Pseudo-Free Groups and Cryptographic Assumptions

Pseudo-Free Groups and Cryptographic Assumptions

by

Masayuki Fukumitsu

Department of Computer and Mathematical Sciences
Graduate School of Information Sciences
Tohoku University

January 20, 2014

Submitted to Tohoku University
in partial fulfillment of the requirements for the degree of
Ph.D. (Information Sciences)

Abstract

In this thesis, we discuss the adaptive pseudo-free group which is proposed as a unified framework to treat cryptographic assumptions and schemes by Catalano, Fiore and Warinschi. We investigate the adaptive pseudo-freeness of the RSA group $\mathbb{Z}_N^\times$ with respect to several cryptographic assumptions.

First, we consider the flexibility of the notion of adaptive pseudo-free groups. In the definition of Catalano-Fiore-Warinschi, the adaptive pseudo-free group is somewhat restricted in a sense that the adaptive behavior of adversaries is restricted by some specific parametric distribution. However, it remains open whether or not the adaptive pseudo-free group with no such restriction, referred to as the strongly-adaptive pseudo-free group, is feasible. For this question, we give a negative circumstantial evidence. We show that the strong adaptive pseudo-freeness of $\mathbb{Z}_N^\times$ cannot be proven from the strong RSA (SRSA, for short) assumption via algebraic reductions, as long as the SRSA assumption holds. This result indicates that it is reasonable to use parametric distributions to construct a concrete adaptive pseudo-free group.

We next focus on the applicability of the existing parametric distributions. Namely we consider the question whether or not the adaptive pseudo-freeness of $\mathbb{Z}_N^\times$ can be shown from some assumptions other than the SRSA assumption, using the parametric distribution of Catalano-Fiore-Warinschi. We show that it cannot be proven from the RSA assumption via algebraic reductions, as long as the RSA assumption holds. By employing this result, we also show that the security of SRSA-based signature schemes may not be proven from the RSA assumption.

The second result implies that another parametric distribution is required to show the adaptive pseudo-freeness of $\mathbb{Z}_N^\times$ under the RSA assumption. As the third result, we propose such a parametric distribution. Namely, we show that the adaptive pseudo-freeness of $\mathbb{Z}_N^\times$ can be proven from the RSA assumption by using our new parametric distribution.

Acknowledgments

First of all, I would like to express my deepest gratitude to my supervisor, Professor Hiroki Shizuya, for his significant guidance since my student days. Without his continual and technical help, this thesis would not have been possible. It has been a great pleasure to study under his tremendous and long-term guidance.

I would particularly like to appreciate the members of the jury Professor Takafumi Aoki, Professor Xiao Zhou and Associate Professor Shuji Isobe for their insightful comments and invaluable suggestions. They helped me improve the thesis so much.

My sincere gratitude must be offered to Assistant Professor Shingo Hasegawa. He has been my dissertation advisor. With his in-deep discussions and appropriate supports, I have completed this thesis.

I would like to show my cordial appreciation to Associate Professor Masao Sakai and Assistant Professor Eisuke Koizumi. I have received their wide-ranging supports and constructive suggestions.

I would also like to thank Associate Professor Eikoh Chida of Ichinoseki National College of Technology for their essential help and for many things.

I would like to acknowledge with sincere thanks the all-out cooperation and services rendered by the members of the Information Security Seminar and the staff of Center for Information Technology in Education, Tohoku University for many things.

Finally, I would like to thank my parents and my sister. My student life has been supported by all of them with their warm encouragement to me.

Contents

1 Introduction **5**

1.1 Background 5

1.2 Summary of Results 9

1.3 Organization 12

2 Preliminaries **13**

2.1 Mathematical Notations 13

2.2 Pseudo-Free Groups 14

2.2.1 Computational Groups 14

2.2.2 Free Abelian Groups 15

2.2.3 Equations over Groups 15

2.2.4 Pseudo-Freeness 16

2.3 Adaptive Pseudo-Free Groups 17

2.4 Signature Schemes 21

2.5 Hash Functions 22

2.6 Algebraic Algorithms 23

2.7 Cryptographic Assumptions 24

3 Impossibility Results on the Adaptive Pseudo-Freeness of the RSA Group $\mathbb{Z}_N^\times$ **27**

3.1 Impossibility of the Strong Adaptive Pseudo-Freeness of $\mathbb{Z}_N^\times$ . . . 27

3.1.1 Strongly-Adaptive Pseudo-Free Groups 28

3.1.2 Main Theorem 29

3.1.3 Proofs of the Claims in Theorem 3.4 37

3.2 Impossibility of the Adaptive Pseudo-Freeness of $\mathbb{Z}_N^\times$ under the RSA Assumption 38

3.2.1 Main Theorem 38

3.2.2 Proofs of the Claims in Theorem 3.9 47
3.2.3 Impossibility for the SRSA-Based Signature Schemes . . . 48
3.3 Concluding Remarks 60

4 The RSA Group $\mathbb{Z}_N^\times$ Is Adaptive Pseudo-Free under the RSA Assumption 61
4.1 Adaptive Pseudo-Free Groups with respect to a Family of Parametric Distributions 61
4.2 Technical Lemmas 63
4.3 Our Parametric Distribution $\varrho^{K,c}$ 64
4.4 Main Theorem 66
4.5 Concluding Remarks 88

5 Conclusion 90

Chapter 1

Introduction

1.1 Background

The public key cryptosystem was born as a concrete application of the computational complexity theory, and has been one of the fundamental tools to communicate securely over a network. This enables one to omit key exchange and to send secret information even over public channels in which several threats including eavesdropping may exist. Moreover, there are many applications based on public key cryptosystems such as digital signature schemes, identification protocols, oblivious transfer protocols, secure multi-party protocols and so on. Therefore, the public key cryptography has been significant in both theoretical and practical senses.

The concept of public key cryptography was introduced by Diffie and Hellman [16] in 1976. In 1978, Rivest, Shamir and Adleman [39] proposed a public key encryption which is referred to as the RSA scheme. They also proposed a digital signature based on the scheme. Afterward, efficient and useful public key encryptions were put forward [4, 8, 9, 15, 20].

The security of many public key cryptosystems depends on the computational complexity of number-theoretic problems such as the integer factoring problem and the discrete logarithm (DL, for short) problem. This means that the security of a cryptosystem is compromised if the underlying number-theoretic problem can be easily solved. On the other hand, the converse implication does not hold in general. Hence the cryptosystems may be vulnerable even if these problems are in fact hard to solve. Therefore, cryptographic schemes are required to be secure as long as solving the number-theoretic problem is hard.

The *provable security* designates the case where the converse above also holds. The security of the cryptographic schemes is usually proved by reduction methodology. A common security proof proceeds as follows. We put on an assumption X that some computational problem Problem_X is hard to solve, namely it cannot be solved by any probabilistic polynomial-time (PPT, for short) algorithm with nonnegligible probability. Then we prove that there exists a polynomial-time reduction from the problem Problem_X to the problem for breaking the scheme. Such a reduction shows that we can efficiently solve the hard problem Problem_X by using an adversary, that is an attacker who is capable of breaking the scheme, as a black-box oracle. However, this is impossible as long as the hardness assumption on the problem Problem_X holds. Eventually, we have proven that the scheme is secure under the hardness assumption X.

A hardness assumption that is usually employed in the security proofs is called a cryptographic assumption. Among cryptographic assumptions, the typical ones are the RSA assumption, the strong RSA (SRSA, for short) assumption [6, 19] and the DL assumption. Although they have different representation, an essential property of these assumptions seems to be the same in a sense that it is assumed hard to solve some specific equation over a group. In fact, the RSA assumption states that it is hard to solve an equation for x of the form $x^e = y$ over the RSA group $\mathbb{Z}_N^\times$, where N is an RSA modulus, i.e. $N = PQ$, a product of two distinct odd primes. The DL assumption asserts that an equation $y = g^x$ cannot be solved for x in PPT over a multiplicative group $\mathbb{Z}_P^\times$ of a finite field $\mathbb{F}_P$. By this observation, the questions arise: Can we construct a unified framework to treat several cryptographic assumptions by using the essential property? Moreover, can we prove the security of generic cryptographic schemes by using such a framework? If the questions are resolved affirmatively, the security of each cryptographic scheme handled by the framework can be proven in a unified manner with respect to a specific cryptographic assumption.

As a unified framework, the notion of pseudo-free group was proposed by Hohenberger [29] in 2003, and was formalized by Rivest [40]. The pseudo-freeness is defined for computational groups. A group is computational if computational operations such as the group law and sampling elements can be efficiently done. Intuitively, a family $\{\mathbb{G}_N\}$ of computational groups is pseudo-free if $\mathbb{G}_N$ is indistinguishable from free groups. Several cryptographic assumptions including the RSA assumption, the DL assumption and the SRSA assumption hold on pseudo-

free groups [40]. This fact indicates that the notion of pseudo-free group could be a unified framework which can treat several cryptographic assumptions.

The indistinguishability of pseudo-free groups $\mathbb{G}_N$ from free groups is described by using equations over a free group. Namely, distinguishing is to find a witness pair (λ, ψ) such that the equation λ has no solution over the free group, but it has a solution ψ over $\mathbb{G}_N$. Such a pair (λ, ψ) witnesses that $\mathbb{G}_N$ is not a free group, since λ should have no solution if $\mathbb{G}_N$ is a free group.

A crucial question was left by Rivest [40] to verify that the pseudo-free group is not a vacuous notion. Namely, it was necessary for us to find a concrete example of pseudo-free groups. In 2005, this question was affirmatively answered by Micciancio [34]. It was shown that the RSA group $\mathbb{Z}_N^\times$ is pseudo-free under the SRSA assumption when the RSA modulus N is a product of two distinct safe primes, a prime P such that $(P-1)/2$ is also prime. This implies that the pseudo-freeness is equivalent to the SRSA assumption over $\mathbb{Z}_N^\times$ of such moduli. Therefore, the pseudo-freeness seems to be a feasible cryptographic assumption. Jhanwar and Barua [31] showed the pseudo-freeness of $\mathbb{Z}_N^\times$ under the SRSA assumption, but with a slightly different condition from Micciancio's one. For an example other than $\mathbb{Z}_N^\times$, Anokhin [5] constructed a family of groups that is shown to be pseudo-free under the integer factoring assumption.

Variants of pseudo-free groups have also been studied in order to cover an assumption that is not known to be captured by the (original) pseudo-free groups. Rivest proposed the variants called weak pseudo-free groups and pseudo-free groups with generalized exponential expressions [40]. Hirano and Tanaka [26] formalized these two notions and showed that several standard cryptographic assumptions hold on the variants as well as on the original pseudo-free groups. Hasegawa, Isobe, Shizuya and Tashiro [25] investigated the relationships among pseudo-free groups and the two variants. They showed that the pseudo-freeness with generalized exponential expressions is equivalent to the original pseudo-freeness, and the pseudo-freeness implies the weak pseudo-freeness. They also showed in [25] that the computational Diffie-Hellman assumption [40] holds on pseudo-free groups in a slightly varied form.

As mentioned above, the notion of pseudo-free group is not a vacuous theory and could be a unified framework. We next focus on the question whether or not a cryptographic scheme can be directly constructed from the pseudo-freeness. If such schemes can be obtained, one may be able to yield a new cryptographic

scheme whose security is proven from a specific cryptographic assumption, only by showing the pseudo-freeness from the specified assumption. In the ordinary security proofs, *adaptive* adversaries which are allowed to obtain auxiliary information are employed. For example, a chosen ciphertext attacker [38] against a public key encryption can adaptively obtain a plaintext for a ciphertext which is requested. However, such adversaries are not considered in the definition of the pseudo-free group. Therefore, if we can extend the notion of the pseudo-freeness so that it enables us to handle adaptive adversaries, we may construct cryptographic schemes directly from pseudo-free groups.

Concerning this issue, Catalano, Fiore and Warinschi [13] introduced the notion of adaptive pseudo-free groups in 2011. Their adaptive pseudo-freeness means that any PPT adversary cannot find a new witness pair (λ^*, ψ^*) as in the Rivest's static setting even if the adversary is allowed to adaptively receive a solution of an equation query. Here, the "adaptive" behavior of the adversary is restricted in a way that the adversary is not allowed to arbitrarily choose equations to be solved. Instead, equations are chosen according to some specified parametric distribution ϱ over the set of possible queries. The adversary queries a parameter M for determining the distribution $\varrho(M)$. Then, it receives a witness (λ, ψ) of the equation λ chosen according to the distribution $\varrho(M)$ and the corresponding solution ψ. Note that the adaptive pseudo-freeness includes the original "static" pseudo-freeness as a special case where the adversary makes no query.

In [13], they gave a class of specific parametric distributions with which $\mathbb{Z}_N^\times$ is adaptive pseudo-free under the SRSA assumption. They also showed a generic construction of secure signature schemes from the notion of adaptive pseudo-free groups as the first direct cryptographic application. Intuitively, a witness pair (λ, ψ) of an equation λ chosen according to a distribution $\varrho(M)$ and its solution ψ is a signature on a message M. The adaptive pseudo-freeness implies that their signatures are strongly existentially unforgeable against the chosen message attack (sEUF-CMA, for short). More specifically, in the security proof of their signature schemes, the adversary against the signature schemes is naturally regarded as an adversary breaking the adaptive pseudo-freeness. In particular, they mentioned that the proof of the adaptive pseudo-freeness of $\mathbb{Z}_N^\times$ is an abstraction for the security proofs of the SRSA-based signature schemes [11, 14, 18, 22, 27, 49], namely the signatures whose security is proven from the

SRSA assumption.

1.2 Summary of Results

As mentioned in the previous section, Catalano, Fiore and Warinschi [13] unified several SRSA-based signature schemes in a sense that these are directly obtained from the adaptive pseudo-freeness of $\mathbb{Z}_N^\times$. This means that the notion of adaptive pseudo-free groups is a candidate of the desirable framework from which one can build a generic construction of cryptographic schemes. However, the construction of signatures given in [13] is the only example of cryptographic schemes from adaptive pseudo-free groups. In this thesis, we discuss flexibility and applicability of the notion of the adaptive pseudo-free groups. Especially, we investigate the adaptive pseudo-freeness of the RSA group $\mathbb{Z}_N^\times$ with respect to several cryptographic assumptions.

First, we focus on the flexibility. The adaptive pseudo-freeness is defined with respect to a parametric distribution that determines the form of equations supplied to adaptive pseudo-free adversaries. In [13], they proposed the class of appropriate parametric distributions ϱ^{CFW} to pick equations that matches ones used in the SRSA-based signatures including [11, 14, 18, 22, 27, 49]. Therefore, if one wants to construct a cryptographic scheme from the adaptive pseudo-free group, he should set up a parametric distribution corresponding to the cryptographic scheme. From this viewpoint, the adaptive pseudo-freeness with respect to a parametric distribution which does not force the form of equations is expected to produce generic cryptographic schemes. Such an adaptive pseudo-free group is already referred to as the strongly-adaptive pseudo-free group in [13]. However, it is not known whether or not the strong adaptive pseudo-freeness is feasible. In particular, it remains open whether or not the RSA group $\mathbb{Z}_N^\times$ is strongly-adaptive pseudo-free. For this question, we give the following negative circumstantial evidence.

Theorem 3.4 (Chapter 3)
The strong adaptive pseudo-freeness of the RSA group $\mathbb{Z}_N^\times$ cannot be proven from the SRSA assumption via algebraic reductions, as long as the SRSA assumption holds.

In Theorem 3.4, a reduction algorithm $\mathcal{R}$ is restricted to being algebraic. The notion of algebraic reductions is introduced by Paillier and Vergnaud [37].

Informally, an algorithm is said to be *algebraic* with respect to a group $\mathbb{G}$ if the algorithm performs only group operations for elements in $\mathbb{G}$ and its execution can be easily traced. We note that employing algebraic algorithms is not of exceedingly restricted setting, because most reductions concerning the pseudo-free group [13, 31, 34], and ordinary security proofs (e.g. [11, 14, 37]) are performed on algebraic algorithms. This notion is employed in order to give impossibility results for constructing security proofs of several cryptographic schemes [1, 2, 21, 24, 43], and investigate relationships among cryptographic assumptions [10, 45].

Theorem 3.4 means that the strong adaptive pseudo-freeness of the RSA group $\mathbb{Z}_N^\times$ cannot be shown under the SRSA assumption, by employing only current proof techniques which are frequently used in ordinary security proofs. Since the SRSA assumption is one of the strongest assumption, this implies that the strong adaptive pseudo-freeness for the RSA group $\mathbb{Z}_N^\times$ may be far from feasibility.

By Theorem 3.4, restricting the form of equations by using some parametric distribution is reasonable to make $\mathbb{Z}_N^\times$ be adaptive pseudo-free. Thus we next consider the applicability of the existing parametric distributions. Namely we discuss the question whether or not one can prove the adaptive pseudo-freeness of the RSA group $\mathbb{Z}_N^\times$ with respect to a parametric distribution belonging to the class of Catalano-Fiore-Warinschi [13] from some assumption other than the SRSA assumption. If this question is positively resolved, a signature scheme whose security is guaranteed by the employed assumption could be constructed from the adaptive pseudo-freeness by using its parametric distribution. For this question, we again give a negative circumstantial evidence.

Theorem 3.9 (Chapter 3)
The adaptive pseudo-freeness of $\mathbb{Z}_N^\times$ with respect to any parametric distribution belonging to the class of Catalano-Fiore-Warinschi cannot be proven from the RSA assumption via algebraic reductions, as long as the RSA assumption holds.

This result means that a new parametric distribution is required to prove the adaptive pseudo-freeness of $\mathbb{Z}_N^\times$ from the RSA assumption.

As an application of Theorem 3.9, we show that the sEUF-CMA security of the SRSA-based signature schemes proposed by [11, 14, 18, 22, 27, 49] may not be proven from the RSA assumption. For the purpose of obtaining such an impossibility result, we prove that the sEUF-CMA security of signatures yielded from the generic construction given by Catalano-Fiore-Warinschi implies the adap-

tive pseudo-freeness of $\mathbb{Z}_N^\times$. Recall that the SRSA-based signatures given in [11, 14, 18, 22, 27, 49] can be obtained from their construction. If the sEUF-CMA security of each of these SRSA-based signatures is proven from the RSA assumption via algebraic reductions, the adaptive pseudo-freeness of $\mathbb{Z}_N^\times$ is also proven from the RSA assumption via algebraic reductions. It follows from Theorem 3.9 that the RSA assumption does not hold. Thus, one can show that the sEUF-CMA security of these SRSA-based signatures cannot be proven from the RSA assumption via algebraic reductions, as long as the RSA assumption holds. This indicates that the adaptive pseudo-free group is useful to discuss whether or not the security of a cryptographic scheme is provable from a specific assumption.

By Theorem 3.9, the parametric distributions of Catalano-Fiore-Warinschi cannot be applied to show the adaptive pseudo-freeness of $\mathbb{Z}_N^\times$ under the RSA assumption. Therefore, in order to show the adaptive pseudo-freeness of $\mathbb{Z}_N^\times$ under the RSA assumption, we need another parametric distribution. We finally explore such a parametric distribution with which the adaptive pseudo-freeness of the RSA group $\mathbb{Z}_N^\times$ can be proven from the RSA assumption. We give the following affirmatively result for this question.

Theorem 4.9 (Chapter 4)
There exists a family $\{\varrho^{K,c}\}$ of parametric distributions so that the adaptive pseudo-freeness of the RSA group $\mathbb{Z}_N^\times$ with respect to the family $\{\varrho^{K,c}\}$ can be proven from the RSA assumption.

Our idea is to utilize the technique presented by Schäge and Schwenk [41]. In [41], they converted an SRSA-based signature scheme into an RSA-based signature scheme with involving the hash function introduced by Hohenberger and Waters [30]. Their hash function is employed to construct our parametric distributions.

Theorem 4.9 means that the adaptive pseudo-freeness of $\mathbb{Z}_N^\times$ is proven from the RSA assumption by using our parametric distributions $\varrho^{K,c}$. Therefore, RSA-based signature schemes could be constructed from the adaptive pseudo-freeness by using our parametric distributions $\varrho^{K,c}$. It is also expected that the adaptive pseudo-freeness of $\mathbb{Z}_N^\times$ is proven with respect to an appropriate parametric distribution corresponding to the applied cryptographic assumption.

1.3 Organization

The rest of this thesis is organized as follows. We introduce notions and notations that are used throughout the thesis in Chapter 2. In Chapter 3, we give several impossibility results on the adaptive pseudo-freeness of the RSA group $\mathbb{Z}_N^\times$. More specifically, in Section 3.1, we show that the strong adaptive pseudo-freeness of $\mathbb{Z}_N^\times$ cannot be proven from the SRSA assumption via algebraic reductions, as long as the SRSA assumption holds. In Section 3.2, we show that the adaptive pseudo-freeness of $\mathbb{Z}_N^\times$ with respect to any parametric distribution belonging to the class of Catalano-Fiore-Warinschi cannot be proven from the RSA assumption via algebraic reductions, as long as the RSA assumption holds. Moreover, we also give the impossibility result on several SRSA-based signatures. Namely, it is shown that SRSA-based signature schemes that can be obtained from the generic construction given in [13] may not be proven from the RSA assumption. In Chapter 4, we describe our parametric distributions $\varrho^{K,c}$, and prove that the RSA group $\mathbb{Z}_N^\times$ is adaptive pseudo-free with the parametric distribution $\varrho^{K,c}$ under the RSA assumption. Conclusion is given in Chapter 5.

Chapter 2

Preliminaries

In this chapter, we describe notions and notations that are used through this thesis.

2.1 Mathematical Notations

A prime P is *safe* if $P = 2P' + 1$ for some prime P'. Let $\mathbb{N}^{\mathrm{safe}}_{\mathsf{RSA}}$ be the set of all RSA composites $N = PQ$ such that P and Q are distinct safe primes, and let $\mathbb{N}^{\mathrm{safe}}_{\mathsf{RSA}(k)}$ be the set of all $N = PQ \in \mathbb{N}^{\mathrm{safe}}_{\mathsf{RSA}}$ such that P and Q are distinct primes of binary length $k/2$, and hence the binary length of N is k. We assume that there are infinitely many safe primes. Although it is open whether or not this assumption holds, this assumption is widely believed to hold [3]. For any $N \in \mathbb{N}$, we use $\mathbb{Z}_N$ and $\mathbb{Z}_N^\times$ to denote the residue ring $\mathbb{Z}/N\mathbb{Z}$ and its group of units, respectively. QR_N designates the group of quadratic residues mod N. Let $\mathbb{P}$ denote the set of all primes, especially $\mathbb{P}_{<n}$ indicates a subset of all primes P such that $P < n$.

For any integers $a \leq b$, let $[a, b]$ ((a, b), resp.) be the set of all integers n such that $a \leq n \leq b$ ($a < n < b$, resp.). We denote by $x \in_{\mathrm{R}} D$ that the element x is chosen at random from the finite set D according to some specific probabilistic distribution. In particular, we write $x \in_{\mathrm{U}} D$ when a uniform distribution on D is designated. By $x := y$, we mean that x is defined or substituted by y. For any algorithm $\mathcal{A}$, $y \leftarrow \mathcal{A}(x)$ indicates that the algorithm $\mathcal{A}$ outputs y on input x. Note that when $\mathcal{A}$ is a probabilistic algorithm, y is distributed according to the internal coin flips of $\mathcal{A}$.

A function $\nu(k)$ is *negligible* if for any polynomial p, there exists a constant

k_0 such that $\nu(k) < 1/p(k)$ for any $k \geq k_0$. We denote by $\mathrm{negl}(k)$ any negligible function in k. A function $\nu(k)$ that is not negligible is called *nonnegligible*. $\nu(k)$ is *overwhelming* if for any polynomial p, there exists a constant k_0 such that $\nu(k) > 1 - 1/p(k)$ for any $k \geq k_0$. Let X and Y be probability distributions over a finite set D. The statistical distance between X and Y is defined by $(1/2) \cdot \sum_{a \in D} |X(a) - Y(a)|$. We write $\mathsf{Supp}(X)$ to denote the *support* of X. Namely, $\mathsf{Supp}(X)$ is the set of all elements $x \in D$ such that the probability that x is selected according to X is strictly greater than 0. Let $\{X_N\}_{N \in \mathbb{N}}$ and $\{Y_N\}_{N \in \mathbb{N}}$ be two ensembles of probability distributions, where for each $N \in \mathbb{N}$, X_N and Y_N are defined over a finite set D_N. $\{X_N\}$ is said to be *statistically close* to $\{Y_N\}$ if the statistical distance between $\{X_N\}$ and $\{Y_N\}$ is negligible in the binary length of N. In particular, $\{X_N\}$ is *almost uniform* if $\{X_N\}$ is statistically close to the ensemble of the uniform distributions over the domain D_N. We say that $\{X_N\}$ is *polynomial-time samplable* if there exists a probabilistic algorithm Samp such that on input N, Samp outputs an element $x \in D_N$ that is distributed according to X_N, and it runs in polynomial-time in the binary length of N.

2.2 Pseudo-Free Groups

In this section, we describe notions and notations necessary for us to discuss pseudo-free groups.

2.2.1 Computational Groups

Let $\{\mathbb{G}_N\}_{N \in \mathcal{N}}$ be a family of finite groups indexed by an index set $\mathcal{N} = \cup_{k \geq 0} \mathcal{N}(k)$, where $\mathcal{N}(k)$ denotes a set of all indices of polynomial length in k. We assume that each group index $N \in \mathcal{N}(k)$ and each element of $\mathbb{G}_N$ (with $N \in \mathcal{N}(k)$) are expressed as a word of polynomial length in k, respectively. Then, $\{\mathbb{G}_N\}_{N \in \mathcal{N}}$ is said to be a *family of computational groups* [40, 34] if the following polynomial-time algorithms are provided:

Composition for any given elements $a, b \in \mathbb{G}_N$ and a group index $N \in \mathcal{N}$, compute $ab \in \mathbb{G}_N$.

Inversion for any given element $a \in \mathbb{G}_N$ and a group index $N \in \mathcal{N}$, compute $a^{-1} \in \mathbb{G}_N$.

Identity for any given group index $N \in \mathcal{N}$, compute the identity element $1 \in \mathbb{G}_N$.

Membership for any given word x and a group index $N \in \mathcal{N}$, determine whether or not $x \in \mathbb{G}_N$.

Identification for any given words x, y and a group index $N \in \mathcal{N}$, determine whether or not $x = y$ in $\mathbb{G}_N$.

Sampling for any given group index $N \in \mathcal{N}$, choose a single element $g \in \mathbb{G}_N$ at random, where the sampling is not necessarily uniform over $\mathbb{G}_N$.

Throughout this thesis, we assume that $\{\mathbb{G}_N\}_{N\in\mathcal{N}}$ is abelian, that is $\mathbb{G}_N$ is abelian for any $N \in \mathcal{N}$.

2.2.2 Free Abelian Groups

Let $A = \{a_1, a_2, \ldots, a_m\}$ be a nonempty set of distinct m symbols, which are the *generators* of a free group. The *identity* of the free group is the empty string $\mathfrak{e}$. For each $a \in A$, a^{-1} denotes the *inverse* of the symbol a. Note that for any $a \in A$, the inverse a^{-1} does not belong to the set A. Let $A^{-1} = \left\{a_1^{-1}, a_2^{-1}, \ldots, a_m^{-1}\right\}$. Then, $A \cup A^{-1}$ is said to be the set of symbols for the free group generated by the set A. We denote by $\mathcal{F}(A)$ the free abelian group generated by the set A. Since $\mathcal{F}(A)$ is now abelian, any element of $\mathcal{F}(A)$ is uniquely expressed by a word of the form $\prod_{i=1}^{m} a_i^{s_i}$ with some exponents $s_1, s_2, \ldots, s_m \in \mathbb{Z}$.

2.2.3 Equations over Groups

We focus only on univariate equations over a free abelian group as in [13]. Let A be a set of m symbols, and let x denote a *variable*. An *equation in x with symbols in A* is a pair $\lambda = (w_1, w_2)$, where w_1 is a word of the form x^E with some exponent $E \in \mathbb{N}$, and w_2 is a word over A of finite length. Since $\mathcal{F}(A)$ is abelian, we may assume that w_2 is expressed in a way that $w_2 = \prod_{i=1}^{m} a_i^{s_i}$ with some exponents $s_1, s_2, \ldots, s_m \in \mathbb{Z}$. Then we write the equation $\lambda = (w_1, w_2)$ by $x^E = \prod_{i=1}^{m} a_i^{s_i}$. We express the equation $\lambda : x^E = \prod_{i=1}^{m} a_i^{s_i}$ with the tuple $(E, \boldsymbol{s})$ of exponents, where $\boldsymbol{s} = (s_1, s_2, \ldots, s_m)$. Equations that have solutions in $\mathcal{F}(A)$ are *trivial*, others are *nontrivial*. The triviality of an equation $x^E = \prod_{i=1}^{m} a_i^{s_i}$ can be easily verified by the following lemma.

Lemma 2.1 ([40])
An equation $x^E = \prod_{i=1}^{m} a_i^{s_i}$ *is trivial over* $\mathcal{F}(A)$ *if and only if* $E \mid s_i$ *for any* $1 \leq i \leq m$.

Let $\mathbb{G}$ be any finite abelian group, and let $\alpha : A \to \mathbb{G}$ be an *assignment* map that interprets each symbol $a \in A$ to a group element $\alpha(a) \in \mathbb{G}$. We write λ_α for the equation $\lambda : x^E = \prod_{i=1}^{m} a_i^{s_i}$ interpreted over $\mathbb{G}$ via α, namely λ_α is the equation $x^E = \prod_{i=1}^{m} \alpha(a_i)^{s_i}$ over $\mathbb{G}$. $\psi \in \mathbb{G}$ is a *solution* for λ_α if $\psi^E = \prod_{i=1}^{m} \alpha(a_i)^{s_i}$ holds over $\mathbb{G}$.

2.2.4 Pseudo-Freeness

We describe the notion of pseudo-free groups formalized by Rivest [40]. Intuitively, the pseudo-free group is a computational group family $\{\mathbb{G}_N\}$ that is indistinguishable from free groups. This distinguishability is formalized by using a *witness pair* (λ^*, ψ^*) such that λ^* is a nontrivial equation and ψ^* is a solution over $\mathbb{G}_N$ of the interpreted equation λ^*_α. Such a pair (λ^*, ψ^*) witnesses that $\mathbb{G}_N$ is not a free group, because λ^*_α should have no solution if $\mathbb{G}_N$ is indeed a free group.

Definition 2.2 (Pseudo-Free Group [40])
Let k *be a security parameter, and let* $m = m(k)$ *be a polynomial in* k. *A computational group family* $\{\mathbb{G}_N\}$ *is* (static) pseudo-free *if there exists no PPT adversary (algorithm)* $\mathcal{A}$ *such that for any set* A *of* m *symbols,* $\mathcal{A}$ *outputs a witness pair* (λ^*, ψ^*) *with nonnegligible probability in* k *on an input pair* (N, α) *of a group index* N *and an assignment map* $\alpha : A \to \mathbb{G}_N$, *where the probability is taken over the random choice of* $N \in \mathcal{N}(k)$ *and* $\alpha(a) \in \mathbb{G}_N$ *for each* $a \in A$ *and the coin flips of* $\mathcal{A}$.

We also define a variant of pseudo-freeness in a sense that an equation that can be output from $\mathcal{A}$ is restricted. We suppose that for each k and m, a class $\mathcal{E}_{k,m}$ of pairs (λ, r), where $\lambda = (E, (s_1, \ldots, s_m))$ is an equation and r is any string, is designated, and assume also that the membership for the class $\mathcal{E}_{k,m}$ can be determined in polynomial-time in k.

Definition 2.3 (Pseudo-Free Group over a Family $\mathcal{E} = \{\mathcal{E}_{k,m}\}_{k,m}$)
Let k *be a security parameter, and let* $m = m(k)$ *be a polynomial in* k. *For each* k *and* m, *let* $\mathcal{E}_{k,m}$ *be a set of pairs* (λ, r). *Then,* $\{\mathbb{G}_N\}$ *is* static pseudo-free over the family $\mathcal{E} = \{\mathcal{E}_{k,m}\}_{k,m}$ *if there exists no PPT adversary* $\mathcal{A}$ *such that for any*

set A of m symbols, $\mathcal{A}$ outputs a tuple $((\lambda^*, r^*), \psi^*)$ satisfying that $(\lambda^*, r^*) \in \mathcal{E}_{k,m}$ and (λ^*, ψ^*) is a witness pair, with nonnegligible probability in k on an input pair (N, α) of a group index N and an assignment map $\alpha : A \to \mathbb{G}_N$, where the probability is taken over the random choice of $N \in \mathcal{N}(k)$ and $\alpha(a) \in \mathbb{G}_N$ for each $a \in A$ and the coin flips of $\mathcal{A}$.

2.3 Adaptive Pseudo-Free Groups

Catalano, Fiore and Warinschi [13] introduced the notion of adaptive pseudo-freeness as a generalization of the Rivest's (static) pseudo-freeness. Intuitively, the adaptive pseudo-freeness means that any PPT adversary cannot output a new witness pair (λ^*, ψ^*) as in the static case in nonnegligible probability, even if he is allowed to adaptively receive a witness pair polynomially many times. The adaptive pseudo-freeness is defined by the adaptive pseudo-free game between two algorithms which are called the challenger and the adversary. The challenger is an algorithm that gives the adversary an instance and replies a witness pair when the adversary queries. The following two points are noted in [13].

