encryption scheme is to convert it from primitives which are secure in a weaker sense such as one-wayness (OW), IND-CPA, etc.

Bellare and Rogaway [7] proposed a generic and simple conversion scheme from a one-way trapdoor permutation into a public-key encryption scheme. The scheme created in this way is called OAEP. Fujisaki, Okamoto, Pointcheval, and Stern [43] proved that OAEP with a partial one-way trapdoor permutation is secure in the sense of IND-CCA. The OAEP conversion has several variants, such as SAEP [9], OAEP+ [78], etc.

Fujisaki and Okamoto [42] proposed a simple conversion scheme from weak public-key and symmetric-key encryption schemes into a public-key encryption scheme which is secure in the sense of IND-CCA. This scheme was used to construct the identity-based encryption scheme proposed by Boneh and Franklin [11]. Pointcheval [70] proposed a similar conversion scheme.

Recently, many conversion schemes which depend on gap problems [67], such as, REACT [66], GEM [25], and the schemes in [28], are proposed.

The public-key encryption schemes derived from the conversion schemes [7, 43, 9, 78, 42, 70, 66, 25, 28] described above meet not only IND-CCA, but also the notion of plaintext awareness (PA). The notion of PA is first proposed by Bellare and Rogaway [7] and refined by Bellare, Desai, Pointcheval, and Rogaway [4] which is, roughly speaking, that nobody can produce a *new* ciphertext without knowing the plaintext. We say that a public-key encryption scheme is secure in the sense of PA if it is secure in the sense of IND-CPA and there exists a knowledge extractor which is a formalization of the above property. In [4], they proved that PA implies IND-CCA. Since it looks much easier to prove that a public-key encryption scheme is secure in the sense of PA than to prove directly it is secure in the sense of IND-CCA, the notion of PA is useful to prove the security of public-key encryption schemes.

Recently, Bellare and Palacio [5] discussed the problem of defining the notion of plaintext-awareness without random oracles and of achieving its concrete schemes.

On the other hand, the notion of PA might be too strong. The schemes described above get a redundant construction. In [69, 29], the conversion schemes without redundancy were proposed. They are secure in the sense of IND-CCA, but does not meet PA. Fujisaki [41] introduced another security notion, called plaintext simulatability (PS). It implies IND-CCA, similar to PA, however, it is a properly weaker notion than PA.

Our Contribution. In this thesis, we propose the notion of plaintext awareness in the two-key setting, called PATK. We say that the public-key encryption scheme Π is secure in the sense of PATK if Π is secure in the sense of IK-CPA and there exists a knowledge extractor for PATK. There are some differences between the definition of a knowledge extractor for PA in [4] and that for PATK. We can see that if there exists a knowledge extractor K for PATK of Π, then we can use K as a knowledge extractor for PA of Π. That is, if the public-key encryption scheme Π is secure in the sense of PATK and IND-CPA, then Π is secure in the sense of PA. However, it is not clear that we can use the knowledge extractor for PA of Π as that for PATK of Π.

We also prove that if a public-key encryption scheme is secure in the sense of PATK, then it is also secure in the sense of IK-CCA. Since it looks much easier to prove that a public-key encryption scheme is secure in the sense of PATK than to prove directly that it is secure in the sense of IK-CCA, the notion of PATK is useful to prove the anonymity property of public-key encryption schemes.

We also propose the first generic conversion scheme for the anonymity from IK-CPA to IK-CCA. We employ the Fujisaki-Okamoto conversion scheme [42]. The public-key

encryption scheme derived from their conversion scheme is secure in the sense of IND-CCA in the random oracle model when it consists of a public-key encryption scheme Π^{pub} and a symmetric-key encryption scheme Π^{sym} where

- Π^{pub} is γ-uniform ($\gamma < 1$) and secure in the sense of OW, and
- Π^{sym} is secure in the sense of find-guess (FG).

We prove that the scheme derived from the Fujisaki-Okamoto conversion scheme with the above two and the following two assumptions is secure in the sense of IK-CCA in the random oracle model.

- In Π^{pub}, the message space and the randomness space are common to each user (each public-key).
- Π^{pub} is secure in the sense of IK-CPA.

We can get the public-key encryption scheme which is secure in the sense of IND-CCA and IK-CCA if we assume the above four conditions.

1.5 Universally Anonymizable Public-Key Encryption

Background. Consider the following situation. In order to send e-mails, all members of the company use the encryption scheme which does not provide the anonymity property. They consider that e-mails sent to the inside of the company do not have to be anonymized and it is sufficient to be encrypted the data. However, when e-mails are sent to the outside of the company, they want to anonymize them for preventing the eavesdropper on the public network.

A trivial answer for this problem is that all members use the encryption scheme with the anonymity property. However, generally speaking, we require some computational costs to create ciphertexts with the anonymity property. In fact, the RSA-based anonymous encryption schemes proposed in [3] and in this thesis, which are based on RSA-OAEP, are not efficient with respect to the encryption cost or the size of ciphertexts, compared with RSA-OAEP (See Figure 1.6. Here, k, k_0, k_1 are security parameters and we assume that N is uniformly distributed in $(2^{k-1}, 2^k)$.). Since the members do not require to anonymize the e-mails, it would be better to use the standard encryption scheme within the company.

Our Contribution. We propose another way to solve this. Consider the situation that not only the person who made the ciphertexts, but also anyone can transform the encrypted data to those with the anonymity property without decrypting these encrypted data. If we have this situation, we can make an e-mail gateway which can transform encrypted e-mails

	RSA-OAEP	Sampling Twice	RSA-RAEP [3]	RSACD	Expanding
anonymity	No	Yes	Yes	Yes	Yes
# of mod. exp. to encrypt (average / worst)	1 / 1	2 / 2	1.5 / k_1	1.5 / 2	1 / 1
# of random bits to encrypt (average / worst)	k_0	$2k_0 + k + 3$ / $2k_0 + k + 3$	$1.5k_0$ / $k_1 k_0$	$1.5k_0$ / $1.5k_0$	$k_0 + 160$ / $k_0 + 160$
size of ciphertexts	k	k	k	k	$k + 160$

Figure 1.6: The costs of the encryption schemes.

to those with the anonymity property without using the corresponding secret key when they are sent to the outside of the company.

Furthermore, we can use this e-mail gateway in order to guarantee the anonymity property for e-mails sent to the outside of the company. The president of the company may consider that all e-mails sent to the outside of the company should be anonymized. In this case, even if someone tries to send e-mails to the outside of the company without anonymization, the e-mails passing through the e-mail gateway are always anonymized.

In this thesis, in order to formalize this idea, we propose a special type of public-key encryption scheme called a *universal anonymizable public-key encryption scheme*. A universal anonymizable public-key encryption scheme consists of a standard public-key encryption scheme $\mathcal{PE}$ and two additional algorithms, that is, an anonymizing algorithm $\mathcal{UA}$ and a decryption algorithm $\mathcal{DA}$ for anonymized ciphertexts. We can use $\mathcal{PE}$ as a standard encryption scheme which is not necessary to have the anonymity property. Furthermore, in this scheme, by using the anonymizing algorithm $\mathcal{UA}$, anyone who has a standard ciphertext can anonymize it with its public key whenever she wants to do that. The receiver can decrypt the anonymized ciphertext by using the decryption algorithm $\mathcal{DA}$ for anonymized ciphertexts. Then, the adversary cannot know under which key the anonymized ciphertext was created.

To formalize the security properties for universal anonymizable public-key encryption, we define three requirements, the key-privacy, the data-privacy on standard ciphertexts, and that on anonymized ciphertexts.

We then propose the universal anonymizable public-key encryption schemes based on the ElGamal encryption scheme, the Cramer-Shoup encryption scheme, and RSA-OAEP, and prove their security.

We show the key-privacy property of our schemes by applying an argument in [3] with modification. Though Bellare, Boldyreva, Desai, and Pointcheval [3] proved that the ElGamal and the Cramer-Shoup encryption schemes provide the anonymity property when all of the users use a common group, the argument in [3] for these schemes depends heavily on the situation where all of the users employ a common group. However, in our discrete-

log based schemes, we do not use the common group for obtaining the key-privacy property. Therefore, we cannot straightforwardly apply their argument to our schemes. To prove the key-privacy property of our schemes, we employ the idea described in [26] by Cramer and Shoup, where we encode the elements of QR_p (a group of quadratic residues modulo p) where $p = 2q + 1$ and p, q are prime to those of $\mathbb{Z}_q$. This encoding plays an important role in our schemes. We also employ the expanding technique. With this technique, if we get the ciphertext, we expand it to the common domain. This technique was proposed by Desmedt [35]. In [44], Galbraith and Mao used this technique for the undeniable signature scheme. In [76], Rivest, Shamir, and Tauman also used this technique for the ring signature scheme.

1.6 Organization

The organization of this thesis is as follows.

In Chapter 2, after reviewing the repeating and the expanding techniques for obtaining the schemes for public-key encryption and signature with anonymity, we construct an RSA family of trap-door permutations with a common domain, and show its property. We also construct the algorithm `ChooseAndShift` and propose the sampling twice technique. By applying our proposed techniques, we propose the schemes for public-key encryption in Chapter 3, those for undeniable and confirmer signature in Chapter 4, and those for ring signature in Chapter 5. In Chapters 3 to 5, we also propose the previously unproposed schemes with anonymity by applying the repeating and expanding techniques.

In Chapter 6, We construct a family of Paillier's trap-door permutations and that with a common domain. We also propose the schemes for public-key encryption with our proposed families of trap-door permutations. In Chapter 7, we propose a new security notion called "strong anonymity," and show the relationships between the data-privacy and the key-privacy for public-key encryption schemes. In Chapter 8 we propose the new security notion of plaintext awareness in the two-key setting, called PATK, and show that PATK implies IK-CCA. We also propose the first generic conversion scheme for the anonymity from IK-CPA to IK-CCA. In Chapter 9, we formalize a special type of public-key encryption scheme called a universally anonymizable public-key encryption scheme. We also propose the universally anonymizable public-key encryption schemes based on the ElGamal encryption scheme, the Cramer-Shoup encryption scheme, and RSA-OAEP, and prove their security. We conclude in Chapter 10.

CHAPTER 2

Techniques for Anonymity

In this chapter, we discuss the techniques for obtaining the schemes for public-key encryption and signature with anonymity. We review the repeating and expanding techniques for obtaining the schemes for public-key encryption and signature with anonymity. We then construct an RSA family of trap-door permutations with a common domain, and show its property. We also construct the algorithm `ChooseAndShift` and propose the sampling twice technique.

The organization of this chapter is as follows. In Section 2.1, we review the repeating technique which is used in the public-key encryption scheme by Bellare, Boldyreva, Desai, and Pointcheval [3]. In Section 2.2, we review the expanding technique which is used in the public-key encryption scheme by Bellare, Boldyreva, Desai, and Pointcheval [3]. In Section 2.3, we construct an RSA family of trap-door permutations with a common domain, and show its property. In Section 2.4, we construct the algorithm `ChooseAndShift` and propose the sampling twice technique.

2.1 The Repeating Technique

In this section, we review the repeating technique.

Repeating Repeating the evaluation of the encryption (respectively the signing) with plaintext x (resp. message m), random r, and the RSA function, each time using

different r until the value is smaller than any public key N of each user.

In [3], Bellare, Boldyreva, Desai, and Pointcheval used this technique for their public-key encryption scheme.

For example, suppose that each user uses an RSA modulus of length k. We set the ciphertext space (or the signature space) to $\{0,1\}^{k-1}$. Then, for any N where $|N| = k$, if the ciphertext is uniformly distributed over $\mathbb{Z}_N^*$, then the distribution of the outputs by the repeating technique is almost the same as the uniform distribution over $\{0,1\}^{k-1}$.

2.2 The Expanding Technique

In this section, we review the expanding technique.

Expanding Doing the evaluation of the encryption (respectively the signing) with plaintext x (resp. message m), random r, and the RSA function, and expanding it to the common domain.

This technique was proposed by Desmedt [35]. In [44], Galbraith and Mao used this technique for the undeniable signature scheme. In [76], Rivest, Shamir, and Tauman also used this technique for the ring signature scheme.

For example, suppose that each user uses an RSA modulus of length k. Then, we set the ciphertext space (or the signature space) to $\{0,1\}^{k+k_b}$. In the expanding technique, we expand the ciphertext (or the signature) $c \in \mathbb{Z}_N^*$ to the common domain $\{0,1\}^{k+k_b}$. In particular, we choose $t \xleftarrow{R} \{0,1,2,\ldots,\lfloor(2^{k+k_b}-c)/N\rfloor\}$ and set $c' \leftarrow c+tN$. Then, for any N where $|N| = k$, if c is uniformly distributed over $\mathbb{Z}_N$, then the distribution of the outputs by the expanding technique is statistically indistinguishable from the uniform distribution over $\{0,1\}^{k+k_b}$, where the statistically distance is less than $1/2^{k_b-1}$. Therefore, for any N where $|N| = k$, if k_b is sufficiently large and c is uniformly distributed over $\mathbb{Z}_N^*$, then the distribution of the outputs by the expanding technique is almost the same as the uniform distribution over $\{0,1\}^{k+k_b}$.

2.3 An RSA family of Trap-Door Permutation with a Common Domain

In this section, we propose an RSA family of trap-door permutations with a common domain denoted by RSACD, and prove that the θ-partial one-wayness of RSACD is equivalent to the one-wayness of RSACD for $\theta > 0.5$, and that the one-wayness of RSACD is equivalent to the one-wayness of RSA.

2.3.1 Prelimilaries

In this section, we review the definitions of families of functions, families of trap-door permutations, and θ-partial one-wayness. We also describe the standard RSA family of trap-door permutations denoted by RSA.

Notations In this thesis, we use the following notations. If A is a probabilistic algorithm, then $A(x_1, x_2, \cdots; r)$ is the result of running A on inputs $x_1, x_2, \cdots$ and coins r. We let $y \leftarrow A(x_1, x_2, \cdots)$ denote the experiment of picking r at random and letting y be $A(x_1, x_2, \cdots; r)$. If S is a finite set then $x \xleftarrow{R} S$ is the operation of picking an element uniformly from S. If α is not an algorithm then $x \leftarrow \alpha$ is a simple assignment statement.

We describe the definitions of families of functions and families of trap-door permutations.

Definition 2.1 (families of functions [3])**.** *A family of functions $F = (K, S, E)$ is specified by three algorithms.*

- *The randomized key-generation algorithm K takes as input a security parameter $k \in \mathbb{N}$ and returns a pair (pk, sk) where pk is a public key and sk is an associated secret key. (In cases where the family is not trap-door, the secret key is simply the empty string.)*
- *The randomized sampling algorithm S takes input pk and returns a random point in a set that we call the domain of pk and denote by $\mathrm{Dom}_F(pk)$.*
- *The deterministic evaluation algorithm E takes input pk and a point $x \in \mathrm{Dom}_F(pk)$ and returns an output we denote by $E_{pk}(x)$. We let $\mathrm{Rng}_F(pk) = \{E_{pk}(x) \,|\, x \in \mathrm{Dom}_F(pk)\}$ denote the range of the function $E_{pk}(\cdot)$.*

Definition 2.2 (families of trap-door permutations [3])**.** *We say that F is a family of trap-door functions if there exists a deterministic inversion algorithm I that takes input sk and a point $y \in \mathrm{Rng}_F(pk)$ and returns a point $x \in \mathrm{Dom}_F(pk)$ such that $E_{pk}(x) = y$. We say that F is a family of trap-door permutations if F is a family of trap-door functions, $\mathrm{Dom}_F(pk) = \mathrm{Rng}_F(pk)$, and E_{pk} is a bijection on this set.*

We describe the definition of θ-partial one-way.

Definition 2.3 (θ-partial one-way [3])**.** *Let $F = (K, S, E)$ be a family of functions. Let $k \in \mathbb{N}$ be a security parameter and $b \in \{0, 1\}$. Let $0 < \theta \leq 1$ be a constant. Let A be an*

adversary. Now, we consider the following experiments:

$$
\begin{array}{l}
\texttt{Experiment } \mathbf{Exp}^{\theta\text{-pow-fnc}}_{F,A}(k) \\
\quad (pk, sk) \leftarrow K(k) \\
\quad x \stackrel{R}{\leftarrow} \mathrm{Dom}_F(pk) \\
\quad y \leftarrow E_{pk}(x) \\
\quad x'_1 \leftarrow A(pk, y) \texttt{ where } |x'_1| = \lceil \theta \cdot |x| \rceil \\
\quad \texttt{if } \left(E_{pk}(x'_1||x'_2) = y \texttt{ for some } x'_2\right) \texttt{ return } 1 \texttt{ else return } 0
\end{array}
$$

Here " $||$ " denotes concatenation. We define the advantages of the adversary via

$$\mathbf{Adv}^{\theta\text{-pow-fnc}}_{F,A}(k) = \Pr[\mathbf{Exp}^{\theta\text{-pow-fnc}}_{F,A}(k) = 1]$$

where the probability is taken over K, $x \stackrel{R}{\leftarrow} \mathrm{Dom}_F(pk)$, E, and A. We say that the family F is θ-partial one-way if the function $\mathbf{Adv}^{\theta-\text{pow}-\text{fnc}}_{F,A}(\cdot)$ is negligible in k for any adversary A whose time complexity is polynomial in k.

The "time-complexity" is the worst case execution time of the experiment plus the size of the code of the adversary, in some fixed RAM model of computation. Note that when $\theta = 1$ the notion of θ-partial one-wayness coincides with the standard notion of one-wayness. In the following, we say that the family F is one-way when F is 1-partial one-way.

We describe the standard RSA family of trap-door permutations denoted by RSA.

Definition 2.4 (the standard RSA family of trap-door permutations [3])**.** *The specifications of the standard RSA family of trap-door permutations $\mathsf{RSA} = (K, S, E)$ are as follows. The key generation algorithm takes as input a security parameter k and picks random, distinct primes p, q in the range $2^{\lceil k/2 \rceil - 1} < p, q < 2^{\lceil k/2 \rceil}$ and $2^{k-1} < N < 2^k$. It sets $N = pq$. It picks $e, d \in \mathbb{Z}^*_{\phi(N)}$ such that $ed = 1 \pmod{\phi(N)}$ where $\phi(N) = (p-1)(q-1)$. The public key is N, e, k and the secret key is N, d, k. The sets $\mathrm{Dom}_{\mathsf{RSA}}(N, e, k)$ and $\mathrm{Rng}_{\mathsf{RSA}}(N, e, k)$ are both equal to $\mathbb{Z}^*_N$. The evaluation algorithm $E_{N,e,k}(x) = f^{\mathsf{RSA}}_{N,e,k}(x) = x^e \bmod N$ and the inversion algorithm $I_{N,d,k}(y) = g^{\mathsf{RSA}}_{N,d,k}(y) = y^d \bmod N$. The sampling algorithm returns a random point in $\mathbb{Z}^*_N$.*

Fujisaki, Okamoto, Pointcheval, and Stern [43] showed that the θ-partial one-wayness of RSA is equivalent to the one-wayness of RSA for $\theta > 0.5$.

2.3.2 An RSA Family of Trap-door Permutations with a Common Domain

In this section, we propose the RSA family of trap-door permutations with a common domain and prove that the θ-partial one-wayness of RSACD is equivalent to the one-wayness of RSACD for $\theta > 0.5$, and the one-wayness of RSACD is equivalent to the one-wayness of RSA.

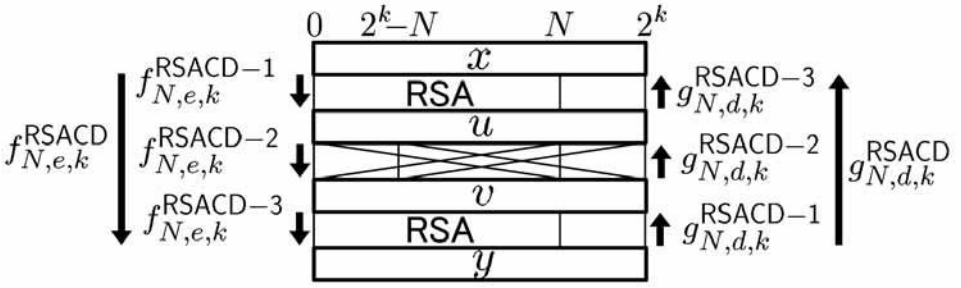

Figure 2.1: Functions $f^{\mathsf{RSACD}}_{N,e,k}$ and $g^{\mathsf{RSACD}}_{N,d,k}$

The Construction of RSACD In this section, we propose the RSA family of trap-door permutations with a common domain denoted by RSACD.

Definition 2.5 (the RSA family of trap-door permutations with a common domain)**.** *The specifications of the RSA family of trap-door permutations with a common domain* $\mathsf{RSACD}= (K, S, E)$ *are as follows. The key generation algorithm is the same as that for* RSA*. The sets* $\mathrm{Dom}_{\mathsf{RSACD}}(N, e, k)$ *and* $\mathrm{Rng}_{\mathsf{RSACD}}(N, e, k)$ *are both equal to* $\{x \mid x \in [0, 2^k) \wedge (x \bmod N) \in \mathbb{Z}^*_N\}$*. The sampling algorithm returns a random point in* $\mathrm{Dom}_{\mathsf{RSACD}}(N, e, k)$*. The evaluation algorithm* $E_{N,e,k}(x) = f^{\mathsf{RSACD}}_{N,e,k}(x)$ *and the inversion algorithm* $I_{N,d,k}(y) = g^{\mathsf{RSACD}}_{N,d,k}(y)$ *are as follows (See Figure 2.1.).*

```
Function f^RSACD_{N,e,k}(x)
  u ← f^RSACD-1_{N,e,k}(x); v ← f^RSACD-2_{N,e,k}(u); y ← f^RSACD-3_{N,e,k}(v)
  return y
```

```
Function f^RSACD-1_{N,e,k}(x)
  if (x < N)
    u ← x^e mod N
  else u ← x
  return u
```

```
Function f^RSACD-2_{N,e,k}(u)
  if (u < 2^k − N) v ← u + N
  elseif (2^k − N ≤ u < N) v ← u
  else v ← u − N
  return v
```

```
Function f^RSACD-3_{N,e,k}(v)
  if (v < N)
    y ← v^e mod N
  else y ← v
  return y
```

```
Function g^RSACD_{N,d,k}(y)
  v ← g^RSACD-1_{N,d,k}(y); u ← g^RSACD-2_{N,d,k}(v); x ← g^RSACD-3_{N,d,k}(u)
  return x
```

```
Function g^RSACD-1_{N,d,k}(y)
  if (y < N)
    v ← y^d mod N
  else v ← y
  return v
```

```
Function g^RSACD-2_{N,d,k}(v)
  if (v < 2^k − N) u ← v + N
  elseif (2^k − N ≤ v < N) u ← v
  else u ← v − N
  return u
```

```
Function g^RSACD-3_{N,d,k}(u)
  if (u < N)
    x ← u^d mod N
  else x ← u
  return x
```

The choice of N from $(2^{k-1}, 2^k)$ ensures that all elements in $\mathrm{Dom}_{\mathsf{RSACD}}(N, e, k)$ are permuted by the standard RSA permutation at least once.

Properties of RSACD In this section, we prove that the θ-partial one-wayness of RSACD is equivalent to the one-wayness of RSACD for $\theta > 0.5$, and that the one-wayness of RSACD is equivalent to that of RSA.

Theorem 2.1. *The θ-partial one-wayness of* RSACD *is equivalent to the one-wayness of* RSACD *for $\theta > 0.5$.*

It is clear that if RSACD is θ-partial one-way then RSACD is one-way. Therefore, we can prove Theorem 2.1 by proving the following lemma.

Lemma 2.1. *If* RSACD *is one-way then* RSACD *is θ-partial one-way for $\theta > 0.5$.*

To prove this lemma, we use the following lemma proved in [43].

Lemma 2.2 ([43])**.** *Consider an equation $\alpha t + u = c \pmod N$ which has solutions t and u smaller than 2^{k_0}. For all values of α, except a fraction $2^{2k_0+6}/N$ of them, (t, u) is unique and can be computed in time $O((\log N)^3)$. (We say "α is a good value" when we can solve the above equation.)*

Proof of Lemma 2.1. Let A be an algorithm that outputs the $k - k_0$ most significant bits of the pre-image of its input $y \in \mathsf{Rng}_{\mathsf{RSACD}}(N, e, k)$ for $2^{k-1} < N < 2^k$ with $k > 2k_0$ (i.e. A is a $((k - k_0)/k)$-partial inverting algorithm for RSACD with $k > 2k_0$), with success probability $\epsilon = \mathbf{Adv}^{\theta-\mathrm{pow}-\mathrm{fnc}}_{\mathsf{RSACD},A}(k)$ where $\theta = (k - k_0)/k > 0.5$, within time bound t. We prove that there exists an algorithm B that outputs a pre-image of y (i.e. B is an inverting algorithm for RSACD) with success probability $\epsilon' = \mathbf{Adv}^{1-\mathrm{pow}-\mathrm{fnc}}_{\mathsf{RSACD},B}(k)$, within time bound t' where

$$\epsilon' \geq \frac{\epsilon^2}{16} \cdot (1 - 2^{2k_0-k+7}), \quad t' \leq 2t + O(k^3).$$

We construct the algorithm B to compute a pre-image of $y \in \mathrm{Rng}_{\mathsf{RSACD}}(N, e, k)$, then we analyze this algorithm and evaluate the success probability and the running time of B.

Algorithm $B((N, e, k), y)$

$\alpha \xleftarrow{R} \mathbb{Z}_N$
pow $\xleftarrow{R} \{1, 2\}$
$y'_{temp} \leftarrow y \cdot \alpha^{e^{\mathrm{pow}}} \bmod N$
$c \xleftarrow{R} \{0, 1\}$
if $(c = 0)$ $y' \leftarrow y'_{temp}$
elseif $(0 \leq y'_{temp} < 2^k - N)$ $y' \leftarrow y'_{temp} + N$
else return fail

} [step 1] set α, pow, and y'

$z \leftarrow A(y)$
$z' \leftarrow A(y')$

} [step 2] run A

find (r, s) **s.t.** $\alpha r - s = (z' - z\alpha) \cdot 2^{k_0} \pmod N$
$x \leftarrow z \cdot 2^{k_0} + r$
return x

} [step 3] compute $g^{\mathsf{RSACD}}_{N,d,k}(y)$

Now, we analyze the advantage of B. For $y \in \mathrm{Rng}_{\mathsf{RSACD}}(N, e, k)$ and $x = g^{\mathsf{RSACD}}_{N,d,k}(y)$, (x, y) satisfies one of the following equations.

$$(1) \quad y = x^e \pmod N$$
$$(2) \quad y = x^{e^2} \pmod N$$

We say $\mathit{type}(y) = 1$ (respectively $\mathit{type}(y) = 2$) if (x, y) satisfies equation 1 (resp. equation 2).

After step 1, if B does not output `fail`, then y' is uniformly distributed over $\mathrm{Rng}_{\mathsf{RSACD}}(N, e, k)$, and for y' and $x' = g^{\mathsf{RSACD}}_{N,d,k}(y')$, (x', y') satisfies one of the following equations.

$$(1') \quad y' = (x')^e \pmod N$$
$$(2') \quad y' = (x')^{e^2} \pmod N$$

We say $\mathit{type}(y') = 1$ (respectively $\mathit{type}(y') = 2$) if (x', y') satisfies equation $1'$ (resp. equation $2'$).

After step 2, if A outputs correctly, namely, z is the $k - k_0$ most significant bits of x and z' is the $k - k_0$ most significant bits of x', then $x = z \cdot 2^{k_0} + r$ and $x' = z' \cdot 2^{k_0} + s$ for some (r, s) where $0 \leq r, s < 2^{k_0}$. Furthermore, if $\mathit{type}(y) = \mathit{type}(y') = \mathrm{pow}$, then $y = x^{e^{\mathrm{pow}}} \pmod N$ and $y' = (x')^{e^{\mathrm{pow}}} \pmod N$. Since $y' = y \cdot \alpha^{e^{\mathrm{pow}}} \pmod N$ and $\gcd(e^{\mathrm{pow}}, N) = 1$, we have $x' = \alpha x \pmod N$. Thus,

$$z' \cdot 2^{k_0} + s = \alpha \cdot (z \cdot 2^{k_0} + r) \pmod N$$
$$\alpha r - s = (z' - z\alpha) \cdot 2^{k_0} \pmod N$$

where $0 \leq r, s < 2^{k_0}$. If α is a good value, algorithm B can solve this equation in step 3 (Lemma 2.2), and outputs $x = z \cdot 2^{k_0} + r$.

Now, we analyze the success probability. We define the following events:

- Fail : B outputs `fail` in step 1,
- GV : α is a good value,
- $\mathsf{Type1}$: $\mathit{type}(y) = \mathit{type}(y') = 1$,
- $\mathsf{Type2}$: $\mathit{type}(y) = \mathit{type}(y') = 2$,
- SucA : $A(y)$ and $A(y')$ are correct.

We have

$$\epsilon = \Pr\left[A(y) \text{ is correct} \wedge \mathit{type}(y) = 1\right] + \Pr[A(y) \text{ is correct} \wedge \mathit{type}(y) = 2]$$

where y is uniformly distributed over $\mathrm{Rng}_{\mathsf{RSACD}}(N, e, k)$. Thus,

$$\Pr[A(y) \text{ is correct} \wedge \mathit{type}(y) = 1] > \frac{\epsilon}{2} \quad \text{or} \quad \Pr[A(y) \text{ is correct} \wedge \mathit{type}(y) = 2] > \frac{\epsilon}{2}.$$

If B does not output `fail` in step 1, then y' is uniformly distributed over $\mathrm{Rng}_{\mathsf{RSACD}}(N, e, k)$. Therefore,

$$\Pr[\mathsf{SucA} \wedge \mathsf{Type1} | \neg\mathsf{Fail}] > \left(\frac{\epsilon}{2}\right)^2 = \frac{\epsilon^2}{4} \quad \text{or} \quad \Pr[\mathsf{SucA} \wedge \mathsf{Type2} | \neg\mathsf{Fail}] > \left(\frac{\epsilon}{2}\right)^2 = \frac{\epsilon^2}{4}.$$

If $A(y)$ and $A(y')$ are correct, $\mathit{type}(y) = \mathit{type}(y') = \mathrm{pow}$, and α is a good value, then B outputs correctly. Since $\Pr[\neg\mathsf{Fail}] > \Pr[c = 1] = 1/2$, $\Pr[\mathrm{pow} = 1] = \Pr[\mathrm{pow} = 2] = 1/2$, and $\Pr[\mathsf{GV}] > 1 - 2^{2k_0-6}/N > 1 - 2^{2k_0-k+7}$ (Lemma 2.2), we have

$$\begin{aligned}
\epsilon' &\geq \Pr[\mathsf{SucA} \wedge \mathit{type}(y) = \mathit{type}(y') = \mathrm{pow} \wedge \mathsf{GV}] \\
&\geq \Pr[\neg\mathsf{Fail}] \times \Pr[\mathsf{GV}] \times \Pr[\mathsf{SucA} \wedge \mathit{type}(y) = \mathit{type}(y') = \mathrm{pow} | \neg\mathsf{Fail}] \\
&\geq \frac{1}{2} \cdot (1 - 2^{2k_0-k+7}) \times (\Pr[\mathsf{SucA} \wedge \mathsf{Type1} \wedge \mathrm{pow} = 1 | \neg\mathsf{Fail}] \\
&\qquad\qquad + \Pr[\mathsf{SucA} \wedge \mathsf{Type2} \wedge \mathrm{pow} = 2 | \neg\mathsf{Fail}]) \\
&= \frac{1}{2} \cdot (1 - 2^{2k_0-k+7}) \times (\Pr[\mathrm{pow} = 1] \times \Pr[\mathsf{SucA} \wedge \mathsf{Type1} | \neg\mathsf{Fail}] \\
&\qquad\qquad + \Pr[\mathrm{pow} = 2] \times \Pr[\mathsf{SucA} \wedge \mathsf{Type2} | \neg\mathsf{Fail}]) \\
&= \frac{1}{4} \cdot (1 - 2^{2k_0-k+7}) \times (\Pr[\mathsf{SucA} \wedge \mathsf{Type1} | \neg\mathsf{Fail}] + \Pr[\mathsf{SucA} \wedge \mathsf{Type2} | \neg\mathsf{Fail}]) \\
&> \frac{\epsilon^2}{16} \cdot (1 - 2^{2k_0-k+7}).
\end{aligned}$$

We estimate the running time of B. B runs A twice. B can solve $\alpha r - s = (z' - z\alpha) \cdot 2^{k_0} \pmod N$ in time $O(k^3)$. Therefore, $t' \leq 2t + O(k^3)$. □

Theorem 2.2. *The one-wayness of* RSACD *is equivalent to the one-wayness of* RSA.

It is easy to see that if RSACD is one-way then RSA is one-way (See Figure 2.1.). Therefore, we can prove Theorem 2.2 by proving the following lemma.

Lemma 2.3. *If* RSA *is one-way then* RSACD *is one-way.*

Proof of Lemma 2.3. We prove that if there exists a polynomial-time inverting algorithm A for RSACD with non-negligible probability $\epsilon = \mathbf{Adv}^{1-\mathrm{pow}-\mathrm{fnc}}_{\mathsf{RSACD},A}(k)$, then there exists a polynomial-time inverting algorithm D for RSA with non-negligible probability $\epsilon' = \mathbf{Adv}^{1-\mathrm{pow}-\mathrm{fnc}}_{\mathsf{RSA},D}(k)$. We specify the algorithm D to compute a pre-image of $Y \in \mathrm{Rng}_{\mathsf{RSA}}(N,e,k)$.

$\mathtt{Algorithm}\ D((N,e,k),Y)$
  $c \overset{R}{\leftarrow} \{0,1\}$
  $\mathtt{if}\ (c=0)$
    $y \leftarrow Y;\ x \leftarrow A((N,e,k),y);\ u \leftarrow f^{\mathsf{RSACD}-1}_{N,e,k}(x);\ v \leftarrow f^{\mathsf{RSACD}-2}_{N,e,k}(u);\ X \leftarrow v$
  $\mathtt{else}$
    $u \leftarrow Y;\ v \leftarrow f^{\mathsf{RSACD}-2}_{N,e,k}(u);\ y \leftarrow f^{\mathsf{RSACD}-3}_{N,e,k}(v);\ x \leftarrow A((N,e,k),y);\ X \leftarrow x$
  $\mathtt{return}\ X$

Now, we analyze the advantage of D. If A outputs correctly then D outputs correctly (See Figure 2.1.). Therefore,

$$\begin{aligned}
\epsilon' &> \Pr[c=0 \ \wedge\ A((N,e,k),Y) \text{ is correct}] \\
&\qquad + \Pr[c=1 \ \wedge\ A((N,e,k),Z) \text{ is correct}] \\
&= \frac{1}{2}\cdot(\Pr[A((N,e,k),Y) \text{ is correct}] + \Pr[A((N,e,k),Z) \text{ is correct}]) \\
&\geq \frac{1}{2}\cdot(\Pr[A((N,e,k),Y) \text{ is correct}] \\
&\qquad + \Pr[A((N,e,k),Z) \text{ is correct} \ \wedge\ N \leq Z < 2^k]).
\end{aligned}$$

where $Z = f^{\mathsf{RSACD}-3}_{N,e,k}(f^{\mathsf{RSACD}-2}_{N,e,k}(Y))$. We have

$$\begin{aligned}
\Pr[A((N,e,k),Y) \text{ is correct}] &= \Pr_1[A((N,e,k),y) \text{ is correct} \,|\, 0 \leq y < N] \\
&> \Pr[A((N,e,k),y) \text{ is correct} \ \wedge\ 0 \leq y < N].
\end{aligned}$$

Furthermore, we have $\Pr[N \leq Z < 2^k] > \Pr[N \leq y < 2^k]$ where Y is uniformly distributed over $\mathbb{Z}^*_N$ and y is uniformly distributed over $\mathrm{Rng}_{\mathsf{RSACD}}(N,e,k)$, since $\Pr[N \leq Z < 2^k] = \Pr[0 \leq Y < 2^k - N]$ and $|\mathbb{Z}^*_N| < |\mathrm{Rng}_{\mathsf{RSACD}}(N,e,k)|$. Since $\Pr[A((N,e,k),Z) \text{ is correct} \,|\, N \leq Z < 2^k] = \Pr[A((N,e,k),y) \text{ is correct} \,|\, N \leq y < 2^k]$, we have

$$\begin{aligned}
&\Pr[A((N,e,k),Z) \text{ is correct} \ \wedge\ N \leq Z < 2^k] \\
&\quad = \Pr[N \leq Z < 2^k] \cdot \Pr[A((N,e,k),Z) \text{ is correct} \,|\, N \leq Z < 2^k] \\
&\quad > \Pr[N \leq y < 2^k] \cdot \Pr[A((N,e,k),Z) \text{ is correct} \,|\, N \leq Z < 2^k] \\
&\quad = \Pr[N \leq y < 2^k] \cdot \Pr[A((N,e,k),y) \text{ is correct} \,|\, N \leq y < 2^k] \\
&\quad > \Pr[A((N,e,k),y) \text{ is correct} \ \wedge\ N \leq y < 2^k].
\end{aligned}$$

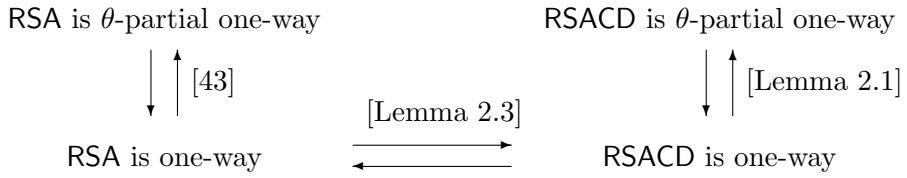

Figure 2.2: Relationships between RSA and RSACD for $\theta > 0.5$.

Therefore,

$$\begin{aligned}\epsilon' &> \frac{1}{2} \cdot (\Pr[A((N,e,k),y) \text{ is correct } \wedge\ 0 \le y < N] \\ &\qquad + \Pr[A((N,e,k),y) \text{ is correct } \wedge\ N \le y < 2^k]) \\ &= \frac{1}{2} \cdot \Pr[A((N,e,k),y) \text{ is correct}] \\ &= \frac{\epsilon}{2}\end{aligned}$$

which is non-negligible in k. □

Hence, we have the relations in Figure 2.2 for $\theta > 0.5$. From these relations, we have that the θ-partial one-wayness of RSACD is equivalent to the one-wayness of RSA. This property is useful to construct the public-key encryption scheme with anonymity.

By using the RSACD function, we propose a new technique for obtaining the anonymity property.

RSACD Doing the evaluation of the encryption (respectively the signing) with plaintext x (resp. message m), random r, and the RSACD function.

In Chapters 3 and 5, by applying the RSACD function we construct the schemes for public-key encryption and ring signature, respectively.

2.4 The Sampling Twice Technique

In this section, we propose a new technique for obtaining the anonymity property of RSA-based cryptosystems. We call this technique "sampling twice." In our technique, we employ an algorithm `ChooseAndShift`. It takes two numbers $x_1, x_2 \in \mathbb{Z}_N$ as input and returns a value $y \in [0, 2^k)$ where $|N| = k$, and if x_1 and x_2 are independently and uniformly chosen from $\mathbb{Z}_N$ then y is uniformly distributed over $[0, 2^k)$.

We describe the algorithm `ChooseAndShift` as follows. It takes two numbers $x_1, x_2 \in$

$\mathbb{Z}_N$ as input and returns a value $y \in [0, 2^k)$ where $|N| = k$.

$$
\begin{array}{l}
\texttt{Algorithm ChooseAndShift}_{N,k}(x_1, x_2) \\
\quad \texttt{if } (0 \le x_1, x_2 < 2^k - N) \\
\quad\quad \texttt{return} \begin{cases} x_1 & \text{with probability } \frac{1}{2} \\ x_1 + N & \text{with probability } \frac{1}{2} \end{cases} \\
\quad \texttt{elseif } (2^k - N \le x_1, x_2 < N) \\
\quad\quad \texttt{return } x_1 \\
\quad \texttt{else} \\
\quad\quad y_1 \leftarrow \min\{x_1, x_2\}; \quad y_2 \leftarrow \max\{x_1, x_2\} \\
\quad\quad \%\%\% \textit{ Note that } 0 \le y_1 < 2^k - N \textit{ and } 2^k - N \le y_2 < N. \ \%\%\% \\
\quad\quad \texttt{return} \begin{cases} y_1 & \text{with probability } (\frac{1}{2} + \frac{N}{2^{k+1}}) \times \frac{1}{2} \\ y_1 + N & \text{with probability } (\frac{1}{2} + \frac{N}{2^{k+1}}) \times \frac{1}{2} \\ y_2 & \text{with probability } \frac{1}{2} - \frac{N}{2^{k+1}} \end{cases}
\end{array}
$$

Note that $2^{k-1} < N < 2^k$ ensures $2^k - N < N$, $0 < \frac{1}{2} - \frac{N}{2^{k+1}} < 1$, and $0 < \frac{1}{2} + \frac{N}{2^{k+1}} < 1$. In order to run this algorithm, it is sufficient to prepare only $k + 3$ random bits.

We prove the following theorem on the property of `ChooseAndShift`.

Theorem 2.3. *If x_1 and x_2 are independently and uniformly chosen from $\mathbb{Z}_N$ then the output of the above algorithm is uniformly distributed over $[0, 2^k)$.*

Proof. To prove this theorem, we show that if x_1 and x_2 are independently and uniformly chosen from $\mathbb{Z}_N$ then $\Pr[\texttt{ChooseAndShift}_{N,k}(x_1, x_2) = z] = 1/2^k$ for any $z \in [0, 2^k)$. For any $z \in [0, 2^k - N)$, we have

$$
\begin{aligned}
&\Pr[\texttt{ChooseAndShift}(x_1, x_2) = z] \\
&\quad = \Pr[x_1 = z \ \wedge \ 0 \le x_2 < 2^k - N] \times \tfrac{1}{2} \\
&\qquad + \Pr[(x_1 = z \ \wedge \ 2^k - N \le x_2 < N) \vee (x_2 = z \ \wedge \ 2^k - N \le x_1 < N)] \times (\tfrac{1}{2} + \tfrac{N}{2^{k+1}}) \times \tfrac{1}{2} \\
&\quad = \tfrac{1}{N} \times \tfrac{2^k - N}{N} \times \tfrac{1}{2} + (\tfrac{1}{N} \times \tfrac{2N - 2^k}{N}) \times 2 \times (\tfrac{1}{2} + \tfrac{N}{2^{k+1}}) \times \tfrac{1}{2} = \tfrac{1}{2^k}.
\end{aligned}
$$

It is clear that $\Pr[\texttt{ChooseAndShift}_{N,k}(x_1, x_2) = z'] = \Pr[\texttt{ChooseAndShift}_{N,k}(x_1, x_2) = z' + N]$ for any $z' \in [0, 2^k - N)$. Therefore, we have $\Pr[\texttt{ChooseAndShift}_{N,k}(x_1, x_2) = z] = 1/2^k$ for any $z \in [N, 2^k)$.

Furthermore, for any $z \in [2^k - N, N)$, we have

$$
\begin{aligned}
&\Pr[\texttt{ChooseAndShift}(x_1, x_2) = z] \\
&\quad = \Pr[x_1 = z \ \wedge \ 2^k - N \le x_2 < N] \\
&\qquad + \Pr[(x_1 = z \ \wedge \ 0 \le x_2 < 2^k - N) \vee (x_2 = z \ \wedge \ 0 \le x_1 < 2^k - N)] \times (\tfrac{1}{2} - \tfrac{N}{2^{k+1}}) \\
&\quad = \tfrac{1}{N} \times \tfrac{2N - 2^k}{N} + (\tfrac{1}{N} \times \tfrac{2^k - N}{N}) \times 2 \times (\tfrac{1}{2} - \tfrac{N}{2^{k+1}}) = \tfrac{1}{2^k}.
\end{aligned}
$$

□

By using the algorithm `ChooseAndShift`, we propose a new technique for obtaining the anonymity property. We call this technique "sampling twice."

Sampling Twice Doing the evaluation of the encryption (respectively the signing) twice with plaintext x (resp. message m), random r_1 and r_2, and the RSA function, and applying our proposed algorithm `ChooseAndShift` for the two resulting values.

In Chapters 3 to 5, by applying the sampling twice technique, we construct the schemes for public-key encryption, undeniable and confirmer signature, and ring signature, respectively.

CHAPTER 3

Anonymity on Public-Key Encryption

In this chapter, we consider the public-key encryption schemes with anonymity. In [3], Bellare, Boldyreva, Desai, and Pointcheval provided the key-privacy encryption scheme, RSA-RAEP, which is a variant of RSA-OAEP (Bellare and Rogaway [7], Fujisaki, Okamoto, Pointcheval, and Stern [43]). They constructed RSA-RAEP by using the repeating technique in order to prove the anonymity property of their scheme. In this chapter, we propose three public-key encryption schemes, which are also variants of RSA-OAEP, with the expanding technique, RSACD, and the sampling twice technique, and prove their security.

The organization of this chapter is as follows. We review the definitions of public-key encryption in Section 3.1, and RSA-RAEP by Bellare, Boldyreva, Desai, Pointcheval in Section 3.2. We propose a key-privacy encryption scheme with the expanding technique in Section 3.3, that with RSACD in Section 3.4, and that with the sampling twice technique in Section 3.5. We compare the efficiency of four schemes in Section 3.6.

3.1 Definitions of Public-Key Encryption

The classical security requirements of public-key encryption schemes provide privacy of the encryption data. Popular formalizations—such as indistinguishability (semantic security) [48] or non-malleability [37], under either chosen-plaintext or various kinds of chosen-ciphertext attacks [65, 74]—are directed at capturing various data-privacy requirements. (See [4] for a comprehensive treatment).

In [3], Bellare, Boldyreva, Desai, and Pointcheval proposed a new (additional) security requirement of encryption schemes called "key-privacy." It asks that the encryption provide (in addition to privacy of the data being encrypted) privacy of the key under which the encryption was performed. In a heterogeneous public-key environment, encryption will probably fail to be anonymous for trivial reasons. For example, different users might be using different cryptosystems, or, if the same cryptosystem, have keys of different lengths. In [3], Bellare, Boldyreva, Desai, and Pointcheval put a common-key generation algorithm into the standard definition of public-key encryption scheme explicitly. The common key consists of some fixed "global" information which the users may share. A public-key encryption scheme with common-key generation [3] is described as follows.

Definition 3.1 (public-key encryption)**.** *A public-key encryption scheme with common-key generation $\mathcal{PE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ consists of four algorithms.*

- *The common-key generation algorithm $\mathcal{G}(k)$ takes as input a security parameter k and returns some common key I.*
- *The key generation algorithm $\mathcal{K}(I)$ is a randomized algorithm that takes as input a common key I and returns a pair (pk, sk) of keys, a public key and a matching secret key. For given pk, the message space $\mathtt{MSPC}(pk)$ and the randomness space $\mathtt{COINS}(pk)$ of Π are uniquely determined.*
- *The encryption algorithm $\mathcal{E}_{pk}(m; r)$ is a randomized algorithm that takes a public key pk and a plaintext $m \in \mathtt{MSPC}(pk)$, and returns a ciphertext c, using random coin $r \in \mathtt{COINS}(pk)$.*
- *The decryption algorithm $\mathcal{D}_{sk}(c)$ is a deterministic algorithm that takes a secret key sk and a ciphertext c, and returns the corresponding plaintext m or a special symbol $\perp$ to indicate that the ciphertext c is invalid.*

We require that, for any $k \in \mathbb{N}$, if $I \leftarrow \mathcal{G}(k)$, $(pk, sk) \leftarrow \mathcal{K}(I)$, $m \in \mathtt{MSPC}(pk)$, and $c \leftarrow \mathcal{E}_{pk}(m)$, then $m = \mathcal{D}_{sk}(c)$.

The notions of security typically considered for encryption schemes are "indistinguishability of encryptions" under either the chosen-plaintext attack, or the (adaptive) chosen-ciphertext attack. These properties ask that the encryption provides privacy of the data being encrypted. Before describing the definition of "key-privacy" by Bellare, Boldyreva, Pointcheval, and Desai, we briefly review the definitions of "indistinguishability of encryptions."

Definition 3.2 (IND-CPA, IND-CCA)**.** *Let $\mathcal{PE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme. Let $b \in \{0, 1\}$ and $k \in \mathbb{N}$. Let $A_{\mathrm{cpa}} = (A^1_{\mathrm{cpa}}, A^2_{\mathrm{cpa}})$, $A_{\mathrm{cca}} = (A^1_{\mathrm{cca}}, A^2_{\mathrm{cca}})$ be adversaries that*

run in two stages and where A_{cca} *has access to the oracle* $\mathcal{D}_{sk}(\cdot)$. *For* atk $\in$ {cpa, cca}, *we consider the following experiment:*

$$\begin{array}{l}\texttt{Experiment } \mathbf{Exp}^{\text{ind-atk-}b}_{\mathcal{PE},A_{\mathrm{atk}}}(k)\\ \quad I \xleftarrow{R} \mathcal{G}(k);\ (pk, sk) \xleftarrow{R} \mathcal{K}(I)\\ \quad (m_0, m_1, \mathsf{si}) \leftarrow A^1_{\mathrm{atk}}(pk);\ c \leftarrow \mathcal{E}_{pk}(m_b);\ d \leftarrow A^2_{\mathrm{atk}}(c, \mathsf{si})\\ \quad \texttt{return } d\end{array}$$

Note that si *is the state information. It contains* m_0, m_1, pk, *and so on. In the above experiment, it is mandated that* A^2_{cca} *never queries* $\mathcal{D}_{sk}(\cdot)$ *on the challenge ciphertext* c. *For* atk $\in$ {cpa, cca}, *we define the advantages via*

$$\mathbf{Adv}^{\text{ind-atk}}_{\mathcal{PE},A_{\mathrm{atk}}}(k) = \Big|\Pr[\mathbf{Exp}^{\text{ind-atk-1}}_{\mathcal{PE},A_{\mathrm{atk}}}(k) = 1] - \Pr[\mathbf{Exp}^{\text{ind-atk-0}}_{\mathcal{PE},A_{\mathrm{atk}}}(k) = 1]\Big|.$$

The scheme $\mathcal{PE}$ *is said to be IND-CPA secure (respectively IND-CCA secure) if the function* $\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{PE},A_{\mathrm{cpa}}}(\cdot)$ *(resp.* $\mathbf{Adv}^{\text{ind-cca}}_{\mathcal{PE},A_{\mathrm{cca}}}(\cdot)$ *) is negligible for any adversary* A *whose time complexity is polynomial in* k.