Adaptive Behavior. The first is that, although the adversary is allowed the adaptive oracle queries, the "adaptive" behavior is somewhat restricted in the following way. We suppose that for each k and m, a class $\mathcal{E}_{k,m}$ of pairs (λ, r) is designated, and we provide a family $\varrho_{k,m} = \{\varrho_{k,m}(M)\}$ of probabilistic distributions $\varrho_{k,m}(M)$ over $\mathcal{E}_{k,m}$. The adversary queries an equation by sending a parameter M to the challenger. The challenger determines the probabilistic distribution $\varrho_{k,m}(M)$ by using the family $\varrho_{k,m}$ and the parameter M. Then, the challenger chooses a pair (λ, r) according to the distribution $\varrho_{k,m}(M)$, and returns the pair (λ, r) with the corresponding solution ψ for λ_α to the adversary. Thus, the adversary does not have a perfect control over his queries.

Nontriviality in the Adaptive Setting. The second is the nontriviality of the equation that the adversary outputs. In the original static setting by Rivest [40], the nontriviality merely means that the equation has no solution over $\mathcal{F}(A)$. However, the adaptive setting requires a more sophisticated condition on the nontriviality. Intuitively, the nontriviality of the equation λ^* (output by the adversary) means that λ^* is independent of the equations $\Lambda = \{\lambda^{(1)}, \lambda^{(2)}, \ldots, \lambda^{(q)}\}$ queried to the challenger, namely λ^* cannot be efficiently deduced from Λ.

In [13], in order to formalize such an independence, they first considered a general deducibility modulo equations as follows. Let $\mathcal{F}$ be a free abelian group, and Λ be a binary relation over $\mathcal{F}$. Then, $\overline{\Lambda}$ denotes the smallest equivalence relation over $\mathcal{F}$ such that $\Lambda \subseteq \overline{\Lambda}$ and for all $e \in \mathbb{N}$ and $w_1, w_2 \in \mathcal{F}$, $(w_1^e, w_2^e) \in \overline{\Lambda}$ implies $(w_1, w_2) \in \overline{\Lambda}$. This condition reflects the simple fact over a computational group $\mathbb{G}$ that for any element $w_1, w_2 \in \mathbb{G}$ and any integer e that is coprime to the order $\mathrm{ord}(\mathbb{G})$ of $\mathbb{G}$, if $w_1^e = w_2^e$ over the group $\mathbb{G}$, then $w_1 = w_2$ follows.

By using the smallest congruence $\overline{\Lambda}$, in [13], they formalized the nontriviality with respect to the query set Λ as follows. Suppose that an adversary now attempts to find a witness pair (λ^*, ψ^*). Then, the adversary is supposed to receive q equations together with their corresponding solutions over $\mathbb{G}_N$. Namely, for each $1 \leq t \leq q$, he receives an equation $\lambda^{(t)} : x^{E_t} = \prod_{i=1}^{m} a_i^{s_{t,i}}$ together with the solution $\psi_t \in \mathbb{G}_N$ for the interpreted equation $\lambda_\alpha^{(t)}$. We set $\Lambda = \left\{\lambda^{(1)}, \ldots, \lambda^{(q)}\right\}$ and $\Psi = \{\psi_1, \ldots, \psi_q\}$. By regarding $\psi_1, \ldots \psi_q$ as new distinct symbols not contained in A, we define a binary relation $\overline{\Lambda}$ over the free abelian group $\mathcal{F} = \mathcal{F}(A \cup \Psi)$ by $\overline{\Lambda} = \left\{(\psi_t^{E_t}, \prod_{i=1}^{m} a_i^{s_{t,i}})\right\}_{t=1}^{q}$. Then, an equation λ^* (output by the adversary) is *trivial with respect to the queried equation set* Λ if λ^* has a solution over the residue group $\mathcal{F}(A \cup \Psi)/\overline{\Lambda}$. We note that when $\Lambda = \emptyset$, the triviality of equations is exactly equivalent to that in the static case, thus the triviality is verified by Lemma 2.1. For the triviality, the following proposition holds. Note that this proposition holds when $\mathcal{F}(A)$ is abelian.

Proposition 2.4 ([13])
An equation $\lambda^* : x^{E^*} = \prod_{i=1}^{m} a_i^{s_i^*}$ *is trivial with respect to an equation set* $\Lambda = \left\{\lambda^{(t)} : x^{E_t} = \prod_{i=1}^{m} a_i^{s_{t,i}}\right\}_{t=1}^{q}$ *if and only if there exist matrices*

$$\boldsymbol{U} = \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_q \end{bmatrix} \in \mathbb{Z}^q \text{ and } \boldsymbol{V} = \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_m \end{bmatrix} \in \mathbb{Z}^m$$

such that

$$E^* \left(\begin{bmatrix} s_{1,1} & s_{2,1} & \ldots & s_{q,1} \\ s_{1,2} & s_{2,2} & \ldots & s_{q,2} \\ \vdots & \vdots & & \vdots \\ s_{1,m} & s_{2,m} & \ldots & s_{q,m} \end{bmatrix} \begin{bmatrix} \frac{1}{E_1} & & & \\ & \frac{1}{E_2} & & \\ & & \ddots & \\ 0 & & & \frac{1}{E_q} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_q \end{bmatrix} + \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_m \end{bmatrix} \right) = \begin{bmatrix} s_1^* \\ s_2^* \\ \vdots \\ s_m^* \end{bmatrix}.$$

Proposition 2.4 implies the following lemma.

Lemma 2.5
If an equation $\lambda^* : x^{E^*} = \prod_{i=1}^{m} a_i^{s_i^*}$ *is nontrivial with respect to a set* $\Lambda = \left\{\lambda^{(t)} : x^{E_t} = \prod_{i=1}^{m} a_i^{s_{t,i}}\right\}_{t=1}^{q}$ *of equations, then for any index* $t \in [1, q]$*, we have*

$$(E^*, s_1^*, s_2^*, \ldots, s_m^*) \neq (E_t, s_{t,1}, s_{t,2}, \ldots, s_{t,m}) .$$

Proof. Assume that there exists an index $t^* \in [1, q]$ such that

$$(E^*, s_1^*, s_2^*, \ldots, s_m^*) = (E_{t^*}, s_{t^*,1}, s_{t^*,2}, \ldots, s_{t^*,m}) . \tag{2.1}$$

Then, we set $\boldsymbol{U} = \begin{bmatrix} U_1 & U_2 & \ldots & U_q \end{bmatrix}^T \in \mathbb{Z}^q$ as $U_{t^*} = 1$ and $U_t = 0$ for any $t \in [1, q] \setminus \{t^*\}$, and set $\boldsymbol{V} = \begin{bmatrix} V_1 & V_2 & \ldots & V_m \end{bmatrix}^T \in \mathbb{Z}^m$ as $V_i = 0$ for any $i \in [1, m]$, respectively. By Eq. (2.1), we have

$$E^* \left(\begin{bmatrix} s_{1,1} & s_{2,1} & \ldots & s_{q,1} \\ s_{1,2} & s_{2,2} & \ldots & s_{q,2} \\ \vdots & \vdots & & \vdots \\ s_{1,m} & s_{2,m} & \ldots & s_{q,m} \end{bmatrix} \begin{bmatrix} \frac{1}{E_1} & & & \\ & \frac{1}{E_2} & & 0 \\ & & \ddots & \\ 0 & & & \frac{1}{E_q} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_q \end{bmatrix} + \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_m \end{bmatrix} \right)$$

$$= E^* \begin{bmatrix} s_{1,1} & s_{2,1} & \ldots & s_{q,1} \\ s_{1,2} & s_{2,2} & \ldots & s_{q,2} \\ \vdots & \vdots & & \vdots \\ s_{1,m} & s_{2,m} & \ldots & s_{q,m} \end{bmatrix} \begin{bmatrix} \frac{1}{E_1} & & & \\ & \frac{1}{E_2} & & 0 \\ & & \ddots & \\ 0 & & & \frac{1}{E_q} \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$

$$= E^* \begin{bmatrix} s_{1,1} & s_{2,1} & \ldots & s_{q,1} \\ s_{1,2} & s_{2,2} & \ldots & s_{q,2} \\ \vdots & \vdots & & \vdots \\ s_{1,m} & s_{2,m} & \ldots & s_{q,m} \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ \frac{1}{E_{t^*}} \\ \vdots \\ 0 \end{bmatrix} = E^* \begin{bmatrix} s_{t^*,1}/E_{t^*} \\ s_{t^*,2}/E_{t^*} \\ \vdots \\ s_{t^*,m}/E_{t^*} \end{bmatrix} = E^* \begin{bmatrix} s_1^*/E^* \\ s_2^*/E^* \\ \vdots \\ s_m^*/E^* \end{bmatrix} = \begin{bmatrix} s_1^* \\ s_2^* \\ \vdots \\ s_m^* \end{bmatrix} .$$

It follows from Proposition 2.4 that $\lambda^* : x^{E^*} = \prod_{i=1}^{m} a_i^{s_i^*}$ is trivial with respect to $\Lambda = \left\{\lambda^{(t)} : x^{E_t} = \prod_{i=1}^{m} a_i^{s_{t,i}}\right\}_{t=1}^{q}$. □

Adaptive Pseudo-Free Game. The adaptive pseudo-free (APF, for short) game is played by a challenger and an adversary $\mathcal{A}$. Let k be a security parameter, and let A be a set of $m = m(k)$ symbols. As before, we suppose that

for each k and m, a class $\mathcal{E}_{k,m}$ of pairs (λ, r) of an equation λ and an auxiliary string r is designated, and we provide a family $\varrho_{k,m} = \{\varrho_{k,m}(M)\}$ of probabilistic distributions $\varrho_{k,m}(M)$ over $\mathcal{E}_{k,m}$. Given k and A, the game proceeds as follows:

Setup. The challenger chooses a random group index $N \in_{\mathrm{U}} \mathcal{N}(k)$. Then, it specifies an assignment map $\alpha : A \to \mathbb{G}_N$ by independently choosing an element $\alpha(a) \in_{\mathrm{R}} \mathbb{G}_N$ at random according to the designated sampling algorithm for each $a \in A$. The adversary $\mathcal{A}$ is given the *game tuple* $(N, \alpha, \varrho_{k,m})$.

Equations queries. The adversary $\mathcal{A}$ is allowed to adaptively query to the challenger in the following manner: on t-th query, $\mathcal{A}$ chooses a parameter M_t for determining a distribution $\varrho_{k,m}(M_t)$, and hands it to the challenger. Then, the challenger chooses a pair $(\lambda^{(t)}, r_t) \in \mathcal{E}_{k,m}$ of an equation $\lambda^{(t)} = (E_t, \boldsymbol{s}_t)$ and an auxiliary string r_t according to the distribution $\varrho_{k,m}(M_t)$, and then returns the pair $(\lambda^{(t)}, r_t)$ and a solution $\psi_t \in \mathbb{G}_N$ of the interpreted equation $\lambda^{(t)}_\alpha : x^{E_t} = \prod_{i=1}^{m} \alpha(a_i)^{s_{t,i}}$ to $\mathcal{A}$.

Challenge. Eventually, the adversary $\mathcal{A}$ outputs a tuple $((\lambda^*, r^*), \psi^*)$ of an equation $\lambda^* = (E^*, \boldsymbol{s}^*)$ and an auxiliary string r^* with a solution ψ^* of the interpreted equation λ^*_α over $\mathbb{G}_N$. The challenger outputs 1 if the following conditions hold, or 0 otherwise:

(A) $(\lambda^*, r^*) \in \mathcal{E}_{k,m}$;

(B) λ^* is *nontrivial with respect to* $\Lambda = \left\{(\lambda^{(t)}, \psi_t)\right\}_t$, the set of queried equations and corresponding solutions appeared in Equations queries phase; and

(C) ψ^* is actually a solution of λ^*_α.

An adversary $\mathcal{A}$ is said to *win the APF game of the family* $\mathcal{G}$ *with respect to the parametric distribution* $\varrho = \{\varrho_{k,m}\}_{k,m}$ if the challenger outputs 1 in the game between the challenger and the adversary $\mathcal{A}$.

Definition 2.6 (Adaptive Pseudo-Free Groups with respect to ϱ [13])
Let k be a security parameter, let m be a polynomial in k, and let $\varrho = \{\varrho_{k,m}\}_{k,m}$ be a parametric distribution. A family $\mathcal{G} = \{\mathbb{G}_N\}_{N \in \mathcal{N}}$ of computational groups is adaptive pseudo-free with respect to ϱ, *if there exists no PPT adversary $\mathcal{A}$ such that for any set A of m symbols, $\mathcal{A}$ wins the APF game of the family $\mathcal{G}$ with respect to ϱ in nonnegligible probability in k, where the probability is taken over the random choices of the index $N \in_{\mathrm{U}} \mathcal{N}(k)$, $\alpha(a) \in_{\mathrm{R}} \mathbb{G}_N$ for each $a \in A$ and*

each pair $(\lambda^{(t)}, r_t) \in_{\mathrm{R}} \mathcal{E}_{k,m}$ *chosen in* Equations queries *phase, and the internal coin flips of* $\mathcal{A}$.

We note that the static pseudo-freeness over the family $\mathcal{E} = \{\mathcal{E}_{k,m}\}_{k,m}$ is exactly the same as the adaptive pseudo-freeness described here in which the adversary makes no adaptive query.

2.4 Signature Schemes

A *signature scheme* $\mathcal{S}$ with a message space $\mathcal{M}$ consists of the following three polynomial-time algorithm (KGen, Sign, Verify):

Key Generator KGen**.** KGen is a PPT algorithm such that on input 1^k, KGen generates a public key pk and a secret key sk.

Signing Algorithm Sign**.** Sign is a PPT algorithm such that on input (sk, pk, M) of a secret key sk, a public key pk and a message $M \in \mathcal{M}$, Sign issues a signature σ on the message M.

Verification Algorithm Verify**.** Verify is a deterministic polynomial-time algorithm such that on input (pk, M, σ) of a public key pk, a message $M \in \mathcal{M}$ and a signature σ, Verify outputs 1 if σ is a signature on the message M under the public key pk, or 0 otherwise.

We now describe security notions of signature schemes. The *existential forgery game under the chosen message attack (EF-CMA game, for short) for a signature scheme* $\mathcal{S} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Verify})$ is defined between a challenger and an adversary $\mathcal{A}$. Given a security parameter k, the game proceeds as follows:

Setup. The challenger generates a pair (pk, sk) of a public key and a secret key by executing $\mathsf{KGen}(1^k)$, and then submits the public key pk to the adversary $\mathcal{A}$.

Signing oracle. When $\mathcal{A}$ (adaptively) queries a message $M_t \in \mathcal{M}$, the challenger issues a signature $\sigma_t \leftarrow \mathsf{Sign}(sk, pk, M_t)$, and then hands σ_t to $\mathcal{A}$.

Challenge. When $\mathcal{A}$ outputs a pair (M^*, σ^*), the challenger outputs 1 if the following conditions hold, or 0 otherwise:

(I) $M^* \notin \{M_t\}_t$, the set of messages queried in Signing oracle phase; and

(II) $\mathsf{Verify}(pk, M^*, \sigma^*) = 1$.

An adversary $\mathcal{A}$ is said to *win the EF-CMA game* if the challenger outputs 1 in the EF-CMA game. Then, a signature scheme $\mathcal{S}$ is *existentially unforgeable against the chosen message attack (EUF-CMA, for short)* if there exists no PPT adversary $\mathcal{A}$ that wins the EF-CMA game with nonnegligible probability, where the probability is taken over the coin flips of KGen, Sign and $\mathcal{A}$. We refer an EF-CMA game in which the condition (I) of the Challenge phase is replaced with "(I') $(M^*, \sigma^*) \notin \{(M_t, \sigma_t)\}_t$, the set of pairs of a message m_t and a signature σ_t appeared in Signing oracle phase" to *a strongly EF-CMA (sEF-CMA, for short) game*. Moreover, a signature scheme $\mathcal{S}$ is *strongly EUF-CMA (sEUF-CMA, for short)* if $\mathcal{S}$ is EUF-CMA even when the condition (I) is replaced with (I'). An EF-CMA game in which an adversary $\mathcal{A}$ is not allowed to query in Signing oracle phase is called a *existential forgery game under the key only attack (EF-KOA game, for short)*. A signature $\mathcal{S}$ is *existentially unforgeable against the key only attack (EUF-KOA, for short)* if there exists no PPT adversary $\mathcal{A}$ that wins the EF-KOA game with nonnegligible probability, where the probability is taken as the same as the notion of EUF-CMA. Note that an EF-KOA challenger is not required to verify the condition (I), because $\mathcal{A}$ does not query in Signing oracle phase. The following proposition follows for the relationship among these security notions.

Proposition 2.7
Let $\mathcal{S}$ be a signature scheme. If $\mathcal{S}$ is sEUF-CMA, then it is also EUF-CMA. If $\mathcal{S}$ is EUF-CMA, then it is also EUF-KOA.

2.5 Hash Functions

A family $\mathcal{H} = \{H_i : D_i \to R_i\}_{i \in I}$ of functions is said to be *a family of hash functions* if the following algorithms (Gen, Samp, H) exist.

Parameter Generator Gen. Gen is a PPT algorithm such that on input 1^k, Gen generates a parameter $i \in I$ such that the length of i is greater than k, where for each parameter i, D_i denotes the domain of the function H_i and R_i denotes the range of the function H_i such that the number of the elements in D_i is strictly greater than that in R_i. We assume that any element in D_i and R_i is polynomial length in k for each i generated by $\mathsf{Gen}(1^k)$.

Sampling Algorithm Samp. Samp is a PPT algorithm such that on a parameter $i \in I$, Samp outputs $r \in D_i$ chosen according to some specific distribution on D_i.

Hash Algorithm H. H is a PPT algorithm such that on input (i, r) of a parameter $i \in I$ and an element $r \in D_i$, H outputs an element $H_i(r) \in R_i$.

We now describe security notions for hash functions. A pair $(r, r') \in D^2$ is said to be a *collision* of a hash function $H : D \to R$ if it holds that $r \neq r'$ but $H(r) = H(r')$. A hash function $\mathcal{H} = \{H_i : D_i \to R_i\}_{i \in I}$ is *collision-resistant* if there exists no PPT adversary $\mathcal{A}$ such that

$$\Pr\left[(r, r') \in D_i^2 \wedge r \neq r' \wedge H_i(r) = H_i(r') : \begin{array}{l} i \leftarrow \mathsf{Gen}(1^k) \\ (r, r') \leftarrow \mathcal{A}(1^k, i) \end{array}\right]$$

is nonnegligible in k, where the probability is taken over the coin flips of Gen and $\mathcal{A}$.

A hash function $\mathcal{H} = \{H_i : D_i \to R_i\}_{i \in I}$ is *division-intractable* if there exists no PPT adversary $\mathcal{A}$ such that

$$\Pr\left[\begin{array}{c} (r_1, r_2, \ldots, r_q, r^*) \in D_i^{q+1} \\ \wedge \forall t \in [1, q], r^* \neq r_t \\ \wedge H_i(r^*) \mid \prod_{t=1}^{q} H_i(r_t) \end{array} : \begin{array}{l} i \leftarrow \mathsf{Gen}(1^k) \\ (r_1, r_2, \ldots, r_q, r^*) \leftarrow \mathcal{A}(1^k, i) \end{array}\right]$$

is nonnegligible in k, where q is a polynomial in k, and the probability is taken over the coin flips of Gen and $\mathcal{A}$.

For a relationship between the collision-resistant property and the division-intractability, the following proposition holds.

Proposition 2.8 ([22])
Let $\mathcal{H}$ be a family of hash functions. $\mathcal{H}$ is collision-resistant provided that $\mathcal{H}$ is division-intractable.

2.6 Algebraic Algorithms

The concept of algebraic algorithms was introduced by Paillier and Vergnaud [37]. Intuitively, an algorithm $\mathcal{R}$ is *algebraic with respect to a computational group* $\mathbb{G}$ if $\mathcal{R}$ performs only the group operation for the elements in $\mathbb{G}$ and the execution of $\mathcal{R}$ can be easily traced. In particular, on any input elements $y_1, \ldots, y_n \in \mathbb{G}$, any

element $g \in \mathbb{G}$ produced in the execution of $\mathcal{R}$ belongs to the subgroup $\langle y_1, \ldots, y_n \rangle$ generated by the input elements, and moreover the expression $g = \prod_{i=1}^{n} y_i^{c_i}$ should be easily retrieved.

We follow the formal definition given in [43]. An algorithm $\mathcal{R}$ is *algebraic* for a computational group family $\{\mathbb{G}_N\}_{N \in \mathcal{N}}$, if the following algorithm $\mathsf{Extract}$ is provided. $\mathsf{Extract}$ receives any tuple $(N, y_1, \ldots, y_n, \mathsf{aux}, g, \omega)$ as input, where $N \in \mathcal{N}$ is a group index, $y_1, \ldots, y_n \in \mathbb{G}_N$ are elements that are given to $\mathcal{R}$ as input, aux is any word given to $\mathcal{R}$ as an auxiliary input, $g \in \mathbb{G}_N$ is a *target* group element and ω denotes a random coin used in $\mathcal{R}$. Then $\mathsf{Extract}$ finds a tuple $(c_1, \ldots, c_n)$ of exponents such that $g = \prod_{i=1}^{n} y_i^{c_i}$, provided that g is actually produced in the execution of $\mathcal{R}$ on the input tuple $(N, y_1, \ldots, y_n, \mathsf{aux})$ with the random coin ω. If there is no correct exponents $(c_1, \ldots, c_n)$, then $\mathsf{Extract}$ may output any word. $\mathsf{Extract}$ is required to run in polynomial-time in the running time of $\mathcal{R}$. In particular, if $\mathcal{R}$ runs in polynomial-time in the security parameter k, then $\mathsf{Extract}$ should run in polynomial-time in k.

We consider an algebraic algorithm $\mathcal{R}$ that has an access to an oracle $\mathcal{A}$. In the case where a target element $g \in \mathbb{G}$ is produced after $\mathcal{R}$ receives the answer for the t-th query, $\mathsf{Extract}$ correctly retrieves exponents $(c_1, \ldots, c_n)$ for the given target g if besides the input tuple $(N, y_1, \ldots, y_n, \mathsf{aux})$, $\mathsf{Extract}$ is also given all the t correct answers for from the first query through t-th query. Note that if the target element $g \in \mathbb{G}_N$ is produced before the first query, it is not required to provide any additional inputs to $\mathsf{Extract}$ as in [10].

2.7 Cryptographic Assumptions

Let $\ell = \ell(k)$ be a polynomial in k, and let φ denote Euler's function. A key generator $\mathsf{KGen}_{\mathsf{RSA}}$ outputs a pair $(N, e) \in \mathbb{N}^{\mathrm{safe}}_{\mathsf{RSA}(\ell)} \times \mathbb{Z}^{\times}_{\varphi(N)}$ for each input 1^k.

Definition 2.9 (RSA Assumption)
An adversary $\mathcal{R}$ is said to break RSA *if $\mathcal{R}$ outputs an element z such that $z^e \equiv y \pmod{N}$ on a given* RSA instance (N, e, y) *of an RSA public key $(N, e) \leftarrow \mathsf{KGen}_{\mathsf{RSA}}(1^k)$ and an element $y \in \mathrm{QR}_N$. The* RSA assumption holds *if there*

exists no PPT adversary $\mathcal{R}$ *such that*

$$\Pr\left[\begin{array}{ll} & (N,e) \leftarrow \mathsf{KGen}_{\mathsf{RSA}}(1^k), \\ z^e \equiv y \pmod N : & y \in_{\mathrm{U}} \mathrm{QR}_N, \\ & z \leftarrow \mathcal{R}(N,e,y) \end{array}\right]$$

is nonnegligible in k*, where the probability is taken over the coin flips of* $\mathsf{KGen}_{\mathsf{RSA}}$ *and* $\mathcal{R}$*, and the uniform random choice* y *from* QR_N*.*

An RSA modulus generator $\mathsf{KGen}_{\mathsf{SRSA}}$ outputs an RSA modulus $N \in \mathbb{N}^{\mathrm{safe}}_{\mathsf{RSA}(\ell)}$ for each input 1^k.

Definition 2.10 (Strong RSA Assumption)
An adversary $\mathcal{R}$ *is said to* break strong RSA (SRSA, for short) *if* $\mathcal{R}$ *outputs a pair* (z,e) *such that* $e > 1$ *and* $z^e \equiv y \pmod N$ *on a given* SRSA instance (N,y) *of an RSA modulus* $N \leftarrow \mathsf{KGen}_{\mathsf{SRSA}}(1^k)$ *and an element* $y \in \mathrm{QR}_N$*. The* SRSA assumption holds *if there exists no PPT adversary* $\mathcal{R}$ *such that*

$$\Pr\left[\begin{array}{ll} & N \leftarrow \mathsf{KGen}_{\mathsf{SRSA}}(1^k), \\ e > 1 \wedge z^e \equiv y \pmod N : & y \in_{\mathrm{U}} \mathrm{QR}_N, \\ & (z,e) \leftarrow \mathcal{R}(N,y) \end{array}\right]$$

is nonnegligible in k*, where the probability is taken over the coin flips of* $\mathsf{KGen}_{\mathsf{SRSA}}$ *and* $\mathcal{R}$*, and the uniform random choice* y *from* QR_N*.*

We will employ the following lemmas.

Lemma 2.11 ([44])
Let $N \in \mathbb{N}$ *with binary length* k*. Let* e *and* E^* *be any integers of length at most polynomial in* k*, and let* $z^*, y \in \mathbb{Z}_N^\times$ *such that* $(z^*)^e \equiv y^{E^*} \pmod N$*. If* $\gcd(e, E^*) = 1$*, then the element* $z \in \mathbb{Z}_N^\times$ *such that* $z^e \equiv y \pmod N$ *can be computed in polynomial-time in* k *on the input* (N, e, E^*, z^*, y)*.*

Lemma 2.12 ([34])
Let $N \in \mathbb{N}^{\mathrm{safe}}_{\mathsf{RSA}}$ *with binary length* k*. Let* e *and* E^* *be any integers of length at most polynomial in* k*, and let* $z^* \in \mathbb{Z}_N^\times$ *and* $y \in \mathrm{QR}_N$ *such that* $(z^*)^e \equiv y^{E^*} \pmod N$*. If* $e \nmid E^*$*, then an element* z *such that* $z^e \equiv y \pmod N$ *can be computed in polynomial-time in* k *on the tuple* (N, e, E^*, z^*, y)*.*

We follow the two setting on the RSA assumption and the SRSA assumption. The first is that an RSA modulus N is restricted to a product of two safe primes,

as in [13, 14, 34, 41]. The second is that y is restricted to a quadratic residue mod N. It should be noted that this is not an essential restriction. This is because breaking RSA for $y \in \mathrm{QR}_N$ leads to breaking RSA for an arbitrary $y \in \mathbb{Z}_N^\times$ in the following way. Assume that we are given a PPT algorithm $\mathcal{A}$ for solving the RSA problem for $y \in \mathrm{QR}_N$. Then, a given instance (N, e, y) with $y \in \mathbb{Z}_N^\times$, one can find an element $z \in \mathbb{Z}_N^\times$ with $z^e \equiv y \pmod{N}$ as follows:

(1) find an element z' such that $z'^e \equiv y^2 \pmod{N}$ by using $\mathcal{A}$,

(2) find an element z such that $z^e \equiv y \pmod{N}$ by employing Lemma 2.11, and then output z.

Note that $y^2 \in \mathrm{QR}_N$, $\gcd(e, 2) = 1$ (because $\varphi(N)$ is even and $\gcd(e, \varphi(N)) = 1$). In a similar manner, one can show that breaking SRSA for $y \in \mathrm{QR}_N$ leads to breaking SRSA for an arbitrary $y \in \mathbb{Z}_N^\times$ [34].

Chapter 3

Impossibility Results on the Adaptive Pseudo-Freeness of the RSA Group $\mathbb{Z}_N^\times$

In this chapter, we show several impossibility results on the adaptive pseudo-freeness of the RSA group family $\{\mathbb{Z}_N^\times\}$, where $\{\mathbb{Z}_N^\times\}$ stands for the RSA group family $\{\mathbb{Z}_N^\times\}_{N\in\mathcal{N}}$ with $\mathcal{N} = \mathbb{N}^{\mathrm{safe}}_{\mathsf{RSA}}$. This chapter is organized as follows. In Section 3.1, we give a negative circumstantial evidence for the adaptive pseudo-freeness of the RSA group family $\{\mathbb{Z}_N^\times\}$ in which the adaptive behavior of the adversary is not restricted. In Section 3.2, we show that it cannot be proven that the adaptive pseudo-freeness of the RSA group family $\{\mathbb{Z}_N^\times\}$ with respect to the parametric distributions proposed in [13] from the RSA assumption via algebraic reductions, as long as the RSA assumption holds. The concluding remarks for this chapter is given in Section 3.3.

3.1 Impossibility of the Strong Adaptive Pseudo-Freeness of $\mathbb{Z}_N^\times$

In this section, we give the following result. In a similar manner to Definition 2.6, we formally describe the strong adaptive pseudo-freeness in which the adaptive behavior of the adversary is not restricted. Then, we show that the strong adaptive pseudo-freeness for the family $\{\mathbb{Z}_N^\times\}$ of the RSA groups cannot be proven from the SRSA assumption via algebraic reductions, as long as the SRSA assumption holds.

Throughout this thesis, we assume that a group index $N \in \mathcal{N}(k)$ of a game tuple is distributed as the same as an RSA modulus N generated by $\mathsf{KGen}_{\mathsf{RSA}}(1^k)$.

We adopt any sampling algorithm for the family $\{\mathbb{Z}_N^\times\}$ which chooses an element g almost uniformly at random over QR_N. For example, Micciancio [34, Lemma 2] showed that if a generator y of QR_N is fixed, then such a sampling can be efficiently done by choosing an exponent $d \in_{\mathrm{U}} \{0, 1, \ldots, B-1\}$ with sufficiently large B and then setting $g := y^d$.

For the group QR_N of quadratic residues with $N \in \mathbb{N}_{\mathsf{RSA}}^{\mathrm{safe}}$, the following proposition holds.

Proposition 3.1 ([34])
Let $N = PQ \in \mathbb{N}_{\mathsf{RSA}}^{\mathrm{safe}}$ and let $y \in \mathrm{QR}_N$. If y is not a generator of QR_N, it holds that $y \equiv 1 \pmod N$ or $\gcd(y-1, N) \in \{P, Q\}$.

By this proposition, the solution z for an RSA instance (N, e, y) (a solution (z, e) for an SRSA instance (N, y), resp.) can be found in polynomial-time provided that y is not a generator of QR_N. Thus, we can assume without loss of generality that y is a generator of QR_N.

3.1.1 Strongly-Adaptive Pseudo-Free Groups

As explained in Section 2.3, Catalano, Fiore and Warinschi [13] introduced the notion of the adaptive pseudo-freeness as a generalization of the Rivest's "static" pseudo-freeness in order to handle adaptive adversaries. In their setting, the queried equations are chosen according to some specific parametric distribution. They also informally define in [13] the *strong* adaptive pseudo-freeness in a way that there is no such restriction, namely the adversary is allowed to freely choose his queries. We formally define the strong version of adaptive pseudo-freeness by the strongly-adaptive pseudo-free (SAPF, for short) game.

The *SAPF game* between the challenger and the adversary is defined as follows. Let $\mathcal{G} = \{\mathbb{G}_N\}_{N \in \mathcal{N}}$ be a computational group family and let $A = \{a_1, a_2, \ldots, a_m\}$ be a set of $m = m(k)$ symbols.

Setup. The challenger chooses a random group index $N \in_{\mathrm{U}} \mathcal{N}(k)$ at random. Then, it sets an assignment $\alpha : A \to \mathbb{G}_N$ by independently choosing an element $\alpha(a) \in_{\mathrm{R}} \mathbb{G}_N$ at random according to the designated sampling algorithm. The adversary $\mathcal{A}$ is given the *game pair* (N, α).

Equations queries. The adversary $\mathcal{A}$ is allowed to adaptively query to the challenger on equations, and to receive their solutions. For each t-th query, $\mathcal{A}$ chooses an arbitrary equation $\lambda^{(t)} = (E_t, \boldsymbol{s}_t)$ and hands it to the challenger.

The challenger returns a correct solution $\psi_t \in \mathbb{G}_N$ for the interpreted equation $\lambda_\alpha^{(t)} : x^{E_t} = \prod_{i=1}^m \alpha(a_i)^{s_{t,i}}$ to the adversary.

Challenge. The adversary outputs a witness pair (λ^*, ψ^*) of an equation $\lambda^* = (E^*, \boldsymbol{s}^*)$ and a solution ψ^* of the interpreted equation λ_α^* over $\mathbb{G}_N$. The challenger outputs 1 if the following conditions hold, or 0 otherwise:

- λ^* is nontrivial with respect to Λ, the set of queried equations and corresponding solutions appeared in Equations queries phase; and
- ψ^* is a correct solution for λ_α^*.

An adversary $\mathcal{A}$ is said to *win the strongly-adaptive pseudo-free* (*SAPF, for short*) *game of the family* $\mathcal{G}$ if the challenger outputs 1 in the game.

Definition 3.2 (Strongly-Adaptive Pseudo-Free Groups)
Let k be a security parameter, and let m be a polynomial in k. A family $\mathcal{G} = \{\mathbb{G}_N\}_{N \in \mathcal{N}}$ of computational groups is strongly-adaptive pseudo-free*, if there exists no PPT adversary $\mathcal{A}$ such that for any set A of m symbols, $\mathcal{A}$ wins the SAPF game of the family $\mathcal{G}$ in nonnegligible probability, where the probability is taken over the random choices of the index $N \in_{\mathrm{U}} \mathcal{N}(k)$, $\alpha(a) \in_{\mathrm{R}} \mathbb{G}_N$ for each $a \in A$, and the internal coin flips of $\mathcal{A}$.*

Remark 3.3
In the adaptive pseudo-free game given in Definition 2.6, the equation queries of the adversary $\mathcal{A}$ is determined by some specific parametric distribution. On the other hand, in the SAPF game, $\mathcal{A}$ can freely choose his queries. It is therefore necessary to consider the situation where $\mathcal{A}$ queries an equation which has no solution over $\mathbb{G}_N$. In this thesis, we assume that the challenger outputs the special symbol $\perp$ provided that a queried equation has no solution over $\mathbb{G}_N$.

3.1.2 Main Theorem

In this section, we show that the strong adaptive pseudo-freeness of the RSA group family $\{\mathbb{Z}_N^\times\}$ cannot be shown from the SRSA assumption via algebraic reductions, as long as the SRSA assumption holds.

Before stating the main result, we give a remark on the strong adaptive pseudo-freeness of the RSA groups $\{\mathbb{Z}_N^\times\}$. In Equations queries phase of the SAPF game, the adversary is allowed to choose an equation query arbitrarily. However, for $\mathbb{Z}_N^\times$, this setting does not seem to work properly without care. For example,

assume that the adversary $\mathcal{A}$ queries the equation $(2, (2, 0, \ldots, 0))$, namely $x^2 = a_1^2$, and receives a solution $\psi \in \mathbb{Z}_N^\times$ such that $\psi \neq \pm\alpha(a_1)$. Then $\mathcal{A}$ can easily factor N. Once N is factored, the adversary can easily find a witness pair (λ^*, ψ^*). Therefore, $\mathbb{Z}_N^\times$ would not be strongly-adaptive pseudo-free in the strict sense. In this section, we exclude such a situation. Instead, for any equation query $\lambda : x^E = \prod_{i=1}^m a_i^{s_i}$, the challenger is assumed to return a *canonical solution* ψ' for the interpreted equation λ_α, namely ψ' is a solution for the interpreted equation λ'_α of the reduced equation $\lambda' : x^{E'} = \prod_{i=1}^m \alpha(a_i)^{s'_i}$, where $E' = E/\gcd(E, s_1, \ldots, s_m)$ and $s'_i = s_i/\gcd(E, s_1, \ldots, s_m)$. For example, the challenger always returns the canonical solution $\psi = \alpha(a_1)$ on the query $x^2 = a_1^2$.

The Situation $\mathsf{SRSA} \leq \mathsf{SAPFG}_{\mathbb{Z}_N^\times}$**.** We describe the situation that *the SRSA assumption implies the strong adaptive pseudo-freeness of the RSA group family* $\{\mathbb{Z}_N^\times\}$, and then we write $\mathsf{SRSA} \leq \mathsf{SAPFG}_{\mathbb{Z}_N^\times}$. We formalize this statement by the following contrapositive setting similarly to [10, 37]: there exist a PPT algorithm $\mathcal{R}$ and polynomials m and q such that for any SAPF adversary $\mathcal{A}$ making at most q queries that wins the SAPF game of the family $\{\mathbb{Z}_N^\times\}$ in nonnegligible probability, where A is assumed to be a set of m symbols, $\mathcal{R}$ breaks SRSA in nonnegligible probability with a black-box access to the adversary $\mathcal{A}$. We may assume without loss of generality that $q \geq 2$. Through the black-box access, $\mathcal{R}$ would play the SAPF game with the adversary $\mathcal{A}$ in which $\mathcal{R}$ is placed at the challenger's position.

Given an SRSA instance (N, y), $\mathcal{R}$ follows Setup phase of the SAPF game, namely $\mathcal{R}$ chooses a game pair (N, α). Especially, the assignment map α is chosen by selecting $\alpha(a)$ almost uniformly at random from QR_N for each $a \in A$. We assume as in [13, 34] that the index N of the game pair is always the same as the modulus N of the given SRSA instance. Moving to Equations queries phase, $\mathcal{A}$ makes equation queries $\lambda = (E, \boldsymbol{s})$ at most q times. Since $\mathcal{R}$ is now playing the role of the challenger, $\mathcal{R}$ replies the answer for each of the queries, but $\mathcal{R}$ may fail to reply the correct answer because the reduction $\mathcal{R}$ is polynomial-time bounded. Eventually, the game completes with $\mathcal{A}$'s output: a "winning" witness pair (λ^*, ψ^*) of the SAPF game, or "losing" symbol $\perp$. After the game, $\mathcal{R}$ would find a correct solution (z, e) for the given SRSA instance (N, y) with nonnegligible probability ϵ_0.

In this thesis, we force the reduction $\mathcal{R}$ to be *algebraic with respect to the*

group QR_N *for any* $N \in \mathcal{N}$. Consequently, any element $g \in \mathrm{QR}_N$ produced in the execution of $\mathcal{R}$ is generated by the given element y and the expression $g = y^d$ is easily retrieved by the extraction algorithm $\mathsf{Extract}$, provided that g is actually produced in the execution of $\mathcal{R}(N, y)$. In particular, for the assignment α and each $a \in A$, $\alpha(a)$ is of the form $\alpha(a) = y^d$ and the exponent d can be easily retrieved.

We now ready to state our main theorem for the strong adaptive pseudo-freeness of the RSA group family $\{\mathbb{Z}_N^\times\}$.

Theorem 3.4
If $\mathsf{SRSA} \leq \mathsf{SAPFG}_{\mathbb{Z}_N^\times}$, *then the SRSA assumption does not hold.*

Proof. Assume that $\mathsf{SRSA} \leq \mathsf{SAPFG}_{\mathbb{Z}_N^\times}$. Then, there exist a PPT algorithm $\mathcal{R}$ and polynomials m and $q \geq 2$ such that $\mathcal{R}$ is algebraic with respect to QR_N for any $N \in \mathcal{N}$, and $\mathcal{R}$ breaks SRSA in nonnegligible probability with a black-box access to any PPT adversary $\mathcal{A}$ making at most q queries that wins the SAPF game of the family $\{\mathbb{Z}_N^\times\}$ in nonnegligible probability. This means that for any security parameter k, $\mathcal{R}$ breaks SRSA with at least nonnegligible probability ϵ_0, where an instance (N, y) is chosen as in Definition 2.10. Namely an RSA modulus N is generated by $\mathsf{KGen}_{\mathsf{SRSA}}(1^k)$ and an element y is uniformly distributed over QR_N. As mentioned just before Section 3.1.1, we may assume without loss of generality that y is a generator of QR_N.

Construction of the Meta-Reduction $\mathcal{M}$. We shall construct a PPT algorithm $\mathcal{M}$ that breaks SRSA with no oracle access at least nonnegligible probability. We shall provide for the reduction $\mathcal{R}$ a simulator $\mathsf{Sim}_{\mathcal{A}}$ that plays a winning SAPF adversary's role to $\mathcal{R}$. In other words, from the $\mathcal{R}$'s viewpoint, $\mathsf{Sim}_{\mathcal{A}}$ looks like a "real" winning adversary, namely it actually wins the SAPF game with nonnegligible probability provided that the reduction $\mathcal{R}$ is supposed to be *ideal* as a challenger in a sense that $\mathcal{R}$ always replies a correct answer to each query from $\mathsf{Sim}_{\mathcal{A}}$. If $\mathsf{Sim}_{\mathcal{A}}$ is set to the adversary's position, then $\mathcal{R}$ would break SRSA via playing the SAPF game with $\mathsf{Sim}_{\mathcal{A}}$. Thus, our meta-reduction $\mathcal{M}$ is constructed by involving $\mathcal{R}$ and $\mathsf{Sim}_{\mathcal{A}}$.

For the algorithm $\mathsf{Sim}_{\mathcal{A}}$, we may assume without loss of generality that the final output of $\mathsf{Sim}_{\mathcal{A}}$ can be a correct solution (z, e) for the given SRSA instance (N, y) if $\mathsf{Sim}_{\mathcal{A}}$ fortunately finds it, instead of a witness pair (λ^*, ψ^*) of the SAPF

game. This does not lower the success probability ϵ_0 of the reduction $\mathcal{R}$. Thus, our simulator $\mathsf{Sim}_{\mathcal{A}}$ is to be a PPT algorithm that on a given game pair (N, α) with an auxiliary input (y, ω), where (N, α) is a game pair presented by $\mathcal{R}$, y is an element of a given SRSA instance (N, y) and ω is a random coin used by $\mathcal{R}$ when $\mathcal{R}$ generates the assignment map α, responds one of the following items (I)–(III):

(I) $\mathsf{Sim}_{\mathcal{A}}$ finds a witness pair (λ^*, ψ^*) of a nontrivial equation λ^* and a solution ψ^* for the interpreted equation λ^*_α.

(II) In a fortunate case, $\mathsf{Sim}_{\mathcal{A}}$ may find a solution (z, e) for the given SRSA instance (N, y) in its execution. If $\mathsf{Sim}_{\mathcal{A}}$ meets its fortunate case, then $\mathsf{Sim}_{\mathcal{A}}$ outputs the solution (z, e).

(III) $\mathsf{Sim}_{\mathcal{A}}$ may abort with the output $\perp$ in an unfortunate case.

We note that $\mathsf{Sim}_{\mathcal{A}}$ is required only to come into either the case (I) or the case (II) with nonnegligible probability and within q queries to $\mathcal{R}$ provided that $\mathcal{R}$ is ideal as a challenger. By using $\mathsf{Sim}_{\mathcal{A}}$, the algorithm $\mathcal{M}$ is constructed as in Fig. 3.1. If $\mathsf{Sim}_{\mathcal{A}}$ is constructed in that way, $\mathcal{R}$ breaks SRSA with at least nonnegligible probability ϵ_0 and consequently the resulting algorithm $\mathcal{M}$ will succeed in nonnegligible probability.

Construction of $\mathsf{Sim}_{\mathcal{A}}$. In order to construct the algorithm $\mathcal{M}$, it suffices to construct the simulator $\mathsf{Sim}_{\mathcal{A}}$. Since $\mathcal{R}$ is algebraic with respect to QR_N for any N, there exists a polynomial time algorithm $\mathsf{Extract}$ that on a tuple (N, y, g, ω), where g is a target element in QR_N that is produced in the execution of $\mathcal{R}$ on the input (N, y) with the random coin ω, returns an exponent d such that $g = y^d$. We involve $\mathsf{Extract}$ in the construction of $\mathsf{Sim}_{\mathcal{A}}$. The algorithm $\mathsf{Sim}_{\mathcal{A}}$ is depicted in Fig. 3.2. On each game pair (N, α) given from $\mathcal{R}$ with the auxiliary input (y, ω),

- if $\alpha(a_i) = 1 \in \mathrm{QR}_N$ for all $i \in [1, m]$, then $\mathsf{Sim}_{\mathcal{A}}$ outputs a correct witness pair (λ^*, ψ^*) in the step (A-1) (the case (I)); or
- if $\alpha(a_{i_0}) \neq 1$ for some i_0, then in the step (A-2), $\mathsf{Sim}_{\mathcal{A}}$ attempts to find a solution (z, e) of the SRSA instance (N, y) by interacting to the challenger $\mathcal{R}$:
  - if $\mathsf{Sim}_{\mathcal{A}}$ has found a solution (z, e) for SRSA, then $\mathsf{Sim}_{\mathcal{A}}$ outputs the solution (z, e) (the case (II)), or

Input. an SRSA instance (N, y).
Output. a solution (z, e) of the given SRSA instance (N, y).

(M-1) $\mathcal{M}$ chooses a random coin ω, and then executes $\mathcal{R}$ on the instance (N, y) with using ω.

(M-2) When $\mathcal{R}$ submits a game pair (N, α) to the adversary, $\mathcal{M}$ invokes $\mathsf{Sim}_{\mathcal{A}}$ on the tuple (N, α) with the auxiliary input (y, ω). Then $\mathcal{R}$ and $\mathsf{Sim}_{\mathcal{A}}$ plays the SAPF game.

(M-3) $\mathsf{Sim}_{\mathcal{A}}$ outputs a response $\gamma \in \{(\lambda^*, \psi^*), \perp, (z, e)\}$, and halts.

(M-4) After receiving the response γ, $\mathcal{M}$ behaves as follows:

(M-4a) either $\gamma = (\lambda^*, \psi^*)$ or $\gamma = \perp$: $\mathcal{M}$ continues to simulate $\mathcal{R}$, and then halts with outputting the final output of $\mathcal{R}$.

(M-4b) $\gamma = (z, e)$: $\mathcal{M}$ halts with outputting (z, e).

Figure 3.1: Configuration of $\mathcal{M}$.

– otherwise, $\mathsf{Sim}_{\mathcal{A}}$ outputs $\perp$ (the case (III)).

Correctness of $\mathsf{Sim}_{\mathcal{A}}$. (A-1) We consider the case where $\alpha(a_i) = 1$ for all $i \in [1, m]$. Note that the triviality of the equations in this case is equivalent to the one in the static case, because $\mathsf{Sim}_{\mathcal{A}}$ makes no query to the challenger $\mathcal{R}$. Namely, the triviality of the equation merely means that it has no solution over the free group $\mathcal{F}(A)$. Therefore, the equation $\lambda^* = (2, (3, \ldots, 3))$ is nontrivial by Lemma 2.1. Moreover, because $\alpha(a_i) = 1$ for all $i \in [1, m]$, $\psi = 1 \in \mathbb{Z}_N^\times$ is a solution for the interpreted equation λ_α^*. Thus the output (λ^*, ψ^*) of $\mathsf{Sim}_{\mathcal{A}}$ is a correct witness pair. This is the case (I).

(A-2) We next consider the case where $\alpha(a_{i_0}) \neq 1$ for some $i_0 \in [1, m]$. We show that $\mathsf{Sim}_{\mathcal{A}}$ outputs a solution (z, e) for the given SRSA instance (N, y) by interacting to the challenger $\mathcal{R}$ or outputs $\perp$. Since the assignment α is generated before the game pair (N, α) is given to $\mathcal{A}$, for each $i \in [1, m]$, $\mathsf{Sim}_{\mathcal{A}}$ can retrieve an exponent d_i such that $\alpha(a_i) = y^{d_i}$ by executing $\mathsf{Extract}$ on the tuple $(N, y, \alpha(a_i), \omega)$. Note that the exponent d_i exists, because the element

Input. the game pair (N, α) with the auxiliary input (y, ω), as in (M-2).
Output. one of the following:

(I) a pair (λ^*, ψ^*) of a nontrivial equation and its interpreted solution;

(II) a pair (z, e) such that $z^e \equiv y \pmod{N}$; or

(III) the special symbol $\perp$.

(A-1) If $\alpha(a_i) = 1$ for all $i \in [1, m]$, then set $\lambda^* := (2, (3, \ldots, 3))$ and $\psi^* := 1$, and then halt with outputting the tuple (λ^*, ψ^*).

(A-2) If $\alpha(a_{i_0}) \neq 1$ for some i_0, then for each $i \in [1, m]$, retrieve an exponent d_i of the element $\alpha(a_i) \in \mathrm{QR}_N$ such that $\alpha(a_i) = y^{d_i}$ by executing $\mathsf{Extract}(N, y, \alpha(a_i), \omega)$.

(A-2a) Choose $s_1, \ldots, s_m \in_{\mathrm{U}} \mathbb{Z}_N$, and set $D := \sum_{i=1}^m d_i s_i$.
If $D = 0$, then reset $s_{i_0} := s_{i_0} + 1$, and $D := \sum_{i=1}^m d_i s_i$.
Set $\boldsymbol{s} := (s_1, \ldots, s_m)$.

(A-2b) Set $E_1 := |D| + 1$ and $E_2 := |D| + 2$.
If $y^{E_1 E_2} \equiv 1 \pmod{N}$, then halt with outputting the pair $(z, e) := (y, E_1 E_2 + 1)$.
Else, proceed to (A-2c).

(A-2c) For each $t \in \{1, 2\}$, submit the equation $\lambda^{(t)} = (E_t, \boldsymbol{s})$ to $\mathcal{R}$, and then receive a solution ψ_t from $\mathcal{R}$.
If ψ_{t_0} is a correct solution for $\lambda_\alpha^{(t_0)}$ for some $t_0 \in \{1, 2\}$, then compute a pair (z, e) such that $z^e \equiv y \pmod{N}$ by Lemma 2.12, and then halt with outputting (z, e).
Else, halt with outputting $\perp$.

Figure 3.2: Configuration of $\mathsf{Sim}_{\mathcal{A}}$

$\alpha(a_i) \in \mathrm{QR}_N$ is produced by the algebraic algorithm $\mathcal{R}$, and hence it belongs to the subgroup $\langle y \rangle$ generated by $y \in \mathrm{QR}_N$. Therefore, for any equation $\lambda = (E, (s_1, \ldots, s_m))$, the interpreted equation λ_α over $\mathbb{Z}_N^\times$ is expressed in a way that $x^E = \prod_{i=1}^m \alpha(a_i)^{s_i} = \prod_{i=1}^m \left(y^{d_i}\right)^{s_i} = y^{\sum_{i=1}^m d_i s_i}$. We say that an equation $\lambda = (E, (s_1, \ldots, s_m))$ is *good* if the interpreted equation $\lambda_\alpha : x^E = y^D$, where $D = \sum_{i=1}^m d_i s_i$, has a solution $\psi \in \langle y \rangle$ and $E \nmid D$. Note that if there is no solution of λ_α in $\langle y \rangle$, $\mathcal{R}$ cannot find the solution of λ_α, although it has a solution in $\mathbb{Z}_N^\times$. This is because $\mathcal{R}$ is algebraic. If $\mathsf{Sim}_\mathcal{A}$ queries a good equation λ to the challenger $\mathcal{R}$ and succeeds to receive a correct solution $\psi \in \langle y \rangle$ for λ_α, then it has obtained the tuple (N, E, D, ψ, y) such that $E \nmid D$ and $\psi^E \equiv y^D \pmod N$. Therefore, $\mathsf{Sim}_\mathcal{A}$ can efficiently find a correct solution (z, e) for the SRSA instance (N, y) by applying Lemma 2.12 to the tuple (N, E, D, ψ, y). This is the case (II).

We now show that one of the following events occurs:

(i) at least one of the equation queries $\lambda^{(1)} = (E_1, \boldsymbol{s})$ and $\lambda^{(2)} = (E_2, \boldsymbol{s})$ generated in the steps (A-2a) and (A-2b) is a good equation; or

(ii) a correct solution (z, e) for the SRSA instance (N, y) is found.

It is easy to observe that the integer $D = \sum_{i=1}^m d_i s_i$ generated in (A-2a) is not zero. In the step (A-2b), $\mathsf{Sim}_\mathcal{A}$ computes integers $E_1 = |D| + 1$ and $E_2 = |D| + 2$. If $y^{E_1 E_2} \equiv 1 \pmod N$, then it is obvious that the pair $(z, e) = (y, E_1 E_2 + 1)$ is a solution for the given SRSA instance (N, y). Otherwise, by the claims **Claim 3.5** and **Claim 3.6**, it is shown that there exists an index $t_0 \in \{1, 2\}$ such that the equation $\lambda^{(t_0)} = (E_{t_0}, \boldsymbol{s})$ is good, namely the interpreted equation $\lambda_\alpha^{(t_0)}$ has a solution over $\langle y \rangle$ and $E_{t_0} \nmid D$, where $\boldsymbol{s} = (s_1, \ldots, s_m)$ has been generated in (A-2a). The proofs are give in Section 3.1.3.

Claim 3.5
Assume that $y^{E_1 E_2} \not\equiv 1 \pmod N$. Then, there exists an index $t_0 \in \{1, 2\}$ such that the interpreted equation $\lambda_\alpha^{(t_0)} : x^{E_{t_0}} = y^D$ has a solution $\psi_{t_0} \in \langle y \rangle$ for the equation $\lambda^{(t_0)} = (E_{t_0}, \boldsymbol{s})$.

Claim 3.6
The integers E_1 and E_2 found in (A-2b) satisfy $E_1 \nmid D$ and $E_2 \nmid D$.

In the step (A-2c), for each $t \in \{1, 2\}$, $\mathsf{Sim}_\mathcal{A}$ queries the equation $\lambda^{(t)} = (E_t, \boldsymbol{s})$ to the challenger $\mathcal{R}$, and then receives the solution ψ_t over $\mathbb{Z}_N^\times$ of the interpreted equation $\lambda_\alpha^{(t)} : x^{E_t} = y^D$.

When ψ_{t_0} is a correct solution for $\lambda_\alpha^{(t_0)}$ for some $t_0 \in \{1,2\}$, we have $\psi_{t_0}^{E_{t_0}} \equiv y^D \pmod N$. Since $E_{t_0} \nmid D$ by **Claim 3.6**, $\mathsf{Sim}_\mathcal{A}$ can find a solution for the given SRSA instance (N, y) by applying Lemma 2.12 to the tuple $(N, E_{t_0}, D, \psi_{t_0}, y)$. This is the case (II).

Otherwise, ψ_t is not a correct solution of $\lambda_\alpha^{(t)}$ for any $t \in \{1,2\}$. By **Claim 3.5**, this means that for some index $t_0 \in \{1,2\}$, $\mathcal{R}$ failed to find a solution of $\lambda_\alpha^{(t_0)}$ despite that $\lambda_\alpha^{(t_0)}$ has a solution in the subgroup $\langle y \rangle$. This is the unfortunate case, namely the case (III). Therefore, $\mathsf{Sim}_\mathcal{A}$ halts with outputting $\perp$.

It immediately follows from the construction that with the probability 1, $\mathsf{Sim}_\mathcal{A}$ outputs either a witness pair (λ^*, ψ^*) or a solution (z, e) for the given SRSA instance (N, y), if $\mathcal{R}$ is ideal as a challenger, namely $\mathsf{Sim}_\mathcal{A}$ can receive a correct reply of the equation query $\lambda^{(t)}$ from $\mathcal{R}$ for each $t \in \{1,2\}$.

The Success Probability of $\mathcal{M}$. Finally, we estimate the success probability of $\mathcal{M}$. We denote by $\Pr[\mathsf{Succ}_\mathcal{M}]$ and $\Pr[\mathsf{Succ}_\mathcal{R}]$ the success probability of $\mathcal{M}$ and $\mathcal{R}$, respectively. It is guaranteed that $\Pr[\mathsf{Succ}_\mathcal{R}] \geq \epsilon_0$ by *Correctness of* $\mathsf{Sim}_\mathcal{A}$.

Unity designates the event that the assignment $\alpha : A \to \mathbb{Z}_N^\times$ of the game pair (N, α) satisfies that $\alpha(a_i) = 1$ for all $i \in [1, m]$.

The event Unity happens. In this case, $\mathsf{Sim}_\mathcal{A}$ outputs the witness pair (λ^*, ψ^*) of a nontrivial equation and its corresponding solution, and then $\mathcal{M}$ outputs the final output of $\mathcal{R}$ at (M-4a). Therefore, we have

$$\Pr\left[\mathsf{Succ}_\mathcal{M} \wedge \mathsf{Unity}\right] = \Pr\left[\mathsf{Succ}_\mathcal{R} \wedge \mathsf{Unity}\right]. \tag{3.1}$$

The event Unity does not happen. Let $\mathsf{SolveEq}$ denote the event that for some index $t_0 \in \{1,2\}$, $\mathcal{R}$ correctly solves the interpreted equation $\lambda_\alpha^{(t_0)} : x^{E_{t_0}} = y^D$ of the queried equation $\lambda^{(t_0)}$ from $\mathsf{Sim}_\mathcal{A}$ during playing the SAPF game.

If the event $\mathsf{SolveEq}$ happens, $\mathcal{M}$ outputs the pair (z, e) returned from $\mathsf{Sim}_\mathcal{A}$ at (M-4b). Since $z^e \equiv y \pmod N$ holds under the event $\mathsf{SolveEq}$ by the correctness of $\mathsf{Sim}_\mathcal{A}$, we have $\Pr\left[\mathsf{Succ}_\mathcal{M} \mid \neg\mathsf{Unity} \wedge \mathsf{SolveEq}\right] = 1$. This implies that

$$\begin{aligned}
&\Pr\left[\mathsf{Succ}_\mathcal{M} \wedge (\neg\mathsf{Unity} \wedge \mathsf{SolveEq})\right] \\
&\quad = \Pr\left[\mathsf{Succ}_\mathcal{M} \mid \neg\mathsf{Unity} \wedge \mathsf{SolveEq}\right] \Pr\left[\neg\mathsf{Unity} \wedge \mathsf{SolveEq}\right] \\
&\quad = \Pr\left[\neg\mathsf{Unity} \wedge \mathsf{SolveEq}\right] \\
&\quad \geq \Pr\left[\mathsf{Succ}_\mathcal{R} \wedge (\neg\mathsf{Unity} \wedge \mathsf{SolveEq})\right].
\end{aligned} \tag{3.2}$$

If the event $\mathsf{SolveEq}$ does not happen, $\mathcal{M}$ outputs the final output of $\mathcal{R}$ in (M-4a). Therefore, we have

$$\Pr\left[\mathsf{Succ}_{\mathcal{M}} \wedge (\neg\mathsf{Unity} \wedge \neg\mathsf{SolveEq})\right] = \Pr\left[\mathsf{Succ}_{\mathcal{R}} \wedge (\neg\mathsf{Unity} \wedge \neg\mathsf{SolveEq})\right]. \quad (3.3)$$

Putting together Eqs. (3.1)–(3.3), we have

$$\begin{aligned}
\Pr[\mathsf{Succ}_{\mathcal{M}}] &= \Pr[\mathsf{Succ}_{\mathcal{M}} \wedge \mathsf{Unity}] + \Pr[\mathsf{Succ}_{\mathcal{M}} \wedge \neg\mathsf{Unity}] \\
&= \Pr[\mathsf{Succ}_{\mathcal{M}} \wedge \mathsf{Unity}] + \Pr[\mathsf{Succ}_{\mathcal{M}} \wedge (\neg\mathsf{Unity} \wedge \mathsf{SolveEq})] \\
&\quad + \Pr[\mathsf{Succ}_{\mathcal{M}} \wedge (\neg\mathsf{Unity} \wedge \neg\mathsf{SolveEq})] \\
&\geq \Pr[\mathsf{Succ}_{\mathcal{R}} \wedge \mathsf{Unity}] + \Pr[\mathsf{Succ}_{\mathcal{R}} \wedge (\neg\mathsf{Unity} \wedge \mathsf{SolveEq})] \\
&\quad + \Pr[\mathsf{Succ}_{\mathcal{R}} \wedge (\neg\mathsf{Unity} \wedge \neg\mathsf{SolveEq})] \\
&= \Pr[\mathsf{Succ}_{\mathcal{R}} \wedge \mathsf{Unity}] + \Pr[\mathsf{Succ}_{\mathcal{R}} \wedge \neg\mathsf{Unity}] \\
&= \Pr[\mathsf{Succ}_{\mathcal{R}}] \\
&\geq \epsilon_0.
\end{aligned}$$

Thus, $\mathcal{M}$ breaks SRSA with probability at least ϵ_0. Hence the SRSA assumption does not hold. □

3.1.3 Proofs of the Claims in Theorem 3.4

We now show the claims employed in Theorem 3.4.

Proof of Claim 3.5. Assume that $y^{E_1E_2} \not\equiv 1 \pmod N$. Then, we now show that one of $E_1 \in \mathbb{Z}^\times_{P'Q'}$ and $E_2 \in \mathbb{Z}^\times_{P'Q'}$ holds. We assume that $E_1, E_2 \notin \mathbb{Z}^\times_{P'Q'}$. Then, we have $\gcd(E_1, P'Q'), \gcd(E_2, P'Q') \in \{P', Q', P'Q'\}$. If either $\gcd(E_1, P'Q') = P'Q'$ or $\gcd(E_2, P'Q') = P'Q'$ holds, then $y^{E_1E_2} \equiv 1 \pmod N$ holds, since $y \in \mathrm{QR}_N$ and the order $\mathrm{ord}(\mathrm{QR}_N)$ of QR_N is $P'Q'$. This is a contradiction. Otherwise, we assume without loss of generality that $\gcd(E_1, P'Q') = P'$ holds. Then, there exists an integer $b_1 \in \mathbb{Z}$ such that $E_1 = b_1P'$. Moreover, we have $E_2 = E_1 + 1 \not\equiv 0 \pmod{P'}$, and hence $P' \nmid E_2$. By the assumption, $\gcd(E_2, P'Q') = Q'$ holds. Then, there exists an integer $b_2 \in \mathbb{Z}$ such that $E_2 = b_2Q'$. It follows that $E_1E_2 = b_1b_2P'Q'$. This implies that $y^{E_1E_2} \equiv 1 \pmod N$. This is a contradiction. Thus, one of $E_1 \in \mathbb{Z}^\times_{P'Q'}$ and $E_2 \in \mathbb{Z}^\times_{P'Q'}$ holds.

Let $t_0 \in \{1, 2\}$ be an index such that $E_{t_0} \in \mathbb{Z}^\times_{P'Q'}$. Since $y \in \mathrm{QR}_N$ and $\mathrm{ord}(\mathrm{QR}_N) = P'Q'$, the interpreted equation $\lambda_\alpha^{(t_0)} : x^{E_{t_0}} = y^D$ has a solution $y^{DE_{t_0}^{-1}}$ that belongs to the subgroup $\langle y \rangle$, where $E_{t_0}^{-1}$ denotes the inverse of E_{t_0} in $\mathbb{Z}^\times_{P'Q'}$. □

Proof of Claim 3.6. It follows from $D \neq 0$ and $0 < |D| < |D|+1 = E_1 < E_2$ that $E_1 \nmid D$ and $E_2 \nmid D$. □

3.2 Impossibility of the Adaptive Pseudo-Freeness of $\mathbb{Z}_N^\times$ under the RSA Assumption

Catalano, Fiore and Warinschi [13] presented a class $\mathfrak{D}^{\mathsf{CFW}}$ of parametric distributions, and showed that the SRSA assumption implies the adaptive pseudo-freeness of the RSA group family $\left\{\mathbb{Z}_N^\times\right\}$ with respect to any parametric distribution in $\mathfrak{D}^{\mathsf{CFW}}$.

In this section, we give the following result. We first define a family $\mathcal{E} = \{\mathcal{E}_{k,m}\}$ of classes $\mathcal{E}_{k,m}$ and a class $\mathfrak{D}$ of parametric distributions. Then, we show that the adaptive pseudo-freeness of $\left\{\mathbb{Z}_N^\times\right\}$ with respect to any parametric distribution $\varrho \in \mathfrak{D}$ cannot be proven from the RSA assumption via algebraic reductions, as long as the RSA assumption holds. Note that our class $\mathcal{E}_{k,m}$ is sufficiently large so that it involves all pairs (λ, r) that can be obtained by following the CFW's setting given in [13]. Moreover, our main theorem holds even when an adversary is *static*, namely the adversary is restricted so that it makes no query during the APF game. Thus, our result indicates that even the "static" pseudo-freeness of $\left\{\mathbb{Z}_N^\times\right\}$ over $\mathcal{E}$ cannot be proven from the RSA assumption via algebraic reductions, as long as the RSA assumption holds.

Consequently, we also show that several SRSA-based signature schemes that are yielded from the construction proposed by Catalano-Fiore-Warinschi [13] cannot be proven to be sEUF-CMA, and even EUF-KOA, from RSA assumption via algebraic reductions, as long as the RSA assumption holds.

3.2.1 Main Theorem

Our class $\mathcal{E}_{k,m}$ is defined in the following manner. We fix any nonconstant polynomials $\ell_{\mathrm{msg}} = \ell_{\mathrm{msg}}(k)$, $\ell_{\mathrm{exp}} = \ell_{\mathrm{exp}}(k)$ and $\ell_{\mathrm{seed}} = \ell_{\mathrm{seed}}(k)$ such that $\ell_{\mathrm{exp}} \leq \ell/2 - 2$, and any (single-valued) collision-resistant hash function $H : \{0,1\}^{\ell_{\mathrm{seed}}} \to \left[0, 2^{\ell_{\mathrm{exp}}} - 1\right]$. For each k and m, let $\mathcal{E}_{k,m}$ be the set of all pairs (λ, r) of an equation $\lambda = (E, (s_1, \ldots, s_m))$ and a string $r \in \{0,1\}^{\ell_{\mathrm{seed}}}$ such that $E = H(r) \in \left[0, 2^{\ell_{\mathrm{exp}}} - 1\right]$, $s_1 = 1$ and $s_2, \ldots, s_m \in \mathbb{Z}$.

In our main theorem, we focus on a class $\mathfrak{D}$ of parametric distributions. Any parametric distribution $\varrho = \{\varrho_{k,m}\}_{k,m}$ belonging to $\mathfrak{D}$ satisfies the following

conditions. For each k and m, $\varrho_{k,m} = \{\varrho_{k,m}(M)\}$ is a family of probabilistic distributions $\varrho_{k,m}(M)$ over the set $\mathcal{E}_{k,m}$ such that for any given parameter $M \in \{0,1\}^{\ell_{\mathrm{msg}}}$,

- $\varrho_{k,m}(M)$ is polynomial-time samplable;
- the description of $\varrho_{k,m}(M)$ can be obtained in polynomial-time in k; and
- a string r is uniformly distributed over $\{0,1\}^{\ell_{\mathrm{seed}}}$.

Note that this requirement is immediately fulfilled also in the CFW's setting. Namely it holds that $\mathfrak{D}^{\mathsf{CFW}} \subseteq \mathfrak{D}$.

We employ the following lemma for the hash function H.