In [3], they formalized the property of "key-privacy." It asks that the encryption provide (in addition to privacy of the data being encrypted) privacy of the key under which the encryption was performed. Similar notions had been proposed Abadi and Rogaway [1], Fischlin [39], Camenisch and Lysyanskaya [15], Sako [77], and Desai [34], however, chosen-ciphertext attacks do not seem to have been considered before in the context of key-privacy. The definition by Bellare, Boldyreva, Desai, and Pointcheval [3] can be considered under either the chosen-plaintext attack or the chosen-ciphertext attack, yielding two notions of security, IK-CPA and IK-CCA. (IK means "indistinguishability of keys".)

Definition 3.3 (IK-CPA, IK-CCA [3])**.** *Let* $\mathcal{PE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *be an encryption scheme. Let* $b \in \{0, 1\}$ *and* $k \in \mathbb{N}$. *Let* $A_{\mathrm{cpa}} = (A^1_{\mathrm{cpa}}, A^2_{\mathrm{cpa}})$, $A_{\mathrm{cca}} = (A^1_{\mathrm{cca}}, A^2_{\mathrm{cca}})$ *be adversaries that run in two stages where* A_{cca} *has access to the oracles* $\mathcal{D}_{sk_0}(\cdot)$ *and* $\mathcal{D}_{sk_1}(\cdot)$. *Note that* si *is the state information. It contains* pk_0, pk_1, *and so on. For* atk $\in$ {cpa, cca}, *we consider the following experiments:*

$$\begin{array}{l}\texttt{Experiment } \mathbf{Exp}^{\text{ik-atk-}b}_{\mathcal{PE},A_{\mathrm{atk}}}(k)\\ \quad I \leftarrow \mathcal{G}(k);\ (pk_0, sk_0) \leftarrow \mathcal{K}(I);\ (pk_1, sk_1) \leftarrow \mathcal{K}(I)\\ \quad (m, \mathsf{si}) \leftarrow A^1_{\mathrm{atk}}(pk_0, pk_1);\ y \leftarrow \mathcal{E}_{pk_b}(m)\\ \quad d \leftarrow A^2_{\mathrm{atk}}(y, \mathsf{si})\\ \quad \texttt{return } d\end{array}$$

Above it is mandated that A^2_{cca} *never queries the challenge ciphertext* y *to either* $\mathcal{D}_{sk_0}(\cdot)$ *or* $\mathcal{D}_{sk_1}(\cdot)$. *For* atk $\in$ {cpa, cca}, *we define the advantages via*

$$\mathbf{Adv}^{\text{ik-atk}}_{\mathcal{PE},A_{\mathrm{atk}}}(k) = \Big|\Pr[\mathbf{Exp}^{\text{ik-atk-1}}_{\mathcal{PE},A_{\mathrm{atk}}}(k) = 1] - \Pr[\mathbf{Exp}^{\text{ik-atk-0}}_{\mathcal{PE},A_{\mathrm{atk}}}(k) = 1]\Big|.$$

The scheme $\mathcal{PE}$ is said to be IK-CPA secure (respectively IK-CCA secure) if the function $\mathbf{Adv}^{\text{ik-cpa}}_{\mathcal{PE},A_{\text{cpa}}}(\cdot)$ *(resp.* $\mathbf{Adv}^{\text{ik-cca}}_{\mathcal{PE},A_{\text{cca}}}(\cdot)$*) is negligible for any adversary A whose time complexity is polynomial in k.*

3.2 RSA-RAEP by Bellare, Boldyreva, Desai, and Pointcheval

In [3], Bellare, Boldyreva, Desai, and Pointcheval proposed an RSA-based encryption scheme which is secure in the sense of IK-CCA. It is RSA-RAEP which is a variant of RSA-OAEP (Bellare and Rogaway [7], Fujisaki, Okamoto, Pointcheval, and Stern [43]). Since their variant chooses N from $(2^{k-1}, 2^k)$, it simply repeats the ciphertext computation, each time using new coins, until the ciphertext y satisfies $y < 2^{k-1}$.

Definition 3.4 (RSA-RAEP [3])**.** *RSA-RAEP $= (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is as follows. The common-key generation algorithm $\mathcal{G}$ takes a security parameter k and returns parameters k, k_0 and k_1 such that $k_0(k) + k_1(k) < k$ for all $k > 1$. This defines an associated plaintext-length function $n(k) = k - k_0(k) - k_1(k)$. The key generation algorithm $\mathcal{K}$ takes k, k_0, k_1, runs the key-generation algorithm of* RSA*, and gets N, e, d. The public key pk is $(N, e), k, k_0, k_1$ and the secret key sk is $(N, d), k, k_0, k_1$. The other algorithms are depicted below. Let $G : \{0,1\}^{k_0} \to \{0,1\}^{n+k_1}$ and $H : \{0,1\}^{n+k_1} \to \{0,1\}^{k_0}$ be hash functions. Note that $[x]^n$ denotes the n most significant bits of x and $[x]_m$ denotes the m least significant bits of x.*

```
Algorithm E^{G,H}_{pk}(x)
  ctr = −1
  repeat
    ctr ← ctr + 1
    r ←R {0,1}^{k0}
    s ← (x || 0^{k1}) ⊕ G(r);  t ← r ⊕ H(s)
    v ← (s||t)^e mod N
  until ((v < 2^{k−1}) ∨ (ctr = k1))
  if (ctr = k1) y ← 1||0^{k0+k1}||x
  else y ← 0||v
  return y
```

```
Algorithm D^{G,H}_{sk}(y)
  b ← [y]^1;  v ← [y]_{k0+k1+n}
  if (b = 1)
    w ← [v]^{k0+k1};  x ← [v]_n
    if (w = 0^{k0+k1}) z ← x else z ←⊥
  else
    s ← [v^d]^{n+k1};  t ← [v^d]_{k0}
    r ← t ⊕ H(s)
    x ← [s ⊕ G(r)]^n;  p ← [s ⊕ G(r)]_{k1}
    if (p = 0^{k1}) z ← x else z ←⊥
  return z
```

They proved RSA-RAEP is secure in the sense of IND-CCA and IK-CCA in the random oracle model assuming RSA is one-way.

Remark 3.1 (random oracle model)**.** *The random oracle model [6] provides a mathematical model of an "ideal" hash function. In this model, a hash function $h : X \to Y$ is chosen randomly from $\mathcal{F}^{X,Y}$ which is the set of all functions from X to Y, and we are only*

permitted oracle access to the function h. This means that we are not given a formula or an algorithm to compute values of the function h. Therefore, the only way to compute the value $h(x)$ is to query the oracle. This can be thought of as looking up the value $h(x)$ in a giant book of random numbers such that, for each possible x, there is completely random value $h(x)$.

3.3 OAEP with Expanding

In this section, we propose an encryption scheme by using the expanding technique.

Definition 3.5. *The common-key generation algorithm, the key generation algorithm, and hash functions are the same as those for RSA-RAEP. The other algorithms are depicted below. The other algorithms are depicted below. Note that the valid ciphertext y satisfies $y \in [0, 2^{k+160})$ and $(y \bmod N) \in \mathbb{Z}_N^*$.*

$\texttt{Algorithm}\ \mathcal{E}_{pk}^{G,H}(m)$	$\texttt{Algorithm}\ \mathcal{D}_{sk}^{G,H}(y)$
$r \xleftarrow{R} \{0,1\}^{k_0}$	$v \leftarrow y \bmod N$
$s \leftarrow (m\|\|0^{k_1}) \oplus G(r)$	$s \leftarrow [v^d \bmod N]^{n+k_1}$
$t \leftarrow r \oplus H(s)$	$t \leftarrow [v^d \bmod N]_{k_0}$
$v \leftarrow (s\|\|t)^e \bmod N$	$r \leftarrow t \oplus H(s)$
$M \leftarrow \lfloor (2^{k+160} - v)/N \rfloor$	$m \leftarrow [s \oplus G(r)]^n$
$\alpha \xleftarrow{R} \{0, 1, \cdots, M\}$	$p \leftarrow [s \oplus G(r)]_{k_1}$
$y \leftarrow v + \alpha N$	$\texttt{if}\ (p = 0^{k_1})\ z \leftarrow m$
$\texttt{return}\ y$	$\texttt{else}\ z \leftarrow \perp$
	$\texttt{return}\ z$

In order to prove that the scheme with N-ary representation is secure in the sense of IK-CCA, we need the restriction as follows.

For a ciphertext y and a public key $pk = ((N, e), k)$, we define the set of ciphertexts $EC(y, pk)$ called "equivalence class" as

$$EC(y, pk) = \{\check{y} \in \{0,1\}^{k+160} | \check{y} = y \ (\text{mod}\ N)\}.$$

If $y \in \{0,1\}^{k+160}$ is a ciphertext of m_0 for $pk_0 = (N_0, e_0, k)$ then any element $\check{y} \in EC(y, pk_0)$ is also a ciphertext of m_0 under pk_0. Therefore, when y is a challenge ciphertext, the adversary can ask a ciphertext $\check{y} \in EC(y, pk_0)$ to the decryption oracle $\mathcal{D}_{sk_0}$, and if the answer of $\mathcal{D}_{sk_0}$ is m_0 then the adversary knows that y is encrypted by pk_0 and the plaintext of y is m_0.

To prevent this attack, we add some restriction to the adversaries in the definition of IK-CCA. That is, it is mandated that the adversary never queries either $\check{y} \in EC(y, pk_0)$ to $\mathcal{D}_{sk_0}$ or $\check{y} \in EC(y, pk_1)$ to $\mathcal{D}_{sk_1}$.

Similarly, in order to prove that the scheme with N-ary representation is secure in the sense of IND-CCA2, we need the same restriction. That is, in the definition of IND-CCA2, it is mandated that the adversary never queries $\check{y} \in EC(y, pk)$ to $\mathcal{D}_{sk}$.

We think these restrictions are natural and reasonable. Actually, in the case of undeniable and confirmer signature schemes, Galbraith and Mao [44] defined the anonymity on undeniable signature schemes with the above restriction.

If we add these restrictions then we can prove that our scheme provides the key-privacy against the adaptive chosen ciphertext attack in the random oracle model assuming RSA is θ-partial one-way for $\theta > 0.5$. More precisely, we show the following theorem.

Theorem 3.1. *For any adversary A attacking the key-privacy of our scheme under the adaptive chosen ciphertext attack, and making at most q_{dec} queries to decryption oracle, q_{gen} G-oracle queries, and q_{hash} H-oracle queries, there exists a θ-partial inverting adversary M for the RSA family, such that for any k, k_0, k_1, and $\theta = \frac{k-k_0}{k}$,*

$$\mathbf{Adv}^{\text{ik-cca}}_{\mathcal{PE},A}(k) \leq 8q_{\mathrm{hash}} \cdot ((1-\epsilon_1)\cdot(1-\epsilon_2))^{-1} \cdot \mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{RSA},M}(k) \ + q_{\mathrm{gen}} \cdot (1-\epsilon_2)^{-1} \cdot 2^{-k+2}$$

where $\epsilon_1 = \frac{2}{2^{k/2-3}-1} + \frac{1}{2^{159}}$ and $\epsilon_2 = \frac{2q_{\mathrm{dec}}}{2^{k_1}} + \frac{2q_{\mathrm{hash}}}{2^{k-k_0}} + \frac{2q_{\mathrm{gen}}+q_{\mathrm{dec}}+2q_{\mathrm{gen}}q_{\mathrm{dec}}}{2^{k_0}}$, and the running time of M is that of A plus $q_{\mathrm{gen}} \cdot q_{\mathrm{hash}} \cdot O(k^3)$.

Proof. The proof is similar to that for RSA-RAEP. We construct the partial inverting algorithm M for the RSA function using a CCA-adversary A attacking anonymity of our encryption scheme.

Intuition. We assume that the challenge ciphertext for A is $Y \in \{0,1\}^{k+160}$ which was encrypted by $pk = (N, e)$, and $y = Y \bmod N$. In order to distinguish under which key the given ciphertext Y was created, the adversary A has to make queries r and s to oracles G and H, respectively, such that $s = (m||0^{k_1}) \oplus G(r)$ and $y = (s||(r \oplus H(s)))^e \bmod N$. Therefore, A asks s to H with non-negligible probability where s is the $n + k_1$ most significant bits of the e-th root of y modulo N.

We now describe the partial inverting algorithm M for RSA using a CCA-adversary A attacking the anonymity of our encryption scheme. M is given $pk = ((N, e), k)$ and a point $y \in \mathbb{Z}_N^*$ where $|y| = k = n + k_0 + k_1$. Let $sk = ((N, d), k)$ be the corresponding secret key. The algorithm is trying to find the $n + k_1$ most significant bits of the e-th root of y modulo N.

1) M picks $\mu \xleftarrow{R} \{0, 1, 2, \ldots, \lfloor (2^{k+160} - y)/N \rfloor\}$ and sets $Y \leftarrow y + \mu N$.

2) M runs the key generation algorithm of RSA with security parameter k to obtain $pk' = ((N', e'), k)$ and $sk' = ((N', d'), k)$. Then it picks a bit $b \xleftarrow{R} \{0,1\}$, sets $pk_b \leftarrow ((N, e), k)$ and $pk_{1-b} \leftarrow ((N', e'), k)$. If the above y does not satisfy $y \in (\mathbb{Z}^*_{N_0} \cap \mathbb{Z}^*_{N_1})$ then M outputs Fail and halts; else it continues.

3) M initializes for lists, called G-list, H-list, Y_0-list, and Y_1-list to empty. It then runs A as follows. Note that M simulates A's oracles G, H, $\mathcal{D}_{sk_0}$, and $\mathcal{D}_{sk_1}$ as described below.

   3-1) M runs $A_1(pk_0, pk_1)$ and gets (m, si) which is the output of A_1.

   3-2) M runs $A_2(Y, \mathsf{si})$ and gets a bit $d \in \{0,1\}$ which is the output of A_2.

4) M chooses a random pair (h, H_h) from the H-list and outputs h as its guess for the $n + k_1$ most significant bits of the e-th root of y modulo N.

M simulates the random oracles G and H, and the decryption oracle as follows:

- When A makes an oracle query g to G, then for each (h, H_h) on the H-list, M builds $z = h||(g \oplus H_h)$, and computes $y_{h,g,0} = z^{e_0} \bmod N_0$ and $y_{h,g,1} = z^{e_1} \bmod N_1$. For $i \in \{0,1\}$, M checks whether $y = y_{h,g,i}$. If for some h and i such a relation holds, then we have inverted y under pk_i, and we can still correctly simulate G by answering $G_g = h \oplus (m||0^{k_1})$. Otherwise, M outputs a random value G_g of length $n + k_1$. In both cases, M adds (g, G_g) to the G-list. Then, for all h, M checks if the k_1 least significant bits of $h \oplus G_g$ are all 0. If they are, then it adds $y_{h,g,0}$ and $y_{h,g,1}$ to the Y_0-list and the Y_1-list, respectively.

- When A makes an oracle query h to H, M provides A with a random string H_h of length k_0 and adds (h, H_h) to the H-list. Then for each (g, G_g) on the G-list, M builds $z = h||(g \oplus H_h)$, and computes $y_{h,g,0} = z^{e_0} \bmod N_0$ and $y_{h,g,1} = z^{e_1} \bmod N_1$. M checks if the k_1 least significant bits of $h \oplus G_g$ are all 0. If they are, then it adds $y_{h,g,0}$ and $y_{h,g,1}$ to the Y_0-list and the Y_1-list, respectively.

- When for $i \in \{0,1\}$, A makes an oracle query $\hat{y} \in \{0,1\}^{k+160}$ to $\mathcal{D}_{sk_i}$, M checks if there exists some $y_{h,g,i}$ in the Y_i-list such that $\hat{y} \bmod N_i = y_{h,g,i}$. If there is, then it returns the n most significant bits of $h \oplus G_g$ to A. Otherwise it returns $\perp$ (indicating that $\hat{y}$ is an invalid ciphertext).

In order to analyze the advantage of M, we define some events. For $i \in \{0,1\}$, let $w_i = y^{d_i} \bmod N_i$, $s_i = [w_i]^{n+k_1}$, and $t_i = [w_i]_{k_0}$. That is, w_i is the e_i-th root of y modulo N_i and s_i is the $n + k_1$ most significant bits of the e_i-th root of y modulo N_i. Note that M wins the game if it outputs s_b. Let r_i be the random variable $t_i \oplus H(s_i)$.

We consider the following events.

- FBad denotes the event that
  - A G-oracle query r_0 was made by A_1 in step 3-1, and $G_{r_0} \neq s_0 \oplus (m||0^{k_1})$, or
  - A G-oracle query r_1 was made by A_1 in step 3-1, and $G_{r_1} \neq s_1 \oplus (m||0^{k_1})$.
- GBad denotes the event that
  - A G-oracle query r_0 was made by A_2 in step 3-2, and at the point in time that it was made, the H-oracle query s_0 was not on the H-list, and $G_{r_0} \neq s_0 \oplus (m||0^{k_1})$, or
  - A G-oracle query r_1 was made by A_2 in step 3-2, and at the point in time that it was made, the H-oracle query s_1 was not on the H-list, and $G_{r_1} \neq s_1 \oplus (m||0^{k_1})$.
- DBad denotes the event that
  - A $\mathcal{D}_{sk_0}$ query is not correctly answered, or
  - A $\mathcal{D}_{sk_1}$ query is not correctly answered.
- $\mathsf{G} = \neg\mathsf{FBad} \wedge \neg\mathsf{GBad} \wedge \neg\mathsf{DBad}$.

We use the events FBad, GBad, and G for proving Lemma 3.1 described below. In this chapter, we omit the proof of Lemma 3.1 since the proof of this lemma is similar to that for RSA-RAEP.

We let $\Pr[\cdot]$ denote the probability distribution in the game defining advantage. We introduce the following additional events:

- YBad denotes the event that $y \notin (\mathbb{Z}^*_{N_0} \cap \mathbb{Z}^*_{N_1})$.
- FAskS denotes the event that H-oracle query s_0 or s_1 was made by A_1 in step 3-1.
- AskR denotes the event that (r_0, G_{r_0}) or (r_1, G_{r_1}) is on the G-list at the end of step 3-2.
- AskS denotes the event that (s_0, H_{s_0}) or (s_1, H_{s_1}) is on the H-list at the end of step 3-2.

We use the event FAskS for proving Lemma 3.1. In this chapter, we omit the proof of Lemma 3.1 since the proof of this lemma is similar to that for RSA-RAEP.

Now, we analyze the advantage of M. The algorithm M wins the game if it outputs s_b. If (s_b, H_{s_b}) is on the H-list, then M outputs s_b with probability at least $1/q_{\mathrm{hash}}$. Thus,

$$\begin{aligned}
&\mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{RSA},M}(k) \\
&\quad \geq \tfrac{1}{q_{\mathrm{hash}}} \cdot \Pr[(s_b, H_{s_b}) \text{ is on the } H\text{-list}] \\
&\quad = \tfrac{1}{2q_{\mathrm{hash}}} \cdot (\Pr[(s_0, H_{s_0}) \text{ is on the } H\text{-list}|b=0] + \Pr[(s_1, H_{s_1}) \text{ is on the } H\text{-list}|b=1]) \\
&\quad \geq \tfrac{1}{2q_{\mathrm{hash}}} \cdot \Pr[\neg\mathsf{YBad}] \cdot (\Pr_1[(s_0, H_{s_0}) \text{ is on the } H\text{-list}|b=0] \\
&\qquad\qquad + \Pr_1[(s_1, H_{s_1}) \text{ is on the } H\text{-list}|b=1])
\end{aligned}$$

where $\Pr_1[\cdot]$ denote the probability distribution in the simulated game where $\neg\mathsf{YBad}$ occurs. Assuming that $\neg\mathsf{YBad}$ occurs, by the random choice of b and symmetry, we have $\Pr_1[(s_i, H_{s_i}) \text{ is on the } H\text{-list}|b=0] = \Pr_1[(s_i, H_{s_i}) \text{ is on the } H\text{-list}|b=1] = \Pr_1[(s_i, H_{s_i}) \text{ is on the } H\text{-list}]$ for $i \in \{0,1\}$. Therefore,

$$\begin{aligned}
&\mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{RSA},M}(k) \\
&\quad \geq \tfrac{1}{2q_{\mathrm{hash}}} \cdot \Pr[\neg\mathsf{YBad}] \cdot (\Pr_1[(s_0, H_{s_0}) \text{ is on the } H\text{-list}] + \Pr_1[(s_1, H_{s_1}) \text{ is on the } H\text{-list}]) \\
&\quad \geq \tfrac{1}{2q_{\mathrm{hash}}} \cdot \Pr[\neg\mathsf{YBad}] \cdot \Pr_1[\mathsf{AskS}].
\end{aligned}$$

We next bound $\Pr_1[\mathsf{AskS}]$. We can bound this probability in a similar way as in the proof of anonymity for RSA-RAEP [3], and we have

$$\Pr_1[\mathsf{AskS}] \geq \frac{1}{2} \cdot \Pr_1[\mathsf{AskR} \wedge \mathsf{AskS}|\neg\mathsf{DBad}] \cdot \Pr_1[\neg\mathsf{DBad}|\neg\mathsf{AskS}].$$

We next bound $\Pr_1[\mathsf{AskR} \wedge \mathsf{AskS}|\neg\mathsf{DBad}]$ and $\Pr_1[\neg\mathsf{DBad}|\neg\mathsf{AskS}]$. Let $\epsilon = \mathbf{Adv}^{\text{ik-cca}}_{\mathcal{PE},A}(k)$. The proofs of the following lemmas are similar to that for RSA-RAEP. Intuitively, Lemma 3.1 states that if M simulates the decryption oracle for the adversary A perfectly, then A makes queries (r, G_r) and (s, H_s) such that $s = (m||0^{k_1}) \oplus G_r$ and $y = (s||(r \oplus H_s))^{e_b} \bmod N_b$ with non-negligible probability. Lemma 3.2 states that M can simulate the decryption oracle with overwhelming probability.

Lemma 3.1.

$$\Pr_1[\mathsf{AskR} \wedge \mathsf{AskS}|\neg\mathsf{DBad}] \geq \frac{\epsilon}{2} \cdot \left(1 - \left(\frac{2q_{\mathrm{gen}}}{2^{k_0}} + \frac{2q_{\mathrm{hash}}}{2^{n+k_1}}\right)\right) - \frac{2q_{\mathrm{gen}}}{2^k}.$$

Lemma 3.2.

$$\Pr_1[\mathsf{DBad}|\neg\mathsf{AskS}] \leq q_{\mathrm{dec}} \cdot \left(\frac{2}{2^{k_1}} + \frac{2q_{\mathrm{gen}}+1}{2^{k_0}}\right).$$

By applying Lemmas 3.1 and 3.2, we have

$$\begin{aligned}
&\Pr_1[\mathsf{AskS}] \\
&\geq \tfrac{1}{2}\cdot\left[\tfrac{\epsilon}{2}\cdot\left(1-\left(\tfrac{2q_{\mathrm{gen}}}{2^{k_0}}+\tfrac{2q_{\mathrm{hash}}}{2^{n+k_1}}\right)\right)-\tfrac{2q_{\mathrm{gen}}}{2^k}\right]\times\left[1-q_{\mathrm{dec}}\cdot\left(\tfrac{2}{2^{k_1}}+\tfrac{2q_{\mathrm{gen}}+1}{2^{k_0}}\right)\right] \\
&= \tfrac{\epsilon}{4}\cdot\left(1-\left(\tfrac{2q_{\mathrm{gen}}}{2^{k_0}}+\tfrac{2q_{\mathrm{hash}}}{2^{n+k_1}}\right)\right)\times\left[1-q_{\mathrm{dec}}\cdot\left(\tfrac{2}{2^{k_1}}+\tfrac{2q_{\mathrm{gen}}+1}{2^{k_0}}\right)\right] \\
&\qquad -\tfrac{1}{2}\cdot\tfrac{2q_{\mathrm{gen}}}{2^k}\cdot\left[1-q_{\mathrm{dec}}\cdot\left(\tfrac{2}{2^{k_1}}+\tfrac{2q_{\mathrm{gen}}+1}{2^{k_0}}\right)\right] \\
&\geq \tfrac{\epsilon}{4}\cdot\left(1-\left(\tfrac{2q_{\mathrm{gen}}}{2^{k_0}}+\tfrac{2q_{\mathrm{hash}}}{2^{n+k_1}}\right)-q_{\mathrm{dec}}\cdot\left(\tfrac{2}{2^{k_1}}+\tfrac{2q_{\mathrm{gen}}+1}{2^{k_0}}\right)\right)-\tfrac{1}{2}\cdot\tfrac{2q_{\mathrm{gen}}}{2^k} \\
&= \tfrac{\epsilon}{4}\cdot\left(1-\left(\tfrac{2q_{\mathrm{gen}}+q_{\mathrm{dec}}+2q_{\mathrm{gen}}q_{\mathrm{dec}}}{2^{k_0}}+\tfrac{2q_{\mathrm{dec}}}{2^{k_1}}+\tfrac{2q_{\mathrm{hash}}}{2^{k-k_0}}\right)\right)-\tfrac{q_{\mathrm{gen}}}{2^k}.
\end{aligned}$$

We next bound the probability that $\neg\mathsf{YBad}$ occurs. Note that we cannot bound $\Pr[\mathsf{YBad}]$ by directly applying a similar argument for RSA-RAEP.

Lemma 3.3.

$$\Pr[\mathsf{YBad}] \leq \frac{2}{2^{k/2-3}-1}+\frac{1}{2^{159}}.$$

Proof of Lemma 3.3. Let $N=pq$ and $N'=p'q'$. Note that $2^{\lceil k/2\rceil-1}<p,q,p',q'<2^{\lceil k/2\rceil}$ and $2^{k-1}<N,N'<2^k$. We define a set $S[N]$ as $\{\tilde{Y}|\tilde{Y}\in[0,2^{k+160})\wedge(\tilde{Y} \bmod N)\in\mathbb{Z}_N^*\}$. Then, we have

$$\begin{aligned}
&\Pr[\mathsf{YBad}] \\
&= \Pr[y\xleftarrow{R}\mathbb{Z}_N^*;\ \mu\xleftarrow{R}\{0,1,2,\ldots,\lfloor(2^{k+160}-y)/N\rfloor\};\ Y\leftarrow y+\mu N:\ Y\notin S[N']] \\
&\leq \Pr[Y'\xleftarrow{R}S[N]:\ Y'\notin S[N']]+1/2^{159}
\end{aligned}$$

since the distribution of Y' is statistically indistinguishable from that of Y, and the statistically distance is less than $1/2^{159}$.

Since $2^{160}\cdot\phi(N)\leq|S[N]|$, we have

$$\begin{aligned}
\Pr[Y'\xleftarrow{R}S[N]:Y'\notin S[N']] &\leq \frac{|\{y\,|\,y\in S[N]\ \wedge\ y\notin S[N']\}|}{|S[N]|} \\
&\leq \frac{|\{y\,|\,y\in[0,2^{k+160})\ \wedge\ y\notin S[N']\}|}{|S[N]|} \\
&\leq \frac{2^{k+160}-|S[N']|}{|S[N]|}\leq\frac{2^{k+160}-|S[N']|}{2^{160}\cdot\phi(N)}.
\end{aligned}$$

Furthermore, we have

$$\begin{aligned}
2^{k+160}-|S[N']| &= \left|\{Y'|Y'\in[0,2^{k+160})\wedge(Y' \bmod N')\notin\mathbb{Z}_{N'}^*\}\right| \\
&\leq \left|\{Y'|Y'\in[0,2N'\cdot2^{160})\wedge(Y' \bmod N')\notin\mathbb{Z}_{N'}^*\}\right| \\
&= 2^{161}\times\left|\{Y'|Y'\in[0,N')\wedge Y'\notin\mathbb{Z}_{N'}^*\}\right| \\
&= 2^{161}(N'-\phi(N')).
\end{aligned}$$

Therefore, we can bound $\Pr[Y' \stackrel{R}{\leftarrow} S[N] : Y' \notin S[N']]$ as

$$\begin{aligned}
&\Pr[Y' \stackrel{R}{\leftarrow} S[N] : Y' \notin S[N']] \\
&\leq \frac{2^{k+160} - |S[N']|}{2^{160} \cdot \phi(N)} \leq \frac{2^{161}(N' - \phi(N'))}{2^{160} \cdot \phi(N)} = \frac{2(p' + q' - 1)}{N - p - q + 1} \leq \frac{2(p' + q')}{N - p - q} \\
&\leq \frac{2(2^{\lceil k/2 \rceil} + 2^{\lceil k/2 \rceil})}{2^{k-1} - 2^{\lceil k/2 \rceil} - 2^{\lceil k/2 \rceil}} = \frac{2(1+1)}{2^{k-1-\lceil k/2 \rceil} - 1 - 1} \leq \frac{4}{2^{k/2-2} - 2} = \frac{2}{2^{k/2-3} - 1}.
\end{aligned}$$

□

Substituting the bounds for the above probabilities, we have

$$\mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{RSA},M}(k) \geq \frac{1}{2q_{\mathrm{hash}}} \cdot (1 - \epsilon_1) \cdot \left(\frac{\epsilon}{4} \cdot (1 - \epsilon_2) - \frac{q_{\mathrm{gen}}}{2^k} \right)$$

where $\epsilon_1 = \frac{2}{2^{k/2-3}-1} + \frac{1}{2^{159}}$ and $\epsilon_2 = \frac{2q_{\mathrm{gen}} + q_{\mathrm{dec}} + 2q_{\mathrm{gen}}q_{\mathrm{dec}}}{2^{k_0}} + \frac{2q_{\mathrm{dec}}}{2^{k_1}} + \frac{2q_{\mathrm{hash}}}{2^{k-k_0}}$, and re-arranging the terms, we get the claimed result. Note that $\epsilon = \mathbf{Adv}^{\text{ik-cca}}_{\mathcal{PE},A}(k)$.

Finally, we estimate the time complexity of M. It is the time complexity of A plus the time for simulating the random oracles. In the random oracle simulation, for each pair $((g, G_g), (h, H_h))$, it is sufficient to compute $y_{h,g,0} = (h||(g \oplus H_h))^{e_0} \bmod N_0$ and $y_{h,g,1} = (h||(g \oplus H_h))^{e_1} \bmod N_1$. Therefore, the time complexity of M is that of A plus $q_{\mathrm{gen}} \cdot q_{\mathrm{hash}} \cdot O(k^3)$.

□

Fujisaki, Okamoto, Pointcheval, and Stern [43] proved that the θ-partial one-wayness of RSA is equivalent to the one-wayness of RSA for $\theta > 0.5$. Therefore, the scheme with N-ary representation is secure in the sense of IK-CCA in the random oracle model assuming that RSA is one-way.

We can also prove that the scheme with N-ary representation is secure in the sense of IND-CCA in the random oracle model assuming RSA is one-way with the restriction mentioned above. More precisely, we prove that if there exists a CCA-adversary $A = (A_1, A_2)$ attacking the indistinguishability of our scheme with advantage ϵ, then there exists a CCA-adversary $B = (B_1, B_2)$ attacking the indistinguishability of RSA-OAEP with the same advantage ϵ. We construct B as follows.

1) B_1 gets pk and passes it to A_1. B_1 gets (m_0, m_1, si) which is an output of A_1, and B_1 outputs it.

2) B_2 gets a challenge ciphertext y, sets $y' \leftarrow y + tN$ where $t \stackrel{R}{\leftarrow} \{0, 1, 2, \cdots, \lfloor (2^{k+160} - y)/N \rfloor\}$, and passes (y', si) to A_2. B_2 gets $d \in \{0, 1\}$ which is an output of A_2, and outputs it.

It is easy to see that the advantage of B is the same as that for A. Since RSA-OAEP is secure in the sense of IND-CCA in the random oracle model assuming RSA is one-way (Fujisaki, Okamoto, Pointcheval, and Stern [43]), our scheme is also secure in the sense of IND-CCA in the random oracle model assuming RSA is one-way.

3.4 OAEP with RSACD

In this section, we propose a key-privacy encryption scheme which uses RSACD, which we have proposed in in Section 2.3.2.

Definition 3.6. *The common-key generation algorithm $\mathcal{G}$ and hash functions are the same as those for RSA-RAEP. The key generation algorithm $\mathcal{K}$ takes k, k_0, k_1, runs the key-generation algorithm of* RSACD*, and gets N, e, d. The public key pk is $(N, e), k, k_0, k_1$ and the secret key sk is $(N, d), k, k_0, k_1$. The other algorithms are described as follows. Note that the valid ciphertext y satisfies $y \in [0, 2^k)$ and $(y \bmod N) \in \mathbb{Z}_N^*$.*

$\mathtt{Algorithm}\ \mathcal{E}_{pk}^{G,H}(x)$	$\mathtt{Algorithm}\ \mathcal{D}_{sk}^{G,H}(y)$
$r \xleftarrow{R} \{0,1\}^{k_0}$	$s \leftarrow [g_{N,d,k}^{\mathsf{RSACD}}(y)]^{n+k_1};\ t \leftarrow [g_{N,d,k}^{\mathsf{RSACD}}(y)]_{k_0}$
$s \leftarrow (x \mid\mid 0^{k_1}) \oplus G(r)$	$r \leftarrow t \oplus H(s)$
$t \leftarrow r \oplus H(s)$	$x \leftarrow [s \oplus G(r)]^n;\ p \leftarrow [s \oplus G(r)]_{k_1}$
$v \leftarrow f_{N,e,k}^{\mathsf{RSACD}}(s\|\|t)$	$\mathtt{if}\ (p = 0^{k_1})\ z \leftarrow x\ \mathtt{else}\ z \leftarrow \perp$
$\mathtt{return}\ y$	$\mathtt{return}\ z$

Fujisaki, Okamoto, Pointcheval, and Stern [43] proved OAEP with the partial one-way function is secure in the sense of IND-CCA. Thus, OAEP with the RSACD function is secure in the sense of IND-CCA assuming RSACD is partial one-way.

Furthermore, we can show the following theorem.

Theorem 3.2. *For any adversary A attacking the key-privacy of our scheme with RSACD under the adaptive chosen ciphertext attack, and making at most q_{dec} queries to decryption oracle, q_{gen} G-oracle queries, and q_{hash} H-oracle queries, there exists a θ-partial inverting adversary M for the RSACD family, such that for any k, k_0, k_1, and $\theta = \frac{k-k_0}{k}$,*

$$\mathbf{Adv}^{\text{ik-cca}}_{\mathcal{PE},A}(k) \leq 8q_{\mathrm{hash}} \cdot ((1-\epsilon_1)\cdot(1-\epsilon_2))^{-1} \cdot \mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{RSACD},M}(k) \ + q_{\mathrm{gen}} \cdot (1-\epsilon_2)^{-1} \cdot 2^{-k+2}$$

where $\epsilon_1 = \frac{2}{2^{k/2-3}-1}$ and $\epsilon_2 = \frac{2q_{\mathrm{dec}}}{2^{k_1}} + \frac{2q_{\mathrm{hash}}}{2^{k-k_0}} + \frac{2q_{\mathrm{gen}}+q_{\mathrm{dec}}+2q_{\mathrm{gen}}q_{\mathrm{dec}}}{2^{k_0}}$, and the running time of M is that of A plus $q_{\mathrm{gen}} \cdot q_{\mathrm{hash}} \cdot O(k^3)$.

Thus, our scheme with RSACD is secure in the sense of IND-CCA and IK-CCA assuming RSACD is partial one-way. Hence, from Theorems 2.1 and 2.2, our scheme with RSACD is secure in the sense of IND-CCA and IK-CCA assuming RSA is one-way.

Proof of Theorem 3.2. The proof is similar to that for our scheme with expanding. We describe the partial inverting algorithm M for RSACD using a CCA-adversary A attacking the anonymity of our encryption scheme with RSACD. M is given $pk = ((N, e), k)$ and a point $y = f^{\mathsf{RSACD}}_{N,e,k}(x)$ where $|y| = k = n + k_0 + k_1$ and $x \xleftarrow{R} \mathrm{Dom}_{\mathsf{RSACD}}(N, e, k)$. Let $sk = ((N, d), k)$ be the corresponding secret key. The algorithm is trying to find the $n + k_1$ most significant bits of x.

1) M runs the key generation algorithm of RSACD with security parameter k to obtain $pk' = ((N', e'), k)$ and $sk' = ((N', d'), k)$. Then it picks a bit $b \xleftarrow{R} \{0, 1\}$, sets $pk_b \leftarrow ((N, e), k)$ and $pk_{1-b} \leftarrow ((N', e'), k)$. If the above y does not satisfy $y \in (\mathrm{Rng}_{\mathsf{RSACD}}(N_0, e_0, k) \cap \mathrm{Rng}_{\mathsf{RSACD}}(N_1, e_1, k))$ then M outputs Fail and halts; else it continues.

2) M initializes for lists, called G-list, H-list, Y_0-list, and Y_1-list to empty. It then runs A as follows. Note that M simulates A's oracles G, H, $\mathcal{D}_{sk_0}$, and $\mathcal{D}_{sk_1}$ as described below.

   2-1) M runs $A_1(pk_0, pk_1)$ and gets (m, si) which is the output of A_1.

   2-2) M runs $A_2(Y, \mathsf{si})$ and gets a bit $d \in \{0, 1\}$ which is the output of A_2.

3) M chooses a random pair (h, H_h) from the H-list and outputs h as its guess for the $n + k_1$ most significant bits of the e-th root of y modulo N.

M simulates the random oracles G and H, and the decryption oracle as follows:

- When A makes an oracle query g to G, then for each (h, H_h) on the H-list, M builds $z = h||(g \oplus H_h)$, and computes $y_{h,g,0} = f^{\mathsf{RSACD}}_{N_0,e_0,k}(z)$ and $y_{h,g,1} = f^{\mathsf{RSACD}}_{N_1,e_1,k}(z)$. For $i \in \{0, 1\}$, M checks whether $y = y_{h,g,i}$. If for some h and i such a relation holds, then we have inverted y under pk_i, and we can still correctly simulate G by answering $G_g = h \oplus (m||0^{k_1})$. Otherwise, M outputs a random value G_g of length $n + k_1$. In both cases, M adds (g, G_g) to the G-list. Then, for all h, M checks if the k_1 least significant bits of $h \oplus G_g$ are all 0. If they are, then it adds $y_{h,g,0}$ and $y_{h,g,1}$ to the Y_0-list and the Y_1-list, respectively.

- When A makes an oracle query h to H, M provides A with a random string H_h of length k_0 and adds (h, H_h) to the H-list. Then for each (g, G_g) on the G-list, M builds $z = h||(g \oplus H_h)$, and computes $y_{h,g,0} = f^{\mathsf{RSACD}}_{N_0,e_0,k}(z)$ and $y_{h,g,1} = f^{\mathsf{RSACD}}_{N_1,e_1,k}(z)$. M checks if the k_1 least significant bits of $h \oplus G_g$ are all 0. If they are, then it adds $y_{h,g,0}$ and $y_{h,g,1}$ to the Y_0-list and the Y_1-list, respectively.

- When for $i \in \{0,1\}$, A makes an oracle query $\hat{y} \in \{0,1\}^k$ to $\mathcal{D}_{sk_i}$, M checks if there exists some $y_{h,g,i}$ in the Y_i-list such that $\hat{y} = y_{h,g,i}$. If there is, then it returns the n most significant bits of $h \oplus G_g$ to A. Otherwise it returns $\perp$ (indicating that $\hat{y}$ is an invalid ciphertext).

In order to analyze the advantage of M, we define some events. For $i \in \{0,1\}$, let $w_i = g^{\mathsf{RSACD}}_{N_i,d_i,k}(y)$, $s_i = [w_i]^{n+k_1}$, and $t_i = [w_i]_{k_0}$. Note that M wins the game if it outputs s_b. Let r_i be the random variable $t_i \oplus H(s_i)$.

We consider the following events.

- FBad denotes the event that
  - A G-oracle query r_0 was made by A_1 in step 2-1, and $G_{r_0} \neq s_0 \oplus (m||0^{k_1})$, or
  - A G-oracle query r_1 was made by A_1 in step 2-1, and $G_{r_1} \neq s_1 \oplus (m||0^{k_1})$.
- GBad denotes the event that
  - A G-oracle query r_0 was made by A_2 in step 2-2, and at the point in time that it was made, the H-oracle query s_0 was not on the H-list, and $G_{r_0} \neq s_0 \oplus (m||0^{k_1})$, or
  - A G-oracle query r_1 was made by A_2 in step 2-2, and at the point in time that it was made, the H-oracle query s_1 was not on the H-list, and $G_{r_1} \neq s_1 \oplus (m||0^{k_1})$.
- DBad denotes the event that
  - A $\mathcal{D}_{sk_0}$ query is not correctly answered, or
  - A $\mathcal{D}_{sk_1}$ query is not correctly answered.
- $\mathsf{G} = \neg\mathsf{FBad} \wedge \neg\mathsf{GBad} \wedge \neg\mathsf{DBad}$.

We use the events FBad, GBad, and G for proving Lemma 3.1 described below. In this chapter, we omit the proof of Lemma 3.1 since the proof of this lemma is similar to that for RSA-RAEP.

We let $\Pr[\cdot]$ denote the probability distribution in the game defining advantage. We introduce the following additional events:

- YBad denotes the event that $y \notin (\mathrm{Rng}_{\mathsf{RSACD}}(N_0, e_0, k) \cap \mathrm{Rng}_{\mathsf{RSACD}}(N_1, e_1, k))$.
- FAskS denotes the event that H-oracle query s_0 or s_1 was made by A_1 in step 2-1.
- AskR denotes the event that (r_0, G_{r_0}) or (r_1, G_{r_1}) is on the G-list at the end of step 2-2.

- AskS denotes the event that (s_0, H_{s_0}) or (s_1, H_{s_1}) is on the H-list at the end of step 2-2.

We use the event FAskS for proving Lemma 3.1. In this chapter, we omit the proof of Lemma 3.1 since the proof of this lemma is similar to that for RSA-RAEP.

Now, we analyze the advantage of M. We can bound the advantage of M in a similar way as that for our scheme with expanding and we have

$$\mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{RSACD},M}(k) \geq \frac{1}{2q_{\mathrm{hash}}} \cdot \Pr[\neg\mathsf{YBad}] \cdot \Pr_1[\mathsf{AskS}].$$

and

$$\Pr_1[\mathsf{AskS}] \geq \frac{\epsilon}{4} \cdot \left(1 - \left(\frac{2q_{\mathrm{gen}} + q_{\mathrm{dec}} + 2q_{\mathrm{gen}}q_{\mathrm{dec}}}{2^{k_0}} + \frac{2q_{\mathrm{dec}}}{2^{k_1}} + \frac{2q_{\mathrm{hash}}}{2^{k-k_0}}\right)\right) - \frac{q_{\mathrm{gen}}}{2^k}$$

where $\Pr_1[\cdot]$ denote the probability distribution in the simulated game where $\neg\mathsf{YBad}$ occurs.

We next bound the probability that $\neg\mathsf{YBad}$ occurs.

Lemma 3.4.

$$\Pr[\mathsf{YBad}] \leq \frac{2}{2^{k/2-3} - 1}.$$

Proof of Lemma 3.4. Let $N = pq$ and $N' = p'q'$. Note that $2^{\lceil k/2\rceil - 1} < p, q, p', q' < 2^{\lceil k/2\rceil}$ and $2^{k-1} < N, N' < 2^k$. Since $\phi(N) \leq |\mathrm{Rng}_{\mathsf{RSACD}}(N, e, k)|$, we have

$$\begin{aligned}
\Pr[\mathsf{YBad}] &\leq \Pr[y \xleftarrow{R} \mathrm{Rng}_{\mathsf{RSACD}}(N, e, k) : y \notin \mathrm{Rng}_{\mathsf{RSACD}}(N', e', k)] \\
&\leq \frac{|\{y \,|\, y \in \mathrm{Rng}_{\mathsf{RSACD}}(N, e, k) \ \wedge \ y \notin \mathrm{Rng}_{\mathsf{RSACD}}(N', e', k)\}|}{|\mathrm{Rng}_{\mathsf{RSACD}}(N, e, k)|} \\
&\leq \frac{|\{y \,|\, y \in [0, 2^k) \ \wedge \ y \notin \mathrm{Rng}_{\mathsf{RSACD}}(N', e', k)\}|}{|\mathrm{Rng}_{\mathsf{RSACD}}(N, e, k)|} \\
&\leq \frac{2^k - |\mathrm{Rng}_{\mathsf{RSACD}}(N', e', k)|}{\phi(N)}.
\end{aligned}$$

Furthermore, we have

$$\begin{aligned}
2^k - |\mathrm{Rng}_{\mathsf{RSACD}}(N', e', k)| &= \left|\{y' \in [0, 2^k) | y' \notin \mathrm{Rng}_{\mathsf{RSACD}}(N', e', k)\}\right| \\
&\leq |\{y' \in [0, 2N') | y' \notin \mathrm{Rng}_{\mathsf{RSACD}}(N', e', k)\}| \\
&= 2 \times |\{y' \in [0, N') | y' \notin \mathrm{Rng}_{\mathsf{RSACD}}(N', e', k)\}| \\
&= 2(N' - \phi(N')).
\end{aligned}$$

Therefore, we can bound $\Pr[\mathsf{YBad}]$ as

$$\begin{aligned}
\Pr[\mathsf{YBad}] &\leq \frac{2^k - |\mathrm{Rng}_{\mathsf{RSACD}}(N', e', k)|}{\phi(N)} \leq \frac{2(N' - \phi(N'))}{\phi(N)} = \frac{2(p' + q' - 1)}{N - p - q + 1} \leq \frac{2(p' + q')}{N - p - q} \\
&\leq \frac{2(2^{\lceil k/2\rceil} + 2^{\lceil k/2\rceil})}{2^{k-1} - 2^{\lceil k/2\rceil} - 2^{\lceil k/2\rceil}} = \frac{2(1+1)}{2^{k-1-\lceil k/2\rceil} - 1 - 1} \leq \frac{4}{2^{k/2-2} - 2} = \frac{2}{2^{k/2-3} - 1}.
\end{aligned}$$

□

Substituting the bounds for the above probabilities, we have

$$\mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{RSACD},M}(k) \geq \frac{1}{2q_{\text{hash}}} \cdot (1-\epsilon_1) \cdot \left(\frac{\epsilon}{4} \cdot (1-\epsilon_2) - \frac{q_{\text{gen}}}{2^k}\right)$$

where $\epsilon_1 = \frac{2}{2^{k/2-3}-1}$ and $\epsilon_2 = \frac{2q_{\text{gen}}+q_{\text{dec}}+2q_{\text{gen}}q_{\text{dec}}}{2^{k_0}} + \frac{2q_{\text{dec}}}{2^{k_1}} + \frac{2q_{\text{hash}}}{2^{k-k_0}}$, and re-arranging the terms, we get the claimed result. Note that $\epsilon = \mathbf{Adv}^{\text{ik-cca}}_{\mathcal{PE},A}(k)$.

Finally, we estimate the time complexity of M. It is the time complexity of A plus the time for simulating the random oracles. In the random oracle simulation, for each pair $((g, G_g), (h, H_h))$, it is sufficient to compute $y_{h,g,0} = f^{\mathsf{RSACD}}_{N_0,e_0,k}(h||(g \oplus H_h))$ and $y_{h,g,1} = f^{\mathsf{RSACD}}_{N_1,e_1,k}(h||(g \oplus H_h))$. Therefore, the time complexity of M is that of A plus $q_{\text{gen}} \cdot q_{\text{hash}} \cdot O(k^3)$. □

3.5 OAEP with Sampling Twice

In this section, we propose a key-privacy encryption scheme with the sampling twice technique.

Definition 3.7. *The common-key generation algorithm, the key generation algorithm, and hash functions are the same as those for RSA-RAEP. The other algorithms are depicted below. Note that the valid ciphertext y satisfies $y \in [0, 2^k)$ and $(y \bmod N) \in \mathbb{Z}^*_N$.*

```
Algorithm E^{G,H}_{pk}(x)
  r_1, r_2 <-R {0,1}^{k_0}
  s_1 ← (x||0^{k_1}) ⊕ G(r_1);  t_1 ← r_1 ⊕ H(s_1)
  v_1 ← (s_1||t_1)^e mod N
  s_2 ← (x||0^{k_1}) ⊕ G(r_2);  t_2 ← r_2 ⊕ H(s_2)
  v_2 ← (s_2||t_2)^e mod N
  y ← ChooseAndShift(v_1, v_2)
  return y
```

```
Algorithm D^{G,H}_{sk}(y)
  v ← y mod N
  s ← [v^d mod N]^{n+k_1};  t ← [v^d mod N]_{k_0}
  r ← t ⊕ H(s)
  x ← [s ⊕ G(r)]^n;  p ← [s ⊕ G(r)]_{k_1}
  if (p = 0^{k_1}) z ← x else z ←⊥
  return z
```

In order to prove that the scheme with sampling twice is secure in the sense of IK-CCA, we need the restriction similar to that for OAEP with expanding.

Since if c is a ciphertext of m for $pk = (N, e, k)$ and $c < 2^k - N$ then $c + N$ is also a ciphertext of m, the adversary can ask $c + N_0$ to decryption oracle $\mathcal{D}_{sk_0}$ where c is a challenge ciphertext such that $c < 2^k - N_0$ and $pk_0 = (N_0, e_0, k)$, and if the answer of $\mathcal{D}_{sk_0}$ is m, then the adversary can know that c was encrypted by pk_0.

To prevent this attack, we add some natural restriction to the adversaries in the definitions of IK-CCA. That is, it is mandated that the adversary never queries either $c' \in [0, 2^k)$ such that $c' = c \pmod{N_0}$ to $\mathcal{D}_{sk_0}$ or $c'' \in [0, 2^k)$ such that $c'' = c \pmod{N_1}$ to $\mathcal{D}_{sk_1}$.

Similarly, in order to prove that the scheme with sampling twice is secure in the sense of IND-CCA, we need the same restriction. That is, in the definition of IND-CCA, it is mandated that the adversary never queries $c' \in [0, 2^k)$ such that $c' = c \pmod N$ to $\mathcal{D}_{sk}$.

If we add these restrictions then we can prove that the scheme with sampling twice is secure in the sense of IK-CCA in the random oracle model assuming RSA is θ-partial one-way for $\theta > 0.5$. More precisely, we prove the following theorem.

Theorem 3.3. *For any adversary A attacking the anonymity of our scheme $\mathcal{PE}$ with sampling twice under an adaptive chosen-ciphertext attack, and making at most q_{dec} decryption oracle queries, q_{gen} G-oracle queries, and q_{hash} H-oracle queries, there exists a θ-partial inverting adversary M for the* RSA *family, such that for any $k, k_0(k), k_1(k)$, and $\theta = \frac{k-k_0(k)}{k}$,*

$$\mathbf{Adv}^{\text{ik-cca}}_{\mathcal{PE},A}(k) \leq 8q_{\mathrm{hash}}((1-\epsilon_1)\cdot(1-\epsilon_2)\cdot(1-\epsilon_3))^{-1} \cdot \mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{RSA},M}(k) + q_{\mathrm{gen}}(1-\epsilon_3)^{-1} \cdot 2^{-k+2}$$

where $\epsilon_1 = \frac{1}{2}$, $\epsilon_2 = \frac{2}{2^{k/2-3}-1}$, and $\epsilon_3 = \frac{2q_{\mathrm{dec}}}{2^{k_1}} + \frac{2q_{\mathrm{hash}}}{2^{k-k_0}} + \frac{2q_{\mathrm{gen}}+q_{\mathrm{dec}}+2q_{\mathrm{gen}}q_{\mathrm{dec}}}{2^{k_0}}$, and the running time of M is that of A plus $q_{\mathrm{gen}} \cdot q_{\mathrm{hash}} \cdot O(k^3)$.

Proof. The proof is similar to that for our scheme with expanding. We describe the RSA partial inverting algorithm M using a CCA-adversary A attacking anonymity of our encryption scheme with sampling twice. M is given $pk = (N, e, k)$ and a point $y \in \mathbb{Z}^*_N$ where $|y| = k = n + k_0 + k_1$. Let $sk = (N, d, k)$ be the corresponding secret key. The algorithm is trying to find the $n + k_1$ most significant bits of the e-th root of y modulo N.

1) M picks a bit $\mu \xleftarrow{R} \{0,1\}$ and sets $Y \leftarrow y + \mu N$. If $Y \geq 2^k$ then outputs Fail and halts; else it continues.

2) M runs the key generation algorithm of RSA with security parameter k to obtain $pk' = (N', e', k)$ and $sk' = (N', d', k)$. Then it picks a bit $b \xleftarrow{R} \{0,1\}$, sets $pk_b \leftarrow (N, e)$ and $pk_{1-b} \leftarrow (N', e')$. If the above y does not satisfy $y \in (\mathbb{Z}^*_{N_0} \cap \mathbb{Z}^*_{N_1})$ then M outputs Fail and halts; else it continues.

3) M initializes for lists, called G-list, H-list, Y_0-list, and Y_1-list to empty. It then runs A as follows. Note that M simulates A's oracles G, H, $\mathcal{D}_{sk_0}$, and $\mathcal{D}_{sk_1}$ as described below.

   3-1) M runs $A_1(pk_0, pk_1)$ and gets (x, si) which is the output of A_1.

   3-2) M runs $A_2(Y, \mathsf{si})$ and gets a bit $d \in \{0,1\}$ which is the output of A_2.

4) M chooses a random pair (h, H_h) from the H-list and outputs h as its guess for the $n + k_1$ most significant bits of the e-th root of y modulo N.

M simulates the random oracles G and H, and the decryption oracle as follows:

- When A makes an oracle query g to G, then for each (h, H_h) on the H-list, M builds $z = h||(g \oplus H_h)$, and computes $y_{h,g,0} = z^{e_0} \bmod N_0$ and $y_{h,g,1} = z^{e_1} \bmod N_1$. For $i \in \{0,1\}$, M checks whether $y = y_{h,g,i}$. If for some h and i such a relation holds, then we have inverted y under pk_i, and we can still correctly simulate G by answering $G_g = h \oplus (x||0^{k_1})$. Otherwise, M outputs a random value G_g of length $n + k_1$. In both cases, M adds (g, G_g) to the G-list. Then, for all h, M checks if the k_1 least significant bits of $h \oplus G_g$ are all 0. If they are, then it adds $y_{h,g,0}$ and $y_{h,g,1}$ to the Y_0-list and the Y_1-list respectively.

- When A makes an oracle query h to H, M provides A with a random string H_h of length k_0 and adds (h, H_h) to the H-list. Then for each (g, G_g) on the G-list, M builds $z = h||(g \oplus H_h)$, and computes $y_{h,g,0} = z^{e_0} \bmod N_0$ and $y_{h,g,1} = z^{e_1} \bmod N_1$. M checks if the k_1 least significant bits of $h \oplus G_g$ are all 0. If they are, then it adds $y_{h,g,0}$ and $y_{h,g,1}$ to the Y_0-list and the Y_1-list respectively.

- When for $i \in \{0,1\}$, A makes an oracle query $y' \in \{0,1\}^k$ to $\mathcal{D}_{sk_i}$, M checks if there exists some $y_{h,g,i}$ in the Y_i-list such that $y' \bmod N_i = y_{h,g,i}$. If there is, then it returns the n most significant bits of $h \oplus G_g$ to A. Otherwise it returns $\perp$ (indicating that y' is an invalid ciphertext).

Now, we analyze the advantage of M. In the following, we consider the experiment where M does not output Fail in the first step. In this experiment, we can consider the distributions of N, e, and Y as $((N, e, k), (N, d, k)) \leftarrow K(k)$; $Y \xleftarrow{R} S[N]$ where $\mathcal{K}$ is the key generation algorithm of RSA and $S[N] = \{Y' \mid Y' \in [0, 2^k) \wedge (Y' \bmod N) \in \mathbb{Z}_N^*\}$.

For $i \in \{0,1\}$, let $w_i = y^{d_i} \bmod N_i$, $s_i = [w_i]^{n+k_1}$, and $t_i = [w_i]_{k_0}$. Let r_i be the random variable $t_i \oplus H(s_i)$. We consider the following events.

- FBad denotes the event that
  - A G-oracle query r_0 was made by A_1 in step 3-1, and $G_{r_0} \neq s_0 \oplus (x||0^{k_1})$, or
  - A G-oracle query r_1 was made by A_1 in step 3-1, and $G_{r_1} \neq s_1 \oplus (x||0^{k_1})$.

- GBad denotes the event that
  - A G-oracle query r_0 was made by A_2 in step 3-2, and at the point in time that it was made, the H-oracle query s_0 was not on the H-list, and $G_{r_0} \neq s_0 \oplus (x||0^{k_1})$, or
  - A G-oracle query r_1 was made by A_2 in step 3-2, and at the point in time that it was made, the H-oracle query s_1 was not on the H-list, and $G_{r_1} \neq s_1 \oplus (x||0^{k_1})$.

- DBad denotes the event that
  - A $\mathcal{D}_{sk_0}$ query is not correctly answered, or
  - A $\mathcal{D}_{sk_1}$ query is not correctly answered.
- $\mathsf{G} = \neg\mathsf{FBad} \wedge \neg\mathsf{GBad} \wedge \neg\mathsf{DBad}$.

We let $\Pr[\cdot]$ denote the probability distribution in the game defining advantage, and $\Pr_0[\cdot]$ denote the probability distribution in the simulated game where M does not output Fail in the first step. We introduce the following additional events:

- YBad denotes the event that $y \in (\mathbb{Z}^*_{N_0} \cap \mathbb{Z}^*_{N_1})$.
- FAskS denotes the event that H-oracle query s_0 or s_1 was made by A_1 in step 3-1.
- AskR denotes the event that (r_0, G_{r_0}) or (r_1, G_{r_1}) is on the G-list at the end of step 3-2.
- AskS denotes the event that (s_0, H_{s_0}) or (s_1, H_{s_1}) is on the H-list at the end of step 3-2.

Now, we analyze the advantage of M. We can bound the advantage of M in a similar way as that for our scheme with expanding and we have

$$\begin{aligned}\mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{RSA},M}(k) &\geq \tfrac{1}{2q_{\mathrm{hash}}} \cdot \Pr[Y < 2^k \wedge \neg\mathsf{YBad}] \cdot \Pr_1[\mathsf{AskS}] \\ &\geq \tfrac{1}{2q_{\mathrm{hash}}} \cdot \Pr[Y < 2^k] \cdot \Pr_0[\neg\mathsf{YBad}] \cdot \Pr_1[\mathsf{AskS}]\end{aligned}$$

and

$$\Pr_1[\mathsf{AskS}] \geq \frac{\epsilon}{4} \cdot \left(1 - \left(\frac{2q_{\mathrm{gen}} + q_{\mathrm{dec}} + 2q_{\mathrm{gen}}q_{\mathrm{dec}}}{2^{k_0}} + \frac{2q_{\mathrm{dec}}}{2^{k_1}} + \frac{2q_{\mathrm{hash}}}{2^{k-k_0}}\right)\right) - \frac{q_{\mathrm{gen}}}{2^k}$$

where $\Pr_0[\cdot]$ denote the probability distribution in the simulated game where M does not output Fail in the first step, and $\Pr_1[\cdot]$ denote the probability distribution in the simulated game where M does not output Fail in the first step and $\neg$YBad occurs.

We next bound the probabilities that Y is in the good range and that $\neg$YBad occurs.

Lemma 3.5.

$$\Pr[Y \geq 2^k] \leq \frac{1}{2} \quad \textit{and} \quad \Pr_0[\mathsf{YBad}] \leq \frac{2}{2^{k/2-3} - 1}.$$

Proof of Lemma 3.5. We first bound $\Pr[Y \geq 2^k]$. Since $Y = y + \mu N$, $y \in \mathbb{Z}^*_N$, and $\mu \xleftarrow{R} \{0,1\}$, we have

$$\Pr[Y \geq 2^k] \leq \Pr[\mu = 1] = \frac{1}{2}.$$

We next bound $\mathrm{Pr}_0[\mathsf{YBad}]$. Let $N = pq$ and $N' = p'q'$. Note that $2^{\lceil k/2 \rceil - 1} < p, q, p', q' < 2^{\lceil k/2 \rceil}$ and $2^{k-1} < N, N' < 2^k$. Since $\phi(N) \leq |S[N]|$, we have

$$\begin{aligned}\mathrm{Pr}_0[\mathsf{YBad}] = \mathrm{Pr}[Y' \xleftarrow{R} S[N] : Y' \notin S[N']] &\leq \frac{|\{y \,|\, y \in S[N] \ \wedge \ y \notin S[N']\}|}{|S[N]|} \\ &\leq \frac{|\{y \,|\, y \in [0, 2^k) \ \wedge \ y \notin S[N']\}|}{|S[N]|} \\ &\leq \frac{2^k - |S[N']|}{|S[N]|} \leq \frac{2^k - |S[N']|}{\phi(N)}.\end{aligned}$$

Furthermore, we have

$$\begin{aligned}2^k - |S[N']| &= \left|\{Y' | Y' \in [0, 2^k) \wedge (Y' \bmod N') \notin \mathbb{Z}^*_{N'}\}\right| \\ &\leq \left|\{Y' | Y' \in [0, 2N') \wedge (Y' \bmod N') \notin \mathbb{Z}^*_{N'}\}\right| \\ &= 2 \times \left|\{Y' | Y' \in [0, N') \wedge Y' \notin \mathbb{Z}^*_{N'}\}\right| \\ &= 2(N' - \phi(N')).\end{aligned}$$

Therefore, we can bound $\mathrm{Pr}_0[\mathsf{YBad}]$ as

$$\begin{aligned}\mathrm{Pr}_0[\mathsf{YBad}] &\leq \frac{2^k - |S[N']|}{\phi(N)} \leq \frac{2(N' - \phi(N'))}{\phi(N)} = \frac{2(p' + q' - 1)}{N - p - q + 1} \leq \frac{2(p' + q')}{N - p - q} \\ &\leq \frac{2(2^{\lceil k/2 \rceil} + 2^{\lceil k/2 \rceil})}{2^{k-1} - 2^{\lceil k/2 \rceil} - 2^{\lceil k/2 \rceil}} = \frac{2(1+1)}{2^{k-1-\lceil k/2 \rceil} - 1 - 1} \leq \frac{4}{2^{k/2-2} - 2} = \frac{2}{2^{k/2-3} - 1}.\end{aligned}$$

□

Substituting the bounds for the above probabilities, we have

$$\mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{RSA},M}(k) \geq \frac{1}{2q_{\mathrm{hash}}} \cdot (1 - \epsilon_1) \cdot (1 - \epsilon_2) \cdot \left(\frac{\epsilon}{4} \cdot (1 - \epsilon_3) - \frac{q_{\mathrm{gen}}}{2^k}\right)$$

where $\epsilon_1 = \frac{1}{2}$, $\epsilon_2 = \frac{2}{2^{k/2-3}-1}$, and $\epsilon_3 = \frac{2q_{\mathrm{gen}} + q_{\mathrm{dec}} + 2q_{\mathrm{gen}}q_{\mathrm{dec}}}{2^{k_0}} + \frac{2q_{\mathrm{dec}}}{2^{k_1}} + \frac{2q_{\mathrm{hash}}}{2^{k-k_0}}$, and re-arranging the terms, we get the claimed result. Note that $\epsilon = \mathbf{Adv}^{\text{ik-cca}}_{\mathcal{PE},A}(k)$.

Finally, we estimate the time complexity of M. It is the time complexity of A plus the time for simulating the random oracles. In the random oracle simulation, for each pair $((g, G_g), (h, H_h))$, it is sufficient to compute $y_{h,g,0} = (h||(g \oplus H_h))^{e_0} \bmod N_0$ and $y_{h,g,1} = (h||(g \oplus H_h))^{e_1} \bmod N_1$. Therefore, the time complexity of M is that of A plus $q_{\mathrm{gen}} \cdot q_{\mathrm{hash}} \cdot O(k^3)$. □

We can also prove that the scheme with sampling twice is secure in the sense of IND-CCA in the random oracle model assuming RSA is θ-partial one-way for $\theta > 0.5$. More precisely, we can prove that if there exists a CCA-adversary $A = (A_1, A_2)$ attacking indistinguishability of our scheme with advantage ϵ, then there exists a CCA-adversary $B = (B_1, B_2)$ attacking indistinguishability of RSA-OAEP with advantage $\epsilon/2$. We construct B as follows.

	Repeating [3]	Expanding	RSACD	Sampling Twice
# of mod. exp. to encrypt (average / worst)	1.5 / k_1	1 / 1	1.5 / 2	2 / 2
# of mod. exp. to decrypt (average / worst)	1 / 1	1 / 1	1.5 / 2	1 / 1
size of ciphertexts	$k+1$	$k+160$	k	k
# of random bits to encrypt (average / worst)	$1.5k_0$ / k_1k_0	k_0+160 / k_0+160	$1.5k_0$ / $1.5k_0$	$2k_0+k+3$ / $2k_0+k+3$

Figure 3.1: The costs of the encryption schemes.

1) B_1 gets pk and passes it to A_1. B_1 gets (m_0, m_1, si) which is an output of A_1, and B_1 outputs it.

2) B_2 gets a challenge ciphertext y and sets $y' \leftarrow y + tN$ where $t \stackrel{R}{\leftarrow} \{0,1\}$. If $y' \geq 2^k$ then B_2 outputs Fail and halts; otherwise B_2 passes (y', si) to A_2. B_2 gets $d \in \{0,1\}$ which is an output of A_2, and B_2 outputs it.

If B does not output Fail, A outputs correctly with advantage ϵ. Since $\Pr[B \text{ outputs } \mathsf{Fail}] < 1/2$, the advantage of B is greater than $\epsilon/2$.

3.6 Efficiency

We show the number of modular exponentiations to encrypt, the number of modular exponentiations to decrypt, the size of ciphertexts, and the number of random bits to encrypt in Figure 3.1. We assume that N is uniformly distributed in $(2^{k-1}, 2^k)$.

CHAPTER 4

Anonymity on Undeniable and Confirmer Signature

In this chapter, we consider the undeniable and confirmer signature schemes with anonymity. In [44], Galbraith and Mao proposed a new RSA-based undeniable and confirmer signature scheme which provides the anonymity property. They constructed the scheme by using the expanding technique in order to prove that their scheme provides the anonymity property. In this chapter, we propose two undeniable and confirmer signature schemes, which are the variants of the Galbraith–Mao scheme, by using the repeating and the sampling twice techniques and prove their security.

The organization of this chapter is as follows. In Section 4.1, we review the definitions of undeniable and confirmer signature schemes, and the attacks on anonymity of undeniable and confirmer signature schemes proposed by Galbraith and Mao [44]. In Section 4.2 we review the undeniable and confirmer signature scheme, which provides the anonymity property, proposed by Galbraith and Mao. We propose a undeniable and confirmer signature scheme with the repeating technique in Section 4.3, and that with the sampling twice technique in Section 4.4. We compare the efficiency of three schemes in Section 4.5.

4.1 Definitions of Undeniable and Confirmer Signature

Digital signatures are easily verified as authentic by anyone using the corresponding public key. This property can be advantageous for many users, but it is unsuitable for many other users. Chaum and Antwerpen provided undeniable signature which cannot be veri-

fied without the signer's cooperation [22, 20]. The validity or invalidity of an undeniable signature can be ascertained by conducting a protocol with the signer, assuming the signer participates. Chaum provided confirmer signature [21] which is undeniable signature where signatures may also be verified by interacting with an entity called the confirmer who has been designated by the signer, and many undeniable and confirmer signature schemes were proposed [47, 63, 16, 45]. We describe the definition of undeniable and confirmer signature.

Definition 4.1. *An undeniable signature scheme* $\mathcal{SIG} = (\textsc{Cgen}, \textsc{Kgen}, \textsc{Sign}, \textsc{Conf}, \textsc{Deny})$ *consists of three algorithms and two protocols.*

- Cgen *is a (randomized) common-key generation algorithm that takes as input some security parameter* k *and returns a common key* I*. The signature space* $\mathcal{S}$ *is uniquely determined by* I*.*
- Kgen *is a (randomized) key generation algorithm that takes as input the common key* I *and returns a pair* (pk, sk) *of keys, the public key and a matching secret key. The message space* $\mathcal{M}_{pk}$ *for* pk *is uniquely determined by* pk*.*
- Sign *is a (randomized) signing algorithm that takes as input a secret key* sk *and a message* m *and outputs a signature* s*. Note that the signature space* $\mathcal{S}_{pk} := \{\textsc{Sign}_{sk}(m) \,|\, m \in \mathcal{M}_{pk}\}$ *for* pk *is a subset of* $\mathcal{S}$ *for any* (pk, sk)*.*
- Conf *is a confirmation protocol between a signer and a verifier which takes as input a message* m*, a signature* s*, and signer's public key* pk *and allows the signer to prove to a verifier that the signature* s *is valid for the message* m *and the key* pk*.*
- Deny *is a denial protocol between a signer and a verifier which takes as input a message* m*, a signature* s*, and signer's public key* pk *and allows the signer to prove to a verifier that the signature* s *is invalid for the message* m *and the key* pk*.*

A confirmer signature scheme is essentially the same as above, except the role of confirmation and denial can also be performed by a third party called a confirmer. The significant modification is that the key generation algorithm produces a confirmation key ck *which is needed for the confirmation or denial protocol.*

The literature on confirmer signature is inconsistent on whether the original signer has the ability to confirm and/or deny signatures. Camenisch and Michels [16] claim that it is undesirable for signers to be able to confirm or deny their signatures and the schemes in [16, 21, 63] do not allow signers to deny signatures. On the other hand, Galbraith and Mao claim that it is important for signers to be able to confirm and/or deny signatures and the schemes in [22, 20, 45, 47] do allow signers to deny signatures. In any case, these distinctions have no bearing on the discussion of the anonymity of the schemes.

Generalized Invisibility. Before describing the definition of the anonymity. We review the security notion called "invisibility." The notion of the invisibility was the strongest notion for undeniable and confirmer signature scheme, which was introduced by Chaum, van Heijst, and Pfitzmann [24]. This is essentially the inability to determine whether a given message-signature pair is valid for a given user. In [24], the invisibility is defined in terms of simulatability. In [17], this notion is phrased in terms of distinguishing whether a signature s corresponds to a message m_0 or m_1. Galbraith and Mao slightly modified the definition in [17], which they call "generalized invisibility." We review the definition of the generalized invisibility. Note that we slightly modify the definition of the generalized invisibility in [44] in order to put a common key generation into it explicitly.

Definition 4.2 (generalized invisibility [44])**.** *Let $\mathcal{SIG} = (\textsc{Cgen}, \textsc{Kgen}, \textsc{Sign}, \textsc{Conf}, \textsc{Deny})$ be an undeniable or confirmer signature scheme. Let $b \in \{0,1\}$ and $k \in \mathbb{N}$ (security parameter). Let $A = (A_1, A_2)$ be adversaries that run in two stages. A has access to the oracles $\textsc{Sign}_{sk}$ and A can execute confirmation and denial protocols $\textsc{Conf}_{sk}, \textsc{Deny}_{sk}$ on any message-signature pair. However, A_2 cannot execute either $\textsc{Conf}_{sk}$ or $\textsc{Deny}_{sk}$ on $(m', \sigma') \in EC(m, \sigma, pk)$. (EC means "equivalence class." If we get a message-signature pair (m, σ) under the key pk, then we can easily compute all elements in $EC(m, \sigma, pk)$. See also Remark 4.1.) Note that si be a state information. It contains common keys, public keys, and so on. Now we consider the following experiments:*

$$
\begin{array}{l}
\texttt{Experiment } \mathbf{Exp}^{\text{Anonym-}b}_{\mathcal{SIG},A}(k) \\
\quad I \leftarrow \textsc{Cgen}(1^k);\ (pk, sk) \leftarrow \textsc{Kgen}(I); \\
\quad (m, \mathsf{si}) \leftarrow A_1(pk); \\
\quad \texttt{if } (b = 0) \texttt{ then } \sigma \leftarrow \textsc{Sign}_{sk}(m) \\
\quad \texttt{if } (b = 1) \texttt{ then } \sigma \xleftarrow{R} \mathcal{S} \\
\quad d \leftarrow A_2(m, \sigma, \mathsf{si}) \\
\quad \texttt{return } d
\end{array}
$$

We define the advantages of the adversaries via:

$$\mathbf{Adv}^{\text{Anonym}}_{\mathcal{SIG},A}(k) = \Big|\Pr[\mathbf{Exp}^{\text{Anonym-1}}_{\mathcal{SIG},A}(k) = 1] - \Pr[\mathbf{Exp}^{\text{Anonym-0}}_{\mathcal{SIG},A}(k) = 1]\Big|.$$

The scheme $\mathcal{SIG}$ provides the generalized invisibility if the function $\mathbf{Adv}^{\text{Anonym}}_{\mathcal{SIG},A}(\cdot)$ is negligible for any adversary A whose time complexity is polynomial in k.

Galbraith and Mao showed that if the scheme meets the generalized invisibility, then the scheme also meets the invisibility in [17], and vice versa.

Anonymity. Galbraith and Mao proposed a new security notion of undeniable and confirmer signatures named "anonymity" in [44]. We say that an undeniable or confirmer

signature scheme provides anonymity when it is infeasible to determine which user generated the message-signature pair. Informally, this security property is as follows. Imagine a system with n users and suppose an adversary is given a valid message-signature pair and is asked to determine which user generated the signature. By running signature confirmation or denial protocols with a given user (or their designated confirmer) one can determine whether or not the user generated the signature. An undeniable or confirmer signature scheme has the anonymity property if it is infeasible to determine whether a user is or is not the signer of the message without interacting with that user or with the $n-1$ other users with given message-signature pair.

We slightly modify the definition of anonymity in [44] in order to put a common key generation into it explicitly.

Definition 4.3 (anonymity [44]). *Let $\mathcal{SIG} = (\textsc{Cgen}, \textsc{Kgen}, \textsc{Sign}, \textsc{Conf}, \textsc{Deny})$ be an undeniable or confirmer signature scheme. Let $b \in \{0,1\}$ and $k \in \mathbb{N}$ (security parameter). Let $A = (A_1, A_2)$ be adversaries that run in two stages. A has access to the oracles $\textsc{Sign}_{sk_0}, \textsc{Sign}_{sk_1}$ and A can execute confirmation and denial protocols $\textsc{Conf}_{sk_0}, \textsc{Conf}_{sk_1}$, $\textsc{Deny}_{sk_0}, \textsc{Deny}_{sk_1}$ on any message-signature pair. However, A_2 cannot execute any one of $\textsc{Conf}_{sk_0}, \textsc{Conf}_{sk_1}, \textsc{Deny}_{sk_0}$, and $\textsc{Deny}_{sk_1}$ on $(m', \sigma') \in EC(m, \sigma, pk_0) \cup EC(m, \sigma, pk_1)$ (EC means "equivalence class." If we get a message-signature pair (m, σ) under the key pk, then we can easily compute all elements in $EC(m, \sigma, pk)$.). Note that si be a state information. It contains common keys, public keys, and so on. Now we consider the following experiments:*

$$
\begin{array}{l}
\texttt{Experiment } \mathbf{Exp}^{\text{Anonym-}b}_{\mathcal{SIG},A}(k) \\
\quad I \leftarrow \textsc{Cgen}(1^k);\ (pk_0, sk_0) \leftarrow \textsc{Kgen}(I);\ (pk_1, sk_1) \leftarrow \textsc{Kgen}(I) \\
\quad (m, \mathsf{si}) \leftarrow A_1(pk_0, pk_1);\ \sigma \leftarrow \textsc{Sign}_{sk_b}(m);\ d \leftarrow A_2(m, \sigma, \mathsf{si}) \\
\quad \texttt{return } d
\end{array}
$$

We define the advantages of the adversaries via:

$$\mathbf{Adv}^{\text{Anonym}}_{\mathcal{SIG},A}(k) = \left|\Pr[\mathbf{Exp}^{\text{Anonym-1}}_{\mathcal{SIG},A}(k) = 1] - \Pr[\mathbf{Exp}^{\text{Anonym-0}}_{\mathcal{SIG},A}(k) = 1]\right|.$$

The scheme $\mathcal{SIG}$ provides anonymity if the function $\mathbf{Adv}^{\text{Anonym}}_{\mathcal{SIG},A}(\cdot)$ is negligible for any adversary A whose time complexity is polynomial in k.

Galbraith and Mao proved the following proposition.

Proposition 4.1. *If the scheme meets the generalized invisibility, then the scheme also meets the anonymity, and vice versa.*

Attacks on Anonymity In [47], Gennaro, Krawczyk and Rabin described an undeniable/confirmer signature scheme based on RSA. In their case the signature for a message m is s where $s = \bar{m}^d \bmod N$ and $\bar{m}$ is a one-way encoding. The signature may be verified by proving that $s^e = \bar{m} \pmod N$ where the verification exponent e is known to the signer/confirmer. This scheme requires that the moduli be products of safe primes. Later the scheme was generalized to use arbitrary RSA moduli [45]. To handle adaptive attacks on anonymity it is clear that the one-way encoding must also be randomized. Hence, a signature becomes a pair (r, s) where r is random and $s = H(m, r)^d \pmod N$ where $H(m, r)$ is the randomized one-way encoding (such as PSS [8]).

In [44], Galbraith and Mao pointed out the Gennaro–Krawczyk–Rabin scheme does not provide anonymity. They showed the following attacks:

Jacobi Symbols Attack Since d is odd it follows that the Jacobi symbols $\left(\frac{s}{N}\right)$ and $\left(\frac{H(m,r)}{N}\right)$ are equal. Hence, given a pair $(H(m, r), s)$ and a user's public key N, if $\left(\frac{s}{N}\right) \neq \left(\frac{H(m,r)}{N}\right)$then the signature is not valid for that user. This shows that the scheme does not have anonymity.

Signature Length Attack A simple observation that seems to be folklore is that standard RSA signature does not provide anonymity, even when all moduli in the system have the same length. Suppose an adversary knows that the signature s is created under one of two keys (N_0, d_0) or (N_1, d_1) (length of N_0 and N_1 are k), and suppose $N_0 \leq N_1$. If $s \geq N_0$ then the adversary knows it was created under (N_1, d_1).

4.2 Undeniable and Confirmer Signature with Expanding by Galbraith and Mao

In [44], Galbraith and Mao proposed a new RSA-based scheme. In this section, we review their scheme.

Definition 4.4 ([44]). *The common-key generation algorithm* CGEN *takes a security parameter k and returns parameters k, k_0 and k_1 such that $k_0(k)+k_1(k) < k$ for all $k > 1$. The key generation algorithm* KGEN *takes k, k_0, k_1, runs the key-generation algorithm of* RSA*, and gets N, e, d, p, q where p, q the safe prime (i.e. $(p-1)/2$ and $(q-1)/2$ are also prime). It picks g from $\mathbb{Z}_N^*$ and sets $h \leftarrow g^d \bmod N$. The public key pk is $(N, g, h), k, k_0, k_1$ and the secret key sk is (N, e, d, p, q), k, k_0, k_1. The signature space is $\mathcal{S} = \{0,1\}^{2k} \times \{0,1\}^{k_0}$. Let $G_0 : \{0,1\}^* \to \{0,1\}^{k_1}$, $G_1 : \{0,1\}^{k_1} \to \{0,1\}^{k_0}$, $G_2 : \{0,1\}^{k_1} \to \{0,1\}^{k-k_0-k_1-1}$, and F*

$: \{0,1\}^k \to \{0,1\}^k$ be hash functions.

	$\textsc{Sign}(m)$	$\textsc{Sign2}(m,r)$
		$w \leftarrow G_0(m\|\|r)$
1	$r \xleftarrow{R} \{0,1\}^{k_0}$	$r^* \leftarrow G_1(w) \oplus r$
2	$\bar{m} \leftarrow \textsc{Sign2}(m,r)$	$M \leftarrow w\|\|r^*\|\|G_2(w)$
3	$t \xleftarrow{R} \{c \in \mathbb{Z}_N \mid c^2 = \pm\bar{m} \pmod N\}$	$\bar{m} \leftarrow M$
4	$s \leftarrow t^d \bmod N$	`while` $\left(\left(\frac{\bar{m}}{N}\right) \neq 1\right)$ `repeat` $\bar{m} \leftarrow F(\bar{m})$
5	$u \xleftarrow{R} \{0,1,\ldots,\lfloor(2^{2k}-s)/N\rfloor\}$	`return` $\bar{m}$
6	$\hat{s} \leftarrow s + uN$	
7	`return` $(\hat{s}, r)$	

Conf *(respectively* Deny*) is a non-interactive designated verifier proof which proves the knowledge of an integer e such that $g = h^e \pmod N$ and $\hat{s}^{2e} = \pm\textsc{Sign2}(m,r) \pmod N$ (resp. $g = h^e \pmod N$ and $\hat{s}^{2e} \neq \pm\textsc{Sign2}(m,r) \pmod N$). Note that $\hat{s} = s + uN = s \pmod N$ and all users can compute $\textsc{Sign2}(m,r)$ given m, r, and N.*

Remark 4.1. *It is clear that if a message-signature pair $(m,(\hat{s},r))$ is valid for $pk = (N,g,h)$ then $(m,(\pm s \pm uN, r))$ is also valid where $s = \hat{s} \bmod N$ and $u \in \{0,1,\ldots,\lfloor(2^{2k}-s)/N\rfloor\}$. Thus, Galbraith and Mao defined the equivalence class for their scheme as*

$$EC(m,(\hat{s},r),pk) = \{(m,(\pm s \pm uN, r)) | s = \hat{s} \bmod N,\ u \in \{0,1,\ldots,\lfloor(2^{2k}-s)/N\rfloor\}\}.$$

Since using a Blum integer N, for every $\bar{m} \in \mathbb{Z}_N^*$ with $\left(\frac{\bar{m}}{N}\right) = 1$, it follows that either $\bar{m}$ or $-\bar{m}$ is a square. One can compute square-root and randomly chooses t from four possibilities in step 3. Since $\left(\frac{t}{N}\right)$ is not fixed, their scheme prevents the Jacobi symbols attack. In step 5 and 6, it extends signatures of length k to be bit-strings of length $2k$. Since $0 \leq \hat{s} < 2^{2k}$ and $\hat{s}$ is indistinguishable from a random $2k$-bit string for any N whose length is k, their scheme prevents the signature length attack (See also [35].).

Galbraith and Mao proved that their scheme provides the generalized invisiblity and the anonymity in the random oracle model under the assumption that the composite decision Diffie-Hellman problem is hard.

Definition 4.5 (composite decision Diffie-Hellman problem)**.** *Let N be a product of two safe primes (i.e. $N = pq$ where p, q, $(p-1)/2$, $(q-1)/2$ are primes). Consider the two sets*

$$\mathcal{T} = \{(g,h,u,v) \in (\mathbb{Z}_N^*)^4 \mid \mathrm{ord}(g) = \mathrm{ord}(h) = 2p'q',\ h \in \langle g\rangle,\ \langle g, v\rangle = \mathbb{Z}_N^*\}$$

and

$$\mathcal{T}_{\mathrm{CDDH}} = \{(g,h,u,v) \in \mathcal{T} \mid h = g^d \pmod N \text{ for some } d \text{ coprime to } \phi(N),\ v = \alpha u^d \pmod N \text{ for some } \alpha \in \mathbb{Z}_N^* \text{ of order } 2\}$$

with the uniform distribution on each. We say that the composite decision Diffie-Hellman problem is hard if it is infeasible to distinguish these two distributions.

To obtain the security result it is necessary that executions of the confirm and deny protocol can be simulated in the random oracle model. This is not possible with interactive proofs so we must use non-interactive proofs. To maintain the security of the system, it is necessary to use non-interactive designated verifier proofs [53].

They also proved that their scheme is existential unforgeable in the random oracle model under the assumption that factoring integers which are products of safe primes is hard.

In the scheme by Galbraith and Mao (and also our schemes we will propose later on), we have to use RSA moduli which are the products of safe primes for obtaining the anonymity property. Gennaro, Krawczyk, and Rabin [47] proposed the RSA-based undeniable signature schemes where RSA moduli are restricted to the products of safe primes, and the confirmation and denial protocols in [47] is more efficient than those by Galbraith, Mao, and Paterson [45]. Therefore, it seems better to use the protocols in [47]. However, if we use the protocols in [47], the prover will have to prove that her RSA modulo has the proper form (i.e. a product of safe primes) during the protocols, and it needs a costly proof. To avoid this, Galbraith, Mao, and Paterson [45] constructed different scheme where there is no restriction for the RSA moduli.

4.3 Undeniable and Confirmer Signature with Repeating

In this section, we propose the undeniable and confirmer signature schemes with the repeating technique.

Definition 4.6. *The common-key generation algorithm* CGEN*, the key generation algorithm* KGEN*, and hash functions G_0, G_1, G_2, F are the same as those for the Galbraith–Mao scheme. The signature space is $\mathcal{S} = \{0,1\}^{k-1} \times \{0,1\}^{k_0}$. The signing algorithm is as follows.*

SIGN(m)
  $ctr \leftarrow -1$
  `repeat`
    $ctr \leftarrow ctr + 1$
    $r \xleftarrow{R} \{0,1\}^{k_0}$; $\bar{m} \leftarrow$ SIGN2(m, r)
    $t \xleftarrow{R} \{c \in [0, N) \mid c^2 = \pm\bar{m} \pmod N\}$; $s \leftarrow t^d \bmod N$
  `until` $(s < 2^{k-1} \vee ctr = k_1)$
  `return` (s, r)

CONF *or* DENY *executes non-interactive designated verifier proofs which prove knowledge of an integer* e *such that* $g = h^e \pmod N$ *and* $s^{2e} \stackrel{?}{=} \pm\text{SIGN2}(m, r) \pmod N$. *In order to construct such proofs, we first employ protocols similar to those in [45] by Galbraith, Mao, and Paterson. Then, we transform them to corresponding non-interactive designated verifier proofs by the method of Jakobsson, Sako, and Impagliazzo [53].*

We now prove the security of the scheme with repeating.

Lemma 4.1. *If the scheme with expanding proposed by Galbraith and Mao provides the generalized invisibility, then the scheme with repeating also provides the generalized invisibility.*

Proof. Suppose that we have an adversary $A = (A_1, A_2)$ attacking the generalized invisibility of the scheme with repeating. We construct the algorithm $B = (B_1, B_2)$ attacking the generalized invisibility of the scheme by Galbraith and Mao, using the algorithm A. Note that B simulates A's oracles as described below.

1) B_1 takes pk and passes it to A_1. B_1 gets (m, si) which is an output of A_1, and B_1 outputs it.

2) B_2 gets a challenge pair $(\hat{s}, r) \in \{0,1\}^{2k} \times \{0,1\}^{k_0}$. Then, B_2 computes $s \leftarrow \hat{s} \bmod N$, and if $s > 2^{k-1}$ then it outputs `fail` and halts.

3) B_2 passes $((s, r), \mathsf{si})$ to A_2. B_2 gets $d \in \{0,1\}$ which is an output of A_2, and outputs it.

B simulates the oracles as follows.

Hash query B uses its random oracles G_0, G_1, G_2, F to answer the query by A.

Signing query To answer the signing query m by A, B uses its signing oracle and gets $(\hat{s}, r)$. Then, B computes $s \leftarrow \hat{s} \bmod N$, and if $s \leq 2^{k-1}$ then it answers (s, r) to A. Otherwise B makes a query to its signing oracle again, and repeats the procedure described above until $s \leq 2^{k-1}$.

Confirmation and Denial query B uses its confirmation and denial oracles, and returns the results to A. (Note that our confirmation and denial protocols of the scheme with repeating and those of the Galbraith–Mao scheme are non-interactive one.)

It is easy to see that the probability that B outputs `fail` (when B makes a challenge) is non-negligible. Furthermore, the distribution of the challenge (s, r) in the above game is indistinguishable from that in the real game for the Galbraith–Mao scheme. Therefore, the advantage of B is non-negligible if that of A is non-negligible. □

From Lemma 4.1, Proposition 4.1, and the result by Galbraith and Mao with respect to the security of their scheme, we have the following theorem.

Theorem 4.1. *The scheme with repeating provides the generalized invisibility and the anonymity in the random oracle model under the assumption that the composite decision Diffie-Hellman problem is hard.*

We next prove the following theorem.

Theorem 4.2. *The scheme with repeating is existential unforgeable in the random oracle model under the assumption that factoring integers which are products of safe primes is hard.*

Proof. Suppose that we have an adversary A attacking the unforgeability of the scheme with repeating. Then, we can construct the algorithm B attacking the unforgeability of the scheme by Galbraith and Mao, using the algorithm A. The algorithm B runs A with simulating A' oracles. B can simulate the oracles for A in a similar way as those in the previous proof. It is easy to see that if A outputs a valid signature of the scheme with repeating, then the signature is also a valid signature of the Galbraith–Mao scheme. Therefore, if A forges the signature of the scheme with repeating with non-negligible probability, then B can forge the signature of the Galbraith–Mao scheme with non-negligible probability. □

4.4 Undeniable and Confirmer Signature with Sampling Twice

In this section, we propose the undeniable and confirmer signature scheme with the sampling twice technique.

Definition 4.7. *The common-key generation algorithm* CGEN*, the key generation algorithm* KGEN*, and hash functions G_0, G_1, G_2, F are the same as those for the Galbraith–Mao scheme. The signature space is $\mathcal{S} = \{0,1\}^k \times \{0,1\}^{k_0}$. The signing algorithm is as follows.*

SIGN(m)
$r_1, r_2 \xleftarrow{R} \{0,1\}^{k_0}$
$\bar{m_1} \leftarrow$ SIGN2(m, r_1); $t_1 \xleftarrow{R} \{c \in \mathbb{Z}_N \mid c^2 = \pm\bar{m_1} \pmod N\}$; $s_1 \leftarrow (t_1)^d \bmod N$
$\bar{m_2} \leftarrow$ SIGN2(m, r_2); $t_2 \xleftarrow{R} \{c \in \mathbb{Z}_N \mid c^2 = \pm\bar{m_2} \pmod N\}$; $s_2 \leftarrow (t_2)^d \bmod N$
$s \leftarrow$ `ChooseAndShift`(s_1, s_2)
`if` $(s \bmod N = s_1)$ $r \leftarrow r_1$ `else` $r \leftarrow r_2$
`return` (s, r)

CONF *(respectively* DENY*) is a non-interactive designated verifier proof which proves the knowledge of an integer e such that $g = h^e \pmod N$ and $s^{2e} = \pm$*SIGN2$(m, r) \pmod N$

	Repeating	Expanding [44]	Sampling Twice
# of mod. exp. to sign (average / worst)	1.5 / k_1	1 / 1	2 / 2
# of computation of square roots (average / worst)	1.5 / k_1	1 / 1	2 / 2
size of signatures	$(k-1)+k_0$	$2k+k_0$	$k+k_0$
# of random bits to sign (average / worst)	$1.5(k_0+2)$ / $k_1(k_0+2)$	k_0+k+2 / k_0+k+2	k_0+k+5 / k_0+k+5

Figure 4.1: The costs of the undeniable and confirmer signature schemes.

(resp. $g = h^e \pmod N$ *and* $s^{2e} \neq \pm\textsc{Sign2}(m,r) \pmod N$*). In order to construct such proofs, we first employ protocols similar to those in [45] by Galbraith, Mao, and Paterson. Then, we transform them to corresponding non-interactive designated verifier proofs by the method of Jakobsson, Sako, and Impagliazzo [53]. The equivalence class of this scheme is* $EC(m,(s,r),pk) = \{(m,(\pm s' \pm uN, r)) \mid s' = s \bmod N \wedge u \in \{0,1,2,\ldots,\lfloor(2^k - s')/N\rfloor\}\}$.

We can prove that the scheme with sampling twice provides the generalized invisiblity and the anonymity in the random oracle model under the assumption that the composite decision Diffie-Hellman problem is hard, and is existential unforgeable in the random oracle model under the assumption that factoring integers which are products of safe primes is hard. The proofs are similar to those for the scheme with repeating.

4.5 Efficiency

We show the number of modular exponentiations to sign, the number of computation of square root, the size of signatures, and the number of random bits to sign in Figure 4.1. We assume that N is uniformly distributed in $(2^{k-1}, 2^k)$.

CHAPTER 5

Anonymity on Ring Signature

In this chapter, we consider the ring signature schemes with anonymity. In [76], Rivest, Shamir, and Tauman proposed the notion of ring signature, which allows a member of an ad hoc collection of users S to prove that a message is authenticated by a member of S without revealing which member actually produced the signature. They constructed the scheme which provides the above property by using the expanding technique. In this chapter, we propose three ring signature schemes, which are variants of the Rivest–Shamir–Tauman scheme, with the repeating technique, RSACD, and the sampling twice technique, and prove their security.

The organization of this chapter is as follows. We review the definitions of ring signature in Section 5.1, and the RSA-based ring signature scheme proposed by Rivest, Shamir, and Tauman in Section 5.2. We propose a ring signature scheme with the repeating technique in Section 5.3, that with RSACD in Section 5.4, and that with the sampling twice technique in Section 5.5. We compare the efficiency of four schemes in Section 5.6.

5.1 Definitions of Ring Signature

In [76], Rivest, Shamir, and Tauman proposed the notion of ring signature, which allows a member of an ad hoc collection of users S to prove that a message is authenticated by a member of S without revealing which member actually produced the signature. Unlike group signature, ring signature has no group managers, no setup procedures, no revocation

procedures, and no coordination.

Definition 5.1 (ring signature [76]). *One assumes that each user (called a ring member) has received (via a PKI or a certificate) a public key P_k, for which the corresponding secret key is denoted by S_k. A ring signature scheme consists of the following algorithms.*

- **ring-sign**$(m, P_1, P_2, \cdots, P_r, s, S_s)$ *which produces a ring signature σ for the message m, given the public keys $P_1, P_2, \cdots, P_r$ of the r ring members, together with the secret key S_s of the s-th member (who is the actual signer).*
- **ring-verify**(m, σ) *which accepts a message m and a signature σ (which includes the public key of all the possible signers), and outputs either "*`valid`*" or "*`invalid`*".*

The signer does not need the knowledge, consent, or assistance of the other ring members to put them in the ring. All he needs is knowledge of their regular public keys. Verification must satisfy the usual soundness and completeness conditions, but in addition the signature scheme must satisfy "signer-ambiguous", which is the property that the verifier should be unable to determine the identity of the actual signer with probability greater than $1/r + \epsilon$, where r is the size of the ring, and ϵ is negligible.

Furthermore, the signature scheme must satisfy "existential unforgeability under adaptive chosen message attack", which is the property that any polynomial time adversary, where she can have access to the signing oracle and get signatures for any message, cannot forge a message-signature pair with non-negligible probability, other than the pairs the signing oracle has previously produced.

5.2 RSA-based Ring Signature Scheme by Rivest, Shamir, and Tauman

In [76], Rivest, Shamir, and Tauman constructed the ring signature scheme in which all the ring member use RSA as their individual signature schemes. Each user can uses the RSA moduli whose lengths are different from other users.

The formal concept of ring signature can be related to an abstract concept called *combining functions*. A combining function $C_{k,v}(y_1, y_2, \cdots, y_r)$ takes as input a key k, an initialization value v, and a list of arbitrary values of the same length ℓ. It outputs a single value $z \in \{0,1\}^\ell$ such that for any k, v, any index s, and any fixed values of $\{y_i\}_{i \neq s}$, $C_{k,v}$ is a permutation over $\{0,1\}^\ell$, when seen as a function of y_s. Moreover, this permutation is efficiently computable as well as its inverse.

In [76], Rivest, Shamir, and Tauman proposed a combining function based on a symmetric encryption scheme E modeled by a (keyed) random permutation

$$C_{k,v}(y_1, \cdots, y_r) = E_k(y_r \oplus E_k(y_{r-1} \oplus \cdots E_k(y_2 \oplus E_k(y_1 \oplus v)) \cdots)). \tag{5.1}$$

For any index s, we can easily verify that $C_{k,v}$ is a combining function by rewriting equation 5.1 as follows:

$$y_s = E_k^{-1}\big(y_{s+1} \oplus \cdots E_k^{-1}(y_r \oplus E_k^{-1}(v))\cdots\big) \oplus E_k\big(y_{s-1} \oplus \cdots E_k(y_1 \oplus v)\cdots\big). \tag{5.2}$$

By using the combining function, Rivest, Shamir, and Tauman proposed the RSA-based ring signature scheme.

Definition 5.2 ([76]). *Let ℓ, k, and b be security parameters. Let E be a symmetric encryption scheme over $\{0,1\}^b$ using ℓ-bit keys and h be a hash function which maps arbitrary strings to ℓ-bit strings. They use h to make a key for E. Each user has an RSA public key $P_i = (N_i, e_i, k_i)$ and secret key $S_i = (N_i, d_i, k_i)$ where $k_i \geq k$ by running the key generation algorithm of* RSA. *Let r be a number of ring member. We define the extended trap-door permutation g_i over $\{0,1\}^b$ as follows: for any b-bits input x_i define nonnegative integers q_i and r_i such that $x_i = q_i N_i + r_i$ and $0 \leq r_i < N_i$. Then*

$$g_i(x_i) = \begin{cases} q_i N_i + f^{\mathsf{RSA}}_{N_i,e_i,k_i}(r_i) & \texttt{if } (q_i+1)N_i \leq 2^b \\ x_i & \texttt{otherwise.} \end{cases}$$

The signing algorithm is as follows.

ring-sign$(m, P_1, P_2, \cdots, P_r, s, S_s)$
  `for each` $i \in \{1, \cdots, s-1, s+1, \cdots, r\}$ `do`
    $x_i \xleftarrow{R} \{0,1\}^b$; $y_i \leftarrow g_i(x_i)$
  $v \xleftarrow{R} \{0,1\}^b$
  `find` y_s `s.t.` $C_{h(m),v}(y_1, \cdots, y_r) = v$
  $x_s \leftarrow g_s^{-1}(y_s)$
  `return` $\sigma = (P_1, P_2, \cdots, P_r, v, x_1, x_2, \cdots, x_r)$

Note that we can find y_s such that $C_{h(m),v}(y_1, \cdots, y_r) = v$ in the signing algorithm by using equation 5.2 (See also Figure 5.1.).
The verification algorithm **ring-verify**(m, σ) *computes $y_i \leftarrow g_i(x_i)$ for each x_i and $z \leftarrow C_{h(m),v}(y_1, \cdots, y_r)$. It returns* `valid` *if $z = v$ (See Figure 5.2.).*

If b is sufficiently large (e.g. 160 bits larger than any of the N_i), g_i is a one-way trap-door permutation, and Rivest, Shamir, and Tauman proved this scheme is unconditionally signer-ambiguous and existential unforgeable under adaptive chosen message attack in the ideal cipher model assuming RSA is one-way.

Remark 5.1 (ideal cipher model). *The ideal cipher model provides a mathematical model of an "ideal" symmetric encryption scheme. In this model, a function (a symmetric encryption scheme) $h_k : X \times K \to X$ is chosen randomly from $\mathcal{P}^{X,K}$ which is a set of functions*

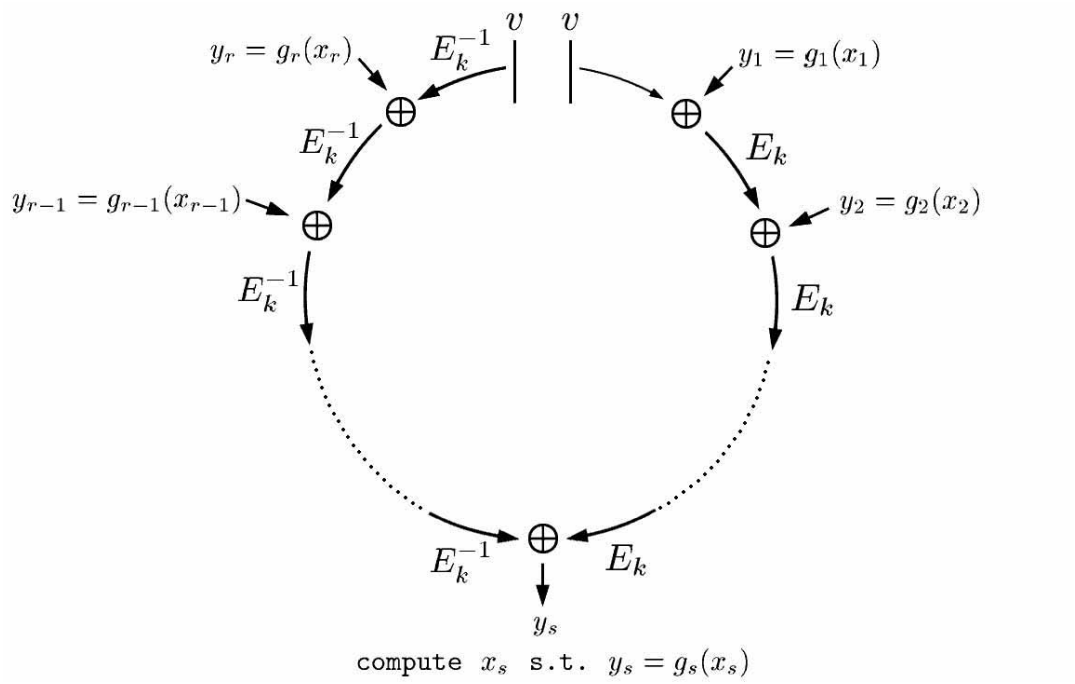

Figure 5.1: The construction of ring signature.