Lemma 3.7
If $H : \{0,1\}^{\ell_{\mathrm{seed}}} \to \left[0, 2^{\ell_{\mathrm{exp}}} - 1\right]$ is a collision-resistant hash function, then for any integer $E \in \left[0, 2^{\ell_{\mathrm{exp}}} - 1\right]$,

$$\Pr_{r \in_{\mathrm{U}} \{0,1\}^{\ell_{\mathrm{seed}}}} [H(r) = E] = \mathrm{negl}(k).$$

Proof. We show the lemma by the contraposition. Assume that there exists an integer $E_0 \in \left[0, 2^{\ell_{\mathrm{exp}}} - 1\right]$ such that $\Pr_{r \in_{\mathrm{U}} \{0,1\}^{\ell_{\mathrm{seed}}}} [H(r) = E_0]$ is nonnegligible, that is there exists a polynomial p in k such that

$$\Pr_{r \in_{\mathrm{U}} \{0,1\}^{\ell_{\mathrm{seed}}}} [H(r) = E_0] \geq \frac{1}{p(k)},$$

for sufficiently large k. This implies that for the set $H^{-1}(E_0)$ of all strings $r \in \{0,1\}^{\ell_{\mathrm{seed}}}$ such that $H(r) = E_0$,

$$\begin{aligned}
\Pr_{r \in_{\mathrm{U}} \{0,1\}^{\ell_{\mathrm{seed}}}} [H(r) = E_0] &= \Pr_{r \in_{\mathrm{U}} \{0,1\}^{\ell_{\mathrm{seed}}}} \left[r \in H^{-1}(E_0)\right] \\
&= \sum_{x \in H^{-1}(E_0)} \Pr_{r \in_{\mathrm{U}} \{0,1\}^{\ell_{\mathrm{seed}}}} [x = r] \\
&= \sum_{x \in H^{-1}(E_0)} \frac{1}{2^{\ell_{\mathrm{seed}}}} \\
&= \frac{|H^{-1}(E_0)|}{2^{\ell_{\mathrm{seed}}}}.
\end{aligned}$$

Namely the density of $H^{-1}(E_0)$ is at least $1/p(k)$ for sufficiently large k. The

following inequality therefore holds.

$$\begin{aligned}
&\Pr_{r_1,r_2\in_{\mathrm{U}}\{0,1\}^{\ell_{\mathrm{seed}}}}[H(r_1)=H(r_2)] \\
&\quad\geq \Pr_{r_1,r_2\in_{\mathrm{U}}\{0,1\}^{\ell_{\mathrm{seed}}}}[H(r_1)=E_0\wedge H(r_2)=E_0] \\
&\quad= \Pr_{r_1,r_2\in_{\mathrm{U}}\{0,1\}^{\ell_{\mathrm{seed}}}}\left[r_1,r_2\in H^{-1}(E_0)\right] \\
&\quad= \Pr_{r_1\in_{\mathrm{U}}\{0,1\}^{\ell_{\mathrm{seed}}}}\left[r_1\in H^{-1}(E_0)\right]\cdot \Pr_{r_2\in_{\mathrm{U}}\{0,1\}^{\ell_{\mathrm{seed}}}}\left[r_2\in H^{-1}(E_0)\right] \\
&\quad= \left(\frac{|H^{-1}(E_0)|}{2^{\ell_{\mathrm{seed}}}}\right)^2 \\
&\quad\geq \frac{1}{p^2(k)}
\end{aligned} \tag{3.4}$$

On the other hand, since H is single-valued, we have

$$\begin{aligned}
&\Pr[H(r_1)=H(r_2)] \\
&\quad= \Pr[H(r_1)=H(r_2)\wedge r_1=r_2]+\Pr[H(r_1)=H(r_2)\wedge r_1\neq r_2] \\
&\quad= \Pr[H(r_1)=H(r_2)\mid r_1=r_2]\Pr[r_1=r_2] \\
&\qquad+\Pr[H(r_1)=H(r_2)\wedge r_1\neq r_2] \\
&\quad= \Pr[r_1=r_2]+\Pr[H(r_1)=H(r_2)\wedge r_1\neq r_2] \\
&\quad= \frac{1}{2^{\ell_{\mathrm{seed}}}}+\Pr[H(r_1)=H(r_2)\wedge r_1\neq r_2].
\end{aligned} \tag{3.5}$$

Putting together Eq. (3.4) and Eq. (3.5), we have

$$\Pr[H(r_1)=H(r_2)\wedge r_1\neq r_2]\geq \frac{1}{p^2(k)}-\frac{1}{2^{\ell_{\mathrm{seed}}}}.$$

This means that a collision (r_1, r_2) can be found by merely choosing two strings r_1 and r_2 uniformly and independently at random from $\{0,1\}^{\ell_{\mathrm{seed}}}$. Thus H is not a collision-resistant hash function. □

The Situation $\mathsf{RSA}\leq\mathsf{APFG}_{\mathbb{Z}_N^\times,\varrho}$. In a similar manner to $\mathsf{SRSA}\leq\mathsf{SAPFG}_{\mathbb{Z}_N^\times}$ formalized in Section 3.1.2, we describe the situation that *the RSA assumption implies the adaptive pseudo-freeness of the RSA group family* $\{\mathbb{Z}_N^\times\}$ *with respect to a parametric distribution* $\varrho=\{\varrho_{k,m}\}_{k,m}$, and then we write $\mathsf{RSA}\leq\mathsf{APFG}_{\mathbb{Z}_N^\times,\varrho}$ to denote such a situation. Namely this is formalized as follows: there exist a PPT algorithm $\mathcal{R}$ and a polynomial m such that for any PPT adversary $\mathcal{A}$ that wins

the APF game of the family $\{\mathbb{Z}_N^\times\}$ with respect to $\varrho = \{\varrho_{k,m}\}_{k,m}$ in nonnegligible probability, $\mathcal{R}$ breaks RSA in nonnegligible probability with a black-box access to such an adversary $\mathcal{A}$. Through the black-box access, $\mathcal{R}$ would play the APF game with the adversary $\mathcal{A}$ in which $\mathcal{R}$ is placed at the challenger's position.

Let (N, e, y) be a given RSA instance. Following Setup phase of the APF game, $\mathcal{R}$ sets a game tuple $(N, \alpha, \varrho_{k,m})$. As in **The Situation** $\mathsf{SRSA} \leq \mathsf{SAPFG}_{\mathbb{Z}_N^\times}$ of Section 3.1.2, we assume that $\mathcal{R}$ sets a game tuple $(N, \alpha, \varrho_{k,m})$ in a way that the index N is always the same as the modulus N of the given RSA instance. Note that the assignment map α is chosen by selecting $\alpha(a)$ almost uniformly at random from QR_N for each $a \in A$. This is because we now adopt a sampling algorithm of $\{\mathbb{Z}_N^\times\}$ such that an element is chosen almost uniformly at random from QR_N. Note also that when $\mathcal{A}$ is a static adversary, $\mathcal{A}$ does not move to Equations queries phase. Eventually, the game completes with $\mathcal{A}$'s output: a "winning" witness $((\lambda^*, r^*), \psi^*)$ of the APF game, or "losing" symbol $\perp$. After the game, $\mathcal{R}$ would find a correct solution z for the given RSA instance (N, e, y) with nonnegligible probability ϵ_0.

We also force the reduction $\mathcal{R}$ to be *algebraic with respect to the group* QR_N *for any* $N \in \mathcal{N}$. For the assignment α and each $a \in A$, $\alpha(a)$ is of the form $\alpha(a) = y^d$ and the exponent d can be retrieved in a similar manner to the situation $\mathsf{SRSA} \leq \mathsf{SAPFG}_{\mathbb{Z}_N^\times}$, where y is an element of the given RSA instance.

We need an additional assumption for the key generator $\mathsf{KGen}_{\mathsf{RSA}}$ in Definition 2.9. We say that a pair $(N, e) \leftarrow \mathsf{KGen}_{\mathsf{RSA}}(1^k)$ is *good* if e is a prime in $\mathbb{Z}_{\varphi(N)}^\times$ and $e \geq 2^{\ell_{\exp}}$. In our main theorem, $\mathsf{KGen}_{\mathsf{RSA}}$ is forced to generate a good pair with probability at least $1/\tau_{\mathsf{Good}}$ for some polynomial $\tau_{\mathsf{Good}}(k)$ for any sufficiently large k. Note that this assumption for $\mathsf{KGen}_{\mathsf{RSA}}$ is not exceedingly strong. For instance, if e is (almost) uniformly distributed over $\mathbb{Z}_{\varphi(N)}^\times$ with respect to each specific modulus N, then our assumption holds as shown in the following Lemma 3.8.

Lemma 3.8
Let ℓ and $\ell_{\exp}$ be polynomials such that $\ell_{\exp} \leq \ell/2 - 1$. Let $N \in \mathbb{N}^{\mathrm{safe}}_{\mathsf{RSA}(\ell)}$. Then, $\Pr_{e \in_{\mathrm{U}} \mathbb{Z}_{\varphi(N)}^\times}\left[e \in \mathbb{P}_{<\varphi(N)} \wedge e \geq 2^{\ell_{\exp}}\right] > 1/\ell - \mathrm{negl}(k)$.

Proof. The prime number theorem [17, Theorem 8.1] states that $\pi(n)/(n/\ln n) \to 1$ as $n \to \infty$, where $\pi(n)$ denotes the number of primes less than or equal to n.

Thus, for any sufficiently large n, we have

$$\left|\frac{\pi(n)}{n/\ln n} - 1\right| < \epsilon,$$

for $\epsilon = 1 - \ln 2 > 0$. This implies that

$$\frac{n}{\log_2 n} = \frac{n \ln 2}{\ln n} = \frac{(1-\epsilon)n}{\ln n} < \pi(n) < \frac{(1+\epsilon)n}{\ln n} = \frac{1+\epsilon}{\ln 2}\frac{n}{\log_2 n} < \frac{2n}{\log_2 n}, \quad (3.6)$$

for any sufficiently large n. Recall that ℓ and $\ell_{\exp}$ are polynomials such that $\ell_{\exp} \leq \ell/2 - 1$. It follows from $N = PQ \in \mathbb{N}_{\mathsf{RSA}}^{\mathrm{safe}}$ that $\varphi(\varphi(N)) < \varphi(N)$. Because the binary length of P and Q are $\ell/2$, we have P and Q are in the interval $\left(2^{\ell/2-1}, 2^{\ell/2}\right)$. This implies that $2^{\ell-4} = (2^{\ell/2-2})^2 < (2^{\ell/2-1}-1)^2 < (P-1)(Q-1) = \varphi(N) < PQ < 2^\ell$. Since the order of $\mathbb{Z}_{\varphi(N)}^\times$ is $\varphi(\varphi(N))$, $e < \varphi(N) \leq 2^\ell$ for any $e \in \mathbb{Z}_{\varphi(N)}^\times$, and both $\varphi(N)$ and $2^{\ell_{\exp}}$ are not primes, we have

$$\begin{aligned}
\Pr_{e \in_{\mathrm{U}} \mathbb{Z}_{\varphi(N)}^\times}\left[e \in \mathbb{P}_{<\varphi(N)} \wedge e \geq 2^{\ell_{\exp}}\right] &= \frac{\pi(\varphi(N)-1) - \pi(2^{\ell_{\exp}}-1)}{\varphi(\varphi(N))} \\
&= \frac{\pi(\varphi(N)) - \pi(2^{\ell_{\exp}})}{\varphi(\varphi(N))} \\
&> \frac{\pi(\varphi(N)) - \pi(2^{\ell_{\exp}})}{\varphi(N)} \\
&> \frac{1}{\varphi(N)}\left(\frac{\varphi(N)}{\log_2 \varphi(N)} - \frac{2 \cdot 2^{\ell_{\exp}}}{\log_2 2^{\ell_{\exp}}}\right) \\
&= \frac{1}{\log_2 \varphi(N)} - \frac{2^{\ell_{\exp}+1}}{\varphi(N)\ell_{\exp}} \\
&> \frac{1}{\log_2 2^\ell} - \frac{2^{\ell_{\exp}+1}}{2^{\ell-4}\ell_{\exp}} \\
&\geq \frac{1}{\ell} - \frac{2^{\ell/2}}{2^{\ell-4}\ell_{\exp}} \\
&= \frac{1}{\ell} - \frac{1}{2^{\ell/2-4}\ell_{\exp}}.
\end{aligned}$$

Thus, it holds that $\Pr_{e \in_{\mathrm{U}} \mathbb{Z}_{\varphi(N)}^\times}\left[e \in \mathbb{P}_{<\varphi(N)} \wedge e \geq 2^{\ell_{\exp}}\right] > 1/\ell - \mathrm{negl}(k)$. □

We now ready to state our theorem for the adaptive pseudo-freeness of $\left\{\mathbb{Z}_N^\times\right\}$ under the RSA assumption.

Theorem 3.9
Assume that $\mathsf{KGen}_{\mathsf{RSA}}$ *outputs a good public key* (N, e) *with probability* $1/\tau_{\mathsf{Good}}$ *for sufficiently large* k*, where* τ_{Good} *is a polynomial in* k*. Let* $\varrho = \{\varrho_{k,m}\}_{k,m}$ *be any parametric distribution in the class* $\mathfrak{D}$*. If* $\mathsf{RSA} \leq \mathsf{APFG}_{\mathbb{Z}_N^\times,\varrho}$*, then the RSA assumption does not hold.*

Proof. Assume that $\mathsf{RSA} \leq \mathsf{APFG}_{\mathbb{Z}_N^\times,\varrho}$. Then, there exist a PPT algorithm $\mathcal{R}$ and a polynomial m such that $\mathcal{R}$ is algebraic with respect to QR_N for any $N \in \mathcal{N}$, and $\mathcal{R}$ breaks RSA in nonnegligible probability ϵ_0 with a black-box access to any PPT adversary $\mathcal{A}$ that wins the APF game of the family $\left\{\mathbb{Z}_N^\times\right\}$ with respect to $\varrho = \{\varrho_{k,m}\}_{k,m}$ in nonnegligible probability. This means that for any security parameter k, on a given RSA instance (N, e, y), $\mathcal{R}$ breaks RSA with at least nonnegligible probability ϵ_0, where the RSA public key (N, e) is generated by $\mathsf{KGen}_{\mathsf{RSA}}(1^k)$ and the RSA ciphertext y is uniformly distributed over QR_N.

Construction of Meta-Reduction $\mathcal{M}$. We shall construct a PPT algorithm that breaks RSA with no oracle access. We shall provide for the reduction $\mathcal{R}$ a simulator $\mathsf{Sim}_{\mathcal{A}}$ that plays the role of a winning APF adversary. In other words, from the $\mathcal{R}$'s viewpoint, $\mathsf{Sim}_{\mathcal{A}}$ looks like an adversary that really wins the APF game with nonnegligible probability. If $\mathsf{Sim}_{\mathcal{A}}$ behaves as the adversary, then $\mathcal{R}$ breaks RSA via playing the game with $\mathsf{Sim}_{\mathcal{A}}$. Thus, our meta-reduction $\mathcal{M}$ is constructed by involving $\mathcal{R}$ and $\mathsf{Sim}_{\mathcal{A}}$. If such a $\mathsf{Sim}_{\mathcal{A}}$ is provided, then $\mathcal{M}$ is constructed as in Fig. 3.3. Let (N, e, y) be a given "target" RSA instance. Note that N is a product of distinct safe primes $P = 2P' + 1$ and $Q = 2Q' + 1$ for some primes P' and Q', $e \in \mathbb{Z}_{\varphi(N)}^\times$ and $y \in_{\mathrm{U}} \mathrm{QR}_N$.

We note that $\mathsf{Sim}_{\mathcal{A}}$ will be constructed so that from the $\mathcal{R}$'s viewpoint, its outcome $((\lambda^*, r^*), \psi^*)$ in the step (c) of (M-4) is indeed a winning witness tuple on the game tuple $(N, \alpha, \varrho_{k,m})$, and hence $\mathcal{R}$ outputs a correct solution z^* with nonnegligible probability in (M-5). We also note that one can construct $\mathsf{Sim}_{\mathcal{A}}$ in a way that it makes no query, namely $\mathsf{Sim}_{\mathcal{A}}$ simulates a static adversary. Hence $\mathcal{M}$ involves no simulation of the Equations queries phase.

By using the following claims, we estimate the probability $\Pr[\mathsf{Succ}_{\mathcal{M}}]$ that $\mathcal{M}$ outputs a correct solution z for the target RSA instance (N, e, y) in (M-6). Note that the proofs of all claims described in this proof are given in Section 3.2.2.

Claim 3.10
$\mathcal{M}$ aborts in (M-2) with negligible probability in k.

Input. an RSA instance (N, e, y).
Output. the solution z of the given RSA instance (N, e, y).

(M-1) $\mathcal{M}$ sets an integer $E^* := H(r^*)$ by choosing a string $r^* \in_{\mathrm{U}} \{0,1\}^{\ell_{\mathrm{seed}}}$.

(M-2) $\mathcal{M}$ aborts if $E^* \leq 1$, or proceeds to the following step otherwise.

(M-3) $\mathcal{M}$ chooses a random coin ω of $\mathcal{R}$, and then executes $\mathcal{R}$ on the RSA instance $\left(N, e, y^{E^*}\right)$ with ω.

(M-4) When $\mathcal{R}$ invokes an APF adversary on a game tuple $(N, \alpha, \varrho_{k,m})$ of a group index N, an assignment $\alpha : A \to \mathbb{Z}_N^\times$ and the distribution family $\varrho_{k,m}$, $\mathcal{M}$ operates as follows:

  (a) $\mathcal{M}$ executes $\mathsf{Sim}_{\mathcal{A}}$ on the game tuple $(N, \alpha, \varrho_{k,m})$ with using the auxiliary tuple (e, y, r^*, ω);

  (b) $\mathcal{M}$ receives from $\mathsf{Sim}_{\mathcal{A}}$ a tuple $((\lambda^*, r^*), \psi^*)$ of a pair $(\lambda^*, r^*) \in \mathcal{E}_{k,m}$ for a nontrivial equation $\lambda^* = (E^*, \boldsymbol{s}^*)$ and a string r^* with a solution ψ^* of the interpreted equation λ^*_α; and

  (c) $\mathcal{M}$ hands the tuple $((\lambda^*, r^*), \psi^*)$ to $\mathcal{R}$ as an adversary's response.

(M-5) $\mathcal{M}$ receives a solution z^* for the RSA instance (N, e, y^{E^*}) from $\mathcal{R}$.

(M-6) $\mathcal{M}$ finds a solution z for the target RSA instance (N, e, y) by using z^* and Lemma 2.11, and halts with outputting z.

Figure 3.3: Configuration of $\mathcal{M}$

Claim 3.11
Assume that $\mathcal{M}$ does not abort in (M-2). For the target RSA instance (N, e, y) and the natural number E^* chosen in (M-1), y^{E^*} is distributed uniformly at random over QR_N.

Claim 3.12
Assume that the given RSA public key (N, e) is good, and $\mathcal{M}$ does not abort in (M-2). If $\mathcal{R}$ outputs a correct solution z^* for the queried RSA instance (N, e, y^{E^*}) in (M-5), $\mathcal{M}$ correctly finds a solution z for the target RSA instance (N, e, y) in (M-6).

By **Claim 3.10**, it is guaranteed that $\mathcal{M}$ aborts in negligible probability. We therefore consider the case where $\mathcal{M}$ does not abort in (M-2). Then, **Claim 3.11** implies that the distribution of the RSA instance (N, e, y^{E^*}) queried by $\mathcal{M}$ in (M-3) is identical to that of the RSA instance (N, e, y) such that $(N, e) \leftarrow \mathsf{KGen}_{\mathsf{RSA}}(1^k)$ and $y \in_{\mathrm{U}} \mathrm{QR}_N$. Further, for the game tuple $(N, \alpha, \varrho_{k,m})$ submitted from $\mathcal{R}$, the simulator $\mathsf{Sim}_{\mathcal{A}}$ returns a winning witness pair $((\lambda^*, r^*), \psi^*)$. Therefore $\mathcal{R}$ would output a correct solution z^* of the queried RSA instance (N, e, y^{E^*}) with at least nonnegligible probability ϵ_0. It follows from **Claim 3.12** that $\mathcal{M}$ outputs a solution z for the target RSA instance (N, e, y) with probability at least ϵ_0 provided that the RSA public key (N, e) is good. Totally, the success probability of $\mathcal{M}$ under the assumption of the good RSA public key (N, e) is evaluated by

$$\Pr\left[\mathsf{Succ}_{\mathcal{M}} \mid (N, e) \text{ is good}\right] \geq \epsilon_0 - \mathrm{negl}(k).$$

Since we now assume that for the polynomial τ_{Good}, $\mathsf{KGen}_{\mathsf{RSA}}$ outputs a good public key (N, e) with probability $1/\tau_{\mathsf{Good}}$ for sufficiently large k, we have

$$\begin{aligned}\Pr\left[\mathsf{Succ}_{\mathcal{M}}\right] &\geq \Pr\left[\mathsf{Succ}_{\mathcal{M}} \wedge (N, e) \text{ is good}\right] \\ &= \Pr\left[(N, e) \text{ is good}\right] \Pr\left[\mathsf{Succ}_{\mathcal{M}} \mid (N, e) \text{ is good}\right] \\ &\geq \frac{1}{\tau_{\mathsf{Good}}}\epsilon_0 - \mathrm{negl}(k).\end{aligned}$$

Thus, $\mathcal{M}$ can break RSA with nonnegligible probability.

Construction of $\mathsf{Sim}_{\mathcal{A}}$. In order to construct the algorithm $\mathcal{M}$, it suffices to construct the simulator $\mathsf{Sim}_{\mathcal{A}}$. Since $\mathcal{R}$ is algebraic with respect to QR_N for any N,

Input. the game tuple $(N, \alpha, \varrho_{k,m})$ with the auxiliary tuple (e, y, r^*, ω), as in (a) of the step (M-4).
Output. a tuple $((\lambda^*, r^*), \psi^*)$.

(A-1) Set $E^* := H(r^*)$, and then for each index $i \in [1, m]$, retrieve an exponent d_i of the element $\alpha(a_i) \in \mathrm{QR}_N$ such that $\alpha(a_i) = \left(y^{E^*}\right)^{d_i}$ by executing $\mathsf{Extract}(N, e, y^{E^*}, \alpha(a_i), \omega)$.

(A-2) Choose a random parameter $M^* \in \{0,1\}^{\ell_{\mathrm{msg}}}$, choose exponents $s_2^*, \ldots, s_m^*$ according to the distribution $\varrho_{k,m}(M^*)$, and then set $\boldsymbol{s}^* := (1, s_2^*, \ldots, s_m^*)$.

(A-3) Set $\lambda^* := (E^*, \boldsymbol{s}^*)$ and $\psi^* := y^{\sum_{i=1}^m d_i s_i^*}$.

(A-4) Halt with outputting the tuple $((\lambda^*, r^*), \psi^*)$.

Figure 3.4: Configuration of $\mathsf{Sim}_{\mathcal{A}}$

there exists a polynomial-time algorithm $\mathsf{Extract}$ that on a tuple $(N, e, y^{E^*}, g, \omega)$, where g is a target element in QR_N that is produced in the execution of $\mathcal{R}$ on the input (N, e, y^{E^*}) given from $\mathcal{M}$ with the random coin ω, returns an exponent d such that $g = \left(y^{E^*}\right)^d$. We involve $\mathsf{Extract}$ in the construction of $\mathsf{Sim}_{\mathcal{A}}$. The algorithm $\mathsf{Sim}_{\mathcal{A}}$ is constructed as in Fig 3.4.

We now show that $\mathsf{Sim}_{\mathcal{A}}$ is a PPT simulator that wins the APF game with probability 1 on each game tuple $(N, \alpha, \varrho_{k,m})$ given from $\mathcal{R}$. Since $\mathsf{Extract}$ is a polynomial-time algorithm, $\mathsf{Sim}_{\mathcal{A}}$ can be run in polynomial-time. For the tuple $((\lambda^*, r^*), \psi^*)$, the following claims hold.

Claim 3.13
The tuple (λ^*, r^*) is chosen according to the distribution $\varrho_{k,m}(M^*)$, and the equation $\lambda^* = (E^*, \boldsymbol{s}^*)$ is nontrivial.

Claim 3.14
ψ^* is a correct solution of the interpreted equation λ_α^*.

Thus, $\mathsf{Sim}_{\mathcal{A}}$ always wins the APF game of the RSA group family $\left\{\mathbb{Z}_N^\times\right\}$ with respect to the parametric distribution ϱ when $\mathcal{R}$ plays the role of a challenger. □

Note that in (A-2) of the description for $\mathsf{Sim}_{\mathcal{A}}$, it suffices for the proof of Theorem 3.9 itself that $\mathsf{Sim}_{\mathcal{A}}$ chooses any random integers $s_2^*, \ldots, s_m^*$. However, we

have constructed $\mathsf{Sim}_{\mathcal{A}}$ in a way that for any parameter M, the distribution of the outcome (λ^*, r^*) from Sim_A is the same as the distribution $\varrho_{k,m}(M)$. We require such a property for the applications of Theorem 3.9 which will be presented in Section 3.2.3. Moreover, we should note that in the proof of Theorem 3.9, $\mathsf{Sim}_{\mathcal{A}}$ has been constructed so that it is static. Due to this, we will be able to show the impossibility results for KOA to the signature schemes, instead of CMA, in Section 3.2.3.

3.2.2 Proofs of the Claims in Theorem 3.9

We now show the claims employed in Theorem 3.9.

Proof of Claim 3.10. It follows from Lemma 3.7 that for the collision-resistant hash function $H : \{0,1\}^{\ell_{\mathrm{seed}}} \to \left[0, 2^{\ell_{\mathrm{exp}}} - 1\right]$,

$$\begin{aligned}\Pr_{r^* \in_{\mathrm{U}} \{0,1\}^{\ell_{\mathrm{seed}}}} \left[E^* = H(r^*) \leq 1\right] &= \Pr_{r^* \in_{\mathrm{U}} \{0,1\}^{\ell_{\mathrm{seed}}}} \left[E^* = 0 \vee E^* = 1\right] \\ &= \Pr_{r^* \in_{\mathrm{U}} \{0,1\}^{\ell_{\mathrm{seed}}}} \left[E^* = 0\right] + \Pr_{r^* \in_{\mathrm{U}} \{0,1\}^{\ell_{\mathrm{seed}}}} \left[E^* = 1\right] \\ &= \mathrm{negl}(k).\end{aligned}$$

Thus, $\mathcal{M}$ aborts in (M-2) with negligible probability. □

Proof of Claim 3.11. We now assume that $E^* \in \mathbb{Z}_{P'Q'}^\times$. Then, we consider for the given RSA modulus N, a map $\mathcal{B}_{N,E^*}$ that maps each element $y \in \mathrm{QR}_N$ to $y^{E^*} \bmod N \in \mathrm{QR}_N$. It follows from $E^* \in \mathbb{Z}_{P'Q'}^\times$ and $\mathrm{ord}\,(\mathrm{QR}_N) = P'Q'$ that $\mathcal{B}_{N,E^*}$ is bijective. Since the element y is chosen uniformly at random from QR_N, y^{E^*} is uniformly distributed over QR_N.

We now show that $E^* = H(r^*) \in \left[0, 2^{\ell_{\mathrm{exp}}} - 1\right]$ set in (M-1) is in $\mathbb{Z}_{P'Q'}^\times$. Since we assume that $\mathcal{M}$ does not abort in (M-2), E^* is in the set $\left(1, 2^{\ell_{\mathrm{exp}}}\right)$. On the other hand, for any $N = PQ \in \mathbb{N}_{\mathsf{RSA}(\ell)}^{\mathrm{safe}}$, it follows from $P = 2P' + 1, Q = 2Q' + 1 \in \left(2^{\ell/2-1}, 2^\ell\right)$ that both P' and Q' are greater than $2^{\ell/2-2} - 1$. Therefore, we have $1 < E^* \leq 2^{\ell_{\mathrm{exp}}} - 1 \leq 2^{\ell/2-2} - 1 < P'$ and $1 < E^* < Q'$ by the inequality $\ell_{\mathrm{exp}} \leq \ell/2 - 2$. This implies that for the exponent E^* chosen in (M-1) and the primes P' and Q', $P' \nmid E^*$ and $Q' \nmid E^*$. Thus, we have $E^* \in \mathbb{Z}_{P'Q'}^\times$. □

Proof of Claim 3.12. Assume that the given RSA public key (N, e) is good. Then, e is a prime and $e \geq 2^{\ell_{\mathrm{exp}}}$. On the other hand, the assumption that $\mathcal{M}$ does not abort implies that $E^* > 1$. It therefore follows from $E^* = H(r^*) < 2^{\ell_{\mathrm{exp}}}$ that

$1 < E^* < 2^{\ell_{\exp}} \leq e$. Since e is prime, we have $\gcd(e, E^*) = 1$. Thus, if $\mathcal{R}$ outputs a correct solution z^* for the queried RSA instance (N, e, y^{E^*}), $\mathcal{M}$ correctly finds a solution z for the target RSA instance (N, e, y) by Lemma 2.11. □

Proof of Claim 3.13. Recall that the string r^* and the exponent E^* are chosen in (M-1) so that $r^* \in_{\mathrm{U}} \{0,1\}^{\ell_{\mathrm{seed}}}$ and $E^* = H(r^*) \in \left[0, 2^{\ell_{\exp}} - 1\right]$. In (A-2), $s_1^*, s_2^*, \ldots, s_m^*$ are chosen according to the distribution $\varrho_{k,m}(M^*)$. Therefore, the tuple (λ^*, r^*) is distributed over the set $\mathcal{E}_{k,m}$ according to the distribution $\varrho_{k,m}(M^*)$.

We now show that the equation λ^* is nontrivial. Since $\mathsf{Sim}_{\mathcal{A}}$ makes no query, the nontriviality is exactly equivalent to that in the case of the static pseudo-free group. Hence, the nontriviality is determined by Lemma 2.1. Recall that the exponent E^* given to $\mathsf{Sim}_{\mathcal{A}}$ is strictly greater than 1 and the integer s_1^* chosen in (A-2) is 1, This implies that $E^* \nmid s_1^*$. It follows from Lemma 2.1 that the equation λ^* is nontrivial. □

Proof of Claim 3.14. It follows from $\alpha(a_i) = (y^{E^*})^{d_i}$ for each $i \in [1, m]$ and $\psi^* = y^{\sum_{i=1}^m d_i s_i^*}$ that for the equation $\lambda^* = (E^*, (s_1^*, s_2^*, \ldots, s_m^*))$, in $\mathbb{Z}_N^\times$,

$$(\psi^*)^{E^*} = \left(y^{\sum_{i=1}^m d_i s_i^*}\right)^{E^*} = \prod_{i=1}^m \left(y^{E^* d_i}\right)^{s_i^*} = \prod_{i=1}^m \alpha(a_i)^{s_i^*}.$$

ψ^* is therefore a correct solution of the interpreted equation λ^*. □

3.2.3 Impossibility for the SRSA-Based Signature Schemes

In this section, we give a negative circumstantial evidence that several SRSA-based signature schemes cannot be proven to be sEUF-CMA, and even EUF-KOA, from RSA assumption. We first describe a generic construction proposed by Catalano-Fiore-Warinschi [13] of a secure signature scheme based on an adaptive pseudo-free group. Next, we show that several SRSA-based signature schemes that are yielded from their construction cannot be proven to be EUF-KOA from RSA assumption via algebraic reductions, as long as the RSA assumption holds.

The Catalano-Fiore-Warinschi Signature Schemes. Catalano, Fiore and Warinschi [13] proposed a generic construction of a signature scheme based on an adaptive pseudo-free group with respect to a parametric distribution ϱ, and formalized a class $\mathfrak{D}^{\mathsf{Sig}}$ of parametric distributions such that the signatures yielded

from their construction by using the adaptive pseudo-free group with respect to $\varrho \in \mathfrak{D}^{\mathsf{Sig}}$ is sEUF-CMA.

We now describe their generic construction of a signature scheme. Let $\mathcal{E} = \{\mathcal{E}_{k,m}\}$ be any family of sets for pairs (λ, r), and let $\varrho = \{\varrho_{k,m}\}$ be any parametric distribution. We require that for each security parameter k and each polynomial m, $\varrho_{k,m} = \{\varrho_{k,m}(M)\}$ is a family of probabilistic distributions $\varrho_{k,m}(M)$ over the set $\mathcal{E}_{k,m}$ such that for any parameter $M \in \{0,1\}^{\ell_{\mathrm{msg}}}$,

(i) $\varrho_{k,m}(M)$ is polynomial-time samplable;

(ii) the description of $\varrho_{k,m}(M)$ can be obtained in polynomial-time in k; and

(iii) the membership of the support $\mathsf{Supp}\left(\varrho_{k,m}(M)\right)$ of the distribution $\varrho_{k,m}(M)$ can be verified in polynomial-time in k.

Let $\mathcal{G} = \{\mathbb{G}_N\}_{N \in \mathcal{N}}$ be a computational group family. Then, the signature scheme $\mathsf{PFSig}_{\mathcal{G},\varrho} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Verify})$ is constructed as in Fig. 3.5, where the message space $\mathcal{M}$ of $\mathsf{PFSig}_{\mathcal{G},\varrho}$ is $\{0,1\}^{\ell_{\mathrm{msg}}}$. Note that a public key $(N, \alpha, \varrho_{k,m})$ generated by $\mathsf{KGen}(1^k, m)$ is chosen as in Setup phase of the APF game defined in Section 2.3, and a pair (M, σ) of a message M and a signature $\sigma = (\lambda, r, \psi)$ is naturally viewed as that of a queried parameter M and an answer (λ, r, ψ) chosen in $\mathsf{Equations\ queries}$ phase.