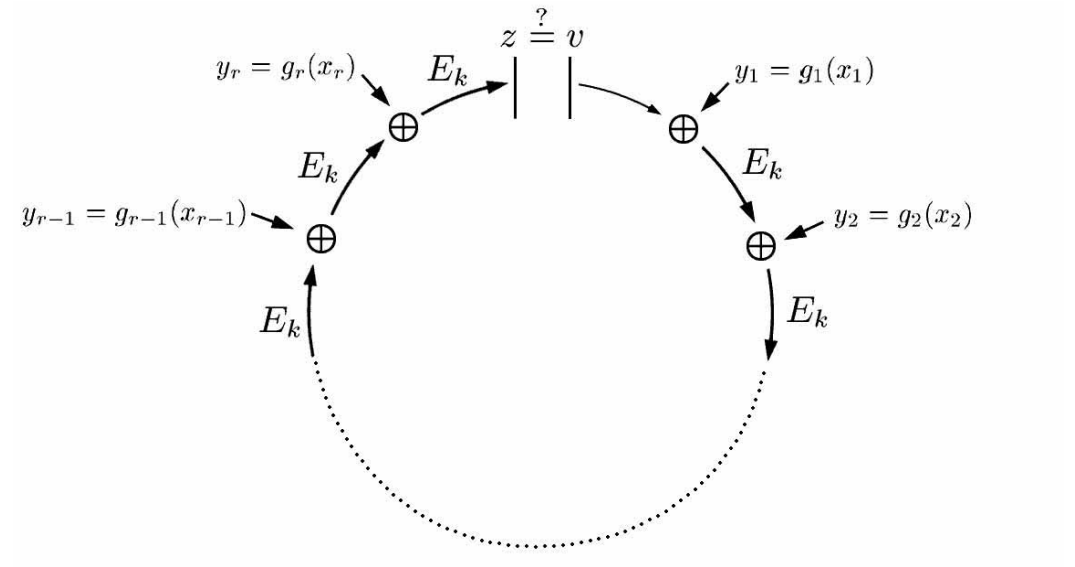

Figure 5.2: The verification of ring signature.

such that for any fixed $k \in K$, h_k is a permutation over X, and we are only permitted oracle access to the function h_k for any $k \in K$. This means that we are not given a formula or an algorithm to compute values of the function h_k for any $k \in K$. Therefore, the only way to compute the value $h_k(x)$ is to query the oracle. It should be noticed that the ideal cipher model is considered to be stronger than the random oracle model.

Bresson, Stern, and Szydlo [14] recently improved the ring signature scheme proposed by Rivest, Shamir, and Tauman. They showed that security can be based on the random oracle model which is strictly weaker complexity assumption than the ideal cipher model. Furthermore, this greatly simplified the security proof provided in [76]. They also provided the threshold ring signature scheme and its applications. For examples of applications of ring signatures, Naor [64] proposed the deniable ring authentication scheme. It is possible to convince a verifier that a member of an ad hoc subset of participants is authenticating a message without revealing which one (source hiding), and the verifier cannot convince a third party that message was indeed authenticated. Zhang and Kim [81] proposed the ID-based ring signature scheme which is based on the bilinear pairings and they also analyzed its security and efficiency.

5.3 Ring Signature with Repeating

In this section, we propose the ring signature scheme by using the repeating technique.

Definition 5.3. *Let ℓ, k, and $b = k - 1$ be security parameters. Let E be a symmetric encryption scheme over $\{0,1\}^b$ using ℓ-bit keys and h a hash function which maps arbitrary strings to ℓ-bit strings. Each user has an RSA public key $P_i = (N_i, e_i, k_i)$ and secret key $S_i = (N_i, d_i, k_i)$ by running the key generation algorithm of* RSA *with security parameter k (i.e. the size of N_i is k). Let r be a number of ring member. The signing algorithm is as*

follows.

ring-sign$(m, P_1, P_2, \cdots, P_r, s, S_s)$
  `for each` $i \in \{1, \cdots, s-1, s+1, \cdots, r\}$ `do`
    $ctr \leftarrow -1$
    `repeat`
      $ctr \leftarrow ctr + 1$
      $x_i \xleftarrow{R} \mathbb{Z}_{N_i}$; $y_i \leftarrow (x_i)^{e_i} \bmod N_i$
    `until` $(y_i < 2^{k-1} \vee ctr = k)$
    `if` $(ctr = k)$ $x_i \leftarrow 1$; $y_i \leftarrow 1$
  $v \xleftarrow{R} \{0,1\}^b$
  `find` y_s `s.t.` $C_{h(m),v}(y_1, \cdots, y_r) = v$
  $x_s \leftarrow y_s^{d_s} \bmod N_s$
  `return` $\sigma = (P_1, P_2, \cdots, P_r, v, x_1, x_2, \cdots, x_r)$

ring-verify(m, σ) *computes* $y_i \leftarrow x_i^{e_i} \bmod N_i$ *for each* x_i *and* $z \leftarrow C_{h(m),v}(y_1, \cdots, y_r)$. *It returns* `valid` *if* $z = v$.

We can prove that our scheme is unconditionally signer-ambiguous, since for each k and v the equation $C_{h(m),v}(y_1, \cdots, y_r) = v$ has exactly $(2^{k-1})^{r-1}$ solutions, and all of them can be chosen by the signature generation procedure with equal probability, regardless of the signer's identity.

We can also prove that our scheme is existential unforgeable under adaptive chosen message attack in the ideal cipher model assuming RSA is one-way. The proof is almost the same as that for the Rivest–Shamir–Tauman scheme. The difference is as follows.

In the proof of unforgeability for the Rivest–Shamir–Tauman scheme, given $y \in \mathbb{Z}_N^*$, one slips y as a "gap" between two consecutive E functions along the ring. Then, the forger has to compute the e-th root of y, and this leads one to obtain the e-th root of y.

In the proof for our scheme, given $y \in \mathbb{Z}_N^*$, if $y \geq 2^{k-1}$ then outputs Fail and halts. Otherwise, one slips y as a "gap" between two consecutive E functions along the ring. We can easily see that the probability of outputting Fail is smaller than $1/2$. The rest of the proof is the same as that for the Rivest–Shamir–Tauman scheme (See Section 3.5 in [76].), and it is not hard to see that if there exists a forger for our scheme with advantage ϵ, then we can invert RSA with probability $\epsilon/2q^2$ where q is a number of oracle queries.

5.4 Ring Signature with RSACD

In this section, we propose a ring signature scheme with the RSACD function. We use $f^{\mathsf{RSACD}}_{N_i,e_i,k}(\cdot)$ instead of $g_i(\cdot)$.

Definition 5.4. *The values ℓ, k, E, h, r are the same as those of Rivest–Shamir–Tauman scheme. Each user has a public key $P_i = (N_i, e_i, k)$ and secret key $S_i = (N_i, d_i, k)$ by running the key generation algorithm of* RSACD *with security parameter k (i.e. the length of N_i is k), and let $b = k$. The signing algorithm is as follows.*

$$
\begin{aligned}
&\textbf{ring-sign}(m, P_1, P_2, \cdots, P_r, s, S_s)\\
&\quad \texttt{for each } i \in \{1, \cdots, s-1, s+1, \cdots, r\} \texttt{ do}\\
&\qquad x_i \xleftarrow{R} \{0,1\}^k;\ y_i \leftarrow f^{\mathsf{RSACD}}_{N_i,e_i,k}(x_i)\\
&\quad v \xleftarrow{R} \{0,1\}^k\\
&\quad \texttt{find } y_s \texttt{ s.t. } C_{h(m),v}(y_1, \cdots, y_r) = v\\
&\quad x_s \leftarrow g^{\mathsf{RSACD}}_{N_s,d_s,k}(y_s)\\
&\quad \texttt{return } \sigma = (P_1, P_2, \cdots, P_r, v, x_1, x_2, \cdots, x_r)
\end{aligned}
$$

The verification algorithm **ring-verify**(m, σ) *computes $y_i \leftarrow f^{\mathsf{RSACD}}_{N_i,e_i,k}(x_i)$ for each x_i and $z \leftarrow C_{h(m),v}(y_1, \cdots, y_r)$. It returns* `valid` *if $z = v$.*

We can prove that the scheme with RSACD is unconditionally signer-ambiguous, since for each k and v the equation $C_{h(m),v}(y_1, \ldots, y_r) = v$ has exactly $(2^k)^{r-1}$ solutions, and all of them are chosen by the signature generation procedure with equal probability, regardless of the signer's identity.

We can also prove that the scheme with RSACD is existential unforgeable under adaptive chosen message attack in the ideal cipher model assuming RSA is one-way. The proof is almost the same as that for the Rivest–Shamir–Tauman scheme. The difference is as follows.

In the proof of unforgeability for the Rivest–Shamir–Tauman scheme, given $y \in \mathbb{Z}_N^*$, one slips y as a "gap" between two consecutive E functions along the ring. Then, the forger has to compute the e-th root of y, and this leads one to obtain the e-th root of y.

In the proof for the scheme with RSACD, given $y' \in \mathrm{Rng}_{\mathsf{RSACD}}(N, e, k)$, one slips y' as a "gap" between two consecutive E functions along the ring. The rest of the proof is the same as that for the Rivest–Shamir–Tauman scheme

5.5 Ring Signature with Sampling Twice

In this section, we propose a ring signature scheme with the sampling twice technique. To verify the signatures deterministically, we add some information c_i to the signature.

Definition 5.5. *Let ℓ, k be security parameters. Let E be a symmetric encryption scheme over $\{0,1\}^k$ using ℓ-bit keys, and let h be a hash function which maps strings of arbitrary length to ℓ-bit strings. Each user U_i has public key $P_i = (N_i, e_i, k)$ and secret key $S_i =$*

(N_i, d_i, k) by running the key generation algorithm of RSA *with security parameter k (i.e. the size of N_i is k). Let r be the number of ring members. The signing algorithm is as follows.*

ring-sign$(m, P_1, P_2, \ldots, P_r, s, S_s)$
  `for each` $i \in \{1, \ldots, s-1, s+1, \ldots, r\}$ `do`
    $x_{i,1}, x_{i,2} \xleftarrow{R} \mathbb{Z}^*_{N_i}$
    $y_{i,1} \leftarrow (x_{i,1})^{e_i} \bmod N_i$; $y_{i,2} \leftarrow (x_{i,2})^{e_i} \bmod N_i$
    $y_i \leftarrow$ `ChooseAndShift`$(y_{i,1}, y_{i,2})$
    `if` $(y_i \bmod N_i = y_{i,1})$ $x_i \leftarrow x_{i,1}$ `else` $x_i \leftarrow x_{i,2}$
    `if` $(y_i \geq N_i)$ $c_i \leftarrow 1$ `else` $c_i \leftarrow 0$
  $v \xleftarrow{R} \{0,1\}^k$
  `find` y_s `s.t.` $C_{h(m),v}(y_1, \ldots, y_r) = v$
  `if` $(y_s \geq N_s)$ $c_s \leftarrow 1$ `else` $c_s \leftarrow 0$
  $x_s \leftarrow (y_s)^{d_s} \bmod N_s$
  `return` $\sigma = (P_1, P_2, \ldots, P_r, v, (x_1, c_1), (x_2, c_2), \ldots, (x_r, c_r))$

The verification algorithm **ring-verify**(m, σ) *computes $y_i \leftarrow ((x_i)^{e_i} \bmod N_i) + c_i \cdot N_i$ for each (x_i, c_i) and $z \leftarrow C_{h(m),v}(y_1, \ldots, y_r)$. It returns* `valid` *if and only if $z = v$.*

We can prove that the scheme with the sampling twice technique is unconditionally signer-ambiguous, since for each k and v the equation $C_{h(m),v}(y_1, \ldots, y_r) = v$ has exactly $(2^k)^{r-1}$ solutions, and all of them are chosen by the signature generation procedure with equal probability, regardless of the signer's identity.

We can also prove that the scheme with the sampling twice technique is existential unforgeable under adaptive chosen message attack in the ideal cipher model assuming RSA is one-way. The proof is almost the same as that for the Rivest–Shamir–Tauman scheme. The difference is as follows.

In the proof of unforgeability for the Rivest–Shamir–Tauman scheme, given $y \in \mathbb{Z}^*_N$, one slips y as a "gap" between two consecutive E functions along the ring. Then, the forger has to compute the e-th root of y, and this leads one to obtain the e-th root of y.

In the proof for the scheme with the sampling twice technique, given $y \in \mathbb{Z}^*_N$, we pick a random bit $t \in \{0,1\}$, set $y' \leftarrow y + tN$. If $y' < 2^k$ then one slips y' as a "gap" between two consecutive E functions along the ring. The rest of the proof is the same as that for the Rivest–Shamir–Tauman scheme

5.6 Efficiency

We show the number of modular exponentiations to sign and to verify, the size of signatures, and the number of random bits to sign in Figure 5.3. We assume that each N_i is uniformly

	Repeating	Expanding [76]	RSACD	Sampling Twice
# of mod. exp. to sign (average / worst)	$1.5r$ / kr	r / r	$1.5r$ / $2r$	$2r$ / $2r$
# of mod. exp. to verify (average / worst)	r / r	r / r	$1.5r$ / $2r$	r / r
size of signatures	$(3r+1)k-1$	$(3r+1)k+160(r+1)$	$(3r+1)k$	$(3r+1)k+r$
# of random bits to sign (average / worst)	$1.5k(r-1)+k-1$ / $k^2(r-1)+k-1$	$(k+160)r$ / $(k+160)r$	kr / kr	$3(k+1)(r-1)+k$ / $3(k+1)(r-1)+k$

Figure 5.3: The comparison of the ring signature schemes ($|N_i| = k$)

distributed in $(2^{k-1}, 2^k)$.

In the schemes with sampling twice and RSACD, it is necessary for each ring member to choose her RSA modulo with the same length, and in the scheme with repeating, it is necessary for each ring member to choose her RSA modulo with almost the same length. In contrast to these schemes, in the scheme with expanding, there is no restriction on the lengths of users' moduli. However, if there is one ring member whose RSA modulo is much larger than the other member's moduli, then the size of the signature and the number of random bits depends on the largest modulo. For example, if there is a user whose RSA modulo has length $k+\ell$ and the other users' moduli have lengths k, then the size of signature is $(3r+1)k+160(r+1)+\ell(r+4)$ and the number of random bits to sign is $r(k+160)+r\ell$.

CHAPTER 6

A Family of Paillier's Trap-door Permutations and its Applications to Public-Key Encryption with Anonymity

In [68], Paillier provided a trap-door one-way bijective function, and proved that the function is one-way if and only if the problem of extracting N-th roots modulo N is hard.

In this chapter, we focus on the four techniques, repeating, expanding, RSACD, and sampling twice, in the case using the Paillier's bijective function instead of the RSA function. We slightly modify his function and construct a family of Paillier's trap-door permutations denoted by Paillier. We also construct a family of Paillier's trap-door permutations with a common domain denoted by PCD, and prove the relations in Figure 6.1 for $\theta > 0.5$. Here, RSA_N denotes an RSA family of trap-door permutations with the fixed exponent N.

We also apply Paillier and PCD to encryption, and obtain Paillier-OAEP (OAEP with Paillier's trap-door permutation) with repeating, that with expanding, that with sampling twice, and PCD-OAEP (OAEP with Paillier's trap-door permutation with a common domain), and prove their security.

The organization of this chapter is as follows. In Section 6.1, after reviewing the Paillier's bijective function [68], we propose a family of Paillier's trap-door permutations denoted by Paillier and a family of Paillier's trap-door permutations with a common domain denote by PCD. We also prove that the θ-partial one-wayness of Paillier is equivalent to the one-wayness of Paillier for $\theta > 0.5$, the θ-partial one-wayness of PCD is equivalent to the

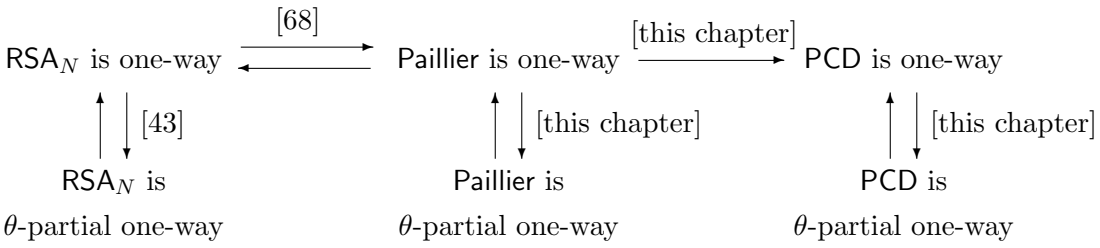

Figure 6.1: Relationships between RSA$_N$, Paillier, and PCD for $\theta > 0.5$.

one-wayness of PCD for $\theta > 0.5$, and that the one-wayness of Paillier is reduced to that of PCD. In Section 6.2, we propose applications of Paillier and PCD to encryption and prove that our schemes provide the anonymity and the indistinguishability in the random oracle model assuming that RSA$_N$ is one-way.

6.1 A Family of Paillier's Trap-door Permutations and that with a Common Domain

In this section, we propose a family of Paillier's trap-door permutations and that with a common domain.

6.1.1 Paillier's Bijective Functions

In [68], Paillier provided the bijective function $g_N : \{x_1 + x_2 \cdot N | x_1 \in \mathbb{Z}_N, x_2 \in \mathbb{Z}_N^*\} \to \mathbb{Z}_{N^2}^*$ such that

$$g_N(x) = (1 + Nx_1)x_2^N \bmod N^2$$

where $x_1 = x \bmod N$ and $x_2 = x \operatorname{div} N$. By using the trap-door $\lambda = \mathrm{lcm}(p-1, q-1)$ where $N = pq$, we can compute $g_N^{-1}(y) = x_1 + x_2 \cdot N$, where $L(u) = (u-1)/N$,

$$x_1 \leftarrow \frac{L(y^\lambda \bmod N^2)}{\lambda} \bmod N, \quad \text{and} \quad x_2 \leftarrow (y \cdot (1 - Nx_1))^{N^{-1} \bmod \lambda} \bmod N^2$$

He proved the following proposition.

Proposition 6.1 ([68])**.** *The family of Paillier's bijective functions is one-way if and only if RSA$_N$ is one-way.*

Definition 6.1 (the RSA family of trap-door permutations with the fixed exponent N)**.** *The RSA family of trap-door permutations with the fixed exponent N RSA$_N = (K, S, E)$ is as follows. The key generation algorithm K takes as input a security parameter k and picks random, distinct primes p, q such that $2^{\lceil k/2 \rceil - 1} < p, q < 2^{\lceil k/2 \rceil}$ and $|p^2 q^2| = 2k$. It sets $N = pq$ (i.e. $2^{2k-1} < N^2 < 2^{2k}$.) and $\lambda = \lambda(N) = \mathrm{lcm}(p-1, q-1)$. It returns a public key*

$pk = (N, k)$ and a secret key $sk = (N, k, \lambda)$. $\mathrm{Dom}_{\mathsf{RSA}_N}(N, k)$ *and* $\mathrm{Rng}_{\mathsf{RSA}_N}(N, k)$ *are both equal to* $\mathbb{Z}_N^*$*. The evaluation algorithm* $E_{N,k}(x) = x^N \bmod N$ *and the inversion algorithm* $I_{N,k,\lambda}(y) = y^{N^{-1} \bmod \lambda} \bmod N$*. The sampling algorithm returns a random point in* $\mathbb{Z}_N^*$*.*

6.1.2 A Family of Paillier's Trap-door Permutations

In this section, we propose a family of Paillier's trap-door permutations denoted by Paillier and prove that the θ-partial one-wayness of Paillier is equivalent to the one-wayness of Paillier for $\theta > 0.5$.

The domain and the range of the Paillier's bijective function are different. In order to construct a *permutation* based on the Paillier's bijective function, we consider a function $h_N : \mathbb{Z}_{N^2}^* \to \{x_1 + x_2 \cdot N | x_1 \in \mathbb{Z}_N, x_2 \in \mathbb{Z}_N^*\}$ such that $h_N(x) = (x \operatorname{div} N) + (x \bmod N) \cdot N$. It is clear that h_N is bijective and $h_N^{-1}(y) = (y \operatorname{div} N) + (y \bmod N) \cdot N$. Therefore, $h_N \circ g_N$ is a trap-door permutation over $\{x_1 + x_2 \cdot N | x_1 \in \mathbb{Z}_N, x_2 \in \mathbb{Z}_N^*\}$.

We now propose the family of Paillier's trap-door permutations denoted by Paillier.

Definition 6.2 (the family of Paillier's trap-door permutations)**.** *The specifications of the family of Paillier's trap-door permutations* $\mathsf{Paillier} = (K, S, E)$ *are as follows. The key generation algorithm K takes as input a security parameter k, runs the key generation algorithm for* RSA_N*, and returns a public key $pk = (N, k)$ and a secret key $sk = (N, k, \lambda)$.* $\mathrm{Dom}_{\mathsf{Paillier}}(N, k)$ *and* $\mathrm{Rng}_{\mathsf{Paillier}}(N, k)$ *are both equal to* $\{x_1 + x_2 \cdot N | x_1 \in \mathbb{Z}_N, x_2 \in \mathbb{Z}_N^*\}$*. The sampling algorithm returns a random point in* $\mathrm{Dom}_{\mathsf{Paillier}}(N, k)$*. The evaluation algorithm* $E_{N,k}(x) = F_N^{\mathsf{P}}(x)$*, and the inversion algorithm* $I_{N,k,\lambda}(y) = G_{N,\lambda}^{\mathsf{P}}(y)$ *are as follows. Note that* $F_N^{\mathsf{P}} = h_N \circ g_N$ *and* $G_{N,\lambda}^{\mathsf{P}} = g_N^{-1} \circ h_N^{-1}$*.*

`Function` $F_N^{\mathsf{P}}(x)$	`Function` $G_{N,\lambda}^{\mathsf{P}}(y)$
$x_1 \leftarrow x \bmod N;\ \ x_2 \leftarrow x \operatorname{div} N$	$y_1 \leftarrow y \bmod N;\ \ y_2 \leftarrow y \operatorname{div} N;\ \ Y \leftarrow y_1 \cdot N + y_2$
$Y \leftarrow (1 + Nx_1)x_2^N \bmod N^2$	$x_1 \leftarrow \dfrac{L(Y^\lambda \bmod N^2)}{\lambda} \bmod N$
$y_1 \leftarrow Y \operatorname{div} N;\ \ y_2 \leftarrow Y \bmod N$	$x_2 \leftarrow (Y \cdot (1 - Nx_1))^{N^{-1} \bmod \lambda} \bmod N^2$
$y \leftarrow y_1 + y_2 \cdot N$	$x \leftarrow x_1 + x_2 \cdot N$
`return` y	`return` x

From Proposition 6.1, we can easily see the following lemma.

Lemma 6.1. Paillier *is one-way if and only if* RSA_N *is one-way.*

We prove the following theorem. Note that we cannot prove the following theorem by directly applying a similar argument for RSA in [43].

Theorem 6.1. *The θ-partial one-wayness of* Paillier *is equivalent to the one-wayness of* Paillier *for $\theta > 0.5$.*

Proof. It is easy to see that if Paillier is θ-partial one-way then Paillier is one-way. Therefore, we prove the opposite direction.

Let A be an algorithm that outputs the $2k - k_0$ most significant bits of the pre-image of its input $y \in \mathrm{Rng}_{\mathsf{Paillier}}(N, k)$ with $k > k_0$ (i.e. A is a $((2k - k_0)/2k)$-partial inverting algorithm for Paillier with $k > k_0$), with success probability $\epsilon = \mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{Paillier},A}(k)$ where $\theta = (2k - k_0)/k > 0.5$, within time bound t. We prove that there exists an algorithm B that outputs a pre-image of y with success probability $\epsilon' = \mathbf{Adv}^{\text{1-pow-fnc}}_{\mathsf{Paillier},B}(k) \geq \epsilon/2$, within time bound $t' \leq t + O(k^3)$. We construct the algorithm B as follows.

$$
\begin{array}{l}
\texttt{Algorithm } B((N,k),y) \\
\quad X \leftarrow A((N,k),y);\ c \xleftarrow{R} \{0,1\};\ x_2 \leftarrow ((2^{k_0} \cdot X) \text{ div } N) + c \\
\quad y_1 \leftarrow y \bmod N;\ y_2 \leftarrow y \text{ div } N;\ Y \leftarrow y_1 \cdot N + y_2 \\
\quad \texttt{find } x_1 \texttt{ s.t. } 1 + Nx_1 = \dfrac{Y}{(x_2)^N} \bmod N^2 \\
\quad x \leftarrow x_1 + x_2 \cdot N;\ \texttt{return } x
\end{array}
$$

Assume that A outputs correctly, that is, X is the most $2k - k_0$ significant bits of x. We know $x = 2^{k_0} \cdot X + R$ for some $0 < R < 2^{k_0}$. Thus, $x_2 = x \text{ div } N = (2^{k_0} \cdot X) \text{ div } N + ((2^{k_0} \cdot X) \bmod N + R) \text{ div } N$. Since $R < 2^{k_0} \leq 2^{k-1} < N$ (Note that $k_0 \leq k - 1$, since $k, k_0 \in \mathbb{N}$ and $k_0 < k$.), we have $(2^{k_0} \cdot X) \bmod N + R < 2N$. Hence, $((2^{k_0} \cdot X) \bmod N + R) \text{ div } N$ is equal to 0 or 1, and we have $x_2 = (2^{k_0} \cdot X) \text{ div } N$ or $(2^{k_0} \cdot X) \text{ div } N + 1$.

It is easy to see that if x_2 is correct then the output of B, that is, $x = x_1 + x_2 \cdot N$ is the pre-image of y. Therefore, $\epsilon' = \mathbf{Adv}^{\text{1-pow-fnc}}_{\mathsf{Paillier},B}(k) \geq \epsilon/2$. It is not hard to see that $t' \leq t + O(k^3)$. □

Fujisaki, Okamoto, Pointcheval, and Stern [43] showed that the θ-partial one-wayness of RSA is equivalent to the one-wayness of RSA for $\theta > 0.5$. In their reduction, they assume the θ-partial inverting algorithm A for RSA with advantage ϵ, and construct the inverting algorithm B for RSA by running A twice. Then, the success probability of B is approximately $\sqrt{\epsilon}$. Furthermore, their reduction can be extended to the case that θ is a constant fraction less than 0.5. That is, B runs A $1/\theta$ times, and the success probability decreases to approximately $\epsilon^{1/\theta}$.

Our reduction for Paillier is tight than that for RSA in [43] with respect to both the success probability and the running time. However, our reduction cannot to be extended to the case that θ is a constant fraction less than 0.5.

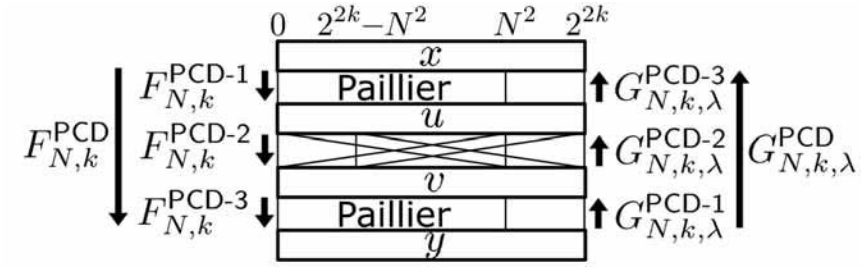

Figure 6.2: The functions $F^{\mathsf{PCD}}_{N,k}$ and $G^{\mathsf{PCD}}_{N,k,\lambda}$.

6.1.3 A Family of Paillier's Trap-door Permutations with a Common Domain

In this section, we construct a family of Paillier's trap-door permutations with a common domain denoted by PCD and prove that the θ-partial one-wayness of PCD is equivalent to the one-wayness of Paillier for $\theta > 0.5$.

The construction of PCD

The construction of PCD is similar to that of RSACDRSACD in Section 2.3.2.

Definition 6.3 (the family of Paillier's trap-door permutations with a common domain)**.** *The family of Paillier's trap-door permutations with a common domain* $\mathsf{PCD} = (K, S, E)$ *is as follows. The key generation algorithm is the same as that of* Paillier. $\mathrm{Dom}_{\mathsf{PCD}}(N,k)$ *and* $\mathrm{Rng}_{\mathsf{PCD}}(N,k)$ *are both equal to* $\{x_1 + x_2 \cdot N | (x_1 + x_2 \cdot N) \in [0, 2^{2k}),\ x_1 \in \mathbb{Z}_N,\ (x_2 \bmod N) \in \mathbb{Z}_N^*\}$*. The sampling algorithm returns a random point in* $\mathrm{Dom}_{\mathsf{PCD}}(N,k)$*. The evaluation algorithm* $E_{N,k}(x) = F^{\mathsf{PCD}}_{N,k}(x)$*, and the inversion algorithm* $I_{N,k,\lambda}(y) = G^{\mathsf{PCD}}_{N,k,\lambda}(y)$ *are as follows. (See also Figure 6.2.)*

```
Function F^PCD_{N,k}(x)
  u ← F^PCD-1_{N,k}(x); v ← F^PCD-2_{N,k}(u); y ← F^PCD-3_{N,k}(v)
  return y
```

```
Function F^PCD-1_{N,k}(x)
  if (x < N^2)
    u ← F^P_N(x)
  else u ← x
  return u
```

```
Function F^PCD-2_{N,k}(u)
  if (u < 2^{2k} − N^2) v ← u + N^2
  elseif (2^{2k} − N^2 ≤ u < N^2) v ← u
  else v ← u − N^2
  return v
```

```
Function F^PCD-3_{N,k}(v)
  if (v < N^2)
    y ← F^P_N(v)
  else y ← v
  return y
```

```
Function G^PCD_{N,k,λ}(y)
  v ← G^PCD-1_{N,k,λ}(y); u ← G^PCD-2_{N,k,λ}(v); x ← G^PCD-3_{N,k,λ}(u)
  return x
```

```
Function G^PCD-1_{N,k,λ}(y)
  if (y < N^2)
    v ← G^P_{N,λ}(y)
  else v ← y
  return v
```

```
Function G^PCD-2_{N,k,λ}(v)
  if (v < 2^{2k} − N^2) u ← v + N^2
  elseif (2^{2k} − N^2 ≤ v < N^2) u ← v
  else u ← v − N^2
  return u
```

```
Function G^PCD-3_{N,k,λ}(u)
  if (u < N^2)
    x ← G^P_{N,λ}(u)
  else x ← u
  return x
```

The choice of N^2 from $(2^{2k-1}, 2^{2k})$ ensures that all elements in $\mathrm{Dom}_{\mathsf{PCD}}(N,k)$ are permuted by F_N^{P} at least once. Since F_N^{P} is a permutation over $\mathrm{Dom}_{\mathsf{Paillier}}(N,k)$, both $F_{N,k}^{\mathsf{PCD\text{-}1}}$ and $F_{N,k}^{\mathsf{PCD\text{-}3}}$ are permutations over $\mathrm{Dom}_{\mathsf{PCD}}(N,k)$. Since it is clear that $F_{N,k}^{\mathsf{PCD\text{-}2}}$ is a permutation over $\mathrm{Dom}_{\mathsf{PCD}}(N,k)$, we have that $F_{N,k}^{\mathsf{PCD}}$ is a permutation over $\mathrm{Dom}_{\mathsf{PCD}}(N,k)$.

Property of PCD

In this section, we prove the θ-partial one-wayness of PCD is equivalent to the one-wayness of PCD for $\theta > 0.5$, and that the one-wayness of PCD is equivalent to the one-wayness of Paillier.

We first prove the partial one-wayness of PCD. Note that we cannot prove this by directly applying a similar argument for that of RSACD.

Theorem 6.2. *The θ-partial one-wayness of* PCD *is equivalent to the one-wayness of* PCD *for $\theta > 0.5$.*

Proof. It is easy to see that if PCD is θ-partial one-way then PCD is one-way. Therefore, we prove the opposite direction.

Let A be an algorithm that outputs the $2k - k_0$ most significant bits of the pre-image of its input $y \in \mathrm{Rng}_{\mathsf{PCD}}(N,k)$ with $k > k_0$ (i.e. A is a $((2k-k_0)/2k)$-partial inverting algorithm for PCD with $k > k_0$), with success probability $\epsilon = \mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{PCD},A}(k)$ where $\theta = (2k-k_0)/2k > 0.5$, within time bound t. We prove that there exists an algorithm B that outputs a pre-image of y with success probability $\epsilon' = \mathbf{Adv}^{\text{1-pow-fnc}}_{\mathsf{PCD},B}(k) \geq \epsilon/2$. within time bound $t' \leq t + O(k^3)$. We construct the algorithm B as follows.

$\texttt{Algorithm}\ B((N,k),y)$
$X \leftarrow A((N,k),y);\ c \xleftarrow{R} \{0,1\};\ x_2 \leftarrow ((2^{k_0}\cdot X) \operatorname{div} N) + c$
$y_1 \leftarrow y \bmod N;\ y_2 \leftarrow y \operatorname{div} N$
$\texttt{if}\ (x_2 \geq N \vee y_2 \geq N)$
$Y \leftarrow y_1 \cdot N + (y_2 \bmod N)$
$\texttt{find}\ x_1\ \texttt{s.t.}\ 1 + Nx_1 = \dfrac{Y}{(x_2 \bmod N)^N} \bmod N^2$
$\texttt{else}$
$Z \leftarrow y_1 \cdot N + y_2;\ w_2 \leftarrow x_2^N \bmod N$
$\texttt{find}\ x_1\ \texttt{s.t.}\ 1 + Nx_1 = \dfrac{1}{x_2^N}\left[\left(\dfrac{Z}{w_2^N} - 1\right) + w_2\right] \bmod N^2$
$x \leftarrow x_1 + x_2 \cdot N$
$\texttt{return}\ x$

Analysis

Assume that $y = y_1 + y_2 \cdot N \in \mathrm{Rng}_{\mathsf{PCD}}(N,k)$ and $x = x_1 + x_2 \cdot N$ which is the pre-image of y, that is, $x = G^{\mathsf{PCD}}_{N,k,\lambda}(y)$.

The algorithm B computes x_2 in a similar way as the inverting algorithm in the proof of Theorem 6.1.

If $x_2 \geq N$ or $y_2 \geq N$, x is permuted by F^{P}_N only once, and then, we have

$$y_1 + (y_2 \bmod N)\cdot N = F^{\mathsf{P}}_N(x_1 + (x_2 \bmod N)\cdot N).$$

Therefore, we can compute x_1 in a similar way as the inverting algorithm in the proof of Theorem 6.1 with replacing x_2 by $x_2 \bmod N$ and y_2 by $y_2 \bmod N$.

If $x_2 < N$ and $y_2 < N$, x is permuted by F^{P}_N twice, that is, $y = F^{\mathsf{P}}_N(F^{\mathsf{P}}_N(x))$. Assume that $w = w_1 + w_2 \cdot N = F^{\mathsf{P}}_N(x)$. By the definition of F^{P}_N, we have

$$w_1 \cdot N + w_2 = (1 + Nx_1)x_2^N \pmod{N^2}$$

and

$$Z = y_1 \cdot N + y_2 = (1 + Nw_1)w_2^N \pmod{N^2}.$$

Thus,

$$(Nw_1 =)\ \ (1 + Nx_1)x_2^N - w_2 = \frac{Z}{w_2^N} - 1 \pmod{N^2},$$

$$1 + Nx_1 = \frac{1}{x_2^N}\left[\left(\frac{Z}{w_2^N} - 1\right) + w_2\right] \pmod{N^2}.$$

Since $1 + Nx_1 < N^2$,

$$1 + Nx_1 = \frac{1}{x_2^N}\left[\left(\frac{Z}{w_2^N} - 1\right) + w_2\right] \bmod N^2. \tag{6.1}$$

Furthermore, since $w_2 = ((1+Nx_1)x_2^N \bmod N^2) \bmod N = x_2^N \bmod N$, B can compute the right term of equation 6.1 and compute x_1.

Hence, if x_2 is correct then $x = x_1 + x_2 \cdot N$ is the pre-image of y, and we have $\epsilon' = \mathbf{Adv}^{\text{1-pow-fnc}}_{\mathsf{PCD},B}(k) \geq \epsilon/2$. It is also clear that $t' \leq t + O(k^3)$. □

We can prove the following theorem in a similar way as that of the relationship between RSA and RSACD.

Theorem 6.3. *If* Paillier *is one-way then* PCD *is one-way.*

Proof. We prove that if there exists a polynomial-time inverting algorithm A for PCD with non-negligible probability $\epsilon = \mathbf{Adv}^{\text{1-pow-fnc}}_{\mathsf{PCD},A}(k)$, then there exists a polynomial-time inverting algorithm D for Paillier with non-negligible probability $\epsilon' = \mathbf{Adv}^{\text{1-pow-fnc}}_{\mathsf{Paillier},D}(k)$. We specify the algorithm D to compute a pre-image of $Y \in \mathrm{Rng}_{\mathsf{Paillier}}(N,k)$.

```
Algorithm D((N,k), Y)
  c ←R {0,1}
  if (c = 0)
    y ← Y; x ← A((N,k), y); u ← F^PCD-1_{N,k}(x); v ← F^PCD-2_{N,k}(u); X ← v
  else
    u ← Y; v ← F^PCD-2_{N,k}(u); y ← F^PCD-3_{N,k}(v); x ← A((N,k), y); X ← x
  return X
```

Now, we analyze the advantage of D. In the following, we assume that Y is uniformly distributed over $\mathrm{Rng}_{\mathsf{Paillier}}(N,k)$ and w is uniformly distributed over $\mathrm{Rng}_{\mathsf{PCD}}(N,k)$. If A outputs correctly then D outputs correctly (See Figure 6.2.). Therefore,

$$\begin{aligned}
\epsilon' &> \Pr[c = 0 \ \wedge \ A((N,k),Y) \text{ is correct}] + \Pr[c = 1 \ \wedge \ A((N,k),Z) \text{ is correct}] \\
&= \frac{1}{2} \cdot (\Pr[A((N,k),Y) \text{ is correct}] + \Pr[A((N,k),Z) \text{ is correct}]) \\
&\geq \frac{1}{2} \cdot (\Pr[A((N,k),Y) \text{ is correct}] + \Pr[A((N,k),Z) \text{ is correct } \wedge \ N^2 \leq Z < 2^{2k}])
\end{aligned}$$

where $Z = F^{\mathsf{PCD-3}}_{N,k}(F^{\mathsf{PCD-2}}_{N,k}(Y))$, and we have

$$\begin{aligned}
\Pr[A((N,k),Y) \text{ is correct}] &= \Pr[A((N,k),w) \text{ is correct} \mid 0 \leq w < N^2] \\
&> \Pr[A((N,k),w) \text{ is correct } \wedge \ 0 \leq w < N^2].
\end{aligned}$$

Noticing that $Z = F^{\mathsf{PCD-3}}_{N,k}(F^{\mathsf{PCD-2}}_{N,k}(Y))$ and $|\mathrm{Rng}_{\mathsf{PCD}}(N,k)| \geq |\mathrm{Rng}_{\mathsf{Paillier}}(N,k)|$, we

have

$$\begin{aligned}
\Pr[N^2 \leq Z < 2^{2k}] &= \Pr[0 \leq Y < 2^{2k} - N^2] \\
&= \frac{|\{Y' | Y' \in [0, 2^{2k} - N^2) \cap \mathrm{Rng}_{\mathsf{Paillier}}(N,k)\}|}{|\mathrm{Rng}_{\mathsf{Paillier}}(N,k)|} \\
&> \frac{|\{Y' | Y' \in [0, 2^{2k} - N^2) \cap \mathrm{Rng}_{\mathsf{Paillier}}(N,k)\}|}{|\mathrm{Rng}_{\mathsf{PCD}}(N,k)|} \\
&= \frac{|\{Y' | Y' \in [0, 2^{2k} - N^2) \cap \mathrm{Rng}_{\mathsf{PCD}}(N,k)\}|}{|\mathrm{Rng}_{\mathsf{PCD}}(N,k)|} \\
&= \Pr[0 \leq w < 2^{2k} - N^2] \\
&= \Pr[N^2 \leq w < 2^{2k}].
\end{aligned}$$

Since $\Pr[A((N,k),Z) \text{ is correct} \,|\, N^2 \leq Z < 2^{2k}] = \Pr[A((N,k),w) \text{ is correct} \,|\, N^2 \leq w < 2^{2k}]$, we have

$$\begin{aligned}
&\Pr[A((N,k),Z) \text{ is correct } \wedge\ N^2 \leq Z < 2^{2k}] \\
&\quad = \Pr[N^2 \leq Z < 2^{2k}] \times \Pr[A((N,k),Z) \text{ is correct} \,|\, N^2 \leq Z < 2^{2k}] \\
&\quad > \Pr[N^2 \leq w < 2^{2k}] \times \Pr[A((N,k),Z) \text{ is correct} \,|\, N^2 \leq Z < 2^{2k}] \\
&\quad = \Pr[N^2 \leq w < 2^{2k}] \times \Pr[A((N,k),w) \text{ is correct} \,|\, N^2 \leq w < 2^{2k}] \\
&\quad = \Pr[A((N,k),w) \text{ is correct } \wedge\ N^2 \leq w < 2^{2k}].
\end{aligned}$$

Therefore,

$$\begin{aligned}
\epsilon' &> \frac{1}{2} \cdot (\Pr[A((N,k),w) \text{ is correct } \wedge\ 0 \leq w < N^2] \\
&\qquad + \Pr[A((N,k),w) \text{ is correct } \wedge\ N^2 \leq w < 2^{2k}]) \\
&= \frac{1}{2} \cdot \Pr[A((N,k),w) \text{ is correct}] = \frac{\epsilon}{2}
\end{aligned}$$

which is non-negligible in k. □

Fujisaki, Okamoto, Pointcheval, and Stern [43] proved that the one-wayness of RSA_N is equivalent to the θ-partial one-wayness of RSA_N for $\theta > 0.5$. Therefore, the relations in Figure 6.1 (in Section 6) are satisfied for $\theta > 0.5$.

6.2 Application to Public-Key Encryption with Anonymity

In this section, we propose public-key encryption schemes with anonymity by using Paillier, PCD, and the four techniques, repeating, expanding, Paillier-CD, and sampling twice, and prove their security.

6.2.1 Our Proposed Schemes

In this section, we propose Paillier-OAEP with repeating, expanding, and sampling twice, and PCD-OAEP.

Definition 6.4 (Paillier-OAEP with repeating)**.** *Paillier-OAEP* $\mathcal{PE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *with repeating is as follows. The common-key generation algorithm* $\mathcal{G}$ *takes a security parameter* k *and returns parameters* k, k_0, *and* k_1 *such that* $k_0 + k_1 < 2k$ *for all* $k > 1$. *This defines an associated plaintext-length function* $n = 2k - k_0 - k_1$. *The key generation algorithm* $\mathcal{K}$ *takes* k, k_0, k_1, *runs the key-generation algorithm of* Paillier, *and gets* N, k, λ. *The public key* pk *is* N, k, k_0, k_1 *and the secret key* sk *is* $(N, \lambda), k, k_0, k_1$. *The other algorithms are depicted below. Let* $G : \{0,1\}^{k_0} \to \{0,1\}^{n+k_1}$ *and* $H : \{0,1\}^{n+k_1} \to \{0,1\}^{k_0}$ *be hash functions. Note that* $[x]^\ell$ *denotes the* ℓ *most significant bits of* x *and* $[x]_\ell$ *denotes the* ℓ *least significant bits of* x.

Algorithm $\mathcal{E}_{pk}(x)$
- $ctr \leftarrow -1$
- **repeat**
  - $ctr \leftarrow ctr + 1$; $r \leftarrow \{0,1\}^{k_0}$
  - $u \leftarrow \mathsf{OAEP}(x, r)$; $v \leftarrow F_N^{\mathsf{P}}(u)$
- **until**$((v < 2^{2k-1}) \vee (ctr = k_1))$
- **if** $(ctr = k_1)$ $y \leftarrow 1 || 0^{k_0+k_1} || x$
- **else** $y \leftarrow 0 || v$
- **return** y

Algorithm $\mathcal{D}_{sk}(y)$
- $b \leftarrow [y]^1$; $v \leftarrow [y]_{k_0+k_1+n}$
- **if** $(b = 1)$
  - $w \leftarrow [v]^{k_0+k_1}$; $x \leftarrow [v]_n$
  - **if** $(w = 0^{k_0+k_1})$ $z \leftarrow x$ **else** $z \leftarrow \perp$
- **else**
  - $u \leftarrow G_{N,\lambda}^{\mathsf{P}}(v)$; $z \leftarrow \mathsf{OAEP}^{-1}(u)$
- **return** z

where

Algorithm $\mathsf{OAEP}(x; r)$
- $s \leftarrow (x || 0^{k_1}) \oplus G(r)$
- $t \leftarrow r \oplus H(s)$
- **return** $s || t$

Algorithm $\mathsf{OAEP}^{-1}(u)$
- $s \leftarrow [u]^{n+k_1}$; $t \leftarrow [u]_{k_0}$; $r \leftarrow t \oplus H(s)$
- $x \leftarrow [s \oplus G(r)]^n$; $p \leftarrow [s \oplus G(r)]_{k_1}$
- **if** $(p = 0^{k_1})$ $z \leftarrow x$ **else** $z \leftarrow \perp$; **return** z

Definition 6.5 (Paillier-OAEP with expanding)**.** *Paillier-OAEP* $\mathcal{PE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *with expanding is as follows. The common-key generation algorithm* $\mathcal{G}$, *the key generation algorithm* $\mathcal{K}$, *and the hash functions* G, H *are the same as those of Paillier-OAEP with repeating. The other algorithms are depicted below. Note that the valid ciphertext* y *satisfies* $y \in [0, 2^{2k+160})$ *and* $(y \bmod N^2) \in \mathrm{Rng}_{\mathsf{Paillier}}(N, k)$.

Algorithm $\mathcal{E}_{pk}(x)$
- $r \leftarrow \{0,1\}^{k_0}$; $u \leftarrow \mathsf{OAEP}(x, r)$; $v \leftarrow F_N^{\mathsf{P}}(u)$
- $\alpha \xleftarrow{R} \{0, 1, 2, \cdots, \lfloor (2^{2k+160} - v)/N^2 \rfloor\}$
- $y \leftarrow v + \alpha N^2$
- **return** y

Algorithm $\mathcal{D}_{sk}(y)$
- $v \leftarrow y \bmod N^2$
- $u \leftarrow G_{N,\lambda}^{\mathsf{P}}(v)$
- $z \leftarrow \mathsf{OAEP}^{-1}(u)$
- **return** z

Definition 6.6 (PCD-OAEP). *PCD-OAEP $\mathcal{PE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is as follows. The common-key generation algorithm $\mathcal{G}$, the key generation algorithm $\mathcal{K}$, and the hash functions G, H are the same as those of Paillier-OAEP with repeating. The other algorithms are depicted below. Note that the valid ciphertext y satisfies $y \in \mathrm{Rng}_{\mathsf{PCD}}(N, k)$.*

$\mathtt{Algorithm}\ \mathcal{E}_{pk}(m)$	$\mathtt{Algorithm}\ \mathcal{D}_{sk}(y)$
$r \xleftarrow{R} \{0,1\}^{k_0};\ u \leftarrow \mathsf{OAEP}(x, r)$	$u \leftarrow G^{\mathsf{PCD}}_{N,k,\lambda}(y);\ z \leftarrow \mathsf{OAEP}^{-1}(u)$
$y \leftarrow F^{\mathsf{PCD}}_{N,k}(u);\ \mathtt{return}\ y$	$\mathtt{return}\ z$

Definition 6.7 (Paillier-OAEP with sampling twice). *Paillier-OAEP $\mathcal{PE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ with sampling twice is as follows. The common-key generation algorithm $\mathcal{G}$, the key generation algorithm $\mathcal{K}$, and the hash functions G, H are the same as those of Paillier-OAEP with repeating. The other algorithms are depicted below. Note that the valid ciphertext y satisfies $y \in [0, 2^{2k})$ and $(y \bmod N^2) \in \mathrm{Rng}_{\mathsf{Paillier}}(N, k)$.*

$\mathtt{Algorithm}\ \mathcal{E}_{pk}(x)$	$\mathtt{Algorithm}\ \mathcal{D}_{sk}(y)$
$r_1 \leftarrow \{0,1\}^{k_0};\ u_1 \leftarrow \mathsf{OAEP}(x, r_1);\ v_1 \leftarrow F^{\mathsf{P}}_N(u_1)$	$v \leftarrow y \bmod N^2$
$r_2 \leftarrow \{0,1\}^{k_0};\ u_2 \leftarrow \mathsf{OAEP}(x, r_2);\ v_2 \leftarrow F^{\mathsf{P}}_N(u_2)$	$u \leftarrow G^{\mathsf{P}}_{N,\lambda}(v)$
$y \leftarrow \mathtt{ChooseAndShift}_{N^2,2k}(v_1, v_2)$	$z \leftarrow \mathsf{OAEP}^{-1}(u)$
$\mathtt{return}\ y$	$\mathtt{return}\ z$

6.2.2 Analysis

In this section, we compare the four schemes proposed in the previous section.

Security

PCD-OAEP Fujisaki, Okamoto, Pointcheval, and Stern [43] proved OAEP with any partial one-way permutation is secure in the sense of IND-CCA in the random oracle model. Thus, PCD-OAEP is secure in the sense of IND-CCA in the random oracle model assuming PCD is partial one-way.

We can also prove PCD-OAEP is secure in the sense of IK-CCA in the random oracle model assuming PCD is partial one-way. More precisely, we prove the following lemma.

Lemma 6.2. *For any adversary A attacking the anonymity of PCD-OAEP $\mathcal{PE}$ under the adaptive chosen ciphertext attack, and making at most q_{dec} decryption oracle queries, q_{gen} G-oracle queries, and q_{hash} H-oracle queries, there exists a θ-partial inverting adversary B for the PCD family, such that for any k, k_0, k_1, and $\theta = \frac{2k-k_0}{2k}$,*

$$\mathbf{Adv}^{\text{ik-cca}}_{\mathcal{PE},A}(k) \leq 8q_{\mathrm{hash}}((1-\epsilon_1)(1-\epsilon_2))^{-1} \cdot \mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{PCD},B}(k) + q_{\mathrm{gen}} \cdot (1-\epsilon_2)^{-1} \cdot 2^{-2k+2}$$

where $\epsilon_1 = \frac{4}{2^{k/2-3}-1}, \epsilon_2 = \frac{2q_{\mathrm{gen}}+q_{\mathrm{dec}}+2q_{\mathrm{gen}}q_{\mathrm{dec}}}{2^{k_0}} + \frac{2q_{\mathrm{dec}}}{2^{k_1}} + \frac{2q_{\mathrm{hash}}}{2^{2k-k_0}}$, *and the running time of* B *is that of* A *plus* $q_{\mathrm{gen}} \cdot q_{\mathrm{hash}} \cdot O(k^3)$.

Proof. The proof is similar to that for RSA-RAEP, which is a variant of RSA-OAEP with repeating by Bellare, Boldyreva, Desai, and Pointcheval [3]. We construct the partial inverting algorithm M for the PCD function using a CCA-adversary A attacking the anonymity of PCD-OAEP.

Intuition. We assume that the challenge ciphertext for A is $y \in \mathrm{Rng}_{\mathsf{PCD}}(N,k)$. In order to distinguish under which key the given ciphertext y was created, the adversary A has to make queries r and s to oracles G and H, respectively, such that $s = (m||0^{k_1}) \oplus G(r)$ and $y = F^{\mathsf{PCD}}_{N,k}(s||(r \oplus H(s)))$. Therefore, A asks s to H with non-negligible probability where s is the $n+k_1$ most significant bits of $G^{\mathsf{PCD}}_{N,k,\lambda}(y)$.

We now describe the partial inverting algorithm M. The algorithm M is given $pk = (N,k)$ and a point $y \in \mathrm{Rng}_{\mathsf{PCD}}(N,k)$ where $|y| = 2k = n + k_0 + k_1$. Let $sk = (N,k,\lambda)$ be the corresponding secret key. The algorithm M is trying to find the $n + k_1 (= 2k - k_0)$ most significant bits of $G^{\mathsf{PCD}}_{N,k,\lambda}(y)$.

1) M runs the key generation algorithm of PCD with security parameter k to obtain $pk' = (N',k)$ and $sk' = (N',k,\lambda')$. Then it picks a bit $b \stackrel{R}{\leftarrow} \{0,1\}$, sets $pk_b \leftarrow (N,k)$ and $pk_{1-b} \leftarrow (N',k)$. If the above y does not satisfy $y \in (\mathrm{Rng}_{\mathsf{PCD}}(N_0,k) \cap \mathrm{Rng}_{\mathsf{PCD}}(N_1,k))$ then M outputs Fail and halts; else it continues.

2) M initializes four lists, called G-list, H-list, Y_0-list, and Y_1-list to empty. It then runs A as follows. Note that M simulates A's oracles G, H, $\mathcal{D}_{sk_0}$, and $\mathcal{D}_{sk_1}$ as described below.

   2-1) M runs $A_1(pk_0, pk_1)$ and gets (m, si) which is the output of A_1.

   2-2) M runs $A_2(y, \mathsf{si})$ and gets a bit $d \in \{0,1\}$ which is the output of A_2.

3) M chooses a random pair (h, H_h) from the H-list and outputs h as its guess for the $n + k_1$ most significant bits of $G^{\mathsf{PCD}}_{N,k,\lambda}(y)$.

M simulates the random oracles G and H, and the decryption oracle as follows:

- When A makes an oracle query g to G, then for each (h, H_h) on the H-list, M builds $z = h||(g \oplus H_h)$, and computes $y_{h,g,0} = F^{\mathsf{PCD}}_{N_0,k}(z)$ and $y_{h,g,1} = F^{\mathsf{PCD}}_{N_1,k}(z)$. For $i \in \{0,1\}$, M checks whether $y = y_{h,g,i}$. If for some h and i such a relation holds, then we have inverted y under pk_i, and we can still correctly simulate G by answering $G_g = h \oplus (m||0^{k_1})$. Otherwise, M outputs a random value G_g of length $n + k_1$. In both cases, M adds (g, G_g) to the G-list. Then, for all h, M checks if the k_1 least

significant bits of $h \oplus G_g$ are all 0. If they are, then it adds $y_{h,g,0}$ and $y_{h,g,1}$ to the Y_0-list and the Y_1-list respectively.

- When A makes an oracle query h to H, M provides A with a random string H_h of length k_0 and adds (h, H_h) to the H-list. Then for each (g, G_g) on the G-list, M builds $z = h||(g \oplus H_h)$, and computes $y_{h,g,0} = F^{\mathsf{PCD}}_{N_0,k}(z)$ and $y_{h,g,1} = F^{\mathsf{PCD}}_{N_1,k}(z)$. M checks if the k_1 least significant bits of $h \oplus G_g$ are all 0. If they are, then it adds $y_{h,g,0}$ and $y_{h,g,1}$ to the Y_0-list and the Y_1-list respectively.
- When for $i \in \{0,1\}$, A makes an oracle query $y' \in \mathrm{Rng}_{\mathsf{PCD}}(N_i, k)$ to $\mathcal{D}_{sk_i}$, M checks if there exists some $y_{h,g,i}$ in the Y_i-list such that $y' = y_{h,g,i}$. If there is, then it returns the n most significant bits of $h \oplus G_g$ to A. Otherwise it returns $\perp$ (indicating that y' is an invalid ciphertext).

In order to analyze the advantage of M, we define some events. For $i \in \{0,1\}$, let $w_i = G^{\mathsf{PCD}}_{N_i,k,\lambda_i}(y)$, $s_i = [w_i]^{n+k_1}$, and $t_i = [w_i]_{k_0}$. Let r_i be the random variable $t_i \oplus H(s_i)$.
We consider the following events.

- FBad denotes the event that
  - A G-oracle query r_0 was made by A_1 in step 3-1, and $G_{r_0} \neq s_0 \oplus (m||0^{k_1})$, or
  - A G-oracle query r_1 was made by A_1 in step 3-1, and $G_{r_1} \neq s_1 \oplus (m||0^{k_1})$.
- GBad denotes the event that
  - A G-oracle query r_0 was made by A_2 in step 3-2, and at the point in time that it was made, the H-oracle query s_0 was not on the H-list, and $G_{r_0} \neq s_0 \oplus (m||0^{k_1})$, or
  - A G-oracle query r_1 was made by A_2 in step 3-2, and at the point in time that it was made, the H-oracle query s_1 was not on the H-list, and $G_{r_1} \neq s_1 \oplus (m||0^{k_1})$.
- DBad denotes the event that
  - A $\mathcal{D}_{sk_0}$ query is not correctly answered, or
  - A $\mathcal{D}_{sk_1}$ query is not correctly answered.
- $\mathsf{G} = \neg\mathsf{FBad} \wedge \neg\mathsf{GBad} \wedge \neg\mathsf{DBad}$.

We use the events FBad, GBad, and G for proving Lemma 6.3 described below. In this chapter, we omit the proof of Lemma 6.3 since the proof of this lemma is similar to that for RSA-RAEP.

We let $\Pr[\cdot]$ denote the probability distribution in the game defining advantage. We introduce the following additional events:

- YBad denotes the event that $y \notin (\mathrm{Rng}_{\mathsf{PCD}}(N_0, k) \cap \mathrm{Rng}_{\mathsf{PCD}}(N_1, k))$.
- FAskS denotes the event that H-oracle query s_0 or s_1 was made by A_1 in step 3-1.
- AskR denotes the event that (r_0, G_{r_0}) or (r_1, G_{r_1}) is on the G-list at the end of step 3-2.
- AskS denotes the event that (s_0, H_{s_0}) or (s_1, H_{s_1}) is on the H-list at the end of step 3-2.

We use the event FAskS for proving Lemma 6.3. In this chapter, we omit the proof of Lemma 6.3 since the proof of this lemma is similar to that for RSA-RAEP.

Now, we analyze the advantage of M. The algorithm M wins the game if it outputs s_b. If (s_b, H_{s_b}) is on the H-list, then M outputs s_b with probability at least $1/q_{\mathrm{hash}}$. Thus,

$$\begin{aligned}
&\mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{PCD},M}(k) \\
&\quad \geq \tfrac{1}{q_{\mathrm{hash}}} \cdot \Pr[(s_b, H_{s_b}) \text{ is on the } H\text{-list}] \\
&\quad = \tfrac{1}{2q_{\mathrm{hash}}} \cdot (\Pr[(s_0, H_{s_0}) \text{ is on the } H\text{-list}|b=0] + \Pr[(s_1, H_{s_1}) \text{ is on the } H\text{-list}|b=1]) \\
&\quad \geq \tfrac{1}{2q_{\mathrm{hash}}} \cdot \Pr[\neg\mathsf{YBad}] \cdot (\Pr_1[(s_0, H_{s_0}) \text{ is on the } H\text{-list}|b=0] \\
&\qquad\qquad + \Pr_1[(s_1, H_{s_1}) \text{ is on the } H\text{-list}|b=1])
\end{aligned}$$

where $\Pr_1[\cdot]$ denote the probability distribution in the simulated game where $\neg\mathsf{YBad}$ occurs. Assuming that $\neg\mathsf{YBad}$ occurs, by the random choice of b and symmetry, we have $\Pr_1[(s_i, H_{s_i}) \text{ is on the } H\text{-list}|b=0] = \Pr_1[(s_i, H_{s_i}) \text{ is on the } H\text{-list}|b=1] = \Pr_1[(s_i, H_{s_i}) \text{ is on the } H\text{-list}]$ for $i \in \{0, 1\}$. Therefore,

$$\begin{aligned}
&\mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{PCD},M}(k) \\
&\quad \geq \tfrac{1}{2q_{\mathrm{hash}}} \cdot \Pr[\neg\mathsf{YBad}] \cdot (\Pr_1[(s_0, H_{s_0}) \text{ is on the } H\text{-list}] + \Pr_1[(s_1, H_{s_1}) \text{ is on the } H\text{-list}]) \\
&\quad \geq \tfrac{1}{2q_{\mathrm{hash}}} \cdot \Pr[\neg\mathsf{YBad}] \cdot \Pr_1[\mathsf{AskS}].
\end{aligned}$$

We next bound $\Pr_1[\mathsf{AskS}]$. We can bound this probability in a similar way as in the proof of anonymity for RSA-RAEP [3], and we have

$$\Pr_1[\mathsf{AskS}] \geq \frac{1}{2} \cdot \Pr_1[\mathsf{AskR} \wedge \mathsf{AskS}|\neg\mathsf{DBad}] \cdot \Pr_1[\neg\mathsf{DBad}|\neg\mathsf{AskS}].$$

We next bound $\Pr_1[\mathsf{AskR} \wedge \mathsf{AskS}|\neg\mathsf{DBad}]$ and $\Pr_1[\neg\mathsf{DBad}|\neg\mathsf{AskS}]$. Let $\epsilon = \mathbf{Adv}^{\text{ik-cca}}_{\mathcal{PE},A}(k)$. The proofs of the following lemmas are similar to those for RSA-RAEP.

Lemma 6.3.

$$\Pr_1[\mathsf{AskR} \wedge \mathsf{AskS}|\neg\mathsf{DBad}] \geq \frac{\epsilon}{2} \cdot \left(1 - 2q_{\mathrm{gen}} \cdot 2^{-k_0} - 2q_{\mathrm{hash}} \cdot 2^{-n-k_1}\right) - 2q_{\mathrm{gen}} \cdot 2^{-2k}.$$

Lemma 6.4.

$$\mathrm{Pr}_1[\mathsf{DBad}|\neg\mathsf{AskS}] \leq q_{\mathrm{dec}} \cdot \left(2 \cdot 2^{-k_1} + (2q_{\mathrm{gen}} + 1) \cdot 2^{-k_0}\right).$$

Intuitively, Lemma 6.3 states that if M simulates the decryption oracle for the adversary A perfectly, then A makes queries (r, G_r) and (s, H_s) such that $s = (m||0^{k_1}) \oplus G_r$ and $y = F^{\mathsf{PCD}}_{N_b,k}(s||(r \oplus H_s))$ with non-negligible probability. Lemma 6.4 states that M can simulate the decryption oracle with overwhelming probability.

By applying Lemmas 6.3 and 6.4, we have

$$\begin{aligned}
&\mathrm{Pr}_1[\mathsf{AskS}] \\
&\geq \tfrac{1}{2} \cdot \left[\tfrac{\epsilon}{2} \cdot \left(1 - \left(\tfrac{2q_{\mathrm{gen}}}{2^{k_0}} + \tfrac{2q_{\mathrm{hash}}}{2^{n+k_1}}\right)\right) - \tfrac{2q_{\mathrm{gen}}}{2^{2k}}\right] \times \left[1 - q_{\mathrm{dec}} \cdot \left(\tfrac{2}{2^{k_1}} + \tfrac{2q_{\mathrm{gen}}+1}{2^{k_0}}\right)\right] \\
&= \tfrac{\epsilon}{4} \cdot \left(1 - \left(\tfrac{2q_{\mathrm{gen}}}{2^{k_0}} + \tfrac{2q_{\mathrm{hash}}}{2^{n+k_1}}\right)\right) \times \left[1 - q_{\mathrm{dec}} \cdot \left(\tfrac{2}{2^{k_1}} + \tfrac{2q_{\mathrm{gen}}+1}{2^{k_0}}\right)\right] \\
&\qquad\qquad - \tfrac{1}{2} \cdot \tfrac{2q_{\mathrm{gen}}}{2^{2k}} \cdot \left[1 - q_{\mathrm{dec}} \cdot \left(\tfrac{2}{2^{k_1}} + \tfrac{2q_{\mathrm{gen}}+1}{2^{k_0}}\right)\right] \\
&\geq \tfrac{\epsilon}{4} \cdot \left(1 - \left(\tfrac{2q_{\mathrm{gen}}}{2^{k_0}} + \tfrac{2q_{\mathrm{hash}}}{2^{n+k_1}}\right) - q_{\mathrm{dec}} \cdot \left(\tfrac{2}{2^{k_1}} + \tfrac{2q_{\mathrm{gen}}+1}{2^{k_0}}\right)\right) - \tfrac{1}{2} \cdot \tfrac{2q_{\mathrm{gen}}}{2^{2k}} \\
&= \tfrac{\epsilon}{4} \cdot \left(1 - \left(\tfrac{2q_{\mathrm{gen}}+q_{\mathrm{dec}}+2q_{\mathrm{gen}}q_{\mathrm{dec}}}{2^{k_0}} + \tfrac{2q_{\mathrm{dec}}}{2^{k_1}} + \tfrac{2q_{\mathrm{hash}}}{2^{2k-k_0}}\right)\right) - \tfrac{q_{\mathrm{gen}}}{2^{2k}}.
\end{aligned}$$

We next bound the probability that $\neg\mathsf{YBad}$ occurs.

Lemma 6.5.

$$\mathrm{Pr}[\mathsf{YBad}] \leq \frac{4}{2^{k/2-3} - 1}.$$

Proof of Lemma 6.5. Let $N = pq$ and $N' = p'q'$. Note that $2^{\lceil k/2 \rceil - 1} < p, q, p', q' < 2^{\lceil k/2 \rceil}$ and $2^{2k-1} < N^2, N'^2 < 2^{2k}$. Since $N \cdot \phi(N) \leq |\mathrm{Rng}_{\mathsf{PCD}}(N, k)|$, we have

$$\begin{aligned}
\mathrm{Pr}[\mathsf{YBad}] &\leq \mathrm{Pr}[y \xleftarrow{R} \mathrm{Rng}_{\mathsf{PCD}}(N, k) : y \notin \mathrm{Rng}_{\mathsf{PCD}}(N', k)] \\
&\leq \frac{|\{y \,|\, y \in \mathrm{Rng}_{\mathsf{PCD}}(N, k) \ \wedge \ y \notin \mathrm{Rng}_{\mathsf{PCD}}(N', k)\}|}{|\mathrm{Rng}_{\mathsf{PCD}}(N, k)|} \\
&\leq \frac{|\{y \,|\, y \in [0, 2^{2k}) \ \wedge \ y \notin \mathrm{Rng}_{\mathsf{PCD}}(N', k)\}|}{|\mathrm{Rng}_{\mathsf{PCD}}(N, k)|} \\
&\leq \frac{2^{2k} - |\mathrm{Rng}_{\mathsf{PCD}}(N', k)|}{N \cdot \phi(N)}.
\end{aligned}$$

Furthermore, we have

$$\begin{aligned}
2^{2k} - |\mathrm{Rng}_{\mathsf{PCD}}(N', k)| &= \left|\{y' \in [0, 2^{2k}) | y' \notin \mathrm{Rng}_{\mathsf{PCD}}(N', k)\}\right| \\
&\leq \left|\{y' \in [0, 2N'^2) | y' \notin \mathrm{Rng}_{\mathsf{PCD}}(N', k)\}\right| \\
&= 2 \times \left|\{y' \in [0, N'^2) | y' \notin \mathrm{Rng}_{\mathsf{PCD}}(N', k)\}\right| \\
&= 2(N'^2 - N' \cdot \phi(N')).
\end{aligned}$$

Therefore, we can bound $\Pr[\mathsf{YBad}]$ as

$$\begin{aligned}\Pr[\mathsf{YBad}] &\leq \frac{2^{2k} - |\mathrm{Rng}_{\mathsf{PCD}}(N', k)|}{N \cdot \phi(N)} \leq \frac{2({N'}^2 - N' \cdot \phi(N'))}{N \cdot \phi(N)} = \frac{2N'(p' + q' - 1)}{N(N - p - q + 1)} \\ &\leq \frac{2N'(p' + q')}{N(N - p - q)} \leq \frac{2 \cdot 2^k(2^{\lceil k/2 \rceil} + 2^{\lceil k/2 \rceil})}{2^{k-1}(2^{k-1} - 2^{\lceil k/2 \rceil} - 2^{\lceil k/2 \rceil})} = \frac{4(1+1)}{2^{k-1-\lceil k/2 \rceil} - 1 - 1} \\ &\leq \frac{8}{2^{k/2-2} - 2} = \frac{4}{2^{k/2-3} - 1}.\end{aligned}$$

□

Substituting the bounds for the above probabilities, we have

$$\mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{PCD},M}(k) \geq \frac{1}{2q_{\mathrm{hash}}} \cdot (1 - \epsilon_1) \cdot \left(\frac{\epsilon}{4} \cdot (1 - \epsilon_2) - \frac{q_{\mathrm{gen}}}{2^{2k}}\right)$$

where $\epsilon_1 = \frac{4}{2^{k/2-3}-1}$ and $\epsilon_2 = \frac{2q_{\mathrm{gen}}+q_{\mathrm{dec}}+2q_{\mathrm{gen}}q_{\mathrm{dec}}}{2^{k_0}} + \frac{2q_{\mathrm{dec}}}{2^{k_1}} + \frac{2q_{\mathrm{hash}}}{2^{2k-k_0}}$, and re-arranging the terms, we get the claimed result. Note that $\epsilon = \mathbf{Adv}^{\text{ik-cca}}_{\mathcal{PE},A}(k)$.

Finally, we estimate the running time of M. It is the running time of A plus the time for simulating the random oracles. In the random oracle simulation, for each pair $((g, G_g), (h, H_h))$, it is sufficient to compute $y_{h,g,0} = F^{\mathsf{PCD}}_{N_0,k}(z)$ and $y_{h,g,1} = F^{\mathsf{PCD}}_{N_1,k}(z)$. Therefore, the running time of M is that of A plus $q_{\mathrm{gen}} \cdot q_{\mathrm{hash}} \cdot O(k^3)$. □

Since if RSA_N is one-way then PCD is θ-partial one-way for $\theta > 0.5$ (See Figure 6.1.), PCD-OAEP is secure in the sense of IND-CCA and IK-CCA in the random oracle model assuming RSA_N is one-way.

Paillier-OAEP with Repeating Fujisaki, Okamoto, Pointcheval, and Stern [43] proved OAEP with any partial one-way permutation is secure in the sense of IND-CCA in the random oracle model. Thus, Paillier-OAEP (OAEP with Paillier's trap-door permutation) is secure in the sense of IND-CCA in the random oracle model assuming $\mathsf{Paillier}$ is partial one-way.

We can prove that if Paillier-OAEP provides the indistinguishability then that with repeating also provides the indistinguishability. More precisely, if there exists a CCA-adversary $A = (A_1, A_2)$ attacking the indistinguishability of Paillier-OAEP with repeating with advantage ϵ, then there exists a CCA-adversary $B = (B_1, B_2)$ attacking the indistinguishability of Paillier-OAEP with advantage $\epsilon/2$. We construct B as follows.

1) B_1 gets pk and passes it to A_1. B_1 gets (m_0, m_1, si) which is an output of A_1, and B_1 outputs it.

2) B_2 gets a challenge ciphertext y. If $y \geq 2^{2k-1}$ then B_2 outputs Fail and halts; otherwise B_2 passes (y', si) to A_2 where $y' \leftarrow 0||y$. B_2 gets $d \in \{0, 1\}$ which is an output of A_2, and B_2 outputs it.

If B does not output Fail, A outputs correctly with advantage ϵ. Since $\Pr[B \text{ outputs } \mathsf{Fail}] < 1/2$, the advantage of B is greater than $\epsilon/2$.

Furthermore, we can prove that Paillier-OAEP with repeating is secure in the sense of IK-CCA in the random oracle model assuming $\mathsf{Paillier}$ is partial one-way. Noticing that the functions $F^{\mathsf{PCD}}_{N,k}$ and $G^{\mathsf{PCD}}_{N,k,\lambda}$ are replaced by F^{P}_N and $G^{\mathsf{P}}_{N,\lambda}$, respectively, and the domain of valid ciphertexts changes, we can prove the following lemma in a similar way as that for PCD-OAEP.

Lemma 6.6. *For any adversary A attacking the anonymity of Paillier-OAEP $\mathcal{PE}$ with repeating under the adaptive chosen ciphertext attack, and making at most q_{dec} decryption oracle queries, q_{gen} G-oracle queries, and q_{hash} H-oracle queries, there exists a θ-partial inverting adversary B for the* $\mathsf{Paillier}$ *family, such that for any k, k_0, k_1, and $\theta = \frac{2k-k_0}{2k}$,*

$$\mathbf{Adv}^{\text{ik-cca}}_{\mathcal{PE},A}(k) \leq 16q_{\mathrm{hash}}((1-\epsilon_1)(1-\epsilon_2))^{-1} \cdot \mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{Paillier},B}(k) + q_{\mathrm{gen}} \cdot (1-\epsilon_2)^{-1} \cdot 2^{-2k+2}$$

where $\epsilon_1 = \frac{1}{2^{k/2-3}-1}$, $\epsilon_2 = \frac{2q_{\mathrm{gen}}+q_{\mathrm{dec}}+2q_{\mathrm{gen}}q_{\mathrm{dec}}}{2^{k_0}} + \frac{2q_{\mathrm{dec}}}{2^{k_1}} + \frac{2q_{\mathrm{hash}}}{2^{2k-k_0}}$, and the running time of B is that of A plus $q_{\mathrm{gen}} \cdot q_{\mathrm{hash}} \cdot O(k^3)$.

Since the θ-partial one-wayness of $\mathsf{Paillier}$ is equivalent to the one-wayness of RSA_N for $\theta > 0.5$ (See Figure 6.1), Paillier-OAEP with repeating is secure in the sense of IND-CCA and IK-CCA in the random oracle model assuming RSA_N is one-way.

Paillier-OAEP with Sampling Twice In order to prove that Paillier-OAEP with sampling twice is secure in the sense of IND-CCA, we need the restriction as follows.

Since if c is a ciphertext of m for $pk = (N, k)$ and $c < 2^{2k} - N^2$ then $c + N^2$ is also a ciphertext of m. Thus, the adversary can ask $c + N^2$ to decryption oracle $\mathcal{D}_{sk}$ where c is a challenge ciphertext such that $c < 2^{2k} - N^2$ and $pk = (N, k)$, and if the answer of $\mathcal{D}_{sk}$ is m, then the adversary knows that c is a ciphertext of m for the key pk.

To prevent this attack, we add some natural restriction to the adversary in the definition of IND-CCA. That is, in the definition of IND-CCA, it is mandated that the adversary never queries D_{sk} on $(c \bmod N^2) + \gamma N^2$ where $\gamma \in \lfloor (2^{2k} - (c \bmod N^2))/N^2 \rfloor$.

We think this restriction is natural and reasonable. Actually, in the case of undeniable and confirmer signature schemes, Galbraith and Mao [44] defined the anonymity on undeniable signature schemes with the above restriction.

If we add this restriction then we can prove that Paillier-OAEP with sampling twice is secure in the sense of IND-CCA in the random oracle model assuming $\mathsf{Paillier}$ is partial one-way. Noticing that the domain of valid ciphertexts changes, we can prove this in a similar way as that for Paillier-OAEP with repeating.

Similarly, in order to prove that Paillier-OAEP with sampling twice is secure in the sense of IK-CCA, we need the same kind of restriction. That is, it is mandated that the adversary

never queries D_{sk_0} on $(c \bmod {N_0}^2) + \beta_0 {N_0}^2$ where $\beta_0 \in \lfloor (2^{2k} - (c \bmod {N_0}^2))/{N_0}^2 \rfloor$, and D_{sk_1} on $(c \bmod {N_1}^2) + \beta_1 {N_1}^2$ where $\beta_1 \in \lfloor (2^{2k} - (c \bmod {N_1}^2))/{N_1}^2 \rfloor$.

If we add this restriction then we can prove that Paillier-OAEP with sampling twice is secure in the sense of IK-CCA in the random oracle model assuming Paillier is partial one-way. More precisely, we can prove the following lemma, and the proof is similar to that for PCD-OAEP.

Lemma 6.7. *For any adversary A attacking the anonymity of Paillier-OAEP $\mathcal{PE}$ with sampling twice under the adaptive chosen ciphertext attack, and making at most q_{dec} decryption oracle queries, q_{gen} G-oracle queries, and q_{hash} H-oracle queries, there exists a θ-partial inverting adversary B for the* Paillier *family, such that for any k, k_0, k_1, and $\theta = \frac{2k-k_0}{2k}$,*

$$\mathbf{Adv}^{\text{ik-cca}}_{\mathcal{PE},A}(k) \leq 16 q_{\mathrm{hash}}((1-\epsilon_1)(1-\epsilon_2))^{-1} \cdot \mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{Paillier},B}(k) + q_{\mathrm{gen}} \cdot (1-\epsilon_2)^{-1} \cdot 2^{-2k+2}$$

where $\epsilon_1 = \frac{4}{2^{k/2-3}-1}$, $\epsilon_2 = \frac{2q_{\mathrm{gen}}+q_{\mathrm{dec}}+2q_{\mathrm{gen}}q_{\mathrm{dec}}}{2^{k_0}} + \frac{2q_{\mathrm{dec}}}{2^{k_1}} + \frac{2q_{\mathrm{hash}}}{2^{2k-k_0}}$, and the running time of B is that of A plus $q_{\mathrm{gen}} \cdot q_{\mathrm{hash}} \cdot O(k^3)$.

Since the θ-partial one-wayness of Paillier is equivalent to the one-wayness of RSA_N for $\theta > 0.5$ (See Figure 6.1), Paillier-OAEP with sampling twice is secure in the sense of IND-CCA and IK-CCA in the random oracle model assuming RSA_N is one-way.

Paillier-OAEP with Expanding In order to prove that Paillier-OAEP with expanding is secure in the sense of IND-CCA and IK-CCA, we need similar restriction as that for Paillier-OAEP with sampling twice. That is, in the definition of IND-CCA, it is mandated that the adversary never queries D_{sk} on $(c \bmod N^2) + \gamma N^2$ where $\gamma \in \lfloor (2^{2k+160} - (c \bmod N^2))/N^2 \rfloor$. Similarly, in the definition of IK-CCA, it is mandated that the adversary never queries D_{sk_0} on $(c \bmod {N_0}^2) + \beta_0 {N_0}^2$ where $\beta_0 \in \lfloor (2^{2k+160} - (c \bmod {N_0}^2))/{N_0}^2 \rfloor$, and D_{sk_1} on $(c \bmod {N_1}^2) + \beta_1 {N_1}^2$ where $\beta_1 \in \lfloor (2^{2k+160} - (c \bmod {N_1}^2))/{N_1}^2 \rfloor$.

If we add these restrictions then we can prove that Paillier-OAEP with expanding is secure in the sense of IND-CCA and IK-CCA in the random oracle model assuming Paillier is partial one-way. Noticing that the domain of valid ciphertexts changes, we can prove them in a similar way as those for Paillier-OAEP with repeating. In particular, we can prove the following lemma for the anonymity property.

Lemma 6.8. *For any adversary A attacking the anonymity of Paillier-OAEP $\mathcal{PE}$ with expanding under the adaptive chosen ciphertext attack, and making at most q_{dec} decryption oracle queries, q_{gen} G-oracle queries, and q_{hash} H-oracle queries, there exists a θ-partial inverting adversary B for the* Paillier *family, such that for any k, k_0, k_1, and $\theta = \frac{2k-k_0}{2k}$,*

$$\mathbf{Adv}^{\text{ik-cca}}_{\mathcal{PE},A}(k) \leq 8 q_{\mathrm{hash}}((1-\epsilon_1)(1-\epsilon_2))^{-1} \cdot \mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{Paillier},B}(k) + q_{\mathrm{gen}} \cdot (1-\epsilon_2)^{-1} \cdot 2^{-2k+2}$$

	Repeating	Expanding	PCD	Sampling Twice
# of mod. exp. to encrypt (average / worst)	1.5 / k_1	1 / 1	1.5 / 2	2 / 2
# of mod. exp. to decrypt (average / worst)	1 / 1	1 / 1	1.5 / 2	1 / 1
size of ciphertexts	$2k+1$	$2k+160$	$2k$	$2k$
# of random bits to encrypt (average / worst)	$1.5k_0$ / k_1k_0	k_0+160 / k_0+160	k_0 / k_0	$2k_0+2k+3$ / $2k_0+2k+3$

Figure 6.3: The costs of the encryption schemes.

where $\epsilon_1 = \frac{4}{2^{k/2-3}-1} + \frac{1}{2^{159}}, \epsilon_2 = \frac{2q_{\mathrm{gen}}+q_{\mathrm{dec}}+2q_{\mathrm{gen}}q_{\mathrm{dec}}}{2^{k_0}} + \frac{2q_{\mathrm{dec}}}{2^{k_1}} + \frac{2q_{\mathrm{hash}}}{2^{2k-k_0}}$*, and the running time of* B *is that of* A *plus* $q_{\mathrm{gen}} \cdot q_{\mathrm{hash}} \cdot O(k^3)$*.*

Since the θ-partial one-wayness of Paillier is equivalent to the one-wayness of RSA_N for $\theta > 0.5$ (See Figure 6.1), Paillier-OAEP with expanding is secure in the sense of IND-CCA and IK-CCA in the random oracle model assuming RSA_N is one-way.

Efficiency

We show the costs of our schemes. We show the number of modular exponentiations to encrypt, the number of modular exponentiations to decrypt, the size of ciphertexts, and the number of random bits to encrypt in Figure 6.3. We assume that N is uniformly distributed in $(2^{2k-1}, 2^{2k})$.

Paillier-OAEP with repeating is inefficient with respect to the encryption cost in the worst case. In this scheme, the number of random bits to encrypt is large in the worst case.

Paillier-OAEP with expanding is efficient with respect to the encryption and the decryption costs. However, the size of ciphertexts is about 160 bits larger than those of the other schemes.

PCD-OAEP is the most efficient among the four schemes with respect to the number of random bits to encrypt. However, the decryption cost is twice as those of the other schemes in the worst case.

Paillier-OAEP with sampling twice requires many random bits to encrypt messages. If $k = k_0/2 = k_1/2$, then the number of random bits to encrypt in Paillier-OAEP with sampling twice is at least four times as many as those of the other schemes in the average case.

CHAPTER 7

Relationships between Data-Privacy and Key-Privacy

In this chapter, we propose a new security notion for public-key encryption of key-privacy, called the strong anonymity. This captures the situation that a public-key encryption scheme provides the anonymity even if the message spaces for each public-key are different, while the anonymity proposed in [3] cannot capture such a situation.

We also show the relationships between data-privacy and key-privacy. We consider the indsitinguishability (IND) as the security notion for the data-privacy, and the anonymity (IK), the anonymity with random messages (IKR), and the strong anonymity (sIK) as those for the key-privacy.

We show the relationships between data-privacy and key-privacy in Figure 7.1. These relations hold under the chosen message attack and the adaptive chosen ciphertext attack. In this figure, for notions of security A and B,

- "$A \longrightarrow B$" means that A implies B, that is, for any public-key encryption scheme which is secure in the sense of A is also secure in the sense of B (We denote it as $A \Rightarrow B$.), and
- "$A \dashrightarrow B$" means that A does *not* imply B, that is, there exists a public-key encryption scheme which is secure in the sense of A and *not* secure in the sense of B (We denote it as $A \not\Rightarrow B$.).

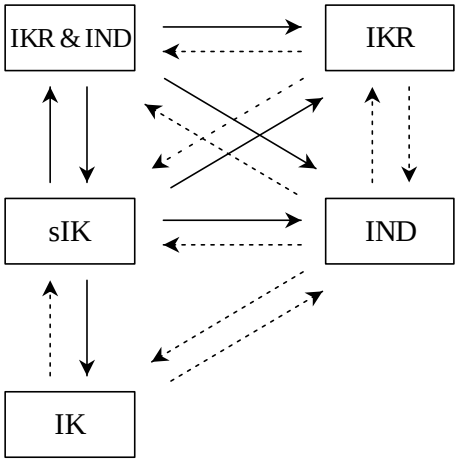

Figure 7.1: Relationships between data-privacy and key-privacy.

In this chapter, we prove the relations in Figure 7.2. In this figure, the number on the arrow refers to the section of this chapter. By using the relations in Figure 7.2 and trivial relations (IKR-atk $\wedge$ IND-atk $\Rightarrow$ IKR-atk, IKR-atk $\wedge$ IND-atk $\Rightarrow$ IND-atk), the relations which are in Figure 7.1 and not in Figure 7.2 are determined automatically.

The organization of this chapter is as follows. In Section 7.1, we review the anonymity with random messages. In Section 7.2, we propsose a new security notion called the strong anonymity. In Section 7.3, we show the relationships between data-privacy and key-privacy.

7.1 Anonymity with Random Messages

In this section, we review the definition of the anonymity with random messages.

Halevi [49] provides a simple sufficient condition for an IND-atk public-key encryption scheme to meet wIK-atk for atk $\in$ {CPA, CCA}. The condition is that even a computationally unbounded adversary, given public keys pk_0, $pk1$ and the encryption of a random message under pk_b, have only a negligible advantage in determining the random challenge bit b. In [2], Abdalla et. al. extended the Halevi's condition to identity-based encryption. They weakened the statistical (i.e. information-theoretic) requirement of to [49] a computational one.

We also consider the computational version of the Halevi's condition for public-key encryption schemes as follows.

Definition 7.1 (IKR-CPA, IKR-CCA)**.** *Let* $\Pi = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *be a public-key encryption scheme. Let* A_{cpa} *and* A_{cca} *be adversary. The adversaries* A_{cpa} *and* A_{cca} *can access to some oracles* $\mathcal{O}_{\mathrm{cpa}}$ *and* O_{cca}*, respectively. For* atk $\in$ {cpa, cca}*, we consider the following*

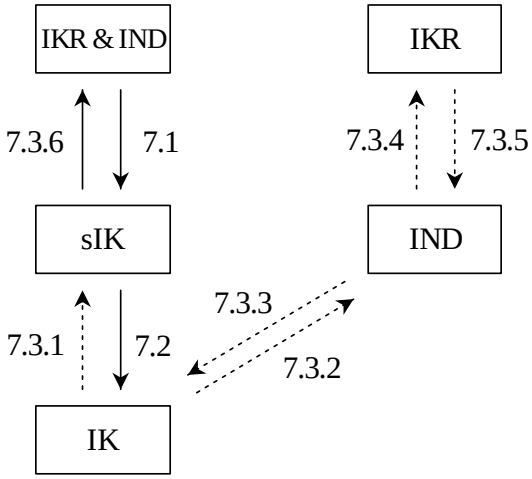

Figure 7.2: Relationships proved in this chapter.

experiment:

$$\begin{aligned}&\texttt{Experiment } \mathbf{Exp}^{\text{ikr-atk-}b}_{\Pi,A_{\text{atk}}}(k)\\ &\quad I \leftarrow \mathcal{G}(k);\ (pk_0, sk_0), (pk_1, sk_1) \leftarrow \mathcal{K}(I)\\ &\quad m \xleftarrow{R} \texttt{MSPC}(pk_b);\ c \leftarrow \mathcal{E}_{pk_b}(m)\\ &\quad d \leftarrow A^{\mathcal{O}_{\text{atk}}}_{\text{atk}}(pk_0, pk_1, c);\ \texttt{return } d\end{aligned}$$

where $\mathcal{O}_{\text{cpa}} = \epsilon$ *and* $\mathcal{O}_{\text{cca}} = \{\mathcal{D}_{sk_0}, \mathcal{D}_{sk_1}\}$. *We require that* A_{cca} *never queries the challenge* c *to either* $\mathcal{D}_{sk_0}$ *or* $\mathcal{D}_{sk_1}$.

For atk $\in$ {cpa, cca}, *we define the advantage via*

$$\mathbf{Adv}^{\text{ikr-atk}}_{\Pi,A_{\text{atk}}}(k) = \Big|\Pr[\mathbf{Exp}^{\text{ikr-atk-1}}_{\Pi,A_{\text{atk}}}(k) = 1] - \Pr[\mathbf{Exp}^{\text{ikr-atk-0}}_{\Pi,A_{\text{atk}}}(k) = 1]\Big|.$$

We say that Π *is secure in the sense of IKR-CPA (resp. IKR-CCA) if* $\mathbf{Adv}^{\text{ikr-cpa}}_{\Pi,A_{\text{cpa}}}(k)$ *(resp.* $\mathbf{Adv}^{\text{ikr-cca}}_{\Pi,A_{\text{cca}}}(k)$*) is negligible for any poly-time adversary* A_{cpa} *(resp.* A_{cca} *).*

Halevi [49] showed that for atk $\in$ {CCA, CPA}, if the public-key encryption scheme is secure in the sense of IND-atk and IKR-atk, then it is also secure in the sense of IK-atk. We can apply his proof to our strong anonymity, and see the following claim.

Claim 7.1. *For atk* $\in$ {*CCA, CPA*}, *IND-atk* $\wedge$ *IKR-atk* $\Rightarrow$ *sIK-atk.*

Proof. The proof is similar to those in [49] and [2], and is a simple hybrid argument. Let A be a poly-time algorithm in the sense of sIK-atk. It is easy to construct poly-time algorithms A_1 and A_3 in the sense of IND-atk and A_2 in the sense of IKR-atk such that

$$\Big|\Pr[\mathbf{Exp}^{\text{sik-atk-1}}_{\Pi,A}(k) = 1] - \Pr[\mathbf{Exp}^{\text{ikr-atk-1}}_{\Pi,A}(k) = 1]\Big| \leq \mathbf{Adv}^{\text{ind-atk}}_{\Pi,A_1}(k),$$

$$\Big|\Pr[\mathbf{Exp}^{\text{ikr-atk-1}}_{\Pi,A}(k) = 1] - \Pr[\mathbf{Exp}^{\text{ikr-atk-0}}_{\Pi,A}(k) = 1]\Big| \leq \mathbf{Adv}^{\text{ikr-atk}}_{\Pi,A_2}(k),$$

$$\left| \Pr[\mathbf{Exp}^{\text{ikr-atk-0}}_{\Pi,A}(k) = 1] - \Pr[\mathbf{Exp}^{\text{sik-atk-0}}_{\Pi,A}(k) = 1] \right| \leq \mathbf{Adv}^{\text{ind-atk}}_{\Pi,A_3}(k).$$

Therefore,

$$\mathbf{Adv}^{\text{sik-atk}}_{\Pi,A}(k) \leq \mathbf{Adv}^{\text{ind-atk}}_{\Pi,A_1}(k) + \mathbf{Adv}^{\text{ikr-atk}}_{\Pi,A_2}(k) + \mathbf{Adv}^{\text{ind-atk}}_{\Pi,A_3}(k)$$

and this concludes the proof. □

7.2 Strong Anonymity

We propose the definition of the strong anonymity.

Definition 7.2 (sIK-CPA, sIK-CCA)**.** *Let* $\Pi = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *be a public-key encryption scheme. Let* A_{cpa} *and* A_{cca} *be adversaries that run in two stages,* find *and* guess*. The adversaries* A_{cpa} *and* A_{cca} *can access to some oracles* $\mathcal{O}_{\text{cpa}}$ *and* O_{cca}*, respectively.*

For atk $\in$ {*cpa, cca*}*, we consider the following experiment:*

$$\begin{array}{l}\texttt{Experiment } \mathbf{Exp}^{\text{sik-atk-}b}_{\Pi,A_{\text{atk}}}(k)\\ \quad I \leftarrow \mathcal{G}(k);\ (pk_0, sk_0), (pk_1, sk_1) \leftarrow \mathcal{K}(I)\\ \quad (m_0, m_1, \mathsf{si}) \leftarrow A^{\mathcal{O}_{\text{atk}}}_{\text{atk}}(\mathsf{find}, pk);\ c \leftarrow \mathcal{E}_{pk_b}(m_b)\\ \quad d \leftarrow A^{\mathcal{O}_{\text{atk}}}_{\text{atk}}(\mathsf{guess}, c, \mathsf{si});\ \texttt{return } d\end{array}$$

where $\mathcal{O}_{\text{cpa}} = \epsilon$ *and* $\mathcal{O}_{\text{cca}} = \{\mathcal{D}_{sk_0}, \mathcal{D}_{sk_1}\}$*. We require that* $m_0 \in \texttt{MSPC}(pk_0)$ *and* $m_1 \in \texttt{MSPC}(pk_1)$*. We also require that* A_{cca} *never queries the challenge* c *to either* $\mathcal{D}_{sk_0}$ *or* $\mathcal{D}_{sk_1}$ *in the guess stage.*

For atk $\in$ {cpa, cca}*, we define the advantage via*

$$\mathbf{Adv}^{\text{sik-atk}}_{\Pi,A_{\text{atk}}}(k) = \left| \Pr[\mathbf{Exp}^{\text{sik-atk-1}}_{\Pi,A_{\text{atk}}}(k) = 1] - \Pr[\mathbf{Exp}^{\text{sik-atk-0}}_{\Pi,A_{\text{atk}}}(k) = 1] \right|.$$

We say that Π *is secure in the sense of sIK-CPA (resp. sIK-CCA) if* $\mathbf{Adv}^{\text{sik-cpa}}_{\Pi,A_{\text{cpa}}}(k)$ *(resp.* $\mathbf{Adv}^{\text{sik-cca}}_{\Pi,A_{\text{cca}}}(k)$*) is negligible for any poly-time adversary* A_{cpa} *(resp.* A_{cca} *).*

There is only one difference between the definition of the anonymity in [3] and that of the strong anonymity.

In the experiment of the definition by [3], the adversary chooses only one message $m \in \texttt{MSPC}(pk_0) \cap \texttt{MSPC}(pk_1)$ and receives a ciphertext of m encrypted with one of two keys pk_0 and pk_1. Therefore, their definition guarantees the anonymity property only when the message is chosen from the set $\texttt{MSPC}(pk_0) \cap \texttt{MSPC}(pk_1)$.

However, in some public-key encryption schemes, the ciphertext space may be common even if the message spaces for each public-key are different. and such schemes may provide the anonymity property.

To consider this situation, in the experiment of our definition, the adversary chooses two messages m_0 and m_1 where m_0 and m_1 are in the message spaces for pk_0 and pk_1, respectively, and receives either a ciphertext of m_0 encrypted with pk_0 or a ciphertext of m_1 encrypted with pk_1.

We can easily see the following claim.

Claim 7.2. *For any atk* $\in \{CPA, CCA\}$, *sIK-atk* $\Rightarrow$ *IK-atk.*

Proof. Let A be an adversary for Π in the sense of IK-atk.

We construct an algorithm B for Π in the sense of sIK-atk by using Aas follows.

1) In the find stage, B takes pk_0 and pk_1, and runs A as $(m, \mathsf{si}) \leftarrow A(\mathsf{find}, pk_0, pk_1)$. Then B outputs (m, m, si).

2) In the guess stage, B takes $c = \mathcal{E}_{pk_b}(m_b)$ and si. (Note that $m_1 = m_1 = m$.) Then, B runs A as $d \leftarrow A(\mathsf{guess}, c, \mathsf{si})$ and outputs d.

Above, in case of atk = CCA, if A makes some decryption queries, B answers to A by using B's decryption oracles. It is easy to see that $\mathbf{Adv}^{\text{sik-atk}}_{\Pi,B}(k) = \mathbf{Adv}^{\text{ik-atk}}_{\Pi,A}(k)$, and the running time of B is that of A. □

7.3 Relationships between Data-Privacy and Key-Privacy

In this section, we show the relationships between data-privacy and key-privacy.

7.3.1 IK-atk $\not\Rightarrow$ sIK-atk

Lemma 7.1. *For atk* $\in \{CPA, CCA\}$, *there exist a public-key encryption scheme* Π *which is secure in the sense of IK-atk, but not secure in the sense of sIK-atk.*

Proof. For atk $\in$ {CPA, CCA}, let $\Pi' = (\mathcal{G}', \mathcal{K}', \mathcal{E}', \mathcal{D}')$ be a public-key encryption scheme which is secure in the sense of IK-atk. Then, consider the public-key encryption scheme Π whose encryption algorithm is defined as $\mathcal{E}_{pk}(m) := \mathcal{E}'_{pk}(m)||m$. We can easily see that Π meets IK-atk, and does not meet sIK-atk. □

7.3.2 IK-atk $\not\Rightarrow$ IND-atk

Lemma 7.2. *For atk* $\in \{CPA, CCA\}$, *there exist a public-key encryption scheme* Π *which is secure in the sense of IK-atk, but not secure in the sense of IND-atk.*

Proof. We can see this by using the encryption scheme Π in the proof of Lemma 7.1. □

7.3.3 IND-atk $\not\Rightarrow$ IK-atk

Lemma 7.3. *For atk $\in$ {CPA, CCA}, there exist a public-key encryption scheme Π which is secure in the sense of IND-atk, but not secure in the sense of IK-atk.*

Proof. For atk $\in$ {CPA, CCA}, let $\Pi' = (\mathcal{G}', \mathcal{K}', \mathcal{E}', \mathcal{D}')$ be a public-key encryption scheme which is secure in the sense of IND-atk. Then, consider the public-key encryption scheme Π whose encryption algorithm is defined as $\mathcal{E}_{pk}(m) := \mathcal{E}'_{pk}(m)||pk$. We can easily see that Π meets IND-atk, and does not meet IK-atk. □

7.3.4 IND-atk $\not\Rightarrow$ IKR-atk

Lemma 7.4. *For atk $\in$ {CPA, CCA}, there exist a public-key encryption scheme Π which is secure in the sense of IND-atk, but not secure in the sense of IKR-atk.*

Proof. We can see this by using the encryption scheme Π in the proof of Lemma 7.3. □

7.3.5 IKR-atk $\not\Rightarrow$ IND-atk

Lemma 7.5. *For atk $\in$ {CPA, CCA}, there exist a public-key encryption scheme Π which is secure in the sense of IKR-atk, but not secure in the sense of IND-atk.*

Proof. For atk $\in$ {CPA, CCA}, let $\Pi' = (\mathcal{G}', \mathcal{K}', \mathcal{E}', \mathcal{D}')$ be a public-key encryption scheme which is secure in the sense of sIK-atk where the message space is common to each public-key (i.e. for any public-keys pk_0 and pk_1, $\mathtt{MSPC}(pk_0) = \mathtt{MSPC}(pk1)$). We consider the public-key encryption scheme Π whose encryption algorithm is defined as $\mathcal{E}_{pk}(m) := \mathcal{E}'_{pk}(m)||m$.

Then Π is secure in the sense of IKR-atk. We show the following claim.

Claim 7.3. *For atk $\in$ {CPA, CCA}, if Π' is secure in the sense of sIK-atk, then Π is secure in the sense of IKR-atk.*

Proof. Let A be an adversary for Π in the sense of IKR-atk.

We construct an algorithm B for Π' in the sense of sIK-atk by using A as follows.

1) In the find stage, B takes pk_0 and pk_1, and picks $m \xleftarrow{R} \mathtt{MSPC}(pk_0)(= \mathtt{MSPC}(pk_1))$. Then B sets $m_0 \leftarrow m$ and $m_1 \leftarrow m$, and outputs (m_0, m_1, si) where si contains two public-keys pk_0 and pk_1.

2) In the guess stage, B takes $c = \mathcal{E}_{pk_b}(m_b)$ and si (Note that $m_0 = m_1 = m$.). Then, B sets $c' \leftarrow c||m$ and runs A as $d \leftarrow A(pk_0, pk_1, c')$. Finally, B outputs d.

Above, in case of atk = CCA, if A makes some decryption queries, B answers to A by using B's decryption oracles. It is easy to see that $\mathbf{Adv}^{\text{sik-atk}}_{\Pi',B}(k) = \mathbf{Adv}^{\text{ikr-atk}}_{\Pi,A}(k)$, and the running time of B is that of A plus $O(k)$. □

It is easy to see that Π does not meet IND-atk. This concludes the proof of Lemma 7.5. □

7.3.6 sIK-atk ⇒ IND-atk ∧ IKR-atk

Before beginning the proof, we define an additional security notion, called strong anonymity with one random message (sIKOR).

Definition 7.3 (sIKOR-CPA, sIKOR-CCA)**.** *Let $\Pi = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme. Let A_{cpa} and A_{cca} be adversaries that run in two stages,* find *and* guess*. The adversaries A_{cpa} and A_{cca} can access to some oracles $\mathcal{O}_{\mathrm{cpa}}$ and O_{cca}, respectively.*

For atk $\in$ {cpa, cca}, we consider the following experiment:

$$
\begin{array}{l}
\texttt{Experiment } \mathbf{Exp}^{\text{sikor-atk-}b}_{\Pi, A_{\mathrm{atk}}}(k) \\
\quad I \leftarrow \mathcal{G}(k);\ (pk_0, sk_0), (pk_1, sk_1) \leftarrow \mathcal{K}(I) \\
\quad (m_0, \mathsf{si}) \leftarrow A^{\mathcal{O}_{\mathrm{atk}}}_{\mathrm{atk}}(\mathsf{find}, pk) \\
\quad m_1 \xleftarrow{R} \mathtt{MSPC}(pk_1);\ c \leftarrow \mathcal{E}_{pk_b}(m_b) \\
\quad d \leftarrow A^{\mathcal{O}_{\mathrm{atk}}}_{\mathrm{atk}}(\mathsf{guess}, c, \mathsf{si});\ \texttt{return } d
\end{array}
$$

where $\mathcal{O}_{\mathrm{cpa}} = \epsilon$ and $\mathcal{O}_{\mathrm{cca}} = \{\mathcal{D}_{sk_0}, \mathcal{D}_{sk_1}\}$. We require that $m_0 \in \mathtt{MSPC}(pk_0)$ and $m_1 \in \mathtt{MSPC}(pk_1)$. We also require that A_{cca} never queries the challenge c to either $\mathcal{D}_{sk_0}$ or $\mathcal{D}_{sk_1}$ in the guess stage.

For atk $\in$ {cpa, cca}*, we define the advantage via*

$$
\mathbf{Adv}^{\text{sikor-atk}}_{\Pi, A_{\mathrm{atk}}}(k) = \left| \Pr[\mathbf{Exp}^{\text{sikor-atk-1}}_{\Pi, A_{\mathrm{atk}}}(k) = 1] - \Pr[\mathbf{Exp}^{\text{sikor-atk-0}}_{\Pi, A_{\mathrm{atk}}}(k) = 1] \right|.
$$

We say that Π is secure in the sense of sIKOR-CPA (resp. sIKOR-CCA) if the function $\mathbf{Adv}^{\text{sikor-cpa}}_{\Pi, A_{\mathrm{cpa}}}(k)$ (resp. $\mathbf{Adv}^{\text{sikor-cca}}_{\Pi, A_{\mathrm{cca}}}(k)$) is negligible for any poly-time adversary A_{cpa} (resp. A_{cca}).

We can easily see the following lemma.