For the security of the signature depicted in Fig. 3.5, Catalano, Fiore and Warinschi [13] defined a class $\mathfrak{D}^{\mathsf{Sig}}$ of parametric distributions $\varrho^{\mathsf{Sig}} = \left\{\varrho^{\mathsf{Sig}}_{k,m}\right\}$ so that for any k, m and M, the form of a pair (λ, r) chosen according to $\varrho^{\mathsf{Sig}}_{k,m}(M)$ is restricted to being in a set $\mathcal{E}^{\mathsf{Sig}}_{k,m}$ of pairs (λ, r) of an equation and a string. We now describe their set $\mathcal{E}^{\mathsf{Sig}}_{k,m}$ of specific pairs (λ, r). In a similar fashion to our class $\mathfrak{D}$ defined in Section 3.2.1, we fix any nonconstant polynomial ℓ_{msg}, $\ell_{\exp}$ and ℓ_{seed}, and any division-intractable hash function $H : \{0,1\}^{\ell_{\mathrm{seed}}} \to \left[0, 2^{\ell_{\exp}} - 1\right]$. For each k and m, let $\mathcal{E}^{\mathsf{Sig}}_{k,m}$ be the set of all pairs (λ, r) of an equation $\lambda = (E, (s_1, \ldots, s_m))$ and a string $r \in \{0,1\}^{\ell_{\mathrm{seed}}}$ such that $E = H(r)$, $s_1 = 1$ and $s_i \in [0, E-1]$ for each $i \in [2, m]$. Then, $\mathfrak{D}^{\mathsf{Sig}}$ is a class of all parametric distributions $\varrho^{\mathsf{Sig}} = \left\{\varrho^{\mathsf{Sig}}_{k,m}\right\}_{k,m}$ such that for each security parameter k and each polynomial m, $\varrho^{\mathsf{Sig}}_{k,m} = \left\{\varrho^{\mathsf{Sig}}_{k,m}(M)\right\}$ is a family of probabilistic distributions $\varrho^{\mathsf{Sig}}_{k,m}(M)$ over $\mathcal{E}^{\mathsf{Sig}}_{k,m}$ satisfying the conditions (i), (ii) and (iii). We also require the additional

Key Generator KGen**.** On input $(1^k, m)$, KGen works as follows:

(1) choose a random group index $N \in_{\mathrm{U}} \mathcal{N}(k)$ together with finding the order $\mathrm{ord}(\mathbb{G}_N)$ of the group $\mathbb{G}_N$;

(2) specify an assignment $\alpha : A \to \mathbb{G}_N$ at random according to the designated sampling algorithm for each $a \in A$; and

(3) output a public key $pk := (N, \alpha, \varrho_{k,m})$ and a secret key $sk := \mathrm{ord}(\mathbb{G}_N)$.

Signing Algorithm Sign**.** On input (sk, pk, M), Sign issues a signature σ on the message M in the following way:

(1) choose $(\lambda, r) \in \mathcal{E}_{k,m}$ according to the distribution $\varrho_{k,m}(M)$;

(2) find a solution $\psi \in \mathbb{G}_N$ of the interpreted equation λ_α by using $sk = \mathrm{ord}(\mathbb{G}_N)$; and

(3) output a signature $\sigma = (\lambda, r, \psi)$ on the message M.

Verification Algorithm Verify On input (pk, M, σ), Verify outputs 1 if $(\lambda, r) \in \mathsf{Supp}\left(\varrho_{k,m}(M)\right)$ and ψ is actually a solution of λ_α, or 0 otherwise.

Figure 3.5: Construction of $\mathsf{PFSig}_{\mathcal{G},\varrho}$

assumption: for any PPT adversary $\mathcal{A}$, any k and any m,

$$\Pr\left[\begin{array}{c} M_1 \neq M_2 \\ \wedge(\lambda, r) \in \mathsf{Supp}\left(\varrho^{\mathsf{Sig}}_{k,m}(M_1)\right) \\ \wedge(\lambda, r) \in \mathsf{Supp}\left(\varrho^{\mathsf{Sig}}_{k,m}(M_2)\right) \end{array} : \begin{array}{l} (M_1, M_2, (\lambda, r)) \\ \leftarrow \mathcal{A}\left(\varrho^{\mathsf{Sig}}_{k,m}\right) \end{array}\right] = \mathrm{negl}(k),$$

where the probability is taken over the coin flips of $\mathcal{A}$. This means that one cannot find a signature σ for two distinct messages. Catalano, Fiore and Warinschi showed in [13, Theorem 1] that for any parametric distribution $\varrho^{\mathsf{Sig}} \in \mathfrak{D}^{\mathsf{Sig}}$, $\mathsf{PFSig}_{\mathcal{G},\varrho^{\mathsf{Sig}}}$ is sEUF-CMA provided that $\mathcal{G}$ is adaptive pseudo-free with respect to ϱ^{Sig}.

We now show the converse direction of Theorem 1 in [13]. Namely, $\mathcal{G}$ is adaptive pseudo-free with respect to $\varrho^{\mathsf{Sig}} \in \mathfrak{D}^{\mathsf{Sig}}$ if $\mathsf{PFSig}_{\mathcal{G},\varrho^{\mathsf{Sig}}}$ is sEUF-CMA. Note that the converse direction is proven for a larger class than $\mathfrak{D}^{\mathsf{Sig}}$. We consider any parametric distribution $\varrho = \{\varrho_{k,m}\}$ such that it satisfies the conditions (i), (ii)

Input. a public key $(N, \alpha, \varrho_{k,m})$ of $\mathsf{PFSig}_{\mathcal{G},\varrho}$.
Output. a tuple (M^*, σ^*), where $\sigma^* = (\lambda^*, r^*, \psi^*)$.

(R-1) $\mathcal{R}$ invokes the APF adversary $\mathcal{A}$ on input $(N, \alpha, \varrho_{k,m})$.

(R-2) When t-th parameter $M_t \in \{0,1\}^{\ell_{\mathrm{msg}}}$ is submitted from $\mathcal{A}$, $\mathcal{R}$ hands M_t to the signing oracle. Note that the parameter M_t is regarded as a message. Then, $\mathcal{R}$ receives a signature $\sigma_t = \left(\lambda^{(t)}, r_t, \psi_t\right)$ on the message M_t, and returns the tuple $\left((\lambda^{(t)}, r_t), \psi_t\right)$ to $\mathcal{A}$.

(R-3) Eventually, $\mathcal{A}$ outputs a tuple $(M^*, ((\lambda^*, r^*), \psi^*))$. Then, $\mathcal{R}$ sets $\sigma^* := (\lambda^*, r^*, \psi^*)$, and outputs (M^*, σ^*).

Figure 3.6: Configuration of an sEF-CMA adversary $\mathcal{R}$

and (iii) explained above, but for any k, m and M, the form of a pair (λ, r) chosen according to $\varrho_{k,m}(M)$ is not restricted. Here, we need the following modification of the APF game: a portion (λ^*, r^*) of an outcome $((\lambda^*, r^*), \psi^*)$ by an APF adversary $\mathcal{A}$ is chosen according to the distribution $\varrho_{k,m}(M^*)$ for some parameter $M^* \in \{0,1\}^{\ell_{\mathrm{msg}}}$. Moreover, $\mathcal{A}$ outputs the parameter M^* together with the tuple $((\lambda^*, r^*), \psi^*)$.

Lemma 3.15
Let ϱ be any parametric distribution satisfying the conditions (i), (ii) and (iii), and let $\mathcal{G} = \{\mathbb{G}_N\}_{N \in \mathcal{N}}$ be any computational group family. If $\mathsf{PFSig}_{\mathcal{G},\varrho}$ is sEUF-CMA, then $\mathcal{G}$ is adaptive pseudo-free with respect to ϱ.

Proof. Assume that $\mathcal{G} = \{\mathbb{G}_N\}_{N \in \mathcal{N}}$ is not adaptive pseudo-free with respect to ϱ. Then, there exist a PPT adversary $\mathcal{A}$ and polynomials m and q such that $\mathcal{A}$ wins the APF game with respect to ϱ with nonnegligible probability ϵ, and $\mathcal{A}$ makes q queries in Equations queries phase of the APF game. We shall construct a PPT adversary $\mathcal{R}$ that wins sEF-CMA game of $\mathsf{PFSig}_{\mathcal{G},\varrho}$ with nonnegligible probability. In our construction, $\mathcal{R}$ plays the APF game with $\mathcal{A}$ in which $\mathcal{R}$ plays the role of an APF challenger. The adversary $\mathcal{R}$ is constructed as in Fig. 3.6.

We now show the correctness of $\mathcal{R}$. Let $(N, \alpha, \varrho_{k,m})$ be a public key generated by $\mathsf{KGen}(1^k, m)$. In particular, the assignment $\alpha : A \to \mathbb{G}_N$ is specified by selecting $\alpha(a) \in \mathbb{G}_N$ according to the designated sampling algorithm for

each $a \in A$. This implies that the public key $(N, \alpha, \varrho_{k,m})$ coincides with the game tuple of the APF game. Moreover, for each t-th query M_t, the signature $\sigma_t = \left(\lambda^{(t)}, r_t, \psi_t\right)$ on M_t given by the signing oracle satisfies that the pair $(\lambda^{(t)}, r_t)$ is distributed according to the distribution $\varrho_{k,m}(M_t)$, and $\psi_t \in \mathbb{G}_N$ is a solution for the interpreted equation $\lambda_\alpha^{(t)}$. This implies that such a signature $\sigma_t = \left(\lambda^{(t)}, r_t, \psi_t\right)$ on M_t can be translated into an answer for the queried parameter M_t from $\mathcal{A}$ in Equations queries phase of the APF game. Therefore, $\mathcal{R}$ actually simulates the role of the APF challenger. Thus, the APF adversary $\mathcal{A}$ outputs the following tuple $(M^*, ((\lambda^*, r^*), \psi^*))$ with nonnegligible probability ϵ. For such a tuple $(M^*, ((\lambda^*, r^*), \psi^*))$, it holds that (A) $(\lambda^*, r^*) \in \mathcal{E}_{k,m}$, (B) λ is nontrivial with respect to the equation set $\left\{\lambda^{(1)}, \lambda^{(2)}, \ldots, \lambda^{(q)}\right\}$, and (C) ψ^* is a solution for the interpreted equation λ_α^* of λ^*. In particular, we have that (A') $(\lambda^*, r^*) \in \mathsf{Supp}\left(\varrho_{k,m}(M^*)\right)$, because we have modified the APF game as mentioned above. It follows from the following claim that $\mathcal{R}$ wins the sEF-CMA game of $\mathsf{PFSig}_{\mathcal{G},\varrho}$ with probability at least ϵ.

Claim 3.16
Assume that the tuple $(M^*, ((\lambda^*, r^*), \psi^*))$ satisfies the conditions (A'), (B) and (C). Then, it holds that for the pair (M^*, σ^*), $(M^*, \sigma^*) \notin \{(M_t, \sigma_t)\}_{t=1}^q$ and $\mathsf{Verify}(pk, M^*, \sigma^*) = 1$, where $\sigma^* = (\lambda^*, r^*, \psi^*)$.

Proof. We first show that the condition (B) implies that $(M^*, \sigma^*) \notin \{(M_t, \sigma_t)\}_{t=1}^q$, namely for any $t \in [1, q]$, we have

$$(M^*, ((E^*, s_1^*, s_2^*, \ldots, s_m^*), r^*, \psi^*)) \neq (M_t, ((E_t, s_{t,1}, s_{t,2}, \ldots, s_{t,m}), r_t, \psi_t)). \quad (3.7)$$

It follows from (B) and Lemma 2.5 that for any $t \in [1, q]$,

$$(E^*, s_1^*, s_2^*, \ldots, s_m^*) \neq (E_t, s_{t,1}, s_{t,2}, \ldots, s_{t,m}).$$

Hence, Eq. (3.7) holds for any $t \in [1, q]$. Thus, we have $(M^*, \sigma^*) \notin \{(M_t, \sigma_t)\}_{t=1}^q$.

We next show that $\mathsf{Verify}(pk, M^*, \sigma^*) = 1$. the condition (A') implies that $(\lambda^*, r^*) \in \mathsf{Supp}\left(\varrho_{k,m}(M^*)\right)$. It follows from the condition (C) that ψ^* is a solution for λ_α^*. These imply that $\mathsf{Verify}(pk, M^*, \sigma^*) = 1$. □

Thus $\mathsf{PFSig}_{\mathcal{G},\varrho}$ is not sEUF-CMA. □

It should be noted that in Lemma 3.15, if the winning APF adversary $\mathcal{A}$ of $\mathcal{G}$ with respect to ϱ is static, then $\mathcal{R}$ depicted in Fig 3.6 can be regarded as a

winning EF-KOA adversary of the signature $\mathsf{PFSig}_{\mathcal{G},\varrho}$. This is because $\mathcal{A}$, and hence $\mathcal{R}$, makes no query in (R-2). Thus, it holds that $\mathcal{G}$ is adaptive pseudo-free with respect to ϱ against a static adversary provided that $\mathsf{PFSig}_{\mathcal{G},\varrho}$ is EUF-KOA.

Results on the SRSA-Based Signatures. We give an impossibility result on several SRSA-based signatures. We focus on the Camenisch-Lysyanskaya (CL, for short) signature [11], the Cramer-Shoup (CS, for short) signature [14], the Fischlin signature [18], the Gennaro-Halevi-Rabin (GHR, for short) signature [22], the Hofheinz-Kiltz (HK, for short) signature [27], and the Zhu signature [48, 49]. In order to prove the impossibility result, we apply Theorem 3.9 and Lemma 3.15 to the RSA group family $\left\{\mathbb{Z}_N^\times\right\}$ with a parametric distribution ϱ belonging to our class $\mathfrak{D}$, and show that $\mathsf{PFSig}_{\mathbb{Z}_N^\times,\varrho}$ may not be proven to be sEUF-CMA, and even EUF-KOA. Note that Theorem 3.9 holds even when the modification of the APF game described just before Lemma 3.15 is required on $\mathsf{Sim}_{\mathcal{A}}$ as follows: in (A-4), $\mathsf{Sim}_{\mathcal{A}}$ outputs the parameter M^* chosen in (A-2) together with the witness $((\lambda^*, r^*), \psi^*)$. By this modification of $\mathsf{Sim}_{\mathcal{A}}$, the following lemma holds.

Lemma 3.17
Let $\varrho = \{\varrho_{k,m}\}$ be a parametric distribution such that $\varrho \in \mathfrak{D}$ and for any k, m and M, the membership of $\mathsf{Supp}\left(\varrho_{k,m}(M)\right)$ can be verified in polynomial-time in k. Assume that the RSA key generator $\mathsf{KGen}_{\mathsf{RSA}}$ outputs a good public key (N, e) with probability $1/\tau_{\mathsf{Good}}$ for sufficiently large k, where τ_{Good} is a polynomial in k. If $\mathsf{PFSig}_{\mathbb{Z}_N^\times,\varrho}$ can be proven to be sEUF-CMA (EUF-KOA, resp.) from the RSA assumption via algebraic reductions, then the RSA assumption does not hold.

Proof. Assume that $\mathsf{PFSig}_{\mathbb{Z}_N^\times,\varrho}$ is proven to be sEUF-CMA from the RSA assumption via algebraic reductions. It follows from Lemma 3.15 that $\mathsf{RSA} \leq \mathsf{APFG}_{\mathbb{Z}_N^\times,\varrho}$. Since $\varrho \in \mathfrak{D}$, Theorem 3.9 implies that the RSA assumption does not hold. Recall that Theorem 3.9 holds even when an APF adversary is static. Thus, it holds that if $\mathsf{PFSig}_{\mathbb{Z}_N^\times,\varrho}$ can be proven to be EUF-KOA from the RSA assumption via algebraic reductions, then the RSA assumption does not hold. □

By employing Lemma 3.17, we give an impossibility result on the SRSA-based signatures. Catalano, Fiore and Warinschi [13] constructed a parametric distribution $\varrho^{\mathsf{CL}} = \left\{\varrho^{\mathsf{CL}}_{k,m}\right\}_{k,m}$ (ϱ^{CS}, ϱ^{Fis}, ϱ^{GHR}, ϱ^{HK} and ϱ^{Zhu}, resp.) so that $\mathsf{PFSig}_{\mathbb{Z}_N^\times,\varrho^{\mathsf{CL}}}$ coincides with the CL (CS, Fischlin, GHR, HK and Zhu, resp.) scheme. We now describe the parametric distributions ϱ^{CL}, ϱ^{CS}, ϱ^{Fis}, ϱ^{GHR}, ϱ^{HK}

and ϱ^{Zhu}, respectively. Let $H_{\mathsf{PRIMES}} : \{0,1\}^{\ell_{\mathrm{seed}}} \to \left(2^{\ell_{\mathrm{exp}}-1}, 2^{\ell_{\mathrm{exp}}}\right)$ be a division-intractable prime-valued hash function, and let H' be an $(\ell_{\mathrm{exp}}-1)$-bit collision-resistant hash function. Let $M \in \{0,1\}^{\ell_{\mathrm{msg}}}$ denote a parameter. Then, all of the parametric distributions are depicted from Fig. 3.7 to Fig. 3.12.

It follows from Proposition 2.8 that $\varrho^{\mathsf{CL}}, \varrho^{\mathsf{Fis}}, \varrho^{\mathsf{HK}}, \varrho^{\mathsf{Zhu}} \in \mathfrak{D}$. This is because a pair (λ, r) chosen according to each of these parametric distributions satisfies that r is uniformly distributed over $\{0,1\}^{\ell_{\mathrm{seed}}}$, E is computed by using the division-intractable (and then collision-resistant) hash function H_{PRIMES}, $s_1 = 1$ and $s_2, \ldots, s_m \in \mathbb{Z}$. Therefore, one can apply Lemma 3.17 to $\mathsf{PFSig}_{\mathbb{Z}_N^\times, \varrho^{\mathsf{CL}}}$, $\mathsf{PFSig}_{\mathbb{Z}_N^\times, \varrho^{\mathsf{Fis}}}$, $\mathsf{PFSig}_{\mathbb{Z}_N^\times, \varrho^{\mathsf{HK}}}$, $\mathsf{PFSig}_{\mathbb{Z}_N^\times, \varrho^{\mathsf{Zhu}}}$, namely the CL signature, the Fischlin signature, the HK signature, the Zhu signature, respectively. Note that Lemma 3.17 can be also applied to $\mathsf{PFSig}_{\mathbb{Z}_N^\times, \varrho^{\mathsf{CS}}}$ (the CS signature) by the following modification for $\mathsf{Sim}_{\mathcal{A}}$. $\mathsf{Sim}_{\mathcal{A}}$ chooses an element $R^* \in_{\mathrm{U}} \mathrm{QR}_N$ in (A-2) to select s_2 according to the distribution $\varrho^{\mathsf{CS}}_{k,m,(N,u_2,E')}(M^*)$, and then it outputs R^* together with the witness $(M^*, ((\lambda^*, r^*), \psi^*))$ in (A-4).

Corollary 3.18
Assume that $\ell_{\mathrm{exp}} \leq \ell/2 - 2$. Assume also that $\mathsf{KGen}_{\mathsf{RSA}}$ outputs a good public key (N, e) with probability $1/\tau_{\mathsf{Good}}$ for sufficiently large k. The CL scheme, the CS scheme, the Fischlin scheme, the HK scheme and the Zhu scheme cannot be proven to be EUF-KOA under the RSA assumption via algebraic reductions, as long as the RSA assumption holds.

The parametric distribution ϱ^{GHR} does not belong to $\mathfrak{D}$. This is because for the parametric distribution ϱ^{GHR}, the exponent E is set as $E := H(M)$ for a given parameter M, and M may not be uniformly distributed over $\{0,1\}^{\ell_{\mathrm{seed}}}$. However, one can show that the adaptive pseudo-freeness of $\left\{\mathbb{Z}_N^\times\right\}$ with respect to ϱ^{GHR} cannot be proven under the RSA assumption, as in Theorem 3.9. The proof is almost the same as that of Theorem 3.9 except that the simulator $\mathsf{Sim}_{\mathcal{A}}$ sets $M^* := r^*$ in (A-2) instead of randomly choosing M^*. Thus, the following corollary holds in a similar manner to Corollary 3.18.

Corollary 3.19
Assume that $\ell_{\mathrm{seed}} = \ell_{\mathrm{msg}}$ and $\ell_{\mathrm{exp}} \leq \ell/2 - 2$. Assume also that $\mathsf{KGen}_{\mathsf{RSA}}$ outputs a good public key (N, e) with probability $1/\tau_{\mathsf{Good}}$ for sufficiently large k. The GHR scheme cannot be proven to be EUF-KOA under the RSA assumption via algebraic reductions, as long as the RSA assumption holds.

Let $\ell_{\mathrm{msg}} + 2 \leq \ell_{\exp}$, and let ℓ_{parm} be a polynomial in k.

Key Generator KGen**.** On input 1^k, KGen works as follows:

(1) choose an RSA modulus $N = PQ \in \mathbb{N}^{\mathrm{safe}}_{\mathsf{RSA}(\ell)}$, and then choose elements $u_1, u_2, u_3 \in_{\mathrm{U}} \mathrm{QR}_N$; and

(2) output a public key $pk = (N, u_1, u_2, u_3)$ and a secret key (P, Q).

Signing Algorithm Sign**.** On input (sk, pk, M), Sign issues a signature σ on the message M in the following way:

(1) choose a random $\ell_{\exp}$-bit prime E and a random $(\ell+\ell_{\mathrm{msg}}+\ell_{\mathrm{parm}})$-bit integer s, and then find an element ψ so that $\psi^E = u_1 u_2^s u_3^M$; and

(2) output a signature $\sigma = (E, s, \psi)$.

Verification Algorithm Verify**.** On input (pk, M, σ), Verify outputs 1 if $E \in \left(2^{\ell_{\exp}-1}, 2^{\ell_{\exp}}\right)$, and $\psi^E = u_1 u_2^s u_3^M$, or 0 otherwise.

For each k and m, $\varrho^{\mathsf{CL}}_{k,m}(M)$ outputs a tuple $((E, \boldsymbol{s}), r)$ by the following rules:

(1) choose $r \in_{\mathrm{U}} \{0,1\}^{\ell_{\mathrm{seed}}}$, and then set $E := H_{\mathsf{PRIMES}}(r)$; and

(2) for the vector $\boldsymbol{s} = (s_1, s_2, \ldots, s_m)$, set $s_1 := 1$, $s_2 \in_{\mathrm{U}} [0, 2^{\ell+\ell_{\mathrm{msg}}+\ell_{\mathrm{parm}}} - 1]$, $s_3 := M$ and $s_i := 0$ for each $i \in [4, m]$.

Figure 3.7: Camenisch-Lysyanskaya Signature Scheme [11] and ϱ^{CL} [13]

Let $\ell_{\exp} \leq \ell/2 - 1$.

Key Generator KGen**.** On input 1^k, KGen works as follows:

(1) choose an RSA modulus $N = PQ \in \mathbb{N}^{\mathrm{safe}}_{\mathsf{RSA}(\ell)}$, elements $u_1, u_2 \in_{\mathrm{U}} \mathrm{QR}_N$ and then a random $\ell_{\exp}$-bit prime E'; and

(2) output a public key $pk = (N, u_1, u_2, E')$ and a secret key (P, Q).

Signing Algorithm Sign**.** On input (sk, pk, M), Sign issues a signature σ on the message M in the following way:

(1) choose an element $R \in_{\mathrm{U}} \mathrm{QR}_N$ and a random $\ell_{\exp}$-bit prime E so that $E \neq E'$;

(2) compute an element $c := R^{E'}/u_2^{H'(M)}$, and then find an element ψ so that $\psi^E = u_1 u_2^{H'(c)}$; and

(3) output a signature $\sigma = (R, E, \psi)$.

Verification Algorithm Verify**.** On input (pk, M, σ), Verify outputs 1 if E is an $\ell_{\exp}$-bit odd integer different from E' and it holds that $c = R^{E'}/u_2^{H'(M)}$ and $\psi^E = u_1 u_2^{H'(c)}$, or 0 otherwise.

For each k, m, an index $N \in \mathcal{N}(k)$, an element $u_2 \in \mathbb{G}_N$, and a prime $E' \in \left(2^{\ell_{\exp}-1}, 2^{\ell_{\exp}}\right)$, $\varrho^{\mathsf{CS}}_{k,m,(N,u_2,E')}(M)$ outputs a tuple $((E, \boldsymbol{s}), (r, R))$ by the following rules:

(1) choose $r \in_{\mathrm{U}} \{0,1\}^{\ell_{\mathrm{seed}}}$, and then set $E := H_{\mathsf{PRIMES}}(r)$;

(2) for the vector $\boldsymbol{s} = (s_1, s_2, \ldots, s_m)$,

(2-1) set $s_1 := 1$ and $s_i := 0$ for each $i \in [3, m]$; and

(2-2) choose $R \in_{\mathrm{R}} \mathbb{G}_N$, compute $c := R^{E'}/u_2^{H'(M)}$, and set $s_2 := H'(c)$.

It should be noted that to determine a family $\varrho^{\mathsf{CS}}_{k,m,(N,u_2,E')}$ of probability distributions, we need a group index $N \in \mathcal{N}(k)$, a group element $u_2 \in \mathbb{G}_N$ and a prime E' besides k and m. For the APF game with respect to ϱ^{CS}, we assume that in Setup phase, the game tuple $(N, \alpha, \varrho^{\mathsf{CS}}_{k,m,(N,u_2,E')})$ is determined as follows: the challenger chooses a group index N and an assignment α as the same as Definition 2.6, then specifies a family $\varrho^{\mathsf{CS}}_{k,m,(N,u_2,E')}$ by setting $u_2 := \alpha(a_2)$ and choosing a prime $E' \in_{\mathrm{R}} \left(2^{\ell_{\exp}-1}, 2^{\ell_{\exp}}\right)$.

Figure 3.8: Cramer-Shoup Signature Scheme [14] and ϱ^{CS} [13]

Key Generator KGen**.** On input 1^k, KGen works as follows:

(1) choose an RSA modulus $N = PQ \in \mathbb{N}^{\mathrm{safe}}_{\mathsf{RSA}(\ell)}$, and then choose elements $u_1, u_2, u_3 \in_{\mathrm{U}} \mathrm{QR}_N$; and

(2) output a public key $pk = (N, u_1, u_2, u_3)$ and a secret key (P, Q).

Signing Algorithm Sign**.** On input (sk, pk, M), Sign issues a signature σ on the message M in the following way:

(1) choose a random ℓ_{exp}-bit prime E, and a random $(\ell_{\mathrm{exp}} - 1)$-bit integer s;

(2) find an element ψ so that $\psi^E = u_1 u_2^s u_3^{s \oplus H'(M)}$, where $\oplus$ denotes the bitwise XOR operator; and

(3) output a signature $\sigma = (E, s, \psi)$.

Verification Algorithm Verify**.** On input (pk, M, σ), Verify outputs 1 if E is an ℓ_{exp}-bit odd integer, s is an $(\ell_{\mathrm{exp}} - 1)$-bit value, and $\psi^E = u_1 u_2^s u_3^{s \oplus H'(M)}$, or 0 otherwise.

For each k and m, $\varrho^{\mathsf{Fis}}_{k,m}(M)$ outputs a tuple $((E, \boldsymbol{s}), r)$ by the following rules:

(1) choose $r \in_{\mathrm{U}} \{0,1\}^{\ell_{\mathrm{seed}}}$, and then set $E := H_{\mathsf{PRIMES}}(r)$; and

(2) for the vector $\boldsymbol{s} = (s_1, s_2, \ldots, s_m)$, set $s_1 := 1$, $s_2 \in_{\mathrm{U}} [0, 2^{\ell_{\mathrm{exp}}-1} - 1]$, $s_3 := s_2 \oplus H'(M)$, and $s_i := 0$ for each $i \in [4, m]$.

Figure 3.9: Fischlin Signature Scheme [18] and ϱ^{Fis} [13]

Key Generator KGen**.** On input 1^k, KGen works as follows:

(1) choose an RSA modulus $N = PQ \in \mathbb{N}^{\mathrm{safe}}_{\mathsf{RSA}(\ell)}$, and then choose an element $u \in_{\mathrm{U}} \mathrm{QR}_N$; and

(2) output a public key $pk = (N, u)$ and a secret key (P, Q).

Signing Algorithm Sign**.** On input (sk, pk, M), Sign issues a signature σ on the message M in the following way:

(1) set $E := H_{\mathsf{DI}}(M)$, where H_{DI} is a division-intractable hash function;

(2) find an element ψ so that $\psi^E = u$; and

(3) output a signature $\sigma = (E, \psi)$.

Verification Algorithm Verify**.** On input (pk, M, σ), Verify outputs 1 if $E = H_{\mathsf{DI}}(M)$, and $\psi^E = u$, or 0 otherwise.

For each k and m, $\varrho^{\mathsf{GHR}}_{k,m}(M)$ outputs a tuple $((E, \boldsymbol{s}), M)$ by the following rules:

(1) set $E := H_{\mathsf{DI}}(M)$; and

(2) for the vector $\boldsymbol{s} = (s_1, s_2, \ldots, s_m)$, set $s_1 := 1$ and $s_i := 0$ for each $i \in [2, m]$.

Figure 3.10: Gennaro-Halevi-Rabin Signature Scheme [22] and ϱ^{GHR} [13]

Let $\ell_{\mathrm{exp}} = \ell/2 - 3$ and $\ell_{\mathrm{msg}} = m - 1$.

Key Generator KGen**.** On input 1^k, KGen works as follows:

(1) choose an RSA modulus $N = PQ \in \mathbb{N}^{\mathrm{safe}}_{\mathsf{RSA}(\ell)}$, and then choose elements $u_0, u_1, \ldots, u_{\ell_{\mathrm{msg}}} \in_{\mathrm{U}} \mathrm{QR}_N$; and

(2) output a public key $pk = (N, u_0, u_1, \ldots, u_{\ell_{\mathrm{msg}}})$ and a secret key (P, Q).

Signing Algorithm Sign**.** On input (sk, pk, M), Sign issues a signature σ on the message M in the following way:

(1) choose a random ℓ_{exp}-bit prime E, and then find an element ψ so that $\psi^E = u_0 \prod_{i=1}^{\ell_{\mathrm{msg}}} u_i^{M_i}$, where $M_i \in \{0,1\}$ denotes the i-th bit of M; and

(2) output a signature $\sigma = (E, \psi)$.

Verification Algorithm Verify**.** On input (pk, M, σ), Verify outputs 1 if E is an ℓ_{exp}-bit odd integer, and $\psi^E = u_0 \prod_{i=1}^{\ell_{\mathrm{msg}}} u_i^{M_i}$, or 0 otherwise.

For each k and m, $\varrho^{\mathsf{HK}}_{k,m}(M)$ outputs a tuple $((E, \boldsymbol{s}), r)$ by the following rules:

(1) choose $r \in_{\mathrm{U}} \{0,1\}^{\ell_{\mathrm{seed}}}$, and then set $E := H_{\mathsf{PRIMES}}(r)$; and

(2) for the vector $\boldsymbol{s} = (s_1, s_2, \ldots, s_m)$, set $s_1 := 1$ and $s_i := M_{i-1}$ for each $i \in [2, m]$.

Figure 3.11: Hofheinz-Kiltz Signature Scheme [27] and ϱ^{HK} [13]

Zhu Signature scheme is basically the same as the CL Signature scheme described in Fig. 3.7 except that an integer s chosen by Sign is of binary length $\ell_{\mathrm{exp}} - 1$. For each k and m, $\varrho^{\mathsf{Zhu}}_{k,m}(M)$ outputs a tuple $((E, \boldsymbol{s}), r)$ by the following rules:

(1) choose $r \in_{\mathrm{U}} \{0,1\}^{\ell_{\mathrm{seed}}}$, and then set $E := H_{\mathsf{PRIMES}}(r)$; and

(2) for the vector $\boldsymbol{s} = (s_1, s_2, \ldots, s_m)$, set $s_1 := 1$, $s_2 \in_{\mathrm{U}} [0, 2^{\ell_{\mathrm{exp}}-1} - 1]$, $s_3 := M$ and $s_i := 0$ for each $i \in [4, m]$.

Figure 3.12: Zhu Signature Scheme [48, 49] and ϱ^{Zhu} [13]

3.3 Concluding Remarks

In this chapter, we have shown two impossibility results on the adaptive pseudo-freeness of the RSA group family $\{\mathbb{Z}_N^\times\}$. First, we have given a negative circumstantial evidence for the question whether or not $\{\mathbb{Z}_N^\times\}$ is strongly-adaptive pseudo-free. More precisely, we have shown that the RSA group family $\{\mathbb{Z}_N^\times\}$ cannot be proven to be strongly-adaptive pseudo-free from the SRSA assumption via algebraic reductions, as long as the SRSA assumption holds. In other words, the strong adaptive pseudo-freeness of $\{\mathbb{Z}_N^\times\}$ cannot be shown under the SRSA assumption, by employing only current proof techniques which are frequently used in ordinary security proofs. Since the SRSA assumption is one of the strongest assumption, this implies that the strong adaptive pseudo-freeness of $\{\mathbb{Z}_N^\times\}$ may be far from feasibility. Hence it is reasonable to use parametric distributions to construct a concrete adaptive pseudo-free group.

As the second result, we have again given a negative circumstantial evidence for the question whether or not the adaptive pseudo-freeness of $\{\mathbb{Z}_N^\times\}$ can be proven from some assumption other than the SRSA assumption, by using the parametric distributions proposed in [13]. Namely, we have shown that it cannot be proven from the RSA assumption via algebraic reductions, as long as the RSA assumption holds. Moreover, Theorem 3.9 holds even when an adversary is static, namely the adversary is restricted so that it makes no query during the APF game. Thus, our result indicates that even the "static" pseudo-freeness of $\{\mathbb{Z}_N^\times\}$ over our class $\mathcal{E}$ of equations cannot be proven from the RSA assumption via algebraic reductions, as long as the RSA assumption holds.