Lemma 7.6. *For any atk $\in$ {CPA, CCA}, sIK-atk ⇒ sIKOR-atk ⇒ IKR-atk.*

Proof. Let A be an adversary for Π in the sense of sIKOR-atk.

We construct an algorithm B for Π in the sense of sIK-atk by using A as follows.

1) In the find stage, B takes pk_0 and pk_1, and runs A as $(m_0, \mathsf{si}) \leftarrow A(\mathsf{find}, pk_0, pk_1)$. Then B picks $m_1 \xleftarrow{R} \mathtt{MSPC}(pk_1)$ and outputs (m_0, m_1, si') where si' contains si and two public-keys pk_0 and pk_1.

2) In the guess stage, B takes $c = \mathcal{E}_{pk_b}(m_b)$ and si'. Then, B runs A as $d \leftarrow A(\mathsf{guess}, c, \mathsf{si})$ and outputs d.

Above, in case of atk = CCA, if A makes some decryption queries, B answers to A by using B's decryption oracles. It is easy to see that $\mathbf{Adv}^{\text{sik-atk}}_{\Pi,B}(k) = \mathbf{Adv}^{\text{sikor-atk}}_{\Pi,A}(k)$, and the running time of B is that of A plus $O(k)$.

We can prove "sIKOR-atk ⇒ IKR-atk" in a similar way. □

Remark 7.1. *We can easily prove that "sIK-atk ⇒ IKR-atk" directly without using the notion sIKOR-atk. We take this approach since we use the relation "sIK-atk ⇒ sIKOR-atk" in the next proof.*

Now, we prove the relation "sIK-atk ⇒ IND-atk ∧ IKR-atk." We can rewrite this relation as "sIK-atk ⇒ IKR-atk ∨ (¬IKR-atk ∧ IND-atk)". We prove "sIK-atk ⇒ IKR-atk" in Lemma 7.6, and it is sufficient to prove "sIK-atk ⇒ ¬IKR-atk ∧ IND-atk." Since sIK-atk implies sIKOR-atk (Lemma 7.6), we prove "sIKOR-atk ⇒ ¬IKR-atk ∧ IND-atk."

Lemma 7.7. *For atk ∈ {CPA, CCA}, if there exists an adversary A for Π in the sense of IND-atk and Π is secure in the sense of IKR-atk, then there exists an adversary B for Π in the sense of sIKOR-CPA where*

$$\mathbf{Adv}^{\text{sikor-atk}}_{\Pi,B}(k) \geq \mathbf{Adv}^{\text{ind-atk}}_{\Pi,A}(k) - \lambda(k)$$

and $\lambda(k)$ is a negligible function in k, and the running time of B is that of A.

Proof. Let A be an adversary for Π in the sense of IND-atk.

We construct an algorithm B for Π in the sense of sIKOR-atk by using A as follows.

1) In the find stage, B takes pk_0 and pk_1, and runs A as $(m_0, \mathsf{si}) \leftarrow A(\mathsf{find}, pk_0)$, and outputs (m_0, si') where si' contains si, pk_0, and pk_1.

2) In the guess stage, B takes $c = \mathcal{E}_{pk_b}(m_b)$ and si' where $b \overset{R}{\leftarrow} \{0,1\}$. (Note that $m_1 \overset{R}{\leftarrow} \mathtt{MSPC}(pk_1)$ in the sIKOR-atk game). Then, B runs A as $d \leftarrow A(\mathsf{guess}, c, \mathsf{si})$ and outputs d.

Above, in case of atk = CCA, if A makes some decryption queries, B answers to A by using B's decryption oracles.

We analyze the advantage of B. If $b = 0$, then the distribution of the input of A simulated by B is identical to that of real A. If $b = 1$, in the guess stage, the input of A simulated by B is $c \leftarrow \mathcal{E}_{pk_1}(m_1)$ where $m_1 \overset{R}{\leftarrow} \mathtt{MSPC}(pk_1)$, and that of real A is $c \leftarrow \mathcal{E}_{pk_0}(m_1)$ where $m_1 \overset{R}{\leftarrow} \mathtt{MSPC}(pk_0)$. Here, in the guess stage, if the probability that the output of A simulated by B and that of real A are different is non-negligible, then it implies that A breaks Π in the sense of IKR-atk. Therefore, the probability that the output of A simulated by B and that of real A are different is negligible.

Thus, we have

$$\begin{aligned}
&\mathbf{Adv}^{\text{sikor-atk}}_{\Pi,B}(k) \\
&\quad = \left|\Pr[\mathbf{Exp}^{\text{sikor-atk-1}}_{\Pi,B}(k) = 1] - \Pr[\mathbf{Exp}^{\text{sikor-atk-0}}_{\Pi,B}(k) = 1]\right| \\
&\quad \geq \left|\Pr[\mathbf{Exp}^{\text{ind-atk-1}}_{\Pi,A}(k) = 1] - \Pr[\mathbf{Exp}^{\text{ind-atk-0}}_{\Pi,A}(k) = 1]\right| - \lambda(k) \\
&\quad = \mathbf{Adv}^{\text{sikor-atk}}_{\Pi,A}(k) - \lambda(k),
\end{aligned}$$

where $\lambda(k)$ is a negligible function in k. It is easy to see that the running time of B is equal to that of A. □

CHAPTER 8

Plaintext Awareness in the Two-Key Setting and a Generic Conversion for Encryption with Anonymity

In this chapter, we propose a new secrity notion of public-key encryption scheme with respect to the anonymity property, called plaintext awareness in the two-key setting (PATK). We also prove that if a public-key encryption scheme is secure in the sense of PATK, then it is also secure in the sense of IK-CCA. Since it looks much easier to prove that a public-key encryption scheme is secure in the sense of PATK than to prove directly that it is secure in the sense of IK-CCA, the notion of PATK is useful to prove the anonymity property of public-key encryption schemes. We also propose the first generic conversion scheme for the anonymity from IK-CPA to IK-CCA.

The organization of this chapter is as follows. In Section 8.1, we review the security notions for public-key encryption. We also review the definition and the security notion of symmetric-key encryption. In Section 8.2, we propose the notion of plaintext awareness in the two-key setting (PATK), and prove that PATK implies IK-CCA. In Section 8.3, we review the conversion scheme to IND-CCA proposed by Fujisaki and Okamoto [42]. In Section 8.4, we propose a generic conversion scheme for the anonymity. More precisely, we prove that the public-key encryption scheme derived from the Fujisaki-Okamoto conversion scheme, where the basic public-key encryption scheme is secure in the sense of IK-CPA, is secure in the sense of IK-CCA in the random oracle model.

8.1 Definitions

In this section, we review the security notions for public-key encryption. We also review the definition and the security notion of symmetric-key encryption. Note that, in this section, we redefine some security notions (IND, IK) for public-key encryption in Section 3.1, which are used only in this chapter. Note that the definitions of IND and IK in this section are essentially equivalent to those in Section 3.1, respectively.

8.1.1 Public-Key Encryption

γ-uniformity

We review a property of public-key encryption, called γ-uniformity, following [42].

Definition 8.1 (γ-uniformity)**.** *Let $\Pi = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme. We say that Π is γ-uniform, if, for any $I \leftarrow \mathcal{G}(1^k)$, $(pk, sk) \leftarrow K(I)$, $m \in \mathtt{MSPC}(pk)$, and $y \in \{0,1\}^*$,*

$$\Pr[r \xleftarrow{R} \mathtt{COINS}(pk) \ : \ y = \mathcal{E}_{pk}(x; r)] < \gamma.$$

One-Wayness

We review a weak security notion for public-key encryption, called one-wayness, following [42].

Definition 8.2 (OW)**.** *Let $\Pi = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme. Let A be an adversary. We define the advantage of A via*

$$\mathbf{Adv}^{\mathrm{ow}}_{\Pi,A}(k) = \Pr[I \leftarrow \mathcal{G}(1^k);\ (pk, sk) \leftarrow \mathcal{K}(I);\ m \xleftarrow{R} \mathtt{MSPC}(pk);\ c \leftarrow \mathcal{E}_{pk}(m) \ : \ A(c, pk) = m].$$

We say that A is a (t, ϵ)-adversary for Π in the sense of OW if A runs in at most time t and archives $\mathbf{Adv}^{\mathrm{ow}}_{\Pi,A}(k) \geq \epsilon$. We say that Π is (t, ϵ)-secure in the sense of OW if there is no (t, ϵ)-adversary for Π in that sense.

Indistinguishability

We review the definition of the indistinguishability of ciphertexts, following [42].

Definition 8.3 (IND-CPA, IND-CCA)**.** *Let $\Pi = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme. Let A_{cpa} and A_{cca} be adversaries that run in two stages,* find *and* guess*. The adversaries A_{cpa} and A_{cca} have access to some oracles $\mathcal{O}_{\mathrm{cpa}}$ and O_{cca}, respectively. For atk $\in$ {cpa, cca}, we define the advantages of A_{atk} via*

$$\mathbf{Adv}^{\text{ind-atk}}_{\Pi,A_{\mathrm{atk}}}(k) = 2 \cdot \Pr[I \leftarrow \mathcal{G}(1^k);\ (pk, sk) \leftarrow \mathcal{K}(I);\ (m_0, m_1, \mathsf{si}) \leftarrow A^{\mathcal{O}_{\mathrm{atk}}}_{\mathrm{atk}}(\mathsf{find}, pk); \ b \xleftarrow{R} \{0,1\};\ c \leftarrow \mathcal{E}_{pk}(m_b) \ : \ A^{\mathcal{O}_{\mathrm{atk}}}_{\mathrm{atk}}(\mathsf{guess}, c, \mathsf{si}) = b] - 1$$

where $\mathcal{O}_{\mathrm{cpa}} = \epsilon$ *and* $\mathcal{O}_{\mathrm{cca}} = \mathcal{D}_{sk}$*. Note that* si *is the state information. It contains the public key* pk*, the messages* m_0 *and* m_1*, and so on. We require that* $m_0 \neq m_1$ *and* $m_0, m_1 \in \mathtt{MSPC}(pk)$*. We also require that* A_{cca} *never queries the challenge* c *to* $\mathcal{D}_{sk}$ *in the guess stage.*

We say that A_{cpa} *is a* (t, ϵ)*-adversary for* Π *in the sense of IND-CPA if* A_{cpa} *runs in at most time* t *and achieves* $\mathbf{Adv}^{\text{ind-cpa}}_{\Pi, A_{\mathrm{cpa}}}(k) \geq \epsilon$*.*

Similarly, we say that A_{cca} *is a* (t, q_d, ϵ)*-adversary for* Π *in the sense of IND-CCA if* A_{cca} *runs in at most time* t*, asks at most* q_d *queries to decryption oracle* $\mathcal{D}_{sk}$*, and achieves* $\mathbf{Adv}^{\text{ind-cca}}_{\Pi, A_{\mathrm{cca}}}(k) \geq \epsilon$*.*

We say that Π *is* (t, ϵ)*-secure (respectively* (t, q_d, ϵ)*-secure) in the sense of IND-CPA (resp. IND-CCA) if there is no* (t, ϵ)*-adversary (resp.* (t, q_d, ϵ)*-adversary) for* Π *in the corresponding sense.*

Indistinguishability in the Random Oracle Model. We can consider the definition of the indistinguishability in the random oracle model in a similar way as that in the standard model described above.

We define Ω as the map family from an appropriate range. The domain and range depend on the underlying encryption scheme. Even if we choose two random functions that have distinct domains and distinct ranges respectively, we just write the experiment, for convenience, as $G, H \leftarrow \Omega$, instead of preparing two map families.

In the random oracle model, we begin the experiment of A_{atk} described above (which defines advantage) by $H \leftarrow \Omega$. Then, we add the random oracle H to both $\mathcal{O}_{\mathrm{cpa}}$ and $\mathcal{O}_{\mathrm{cca}}$, and allow that $\mathcal{E}_{pk}$ and $\mathcal{D}_{sk}$ may depend on H (which we write $\mathcal{E}^H_{pk}$ and $\mathcal{D}^H_{sk}$, respectively).

We define the adversaries in a similar way as those in the standard model, that is, we define a (t, q_h, ϵ)-adversary in the sense of IND-CPA in the random oracle model and a (t, q_h, q_d, ϵ)-adversary in the sense of IND-CCA in the random oracle model where the adversary makes at most q_h queries to H.

We say that Π is (t, q_h, ϵ)-secure (respectively (t, q_h, q_d, ϵ)-secure) in the sense of IND-CPA (resp. IND-CCA) in the random oracle model if there is no (t, q_h, ϵ)-adversary (resp. (t, q_h, q_d, ϵ)-adversary) for Π in the corresponding sense in the random oracle model.

Knowledge Extractor and Plaintext Awareness

The notion of knowledge extractor and plaintext awareness for a public-key encryption scheme is defined in [7, 4]. We describe the definitions by Bellare, Desai, Pointcheval, and Rogaway [4].

Definition 8.4 (Knowledge Extractor and Plaintext Awareness)**.** *Let* $\Pi = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *be a public-key encryption scheme. Let* B *and* K *be algorithms, called adversary and knowledge extractor, respectively. They work in the random oracle model as follows:*

- *B is a (q_h, q_e)-adversary that takes a public-key pk and makes queries at most q_h and q_e times to the random oracle H and the encryption oracle $\mathcal{E}_{pk}^H$, respectively. B finally outputs $c \notin C$, where*
  - *T_H denotes the set of all pairs of B's queries and the corresponding answers from H,*
  - *C denotes the set of all answers from $\mathcal{E}_{pk}^H$.*

  We write the above experiment as $(T_H, C, c, pk) \leftarrow \mathtt{run}\ B^{H,\mathcal{E}_{pk}^H}(pk)$.
- *Knowledge extractor K takes (T_H, C, c, pk) and output a string m.*

For any $k \in \mathbb{N}$, we define

$$\mathbf{Succ}^{\mathrm{pa}}_{K,B,\Pi}(k) = \Pr[H \leftarrow \Omega;\ I \leftarrow \mathcal{G}(1^k);\ (pk, sk) \leftarrow \mathcal{K}(I);\\ (T_H, C, c, pk) \leftarrow \mathtt{run}\ B^{H,\mathcal{E}_{pk}^H}(pk)\ :\ K(T_H, C, c, pk) = \mathcal{D}_{sk}^H(c)].$$

We say that K is a $(t_{\mathrm{KE}}, \lambda, q_h, q_e)$-knowledge extractor for PA of Π if for any (q_h, q_e)-adversary B, K runs in at most time t_{KE} and achieves $\mathbf{Succ}^{\mathrm{pa}}_{K,B,\Pi}(k) \geq \lambda$.

We say that Π is $(t_{\mathrm{cpa}}, t_{\mathrm{KE}}, q_h, q_e, \epsilon, \lambda)$-secure in the sense of PA if Π is $(t_{\mathrm{cpa}}, q_h, \epsilon)$-secure in the sense of IND-CPA, and there exists a $(t_{\mathrm{KE}}, \lambda, q_h, q_e)$-knowledge extractor K for PA of Π.

Bellare, Desai, Pointcheval, and Rogaway [4] showed that if the public-key encryption scheme is secure in the sense of PA, then it is also secure in the sense of IND-CCA.

Anonymity

We describe the definition of the anonymity, following [3].

Definition 8.5 (IK-CPA, IK-CCA [3])**.** *Let $\Pi = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme. Let A_{cpa} and A_{cca} be adversaries that run in two stages,* find *and* guess*. The adversaries A_{cpa} and A_{cca} have access to some oracles $\mathcal{O}_{\mathrm{cpa}}$ and O_{cca}, respectively. For atk $\in \{$cpa, cca$\}$, we define the advantages of A_{atk} via*

$$\mathbf{Adv}^{\text{ik-atk}}_{\Pi,A_{\mathrm{atk}}}(k) = 2 \cdot \Pr[I \leftarrow \mathcal{G}(1^k);\ (pk_0, sk_0), (pk_1, sk_1) \leftarrow \mathcal{K}(I);\\ (m, \mathsf{si}) \leftarrow A_{\mathrm{atk}}^{\mathcal{O}_{\mathrm{atk}}}(\mathsf{find}, pk_0, pk_1);\ b \xleftarrow{R} \{0,1\};\ c \leftarrow \mathcal{E}_{pk_b}(m)\ :\ A_{\mathrm{atk}}^{\mathcal{O}_{\mathrm{atk}}}(\mathsf{guess}, c, \mathsf{si}) = b] - 1$$

where $\mathcal{O}_{\mathrm{cpa}} = \epsilon$ and $\mathcal{O}_{\mathrm{cca}} = (\mathcal{D}_{sk_0}, \mathcal{D}_{sk_1})$. Note that si *is the state information. It contains the public keys pk_0, pk_1, the message m, and so on. We require that $m \in \mathtt{MSPC}(pk_0) \cap \mathtt{MSPC}(pk_1)$. We also require that A_{cca} never queries the challenge c to either $\mathcal{D}_{sk_0}$ or $\mathcal{D}_{sk_1}$ in the guess stage.*

We say that A_{cpa} *is a* (t, ϵ)*-adversary for* Π *in the sense of IK-CPA if* A_{cpa} *runs in at most time* t *and achieves* $\mathbf{Adv}^{\text{ik-cpa}}_{\Pi, A_{\mathrm{cpa}}}(k) \geq \epsilon$.

Similarly, we say that A_{cca} *is a* (t, q_d, ϵ)*-adversary for* Π *in the sense of IK-CCA if* A_{cca} *runs in at most time* t*, makes a total number of* q_d *queries to decryption oracles* $\mathcal{D}_{sk_0}$ *and* $\mathcal{D}_{sk_1}$*, and achieves* $\mathbf{Adv}^{\text{ik-cca}}_{\Pi, A_{\mathrm{cca}}}(k) \geq \epsilon$.

We say that Π *is* (t, ϵ)*-secure (respectively* (t, q_d, ϵ)*-secure) in the sense of IK-CPA (resp. IK-CCA) if there is no* (t, ϵ)*-adversary (resp.* (t, q_d, ϵ)*-adversary) for* Π *in the corresponding sense.*

Anonymity in the Random Oracle Model. We can consider the definition of the anonymity in the random oracle model in a similar way as that in the standard model described above.

We define Ω as the map family from an appropriate range. The domain and range depend on the underlying encryption scheme. Even if we choose two random functions that have distinct domains and distinct ranges respectively, we just write the experiment, for convenience, as $G, H \leftarrow \Omega$, instead of preparing two map families.

In the random oracle model, we begin the experiment of A_{atk} described above (which defines advantage) by $H \leftarrow \Omega$. Then, we add the random oracle H to both $\mathcal{O}_{\mathrm{cpa}}$ and $\mathcal{O}_{\mathrm{cca}}$, and allow that for $i \in \{0, 1\}$, $\mathcal{E}_{pk_i}$ and $\mathcal{D}_{sk_i}$ may depend on H (which we write $\mathcal{E}^H_{pk_i}$ and $\mathcal{D}^H_{sk_i}$, respectively).

We define the adversaries in a similar way as those in the standard model, that is, we define a (t, q_h, ϵ)-adversary in the sense of IK-CPA in the random oracle model and a (t, q_h, q_d, ϵ)-adversary in the sense of IK-CCA in the random oracle model where the adversary makes at most q_h queries to H.

We say that Π is (t, q_h, ϵ)-secure (respectively (t, q_h, q_d, ϵ)-secure) in the sense of IK-CPA (resp. IK-CCA) in the random oracle model if there is no (t, q_h, ϵ)-adversary (resp. (t, q_h, q_d, ϵ)-adversary) for Π in the corresponding sense in the random oracle model.

8.1.2 Symmetric-Key Encryption

The Definition of Symmetric-Key Encryption

We review the definition of symmetric-key encryption schemes.

Definition 8.6. *A symmetric-key encryption scheme* $\Pi = (\mathcal{E}, \mathcal{D})$ *consists of two algorithms.*

- *The encryption algorithm* $\mathcal{E}_x(m)$ *is a deterministic algorithm that takes a symmetric-key* $x \in \mathtt{KSPC}(k)$ *and a message* $m \in \mathtt{MSPC}(k)$*, and returns a ciphertext* c*. Note that*

$\mathtt{KSPC}(k)$ *and* $\mathtt{MSPC}(k)$ *are the key space and the message space for* k*, respectively. They are uniquely determined by a security parameter* 1^k*.*

- *The decryption algorithm* $\mathcal{D}_x(c)$ *is a deterministic algorithm that takes a symmetric key* x *and a ciphertext* c*, and returns the corresponding plaintext* m*.*

We require that, for any $k \in \mathbb{N}$*, if* $x \in \mathtt{KSPC}(k)$*,* $m \in \mathtt{MSPC}(k)$*, and* $c \leftarrow \mathcal{E}_x(m)$*, then* $m = \mathcal{D}_x(c)$*.*

Find-Guess

We review a security notion for symmetric-key encryption, called find-guess (FG), following [42].

Definition 8.7 (FG)**.** *Let* $\Pi = (\mathcal{E}, \mathcal{D})$ *be a symmetric-key encryption scheme. Let* A *be an adversary that runs in two stages,* find *and* guess*. We define the advantage of* A *via*

$$\mathbf{Adv}^{\mathrm{fg}}_{\Pi}(k) = 2 \cdot \Pr[x \xleftarrow{R} \mathtt{KSPC}(k);\ (m_0, m_1, \mathsf{si}) \leftarrow A(\mathsf{find}, k);\\ b \xleftarrow{R} \{0,1\};\ c \leftarrow \mathcal{E}_x(m_b)\ :\ A(\mathsf{guess}, c, \mathsf{si}) = b] - 1.$$

We require that $m_0 \neq m_1$ *and* $m_0, m_1 \in \mathtt{MSPC}(k)$*.*

We say that A *is a* (t, ϵ)*-adversary for* Π *in the sense of FG if* A *runs in at most time* t *and achieves* $\mathbf{Adv}^{\mathrm{fg}}_{\Pi,A}(k) \geq \epsilon$*.*

We say that Π *is* (t, ϵ)*-secure in the sense of FG if there is no* (t, ϵ)*-adversary for* Π *in the sense of FG.*

8.2 Plaintext Awareness in the Two-Key Setting

In this section, we propose the notion of plaintext awareness in the two-key setting (PATK), and prove that PATK implies IK-CCA.

We describe the definition of plaintext awareness in the two-key setting.

Definition 8.8 (Plaintext Awareness in the two-key setting and Knowledge Extractor for PATK)**.** *Let* $\Pi = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *be a public-key encryption scheme. Let* B *and* K *be algorithms, called an adversary for PATK and a knowledge extractor for PATK, respectively. They work in the random oracle model as follows:*

- B *is a* (q_h, q_e)*-adversary for PATK that takes two public-keys* pk_0, pk_1 *and an index* $i \in \{0,1\}$*, and makes at most* q_h *queries to* H *and* q_e *queries to the encryption oracles,* $\mathcal{E}^H_{pk_0}$ *and* $\mathcal{E}^H_{pk_1}$*.* B *finally outputs* $c \notin C$*, where*
  - T_H *denotes the set of all pairs of a* B*'s query and the corresponding answer from* H*, and*

– *C denotes the set of all answers from $\mathcal{E}^H_{pk_0}$ and $\mathcal{E}^H_{pk_1}$. (Note that C does not contain an information of which encryption oracle responded.)*

We write this experiment as $(T_H, C, c, pk_i) \leftarrow \mathtt{run}\ B^{H,\mathcal{E}^H_{pk_0},\mathcal{E}^H_{pk_1}}(pk_0, pk_1, i)$.

- *Knowledge extractor K for PATK takes (T_H, C, c, pk_i) and outputs a string m.*

For any $k \in \mathbb{N}$ and $i \in \{0,1\}$, we define

$$\mathbf{Succ}^{\mathrm{patk}}_{K,B,\Pi,i}(k) = \Pr[H \leftarrow \Omega;\ I \leftarrow \mathcal{G}(1^k);\ (pk_0, sk_0), (pk_1, sk_1) \leftarrow \mathcal{K}(I);\\ (T_H, C, c, pk_i) \leftarrow \mathtt{run}\ B^{H,\mathcal{E}^H_{pk_0},\mathcal{E}^H_{pk_1}}(pk_0, pk_1, i)\ :\ K(T_H, C, c, pk_i) = \mathcal{D}^H_{sk_i}(c)].$$

We say that K is a $(t_{\mathrm{KETK}}, \lambda, q_h, q_e)$-knowledge extractor for PATK of Π if for any (q_h, q_e)-adversary B and for any index $i \in \{0,1\}$, K runs in at most time t_{KETK} and achieves $\mathbf{Succ}^{\mathrm{patk}}_{K,B,\Pi,i}(k) \geq \lambda$.

We say that Π is $(t_{\mathrm{cpa}}, t_{\mathrm{KETK}}, q_h, q_e, \epsilon, \lambda)$-secure in the sense of PATK if Π is $(t_{\mathrm{cpa}}, q_h, \epsilon)$-secure in the sense of IK-CPA, and there exists a $(t_{\mathrm{KETK}}, \lambda, q_h, q_e)$-knowledge extractor K for PATK of Π.

There are some differences between the definition of PA in [4] and that of PATK. First, the adversary B in our definition receives two public keys and two encryption oracles, while the adversary in the definition of PA receives one public key and one encryption oracle. Second, we define the success probability of B for any index $i \in \{0,1\}$. This indicates under which key, pk_0 or pk_1, the knowledge extractor K for PATK should decrypt c. Third, in the definition of PA, the list C contains the answers (ciphertexts) from only one encryption oracle $\mathcal{E}^H_{pk}$. When we prove that PA implies IND-CCA, C plays an important role, that is, C contains the challenge ciphertext of IND-CCA game to give it to the adversary B for PA. In our definition, if we use C to prove that PATK implies IK-CCA, C has to contain the challenge ciphertext of IK-CCA game and the challenge ciphertext is encrypted by either pk_0 or pk_1. Therefore, in our definition, we define that the list C consists of the answers (ciphertexts) from *both* $\mathcal{E}^H_{pk_0}$ *and* $\mathcal{E}^H_{pk_1}$.

It is easy to see that if there exists a knowledge extractor K for PATK of Π, then we can use K as a knowledge extractor for PA of Π. That is, if the public-key encryption scheme Π is secure in the sense of PATK and IND-CPA, then Π is secure in the sense of PA. However, it is not clear that we can use the knowledge extractor for PA of Π as that for PATK of Π. The difficulty of proving this seems to depend on the third difference described above.

We prove the following theorem.

Theorem 8.1. *If the public encryption scheme Π is $(t_{\mathrm{cpa}}, t_{\mathrm{KETK}}, q_h, 1, \epsilon, \lambda)$-secure in the sense of PATK, then Π is $(t_{\mathrm{cca}}, q_h, q_d, \epsilon')$-secure in the sense of IK-CCA where*

$$t_{\mathrm{cca}} = t_{\mathrm{cpa}} - q_d \cdot t_{\mathrm{KETK}} \text{ and } \epsilon' = \epsilon + 2q_d \cdot (1 - \lambda).$$

Proof. In [4], Bellare, Desai, Pointcheval, and Rogaway proved that PA implies IND-CCA. We prove Theorem 8.1 in a similar way.

Let A_{cca} be an $(t_{\mathrm{cca}}, q_h, q_d, \epsilon)$-adversary of Π in the sense of IK-CCA. We construct an adversary A_{cpa} of Π in the sense of IK-CPA by using A_{cca}.

We construct the algorithm A_{cpa} as follows. Note that A_{cpa} simulates A_{cca}'s oracles H, $\mathcal{D}_{sk_0}$, and $\mathcal{D}_{sk_1}$ as described below.

1) A_{cpa} initializes two lists, T_H and C to empty.

2) $A_{\mathrm{cpa}}(\mathsf{find}, pk_0, pk_1)$ runs A_{cca} as $(m, \mathsf{si}) \leftarrow A_{\mathrm{cca}}(\mathsf{find}, pk_0, pk_1)$ and outputs (m, si).

3) A_{cpa} receives a challenge ciphertext $\hat{c} = \mathcal{E}^H_{pk_b}(m)$ where $b \stackrel{R}{\leftarrow} \{0, 1\}$.

4) $A_{\mathrm{cpa}}(\mathsf{guess}, \hat{c})$ runs A_{cca} as $d \leftarrow A_{\mathrm{cca}}(\mathsf{guess}, \hat{c})$ and outputs d.

A_{cpa} simulates A_{cca}'s oracle as follows:

- When A_{cca} makes a query h to H, A_{cpa} makes a query h to *its* oracle H and obtains an answer $H(h)$. Then, A_{cpa} returns $H(h)$ to A_{cca} and puts $(h, H(h))$ into the list T_H.
- When A_{cca} makes a decryption query c to $\mathcal{D}^H_{sk_i}$, A_{cpa} runs the knowledge extractor K as follows.
  - In the find stage, A_{cpa} runs K as $m \leftarrow K(T_H, \epsilon, c, pk_i)$ and returns m to A_{cca}.
  - In the guess stage, A_{cpa} runs K as $m \leftarrow K(T_H, \hat{c}, c, pk_i)$ and returns m to A_{cca}.

To guarantee that the knowledge extractor K for PATK outputs a correct answer (a corresponding plaintext m or an invalid symbol $\perp$), for $j \in \{1, 2, \cdots, q_d\}$ we construct the adversary B_j for PATK as follows. Note that B_j simulates A_{cca}'s oracles H, $\mathcal{D}_{sk_0}$, and $\mathcal{D}_{sk_1}$ as described below. Note that $B_j(pk_0, pk_1, i)$ returns some value and halts when A_{cca} makes its j-th decryption query.

1) B_j initializes two lists, T_H and C to empty.

2) B_j runs A_{cca} as $(m, \mathsf{si}) \leftarrow A_{\mathrm{cca}}(\mathsf{find}, pk_0, pk_1)$.

3) B_j picks a random bit $b \stackrel{R}{\leftarrow} \{0, 1\}$ and makes an oracle query as $\hat{c} \leftarrow \mathcal{E}^H_{pk_b}(m)$.

4) B_j runs $A_{\mathrm{cca}}(\mathsf{guess}, \hat{c})$. (Note that B_j is sure to halt before A_{cca} outputs d. See below.)

$B_j(pk_0, pk_1, i)$ simulates A_{cca}'s oracle as follows:

- When A_{cca} makes a query h to H, A_{cpa} makes a query h to *its* oracle H and obtains an answer $H(h)$. Then, A_{cpa} returns $H(h)$ to A_{cca} and puts $(h, H(h))$ into the list T_H.
- When A_{cca} makes a j'-th decryption query c to $\mathcal{D}^H_{sk_i}$, A_{cpa} runs the knowledge extractor K as follows.
  - In the find stage, if $j' = j$ then B_j returns c and halts; otherwise, A_{cpa} runs K as $m \leftarrow K(T_H, \epsilon, c, pk_i)$ and returns m to A_{cca}.
  - In the guess stage, if $j' = j$ then B_j returns c and halts; otherwise, A_{cpa} runs K as $m \leftarrow K(T_H, \hat{c}, c, pk_i)$ and returns m to A_{cca}.

Since $j \leq q_d$ and A_{cca} makes at most q_d queries to the decryption oracles, B_j is sure to output c and halt before A_{cca} outputs d in the guess stage.

We analyze the success probability of A_{cpa}. We have that for any $j \in \{1, 2, \cdots, q_d\}$ the distribution of $(T_H, C, c, pk_i) \leftarrow \mathtt{run}\ B_j^{H, \mathcal{E}^H_{pk_0}, \mathcal{E}^H_{pk_1}}(pk_0, pk_1, i)$ where

$$H \leftarrow \Omega;\ I \leftarrow \mathcal{G}(1^k);\ (pk_0, sk_0), (pk_1, sk_1) \leftarrow \mathcal{K}(I)$$

and the distribution of the j-th input for K in the above adversary A_{cpa} is identical. Therefore,

$$\Pr[A_{\mathrm{cpa}}(\mathsf{find}, pk_0, pk_1) = A_{\mathrm{cca}}(\mathsf{find}, pk_0, pk_1)] \geq 1 - q_d^{\mathrm{find}} \cdot (1 - \lambda)$$

and

$$\begin{aligned}\Pr[&A_{\mathrm{cpa}}(\mathsf{guess}, c, (\mathsf{si}, T_H)) = A_{\mathrm{cca}}(\mathsf{guess}, c, \mathsf{si}) \\ &|A_{\mathrm{cpa}}(\mathsf{find}, pk_0, pk_1) = A_{\mathrm{cca}}(\mathsf{find}, pk_0, pk_1)] \geq 1 - (q_d - q_d^{\mathrm{find}}) \cdot (1 - \lambda)\end{aligned}$$

where q_d^{find} is a number of decryption queries of A_{cca} in the find stage. Hence, $\epsilon' \geq \epsilon - 2q_d(1 - \lambda)$.

It is easy to see that the running time of A_{cpa} is less than $t_{\mathrm{cca}} + q_d \cdot t_{\mathrm{KETK}}$. □

8.3 The Fujisaki–Okamoto Conversion

In this section, we review the conversion proposed by Fujisaki and Okamoto [42].

Let $\Pi^{\mathrm{pub}} = (\mathcal{G}^{\mathrm{pub}}, \mathcal{K}^{\mathrm{pub}}, \mathcal{E}^{\mathrm{pub}}, \mathcal{D}^{\mathrm{pub}})$ be a public-key encryption scheme and let $\Pi^{\mathrm{sym}} = (\mathcal{E}^{\mathrm{sym}}, \mathcal{D}^{\mathrm{sym}})$ be a symmetric-key encryption scheme. Let $G : \mathtt{MSPC}^{\mathrm{pub}} \rightarrow \mathtt{KSPC}^{\mathrm{sym}}$ and $H : \mathtt{MSPC}^{\mathrm{pub}} \times \mathtt{MSPC}^{\mathrm{sym}} \rightarrow \mathtt{COINS}^{\mathrm{pub}}$ be hash functions.

A public-key encryption scheme $\Pi^{\mathrm{hy}} = (\mathcal{G}^{\mathrm{hy}}, \mathcal{K}^{\mathrm{hy}}, \mathcal{E}^{\mathrm{hy}}, \mathcal{D}^{\mathrm{hy}})$ derived from the Fujisaki-Okamoto conversion is as follows:

- Common key generation and key generation: $\mathcal{G}^{\mathrm{hy}}$ and $\mathcal{K}^{\mathrm{hy}}$ are the same as $\mathcal{G}^{\mathrm{pub}}$ and $\mathcal{K}^{\mathrm{pub}}$, respectively.
- Encryption:
$$\mathcal{E}^{\mathrm{hy}}_{pk}(m;\sigma) = \mathcal{E}^{\mathrm{pub}}_{pk}(\sigma; H(\sigma,m)) \,||\, \mathcal{E}^{\mathrm{sym}}_{G(\sigma)}(m)$$
where $\mathtt{COINS}^{\mathrm{hy}} = \mathtt{MSPC}^{\mathrm{pub}}$ and $\mathtt{MSPC}^{\mathrm{hy}} = \mathtt{MSPC}^{\mathrm{sym}}$.
- Decryption:
$$\mathcal{D}^{\mathrm{hy}}_{sk}(c_1||c_2) = \begin{cases} \hat{m} & \texttt{if } c_1 = \mathcal{E}^{\mathrm{pub}}_{pk}(\hat{\sigma}; H(\hat{\sigma},\hat{m})) \\ \perp & \texttt{otherwise} \end{cases}$$
where $\hat{\sigma} \leftarrow \mathcal{D}^{\mathrm{pub}}_{sk}(c_1)$ and $\hat{m} \leftarrow \mathcal{D}^{\mathrm{sym}}_{G(\hat{\sigma})}(c_2)$.

Fujisaki and Okamoto showed that the public-key encryption scheme Π^{hy} is secure in the sense of IND-CCA in the random oracle model when

- Π^{pub} is γ-uniform ($\gamma < 1$) and secure in the sense of OW, and
- Π^{sym} is secure in the sense of FG.

8.4 A Generic Conversion for the Anonymity

In this section, we propose the generic conversion for the anonymity, that is, we prove that the public-key encryption scheme derived from the Fujisaki-Okamoto conversion with the following assumptions is secure in the sense of IK-CCA in the random oracle model.

- Π^{pub} use the common message space $\mathtt{MSPC}^{\mathrm{pub}}(I)$ and the common randomness space $\mathtt{COINS}^{\mathrm{pub}}(I)$ as the message space $\mathtt{MSPC}^{\mathrm{pub}}(pk)$ and the randomness space $\mathtt{COINS}^{\mathrm{pub}}(pk)$, respectively, for any public key pk outputted by $K(I)$,
- Π^{pub} is secure in the sense of IK-CPA,
- Π^{pub} is γ-uniform ($\gamma < 1$) and secure in the sense of OW, and
- Π^{sym} is secure in the sense of FG.

Since these conditions are sufficient that Π^{hy} meets IND-CCA, we can get a public-key encryption scheme which is secure in the sense of IND-CCA and IK-CCA in the random oracle model when we assume the above four conditions.

IK-CPA Security. We prove the following lemma with respect to the anonymity property.

Lemma 8.1. *Let* Π^{pub} *be a public-key encryption scheme where* Π^{pub} *uses the common message space* $\mathtt{MSPC}^{\mathrm{pub}}(I)$ *and the common randomness space* $\mathtt{COINS}^{\mathrm{pub}}(I)$ *as the message space* $\mathtt{MSPC}^{\mathrm{pub}}(pk)$ *and the randomness space* $\mathtt{COINS}^{\mathrm{pub}}(pk)$*, respectively, for any public key* pk *outputted by* $K(I)$*.*

Suppose that Π^{pub} *is* (t_1, ϵ_1)*-secure in the sense of IK-CPA, and* (t_2, ϵ_2)*-secure in the sense of OW. Let* ℓ_2 *be the size of* $\mathtt{MSPC}^{\mathrm{sym}}$*. Then,* Π^{hy} *is* (t, q_g, q_h, ϵ)*-secure in the sense of IK-CPA in the random oracle model, where* $t = \min\{t_1, t_2\} - poly(\ell_2)$ *and* $\epsilon = \epsilon_1 + 2(q_g + q_h) \cdot \epsilon_2$*.*

Remark 8.1. *Note that IK-CPA does not imply OW. For example, let* $\Pi = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *be a public-key encryption scheme which is secure in the sense of IK-CPA. Then, consider the public-key encryption scheme* Π' *whose encryption algorithm is defined as* $\mathcal{E}'_{pk}(m) := \mathcal{E}_{pk}(m)||m$*. We can easily see that* Π' *meets IK-CPA, and does not meet OW.*

Proof. Suppose that A is a (t, q_g, q_h, ϵ)-adversary for Π^{hy} in the sense of IK-CPA in the random oracle model. We show that there exists a (t_1, ϵ_1)-adversary B for Π^{pub} in the sense of IK-CPA and a (t_2, ϵ_2)-adversary C for Π^{pub} in the sense of OW, where $t = \min\{t_1, t_2\} - poly(\ell_2)$ and $\epsilon = \epsilon_1 + 2(q_g + q_h) \cdot \epsilon_2$.

We construct the adversaries B and C by using the adversary A. B and C have to simulate the random oracles G and H for A. We describe how to simulate the random oracles in both B and C. We use the lists $\mathcal{T}_G$ and $\mathcal{T}_H$ which are initially empty lists.

- *The simulation of* G*.* For a query σ, if there exist an entry $(\sigma', g') \in \mathcal{T}_G$ such that $\sigma = \sigma'$, it returns g' to A. Otherwise, it picks a string $g \xleftarrow{R} \mathtt{KSPC}^{\mathrm{sym}}(k)$, returns g to A, and puts (σ, g) on the list $\mathcal{T}_G$.

- *The simulation of* H*.* For a query (σ, m), if there exist an entry $(\sigma', m', h') \in \mathcal{T}_H$ such that $\sigma = \sigma'$ and $m = m'$, it returns h' to A. Otherwise, it picks a string $h \xleftarrow{R} \mathtt{COINS}^{\mathrm{pub}}(I)$, returns h to A, and puts (σ, m, h) on the list $\mathcal{T}_H$.

We construct the adversary B in the sense of IK-CPA as follows.

$\mathtt{Algorithm}\ B(\mathsf{find}, pk_0, pk_1)$	$\mathtt{Algorithm}\ B(\mathsf{guess}, c, \mathsf{si}')$
$(m, \mathsf{si}) \leftarrow A(\mathsf{find}, pk_0, pk_1)$	$x \xleftarrow{R} \mathtt{KSPC}^{\mathrm{sym}}(k)$
$\sigma \xleftarrow{R} \mathtt{MSPC}^{\mathrm{pub}}(I)$	$c' \leftarrow c\|\|\mathcal{E}^{\mathrm{sym}}_x(m)$
$\mathsf{si}' \leftarrow (\mathsf{si}, m)$	$b' \leftarrow A(\mathsf{guess}, c, \mathsf{si})$
$\mathtt{return}\ (\sigma, \mathsf{si}')$	$\mathtt{return}\ b'$

We construct the adversary C in the sense of OW as follows.

$$
\begin{array}{l}
\texttt{Algorithm } C(c, pk) \\
\quad (pk', sk') \leftarrow \mathcal{K}^{\mathrm{pub}}(I) \\
\quad d \xleftarrow{R} \{0,1\};\ pk_d \leftarrow pk;\ pk_{1-d} \leftarrow pk' \\
\quad (m, \mathsf{si}) \leftarrow A(\mathsf{find}, pk_0, pk_1) \\
\quad b \xleftarrow{R} \{0,1\};\ x \xleftarrow{R} \texttt{KSPC}^{\mathrm{sym}}(k);\ c' \leftarrow c || \mathcal{E}_x^{\mathrm{sym}}(m) \\
\quad b' \leftarrow A(\mathsf{guess}, c') \\
\quad \hat{\sigma} \xleftarrow{R} \{\sigma' | (\sigma', g') \in \mathcal{T}_G \text{ or } (\sigma', m', h') \in \mathcal{T}_H\} \\
\quad \texttt{return } \hat{\sigma}
\end{array}
$$

It is easy to see that the running times of B and C is at most that of A plus the time for computing $\mathcal{E}_x^{\mathrm{sym}}(m)$, that is, $t_1, t_2 < t + poly(\ell_2)$.

We analyze the advantages of B and C. We define the following events.

- $\mathsf{AskA} = [A$ asks σ to the oracle G or asks (σ, m) to the oracle H where the challenge ciphertext is $c' = \mathcal{E}_{pk_b}^{\mathrm{pub}}(\sigma; H(\sigma, m)) || \mathcal{E}_{G(\sigma)}^{\mathrm{sym}}(m).]$
- $\mathsf{SuccA} = [G, H \leftarrow \Omega;\ I \leftarrow \mathcal{G}^{\mathrm{hy}}(1^k);\ (pk_0, sk_0), (pk_1, sk_1) \leftarrow \mathcal{K}^{\mathrm{hy}}(I);$
  $(m, \mathsf{si}) \leftarrow A^{G,H}(\mathsf{find}, pk);\ b \xleftarrow{R} \{0,1\};\ c' \leftarrow \mathcal{E}_{pk_b}^{\mathrm{hy}}(m)\ :\ A^{G,H}(\mathsf{guess}, c', \mathsf{si}) = b]$
- $\mathsf{SuccB} = [I \leftarrow \mathcal{G}^{\mathrm{pub}}(1^k);\ (pk_0, sk_0), (pk_1, sk_1) \leftarrow \mathcal{K}^{\mathrm{pub}}(I);$
  $(\sigma, \mathsf{si}) \leftarrow B(\mathsf{find}, pk);\ b \xleftarrow{R} \{0,1\};\ c \leftarrow \mathcal{E}_{pk_b}^{\mathrm{pub}}(\sigma)\ :\ B(\mathsf{guess}, c, \mathsf{si}) = b]$
- $\mathsf{SuccC} = [I \leftarrow \mathcal{G}^{\mathrm{pub}}(1^k);\ (pk, sk) \leftarrow \mathcal{K}^{\mathrm{pub}}(I);\ \sigma \xleftarrow{R} \texttt{MSPC}^{\mathrm{pub}}(pk)$
  $c \leftarrow \mathcal{E}_{pk}^{\mathrm{pub}}(\sigma)\ :\ C(c, pk) = \sigma]$

In the experiment of B, if the event $\neg\mathsf{AskA}$ holds, the view of A simulated in B is identical to the real A's view. Therefore, $\Pr[\mathsf{SuccB}] \geq \Pr[\mathsf{SuccA} | \neg\mathsf{AskA}] \cdot \Pr[\neg\mathsf{AskA}]$.

In the experiment of C, if the event AskA holds, there exist a string σ such that $c = \mathcal{E}_{pk_b}^{\mathrm{pub}}(\sigma)$ in $\{\sigma' | (\sigma', g') \in \mathcal{T}_G \text{ or } (\sigma', m', h') \in \mathcal{T}_H\}$ and C can output the correct answer with probability at least $1/(q_G + q_H)$. Furthermore, if $b = d$ holds, the probability that C asks such σ is the same as the probability that the real A asks such σ. Therefore, $\Pr[\mathsf{SuccC}] \geq \Pr[b = d] \times \Pr[\mathsf{SuccC} | b = d] \geq 1/(2(q_G + q_H)) \cdot \Pr[\mathsf{AskA}]$.

Hence, we have

$$
\begin{aligned}
\Pr[\mathsf{SuccA}] &= \Pr[\mathsf{SuccA} | \neg\mathsf{AskA}] \cdot \Pr[\neg\mathsf{AskA}] + \Pr[\mathsf{SuccA} | \mathsf{AskA}] \cdot \Pr[\mathsf{AskA}] \\
&\leq \Pr[\mathsf{SuccA} | \neg\mathsf{AskA}] \cdot \Pr[\neg\mathsf{AskA}] + \Pr[\mathsf{AskA}] \\
&\leq \Pr[\mathsf{SuccB}] + 2(q_G + q_H) \cdot \Pr[\mathsf{SuccC}].
\end{aligned}
$$

Since $\epsilon = 2 \cdot \Pr[\mathsf{SuccA}] - 1$, $\epsilon_1 = 2 \cdot \Pr[\mathsf{SuccB}] - 1$, and $\epsilon_2 = \Pr[\mathsf{SuccC}]$, we have $\epsilon \leq \epsilon_1 + (q_G + q_H) \cdot \epsilon_2$. □

Knowledge Extractor for PATK. We show the existence of the knowledge extractor for PATK of our scheme.

Though we mentioned that we could not use the knowledge extractor for PA directly as that for PATK, fortunately, we can use the knowledge extractor for PA as that for PATK in the case of the Fujisaki-Okamoto conversion.

We show the following lemma.

Lemma 8.2. *Suppose that* Π^{pub} *is* γ*-uniform and* (t_2, ϵ_2)*-secure in the sense of OW. Suppose that* Π^{sym} *is* (t_3, ϵ_3)*-secure in the sense of FG. Let* ℓ_1 *and* ℓ_2 *be the sizes of* $\mathtt{MSPC}^{\mathrm{pub}}$ *and* $\mathtt{MSPC}^{\mathrm{sym}}$*, respectively. Then, there exist a* $(t, \lambda, q_g, q_h, q_e)$*-knowledge extractor* K *for PATK of* Π^{hy} *such that* $t = (q_g + q_h) \cdot poly(\ell_1 + \ell_2)$ *and* $\lambda = 1 - 2q_e \cdot \epsilon_2 - 2\epsilon_3 - \gamma - 2^{-\ell_2}$*.*

Proof. The construction of the knowledge extractor for PATK is the same as that for PA in [42]. We first describe the knowledge extractor $K(T_G, T_H, C, c, pk)$ as follows. Here, let $T_G = \{(\sigma_i, g_i) | i = 1, \ldots, q_g\}$ and $T_H = \{(\sigma'_j, m_j, h_j) | j = 1, \ldots, q_h\}$.

1) Set two empty lists, S_1 and S_2.

2) Find all elements in T_H such that $c_1 = \mathcal{E}^{\mathrm{pub}}_{pk}(\sigma'_j, h_j)$ and put them into list S_1. If $S_1 = \emptyset$, then output $\perp$.

3) For every $(\sigma'_j, m_j, h_j) \in S_1$, find all elements in T_G such that $\sigma_i = \sigma'_j$ and put them (i.e. $(\sigma'_j, m_j, h_j)||(\sigma_i, g_i)$'s) into S_2. If $S_2 = \emptyset$, then output $\perp$.

4) Check in S_2 if there exists a $(\sigma'_j, m_j, h_j)||(\sigma_i, g_i)$ such that $c_2 = \mathcal{E}^{\mathrm{sym}}_{g_i}(m_j)$. If it exists in S_2, then output m_j otherwise output $\perp$.

This protocol runs in $(q_g + q_h) \cdot poly(\ell_1 + \ell_2)$.

Next, we examine the advantage of the knowledge extractor for PATK. We define the following events.

- $\mathsf{Inv0}$ is true if there exists $(c^*_1, c^*_2) \in C$ and $(\sigma_i, g_i) \in T_G$ or $(\sigma_j, m_j, h_j) \in T_H$ such that $\sigma_i = \mathcal{D}^{\mathrm{pub}}_{sk_0}(c^*_1)$ or $\sigma_j = \mathcal{D}^{\mathrm{pub}}_{sk_0}(c^*_1)$.
- $\mathsf{Inv1}$ is true if there exists $(c^*_1, c^*_2) \in C$ and $(\sigma_i, g_i) \in T_G$ or $(\sigma_j, m_j, h_j) \in T_H$ such that $\sigma_i = \mathcal{D}^{\mathrm{pub}}_{sk_1}(c^*_1)$ or $\sigma_j = \mathcal{D}^{\mathrm{pub}}_{sk_1}(c^*_1)$.
- $\mathsf{Inv} = \mathsf{Inv0} \vee \mathsf{Inv1}$.
- $p(S_1)$ true if $S_1 \neq \emptyset$.
- $p(S_2)$ true if $S_2 \neq \emptyset$.
- Find is true if there exists a $(\sigma'_j, m_j, h_j)||(\sigma_i, g_i)$ in S_2 such that $c_2 = \mathcal{E}^{\mathrm{sym}}_{g_i}(m_j)$.

- Fail is true if "the output of knowledge extractor K for PATK" $\neq \mathcal{D}_{sk}^{\mathrm{hy}}(c_1, c_2)$.