As a consequence of Theorem 3.9 and Lemma 3.15, it follows that the SRSA-based signature schemes proposed by [11, 14, 18, 22, 27, 49] may not be proven to be EUF-KOA from the RSA assumption via algebraic reductions, as long as the RSA assumption holds. This impossibility result on the SRSA-based signatures indicates that the adaptive pseudo-free group is useful to discuss whether or not the security of cryptographic schemes is provable from a specific assumption.

Chapter 4

The RSA Group $\mathbb{Z}_N^\times$ Is Adaptive Pseudo-Free under the RSA Assumption

In this chapter, we develop a new parametric distribution and show that the RSA group family $\left\{\mathbb{Z}_N^\times\right\}$ is adaptive pseudo-free with the parametric distribution under the RSA assumption. In Section 4.1, we define a slightly different variant of the adaptive pseudo-free group which is used throughout this chapter. We describe technical lemmas in Section 4.2. Our parametric distribution is given in Section 4.3, and we show in Section 4.4 that the RSA group family $\left\{\mathbb{Z}_N^\times\right\}$ is adaptive pseudo-free under the RSA assumption by using our parametric distribution. In Section 4.5, we give a concluding remarks for this chapter.

4.1 Adaptive Pseudo-Free Groups with respect to a Family of Parametric Distributions

We employ another definition of the adaptive pseudo-free group which is slightly different from the original definition described in Definition 2.6.

Let k be a security parameter, let A be a set of $m = m(k)$ symbols and let $I = \cup_{k\geq 0} I(k)$ be an index set. We assume that each index $\xi \in I(k)$ is of polynomial length in k. We suppose that for each k, m and each index $\xi \in I(k)$, a class $\mathcal{E}_{k,m}^{\xi}$ of a pair (λ, r) is designated, and we provide a family $\varrho_{k,m}^{\xi} = \left\{\varrho_{k,m}^{\xi}(M)\right\}$ of probabilistic distributions $\varrho_{k,m}^{\xi}(M)$ over $\mathcal{E}_{k,m}^{\xi}$. We also assume that the description $\varrho_{k,m}^{\xi}$ can be easily obtained by k, m and ξ.

Setup. The challenger chooses a random group index $N \in_{\mathrm{U}} \mathcal{N}(k)$. Then, it specifies an assignment $\alpha : A \to \mathbb{G}_N$ by independently choosing an element $\alpha(a) \in_{\mathrm{R}} \mathbb{G}_N$ at random according to the designated sampling algorithm for each $a \in A$. Moreover, the challenger specifies a distribution family $\varrho^{\xi}_{k,m}$ by choosing an index $\xi \in_{\mathrm{U}} I(k)$. The adversary is given the game tuple $\left(N, \alpha, \varrho^{\xi}_{k,m}\right)$.

Equations queries. For each t-th query, $\mathcal{A}$ chooses a parameter M_t and hands it to the challenger. The challenger chooses a pair $(\lambda^{(t)}, r_t)$ of an equation $\lambda^{(t)} = (E_t, \boldsymbol{s}_t)$ and a string r_t according to the distribution $\varrho^{\xi}_{k,m}(M_t)$. Then it returns the pair $(\lambda^{(t)}, r_t)$ and a solution $\psi_t \in \mathbb{G}_N$ for the interpreted equation $\lambda^{(t)}_{\alpha} : x^{E_t} = \prod_{i=1}^{m} \alpha(a_i)^{s_{t,i}}$ to $\mathcal{A}$.

Challenge. Eventually, the adversary $\mathcal{A}$ outputs a tuple $((\lambda^*, r^*), \psi^*)$ of an equation $\lambda^* = (E^*, \boldsymbol{s}^*)$ and a string r^* together with a solution ψ^* of the interpreted equation λ^*_{α} over $\mathbb{G}_N$. The challenger outputs 1 if the following conditions hold, or 0 otherwise:

- $(\lambda^*, r^*) \in \mathcal{E}^{\xi}_{k,m}$;
- λ^* is nontrivial with respect to Λ, the set of queried equations and corresponding solutions appeared in Equations queries phase; and
- ψ^* is actually a solution of λ^*_{α}.

An adversary $\mathcal{A}$ is said to *win the adaptive pseudo-free game of the group family $\mathcal{G}$ with respect to the family $\left\{\varrho^{\xi}\right\}_{\xi \in I}$ of the parametric distributions $\varrho^{\xi} = \left\{\varrho^{\xi}_{k,m}\right\}$* if the challenger outputs 1 in the game between the challenger and the adversary $\mathcal{A}$.

Definition 4.1 (Adaptive Pseudo-Free Groups w.r.t. $\left\{\varrho^{\xi}\right\}_{\xi \in I}$)
Let k be a security parameter, let q and m be polynomials in k, and let $\left\{\varrho^{\xi}\right\}_{\xi \in I}$ be a family of parametric distributions ϱ^{ξ}. A family $\mathcal{G} = \{\mathbb{G}_N\}_{N \in \mathcal{N}}$ of groups is (m, q, ϵ)-adaptive pseudo-free with respect to the family $\left\{\varrho^{\xi}\right\}_{\xi \in I}$ of the parametric distributions*, if for any PPT adversary $\mathcal{A}$ that makes at most $q = q(k)$ queries, and any set A of $m = m(k)$ symbols, the probability that $\mathcal{A}$ wins the adaptive pseudo-free game of the group family $\mathcal{G}$ with respect to $\left\{\varrho^{\xi}\right\}_{\xi \in I}$ is at most $\epsilon = \epsilon(k)$, where the probability is taken over the random choices of the index $N \in_{\mathrm{U}} \mathcal{N}(k)$, $\alpha(a) \in_{\mathrm{R}} \mathbb{G}_N$ for each $a \in A$, $\xi \in_{\mathrm{U}} I(k)$ and the equation $(\lambda^{(t)}, r_t) \in_{\mathrm{R}} \mathcal{E}^{\xi}_{k,m}$ for each $1 \leq t \leq q$, and the internal coin flips of $\mathcal{A}$.*

We note that Definition 4.1 can be viewed as a generalization of the original

definition in Definition 2.6. Definition 4.1 involves a family $\left\{\varrho^\xi\right\}_{\xi\in I}$ of parametric distributions, whereas the original definition involves a single parametric distribution.

4.2 Technical Lemmas

We use a series of lemmas in this chapter. Lemma 4.3 is trivial.

Lemma 4.2
Let $\{n_1(k)\}_{k\in\mathbb{N}}$ and $\{n_2(k)\}_{k\in\mathbb{N}}$ be two sequences of natural numbers. Assume that n_2/n_1 is negligible in k. If $z \in_{\mathrm{U}} \mathbb{Z}_{n_1}$, then the distribution of $z \bmod n_2$ is statistically close to the uniform distribution over $\mathbb{Z}_{n_2}$.

Proof. Since $n_1 > n_2$, let $n_1 = bn_2 + c$, where $0 \le c < n_2$. For any $a \in \mathbb{Z}_{n_2}$, let $P_1(a) := \Pr_{z\in_{\mathrm{U}}\mathbb{Z}_{n_1}}[z \equiv a \pmod{n_2}]$ and $P_2(a) := \Pr_{z\in_{\mathrm{U}}\mathbb{Z}_{n_2}}[z \equiv a \pmod{n_2}]$. Then we have $P_1(a) = (b+1)/n_1$ if $0 \le a < c$, or $P_1(a) = b/n_1$ if $c \le a < n_2$, and $P_2(a) = 1/n_2$. It follows from $0 \le c < n_2$ that

$$\begin{aligned}
\frac{1}{2}\sum_{a\in\mathbb{Z}_{n_2}} |P_1(a) - P_2(a)| &= \frac{c}{2}\left|\frac{b+1}{n_1} - \frac{1}{n_2}\right| + \frac{n_2-c}{2}\left|\frac{b}{n_1} - \frac{1}{n_2}\right| \\
&= \frac{c}{2}\left|\frac{bn_2 + n_2 - n_1}{n_1 n_2}\right| + \frac{n_2-c}{2}\left|\frac{bn_2 - n_1}{n_1 n_2}\right| \\
&= \frac{c}{2}\left|\frac{-c + n_2}{n_1 n_2}\right| + \frac{n_2-c}{2}\left|\frac{-c}{n_1 n_2}\right| \\
&= \frac{c}{2}\left(\frac{n_2-c}{n_1 n_2}\right) + \frac{n_2-c}{2}\left(\frac{c}{n_1 n_2}\right) \\
&= \frac{c(n_2-c)}{n_1 n_2} \\
&< \frac{n_2}{n_1}.
\end{aligned}$$

By the assumption that n_2/n_1 is negligible in k, if $z \in_{\mathrm{U}} \mathbb{Z}_{n_1}$, then the distribution of $z \bmod n_2$ is statistically close to the uniform distribution over $\mathbb{Z}_{n_2}$. □

Lemma 4.3
Let C and n be any natural numbers. Let $\mathrm{ADD}_{C,n}$ map each element $x \in \mathbb{Z}_n$ to $(x + C) \bmod n \in \mathbb{Z}_n$, and let $\mathrm{MUL}_{C,n}$ map each element $x \in \mathbb{Z}_n$ to $Cx \bmod n \in \mathbb{Z}_n$. Then, $\mathrm{ADD}_{C,n}$ is bijective, and $\mathrm{MUL}_{C,n}$ is bijective provided that $C \in \mathbb{Z}_n^\times$.

4.3 Our Parametric Distribution $\varrho^{K,c}$

In this section, we propose a new parametric distribution $\varrho^{K,c}$. Our construction of $\varrho^{K,c}$ involves a hash function $H_{K,c}$ that is similar to the one introduced by Hohenberger and Waters [30]. Their hash function has been employed in many RSA-based schemes in the standard model [12, 30, 28, 33, 36, 41, 42, 46, 47].

The Construction of a Hash Function $H_{K,c}$. In order to introduce our parametric distribution $\varrho^{K,c}$, we establish a hash function $H_{K,c}$. Let k be a security parameter. In order to establish $H_{K,c}$, we define a keyed pseudo-random function.

Definition 4.4 (Keyed Pseudo-Random Function [32])
Let $\mathcal{K} := \cup_{k\geq 0}\mathcal{K}(k)$ denote a key space, and let ℓ_{dom} and ℓ_{range} be polynomials in k. We say that $F : \mathcal{K} \times \{0,1\}^{\ell_{\mathrm{dom}}} \to \{0,1\}^{\ell_{\mathrm{range}}}$ is a keyed pseudo-random function *with the key space $\mathcal{K}$ if there exists no PPT adversary $\mathcal{D}$ such that*

$$\left| \Pr_{K\in_{\mathrm{U}}\mathcal{K}(k)}\left[\mathcal{D}^{F_K(\cdot)}(1^k)=1\right] - \Pr_{f_k\in_{\mathrm{U}}\mathcal{F}_k^{\ell_{\mathrm{dom}},\ell_{\mathrm{range}}}}\left[\mathcal{D}^{f_k(\cdot)}(1^k)=1\right]\right|,$$

is nonnegligible in k, where $F_K(\cdot) := F(K,\cdot)$ and for each k, $\mathcal{F}_k^{\ell_{\mathrm{dom}},\ell_{\mathrm{range}}}$ denotes a set of all functions mapping each $\ell_{\mathrm{dom}}(k)$-bits string to an $\ell_{\mathrm{range}}(k)$-bits string. The probability is taken over the random choice of a key $K \in_{\mathrm{U}} \mathcal{K}(k)$, that of $f_k \in_{\mathrm{U}} \mathcal{F}_k^{\ell_{\mathrm{dom}},\ell_{\mathrm{range}}}$ and coin flips of $\mathcal{D}$.

Let $N \in \mathbb{N}^{\mathrm{safe}}_{\mathsf{RSA}(\ell)}$ be an RSA composite with length $\ell = \ell(k)$, and let $\ell_F := \ell - 1$. Let $F : \mathcal{K} \times \{0,1\}^* \to \{0,1\}^{\ell_F}$ be a keyed pseudo-random polynomial-time computable function with a key space $\mathcal{K}$, where the length of any key $K \in \mathcal{K}(k)$ is a polynomial in k. In particular, we require that $F_K(r)$ is almost uniformly distributed over $\{0,1\}^{\ell_F}$ with $K \in_{\mathrm{U}} \mathcal{K}(k)$ for any given string r of polynomial length in k. For any $K \in \mathcal{K}$ and any $c \in \{0,1\}^{\ell_F}$, we define a function $h_{K,c} : \{0,1\}^* \to \mathbb{P}_{<2^\ell}$ by

$$h_{K,c}(r) := \mathsf{nextprime}(F_K(r) \oplus c),$$

where $F_K(\cdot) := F(K,\cdot)$, the function $\mathsf{nextprime} : \{0,1\}^{\ell_F} \to \mathbb{P}_{<2^\ell}$ maps each string $x \in \{0,1\}^{\ell_F}$ (regarded as a binary representation of a natural number $\tilde{x}$) to the smallest odd prime $P \in \mathbb{P}_{<2^\ell}$ that is equal to or greater than $\tilde{x}$, and $\oplus$ denotes

the bitwise XOR operator. The function nextprime is computed in PPT with negligible error probability [35]. Hence we assume that one can always correctly compute nextprime and $h_{K,c}$ throughout this thesis.

Note that for any $P \in \mathbb{P}_{<2^\ell}$, nextprime is efficiently invertible in the sense that a preimage of nextprime is computable in polynomial-time in the binary length of P.

For any string $r \in \{0,1\}^*$ of length ℓ, we write $r = r_1 r_2 \cdots r_\ell$, where $r_i \in \{0,1\}$ is the i-th bit of r. For each $j \in [1,\ell]$, the *j-prefix* of r is the string $r^{(j)} = r_1 r_2 \cdots r_j$ consisting of the leading j bits of r. The *0-prefix* $r^{(0)}$ of r is the empty string. Let ℓ_H be a polynomial in k. For each $K \in \mathcal{K}$ and $c \in \{0,1\}^{\ell_F}$, we define a function $H_{K,c} : \{0,1\}^{\ell_H} \to \mathbb{Z}$ by

$$H_{K,c}(r) := \prod_{j=1}^{\ell_H} h_{K,c}(r^{(j)}). \tag{4.1}$$

For any fixed key K and string c, $H_{K,c}(r) \neq H_{K,c}(r')$ implies that $r \neq r'$ for any $r, r' \in \{0,1\}^{\ell_H}$, because $H_{K,c}$ is single-valued. For any $c \in \{0,1\}^{\ell_F}$, $h_{K,c}$ is collision-resistant with $K \in_{\mathrm{U}} \mathcal{K}(k)$, because F_K is a pseudo-random function [30, 41]. Therefore, one can observe that for any $c \in \{0,1\}^{\ell_H}$, $H_{K,c}$ is also collision-resistant with $K \in_{\mathrm{U}} \mathcal{K}(k)$ as follows. Assume that $H_{K,c}$ is not collision-resistant, namely distinct strings $r, r' \in \{0,1\}^{\ell_H}$ satisfying $H_{K,c}(r) = H_{K,c}(r')$ can be efficiently found. Since $h_{K,c}(r) = h_{K,c}(r^{(\ell_H)})$ is a prime factor of $H_{K,c}(r) = H_{K,c}(r')$ and $H_{K,c}(r') = \prod_{j=1}^{\ell_H} h_{K,c}(r'^{(j)})$ is a prime factorization, we have $h_{K,c}(r) = h_{K,c}(r'^{(j)})$ for some $1 \leq j \leq \ell_H$ and such an index j can be found in polynomial-time because $h_{K,c}$ is polynomial-time computable. If $j < \ell_H$, then we have $r \neq r'^{(j)}$ because their lengths are different. If $j = \ell_H$, then we have $r \neq r' = r'^{(\ell_H)}$. In either case, one can find a collision of $h_{K,c}$ in polynomial-time. However, this is impossible because $h_{K,c}$ is collision-resistant.

Remark 4.5

Catalano, Fiore and Warinschi [13] used the notion of the division intractability in order to extract an answer for the SRSA instance (N, y) by employing Lemma 2.12. However, it seems to be hard to straightforwardly apply their way to the case of the RSA problem. The difficulty of breaking RSA rather than SRSA is that the exponent e of the final output $z^e \equiv y \pmod N$ is forced by the given instance (N, e, y). Our strategy of breaking RSA for a given instance (N, e, y) is to employ the hash function $H_{K,c}$ in a similar fashion to [41]. Namely,

we embed the given RSA exponent e into a prime factor of $H_{K,c}$ by choosing appropriate strings r and c so that $e = h_{K,c}(r)$. When e is actually a prime, this embedding can be done in polynomial-time as $\mathsf{nextprime}$, and hence also $h_{K,c}$, is polynomial-time invertible. We follow only the case where e is a sufficiently large prime. We will explain that this is enough for our purpose in **Remark 4.7** later.

A Parametric Distribution $\varrho^{K,c}$. We fix any polynomials $\ell_{\mathrm{exp1}} = \ell_{\mathrm{exp1}}(k)$, $\ell_{\mathrm{exp2}} = \ell_{\mathrm{exp2}}(k)$ and $\ell_{\mathrm{diff}} = \ell_{\mathrm{diff}}(k)$ such that $\ell_{\mathrm{exp1}} + 1 \leq \ell/2$, ℓ_{diff} is not a constant and $\ell_{\mathrm{exp1}} = \ell_{\mathrm{exp2}} + \ell_{\mathrm{diff}}$. For each k, m, $K \in \mathcal{K}$ and $c \in \{0,1\}^{\ell_F}$, we denote by $\mathcal{E}_{k,m}^{K,c}$ the set of all pairs (λ, r) of an equation $\lambda = (E, (s_1, \ldots, s_m))$ and a string $r \in \{0,1\}^{\ell_H}$ such that $E = H_{K,c}(r)$, $s_1 = 1$, $s_2 \in \mathcal{I}_{\mathrm{exp1}} := \left[0, 2^{\ell_{\mathrm{exp1}}} - 1\right]$, and $s_3, \ldots, s_m \in \mathcal{I}_{\mathrm{exp2}} := \left[0, 2^{\ell_{\mathrm{exp2}}} - 1\right]$.

Our parametric distribution $\varrho_{k,m}^{K,c}$ is described in the following manner. For each k, m, K and c, $\varrho_{k,m}^{K,c}$ is a family $\varrho_{k,m}^{K,c} = \left\{\varrho_{k,m}^{K,c}(M)\right\}$ of probabilistic distributions $\varrho_{k,m}^{K,c}(M)$ over the set $\mathcal{E}_{k,m}^{K,c}$ such that for any given parameter M,

- $\varrho_{k,m}^{K,c}(M)$ is polynomial-time samplable;
- the description of $\varrho_{k,m}^{K,c}(M)$ can be obtained in polynomial-time in k;
- r is uniformly distributed over $\{0,1\}^{\ell_H}$; and
- s_2 is uniformly distributed over $\mathcal{I}_{\mathrm{exp1}}$.

4.4 Main Theorem

In this section, we show that the RSA group $\mathbb{Z}_N^\times$ is adaptive pseudo-free with our parametric distribution $\varrho^{K,c}$ under the RSA assumption, rather than the SRSA assumption.

As in [13, 34] and in Chapter 3, we adopt any sampling algorithm for the family $\left\{\mathbb{Z}_N^\times\right\}_{N \in \mathcal{N}}$ which chooses an element $g \in \mathrm{QR}_N$ almost uniformly at random. We also assume that the challenger of the adaptive pseudo-free game chooses an index $N \in \mathcal{N}(k)$ according to the same distribution of N chosen from $\mathsf{KGen}_{\mathsf{RSA}}(1^k)$ for each security parameter k. In this section, we use the following specified version of the RSA assumption.

Definition 4.6 (ϵ_{RSA}-RSA assumption)

The ϵ_{RSA}-RSA assumption *holds if for any PPT adversary* $\mathcal{R}$*,*

$$\Pr\left[z^e \equiv y \pmod N : \begin{array}{l}(N,e) \leftarrow \mathsf{KGen}_{\mathsf{RSA}}(1^k),\\ y \in_{\mathrm{U}} \mathrm{QR}_N,\\ z \leftarrow \mathcal{R}(N,e,y)\end{array}\right] \leq \epsilon_{\mathsf{RSA}},$$

where the probability is taken over the coin flips of $\mathsf{KGen}_{\mathsf{RSA}}$ *and* $\mathcal{R}$*, and the uniform random choice* y *from* QR_N*.*

Remark 4.7

We need an additional assumption for the key generator $\mathsf{KGen}_{\mathsf{RSA}}$ in Definition 4.6. We say that a pair $(N,e) \leftarrow \mathsf{KGen}_{\mathsf{RSA}}(1^k)$ is *good* if e is a prime in $\mathbb{Z}_{\varphi(N)}^\times$ and $2^{\ell_{\exp 1}} \leq e < 2^{\ell_F}$. In our main theorem, $\mathsf{KGen}_{\mathsf{RSA}}$ is forced to generate a good pair with probability at least $1/\tau_{\mathsf{Good}}$ for some polynomial $\tau_{\mathsf{Good}}(k)$ for any sufficiently large k. Then, the ϵ_{RSA}-RSA assumption for such a key generator is referred to as an $(\epsilon_{\mathsf{RSA}}, \tau_{\mathsf{Good}})$-*RSA assumption*. Note that this assumption for $\mathsf{KGen}_{\mathsf{RSA}}$ is not exceedingly strong. For instance, if e is (almost) uniformly distributed over $\mathbb{Z}_{\varphi(N)}^\times$ with respect to each specific modulus N, then our assumption holds as shown in the following Lemma 4.8 in a similar manner to Lemma 3.8.

Lemma 4.8

Let ℓ*,* ℓ_F *and* $\ell_{\exp 1}$ *be polynomials defined as above. Let* $N \in \mathbb{N}^{\mathrm{safe}}_{\mathsf{RSA}(\ell)}$*. Then,* $\Pr_{e \in_{\mathrm{U}} \mathbb{Z}_{\varphi(N)}^\times}\left[e \in \mathbb{P}_{<\varphi(N)} \wedge 2^{\ell_{\exp 1}} \leq e < 2^{\ell_F}\right] > 1/(2\ell) - \mathrm{negl}(k)$*.*

Proof. As described in Eq. (3.6), it holds that for the number $\pi(n)$ of primes less than or equal to n,

$$\frac{n}{\log_2 n} < \pi(n) < \frac{2n}{\log_2 n},$$

for any sufficiently large n. Recall that ℓ, ℓ_F and $\ell_{\exp 1}$ are the polynomials such that $\ell_F = \ell - 1$ and $\ell_{\exp 1} + 1 \leq \ell/2$. It follows from $N = PQ \in \mathbb{N}^{\mathrm{safe}}_{\mathsf{RSA}}$ that $\varphi(\varphi(N)) < \varphi(N)$. Because the binary length of P and Q are $\ell/2$, we have P and Q are in the interval $\left[2^{\ell/2-1}, 2^{\ell/2} - 1\right]$. This implies that $2^{\ell-4} = (2^{\ell/2-2})^2 < (2^{\ell/2-1} - 1)^2 < (P-1)(Q-1) = \varphi(N) < PQ < 2^\ell$.

We now assume that $\varphi(N) \leq 2^{\ell_F}$. Since the order of $\mathbb{Z}_{\varphi(N)}^\times$ is $\varphi(\varphi(N))$, $e < \varphi(N) \leq 2^{\ell_F}$ for any $e \in \mathbb{Z}_{\varphi(N)}^\times$, and both $\varphi(N)$ and $2^{\ell_{\exp 1}}$ are not primes, we

have

$$\begin{aligned}
&\Pr_{e\in_{\mathrm{U}}\mathbb{Z}_{\varphi(N)}^\times}\left[e\in\mathbb{P}_{<\varphi(N)}\wedge 2^{\ell_{\exp 1}}\le e<2^{\ell_F}\right]\\
&=\Pr_{e\in_{\mathrm{U}}\mathbb{Z}_{\varphi(N)}^\times}\left[e\in\mathbb{P}_{<\varphi(N)}\wedge 2^{\ell_{\exp 1}}\le e<\varphi(N)\right]\\
&=\frac{\pi(\varphi(N)-1)-\pi(2^{\ell_{\exp 1}}-1)}{\varphi(\varphi(N))}\\
&=\frac{\pi(\varphi(N))-\pi(2^{\ell_{\exp 1}})}{\varphi(\varphi(N))}\\
&>\frac{\pi(\varphi(N))-\pi(2^{\ell_{\exp 1}})}{\varphi(N)}\\
&>\frac{1}{\varphi(N)}\left(\frac{\varphi(N)}{\log_2\varphi(N)}-\frac{2\cdot 2^{\ell_{\exp 1}}}{\log_2 2^{\ell_{\exp 1}}}\right)\\
&=\frac{1}{\log_2\varphi(N)}-\frac{2^{\ell_{\exp 1}+1}}{\varphi(N)\ell_{\exp 1}}\\
&>\frac{1}{\log_2 2^{\ell}}-\frac{2^{\ell_{\exp 1}+1}}{2^{\ell-4}\ell_{\exp 1}}\\
&\ge\frac{1}{\ell}-\frac{2^{\ell/2}}{2^{\ell-4}\ell_{\exp 1}}\\
&=\frac{1}{\ell}-\frac{1}{2^{\ell/2-4}\ell_{\exp 1}}\\
&>\frac{1}{2\ell}-\frac{1}{2^{\ell/2-4}\ell_{\exp 1}}.
\end{aligned}\tag{4.2}$$

On the other hand, we assume that $\varphi(N)>2^{\ell_F}$. In a similar manner to

Eq. (4.2), we have

$$\begin{aligned}
&\Pr_{e\in_{\mathrm{U}}\mathbb{Z}^\times_{\varphi(N)}}\left[e\in\mathbb{P}_{<\varphi(N)}\wedge 2^{\ell_{\exp 1}}\le e<2^{\ell_F}\right]\\
&=\frac{\pi(2^{\ell_F}-1)-\pi(2^{\ell_{\exp 1}}-1)}{\varphi(\varphi(N))}\\
&=\frac{\pi(2^{\ell_F})-\pi(2^{\ell_{\exp 1}})}{\varphi(\varphi(N))}\\
&>\frac{\pi(2^{\ell_F})-\pi(2^{\ell_{\exp 1}})}{N}\\
&>\frac{1}{N}\left(\frac{2^{\ell_F}}{\log_2 2^{\ell_F}}-\frac{2\cdot 2^{\ell_{\exp 1}}}{\log_2 2^{\ell_{\exp 1}}}\right)\\
&=\frac{1}{N}\left(\frac{2^{\ell_F}}{\ell_F}-\frac{2^{\ell_{\exp 1}+1}}{\ell_{\exp 1}}\right)\\
&>\frac{1}{2^\ell}\left(\frac{2^{\ell_F}}{\ell_F}-\frac{2^{\ell_{\exp 1}+1}}{\ell_{\exp 1}}\right)\\
&\ge\frac{1}{2^\ell}\left(\frac{2^{\ell-1}}{\ell-1}-\frac{2^{\ell/2}}{\ell_{\exp 1}}\right)\\
&=\frac{1}{2(\ell-1)}-\frac{1}{2^{\ell/2}\ell_{\exp 1}}\\
&>\frac{1}{2\ell}-\frac{1}{2^{\ell/2}\ell_{\exp 1}}.
\end{aligned}$$

Thus, it holds that $\Pr_{e\in_{\mathrm{U}}\mathbb{Z}^\times_{\varphi(N)}}\left[e\in\mathbb{P}_{<\varphi(N)}\wedge 2^{\ell_{\exp 1}}\le e<2^{\ell_F}\right] > 1/(2\ell) - \mathrm{negl}(k)$. □

We are now ready to state our main theorem.

Theorem 4.9

Let k be a security parameter, and let m and q be polynomials in k. The RSA assumption implies the adaptive pseudo-freeness of $\{\mathbb{Z}_N^\times\}$ with respect to the family $\{\varrho^{K,c}\}_{K,c}$ of parametric distributions.

More specifically, assume that the $(\epsilon_{\mathsf{RSA}},\tau_{\mathsf{Good}})$-RSA assumption holds. If there exists a PPT adversary $\mathcal{A}$ that breaks (m,q,ϵ)-adaptive pseudo-freeness of $\{\mathbb{Z}_N^\times\}$ of the RSA groups with respect to the family $\{\varrho^{K,c}\}_{K,c}$ of the parametric

distributions, then it must hold that

$$\epsilon < \frac{9}{2} mq\ell_H \tau_{\mathsf{Good}} \epsilon_{\mathsf{RSA}} + \mathrm{negl}(k).$$

Proof. Let $\mathcal{A}$ be the supposed PPT adversary. Namely, $\mathcal{A}$ wins the adaptive pseudo-free game for the family $\{\mathbb{Z}_N^\times\}$ with respect to the family $\{\varrho^{K,c}\}_{K,c}$ of the parametric distributions with probability greater than $\epsilon = \epsilon(k)$ for infinitely many k, where $\mathcal{A}$ makes at most $q = q(k)$ queries and a symbol set A is of size $m = m(k)$. In more detail, on input $(N, \alpha, \varrho_{k,m}^{K,c})$ of a group index $N \in \mathcal{N}(k)$, an assignment $\alpha : A \to \mathbb{Z}_N^\times$ and a distribution family $\varrho_{k,m}^{K,c}$, $\mathcal{A}$ outputs a tuple $((\lambda^*, r^*), \psi^*)$ of a pair $(\lambda^*, r^*) \in \mathcal{E}_{k,m}^{K,c}$ of an equation $\lambda^* = (E^*, \boldsymbol{s}^*)$ and a string r^* together with a solution ψ^* of the interpreted equation λ_α^* over $\mathbb{Z}_N^\times$, with probability greater than ϵ, where λ^* is nontrivial with respect to the set Λ of q previously queried equations, and $\boldsymbol{s}^* = (s_1^*, \ldots, s_m^*)$.

We construct a PPT algorithm $\mathcal{R}$ that breaks RSA with probability greater than $2\epsilon/(9mq\ell_H\tau_{\mathsf{Good}}) - \mathrm{negl}(k)$. In our construction, $\mathcal{R}$ plays the adaptive pseudo-free game with the adversary $\mathcal{A}$ in which $\mathcal{R}$ plays the role of the challenger.

Throughout the proof, we use the following notations. t denotes the index of the $\mathcal{A}$'s queries in the game, then $1 \le t \le q$. For each t, let $\big((\lambda^{(t)}, r_t), \psi_t\big)$ denote the answer to the adversary $\mathcal{A}$ determined by $\varrho_{k,m}^{K,c}(M_t)$ on the $\mathcal{A}$'s t-th query M_t. Namely, the t-th query is the pair $(\lambda^{(t)}, r_t)$ of the equation $\lambda^{(t)} : x^{E_t} = \prod_{i=1}^m a_i^{s_{t,i}}$, where $\boldsymbol{s}_t = (s_{t,1}, \ldots, s_{t,m})$, and the string $r_t \in \{0,1\}^{\ell_H}$ satisfying $H_{K,c}(r_t) = E_t$. $\psi_t \in \mathbb{Z}_N^\times$ is a solution for the interpreted equation $\lambda_\alpha^{(t)}$.

For each t, let j_t be the length of the longest common prefix of r_t and r^*. Let $j^* = \max_{1 \le t \le q} j_t$, and let t^* be any fixed index such that $j^* = j_{t^*}$. Then the j^*-prefix $r^{*(j^*)}$ of r^* is the same as the j^*-prefix $r_{t^*}^{(j^*)}$ of r_{t^*}. By the maximality of j^*, $r^{*(j^*+1)}$ is not a prefix of any r_t. For example, if $r^* = 101001$, $r_1 = 101111$ and $r_2 = 111001$, then $j_1 = 3$ and $j_2 = 1$, and hence $j^* = 3$ and $t^* = 1$.

We may assume without loss of generality that $H_{K,c}(r) \in \mathbb{Z}_{P'Q'}^\times$ for any $r \in \{0,1\}^{\ell_H}$, where P' and Q' are distinct primes such that $P = 2P' + 1$ and $Q = 2Q' + 1$ for an RSA composite $N = PQ \in \mathbb{N}_{\mathsf{RSA}}^{\mathrm{safe}}$. This is because N is easily factored if $H_{K,c}(r) \notin \mathbb{Z}_{P'Q'}^\times$ as follows. Assume that $H_{K,c}(r) = \prod_{j=1}^{\ell_H} h_{K,c}(r^{(j)}) \notin \mathbb{Z}_{P'Q'}^\times$. Since $h_{K,c}$ is prime-valued, we have $h_{K,c}(r^{(j_0)}) \in \{P', Q'\}$ for some $1 \le j_0 \le \ell_H$. Moreover, such an index j_0 can be found in polynomial-time. Therefore, P' or Q' is revealed, and hence $N = PQ$ is factored.

Our algorithm $\mathcal{R}$ aims to output the value $z \in \mathbb{Z}_N^\times$ such that $z^e \equiv y \pmod N$

on any given RSA instance (N, e, y), where $(N, e) \leftarrow \mathsf{KGen}_{\mathsf{RSA}}(1^k)$ and $y \in_{\mathrm{U}} \mathrm{QR}_N$. Since we are now under the $(\epsilon_{\mathsf{RSA}}, \tau_{\mathsf{Good}})$-RSA assumption, the pair (N, e) is good with probability at least $1/\tau_{\mathsf{Good}}$. Hence, we hereafter assume that (N, e) is a good pair. As mentioned just before Section 3.1.1, we may assume without loss of generality that the RSA ciphertext y is a generator of QR_N. Thus, the RSA instance (N, e, y) given to $\mathcal{R}$ always satisfies that $N \in \mathbb{N}_{\mathsf{RSA}}^{\mathrm{safe}}$, (N, e) is good and y is a generator of QR_N.