We further define the following events:

$$\begin{aligned}
\text{`}\mathtt{1}' &= \mathsf{Inv}. \\
\text{`}\mathtt{00}' &= \neg\mathsf{Inv} \wedge \neg p(S_1). \\
\text{`}\mathtt{010}' &= \neg\mathsf{Inv} \wedge p(S_1) \wedge \neg p(S_2). \\
\text{`}\mathtt{0110}' &= \neg\mathsf{Inv} \wedge p(S_1) \wedge p(S_2) \wedge \neg\mathsf{Find}. \\
\text{`}\mathtt{0111}' &= \neg\mathsf{Inv} \wedge p(S_1) \wedge p(S_2) \wedge \mathsf{Find}.
\end{aligned}$$

We have

$$\begin{aligned}
\Pr[\mathsf{Fail}] &= \Pr[\mathsf{Fail}|\mathtt{1}] \cdot \Pr[\mathtt{1}] + \Pr[\mathsf{Fail}|\mathtt{00}] \cdot \Pr[\mathtt{00}] + \Pr[\mathsf{Fail}|\mathtt{010}] \cdot \Pr[\mathtt{010}] \\
&\quad + \Pr[\mathsf{Fail}|\mathtt{0110}] \cdot \Pr[\mathtt{0110}] + \Pr[\mathsf{Fail}|\mathtt{0111}] \cdot \Pr[\mathtt{0111}] \\
&\leq \Pr[\mathtt{1}] + \Pr[\mathsf{Fail}|\mathtt{00}] + \Pr[\mathsf{Fail}|\mathtt{010}] + \Pr[\mathsf{Fail}|\mathtt{0110}] + \Pr[\mathsf{Fail}|\mathtt{0111}] \\
&= \Pr[\mathtt{1}] + \Pr[\mathsf{Fail}|\mathtt{00}] + \Pr[\mathsf{Fail}|\mathtt{010}].
\end{aligned}$$

We prove the following claim.

Claim 8.1. $\Pr[\mathtt{1}] \leq 2q_e \cdot \epsilon_2$.

Proof. We first consider $\Pr[\mathsf{Inv0}]$. For any $i \in \{0,1\}$, when the adversary B makes a query m to the encryption oracle $\mathcal{E}_{pk_i}^{hy}$, the oracle picks random coins σ and returns $(\mathcal{E}_{pk_i}^{\mathrm{pub}}(\sigma, H(\sigma, m))||\mathcal{E}_{G(\sigma)}^{\mathrm{sym}}(m))$ to B. B makes at most q_e to the encryption oracles. Therefore, $\Pr[\mathtt{Inv0}] \leq q_e \cdot \epsilon_2$. Similarly, we have $\Pr[\mathtt{Inv1}] \leq q_e \cdot \epsilon_2$. Hence, $\Pr[\mathtt{1}] = \Pr[\mathtt{Inv}] \leq 2q_e \cdot \epsilon_2$ □

The proofs of the following claims are the same as those in [42].

Claim 8.2. $\Pr[\mathtt{Fail}|\mathtt{00} \leq \gamma$.

Claim 8.3. $\Pr[\mathtt{Fail}|\mathtt{010}] \leq 2\epsilon_3 + 2^{-\ell_2}$.

Therefore, $\Pr[\mathsf{Fail}] \leq 2q_e \cdot \epsilon_2 + \gamma + 2\epsilon_3 + 2^{-\ell_2}$. Hence,

$$\lambda = 1 - \Pr[\mathsf{Fail}] \geq 1 - (2q_e \cdot \epsilon_2 + \gamma + 2\epsilon_3 + 2^{-\ell_2}).$$

□

From Theorem 8.1 and Lemmas 8.1 and 8.2, we have the following theorem.

Theorem 8.2. *Let Π^{pub} be a public-key encryption scheme where Π^{pub} uses the common message space $\mathtt{MSPC}^{\mathrm{pub}}(I)$ and the common randomness space $\mathtt{COINS}^{\mathrm{pub}}(I)$ as the message space $\mathtt{MSPC}^{\mathrm{pub}}(pk)$ and the randomness space $\mathtt{COINS}^{\mathrm{pub}}(pk)$ for any public key pk outputted by $K(I)$, respectively.*

Suppose that Π^{pub} *is* γ*-uniform,* (t_1, ϵ_1)*-secure in the sense of IK-CPA, and* (t_2, ϵ_2)*-secure in the sense of OW. Suppose that* Π^{sym} *is* (t_3, ϵ_3)*-secure in the sense of FG. Let* ℓ_1 *and* ℓ_2 *be the sizes of* $\mathtt{MSPC}^{\mathrm{pub}}$ *and* $\mathtt{MSPC}^{\mathrm{sym}}$*, respectively. Then,* Π^{hy} *is* $(t, q_g, q_h, q_d, \epsilon)$*-secure in the sense of IK-CCA in the random oracle model where* $t = \min\{t_1, t_2\} - (q_g + q_h) \cdot \mathit{poly}(\ell_1 + \ell_2)$*. and* $\epsilon = \epsilon_1 + 2(q_g + q_h)\epsilon_2 + 2q_d(2\epsilon_2 + 2\epsilon_3 + \gamma + 2^{-\ell_2})$.

CHAPTER 9

Universally Anonymizable Public-Key Encryption

In this chapter, we consider the following situation. In order to send e-mails, all members of the company use the encryption scheme which does not provide the anonymity property. They consider that e-mails sent to the inside of the company do not have to be anonymized and it is sufficient to be encrypted the data. However, when e-mails are sent to the outside of the company, they want to anonymize them for preventing the eavesdropper on the public network.

We propose a solution to solve this as follows. Consider the situation that not only the person who made the ciphertexts, but also anyone can transform the encrypted data to those with the anonymity property without decrypting these encrypted data. If we have this situation, we can make an e-mail gateway which can transform encrypted e-mails to those with the anonymity property without using the corresponding secret key when they are sent to the outside of the company.

In this chapter, in order to formalize this idea, we propose a special type of public-key encryption scheme called a *universally anonymizable public-key encryption scheme*.

The organization of this chapter is as follows. In Section 9.1, we review the definitions of the Decisional Diffie-Hellman problem and the families of hash functions. In Section 9.2, we formulate the notion of universally anonymizable public-key encryption and its security properties. We propose the universally anonymizable public-key encryption scheme based on the ElGamal encryption scheme in Section 9.3, that based on the Cramer-Shoup encryption scheme in Section 9.4, and that based on RSA-OAEP in Section 9.5.

9.1 Preliminaries

The Decisional Diffie-Hellman Problem We review the decisional Diffie-Hellman Problem.

Definition 9.1 (decisional Diffie-Hellman problem)**.** *Let $\mathcal{G}$ be a group generator which takes as input a security parameter k and returns (q, g) where q is a k-bit integer and g is a generator of a cyclic group G_q of order q. Let D be an adversary. We consider the following experiments:*

$\texttt{Experiment } \mathbf{Exp}^{\text{ddh-real}}_{\mathcal{G},D}(k)$	$\texttt{Experiment } \mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D}(k)$
$(q, g) \leftarrow \mathcal{G}(k);\ x, y \xleftarrow{R} \mathbb{Z}_q$	$(q, g) \leftarrow \mathcal{G}(k);\ x, y \xleftarrow{R} \mathbb{Z}_q$
$X \leftarrow g^x;\ Y \leftarrow g^y;\ T \leftarrow g^{xy}$	$X \leftarrow g^x;\ Y \leftarrow g^y;\ T \xleftarrow{R} G_q$
$d \leftarrow D(q, g, X, Y, T)$	$d \leftarrow D(q, g, X, Y, T)$
$\texttt{return } d$	$\texttt{return } d$

The advantage of D in solving the decisional Diffie-Hellman (DDH) problem for $\mathcal{G}$ is defined by

$$\mathbf{Adv}^{\text{ddh}}_{\mathcal{G},D}(k) = \left|\Pr[\mathbf{Exp}^{\text{ddh-real}}_{\mathcal{G},D}(k) = 1] - \Pr[\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D}(k) = 1]\right|.$$

We say that the DDH problem for $\mathcal{G}$ is hard if the function $\mathbf{Adv}^{\text{ddh}}_{\mathcal{G},D}(k)$ is negligible for any algorithm D whose time-complexity is polynomial in k.

The "time-complexity" is the worst case execution time of the experiment plus the size of the code of the adversary, in some fixed RAM model of computation.

Families of Hash Functions We describe the definitions of families of hash functions and universal one-wayness.

Definition 9.2 (families of hash functions)**.** *A family of hash functions $\mathcal{H} = (\mathcal{GH}, \mathcal{EH})$ is defined by two algorithms. A probabilistic generator algorithm $\mathcal{GH}$ takes the security parameter k as input and returns a key K. A deterministic evaluation algorithm $\mathcal{EH}$ takes the key K and a string $M \in \{0,1\}^*$ and returns a string $\mathcal{EH}_K(M) \in \{0,1\}^{k-1}$.*

Definition 9.3 (universal one-wayness)**.** *Let $\mathcal{H} = (\mathcal{GH}, \mathcal{EH})$ be a family of hash functions and let $C = (C_1, C_2)$ be an adversary. We consider the following experiment:*

$\texttt{Experiment } \mathbf{Exp}^{\text{uow}}_{\mathcal{H},C}(k)$
$(x_0, \mathsf{si}) \leftarrow C_1(k);\ K \leftarrow \mathcal{GH}(k);\ x_1 \leftarrow C_2(K, x_0, \mathsf{si})$
$\texttt{if } ((x_0 \neq x_1) \wedge (\mathcal{EH}_K(x_0) = \mathcal{EH}_K(x_1))) \texttt{ then return } 1 \texttt{ else return } 0$

Note that si *is the state information. We define the advantage of C via*

$$\mathbf{Adv}^{\text{uow}}_{\mathcal{H},C}(k) = \Pr[\mathbf{Exp}^{\text{uow}}_{\mathcal{H},C}(k) = 1].$$

We say that the family of hash functions $\mathcal{H}$ is universal one-way if $\mathbf{Adv}^{\mathrm{uow}}_{\mathcal{H},C}(k)$ is negligible for any algorithm C whose time-complexity is polynomial in k.

9.2 Universally Anonymizable Public-Key Encryption

In this section, we propose the definition of universally anonymizable public-key encryption schemes and its security properties.

9.2.1 Definition

We formalize the notion of universally anonymizable public-key encryption schemes as follows.

Definition 9.4. *A universally anonymizable public-key encryption scheme $\mathcal{UAPE} = ((\mathcal{K}, \mathcal{E}, \mathcal{D}), \mathcal{UA}, \mathcal{DA})$ consists of a public-key encryption scheme $\mathcal{PE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ and two other algorithms.*

- *The key generation algorithm $\mathcal{K}$ is a randomized algorithm that takes as input a security parameter k and returns a pair (pk, sk) of keys, a public key and a matching secret key. For any pk, the message space $\mathcal{M}(pk)$ is uniquely determined.*
- *The encryption algorithm $\mathcal{E}$ is a randomized algorithm that takes the public key pk and a plaintext m and returns a standard ciphertext c.*
- *The decryption algorithm $\mathcal{D}$ for standard ciphertexts is a deterministic algorithm that takes the secret key sk and a standard ciphertext c and returns the corresponding plaintext m or a special symbol $\perp$ to indicate that the standard ciphertext is invalid.*
- *The anonymizing algorithm $\mathcal{UA}$ is a randomized algorithm that takes the public key pk and a standard ciphertext c and returns an anonymized ciphertext c'.*
- *The decryption algorithm $\mathcal{DA}$ for anonymized ciphertexts is a deterministic algorithm that takes the secret key sk and an anonymized ciphertext c' and returns the corresponding plaintext m or a special symbol $\perp$ to indicate that the anonymized ciphertext is invalid.*

We require the standard correctness condition. That is, for any (pk, sk) outputted by $\mathcal{K}$ and $m \in \mathcal{M}(pk)$, we have $m = \mathcal{D}_{sk}(\mathcal{E}_{pk}(m))$ and $m = \mathcal{DA}_{sk}(\mathcal{UA}_{pk}(\mathcal{E}_{pk}(m)))$.

In the universally anonymizable public-key encryption scheme, we can use $\mathcal{PE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ as a standard encryption scheme. Furthermore, in this scheme, by using the anonymizing algorithm $\mathcal{UA}$, anyone who has a standard ciphertext can anonymize it whenever she wants to do that. The receiver can decrypt the anonymized ciphertext by using the decryption algorithm $\mathcal{DA}$ for anonymized ciphertexts.

9.2.2 Security Properties

We now define security properties with respect to universally anonymizable public-key encryption schemes.

Data-Privacy

We define the security property called *data-privacy* of universally anonymizable public-key encryption schemes. The definition is based on the indistinguishability for standard public-key encryption schemes.

We can consider two types of data-privacy, that is, the data-privacy on standard ciphertexts and that on anonymized ciphertexts. We first describe the definition of the data-privacy on standard ciphertexts.

Definition 9.5 (data-privacy on standard ciphertexts)**.** *Let $b \in \{0,1\}$ and $k \in \mathbb{N}$. Let $A_{\mathrm{cpa}} = (A^1_{\mathrm{cpa}}, A^2_{\mathrm{cpa}})$, $A_{\mathrm{cca}} = (A^1_{\mathrm{cca}}, A^2_{\mathrm{cca}})$ be adversaries that run in two stages and where A_{cca} has access to the oracles $\mathcal{D}_{sk_0}(\cdot)$, $\mathcal{D}_{sk_1}(\cdot)$, $\mathcal{DA}_{sk_0}(\cdot)$, and $\mathcal{DA}_{sk_1}(\cdot)$. Note that si is the state information. It contains pk, m_0, m_1, and so on. For $\mathrm{atk} \in \{\mathrm{cpa}, \mathrm{cca}\}$, we consider the following experiment:*

$\mathtt{Experiment}\ \mathbf{Exp}^{\text{dataS-atk-}b}_{\mathcal{UAPE},A_{\mathrm{atk}}}(k)$
$(pk, sk) \leftarrow \mathcal{K}(k);\ (m_0, m_1, \mathsf{si}) \leftarrow A^1_{\mathrm{atk}}(pk);\ c \leftarrow \mathcal{E}_{pk}(m_b);\ d \leftarrow A^2_{\mathrm{atk}}(c, \mathsf{si})$
$\mathtt{return}\ d$

Note that $m_0, m_1 \in \mathcal{M}(pk)$. Above it is mandated that A^2_{cca} never queries the challenge c to either $\mathcal{D}_{sk_0}(\cdot)$ or $\mathcal{D}_{sk_1}(\cdot)$. It is also mandated that A^2_{cca} never queries either the anonymized ciphertext $\tilde{c} \in \{\mathcal{UA}_{pk_0}(c)\}$ to $\mathcal{DA}_{sk_0}(\cdot)$ or $\tilde{c} \in \{\mathcal{UA}_{pk_1}(c)\}$ to $\mathcal{DA}_{sk_1}(\cdot)$. For $\mathrm{atk} \in \{\mathrm{cpa}, \mathrm{cca}\}$, we define the advantage via

$$\mathbf{Adv}^{\text{dataS-atk}}_{\mathcal{UAPE},A_{\mathrm{atk}}}(k) = \Big|\Pr[\mathbf{Exp}^{\text{dataS-atk-1}}_{\mathcal{UAPE},A_{\mathrm{atk}}}(k) = 1] - \Pr[\mathbf{Exp}^{\text{dataS-atk-0}}_{\mathcal{UAPE},A_{\mathrm{atk}}}(k) = 1]\Big|.$$

We say that the universally anonymizable public-key encryption scheme $\mathcal{UAPE}$ provides the data-privacy on standard ciphertexts against the chosen plaintext attack (respectively the adaptive chosen ciphertext attack) if $\mathbf{Adv}^{\text{dataS-cpa}}_{\mathcal{UAPE},A_{\mathrm{cpa}}}(k)$ (resp. $\mathbf{Adv}^{\text{dataS-cca}}_{\mathcal{UAPE},A_{\mathrm{cca}}}(k)$) is negligible for any adversary A whose time complexity is polynomial in k.

In the above experiment, if the challenge is c, then anyone can compute $\mathcal{UA}_{pk_0}(c)$. Therefore, in the CCA setting, we restrict the oracle access to $\mathcal{DA}$ as described above.

We next describe the definition of the data-privacy on anonymized ciphertexts.

Definition 9.6 (data-privacy on anonymized ciphertexts)**.** *Let $b \in \{0,1\}$ and $k \in \mathbb{N}$. Let $A_{\mathrm{cpa}} = (A^1_{\mathrm{cpa}}, A^2_{\mathrm{cpa}})$, $A_{\mathrm{cca}} = (A^1_{\mathrm{cca}}, A^2_{\mathrm{cca}})$ be adversaries that run in two stages and where*

A_{cca} has access to the oracles $\mathcal{D}_{sk_0}(\cdot)$, $\mathcal{D}_{sk_1}(\cdot)$, $\mathcal{DA}_{sk_0}(\cdot)$, and $\mathcal{DA}_{sk_1}(\cdot)$. For atk $\in$ {cpa, cca}*, we consider the following experiment:*

$$\begin{array}{l}\texttt{Experiment } \mathbf{Exp}^{\text{dataA-atk-}b}_{\mathcal{UAPE},A_{\mathrm{atk}}}(k)\\ \quad (pk,sk)\leftarrow\mathcal{K}(k);\ (m_0,m_1,\mathsf{si})\leftarrow A^1_{\mathrm{atk}}(pk)\\ \quad c\leftarrow\mathcal{E}_{pk}(m_b);\ c'\leftarrow\mathcal{UA}_{pk}(c);\ d\leftarrow A^2_{\mathrm{atk}}(c',\mathsf{si})\\ \quad \texttt{return } d\end{array}$$

Note that $m_0, m_1 \in \mathcal{M}(pk)$. Above it is mandated that A^2_{cca} never queries the challenge c' to either $\mathcal{DA}_{sk_0}(\cdot)$ or $\mathcal{DA}_{sk_1}(\cdot)$. For atk $\in$ {cpa, cca}*, we define the advantage via*

$$\mathbf{Adv}^{\text{dataA-atk}}_{\mathcal{UAPE},A_{\mathrm{atk}}}(k)=\Big|\Pr[\mathbf{Exp}^{\text{dataA-atk-1}}_{\mathcal{UAPE},A_{\mathrm{atk}}}(k)=1]-\Pr[\mathbf{Exp}^{\text{dataA-atk-0}}_{\mathcal{UAPE},A_{\mathrm{atk}}}(k)=1]\Big|.$$

We say that the universally anonymizable public-key encryption scheme $\mathcal{UAPE}$ provides the data-privacy on anonymized ciphertexts against the chosen plaintext attack (resp. the adaptive chosen ciphertext attack) if $\mathbf{Adv}^{\text{dataA-cpa}}_{\mathcal{UAPE},A_{\mathrm{cpa}}}(k)$ (resp. $\mathbf{Adv}^{\text{dataA-cca}}_{\mathcal{UAPE},A_{\mathrm{cca}}}(k)$) is negligible for any adversary A whose time complexity is polynomial in k.

Remark 9.1. *In the CPA setting, if there exists an algorithm which breaks the data-privacy on anonymized ciphertexts, then we can break that on standard ciphertexts by applying the anonymizing algorithm to the standard ciphertexts and passing the resulting anonymized ciphertexts to the adversary which breaks the data-privacy on anonymized ciphertexts. Therefore, in the CPA setting, it is sufficient that the universally anonymizable public-key encryption scheme provides the data-privacy of standard ciphertexts.*

On the other hand, in the CCA setting, the data privacy on standard ciphertexts does not always imply that on anonymized ciphertexts, since the oracle access of the adversary attacking the data privacy on standard ciphertexts is restricted more strictly than that on anonymized ciphertexts.

Key-Privacy

We define the security property called *key-privacy* of universally anonymizable public-key encryption schemes. If the scheme provides the key-privacy, the adversary cannot know under which key the anonymized ciphertext was created.

Definition 9.7 (key-privacy)**.** *Let $b \in \{0,1\}$ and $k \in \mathbb{N}$. Let $A_{\mathrm{cpa}} = (A^1_{\mathrm{cpa}}, A^2_{\mathrm{cpa}})$, $A_{\mathrm{cca}} = (A^1_{\mathrm{cca}}, A^2_{\mathrm{cca}})$ be adversaries that run in two stages and where A_{cca} has access to the oracles $\mathcal{D}_{sk_0}(\cdot)$, $\mathcal{D}_{sk_1}(\cdot)$, $\mathcal{DA}_{sk_0}(\cdot)$, and $\mathcal{DA}_{sk_1}(\cdot)$. For* atk $\in$ {cpa, cca}*, we consider the following*

experiment:

$$\begin{array}{l}\texttt{Experiment } \mathbf{Exp}^{\text{key-atk-}b}_{\mathcal{UAPE},A_{\text{atk}}}(k)\\ \quad (pk_0, sk_0) \leftarrow \mathcal{K}(k);\ (pk_1, sk_1) \leftarrow \mathcal{K}(k)\\ \quad (m_0, m_1, \mathsf{si}) \leftarrow A^1_{\text{atk}}(pk_0, pk_1);\ c \leftarrow \mathcal{E}_{pk_b}(m_b);\ c' \leftarrow \mathcal{UA}_{pk_b}(c);\ d \leftarrow A^2_{\text{atk}}(c', \mathsf{si})\\ \quad \texttt{return } d\end{array}$$

Note that $m_0 \in \mathcal{M}(pk_0)$ and $m_1 \in \mathcal{M}(pk_1)$. Above it is mandated that A^2_{cca} never queries the challenge c' to either $\mathcal{DA}_{sk_0}(\cdot)$ or $\mathcal{DA}_{sk_1}(\cdot)$. For atk $\in$ {cpa, cca}*, we define the advantage via*

$$\mathbf{Adv}^{\text{key-atk}}_{\mathcal{UAPE},A_{\text{atk}}}(k) = \left|\Pr[\mathbf{Exp}^{\text{key-atk-1}}_{\mathcal{UAPE},A_{\text{atk}}}(k) = 1] - \Pr[\mathbf{Exp}^{\text{key-atk-0}}_{\mathcal{UAPE},A_{\text{atk}}}(k) = 1]\right|.$$

We say that the universally anonymizable public-key encryption scheme $\mathcal{UAPE}$ provides the key-privacy against the chosen plaintext attack (resp. the adaptive chosen ciphertext attack) if $\mathbf{Adv}^{\text{key-cpa}}_{\mathcal{UAPE},A_{\text{cpa}}}(k)$ (resp. $\mathbf{Adv}^{\text{key-cca}}_{\mathcal{UAPE},A_{\text{cca}}}(k)$) is negligible for any adversary A whose time complexity is polynomial in k.

Bellare, Boldyreva, Desai, and Pointcheval [3] proposed a security requirement of encryption schemes called "key-privacy." Similar to the above definition, it asks that the encryption provides privacy of the key under which the encryption was performed. In addition to the property of the universal anonymizability, there are two differences between their definition and ours.

In [3], they defined the encryption scheme with some *common-key* which contains the common parameter for all users to obtain the key-privacy property. For example, in the discrete-log based schemes such that the ElGamal and the Cramer-Shoup encryption schemes, the common key contains a common group G, and the encryption is performed over the common group for all uses.

On the other hand, in our definition, we do not prepare any common key for obtaining the key-privacy property. In the universally anonymizable public-key encryption scheme, we can use the standard encryption scheme which is not necessary to have the key-privacy property. In addition to it, anyone can anonymize the ciphertext by using its public key whenever she want to do that, and the adversary cannot know under which key the anonymized ciphertext was created.

The definition in [3], they considered the situation that the message space was common to each user. Therefore, in the experiment of their definition, the adversary chooses only one message m from the common message space and receives a ciphertext of m encrypted with one of two keys pk_0 and pk_1.

In our definition, we do not use common parameter and the message spaces for users may be different even if the security parameter is fixed. In fact, in Sections 9.3 and 9.4, we

propose the encryption schemes whose message spaces for users are different. Therefore, in the experiment of our definition, the adversary chooses two messages m_0 and m_1 where m_0 and m_1 are in the message spaces for pk_0 and pk_1, respectively, and receives either a ciphertext of m_0 encrypted with pk_0 or a ciphertext of m_1 encrypted with pk_1. The ability of the adversary with two messages m_0 and m_1 might be stronger than that with one message m.

We say that a universally anonymizable public-key encryption scheme $\mathcal{UAPE}$ is CPA-secure (resp. CCA-secure) if the scheme $\mathcal{UAPE}$ provides the data-privacy on standard ciphertexts, that on anonymized ciphertexts, and the key-privacy against the chosen plaintext attack (resp. the adaptive chosen ciphertext attack).

9.3 ElGamal and its Universal Anonymizability

In this section, we propose a universally anonymizable ElGamal encryption scheme.

9.3.1 The ElGamal Encryption Scheme

Definition 9.8 (ElGamal)**.** *The ElGamal encryption scheme* $\mathcal{PE}^{\mathsf{EG}} = (\mathcal{K}^{\mathsf{EG}}, \mathcal{E}^{\mathsf{EG}}, \mathcal{D}^{\mathsf{EG}})$ *is as follows. Note that* $\mathcal{Q}$ *is a QR-group generator with a safe prime which takes as input a security parameter* k *and returns* (q, g) *where* q *is* k*-bit prime,* $p = 2q + 1$ *is prime, and* g *is a generator of a cyclic group* QR_p *(a group of quadratic residues modulo* p*) of order* q*.*

$\mathtt{Algorithm}\ \mathcal{K}^{\mathsf{EG}}(k)$	$\mathtt{Algorithm}\ \mathcal{E}^{\mathsf{EG}}_{pk}(m)$	$\mathtt{Algorithm}\ \mathcal{D}^{\mathsf{EG}}_{sk}(c_1, c_2)$
$(q, g) \leftarrow \mathcal{Q}(k)$	$r \xleftarrow{R} \mathbb{Z}_q$	$m \leftarrow c_2 \cdot c_1^{-x}$
$x \xleftarrow{R} \mathbb{Z}_q;\ y \leftarrow g^x$	$c_1 \leftarrow g^r$	$\mathtt{return}\ m$
$\mathtt{return}\ pk = (q, g, y)\ \mathtt{and}\ sk = x$	$c_2 \leftarrow m \cdot y^r$	
	$\mathtt{return}\ (c_1, c_2)$	

The ElGamal encryption scheme is secure in the sense of IND-CPA if the DDH problem for $\mathcal{Q}$ is hard.

9.3.2 Universal Anonymizability of the ElGamal Encryption Scheme

We now consider the situation that there exists no common key, and in the above definition of the ElGamal encryption scheme, each user chooses an arbitrary prime q where $|q| = k$ and $p = 2q + 1$ is also prime, and uses a group of quadratic residues modulo p. Therefore, each user U_i uses a different groups G_i for her encryption scheme and if she publishes the ciphertext directly (without anonymization) then the scheme does not provide the key-privacy. In fact, the adversary simply checks whether the ciphertext y is in the group G_i, and if $y \notin G_i$ then y was not encrypted by U_i. To anonymize the standard ciphertext

of the ElGamal encryption scheme, we consider the following strategy in the anonymizing algorithm.

1) Compute a ciphertext c over each user's prime-order group.

2) Encode c to an element $\bar{c} \in \mathbb{Z}_q$ (the encoding function).

3) Expand $\bar{c}$ to the common domain (the expanding technique).

The Encoding Function

Generally speaking, it is not easy to encode the elements of a prime-order group of order q to those of $\mathbb{Z}_q$. We employ the idea described in [26] by Cramer and Shoup. We can encode the elements of QR_p where $p = 2q + 1$ and p, q are prime to those of $\mathbb{Z}_q$.

Let p be safe prime (i.e. $q = (p-1)/2$ is also prime) and $QR_p \subset \mathbb{Z}_p^*$ a group of quadratic residues modulo p. Then we have $|QR_p| = q$ and $QR_p = \{1^2 \bmod p,\ 2^2 \bmod p, \cdots,\ q^2 \bmod p\}$. It is easy to see that QR_p is a cyclic group of order q, and each $g \in QR_p \backslash \{1\}$ is a generator of QR_p.

We now define a function $F_q : QR_p \to \mathbb{Z}_q$ as

$$F_q(x) = \min \left\{ \pm x^{\frac{p-1}{4}} \bmod p \right\}.$$

Noticing that $\pm\, x^{\frac{p-1}{4}} \bmod p$ are the square roots of x modulo p, the function F_q is bijective and we have $F_q^{-1}(y) = y^2 \bmod p$. We call the function F_q an *encoding function*. We also define a *t-encoding function* $\bar{F}_{q,t} : (QR_p)^t \to (\mathbb{Z}_q)^t$. $\bar{F}_{q,t}$ takes as input $(x_1, \cdots, x_t) \in (QR_p)^t$ and returns $(y_1, \cdots, y_t) \in (\mathbb{Z}_q)^t$ where $y_i = F_q(x_i)$ for each $i \in \{1, \cdots, t\}$. It is easy to see that $\bar{F}_{q,t}$ is bijective and we can define $\bar{F}_{q,t}^{-1}$.

Our Scheme

We now propose our universally anonymizable ElGamal encryption scheme. Our scheme provides the key-privacy against the chosen plaintext attack even if each user chooses an arbitrary prime q where $|q| = k$ and $p = 2q+1$ is also prime, and uses a group of quadratic residues modulo p.

Definition 9.9. *Our universally anonymizable ElGamal encryption scheme* $\mathcal{UAPE}^{\mathsf{EG}} = ((\mathcal{K}^{\mathsf{EG}}, \mathcal{E}^{\mathsf{EG}}, \mathcal{D}^{\mathsf{EG}}), \mathcal{UA}^{\mathsf{EG}}, \mathcal{DA}^{\mathsf{EG}})$ *consists of the ElGamal encryption scheme* $\mathcal{PE}^{\mathsf{EG}} = (\mathcal{K}^{\mathsf{EG}},$

$\mathcal{E}^{\mathsf{EG}}, \mathcal{D}^{\mathsf{EG}}$) *and two algorithms described as follows.*

$\texttt{Algorithm}\ \mathcal{UA}^{\mathsf{EG}}_{pk}(c_1, c_2)$	$\texttt{Algorithm}\ \mathcal{DA}^{\mathsf{EG}}_{sk}(c'_1, c'_2)$
$(\bar{c}_1, \bar{c}_2) \leftarrow \bar{F}_{q,2}(c_1, c_2)$	$\bar{c}_1 \leftarrow c'_1 \bmod q;\ \bar{c}_2 \leftarrow c'_2 \bmod q$
$t_1 \xleftarrow{R} \{0, 1, 2, \cdots, \lfloor (2^{k+160} - \bar{c}_1)/q \rfloor\}$	$(c_1, c_2) \leftarrow \bar{F}^{-1}_{q,2}(\bar{c}_1, \bar{c}_2)$
$t_2 \xleftarrow{R} \{0, 1, 2, \cdots, \lfloor (2^{k+160} - \bar{c}_2)/q \rfloor\}$	$m \leftarrow \mathcal{D}^{\mathsf{EG}}_{sk}(c_1, c_2)$
$c'_1 \leftarrow \bar{c}_1 + t_1 q;\ c'_2 \leftarrow \bar{c}_2 + t_2 q$	$\texttt{return}\ m$
$\texttt{return}\ (c'_1, c'_2)$	

9.3.3 Security

In this section, we prove that our universally anonymizable ElGamal encryption scheme $\mathcal{UAPE}^{\mathsf{EG}}$ is CPA-secure assuming that the DDH problem for $\mathcal{Q}$ is hard.

We can easily see that our scheme provides the data-privacy on standard ciphertexts against the chosen plaintext attack if the DDH problem for $\mathcal{Q}$ is hard. More precisely, we can prove that if there exists a CPA-adversary attacking the data-privacy on standard ciphertexts of our scheme with advantage ϵ, then there exists a CPA-adversary attacking the indistinguishability of the ElGamal encryption scheme with the same advantage ϵ.

Note that this implies our scheme provides the data-privacy on anonymized ciphertexts against the chosen plaintext attack if the DDH problem for $\mathcal{Q}$ is hard.

We now prove our scheme provides the key-privacy against the chosen plaintext attack. To prove this, we use the idea of Halevi [49].

Lemma 9.1 (Halevi [49])**.** *Let $\mathcal{PE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a (standard) encryption scheme that is CCA secure (resp. CPA secure) for the indistinguishability (data-privacy). Then a sufficient condition for $\mathcal{PE}$ to be also CCA secure (resp. CPA secure) for the key-privacy (defined by Bellare, Boldyreva, Desai, and Pointcheval) if the statistical distance between the two distributions*

$$D_0 = \{(pk_0, pk_1, \mathcal{E}_{pk_0}(m)) : (pk_0, sk_0), (pk_1, sk_1) \leftarrow \mathcal{K}(k);\ m \xleftarrow{R} \mathcal{M}(pk_0)\}$$
$$D_1 = \{(pk_0, pk_1, \mathcal{E}_{pk_1}(m)) : (pk_0, sk_0), (pk_1, sk_1) \leftarrow \mathcal{K}(k);\ m \xleftarrow{R} \mathcal{M}(pk_1)\}$$

is negligible.

This lemma shows the relation between the indistinguishability and the key-privacy for *standard* encryption scheme. We can apply this lemma to our universally anonymizable encryption scheme. That is, if the universally anonymizable encryption scheme $\mathcal{UAPE} = ((\mathcal{K}, \mathcal{E}, \mathcal{D}), \mathcal{UA}, \mathcal{DA})$ provides the data-privacy on *anonymized* ciphertexts against CCA (resp. CPA) and the statistical distance between the two distributions

$$D'_0 = \{(pk_0, pk_1, \mathcal{UA}_{pk_0}(\mathcal{E}_{pk_0}(m))) : (pk_0, sk_0), (pk_1, sk_1) \leftarrow \mathcal{K}(k);\ m \xleftarrow{R} \mathcal{M}(pk_0)\}$$
$$D'_1 = \{(pk_0, pk_1, \mathcal{UA}_{pk_1}(\mathcal{E}_{pk_1}(m))) : (pk_0, sk_0), (pk_1, sk_1) \leftarrow \mathcal{K}(k);\ m \xleftarrow{R} \mathcal{M}(pk_1)\}$$

is negligible, then $\mathcal{UAPE}$ provides the key-privacy against CCA (resp. CPA).

By using this, in order to prove that our scheme provides the key-privacy against the chosen plaintext attack, all we have to do is to see that the two distributions D'_0 and D'_1 derived by our scheme satisfy the property defined above. It is easy to see that the statistical distance between D'_0 and D'_1 is less than $2 \times (1/2^{159})^2$.

In conclusion, our universally anonymizable ElGamal encryption scheme is CPA-secure assuming that the DDH problem for $\mathcal{Q}$ is hard.

9.4 Cramer-Shoup and its Universal Anonymizability

In this section, we propose a universally anonymizable Cramer-Shoup encryption scheme.

9.4.1 The Cramer-Shoup Encryption Scheme

Definition 9.10 (Cramer-Shoup)**.** *The Cramer-Shoup encryption scheme* $\mathcal{PE}^{\mathsf{CS}} = (\mathcal{K}^{\mathsf{CS}}, \mathcal{E}^{\mathsf{CS}}, \mathcal{D}^{\mathsf{CS}})$ *is defined as follows. Let* $\mathcal{H} = (\mathcal{GH}, \mathcal{EH})$ *be a family of hash functions. Note that* $\mathcal{Q}$ *is a QR-group generator with a safe prime.*

$\mathtt{Algorithm}\ \mathcal{K}^{\mathsf{CS}}(k)$	$\mathtt{Algorithm}\ \mathcal{E}^{\mathsf{CS}}_{pk}(m)$	$\mathtt{Algorithm}\ \mathcal{D}^{\mathsf{CS}}_{sk}(u_1, u_2, e, v)$
$(q, g) \leftarrow \mathcal{Q}(k);\ K \leftarrow \mathcal{GH}(k)$	$r \xleftarrow{R} \mathbb{Z}_q$	$\alpha \leftarrow \mathcal{EH}_K(u_1, u_2, e)$
$g_1 \leftarrow g;\ g_2 \xleftarrow{R} QR_p$	$u_1 \leftarrow g_1^r;\ u_2 \leftarrow g_2^r$	$\mathtt{if}\ (u_1^{x_1+y_1\alpha} u_2^{x_2+y_2\alpha} = v)$
$x_1, x_2, y_1, y_2, z \xleftarrow{R} \mathbb{Z}_q$	$e \leftarrow h^r m$	$\quad \mathtt{then}\ m \leftarrow e/u_1^z$
$c \leftarrow g_1^{x_1} g_2^{x_2};\ d \leftarrow g_1^{y_1} g_2^{y_2}$	$\alpha \leftarrow \mathcal{EH}_K(u_1, u_2, e)$	$\mathtt{else}\ m \leftarrow \perp$
$h \leftarrow g_1^z$	$v \leftarrow c^r d^{r\alpha}$	$\mathtt{return}\ m$
$pk \leftarrow (q, g_1, g_2, c, d, h, K)$	$\mathtt{return}\ (u_1, u_2, e, v)$	
$sk \leftarrow (x_1, x_2, y_1, y_2, z)$		
$\mathtt{return}\ (pk, sk)$		

Cramer and Shoup [26] proved that the Cramer-Shoup encryption scheme is secure in the sense of IND-CCA2 assuming that $\mathcal{H}$ is universal one-way and the DDH problem for $\mathcal{Q}$ is hard. Lucks [61] recently proposed a variant of the Cramer-Shoup encryption scheme for groups of unknown order. This scheme is secure in the sense of IND-CCA2 assuming that the family of hash functions in the scheme is universal one-way, and both the Decisional Diffie-Hellman problem in QR_N (a set of quadratic residues modulo N) and factoring N are hard.

9.4.2 Universal Anonymizability of the Cramer-Shoup Encryption Scheme

We propose our universally anonymizable Cramer-Shoup encryption scheme. Our scheme provides the key-privacy against the adaptive chosen ciphertext attack even if each user chooses an arbitrary prime q where $|q| = k$ and $p = 2q + 1$ is also prime, and uses a group of quadratic residues modulo p.

Note that in our scheme we employ the encoding function and the expanding technique appeared in Section 9.3.

Definition 9.11. *Our universally anonymizable Cramer-Shoup encryption scheme* $\mathcal{UAPE}^{\mathsf{CS}} = ((\mathcal{K}^{\mathsf{CS}}, \mathcal{E}^{\mathsf{CS}}, \mathcal{D}^{\mathsf{CS}}), \mathcal{UA}^{\mathsf{CS}}, \mathcal{DA}^{\mathsf{CS}})$ *consists of the Cramer-Shoup encryption scheme* $\mathcal{PE}^{\mathsf{CS}} = (\mathcal{K}^{\mathsf{CS}}, \mathcal{E}^{\mathsf{CS}}, \mathcal{D}^{\mathsf{CS}})$ *and two algorithms described as follows.*

$\mathtt{Algorithm}\ \mathcal{UA}^{\mathsf{CS}}_{pk}(u_1, u_2, e, v)$
$(\bar{u}_1, \bar{u}_2, \bar{e}, \bar{v}) \leftarrow \bar{F}_{q,4}(u_1, u_2, e, v)$
$t_1 \overset{R}{\leftarrow} \{0, 1, 2, \cdots, \lfloor (2^{k+160} - \bar{u}_1)/q \rfloor\}$
$t_2 \overset{R}{\leftarrow} \{0, 1, 2, \cdots, \lfloor (2^{k+160} - \bar{u}_2)/q \rfloor\}$
$t_3 \overset{R}{\leftarrow} \{0, 1, 2, \cdots, \lfloor (2^{k+160} - \bar{e})/q \rfloor\}$
$t_4 \overset{R}{\leftarrow} \{0, 1, 2, \cdots, \lfloor (2^{k+160} - \bar{v})/q \rfloor\}$
$u'_1 \leftarrow \bar{u}_1 + t_1 q;\ u'_2 \leftarrow \bar{u}_2 + t_2 q$
$e' \leftarrow \bar{e} + t_3 q;\ v' \leftarrow \bar{v} + t_4 q$
$\mathtt{return}\ (u'_1, u'_2, e', v')$

$\mathtt{Algorithm}\ \mathcal{DA}^{\mathsf{CS}}_{sk}(u'_1, u'_2, e', v')$
$\bar{u}_1 \leftarrow u'_1 \bmod q;\ \bar{u}_2 \leftarrow u'_2 \bmod q$
$\bar{e} \leftarrow e' \bmod q;\ \bar{v} \leftarrow v' \bmod q$
$(u_1, u_2, e, v) \leftarrow \bar{F}^{-1}_{q,4}(\bar{u}_1, \bar{u}_2, \bar{e}, \bar{v})$
$m \leftarrow \mathcal{D}^{\mathsf{CS}}_{sk}(u_1, u_2, e, v)$
$\mathtt{return}\ m$

9.4.3 Security

In this section, we prove that our universally anonymizable Cramer-Shoup encryption scheme $\mathcal{UAPE}^{\mathsf{EG}}$ is CCA-secure assuming that the DDH problem for $\mathcal{Q}$ is hard and $\mathcal{H}$ is universal one-way.

We can prove that our scheme provides the data-privacy on standard ciphertexts against the adaptive chosen ciphertext attack if the DDH problem for $\mathcal{Q}$ is hard and $\mathcal{H}$ is universal one-way. More precisely, we can prove that if there exists a CCA-adversary A attacking the data-privacy on standard ciphertexts of our scheme with advantage ϵ, then there exists a CCA2-adversary B attacking the indistinguishability of the Cramer-Shoup encryption scheme with the same advantage ϵ. In the reduction of the proof, we have to simulate the decryption oracles for anonymized ciphertexts for A. If A makes a query $\vec{c}' = (u'_1, u'_2, e', v')$ to $\mathcal{DA}_{sk_0}(\cdot)$, we simply compute $\vec{c} = (u'_1 \bmod q_0, u'_2 \bmod q_0, e' \bmod q_0, v' \bmod q_0)$ and decrypt $\vec{c}$ by using the decryption algorithm $\mathcal{D}_{sk_0}(\cdot)$ for standard ciphertexts for B. We can simulate $\mathcal{DA}_{sk_1}(\cdot)$ in a similar way.

In order to prove that our scheme provides the key-privacy and the data-privacy on anonymized ciphertexts against the adaptive chosen ciphertext attack, we need restriction

as follows.

We define the set of ciphertexts $EC_{\mathsf{CS}}((u'_1, u'_2, e', v'), pk)$ called "equivalence class" as

$$EC_{\mathsf{CS}}((u'_1, u'_2, e', v'), pk) = \{(\check{u}_1, \check{u}_2, \check{e}, \check{v}) \in (\{0,1\}^{k+160})^4 | \\ \check{u}_1 = u'_1 \pmod q \wedge \check{u}_2 = u'_2 \pmod q \wedge \check{e} = e' \pmod q \wedge \check{v} = v' \pmod q\}.$$

If $\vec{c}' = (u'_1, u'_2, e', v') \in (\{0,1\}^{k+160})^4$ is an anonymized ciphertext of m under $pk = (q, g_1, g_2, c, d, h, K)$ then any element $\check{\vec{c}} = (\check{u}_1, \check{u}_2, \check{e}, \check{v}) \in EC_{\mathsf{CS}}(\vec{c}', pk)$ is also an anonymized ciphertext of m under pk. Therefore, when $\vec{c}'$ is a challenge anonymized ciphertext, the adversary can ask an anonymized ciphertext $\check{\vec{c}} \in EC_{\mathsf{CS}}(\vec{c}', pk_0)$ to the decryption oracle $\mathcal{DA}^{\mathsf{CS}}_{sk_0}$ for anonymized ciphertexts, and if the answer of $\mathcal{DA}^{\mathsf{CS}}_{sk_0}$ is m_0 then the adversary knows that $\vec{c}'$ is encrypted by pk_0 and the plaintext of $\vec{c}'$ is m_0.

Furthermore, the adversary can ask $(u'_1 \bmod q_0, u'_2 \bmod q_0, e' \bmod q_0, v' \bmod q_0)$ to the decryption oracle $\mathcal{D}^{\mathsf{CS}}_{sk_0}$ for standard ciphertexts. If the answer of $\mathcal{D}^{\mathsf{CS}}_{sk_0}$ is m_0, then the adversary knows that $\vec{c}'$ is encrypted by pk_0 and the plaintext of $\vec{c}'$ is m_0.

To prevent these attacks, we add some natural restriction to the adversaries in the definitions of the key-privacy and the data-privacy on anonymized ciphertexts. That is, it is mandated that the adversary never queries either $\check{\vec{c}} \in EC_{\mathsf{CS}}(\vec{c}', pk_0)$ to $\mathcal{DA}^{\mathsf{CS}}_{sk_0}$ or $\check{\vec{c}} \in EC_{\mathsf{CS}}(\vec{c}', pk_1)$ to $\mathcal{DA}^{\mathsf{CS}}_{sk_1}$. It is also mandated that the adversary never queries either $(u'_1 \bmod q_0, u'_2 \bmod q_0, e' \bmod q_0, v' \bmod q_0)$ to $\mathcal{D}^{\mathsf{CS}}_{sk_0}$ or $(u'_1 \bmod q_1, u'_2 \bmod q_1, e' \bmod q_1, v' \bmod q_1)$ to $\mathcal{D}^{\mathsf{CS}}_{sk_1}$.

We think these restrictions are natural and reasonable. Actually, in the case of undeniable and confirmer signature schemes, Galbraith and Mao [44] defined the anonymity on undeniable signature schemes with the above restriction. Incidentally, Canetti, Krawczyk, and Nielsen [18] proposed a relaxed notion of CCA security, called Replayable CCA (RCCA). In their security model, the schemes which require restriction such as equivalence class for proving their CCA security satisfy a variant of RCCA, pd-RCCA (publicly-detectable replayable-CCA) secure.

If we add these restrictions then we can prove that our scheme provides the data-privacy on anonymized ciphertexts against the adaptive chosen ciphertext attack if the DDH problem for $\mathcal{Q}$ is hard and $\mathcal{H}$ is universal one-way. More precisely, we can prove that if there exists a CCA-adversary attacking the data-privacy on anonymized ciphertexts of our scheme with advantage ϵ, then there exists a CCA-adversary attacking the data-privacy on standard ciphertexts of our scheme with the same advantage ϵ.

We now prove our scheme provides the key-privacy against the adaptive chosen ciphertext attack. If we add the restrictions described above, we can prove this in a similar way as that for our universally anonymizable ElGamal encryption scheme. Note that the statistical distance between D'_0 and D'_1 (See Section 9.3.3.) is less than $2 \times (1/2^{159})^4$.

In conclusion, our universally anonymizable Cramer-Shoup encryption scheme is CCA-secure assuming that the DDH problem for $\mathcal{Q}$ is hard and $\mathcal{H}$ is universal one-way.

9.5 RSA-OAEP and its Universal Anonymizability

In this section, we propose a universally anonymizable RSA-OAEP scheme.

9.5.1 RSA-OAEP

Definition 9.12 (RSA-OAEP)**.** *RSA-OAEP* $\mathcal{PE}^{\mathsf{RO}} = (\mathcal{K}^{\mathsf{RO}}, \mathcal{E}^{\mathsf{RO}}, \mathcal{D}^{\mathsf{RO}})$ *is as follows. Let* k*,* k_0 *and* k_1 *be security parameters such that* $k_0 + k_1 < k$*. This defines an associated plaintext-length* $n = k - k_0 - k_1$*. The key generation algorithm* $\mathcal{K}^{\mathsf{RO}}$ *takes as input a security parameter* k *and runs the key generation algorithm of* RSA *to get* N, e, d*. It outputs the public key* $pk = (N, e)$ *and the secret key* $sk = d$*. The other algorithms are depicted below. Let* $G : \{0,1\}^{k_0} \to \{0,1\}^{n+k_1}$ *and* $H : \{0,1\}^{n+k_1} \to \{0,1\}^{k_0}$ *be hash functions. Note that* $[x]^{\ell}$ *denotes the* ℓ *most significant bits of* x*, and* $[x]_{\ell'}$ *denotes the* ℓ' *least significant bits of* x*.*

$\mathtt{Algorithm}\ \mathcal{E}^{\mathsf{RO}}_{pk}(m)$	$\mathtt{Algorithm}\ \mathcal{D}^{\mathsf{RO}}_{sk}(c)$
$r \xleftarrow{R} \{0,1\}^{k_0}$	$s \leftarrow [c^d \bmod N]^{n+k_1};\ t \leftarrow [c^d \bmod N]_{k_0}$
$s \leftarrow (m \Vert 0^{k_1}) \oplus G(r)$	$r \leftarrow t \oplus H(s)$
$t \leftarrow r \oplus H(s)$	$m \leftarrow [s \oplus G(r)]^n;\ p \leftarrow [s \oplus G(r)]_{k_1}$
$c \leftarrow (s \Vert t)^e \bmod N$	$\mathtt{if}\ (p = 0^{k_1})\ z \leftarrow m\ \mathtt{else}\ z \leftarrow \perp$
$\mathtt{return}\ c$	$\mathtt{return}\ z$

Fujisaki, Okamoto, Pointcheval, and Stern [43] proved that OAEP with partial one-way permutations is secure in the sense of IND-CCA2 in the random oracle model. They also showed that RSA is one-way if and only if RSA is θ-partial one-way for $\theta > 0.5$. Thus, RSA-OAEP is secure in the sense of IND-CCA2 in the random oracle model assuming RSA is one-way.

9.5.2 Universal Anonymizability of RSA-OAEP

A simple observation that seems to be folklore is that if one publishes the ciphertext of the RSA-OAEP scheme directly (without anonymization) then the scheme does not provide the key-privacy. Suppose an adversary knows that the ciphertext c is created under one of two keys (N_0, e_0) or (N_1, e_1), and suppose $N_0 \leq N_1$. If $c \geq N_0$ then the adversary bets it was created under (N_1, e_1), else the adversary bets it was created under (N_0, e_0). It is not hard to see that this attack has non-negligible advantage.

To anonymize ciphertexts of RSA-OAEP, we do not have to employ the encoding function and we only use the expanding technique.

Definition 9.13. *Our universally anonymizable RSA-OAEP scheme* $\mathcal{UAPE}^{\mathsf{RO}} = ((\mathcal{K}^{\mathsf{RO}}, \mathcal{E}^{\mathsf{RO}}, \mathcal{D}^{\mathsf{RO}}), \mathcal{UA}^{\mathsf{RO}}, \mathcal{DA}^{\mathsf{RO}})$ *consists of RSA-OAEP* $\mathcal{PE}^{\mathsf{RO}} = (\mathcal{K}^{\mathsf{RO}}, \mathcal{E}^{\mathsf{RO}}, \mathcal{D}^{\mathsf{RO}})$ *and two algorithms described as follows.*

$\mathtt{Algorithm}\ \mathcal{UA}^{\mathsf{RO}}_{pk}(c)$	$\mathtt{Algorithm}\ \mathcal{DA}^{\mathsf{RO}}_{sk}(c')$
$\alpha \xleftarrow{R} \{0, 1, 2, \cdots, \lfloor (2^{k+160} - c)/N \rfloor\}$	$c \leftarrow c' \bmod N$
$c' \leftarrow c + \alpha N$	$z \leftarrow \mathcal{D}^{\mathsf{RO}}_{sk}(c)$
$\mathtt{return}\ c'$	$\mathtt{return}\ z$

9.5.3 Security

In this section, we prove that our universally anonymizable RSA-OAEP scheme $\mathcal{UAPE}^{\mathsf{RO}}$ is CCA-secure in the random oracle model assuming RSA is one-way.