For the nontrivial equation $\lambda^* = (E^*, \boldsymbol{s}^*)$ with respect to Λ, Lemma 2.5 implies that $(E^*, (s_1^*, \ldots, s_m^*)) \notin \{(E_t, (s_{t,1}, \ldots, s_{t,m}))\}_{t=1}^q$. At the beginning, $\mathcal{R}$ therefore guesses uniformly at random the following three types of the adversary's output exponent E^* and $\boldsymbol{s}^*$:

Type I: $E^* \neq E_t$ for any $1 \leq t \leq q$.
Type II: There exists an index $1 \leq t \leq q$ such that $E^* = E_t$ and $s_2^* \neq s_{t,2}$.
Type III: The remaining case. Namely, there exists an index $1 \leq t \leq q$ such that $E^* = E_t$. Moreover, for any index $1 \leq t \leq q$ satisfying that $E^* = E_t$, $s_2^* = s_{t,2}$ holds.

Whenever $\mathcal{R}$ has noticed that his guess is wrong, then $\mathcal{R}$ aborts.

Type I. In this type, $E^* \neq E_t$ for any t. In Setup phase, $\mathcal{R}$ determines a distribution family $\varrho_{k,m}^{K,c}$ and a random assignment $\alpha : A \to \mathbb{Z}_N^\times$, where $A = \{a_1, \ldots, a_m\}$. For selecting the distribution family $\varrho_{k,m}^{K,c}$, K is randomly chosen from the key space $\mathcal{K}(k)$, and $c \in \{0,1\}^{\ell_F}$ is chosen so that $h_{K,c}(r^{*(j^*+1)}) = e$ holds for the given RSA exponent e. However, $\mathcal{R}$ does not know the string r^* and the index j^* before $\mathcal{A}$ outputs r^* in $\mathsf{Challenge}$ phase, and hence $\mathcal{R}$ does not know the string $r^{*(j^*+1)}$ at this time. Therefore, $\mathcal{R}$ guesses an index $t^* \in [1, q]$ at random and the length $j^* \in [0, \ell_H - 1]$ at random, and $\mathcal{R}$ computes the string c by using these guessed values.

Remark 4.10
$\mathcal{R}$ guesses the length j^* in the interval $[0, \ell_H - 1]$, because $j^* < \ell_H$ as follows. Since we are now in *Type I*, we have $E^* \neq E_t$ for any t. Since E^* and E_t will be given by $E^* = H_{K,c}(r^*)$ and $E_t = H_{K,c}(r_t)$, we have $r^* \neq r_t$. Thus, we have $j^* < \ell_H$.

We now describe the algorithm $\mathcal{R}$ when *Type I* is guessed. Assume that an RSA instance (N, e, y) is given.

Reduction Algorithm $\mathcal{R}$ for Type I.

Setup. (I-1) $\mathcal{R}$ chooses random strings $r_1, r_2, \ldots, r_q \in_{\mathrm{U}} \{0,1\}^{\ell_H}$.

(I-2) $\mathcal{R}$ chooses a random index $t^* \in_{\mathrm{U}} [1, q]$ and chooses a length $j^* \in_{\mathrm{U}} [0, \ell_H - 1]$.

$\mathcal{R}$ now determines a distribution family $\varrho_{k,m}^{K,c}$ and an assignment $\alpha : A \to \mathbb{Z}_N^\times$ in the following way.

(I-3) $\mathcal{R}$ chooses a key $K \in_{\mathrm{U}} \mathcal{K}(k)$, and then sets $c := F_K(r^{*(j^*+1)}) \oplus e$.

Note that if j^* and t^* are correctly guessed, $r^{*(j^*+1)}$ is the (j^*+1)-bits string that is obtained by appending the complement of r_{t^*}'s (j^*+1)-th bit to the tail of the j^*-prefix $r_{t^*}^{(j^*)}$ of r_{t^*}. Note also that since (N, e) is assumed to be good, and hence e is a prime less than 2^{ℓ_F}, the prime e can be viewed as an ℓ_F-bit string. Moreover, it holds that

$$\begin{aligned} h_{K,c}(r^{*(j^*+1)}) &= \mathsf{nextprime}(F_K(r^{*(j^*+1)}) \oplus c) \\ &= \mathsf{nextprime}(F_K(r^{*(j^*+1)}) \oplus (F_K(r^{*(j^*+1)}) \oplus e)) \\ &= \mathsf{nextprime}(e) \\ &= e. \end{aligned} \tag{4.3}$$

(I-4) For each $1 \le t \le q$, let $E_t = H_{K,c}(r_t)$.

(I-5) $\mathcal{R}$ fixes an assignment $\alpha : A \to \mathbb{Z}_N^\times$ in a way that:

- Choose $z_1, z_2, \ldots, z_m \in_{\mathrm{U}} \mathbb{Z}_{N^2}$;
- Let $C = \prod_{t=1}^{q} E_t$; and
- Set $\alpha(a_i) := y^{Cz_i} \bmod N$ for each $1 \le i \le m$.

We will show that each $\alpha(a_i)$ is distributed almost uniformly at random in QR_N later in the description of Game 2. Finally, $\mathcal{R}$ submits the index N, the assignment α and the tuple (K, c) to the adversary $\mathcal{A}$.

Equations queries. When the t-th parameter $M_t \in \{0,1\}^\ell$ is queried from the adversary $\mathcal{A}$, "the challenger" $\mathcal{R}$ should reply the tuple $\big((\lambda^{(t)}, r_t), \psi_t\big)$ to $\mathcal{A}$, where $\lambda^{(t)} = (E_t, \boldsymbol{s}_t)$.

(I-6) $\mathcal{R}$ generates $\boldsymbol{s}_t$ and ψ_t by using $E_1, \ldots, E_q$ generated in (I-4) as follows:

- Choose $s_{t,2} \in \mathcal{I}_{\mathrm{exp1}}$ and $s_{t,3}, \ldots, s_{t,m} \in \mathcal{I}_{\mathrm{exp2}}$, according to the distribution $\varrho_{k,m}^{K,c}(M_t)$;
- Set $\mu_t := C(z_1 + \sum_{i=2}^{m} z_i s_{t,i})$; and

- Set $\psi_t := y^{\mu_t/E_t} \bmod N$. Note that μ_t/E_t can be easily computed although the order $\mathrm{ord}(\mathrm{QR}_N)$ of QR_N is unknown, because $\mathcal{R}$ can efficiently compute each E_t.

$\mathcal{R}$ replies the tuple $((E_t, \boldsymbol{s}_t, r_t), \psi_t)$ to $\mathcal{A}$. Noting that $s_{t,1} = 1$, we have

$$\prod_{i=1}^{m} \alpha(a_i)^{s_{t,i}} = \prod_{i=1}^{m} \left(y^{Cz_i}\right)^{s_{t,i}} = y^{\mu_t}. \tag{4.4}$$

Then ψ_t is a valid solution for the equation $\lambda_\alpha^{(t)}$.

Challenge. Eventually, $\mathcal{A}$ outputs a tuple $((\lambda^*, r^*), \psi^*)$, where $\lambda^* = (E^*, \boldsymbol{s}^*)$. $\mathcal{R}$ checks whether or not his guess is correct.

(I-7) If $E^* = E_t$ for some t, then this is not the case for *Type I*, and hence $\mathcal{R}$'s guess is wrong. Then $\mathcal{R}$ aborts. Assume that $\mathcal{R}$ correctly guesses *Type I*. Then, $\mathcal{R}$ next verifies whether or not j^* and t^* are correctly guessed, namely $j^* = \max_{1 \le t \le q} j_t$ and $r^{*(j^*)} = r_{t^*}^{(j^*)}$ hold. If the guess is wrong, then $\mathcal{R}$ aborts.

Suppose that all guesses are correct. $\mathcal{R}$ tries to extract z such that $z^e \equiv y \pmod N$ as follows. For the case where $\mathcal{A}$ succeeds, we have the following equation:

$$\psi^{*E^*} = \prod_{i=1}^{m} \alpha(a_i)^{s_i^*} = y^{\mu^*}, \tag{4.5}$$

where $\mu^* = C(z_1 + \sum_{i=2}^{m} z_i s_i^*)$. Furthermore, we have

$$E^* = H_{K,c}(r^*) = \prod_{j=1}^{\ell_H} h_{K,c}(r^{*(j)}). \tag{4.6}$$

Since the guess of j^* and t^* is correct, we have $h_{K,c}(r^{*(j^*+1)}) = e$. Then it follows from Eqs. (4.5) and (4.6) that

$$\left(\psi^{*C^*}\right)^e = y^{\mu^*}, \tag{4.7}$$

where $C^* = \prod_{j=1, j\neq j^*+1}^{\ell_H} h_{K,c}(r^{*(j)}) = E^*/e$.

(I-8) If $\gcd(e, \mu^*) \neq 1$, then $\mathcal{R}$ aborts.

(I-9) Otherwise (namely, $\gcd(e, \mu^*) = 1$), then $\mathcal{R}$ obtains z such that $z^e \equiv y \pmod N$ by Eq. (4.7) and Lemma 2.11.

Analysis of Type I. We analyze the success probability ϵ_{RSA} of *Reduction Algorithm $\mathcal{R}$ for Type I* by the standard hybrid argument. In the following series of games, we adopt the behavior (I-1), (I-2), ..., (I-9) of $\mathcal{R}$ so that the first game Game 0 is the ordinary adaptive pseudo-free game and the final game Compute RSA coincides with *Reduction Algorithm $\mathcal{R}$ for Type I* that we have just described. For each $0 \leq i \leq 5$, let $\Pr[\mathsf{Succ}_i]$ be the probability that the adversary $\mathcal{A}$ wins Game i.

Game 0. This game is the ordinary adaptive pseudo-free game. Namely $\mathcal{R}$ invokes $\mathcal{A}$ with a random index $N' \in \mathcal{N}(k)$ ($N' = PQ$ with two safe primes $P = 2P' + 1$ and $Q = 2Q' + 1$), a random assignment α', and a key $K \in_{\mathrm{U}} \mathcal{K}(k)$ and a string $c \in_{\mathrm{U}} \{0,1\}^{\ell_F}$ for the distribution family $\varrho_{k,m}^{K,c}$. By the assumption, we have

$$\Pr[\mathsf{Succ}_0] > \epsilon. \tag{4.8}$$

Note that the randomly chosen modulus $N' \in \mathcal{N}(k)$ will be used until Game 5 instead of the given RSA modulus N.

Game 1. This game proceeds in the same way as Game 0 with the exception that $\mathcal{R}$ executes (I-1), (I-2) and (I-3) in Setup phase. Namely $\mathcal{R}$ chooses the random strings r_t for each $1 \leq t \leq q$, and guesses the indices t^* and j^*. Then, $\mathcal{R}$ chooses a random exponent $e' \in \mathbb{Z}^\times_{\varphi(N')}$ so that (N', e') is good instead of the given RSA exponent e, and sets $c := F_K(r^{*(j^*+1)}) \oplus e'$. (Note that c was randomly chosen in Game 0.)

The key K and the assignment α' are chosen as in Game 0. We now show that the string c chosen in this game is almost uniformly distributed over $\{0,1\}^{\ell_F}$. We define a bijection $\oplus_{e'}$ that maps $r \in \{0,1\}^{\ell_F}$ to $r \oplus e' \in \{0,1\}^{\ell_F}$. Since $F_K(r)$ is assumed to be almost uniformly distributed over $\{0,1\}^{\ell_F}$ with $K \in_{\mathrm{U}} \mathcal{K}(k)$ for any string r, the string $c = F_K(r^{*(j^*+1)}) \oplus e' = \oplus_{e'}(F_K(r^{*(j^*+1)}))$ is almost uniformly distributed over $\{0,1\}^{\ell_F}$. Therefore,

$$\Pr[\mathsf{Succ}_1] \geq \Pr[\mathsf{Succ}_0] - \mathrm{negl}(k). \tag{4.9}$$

Game 2. This game proceeds in the same way as Game 1 with the exception that $\mathcal{R}$ also executes (I-4) and (I-5) in Setup phase. Namely, $\mathcal{R}$ computes each $E_t = H_{K,c}(r_t)$, where r_t's are random strings as chosen in Game 1, and chooses a generator y' of $\mathrm{QR}_{N'}$ uniformly at random instead of the given RSA ciphertext y. Then $\mathcal{R}$ fixes the assignment α as in (I-5), and submits the assignment α instead of the random assignment α' (constructed in Game 0).

It follows from Lemma 4.2 that the distribution of each $z_i \bmod P'Q'$ is almost uniform over $\mathbb{Z}_{P'Q'}$, because $(P'Q')/N'^2 < 1/N'$ is negligible in k. Since y' is a generator of $\mathrm{QR}_{N'}$, y' is of order $P'Q'$ and y'^C is of order $P'Q'/\gcd(C, P'Q')$, respectively. Since each $E_t = H_{K,c}(r_t)$ can be assumed to be in $\mathbb{Z}^\times_{P'Q'}$, we have $C = \prod_{t=1}^q E_t \in \mathbb{Z}^\times_{P'Q'}$. Hence, y'^C is also a generator of $\mathrm{QR}_{N'}$. Therefore, each $\alpha(a_i) = (y'^C)^{z_i}$ is distributed almost uniformly at random over $\mathrm{QR}_{N'}$. Thus, we have

$$\Pr[\mathsf{Succ}_2] \geq \Pr[\mathsf{Succ}_1] - \mathrm{negl}(k). \tag{4.10}$$

Game 3. This game proceeds in the same way as Game 2 with the exception that $\mathcal{R}$ also executes (I-6) in Equations queries phase. In this game, for each t-th queried parameter M_t from $\mathcal{A}$, $\mathcal{R}$ chooses $s_{t,2} \in \mathcal{I}_{\mathrm{exp1}}$ and $s_{t,3}, \ldots, s_{t,m} \in \mathcal{I}_{\mathrm{exp2}}$, according to the distribution $\varrho^{K,c}_{k,m}(M_t)$, and then computes $\mu_t = C(z_1 + \sum_{i=2}^m z_i s_{t,i})$ by using C as computed in Game 2, and computes $\psi_t = y'^{\mu_t/E_t} \bmod N'$.

For each t-th parameter M_t, $\mathcal{R}$ can correctly reply the solution ψ_t for the interpreted equation $\lambda_\alpha^{(t)}$ to $\mathcal{A}$ by Eq. (4.4). Then we have

$$\Pr[\mathsf{Succ}_3] = \Pr[\mathsf{Succ}_2]. \tag{4.11}$$

Game 4. This game proceeds in the same way as Game 3 with the exception that $\mathcal{R}$ also executes (I-7) in Challenge phase. In this game, as in (I-7), $\mathcal{R}$ aborts if $\mathcal{R}$'s guess is wrong, namely $E^* = E_t$ for some t, $j^* \neq \max_{1 \leq t \leq q} j_t$ or $r^{*(j^*)} \neq r_{t^*}^{(j^*)}$.

The probability that $\mathcal{R}$ correctly guesses *Type I* is $1/3$, since $\mathcal{R}$ guesses the type of the adversary's output uniformly at random out of *Type I–III*. Since t^* is chosen uniformly at random from the set $[1, q]$, and j^* is chosen uniformly at random from the set $[0, \ell_H - 1]$, respectively, the success probability of $\mathcal{R}$'s guess for j^* and t^* is $1/(q\ell_H)$. Therefore, we have

$$\Pr[\mathsf{Succ}_4] \geq \frac{1}{3q\ell_H} \Pr[\mathsf{Succ}_3]. \tag{4.12}$$

Game 5. This game proceeds in the same way as Game 4 with the exception that $\mathcal{R}$ also executes (I-8) in Challenge phase. Namely $\mathcal{R}$ aborts if $\gcd(e', \mu^*) \neq 1$, where $\mu^* = C(z_1 + \sum_{i=2}^m z_i s_i^*)$.

We first show that $e' \nmid C$ with overwhelming probability. Since we have $j^* = \max_{1 \leq t \leq q} j_t$ (the correct guess), it holds that $r^{*(j^*+1)} \neq r_t^{(j^*+1)}$ for any $1 \leq t \leq q$. Note that for any $r \in \{0,1\}^{\ell_H}$ and any $1 \leq j \leq \ell_H$ with $j \neq$

$j^* + 1$, we have $r^{*(j^*+1)} \neq r^{(j)}$, because these are of different length. Hence, $r^{*(j^*+1)} \neq r_t^{(j)}$ for each $1 \leq t \leq q$ and each $1 \leq j \leq \ell_H$. Therefore it follows that $e' = h_{K,c}(r^{*(j^*+1)}) \nmid \prod_{t=1}^{q} \prod_{j=1}^{\ell_H} h_{K,c}(r_t^{(j)}) = C$ with overwhelming probability, since $h_{K,c}$ is a collision-resistant prime-valued function.

We then estimate the probability that $e' \mid (z_1 + \sum_{i=2}^{m} z_i s_i^*)$ for any fixed $\mathcal{A}$'s output $\boldsymbol{s}^*$. By Lemma 4.3, $\mathrm{ADD}_{\mu^{*\prime}, N'^2}$ is bijective, where $\mu^{*\prime} = \sum_{i=2}^{m} z_i s_i^*$. Note that $\boldsymbol{s}^*$ is independent of z_1, because $\mathcal{A}$ does not know z_1. It follows from $z_1 \in_{\mathrm{U}} \mathbb{Z}_{N'^2}$ that $z_1 + \sum_{i=2}^{m} z_i s_i^* \bmod N'^2 = \mathrm{ADD}_{\mu^{*\prime}, N'^2}(z_1)$ is also distributed uniformly at random over $\mathbb{Z}_{N'^2}$. Since $3/N'^2$ is negligible in k, the distribution of $z_1 + \sum_{i=2}^{m} z_i s_i^* \bmod 3$ is almost uniform over $\mathbb{Z}_3$ by Lemma 4.2. Therefore, we have

$$\begin{aligned}\Pr_{z_1 \in_{\mathrm{U}} \mathbb{Z}_{N'^2}}\left[z_1 + \sum_{i=2}^{m} z_i s_i^* \equiv 0 \pmod{e'}\right] &\leq \Pr_{z_1 \in_{\mathrm{U}} \mathbb{Z}_{N'^2}}\left[z_1 + \sum_{i=2}^{m} z_i s_i^* \equiv 0 \pmod{3}\right] \\ &= \frac{1}{3} + \mathrm{negl}(k),\end{aligned}$$

since e' is a prime greater than 3. Note that $\mathcal{A}$ always wins Game 5 whenever $\mathcal{A}$ wins Game 4, provided that $\gcd(e', \mu^*) = 1$. Therefore, the probability that $\mathcal{A}$ wins this game is

$$\Pr[\mathsf{Succ}_5] \geq \frac{2}{3}\Pr[\mathsf{Succ}_4] - \mathrm{negl}(k). \tag{4.13}$$

Compute RSA. This is the final game in which the challenger coincides with *Reduction Algorithm $\mathcal{R}$ for Type I*. Namely, this game proceeds in the same way as Game 5, with the exception that $\mathcal{R}$ uses the input tuple (N, e, y) instead of (N', e', y'), and executes (I-9) in Challenge phase. Therefore, $\mathcal{R}$ computes z such that $z^e \equiv y \pmod{N}$ if $\gcd(e, \mu^*) = 1$ as checked in Game 5.

Since $\mathcal{R}$ obtains z such that $z^e \equiv y \pmod{N}$ by Lemma 2.11 provided that $\mathcal{A}$ wins Game 5, we have

$$\epsilon_{\mathsf{RSA}} \geq \Pr[\mathsf{Succ}_5]. \tag{4.14}$$

Putting together Eqs. (4.8)–(4.14), we have

$$\epsilon_{\mathsf{RSA}} > \frac{2}{9q\ell_H}\epsilon - \mathrm{negl}(k). \tag{4.15}$$

Type II. In this case, there exists an index $1 \leq t \leq q$ such that $E^* = E_t$ and $s_2^* \neq s_{t,2}$. Since $H_{K,c}$ is collision-resistant, we have $r^* = r_t$ with overwhelming probability.

In the proof of *Type II*, let $1 \leq t^* \leq q$ be any fixed index such that $r^* = r_{t^*}$ and $s_2^* \neq s_{t^*,2}$, and let $j^* = \max_{1 \leq t \leq q, t \neq t^*} j_t$. (Note that these definitions are slightly different from in *Type I*.)

In a similar fashion to *Type I*, for the adaptive pseudo-free game with the adversary $\mathcal{A}$, $\mathcal{R}$ plays the role of the challenger. In Setup phase, $\mathcal{R}$ determines a distribution family $\varrho_{k,m}^{K,c}$ and a random assignment $\alpha : A \to \mathbb{Z}_N^\times$. For selecting the distribution family $\varrho_{k,m}^{K,c}$, K is randomly chosen from the key space $\mathcal{K}(k)$, and $c \in \{0,1\}^{\ell_F}$ is chosen so that $h_{K,c}(r^{*(j^*+1)}) = e$ holds for the given RSA exponent e. However, $\mathcal{R}$ does not know the string r^* and the index j^* before $\mathcal{A}$ outputs r^* in Challenge phase. $\mathcal{R}$ therefore guesses an index $t^* \in [1, q]$ at random, and finds the length j^* by finding the length j_t of the longest common prefix of r_t and r^* with using r_{t^*} for each $1 \leq t \leq q$ with $t \neq t^*$. Then, $\mathcal{R}$ computes $c := F_K(r^{*(j^*+1)}) \oplus e$ by using these values t^* and j^*. In a similar manner to Eq. (4.3), it holds that $h_{K,c}(r^{*(j^*+1)}) = e$ for the resulting c.

Remark 4.11
If there exists an index t' ($1 \leq t' \leq q$ and $t' \neq t^*$) such that $r_{t'} = r^*$, then we have $j^* = \max_{1 \leq t \leq q, t \neq t^*} j_t = \ell_H$. However, $(\ell_H + 1)$-prefix of r^* does not defined, namely c cannot be determined to be $h_{K,c}(r^{*(j^*+1)}) = e$. Therefore, if $\mathcal{R}$ finds such an index t', $\mathcal{R}$ aborts. Thus j^* is in $[0, \ell_H - 1]$.

We now describe the algorithm $\mathcal{R}$ when *Type II* is guessed. Assume that an RSA instance (N, e, y) is given.

Reduction Algorithm $\mathcal{R}$ for Type II.
Setup. (II-1) $\mathcal{R}$ chooses random strings $r_1, r_2, \ldots, r_q \in_{\mathrm{U}} \{0,1\}^{\ell_H}$.
(II-2) $\mathcal{R}$ guesses an index $t^* \in_{\mathrm{U}} [1, q]$ such that $r^* = r_{t^*}$ and $s_2^* \neq s_{t^*,2}$.
(II-3) If there exists an index $1 \leq t \leq q$ such that $t \neq t^*$ and $r_t = r_{t^*}$, then $\mathcal{R}$ aborts. (See **Remark 4.11**.) Otherwise, $\mathcal{R}$ finds the longest length $j^* = \max_{1 \leq t \leq q, t \neq t^*} j_t$ by using r_{t^*}.

$\mathcal{R}$ now determines a distribution family $\varrho_{k,m}^{K,c}$ and an assignment $\alpha : A \to \mathbb{Z}_N^\times$ in the following way.
(II-4) $\mathcal{R}$ chooses a key $K \in_{\mathrm{U}} \mathcal{K}(k)$, and then sets $c := F_K(r^{*(j^*+1)}) \oplus e$ as in (I-3)

of *Type I*.
(II-5) For each $1 \leq t \leq q$, let $E_t = H_{K,c}(r_t)$.
(II-6) $\mathcal{R}$ fixes the assignment $\alpha : A \to \mathbb{Z}_N^\times$ as follows:

- Choose $z_1, z_2, \ldots, z_m \in_{\mathrm{U}} \mathbb{Z}_{N^2}$, and $\beta \in_{\mathrm{U}} \mathcal{I}_{\mathrm{exp1}}$;
- Let $C = \prod_{t=1}^{q} E_t$, and $C_{t^*} = \prod_{t=1, t\neq t^*}^{q} E_t = C/E_{t^*}$; and
- Set $\alpha(a_1) := y^{Cz_1} \cdot y^{\beta C_{t^*}} \bmod N$, $\alpha(a_2) := y^{Cz_2} \cdot y^{-C_{t^*}} \bmod N$, and $\alpha(a_i) := y^{Cz_i} \bmod N$ for each $3 \leq i \leq m$.

We will show that each $\alpha(a_i)$ is distributed almost uniformly at random in QR_N later in the description of Game 3. Finally, $\mathcal{R}$ submits the index N, the assignment α and the tuple (K, c) to the adversary $\mathcal{A}$.

Equations queries. (II-7) When the t-th parameter $M_t \in \{0,1\}^\ell$ is queried from the adversary $\mathcal{A}$, "the challenger" $\mathcal{R}$ generates the tuple $((E_t, \boldsymbol{s}_t, r_t), \psi_t)$ as follows:

- $\mathcal{R}$ chooses $s_{t,2} \in \mathcal{I}_{\mathrm{exp1}}$ and $s_{t,3}, \ldots, s_{t,m} \in \mathcal{I}_{\mathrm{exp2}}$, according to the distribution $\varrho_{k,m}^{K,c}(M_t)$;
- If $t \neq t^*$, then $\mathcal{R}$ computes ψ_t as follows:

  $$\psi_t = y^{\nu_t/E_t} \bmod N,$$

  where $\nu_t = C(z_1 + \sum_{i=2}^{m} z_i s_{t,i}) + C_{t^*}(\beta - s_{t,2})$.
- Otherwise (i.e. $t = t^*$), $\mathcal{R}$ sets $s_{t,2} := \beta$, and $\mathcal{R}$ computes ψ_t as follows:

  $$\psi_t = y^{\nu_t/E_t} \bmod N,$$

  where $\nu_t = C(z_1 + \sum_{i=2}^{m} z_i s_{t,i})$.

Note that ν_t/E_t can be easily computed although the order $\mathrm{ord}(\mathrm{QR}_N)$ of QR_N is unknown as in (I-6) of *Type I*. Then $\mathcal{R}$ replies the tuple $((E_t, \boldsymbol{s}_t, r_t), \psi_t)$ to $\mathcal{A}$. Noting that $s_{t,1} = 1$, we have

$$\prod_{i=1}^{m} \alpha(a_i)^{s_{t,i}} = (y^{\beta C_{t^*}})^{s_{t,1}} (y^{-C_{t^*}})^{s_{t,2}} \prod_{i=1}^{m} (y^{Cz_i})^{s_{t,i}} = y^{\nu_t}, \tag{4.16}$$

in either case of $t \neq t^*$ and $t = t^*$. Then, ψ_t is a valid solution for the interpreted equation $\lambda_\alpha^{(t)} : x^{E_t} = \prod_{i=1}^{m} \alpha(a_i)^{s_{t,i}}$.

Challenge. Eventually, $\mathcal{A}$ outputs a tuple $((E^*, \boldsymbol{s}^*, r^*), \psi^*)$. $\mathcal{R}$ checks whether or not his guess is correct.

(II-8) If there exists no index $1 \leq t_0 \leq q$ such that $E^* = E_{t_0}$ and $s_2^* \neq s_{t_0,2}$, then this is not the case for *Type II*, and hence $\mathcal{R}$'s guess is wrong. Then $\mathcal{R}$ aborts. Assume that $\mathcal{R}$ correctly guesses *Type II*. Then, $\mathcal{R}$ next verifies whether or not t^* is correctly guessed, namely $r^* = r_{t^*}$ and $s_2^* \neq s_{t^*,2}$ hold. If the guess is wrong, then $\mathcal{R}$ aborts.

Suppose that all guesses are correct. $\mathcal{R}$ tries to extract z such that $z^e \equiv y \pmod N$ as follows. For the case where $\mathcal{A}$ succeeds, we have the following equation:

$$\begin{aligned} \psi^{*E^*} &= \alpha(a_1)\alpha(a_2)^{s_2^*} \prod_{i=3}^{m} \alpha(a_i)^{s_i^*} \\ &= (y^{Cz_1} \cdot y^{\beta C_{t^*}})(y^{Cz_2} \cdot y^{-C_{t^*}})^{s_2^*} \prod_{i=3}^{m} (y^{Cz_i})^{s_i^*} \\ &= y^{\nu^*}, \end{aligned} \tag{4.17}$$

where $\nu^* = C(z_1 + \sum_{i=2}^m z_i s_i^*) + C_{t^*}(\beta - s_2^*)$. Since the guess of t^* is correct, it follows from $e = h_{K,c}(r_{t^*}^{(j^*+1)}) \mid \prod_{j=1}^{\ell_H} h_{K,c}(r_{t^*}^{(j)}) = E_{t^*} = E^*$, $E_{t^*} \mid \prod_{t=1}^{q} E_t = C$ and Eq. (4.17) that

$$\left(\psi^{*E^*/e} \cdot y^{\nu^{*\prime}}\right)^e = y^{C_{t^*}(\beta - s_2^*)}, \tag{4.18}$$

where $\nu^{*\prime} = -(C/e)(z_1 + \sum_{i=2}^m z_i s_i^*)$.
(II-9) If $\gcd(e, C_{t^*}(\beta - s_2^*)) \neq 1$, then $\mathcal{R}$ aborts.
(II-10) Otherwise (namely, $\gcd(e, C_{t^*}(\beta - s_2^*)) = 1$), then $\mathcal{R}$ obtains z such that $z^e \equiv y \pmod N$ by Eq. (4.18) and Lemma 2.11.

Analysis of Type II. We analyze the success probability ϵ_{RSA} of *Reduction Algorithm $\mathcal{R}$ for Type II*. In the following series of games, as in the case of *Type I*, we adopt the behavior (II-1), (II-2), ..., (II-10) of $\mathcal{R}$ so that the first game Game 0 is the ordinary adaptive pseudo-free game, and the final game Compute RSA coincides with *Reduction Algorithm $\mathcal{R}$ for Type II* that we have just described. For each $0 \leq i \leq 6$, let $\Pr[\mathsf{Succ}_i]$ be the probability that the adversary $\mathcal{A}$ wins Game i.

Game 0. This game is the same as Game 0 of *Type I*.

$$\Pr[\mathsf{Succ}_0] > \epsilon. \tag{4.19}$$

Note that the randomly chosen index $N' \in \mathcal{N}(k)$ will be used until Game 6.

Game 1. This game proceeds in the same way as Game 0 with the exception that $\mathcal{R}$ executes (II-1), (II-2) and (II-3) in Setup phase. Namely $\mathcal{R}$ chooses the random strings r_t for each $1 \leq t \leq q$, and guesses the index t^* such that $r^* = r_{t^*}$ and $s_2^* \neq s_{t^*,2}$. Then, $\mathcal{R}$ aborts in Setup phase if there exists an index $1 \leq t \leq q$ such that $t \neq t^*$ and $r_t = r_{t^*}$.

Since the random index N, the assignment α' and the tuple (K, c) of the distribution family $\varrho_{k,m}^{K,c}$ are the same as Game 0. Moreover, the probability that $\mathcal{R}$ aborts in (II-3) is at most $q \cdot \left(1/2^{\ell_H}\right)$. Therefore, we have

$$\Pr[\mathsf{Succ}_1] \geq \Pr[\mathsf{Succ}_0] - \frac{q}{2^{\ell_H}}. \tag{4.20}$$

Game 2. This game proceeds in the same way as Game 1 with the exception that $\mathcal{R}$ also executes (II-4) in Setup phase. In this game, $\mathcal{R}$ chooses a random exponent $e' \in \mathbb{Z}_{\varphi(N')}^\times$ so that (N', e') is good instead of the given RSA exponent e, and sets $c := F_K(r^{*(j^*+1)}) \oplus e'$.

In the similar fashion to Game 1 for *Type I*, we have

$$\Pr[\mathsf{Succ}_2] \geq \Pr[\mathsf{Succ}_1] - \mathrm{negl}(k). \tag{4.21}$$

Game 3. This game proceeds in the same way as Game 2 with the exception that $\mathcal{R}$ also executes (II-5) and (II-6) in Setup phase. In this game, $\mathcal{R}$ computes each $E_t = H_{K,c}(r_t)$, where r_t's are random strings as chosen in Game 1, and $\mathcal{R}$ chooses a generator y' of $\mathrm{QR}_{N'}$ uniformly at random instead of the given RSA instance y. Then $\mathcal{R}$ fixes the assignment α as in (II-6).

Similarly to Game 2 of *Type I*, each ${y'}^{Cz_i}$ is distributed almost uniformly at random over $\mathrm{QR}_{N'}$. $\mathrm{MUL}_{{y'}^{\beta C_{t^*}},N'}$ and $\mathrm{MUL}_{{y'}^{-C_{t^*}},N'}$ are bijective by Lemma 4.3, because ${y'}^{\beta C_{t^*}}, {y'}^{-C_{t^*}} \in \mathrm{QR}_{N'}$. As in Game 2 of *Type I*, we may assume that $C \in \mathbb{Z}_{P'Q'}^\times$, and hence ${y'}^{Cz_1}$ and ${y'}^{Cz_2}$ are almost uniformly distributed over $\mathrm{QR}_{N'}$. Therefore, it looks that both $\alpha(a_1) = {y'}^{Cz_1} \cdot {y'}^{\beta C_{t^*}} = \mathrm{MUL}_{{y'}^{\beta C_{t^*}},N'}({y'}^{Cz_1})$ and $\alpha(a_2) = {y'}^{Cz_2} \cdot {y'}^{-C_{t^*}} = \mathrm{MUL}_{{y'}^{-C_{t^*}},N'}({y'}^{Cz_2})$ are distributed almost uniformly at random over $\mathrm{QR}_{N'}$. Therefore, each $\alpha(a_i)$ is distributed almost uniformly at random over $\mathrm{QR}_{N'}$. Thus, we have

$$\Pr[\mathsf{Succ}_3] \geq \Pr[\mathsf{Succ}_2] - \mathrm{negl}(k). \tag{4.22}$$

Game 4. This game proceeds in the same way as Game 3 with the exception that $\mathcal{R}$ also executes (II-7) in Equations queries phase. In this game, for each t-th queried

parameter M_t from $\mathcal{A}$, $\mathcal{R}$ chooses $s_{t,2} \in \mathcal{I}_{\mathrm{exp1}}$ and $s_{t,3}, \ldots, s_{t,m} \in \mathcal{I}_{\mathrm{exp2}}$, according to the distribution $\varrho_{k,m}^{K,c}(M_t)$. Then, when $t \neq t^*$, $\mathcal{R}$ computes $\psi_t = y'^{\nu_t/E_t} \bmod N'$ for $\nu_t = C(z_1 + \sum_{i=2}^m z_i s_{t,i}) + C_{t^*}(\beta - s_{t,2})$. Otherwise, $\mathcal{R}$ sets $s_{t,2} := \beta$, and computes $\psi_t = y'^{\nu_t/E_t} \bmod N'$ for $\nu_t = C(z_1 + \sum_{i=2}^m z_i s_{t,i})$. $\mathcal{R}$ finally replies the equation $\lambda^{(t)} = (E_t, \boldsymbol{s}_t)$, the corresponding answer ψ_t, and the random string r_t as chosen in Game 1.