We can prove that our scheme provides the data-privacy on standard ciphertexts against the adaptive chosen ciphertext attack in the random oracle model assuming RSA is θ-partial one-way for $\theta > 0.5$. More precisely, if RSA-OAEP is secure in the sense of IND-CCA2 then our scheme provides the data-privacy on standard ciphertexts against the adaptive chosen ciphertext attack. The proof is similar to that for our universally anonymizable Cramer-Shoup encryption scheme.

In order to prove that our scheme provides the key-privacy and the data-privacy on anonymized ciphertexts against the adaptive chosen ciphertext attack, we need the restrictions similar to those for our universally anonymizable Cramer-Shoup encryption scheme. We define the equivalence class for our universally anonymizable RSA-OAEP scheme as

$$EC_{\mathsf{RO}}(c', pk) = \{\check{c} \in \{0,1\}^{k+160} | \check{c} = c' \pmod N\}$$

where $pk = (N, e)$ and it is mandated that the adversary never queries either $\check{c} \in EC_{\mathsf{RO}}(c', pk_0)$ to $\mathcal{DA}^{\mathsf{RO}}_{sk_0}$ or $\check{c} \in EC_{\mathsf{RO}}(c', pk_1)$ to $\mathcal{DA}^{\mathsf{RO}}_{sk_1}$. It is also mandated that the adversary never queries either $c' \bmod N_0$ to $\mathcal{D}^{\mathsf{RO}}_{sk_0}$ or $c' \bmod N_1$ to $\mathcal{D}^{\mathsf{RO}}_{sk_1}$.

If we add these restrictions then we can prove that our scheme provides the data-privacy on anonymized ciphertexts against the adaptive chosen ciphertext attack in the random oracle model assuming RSA is θ-partial one-way for $\theta > 0.5$ in a similar way as that for our universally anonymizable Cramer-Shoup encryption scheme.

Furthermore, if we add the restrictions described above, then we can prove that our scheme provides the key-privacy against the adaptive chosen ciphertext attack in the random oracle model assuming RSA is θ-partial one-way for $\theta > 0.5$. More precisely, we show the following theorem [1].

[1] Halevi [49] noted that we cannot apply Lemma 9.1 directly to the schemes analyzed in the random oracle model.

Theorem 9.1. *For any adversary A attacking the key-privacy of our scheme under the adaptive chosen ciphertext attack, and making at most q_{dec} queries to decryption oracle for standard ciphertexts, q'_{dec} queries to decryption oracle for anonymized ciphertexts, q_{gen} G-oracle queries, and q_{hash} H-oracle queries, there exists a θ-partial inverting adversary M for* RSA*, such that for any k, k_0, k_1, and $\theta = \frac{k-k_0}{k}$,*

$$\mathbf{Adv}^{\text{key-cca}}_{\mathcal{UAPE}^{\mathsf{RO}},A}(k) \leq 8q_{\mathrm{hash}} \cdot ((1-\epsilon_1)\cdot(1-\epsilon_2))^{-1} \cdot \mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{RSA},M}(k) + q_{\mathrm{gen}} \cdot (1-\epsilon_2)^{-1} \cdot 2^{-k+2}$$

where $\epsilon_1 = \frac{2}{2^{k/2-3}-1} + \frac{1}{2^{159}}$, $\epsilon_2 = \frac{2q_{\mathrm{gen}}+q_{\mathrm{dec}}+q'_{\mathrm{dec}}+2q_{\mathrm{gen}}(q_{\mathrm{dec}}+q'_{\mathrm{dec}})}{2^{k_0}} + \frac{2(q_{\mathrm{dec}}+q'_{\mathrm{dec}})}{2^{k_1}} + \frac{2q_{\mathrm{hash}}}{2^{k-k_0}}$, *and the running time of B is that of A plus $q_{\mathrm{gen}} \cdot q_{\mathrm{hash}} \cdot O(k^3)$.*

In conclusion, since RSA is θ-partial one-way if and only if RSA is one-way for $\theta > 0.5$, our universally anonymizable RSA-OAEP scheme is CCA-secure in the random oracle model assuming RSA is one-way.

Proof of Theorem 9.1. The proof is similar to that for OAEP with expanding in Section 3.3. We describe the partial inverting algorithm M for RSA using a CCA-adversary A attacking the anonymity of our encryption scheme. M is given $pk = (N, e, k)$ and a point $y \in \mathbb{Z}_N^*$ where $|y| = k = n + k_0 + k_1$. Let $sk = (N, d, k)$ be the corresponding secret key. The algorithm is trying to find the $n + k_1$ most significant bits of the e-th root of y modulo N. *Intuition.* We assume that the challenge ciphertext for A is $Y \in \{0,1\}^{k+160}$ which was encrypted by $pk = (N, e)$, and $y = Y \bmod N$. In order to distinguish under which key the given ciphertext Y was created, the adversary A has to make queries r and s to oracles G and H, respectively, such that $s = (m||0^{k_1}) \oplus G(r)$ and $y = (s||(r \oplus H(s)))^e \bmod N$. Therefore, A asks s to H with non-negligible probability where s is the $n + k_1$ most significant bits of the e-th root of y modulo N.

1) M picks $\mu \xleftarrow{R} \{0, 1, 2, \ldots, \lfloor (2^{k+160} - y)/N \rfloor\}$ and sets $Y \leftarrow y + \mu N$.

2) M runs the key generation algorithm of RSA with security parameter k to obtain $pk' = (N', e', k)$ and $sk' = (N', d', k)$. Then it picks a bit $b \xleftarrow{R} \{0, 1\}$, and sets $pk_b \leftarrow (N, e)$ and $pk_{1-b} \leftarrow (N', e')$. If the above y does not satisfy $y \in (\mathbb{Z}_{N_0}^* \cap \mathbb{Z}_{N_1}^*)$ then M outputs Fail and halts; else it continues.

3) M initializes four lists, called G-list, H-list, Y_0-list, and Y_1-list to empty. It then runs A as follows. Note that M simulates A's oracles G, H, $\mathcal{D}_{sk_0}$, and $\mathcal{D}_{sk_1}$ as described below.

   3-1) M runs $A_1(pk_0, pk_1)$ and gets (m_0, m_1, si) which is the output of A_1.

   3-2) M runs $A_2(Y, \mathsf{si})$ and gets a bit $d \in \{0, 1\}$ which is the output of A_2.

4) M chooses a random pair (h, H_h) from the H-list and outputs h as its guess for the $n + k_1$ most significant bits of the e-th root of y modulo N.

M simulates A's random oracles G and H, the decryption oracles $\mathcal{D}_{sk_0}$ and $\mathcal{D}_{sk_1}$ for standard ciphertexts, and the decryption oracles $\mathcal{DA}_{sk_0}$ and $\mathcal{DA}_{sk_1}$ for anonymized ciphertexts as follows:

- When A makes an oracle query g to G, then for each (h, H_h) on the H-list, M builds $z = h||(g \oplus H_h)$, and computes $y_{h,g,0} = z^{e_0} \bmod N_0$ and $y_{h,g,1} = z^{e_1} \bmod N_1$. For $i \in \{0, 1\}$, M checks whether $y = y_{h,g,i}$. If for some h and i such a relation holds, then we have inverted y under pk_i, and we can still correctly simulate G by answering $G_g = h \oplus (m_i||0^{k_1})$. Otherwise, M outputs a random value G_g of length $n + k_1$. In both cases, M adds (g, G_g) to the G-list. Then, for all h, M checks if the k_1 least significant bits of $h \oplus G_g$ are all 0. If they are, then it adds $y_{h,g,0}$ and $y_{h,g,1}$ to the Y_0-list and the Y_1-list, respectively.

- When A makes an oracle query h to H, M provides A with a random string H_h of length k_0 and adds (h, H_h) to the H-list. Then for each (g, G_g) on the G-list, M builds $z = h||(g \oplus H_h)$, and computes $y_{h,g,0} = z^{e_0} \bmod N_0$ and $y_{h,g,1} = z^{e_1} \bmod N_1$. M checks if the k_1 least significant bits of $h \oplus G_g$ are all 0. If they are, then it adds $y_{h,g,0}$ and $y_{h,g,1}$ to the Y_0-list and the Y_1-list, respectively.

- When for $i \in \{0, 1\}$, A makes an oracle query $\hat{y} \in \mathbb{Z}^*_{N_i}$ to $\mathcal{D}_{sk_i}$, M checks if there exists some $y_{h,g,i}$ in the Y_i-list such that $\hat{y} = y_{h,g,i}$. If there is, then it returns the n most significant bits of $h \oplus G_g$ to A. Otherwise it returns $\perp$ (indicating that $\hat{y}$ is an invalid ciphertext).

- When for $i \in \{0, 1\}$, A makes an oracle query $\hat{Y} \in \{0, 1\}^{k+160}$ to $\mathcal{DA}_{sk_i}$, M checks if there exists some $y_{h,g,i}$ in the Y_i-list such that $\hat{Y} \bmod N_i = y_{h,g,i}$. If there is, then it returns the n most significant bits of $h \oplus G_g$ to A. Otherwise it returns $\perp$ (indicating that $\hat{Y}$ is an invalid anonymized ciphertext).

In order to analyze the advantage of M, we define some events. For $i \in \{0, 1\}$, let $w_i = y^{d_i} \bmod N_i$, $s_i = [w_i]^{n+k_1}$, and $t_i = [w_i]_{k_0}$. That is, w_i is the e_i-th root of y modulo N_i and s_i is the $n + k_1$ most significant bits of the e_i-th root of y modulo N_i. Note that M wins the game if it outputs s_b. Let r_i be the random variable $t_i \oplus H(s_i)$.

We consider the following events.

- FBad denotes the event that
  - A G-oracle query r_0 was made by A_1 in step 3-1, and $G_{r_0} \neq s_0 \oplus (m_0||0^{k_1})$, or

  - A G-oracle query r_1 was made by A_1 in step 3-1, and $G_{r_1} \neq s_1 \oplus (m_1||0^{k_1})$.

- GBad denotes the event that
  - A G-oracle query r_0 was made by A_2 in step 3-2, and at the point in time that it was made, the H-oracle query s_0 was not on the H-list, and $G_{r_0} \neq s_0 \oplus (m_0||0^{k_1})$, or
  - A G-oracle query r_1 was made by A_2 in step 3-2, and at the point in time that it was made, the H-oracle query s_1 was not on the H-list, and $G_{r_1} \neq s_1 \oplus (m_1||0^{k_1})$.
- DSBad denotes the event that
  - A $\mathcal{D}_{sk_0}$ query is not correctly answered, or
  - A $\mathcal{D}_{sk_1}$ query is not correctly answered.
- DABad denotes the event that
  - A $\mathcal{DA}_{sk_0}$ query is not correctly answered, or
  - A $\mathcal{DA}_{sk_1}$ query is not correctly answered.
- DBad = DSBad $\vee$ DABad.
- G = $\neg$FBad $\wedge$ $\neg$GBad $\wedge$ $\neg$DBad.

We use the events FBad, GBad, and G for proving Lemma 9.2 described below. In this chapter, we omit the proof of Lemma 9.2 since the proof of this lemma is similar to that for RSA-RAEP.

We let Pr[$\cdot$] denote the probability distribution in the game defining advantage. We introduce the following additional events:

- YBad denotes the event that $y \notin (\mathbb{Z}_{N_0}^* \cap \mathbb{Z}_{N_1}^*)$.
- FAskS denotes the event that H-oracle query s_0 or s_1 was made by A_1 in step 3-1.
- AskR denotes the event that (r_0, G_{r_0}) or (r_1, G_{r_1}) is on the G-list at the end of step 3-2.
- AskS denotes the event that (s_0, H_{s_0}) or (s_1, H_{s_1}) is on the H-list at the end of step 3-2.

We use the event FAskS for proving Lemma 9.2. In this chapter, we omit the proof of Lemma 9.2 since the proof of this lemma is similar to that for RSA-RAEP.

Now, we analyze the advantage of M. The algorithm M wins the game if it outputs s_b. If (s_b, H_{s_b}) is on the H-list, then M outputs s_b with probability at least $1/q_{\mathrm{hash}}$. Thus,

$$\begin{aligned}
&\mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{RSA},M}(k) \\
&\geq \tfrac{1}{q_{\mathrm{hash}}} \cdot \Pr[(s_b, H_{s_b}) \text{ is on the } H\text{-list}] \\
&= \tfrac{1}{2q_{\mathrm{hash}}} \cdot (\Pr[(s_0, H_{s_0}) \text{ is on the } H\text{-list}|b=0] + \Pr[(s_1, H_{s_1}) \text{ is on the } H\text{-list}|b=1]) \\
&\geq \tfrac{1}{2q_{\mathrm{hash}}} \cdot \Pr[\neg\mathsf{YBad}] \cdot (\Pr_1[(s_0, H_{s_0}) \text{ is on the } H\text{-list}|b=0] \\
&\qquad\qquad + \Pr_1[(s_1, H_{s_1}) \text{ is on the } H\text{-list}|b=1])
\end{aligned}$$

where $\Pr_1[\cdot]$ denote the probability distribution in the simulated game where $\neg\mathsf{YBad}$ occurs. Assuming that $\neg\mathsf{YBad}$ occurs, by the random choice of b and symmetry, we have $\Pr_1[(s_i, H_{s_i}) \text{ is on the } H\text{-list}|b=0] = \Pr_1[(s_i, H_{s_i}) \text{ is on the } H\text{-list}|b=1] = \Pr_1[(s_i, H_{s_i}) \text{ is on the } H\text{-list}]$ for $i \in \{0,1\}$. Therefore,

$$\begin{aligned}
&\mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{RSA},M}(k) \\
&\geq \tfrac{1}{2q_{\mathrm{hash}}} \cdot \Pr[\neg\mathsf{YBad}] \cdot (\Pr_1[(s_0, H_{s_0}) \text{ is on the } H\text{-list}] + \Pr_1[(s_1, H_{s_1}) \text{ is on the } H\text{-list}]) \\
&\geq \tfrac{1}{2q_{\mathrm{hash}}} \cdot \Pr[\neg\mathsf{YBad}] \cdot \Pr_1[\mathsf{AskS}].
\end{aligned}$$

We next bound $\Pr_1[\mathsf{AskS}]$. We can bound $\Pr_1[\mathsf{AskS}]$ in a similar way as in the proof of the anonymity for RSA-RAEP [3], and we have

$$\Pr_1[\mathsf{AskS}] \geq \frac{1}{2} \cdot \Pr_1[\mathsf{AskR} \wedge \mathsf{AskS}|\neg\mathsf{DBad}] \cdot \Pr_1[\neg\mathsf{DBad}|\neg\mathsf{AskS}].$$

We next bound $\Pr_1[\mathsf{AskR} \wedge \mathsf{AskS}|\neg\mathsf{DBad}]$. Let $\epsilon = \mathbf{Adv}^{\text{key-cca}}_{\mathcal{UAPE}^{\mathsf{RO}},A}(k)$. The proof of the following lemma is similar to that for RSA-RAEP. Intuitively, this lemma states that if M simulates the decryption oracle for the adversary A perfectly, then A makes queries (r, G_r) and (s, H_s) such that $s = (m||0^{k_1}) \oplus G_r$ and $y = (s||(r \oplus H_s))^{e_b} \bmod N_b$ with non-negligible probability.

Lemma 9.2.

$$\Pr_1[\mathsf{AskR} \wedge \mathsf{AskS}|\neg\mathsf{DBad}] \geq \frac{\epsilon}{2} \cdot \left(1 - \left(\frac{2q_{\mathrm{gen}}}{2^{k_0}} + \frac{2q_{\mathrm{hash}}}{2^{n+k_1}}\right)\right) - \frac{2q_{\mathrm{gen}}}{2^k}.$$

We next bound $\Pr_1[\neg\mathsf{DBad}|\neg\mathsf{AskS}]$. It is easy to see that

$$\Pr_1[\neg\mathsf{DBad}|\neg\mathsf{AskS}] \leq \Pr_1[\neg\mathsf{DSBad}|\neg\mathsf{AskS}] + \Pr_1[\neg\mathsf{DABad}|\neg\mathsf{AskS}],$$

and the proof of the following lemma is similar to that for RSA-RAEP. Intuitively, this lemma states that M can simulate the decryption oracle for standard ciphertexts with overwhelming probability.

Lemma 9.3.

$$\Pr_1[\mathsf{DSBad}|\neg\mathsf{AskS}] \leq q_{\mathrm{dec}} \cdot \left(\frac{2}{2^{k_1}} + \frac{2q_{\mathrm{gen}}+1}{2^{k_0}}\right).$$

Furthermore, we can prove the following lemma in a similar way as that for Lemma 9.3. Intuitively, this lemma states that M can simulate the decryption oracle for anonymized ciphertexts with overwhelming probability.

Lemma 9.4.

$$\Pr_1[\mathsf{DABad}|\neg\mathsf{AskS}] \leq q'_{\mathrm{dec}} \cdot \left(\frac{2}{2^{k_1}} + \frac{2q_{\mathrm{gen}}+1}{2^{k_0}}\right).$$

By applying Lemmas 9.2, 9.3, and 9.4, we can bound $\Pr_1[\mathsf{AskS}]$ as

$$\begin{aligned}
&\Pr_1[\mathsf{AskS}] \\
&\geq \tfrac{1}{2} \cdot \left[\tfrac{\epsilon}{2} \cdot \left(1 - \left(\tfrac{2q_{\mathrm{gen}}}{2^{k_0}} + \tfrac{2q_{\mathrm{hash}}}{2^{n+k_1}}\right)\right) - \tfrac{2q_{\mathrm{gen}}}{2^k}\right] \times \left[1 - (q_{\mathrm{dec}} + q'_{\mathrm{dec}}) \cdot \left(\tfrac{2}{2^{k_1}} + \tfrac{2q_{\mathrm{gen}}+1}{2^{k_0}}\right)\right] \\
&= \tfrac{\epsilon}{4} \cdot \left(1 - \left(\tfrac{2q_{\mathrm{gen}}}{2^{k_0}} + \tfrac{2q_{\mathrm{hash}}}{2^{n+k_1}}\right)\right) \times \left[1 - (q_{\mathrm{dec}} + q'_{\mathrm{dec}}) \cdot \left(\tfrac{2}{2^{k_1}} + \tfrac{2q_{\mathrm{gen}}+1}{2^{k_0}}\right)\right] \\
&\qquad\qquad - \tfrac{1}{2} \cdot \tfrac{2q_{\mathrm{gen}}}{2^k} \cdot \left[1 - (q_{\mathrm{dec}} + q'_{\mathrm{dec}}) \cdot \left(\tfrac{2}{2^{k_1}} + \tfrac{2q_{\mathrm{gen}}+1}{2^{k_0}}\right)\right] \\
&\geq \tfrac{\epsilon}{4} \cdot \left(1 - \left(\tfrac{2q_{\mathrm{gen}}}{2^{k_0}} + \tfrac{2q_{\mathrm{hash}}}{2^{n+k_1}}\right) - (q_{\mathrm{dec}} + q'_{\mathrm{dec}}) \cdot \left(\tfrac{2}{2^{k_1}} + \tfrac{2q_{\mathrm{gen}}+1}{2^{k_0}}\right)\right) - \tfrac{1}{2} \cdot \tfrac{2q_{\mathrm{gen}}}{2^k} \\
&= \tfrac{\epsilon}{4} \cdot \left(1 - \tfrac{2q_{\mathrm{gen}}+q_{\mathrm{dec}}+q'_{\mathrm{dec}}+2q_{\mathrm{gen}}(q_{\mathrm{dec}}+q'_{\mathrm{dec}})}{2^{k_0}} - \tfrac{2(q_{\mathrm{dec}}+q'_{\mathrm{dec}})}{2^{k_1}} - \tfrac{2q_{\mathrm{hash}}}{2^{k-k_0}}\right) - \tfrac{q_{\mathrm{gen}}}{2^k}.
\end{aligned}$$

We next bound the probability that $\neg\mathsf{YBad}$ occurs.

Lemma 9.5.

$$\Pr[\mathsf{YBad}] \leq \frac{2}{2^{k/2-3}-1} + \frac{1}{2^{159}}.$$

Lemma 9.5. Let $N = pq$ and $N' = p'q'$. Note that $2^{\lceil k/2\rceil-1} < p, q, p', q' < 2^{\lceil k/2\rceil}$ and $2^{k-1} < N, N' < 2^k$. We define a set $S[N]$ as $\{\tilde{Y}|\tilde{Y} \in [0, 2^{k+160}) \wedge (\tilde{Y} \bmod N) \in \mathbb{Z}_N^*\}$. Then, we have

$$\begin{aligned}
&\Pr[\mathsf{YBad}] \\
&= \Pr[y \xleftarrow{R} \mathbb{Z}_N^*;\ \mu \xleftarrow{R} \{0, 1, 2, \ldots, \lfloor(2^{k+160} - y)/N\rfloor\};\ Y \leftarrow y + \mu N :\ Y \notin S[N']] \\
&\leq \Pr[Y' \xleftarrow{R} S[N] :\ Y' \notin S[N']] + 1/2^{159}
\end{aligned}$$

since the distribution of Y' is statistically indistinguishable from that of Y, and the statistically distance is less than $1/2^{159}$.

Since $2^{160} \cdot \phi(N) \leq |S[N]|$, we have

$$\begin{aligned}
\Pr[Y' \xleftarrow{R} S[N] : Y' \notin S[N']] &\leq \frac{|\{y \,|\, y \in S[N] \ \wedge\ y \notin S[N']\}|}{|S[N]|} \\
&\leq \frac{|\{y \,|\, y \in [0, 2^{k+160}) \ \wedge\ y \notin S[N']\}|}{|S[N]|} \\
&\leq \frac{2^{k+160} - |S[N']|}{|S[N]|} \leq \frac{2^{k+160} - |S[N']|}{2^{160} \cdot \phi(N)}.
\end{aligned}$$

Furthermore, we have

$$\begin{aligned} 2^{k+160} - |S[N']| &= \left|\{Y' | Y' \in [0, 2^{k+160}) \wedge (Y' \bmod N') \notin \mathbb{Z}^*_{N'}\}\right| \\ &\leq \left|\{Y' | Y' \in [0, 2N' \cdot 2^{160}) \wedge (Y' \bmod N') \notin \mathbb{Z}^*_{N'}\}\right| \\ &= 2^{161} \times \left|\{Y' | Y' \in [0, N') \wedge Y' \notin \mathbb{Z}^*_{N'}\}\right| \\ &= 2^{161}(N' - \phi(N')). \end{aligned}$$

Therefore, we can bound $\Pr[Y' \xleftarrow{R} S[N] : \ Y' \notin S[N']]$ as

$$\begin{aligned} &\Pr[Y' \xleftarrow{R} S[N] : \ Y' \notin S[N']] \\ &\leq \frac{2^{k+160} - |S[N']|}{2^{160} \cdot \phi(N)} \leq \frac{2^{161}(N' - \phi(N'))}{2^{160} \cdot \phi(N)} = \frac{2(p' + q' - 1)}{N - p - q + 1} \leq \frac{2(p' + q')}{N - p - q} \\ &\leq \frac{2(2^{\lceil k/2 \rceil} + 2^{\lceil k/2 \rceil})}{2^{k-1} - 2^{\lceil k/2 \rceil} - 2^{\lceil k/2 \rceil}} = \frac{2(1+1)}{2^{k-1-\lceil k/2 \rceil} - 1 - 1} \leq \frac{4}{2^{k/2-2} - 2} = \frac{2}{2^{k/2-3} - 1}. \end{aligned}$$

□

Substituting the bounds for the above probabilities, we have

$$\mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{RSA},M}(k) \geq \frac{1}{2q_{\text{hash}}} \cdot (1 - \epsilon_1) \cdot \left(\frac{\epsilon}{4} \cdot (1 - \epsilon_2) - \frac{q_{\text{gen}}}{2^k}\right)$$

where $\epsilon_1 = \frac{2}{2^{k/2-3}-1} + \frac{1}{2^{159}}$ and $\epsilon_2 = \frac{2q_{\text{gen}} + q_{\text{dec}} + q'_{\text{dec}} + 2q_{\text{gen}}(q_{\text{dec}} + q'_{\text{dec}})}{2^{k_0}} + \frac{2(q_{\text{dec}} + q'_{\text{dec}})}{2^{k_1}} + \frac{2q_{\text{hash}}}{2^{k-k_0}}$, and re-arranging the terms, we get the claimed result. Note that $\epsilon = \mathbf{Adv}^{\text{key-cca}}_{\mathcal{UAPE}^{\mathsf{RO}},A}(k)$.

Finally, we estimate the time complexity of M. It is the time complexity of A plus the time for simulating the random oracles. In the random oracle simulation, for each pair $((g, G_g), (h, H_h))$, it is sufficient to compute $y_{h,g,0} = (h||(g \oplus H_h))^{e_0} \bmod N_0$ and $y_{h,g,1} = (h||(g \oplus H_h))^{e_1} \bmod N_1$. Therefore, the time complexity of M is that of A plus $q_{\text{gen}} \cdot q_{\text{hash}} \cdot O(k^3)$.

□

CHAPTER 10

Conclusion

In this thesis, we have focused on the security property for encryption and signature schemes, called "anonymity."

In Chapters 2 to 5, we have studied the techniques which can be used to obtain the anonymity property. We have proposed two techniques for anonymity. We have also constructed the schemes for public-key encryption, undeniable and confirmer, and ring signature, by applying our proposed techniques.

In Chapter 2, we have provided the RSA family of trap-door permutations with a common domain. The domain and range of RSACD are common to each user when each user has an RSA modulus of the same size. We have proved that the θ-partial one-wayness of RSACD is equivalent to the one-wayness of RSACD for $\theta > 0.5$, and that the one-wayness of RSACD is equivalent to the one-wayness of RSA. We have also proposed a new technique for obtaining the anonymity property of RSA-based cryptosystems, which we call "sampling twice." In our technique, we employ an algorithm `ChooseAndShift` which takes two numbers $x_1, x_2 \in \mathbb{Z}_N$ as input and returns a value $y \in [0, 2^k)$ where $|N| = k$. Then, for any N where $|N| = k$, the output is uniformly distributed over $[0, 2^k)$ if x_1 and x_2 are independently and uniformly chosen from $\mathbb{Z}_N$.

We have proposed new schemes for public-key encryption in Chapter 3, those for undeniable and confirmer signature in Chapter 4, and those for ring signature in Chapter 5, by applying the previously proposed and our new techniques, repeating, expanding, RSACD,

sampling twice. We have also proved the anonymity property and other required security of the schemes.

We again describe the (dis)advantage of the schemes with four techniques.

The scheme with repeating is efficient with respect to the sizes of ciphertexts and signatures, the computational costs to encrypt messages and to sign messages in the average case, and those to decrypt ciphertexts and to verify signatures. However, it is inefficient with respect to the computational costs to encrypt messages and to sign messages in the worst case. In order to obtain the anonymity property, it is necessary for each user to choose a public key with almost the same size.

The scheme with expanding provides anonymity even if each user uses the public key of different length. It is efficient with respect to the computational costs to encrypt messages, to sign messages, to decrypt ciphertexts, and to verify signatures. However, the sizes of ciphertexts and signatures are larger than those of the other schemes.

The scheme with RSACD is efficient with respect to the sizes of ciphertexts and signatures, and the computational costs to encrypt messages and to sign messages. However, it is inefficient with respect to the computational costs to decrypt a ciphertext and to verify a signature. In order to obtain the anonymity property, it is necessary for each user to choose a public key with exact the same size.

The scheme with sampling twice is efficient with respect to the sizes of ciphertexts and signatures, the computational costs to decrypt ciphertexts and to verify signatures, and the computational costs to encrypt messages and to sign messages in the worst case. However, the number of exponentiations for encryption or signing is two, while that of the other schemes is one or 1.5 in the average case. Similar to the scheme with RSACD, in order to obtain the anonymity property, it is necessary for each user to choose a public key with exact the same size.

In this thesis, we have not succeeded to construct the undeniable and confirmer signature scheme with anonymity by applying the RSACD function. It might be interesting to construct such schemes.

It would be also interesting to consider other applications of our proposed techniques. There are many schemes which required the anonymity property, such as (hybrid) ID-based encryption [11, 2], group signatures [23], anonymous group identification [32, 60] signcryption [82, 13], designated verifier signature [53, 59], and so on. Our proposed techniques seem to be useful to construct such schemes with the anonymity property.

In Chapter 6, we have considered the schemes with anonymity using the Paillier's bijective function. We have applied the four techniques described above in the case using

the Paillier's bijective function instead of the RSA function. We have constructed a family of Paillier's trap-door permutations and that with a common, and prove the properties of them. We have also proposed the public-key encryption schemes with the above families of permutations by applying the four techniques, that is, Paillier-OAEP (OAEP with Paillier's trap-door permutation) with repeating, that with expanding, that with sampling twice, and PCD-OAEP (OAEP with Paillier's trap-door permutation with a common domain).

It would be interesting to consider the construction of families of trap-door permutations with a common domain based on the variants of Paillier's permutation. After the paper of Paillier, several variants of Paillier's scheme were proposed. Catalano, Gennaro, Howgrave-Graham, and Nguyen [19] proposed a mix of Paillier's scheme with the RSA scheme, in order to obtain an IND-CPA cryptosystem in the standard model with efficiency similar to that of the RSA cryptosystem. It is based on the permutation $(m, r) \mapsto r^e(1+mN) \bmod N^2$ where $\gcd(e, \lambda(N^2)) = 1$. The encryption scheme is semantically secure under the Decisional Small e-Residues assumption. Galindo, Martín, Morillo, and Villar [46] proposed a encryption scheme based on the permutation: $(m, r) \mapsto r^{2e}+mN \bmod N^2$ where $p = q = 3 \bmod 4$ and $\gcd(e, \lambda(N)) = 1$. This function is one-way under the Factoring assumption. Damgård and Jurik [30] proposed a generalization of Paillier's scheme, in which the expansion factor is reduced and which allows to adjust the block length of the scheme even after the public key has been fixed, without loosing the homomorphic property. They also constructed its threshold variant. Forque, Poupard and Stern [40] also proposed the threshold version of Paillier's scheme.

In Chapter 7, we have proposed the new security notion for public-key encryption with anonymity, called "strong anonymity," and show the relationships between the data-privacy and the key-privacy for public-key encryption schemes. From our results, we have that the strong anonymity (sIK) is a sufficient condition in order to satisfy that a public-key encryption scheme provides the data-privacy (IND) and the key-privacy (IK).

The motivation to propose the strong anonymity is capturing the situation that the schemes whose message spaces for each public-key are different provide the anonymity property. However, our proposed security notion not only captures such situation but also implies the data-privacy (the indistinguishability of ciphertexts). Therefore, it might be interesting to consider a security notion which captures the above situation, while it does not implies the data-privacy.

In Chapter 8 we have proposed the notion of plaintext awareness in the two-key setting, called PATK, and proved that if a public-key encryption scheme is secure in the sense of PATK, then it is also secure in the sense of IK-CCA. Since it looks much easier to prove that a public-key encryption scheme is secure in the sense of PATK than to prove directly that it is secure in the sense of IK-CCA, the notion of PATK is useful to prove the anonymity

property of public-key encryption schemes. The previously proposed public-key encryption schemes in [3, 50, 51] which are based on RSA-OAEP and secure in the sense of IK-CCA seem to meet PAKE.

We have also proposed the first generic conversion scheme for the anonymity from IK-CPA to IK-CCA. More precisely, we have proved that the public-key encryption scheme derived from the Fujisaki-Okamoto conversion scheme, where the basic public-key encryption scheme is secure in the sense of IK-CPA, is secure in the sense of IK-CCA in the random oracle model. Recently, Bellare and Palacio [5] proposed the definition of the plaintext-awareness in the standard model (i.e. without random oracles). Dent [33] showed that the Cramer-Shoup hybrid encryption scheme [27] satisfies the plaintext-awareness in the standard model. It might be interesting to consider the definition of the plaintext awareness in the two-key setting without random oracles and the schemes in the standard model which meet the plaintext awareness in the two-key setting.

In Chapter 9, we have formalized a special type of public-key encryption scheme called a universally anonymizable public-key encryption scheme. A universally anonymizable public-key encryption scheme consists of a standard public-key encryption scheme $\mathcal{PE}$ and two additional algorithms, that is, an anonymizing algorithm $\mathcal{UA}$ and a decryption algorithm $\mathcal{DA}$ for anonymized ciphertexts. We can use $\mathcal{PE}$ as a standard encryption scheme which is not necessary to have the anonymity property. Furthermore, in this scheme, by using the anonymizing algorithm $\mathcal{UA}$, anyone who has a standard ciphertext can anonymize it with its public key whenever she wants to do that. The receiver can decrypt the anonymized ciphertext by using the decryption algorithm $\mathcal{DA}$ for anonymized ciphertexts. Then, the adversary cannot know under which key the anonymized ciphertext was created.

We have also proposed the universally anonymizable public-key encryption schemes based on the ElGamal encryption scheme, the Cramer-Shoup encryption scheme, and RSA-OAEP, and proved their security. It might be interesting to consider the application of our proposed primitive, or construct other schemes for universally anonymizable public-key encryption.

Bibliography

[1] Abadi, M. and Rogaway, P. Reconciling Two Views of Cryptography (The Computational Soundness of Formal Encryption). In *Proceedings of the First IFIP International Conference on Theoretical Computer Science* (Sendai, Japan, August 2000), vol. 1872 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 3–22.

[2] Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P., and Shi, H. Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions. In *Advances in Cryptology – CRYPTO 2005* (Santa Barbara, California, USA, August 2005), V. Shoup, Ed., vol. 3621 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 205–222.

[3] Bellare, M., Boldyreva, A., Desai, A., and Pointcheval, D. Key-Privacy in Public-Key Encryption. In Boyd [12], pp. 566–582. Full version of this paper, available via `http://www-cse.ucsd.edu/users/mihir/`.

[4] Bellare, M., Desai, A., Pointcheval, D., and Rogaway, P. Relations among Notions of Security for Public-Key Encryption Schemes. In Krawczyk [58], pp. 26–45.

[5] Bellare, M. and Palacio, A. Towards Plaintext-Aware Public-Key Encryption without Random Oracles. In *Advances in Cryptology – ASIACRYPT 2004* (Jeju Island, Korea, December 2004), P. J. Lee, Ed., vol. 3329 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 48–62.

[6] Bellare, M. and Rogaway, P. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *1st ACM Conference on Computer and Communications Security* (Fairfax, Virginia, USA, 1993), ACM, pp. 62–73.

[7] Bellare, M. and Rogaway, P. Optimal Asymmetric Encryption – How to Encrypt with RSA. In De Santis [31], pp. 92–111.

[8] Bellare, M. and Rogaway, P. The Exact Security of Digital Signatures: How to Sign with RSA and Rabin. In Maurer [62], pp. 339–416.

[9] Boneh, D. Simplified OAEP for the RSA and Rabin functions. In Kilian [55], pp. 275–291.

[10] Boneh, D., Ed. *Advances in Cryptology – CRYPTO 2003* (Santa Barbara, California, USA, August 2003), vol. 2729 of *Lecture Notes in Computer Science*, Springer-Verlag.

[11] Boneh, D. and Franklin, M. K. Identity-Based Encryption from the Weil Pairing. In Kilian [55], pp. 213–229.

[12] Boyd, C., Ed. *Advances in Cryptology – ASIACRYPT 2001* (Gold Coast, Australia, December 2001), vol. 2248 of *Lecture Notes in Computer Science*, Springer-Verlag.

[13] Boyen, X. Multipurpose Identity-Based Signcryption: A Swiss Army Knife for Identity-Based Cryptography. In Boneh [10], pp. 382–398.

[14] Bresson, E., Stern, J., and Szydlo, M. Threshold Ring Signatures and Applications to Ad-hoc Groups. In Yung [80], pp. 465–480.

[15] Camenisch, J. and Lysyanskaya, A. Efficient Non-Transferable Anonymous Multi-Show Credential System with Optional Anonymity Revocation. In *Advances in Cryptology – EUROCRYPT 2001* (Innsbruck, Austria, May 2001), B. Pfitzmann, Ed., vol. 2045 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 93–118.

[16] Camenisch, J. and Michels, M. Confirmer Signature Schemes Secure against Adaptive Adversaries. In Preneel [71], pp. 243–258.

[17] Camenisch, J. and Michels, M. Confirmer Signature Schemes Secure against Adaptive Adversaries. In Preneel [71], pp. 243–258.

[18] Canetti, R., Krawczyk, H., and Nielsen, J. B. Relaxing Chosen-Ciphertext Security. In Boneh [10], pp. 565–582.

[19] Catalano, D., Gennaro, R., Howgrave-Graham, N., and Nguyen, P. Q. Paillier's Cryptosystem Revisited. In *Proceedings of the 8th ACM conference on Computer and Communications Security* (2001), pp. 206–214.

[20] Chaum, D. Zero-Knowledge Undeniable Signatures. In *Advances in Cryptology – EUROCRYPT '90* (Aarhus, Denmark, May 1990), I. Damgård, Ed., vol. 473 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 458–464.

[21] Chaum, D. Designated Confirmer Signatures. In De Santis [31], pp. 86–91.

[22] Chaum, D. and Antwerpen, H. V. Undeniable Signatures. In *Advances in Cryptology – CRYPTO '89* (Santa Barbara, California, USA, August 1989), G. Brassard, Ed., vol. 435 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 212–217.

[23] Chaum, D. and van Heyst, E. Non-Interactive Zero Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In Feigenbaum [38], pp. 433–444.

[24] Chaum, D., van Heyst, E., and Pfitzmann, B. Cryptographically Strong Undeniable Signatures, Unconditionally Secure for the Signer. In Feigenbaum [38], pp. 470–484.

[25] Coron, J.-S., Handschuh, H., Joye, M., Paillier, P., Pointcheval, D., and Tymen, C. GEM: A Generic Chosen-Ciphertext Secure Encryption Method. In Preneel [72], pp. 263–276.

[26] Cramer, R. and Shoup, V. A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. In Krawczyk [58], pp. 13–25.

[27] Cramer, R. and Shoup, V. Design and Analysis of Practical Public-Key Encryption Schemes Secure Against Adaptive Chosen Ciphertext Attack. *SIAM Journal on Computing 33*, 1 (2004), 167–226.

[28] Cui, Y., Kobara, K., and Imai, H. Compact Conversion Schemes for the Probabilistic OW-PCA Primitives. In *Information and Communications Security, 5th International Conference, ICICS 2003* (Huhehaote, China, October 2003), S. Qing, D. Gollmann, and J. Zhou, Eds., vol. 2836 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 269–279.

[29] Cui, Y., Kobara, K., and Imai, H. A Generic Conversion with Optimal Redundancy. In *Topics in Cryptology – CT-RSA 2005* (San Francisco, CA, USA, February 2005), A. Menezes, Ed., vol. 3376 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 104–117.

[30] Damgård, I. and Jurik, M. A Generalisation, a Simplification and some Applications of Paillier's Probabilistic Public-Key System. In Kim [56], pp. 119–136.

[31] De Santis, A., Ed. *Advances in Cryptology – EUROCRYPT '94* (Perugia, Italy, May 1994), vol. 950 of *Lecture Notes in Computer Science*, Springer-Verlag.

[32] De Santis, A., Di Crescenzo, G., and Persiano, G. Communication-Efficient Anonymous Group Identification. In *ACM Conference on Computer and Communications Security* (1998), pp. 73–82.

[33] Dent, A. W. Cramer-Shoup is Plaintext-Aware in the Standard Model. In *Advances in Cryptology – EUROCRYPT 2006* (St. Petersburg, Russia, May 2006), S. Vaudenay, Ed., vol. 4004 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 289–307.

[34] Desai, A. The Security of All-or-Nothing Encryption: Protecting against Exhaustive Key Search. In *Advances in Cryptology – CRYPTO 2000* (Santa Barbara, California, USA, August 2000), M. Bellare, Ed., vol. 1880 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 359–375.

[35] Desmedt, Y. Securing traceability of ciphertexts: Towards a secure software escrow scheme. In *Advances in Cryptology – EUROCRYPT '95* (Saint-Malo, France, May 1995), L. C. Guillou and J.-J. Quisquater, Eds., vol. 921 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 147–157.

[36] Diffie, W. and Hellman, M. E. New Directions in Cryptography. *IEEE Transactions on Information Theory IT-22* (1976), 644–654.

[37] Dolev, D., Dwork, C., and Naor, M. Non-Malleable Cryptography. *SIAM Journal on Computing 30*, 2 (2000), 391–437.

[38] Feigenbaum, J., Ed. *Advances in Cryptology – CRYPTO '91* (Santa Barbara, California, USA, August 1991), vol. 576 of *Lecture Notes in Computer Science*, Springer-Verlag.

[39] Fischlin, M. Pseudorandom Function Tribe Ensembles Based on One-Way Permutations. In Stern [79], pp. 432–445.

[40] Fouque, P., Poupard, G., and Stern, J. Sharing Decryption in the Context of Voting or Lotteries. In *Financial Cryptography – FC 2000* (Anguilla, British West Indies, February 2000), Y. Frankel, Ed., vol. 1962 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 90–104.

[41] Fujisaki, E. Plaintext-Simulatability. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Special Section on Cryptography and Information Security E89-A*, 1 (January 2006), 55–65.

[42] Fujisaki, E. and Okamoto, T. Secure Integration of Asymmetric and Symmetric Encryption Schemes. In *Advances in Cryptology – CRYPTO '99* (Santa Barbara, California, USA, August 1999), M. Wiener, Ed., vol. 1666 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 537–554.

[43] Fujisaki, E., Okamoto, T., Pointcheval, D., and Stern, J. RSA-OAEP is Secure under the RSA Assumption. In Kilian [55], pp. 260–274.

[44] Galbraith, S. D. and Mao, W. Invisibility and Anonymity of Undeniable and Confirmer Signatures. In *Topics in Cryptology – CT-RSA 2003* (San Francisco, CA, USA, April 2003), M. Joye, Ed., vol. 2612 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 80–97.

[45] Galbraith, S. D., Mao, W., and Paterson, K. G. RSA-based Undeniable Signatures for General Moduli. In Preneel [72], pp. 200–217.

[46] Galindo, D., Mollevĺ, S. M., Morillo, P., and Villar, J. L. A Practical Public Key Cryptosystem from Paillier and Rabin Schemes. In *PKC 2003 – 6th International Workshop on Theory and Practice in Public Key Cryptography* (Miami, Florida, USA, January 2003), Y. Desmedt, Ed., vol. 2567 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 279–291.

[47] Gennaro, R., Krawczyk, H., and Rabin, T. RSA-based Undeniable Signatures. In Kaliski [54], pp. 132–149.

[48] Goldwasser, S. and Micali, S. Probabilistic Encryption. *Journal of Computer and System Sciences 28* (April 1984), 270–299.

[49] Halevi, S. A Sufficient Condition for Key-Privacy. IACR Cryptology ePrint Archive, `http://eprint.iacr.org/2005/005.pdf`, 2005.

[50] Hayashi, R., Okamoto, T., and Tanaka, K. An RSA Family of Trap-door Permutations with a Common Domain and its Applications. In *PKC 2004 – 7th International Workshop on Theory and Practice in Public Key Cryptography* (Singapore, March 2004), F. Bao, R. H. Deng, and J. Zhou, Eds., vol. 2947 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 291–304.

[51] Hayashi, R. and Tanaka, K. The Sampling Twice Technique for the RSA-based Cryptosystems with Anonymity. In *PKC 2005 – 8th International Workshop on Theory and Practice in Public Key Cryptography* (Les Diablerets, Switzerland, January 2005), S. Vaudenay, Ed., vol. 3386 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 216–233.

[52] Imai, H. and Zheng, Y., Eds. *PKC 2000 – 3rd International Workshop on Theory and Practice in Public Key Cryptography* (Melbourne, Victoria, Australia, January 2000), vol. 1751 of *Lecture Notes in Computer Science*, Springer-Verlag.

[53] Jakobsson, M., Sako, K., and Impagliazzo, R. Designated Verifier Proofs and their Applications. In Maurer [62], pp. 143–154.

[54] Kaliski, Jr., B. S., Ed. *Advances in Cryptology – CRYPTO '97* (Santa Barbara, California, USA, August 1997), vol. 1294 of *Lecture Notes in Computer Science*, Springer-Verlag.

[55] Kilian, J., Ed. *Advances in Cryptology – CRYPTO 2001* (Santa Barbara, California, USA, August 2001), vol. 2139 of *Lecture Notes in Computer Science*, Springer-Verlag.

[56] Kim, K., Ed. *PKC 2001 – 4th International Workshop on Theory and Practice in Public Key Cryptography* (Cheju Island, Korea, February 2001), vol. 1992 of *Lecture Notes in Computer Science*, Springer-Verlag.

[57] Krawczyk, H. SKEME: A Versatile Secure Key Exchange Mechanism for Internet. In *Proceedings of the 1996 Internet Society Symposium on Network and Distributed System Security* (San Diego, CA, USA, February 1996), pp. 114–127.

[58] Krawczyk, H., Ed. *Advances in Cryptology – CRYPTO '98* (Santa Barbara, California, USA, August 1998), vol. 1462 of *Lecture Notes in Computer Science*, Springer-Verlag.

[59] Laguillaumie, F. and Vergnaud, D. Designated Verifier Signatures: Anonymity and Efficient Construction from Any Bilinear Map. In *Security in Communication Networks, 4th International Conference, SCN 2004* (Amalfi, Italy, September 2005), C. Blundo and S. Cimato, Eds., vol. 3352 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 105–119.

[60] Lee, C. H., Deng, X., and Zhu, H. Design and Security Analysis of Anonymous Group Identification Protocols. In *PKC 2002 – 5th International Workshop on Theory and Practice in Public Key Cryptography* (Paris, France, February 2002), D. Naccache and P. Paillier, Eds., vol. 2274 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 188–198.

[61] Lucks, S. A Variant of the Cramer-Shoup Cryptosystem for Groups of Unknown Order. In Zheng [83], pp. 27–45.

[62] Maurer, U., Ed. *Advances in Cryptology – EUROCRYPT '96* (Saragossa, Spain, May 1996), vol. 1070 of *Lecture Notes in Computer Science*, Springer-Verlag.

[63] Michels, M. and Stadler, M. Generic Constructions for Secure and Efficient Confirmer Signature Schemes. In *Advances in Cryptology – EUROCRYPT '98* (Espoo, Finland, May 1998), K. Nyberg, Ed., vol. 1403 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 406–421.

[64] Naor, M. Deniable Ring Authentication. In Yung [80], pp. 481–498.

[65] Naor, M. and Yung, M. Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In *Proceedings on 22nd Annual ACM Symposium on Theory of Computing (STOC '90)* (New Orleans, Louisiana, USA, May 1990), ACM, pp. 427–437.

[66] Okamoto, T. and Pointcheval, D. REACT: Rapid Enhanced-Security Asymmetric Cryptosystem Transform. In *Topics in Cryptology – CT-RSA 2003* (San Francisco, CA, USA, April 2001), D. Naccache, Ed., vol. 2020 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 159–175.

[67] Okamoto, T. and Pointcheval, D. The Gap-Problems: A New Class of Problems for the Security of Cryptographic Schemes. In Kim [56], pp. 104–118.

[68] Paillier, P. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In Stern [79], pp. 223–238.

[69] Phan, D. H. and Pointcheval, D. Chosen-Ciphertext Security without Redundancy. In *Advances in Cryptology – ASIACRYPT 2003* (Taipei, Taiwan, November 2003), C. S. Laih, Ed., vol. 2894 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 1–18.

[70] Pointcheval, D. Chosen-Ciphertext Security for Any One-Way Cryptosystem. In Imai and Zheng [52], pp. 129–146.

[71] Preneel, B., Ed. *Advances in Cryptology – EUROCRYPT 2000* (Bruges, Belgium, May 2000), vol. 1807 of *Lecture Notes in Computer Science*, Springer-Verlag.

[72] Preneel, B., Ed. *Topics in Cryptology – CT-RSA 2002* (San Jose, CA, USA, February 2002), vol. 2271 of *Lecture Notes in Computer Science*, Springer-Verlag.

[73] Rackoff, C. and Simon, D. Non-Interactive Zero Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In Feigenbaum [38], pp. 433–444.

[74] Rackoff, C. and Simon, D. R. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In Feigenbaum [38], pp. 433–444.

[75] Rivest, R., Shamir, A., and Adleman, L. M. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM 21*, 2 (1978), 120–126.

[76] Rivest, R. L., Shamir, A., and Tauman, Y. How to Leak a Secret. In Boyd [12], pp. 552–565.

[77] Sako, K. An Auction Protocol Which Hides Bids of Losers. In Imai and Zheng [52], pp. 422–432.

[78] Shoup, V. OAEP Reconsidered. In Kilian [55], pp. 239–259.

[79] Stern, J., Ed. *Advances in Cryptology – EUROCRYPT '99* (Prague, Czech Republic, May 1999), vol. 1592 of *Lecture Notes in Computer Science*, Springer-Verlag.

[80] Yung, M., Ed. *Advances in Cryptology – CRYPTO 2002* (Santa Barbara, California, USA, August 2002), vol. 2442 of *Lecture Notes in Computer Science*, Springer-Verlag.

[81] Zhang, F. and Kim, K. ID-Based Blind Signature and Ring Signature from Pairings . In Zheng [83], pp. 354–368.

[82] Zheng, Y. Digital Signcryption or How to Achieve Cost(Signature & Encryption) << Cost(Signature) + Cost(Encryption). In Kaliski [54], pp. 165–179.

[83] Zheng, Y., Ed. *Advances in Cryptology – ASIACRYPT 2002* (Queenstown, New Zealand, December 2002), vol. 2501 of *Lecture Notes in Computer Science*, Springer-Verlag.

Publications

[1] Hayashi, R., Okamoto, T., and Tanaka, K. An RSA Family of Trap-door Permutations with a Common Domain and its Applications. In *PKC 2004 – 7th International Workshop on Theory and Practice in Public Key Cryptography* (Singapore, March 2004), F. Bao, R. H. Deng, and J. Zhou, Eds., vol. 2947 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 291–304.

[2] Hayashi, R., and Tanaka, K. The Sampling Twice Technique for the RSA-based Cryptosystems with Anonymity. In *PKC 2005 – 8th International Workshop on Theory and Practice in Public Key Cryptography* (Les Diablerets, Switzerland, January 2005), S. Vaudenay, Ed., vol. 3386 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 216–233.

[3] Hayashi, R., and Tanaka, K. Universally Anonymizable Public-Key Encryption. In *Advances in Cryptology – ASIACRYPT 2005* (Chennai, India, December 2005), B. Roy, Ed., vol. 3788 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 293–312.

[4] Hayashi, R., and Tanaka, K. PA in the Two-Key Setting and a Generic Conversion for Encryption with Anonymity. In *Information Security and Privacy, 11th Australasian Conference, ACISP 2006* (Melbourne, Australia, July 2006), L. M. Batten and R. Safavi-Naini, Eds., vol. 4058 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 271–282.

[5] Hayashi, R., and Tanaka, K. Schemes for Encryption with Anonymity and Ring Signature. *IEICE Transactions on Fundamentals of Electronics, Communications*

and Computer Sciences, Special Section on Cryptography and Information Security E89-A, 1 (January 2006), 66–73.