For each t-th parameter M_t, $\mathcal{R}$ can correctly reply the solution ψ_t for the interpreted equation $\lambda_\alpha^{(t)}$ to $\mathcal{A}$ by Eq. (4.16). Recall that for the distribution $\varrho_{k,m}^{K,c}(M_{t^*})$, $s_{t^*,2} = \beta$ is chosen uniformly at random from $\mathcal{I}_{\mathrm{exp1}}$. Thus, we have

$$\Pr[\mathsf{Succ}_4] = \Pr[\mathsf{Succ}_3]. \tag{4.23}$$

Game 5. This game proceeds in the same way as Game 4 with the exception that $\mathcal{R}$ also executes (II-8) in Challenge phase. In this game, as in (II-8), $\mathcal{R}$ aborts if there exists no index $1 \leq t \leq q$ such that $E^* = E_t$ and $s_2^* \neq s_{t,2}$. $\mathcal{R}$ also aborts if $r^* \neq r_{t^*}$ or $s_2^* = s_{t^*,2}$.

As in Game 4 of *Type I*, the success probability that $\mathcal{R}$ correctly guesses *Type II* is $1/3$. Since t^* is chosen uniformly at random from the set $[1, q]$, the probability that $E^* = E_{t^*}$ and $s_2^* \neq s_{t^*,2}$ is $1/q$. If $E^* = E_{t^*}$, then $r^* = r_{t^*}$ with overwhelming probability, because $H_{K,c}$ is collision-resistant. Therefore, we have

$$\Pr[\mathsf{Succ}_5] \geq \frac{1}{3q} \Pr[\mathsf{Succ}_4] - \mathrm{negl}(k). \tag{4.24}$$

Game 6. This game proceeds in the same way as Game 5 with the exception that $\mathcal{R}$ also executes (II-9) in Challenge phase. $\mathcal{R}$ aborts if $\gcd(e', C_{t^*}(\beta - s_2^*)) \neq 1$.

We first show that $e' \nmid C_{t^*}$ with overwhelming probability as in Game 5 for *Type I*. Since $j^* = \max_{1 \leq t \leq q, t \neq t^*} j_t$ (the correct guess), for any $1 \leq t \leq q$ with $t \neq t^*$, we have that $r^{*(j^*+1)} \neq r_t^{(j^*+1)}$. Note that for any $r \in \{0,1\}^{\ell_H}$ and any $1 \leq j \leq \ell_H$ with $j \neq j^* + 1$, we have that $r^{*(j^*+1)} \neq r^{(j)}$ because these are of different length. Hence we have $r^{*(j^*+1)} \neq r_t^{(j)}$ for each $1 \leq t \leq q$ with $t \neq t^*$ and each $1 \leq j \leq \ell_H$. Therefore it follows that $e' = h_{K,c}(r^{*(j^*+1)}) \nmid \prod_{t=1, t \neq t^*}^q \prod_{j=1}^{\ell_H} h_{K,c}(r_t^{(j)}) = C_{t^*}$ with overwhelming probability, since $h_{K,c}$ is a collision-resistant prime-valued function.

We show that $e' \nmid (\beta - s_2^*)$ for any fixed $\mathcal{A}$'s output s_2^*. Note that (N', e') is good. Since $\beta = s_{t^*,2} \neq s_2^*$ in *Type II*, $\beta, s_{t^*,2} \in \mathcal{I}_{\mathrm{exp1}}$ and $e' \geq 2^{\ell_{\mathrm{exp1}}}$, we have $0 < |\beta - s_2^*| < 2^{\ell_{\mathrm{exp1}}} \leq e'$. Therefore, we have $e' \nmid (\beta - s_2^*)$. Thus, the probability

that $\mathcal{A}$ wins this game is

$$\Pr[\mathsf{Succ}_6] \geq \Pr[\mathsf{Succ}_5] - \mathrm{negl}(k). \tag{4.25}$$

Compute RSA. This is the final game in which the challenger coincides with *Reduction Algorithm $\mathcal{R}$ for Type II*. Namely, this game proceeds in the same way as Game 6, with the exception that $\mathcal{R}$ uses the input tuple (N, e, y) instead of (N', e', y'), and executes (II-10) in Challenge phase. Therefore, $\mathcal{R}$ computes z such that $z^e \equiv y \pmod{N}$ if $\gcd(e, C_{t^*}(\beta - s_2^*)) = 1$ as checked in Game 6.

Since $\mathcal{R}$ obtains z such that $z^e \equiv y \pmod{N}$ by Lemma 2.11 provided that $\mathcal{A}$ wins Game 6, we have

$$\epsilon_{\mathsf{RSA}} \geq \Pr[\mathsf{Succ}_6]. \tag{4.26}$$

Putting together Eqs. (4.19)–(4.26), we have

$$\epsilon_{\mathsf{RSA}} > \frac{1}{3q}\epsilon - \mathrm{negl}(k). \tag{4.27}$$

Type III. In this type, there exists an index $1 \leq t \leq q$ such that $E^* = E_t$. Moreover, for any such index t, $s_2^* = s_{t,2}$ holds. The idea for this type is almost the same as that for *Type II*. As in *Type II*, $\mathcal{R}$ guesses an index $1 \leq t^* \leq q$ such that $r^* = r_{t^*}$. Recall that $r^* = r_{t^*}$ with overwhelming probability provided that $E^* = E_{t^*}$. However, since $s_2^* = s_{t^*,2}$ in this type, an unfortunate event may happen in Game 6 of *Type II*, that is $e \mid C_{t^*}(\beta - s_2^*)$, with nonnegligible probability. In order to avoid this problem, we employ Lemma 2.5. It implies that for a nontrivial equation $\lambda^* : x^{E^*} = \prod_{i=1}^m a_i^{s_i^*}$ with respect to the queried equation set $\Lambda = \{\lambda^{(t)}\}_{t=1}^q$ from the adversary $\mathcal{A}$, it is guaranteed that $(E^*, s_1^*, s_2^*, \ldots, s_m^*) \neq (E_{t^*}, s_{t^*,1}, s_{t^*,2}, \ldots, s_{t^*,m})$. Since we have $E^* = E_{t^*}$, $s_1^* = s_{t^*,1} = 1$ and $s_2^* = s_{t^*,2}$, there exists an index $i^* \in [3, m]$ such that $s_{i^*}^* \neq s_{t^*,i^*}$. Therefore, $\mathcal{R}$ guesses an index $t^* \in [1, q]$ such that $r^* = r_{t^*}$ together with an index $i^* \in [3, m]$ such that $s_{i^*}^* \neq s_{t^*,i^*}$.

We now describe the algorithm $\mathcal{R}$ when *Type III* is guessed. Assume that an RSA instance (N, e, y) is given.

Reduction Algorithm $\mathcal{R}$ for Type III.

Setup. (III-1) $\mathcal{R}$ chooses random strings $r_1, r_2, \ldots, r_q \in_{\mathrm{U}} \{0,1\}^{\ell_H}$.

(III-2) $\mathcal{R}$ guesses an index $t^* \in_{\mathrm{U}} [1, q]$ such that $r^* = r_{t^*}$, and guesses an index $i^* \in_{\mathrm{U}} [3, m]$ such that $s^*_{i^*} \neq s_{t^*,i^*}$.
(III-3) If there exists an index $1 \leq t \leq q$ such that $t \neq t^*$ and $r_t = r_{t^*}$, then $\mathcal{R}$ aborts. Otherwise, $\mathcal{R}$ finds the longest length $j^* = \max_{1 \leq t \leq q, t \neq t^*} j_t$.

$\mathcal{R}$ now determines a distribution family $\varrho^{K,c}_{k,m}$ and an assignment $\alpha : A \to \mathbb{Z}_N^\times$ in the following way.
(III-4) $\mathcal{R}$ chooses a key $K \in_{\mathrm{U}} \mathcal{K}(k)$, and then sets $c := F_K(r^{*(j^*+1)}) \oplus e$ as in (I-3) of *Type I*.
(III-5) For each $1 \leq t \leq q$, let $E_t = H_{K,c}(r_t)$.
(III-6) $\mathcal{R}$ fixes the assignment $\alpha : A \to \mathbb{Z}_N^\times$ as follows:

- Choose $z_1, z_2, \ldots, z_m \in_{\mathrm{U}} \mathbb{Z}_{N^2}$, and $\beta \in_{\mathrm{U}} \mathbb{Z}_B$, where $B = 2^{\ell_{\mathrm{exp1}}} + 2^{\ell_{\mathrm{exp2}}} - 1$;
- Let $C = \prod_{t=1}^{q} E_t$, and $C_{t^*} = \prod_{t=1, t \neq t^*}^{q} E_t = C/E_{t^*}$; and
- Set $\alpha(a_1) := y^{Cz_1} \cdot y^{\beta C_{t^*}} \bmod N$, $\alpha(a_2) := y^{Cz_2} \cdot y^{-C_{t^*}} \bmod N$, $\alpha(a_{i^*}) := y^{Cz_{i^*}} \cdot y^{-C_{t^*}} \bmod N$ and $\alpha(a_i) := y^{Cz_i} \bmod N$ for each $3 \leq i \leq m$ with $i \neq i^*$.

We will show that each $\alpha(a_i)$ is distributed almost uniformly at random in QR_N later in the description of Game 3. Finally, $\mathcal{R}$ submits the index N, the assignment α and the tuple (K, c) to the adversary $\mathcal{A}$.

Equations queries. (III-7) When the t-th parameter $M_t \in \{0,1\}^\ell$ is queried from the adversary $\mathcal{A}$, "the challenger" $\mathcal{R}$ generates the tuple $((E_t, \boldsymbol{s}_t, r_t), \psi_t)$ as follows:

- $\mathcal{R}$ chooses $s_{t,2} \in \mathcal{I}_{\mathrm{exp1}}$ and $s_{t,3}, \ldots, s_{t,m} \in \mathcal{I}_{\mathrm{exp2}}$, according to the distribution $\varrho^{K,c}_{k,m}(M_t)$;
- If $t \neq t^*$, then $\mathcal{R}$ computes ψ_t as follows:

  $$\psi_t = y^{\nu_t / E_t} \bmod N,$$

  where $\nu_t = C(z_1 + \sum_{i=2}^{m} z_i s_{t,i}) + C_{t^*}(\beta - s_{t,2} - s_{t,i^*})$.
- Otherwise (i.e. $t = t^*$), $\mathcal{R}$ aborts if $\beta - s_{t,i^*} \notin \mathcal{I}_{\mathrm{exp1}}$. If $\beta - s_{t,i^*} \in \mathcal{I}_{\mathrm{exp1}}$, then sets $s_{t,2} := \beta - s_{t,i^*}$, and $\mathcal{R}$ computes ψ_t as follows:

  $$\psi_t = y^{\nu_t / E_t} \bmod N,$$

  where $\nu_t = C(z_1 + \sum_{i=2}^{m} z_i s_{t,i})$.

Then $\mathcal{R}$ replies the tuple $((E_t, \boldsymbol{s}_t, r_t), \psi_t)$ to $\mathcal{A}$. Noting that $s_{t,1} = 1$, we have

$$\prod_{i=1}^{m} \alpha(a_i)^{s_{t,i}} = (y^{\beta C_{t^*}})^{s_{t,1}} (y^{-C_{t^*}})^{s_{t,2}} (y^{-C_{t^*}})^{s_{t,i^*}} \prod_{i=1}^{m} (y^{Cz_i})^{s_{t,i}} = y^{\nu_t}, \tag{4.28}$$

in either case of $t = t^*$ and $t \neq t^*$. Then, ψ_t is a valid solution for the interpreted equation $\lambda_\alpha^{(t)} : x^{E_t} = \prod_{i=1}^m \alpha(a_i)^{s_{t,i}}$.

Challenge. Eventually, $\mathcal{A}$ outputs a tuple $((E^*, \boldsymbol{s}^*, r^*), \psi^*)$. $\mathcal{R}$ checks whether or not his guess is correct.

(III-8) If there exists no index $1 \leq t_0 \leq q$ such that $E^* = E_{t_0}$, or there exists an index $1 \leq t_0 \leq q$ such that $E^* = E_{t_0}$ and $s_2^* \neq s_{t_0,2}$, then this is not the case for *Type III*, and hence $\mathcal{R}$'s guess is wrong. Then $\mathcal{R}$ aborts. Assume that $\mathcal{R}$ correctly guesses *Type III*. Then, $\mathcal{R}$ next verifies whether or not both t^* and i^* are correctly guessed, namely $r^* = r_{t^*}$ and $s_{i^*}^* \neq s_{t^*,i^*}$ hold. If the guess is incorrect, then $\mathcal{R}$ aborts.

Suppose that all guesses are correct. $\mathcal{R}$ tries to extract z such that $z^e \equiv y \pmod N$ as follows. For the case where $\mathcal{A}$ succeeds, we have the following equation:

$$\psi^{*E^*} = y^{\nu^*}, \tag{4.29}$$

where $\nu^* = C(z_1 + \sum_{i=2}^m z_i s_i^*) + C_{t^*}(\beta - s_2^* - s_{i^*}^*)$. Since the guess of t^* is correct, it follows from $e = h_{K,c}(r_{t^*}^{(j^*+1)}) \mid \prod_{j=1}^{\ell_H} h_{K,c}(r_{t^*}^{(j)}) = E_{t^*} = E^*$, $E_{t^*} \mid \prod_{t=1}^q E_t = C$ and Eq. (4.29) that

$$\left(\psi^{*E^*/e} \cdot y^{\nu^{*\prime}}\right)^e = y^{C_{t^*}(\beta - s_2^* - s_{i^*}^*)}, \tag{4.30}$$

where $\nu^{*\prime} = -(C/e)(z_1 + \sum_{i=2}^m z_i s_i^*)$.

(III-9) If $\gcd(e, C_{t^*}(\beta - s_2^* - s_{i^*}^*)) \neq 1$, then $\mathcal{R}$ aborts.

(III-10) Otherwise (namely, $\gcd(e, C_{t^*}(\beta - s_2^* - s_{i^*}^*)) = 1$), then $\mathcal{R}$ obtains z such that $z^e \equiv y \pmod N$ by Eq. (4.30) and Lemma 2.11.

Analysis of Type III. We analyze the success probability ϵ_{RSA} of *Reduction Algorithm $\mathcal{R}$ for Type III*. In the following series of games, as in the case of the previous types, we adopt the behavior (III-1), (III-2), ..., (III-10) of $\mathcal{R}$ so that the first game Game 0 is the ordinary adaptive pseudo-free game, and the final

game Compute RSA coincides with *Reduction Algorithm $\mathcal{R}$ for Type III* that we have just described. For each $0 \leq i \leq 6$, let $\Pr[\mathsf{Succ}_i]$ be the probability that the adversary $\mathcal{A}$ wins Game i. Game 0 and Game 2 of this type are the same as those of *Type II*, respectively. Game 1 is almost the same as that of *Type II* with the exception that $\mathcal{R}$ also guesses $i^* \in_{\mathrm{U}} [3, m]$. Nevertheless, $\Pr[\mathsf{Succ}_1]$ is the same as that of *Type II*, because the guess of i^* does not cause the abortion in (III-3).

Game 3. This game proceeds in the same way as Game 2 with the exception that $\mathcal{R}$ also executes (III-5) and (III-6) in Setup phase. In this game, $\mathcal{R}$ computes each $E_t = H_{K,c}(r_t)$, where r_t's are random strings as chosen in Game 1, and chooses a generator y' of $\mathrm{QR}_{N'}$ uniformly at random instead of the given RSA instance y. Then $\mathcal{R}$ fixes the assignment α as in (III-6).

Similarly to Game 2 of *Type II*, each ${y'}^{Cz_i}$, $\alpha(a_1) = {y'}^{Cz_1} \cdot {y'}^{\beta C_{t^*}}$ and $\alpha(a_2) = {y'}^{Cz_2} \cdot {y'}^{-C_{t^*}}$ are distributed almost uniformly at random over $\mathrm{QR}_{N'}$, respectively. $\alpha(a_{i^*}) = {y'}^{Cz_{i^*}} \cdot {y'}^{-C_{t^*}} \bmod N'$ is also distributed almost uniformly at random over $\mathrm{QR}_{N'}$. Therefore, each $\alpha(a_i)$ is distributed almost uniformly at random over $\mathrm{QR}_{N'}$. We have

$$\Pr[\mathsf{Succ}_3] \geq \Pr[\mathsf{Succ}_2] - \mathrm{negl}(k). \tag{4.31}$$

Game 4. This game proceeds in the same way as Game 3 with the exception that $\mathcal{R}$ also executes (III-7) in Equations queries phase. In this game, for each t-th parameter M_t queried from $\mathcal{A}$, $\mathcal{R}$ chooses $s_{t,2} \in \mathcal{I}_{\mathrm{exp1}}$ and $s_{t,3}, \ldots, s_{t,m} \in \mathcal{I}_{\mathrm{exp2}}$, according to the distribution $\varrho_{k,m}^{K,c}(M_t)$. Then, when $t \neq t^*$, $\mathcal{R}$ sets $\nu_t := C(z_1 + \sum_{i=2}^m z_i s_{t,i}) + C_{t^*}(\beta - s_{t,2} - s_{t,i^*})$. Otherwise, $\mathcal{R}$ aborts if $\beta - s_{t,i^*} \notin \mathcal{I}_{\mathrm{exp1}}$, or $\mathcal{R}$ sets $s_{t,2} := \beta - s_{t,i^*}$ and $\nu_t := C(z_1 + \sum_{i=2}^m z_i s_{t,i})$ otherwise. $\mathcal{R}$ computes $\psi_t = {y'}^{\nu_t/E_t} \bmod N'$, and then replies the equation $\lambda^{(t)} = (E_t, \boldsymbol{s}_t)$, the corresponding answer ψ_t, and the random string r_t as chosen in Game 1.

For each t-th parameter M_t, $\mathcal{R}$ can correctly reply the solution ψ_t for the interpreted equation $\lambda_\alpha^{(t)}$ to $\mathcal{A}$ by Eq. (4.28) unless $\mathcal{R}$ aborts at (III-7).

We estimate the probability that $\mathcal{R}$ aborts. For any fixed $s_{t^*,i^*} \in \mathcal{I}_{\mathrm{exp2}}$, there are $2^{\ell_{\mathrm{exp1}}}$ values $\beta \in \mathbb{Z}_B$, where $B = 2^{\ell_{\mathrm{exp1}}} + 2^{\ell_{\mathrm{exp2}}} - 1$, such that $0 \leq \beta - s_{t^*,i^*} \leq 2^{\ell_{\mathrm{exp1}}} - 1$, namely $\beta - s_{t^*,i^*} \in \mathcal{I}_{\mathrm{exp1}}$. (Note that we have $s_{t^*,i^*} \leq 2^{\ell_{\mathrm{exp2}}} - 1$ and hence $s_{t^*,i^*} + 2^{\ell_{\mathrm{exp1}}} - 1 < B$.) Since $\ell_{\mathrm{exp1}} = \ell_{\mathrm{exp2}} + \ell_{\mathrm{diff}}$, for any $s_{t^*,i^*} \in \mathcal{I}_{\mathrm{exp2}}$, we

have

$$\begin{aligned}\Pr[\mathcal{R} \text{ aborts}] &= \Pr_{\beta \in_{\mathrm{U}} \mathbb{Z}_B}[s_{t^*,2} = \beta - s_{t^*,i^*} \notin \mathcal{I}_{\mathrm{exp1}}] \\ &= 1 - \frac{2^{\ell_{\mathrm{exp1}}}}{B} \\ &= \frac{2^{\ell_{\mathrm{exp2}}} - 1}{2^{\ell_{\mathrm{exp1}}} + 2^{\ell_{\mathrm{exp2}}} - 1} \\ &< \frac{2^{\ell_{\mathrm{exp2}}}}{2^{\ell_{\mathrm{exp1}}} + 2^{\ell_{\mathrm{exp2}}} - 1} \\ &= \frac{1}{2^{\ell_{\mathrm{diff}}} + 1 - 2^{-\ell_{\mathrm{exp2}}}} \\ &< 2^{-\ell_{\mathrm{diff}}} \\ &= \mathrm{negl}(k).\end{aligned}$$

We show that for any fixed $s_{t^*,i^*} \in \mathcal{I}_{\mathrm{exp2}}$, the distribution of the random variable $s_{t^*,2} = \beta - s_{t^*,i^*}$, where $\beta \in_{\mathrm{U}} \mathbb{Z}_B$, is statistically close to the distribution of $s_2 \in_{\mathrm{R}} \mathcal{E}_{k,m}^{K,c}$ (namely $s_2 \in_{\mathrm{U}} \mathcal{I}_{\mathrm{exp1}}$) over the interval $\mathcal{I} = [-s_{t^*,i^*}, (B-1) - s_{t^*,i^*}]$. Note that $\mathcal{I} \supseteq \mathcal{I}_{\mathrm{exp1}}$. For any $a \in \mathcal{I}$, we denote by $P_1(a) := \Pr_{\beta \in_{\mathrm{U}} \mathbb{Z}_B}[\beta - s_{t^*,i^*} = a]$ and $P_2(a) := \Pr_{s_2 \in_{\mathrm{U}} \mathcal{I}_{\mathrm{exp1}}}[s_2 = a]$. Then, we have $P_1(a) = 1/B$, and $P_2(a) = 1/2^{\ell_{\mathrm{exp1}}}$ if $a \in \mathcal{I}_{\mathrm{exp1}}$ and $P_2(a) = 0$ otherwise. Therefore, the statistical distance between $s_{t^*,2}$ and s_2 over $\mathcal{I}$ is

$$\begin{aligned}\frac{1}{2}\sum_{a \in \mathcal{I}} |P_1(a) - P_2(a)| &= \frac{2^{\ell_{\mathrm{exp1}}}}{2}\left|\frac{1}{B} - \frac{1}{2^{\ell_{\mathrm{exp1}}}}\right| + \frac{B - 2^{\ell_{\mathrm{exp1}}}}{2}\left|\frac{1}{B} - 0\right| \\ &= 1 - \frac{2^{\ell_{\mathrm{exp1}}}}{B} \\ &< 2^{-\ell_{\mathrm{diff}}} \\ &= \mathrm{negl}(k).\end{aligned}$$

Thus we have

$$\Pr[\mathsf{Succ}_4] \geq \Pr[\mathsf{Succ}_3] - \mathrm{negl}(k). \tag{4.32}$$

Game 5. This game proceeds in the same way as Game 4 with the exception that $\mathcal{R}$ also executes (III-8) in Challenge phase. In this game, as in (III-8), $\mathcal{R}$ aborts if there exists no index $1 \leq t_0 \leq q$ such that $E^* = E_{t_0}$ or there exists an index $1 \leq t_0 \leq q$ such that $E^* = E_{t_0}$ and $s_2^* \neq s_{t,2}$. Moreover, $\mathcal{R}$ aborts if $r^* \neq r_{t^*}$ or $s_{i^*}^* = s_{t^*,i^*}$.

As in Game 4 of *Type I*, the probability that $\mathcal{R}$ correctly guesses *Type III* is $1/3$. Since t^* is chosen uniformly at random from the set $[1, q]$, the probability that $E^* = E_{t^*}$ is $1/q$. If $E^* = E_{t^*}$, then $r^* = r_{t^*}$ with overwhelming probability, because $H_{K,c}$ is collision-resistant. Since i^* is chosen uniformly at random from the set $[3, m]$, the probability of $s^*_{i^*} \neq s_{t^*,i^*}$ is at least $1/m$. Therefore, we have

$$\Pr[\mathsf{Succ}_5] \geq \frac{1}{3qm} \Pr[\mathsf{Succ}_4] - \mathrm{negl}(k). \tag{4.33}$$

Game 6. This game proceeds in the same way as Game 5 with the exception that $\mathcal{R}$ also executes (III-9) in Challenge phase. Namely $\mathcal{R}$ aborts if $\gcd(e', C_{t^*}(\beta - s^*_2 - s^*_{i^*})) \neq 1$.

In a similar manner to Game 6 of *Type II*, we have $e' \nmid C_{t^*}$ with overwhelming probability.

We show that $e' \nmid (\beta - s^*_2 - s^*_{i^*})$ for any fixed $\mathcal{A}$'s output s^*_2. It follows from $\beta = s_{t^*,2} + s_{t^*,i^*}$ and $s^*_2 = s_{t^*,2}$ in this type that $\beta - s^*_2 - s^*_{i^*} = s_{t^*,i^*} - s^*_{i^*}$. Note that (N', e') is good. Since $s^*_{i^*} \neq s_{t^*,i^*}$ (the correct guess), $s_{t^*,i^*}, s^*_{i^*} \in \mathcal{I}_{\mathrm{exp2}}$ and $e' \geq 2^{\ell_{\mathrm{exp1}}}$, this implies that $0 < |s_{t^*,i^*} - s^*_{i^*}| = |\beta - s^*_2 - s^*_{i^*}| < 2^{\ell_{\mathrm{exp2}}} \leq 2^{\ell_{\mathrm{exp1}}} \leq e'$. Therefore, we have $e' \nmid (\beta - s^*_2 - s^*_{i^*})$. Thus, the probability that $\mathcal{A}$ wins this game is

$$\Pr[\mathsf{Succ}_6] \geq \Pr[\mathsf{Succ}_5] - \mathrm{negl}(k). \tag{4.34}$$

Compute RSA. This is the final game whose challenger coincides with *Reduction Algorithm $\mathcal{R}$ for Type III*. Namely, this game proceeds in the same way as Game 6, with the exception that $\mathcal{R}$ uses the input tuple (N, e, y) instead of (N', e', y'), and executes (III-10) in Challenge phase. Therefore, $\mathcal{R}$ computes z such that $z^e \equiv y \pmod{N}$ if $\gcd(e, C_{t^*}(\beta - s^*_2 - s^*_{i^*})) = 1$ as checked in Game 6.

Since $\mathcal{R}$ obtains z such that $z^e \equiv y \pmod{N}$ by Lemma 2.11 provided that $\mathcal{A}$ wins Game 6, we have

$$\epsilon_{\mathsf{RSA}} \geq \Pr[\mathsf{Succ}_6]. \tag{4.35}$$

Putting together Eqs. (4.19)–(4.21) and (4.31)–(4.35), we have

$$\epsilon_{\mathsf{RSA}} > \frac{1}{3qm}\epsilon - \mathrm{negl}(k). \tag{4.36}$$

The Success Probability of $\mathcal{R}$. For each type, the distribution of $\mathcal{R}$'s outputs is statistically close to the distribution of the outputs from the challenger of the ordinary adaptive pseudo-free game. Namely the distribution of the tuple $(N, \alpha, (K, c))$ submitted from $\mathcal{R}$ is statistically close to that submitted from the challenger of the ordinary game. Moreover, for any parameter M_t queried from $\mathcal{A}$, the distribution of the tuple $(E_t, \boldsymbol{s}_t, r_t)$ replied from $\mathcal{R}$ is statistically close to the distribution $\varrho_{k,m}^{K,c}(M_t)$. Therefore, the PPT adversary $\mathcal{A}$ cannot distinguish the type chosen by the algorithm $\mathcal{R}$. It therefore follows from Eq. (4.15) for *Type I*, Eq. (4.27) for *Type II* and Eq. (4.36) for *Type III* that

$$\epsilon_{\mathsf{RSA}} > \frac{2}{9mq\ell_H}\epsilon - \mathrm{negl}(k)$$

holds for any *Type I–III*, provided that (N, e) is good: this lower bound can be obtained by simply taking the trivial common lower bound for the bounds given in Eqs. (4.15), (4.27) and (4.36).

Recall that, as we are under the $(\epsilon_{\mathsf{RSA}}, \tau_{\mathsf{Good}})$-RSA assumption, (N, e) is a good pair with probability at least $1/\tau_{\mathsf{Good}}$. Therefore, the probability that $\mathcal{R}$ extracts $z \in \mathbb{Z}_N^\times$ such that $z^e \equiv y \pmod N$ is

$$\begin{aligned}\epsilon_{\mathsf{RSA}} &> \frac{1}{\tau_{\mathsf{Good}}}\left(\frac{2}{9mq\ell_H}\epsilon - \mathrm{negl}(k)\right)\\ &\geq \frac{2}{9mq\ell_H\tau_{\mathsf{Good}}}\epsilon - \mathrm{negl}(k).\end{aligned}$$

This implies that

$$\epsilon < \frac{9}{2}mq\ell_H\tau_{\mathsf{Good}}\epsilon_{\mathsf{RSA}} + \mathrm{negl}(k).$$

This completes the proof. □

4.5 Concluding Remarks

In this chapter, we have proposed a new class of parametric distributions for adaptive pseudo-free groups, and have shown that the RSA group family $\left\{\mathbb{Z}_N^\times\right\}$ is adaptive pseudo-free with respect to $\left\{\varrho^{K,c}\right\}_{K,c}$ under the RSA assumption provided that N is a product of two distinct safe primes. Therefore, RSA-based signature schemes could be constructed from the adaptive pseudo-freeness by

using our family $\left\{\varrho^{K,c}\right\}_{K,c}$ of parametric distributions. It is also expected that the adaptive pseudo-freeness of $\left\{\mathbb{Z}_N^\times\right\}$ is proven for an appropriate parametric distribution corresponding to the applied cryptographic assumption.

Chapter 5

Conclusion

In this thesis, we have discussed the flexibility of the notion of the adaptive pseudo-free groups, and the applicability of the existing parametric distributions. Especially, we have studied the property of the adaptive pseudo-freeness of the RSA group $\mathbb{Z}_N^\times$ with respect to several cryptographic assumptions.

In Chapter 3, we have shown two impossibility results on the adaptive pseudo-freeness of the RSA group $\mathbb{Z}_N^\times$. First, we have given a negative circumstantial evidence for the question whether or not $\mathbb{Z}_N^\times$ is strongly-adaptive pseudo-free. More precisely, we have shown that the RSA group $\mathbb{Z}_N^\times$ cannot be proven to be strongly-adaptive pseudo-free from the strong RSA (SRSA, for short) assumption via algebraic reductions, as long as the SRSA assumption holds. In other words, the strong adaptive pseudo-freeness of $\mathbb{Z}_N^\times$ cannot be shown under the SRSA assumption, by employing only current proof techniques which are frequently used in ordinary security proofs. Since the SRSA assumption is one of the strongest assumption, this implies that the strong adaptive pseudo-freeness for the RSA group $\mathbb{Z}_N^\times$ may be far from feasibility. Hence it is reasonable to use parametric distributions to construct a concrete adaptive pseudo-free group.

As the second result of Chapter 3, we have again given a negative circumstantial evidence for the question whether or not the adaptive pseudo-freeness of $\mathbb{Z}_N^\times$ can be proven from some assumption other than the SRSA assumption, by using the parametric distributions of Catalano-Fiore-Warinschi [13]. Namely, we have shown that it cannot be proven from the RSA assumption via algebraic reductions, as long as the RSA assumption holds.

As an application of the second result, we have shown that the strongly existential unforgeability of the SRSA-based signature schemes proposed by [11,

14, 18, 22, 27, 49] against the chosen message attack cannot be proven from the RSA assumption via algebraic reductions, as long as the RSA assumption holds. In particular, we have also shown that even existential unforgeability of these signatures against the key only attack may not be proven from the RSA assumption. These impossibility results on the SRSA-based signatures indicates that the adaptive pseudo-free group is useful to discuss whether or not the security of a cryptographic scheme is provable from a specific assumption.

Our second result means that another parametric distribution is required to prove the adaptive pseudo-freeness of $\mathbb{Z}_N^\times$ from the RSA assumption. In Chapter 4, we have developed the new parametric distributions $\varrho^{K,c}$ as the third result. Namely, we have shown that the adaptive pseudo-freeness of $\mathbb{Z}_N^\times$ can be proven from the RSA assumption by using our parametric distributions $\varrho^{K,c}$. Therefore, RSA-based signature schemes could be constructed from the adaptive pseudo-freeness via the parametric distributions $\varrho^{K,c}$. It is also expected that the adaptive pseudo-freeness of $\mathbb{Z}_N^\times$ is proven with respect to an appropriate parametric distribution corresponding to the applied cryptographic assumption.

We now describe several questions that emerged in this thesis and are left unresolved. The first one is whether or not a cryptographic scheme other than signature schemes can be constructed from the adaptive pseudo-freeness. If such a scheme can be obtained, one may be able to yield a new cryptographic scheme only by showing the adaptive pseudo-freeness of specific groups.

The second one is to explore a parametric distribution such that the adaptive pseudo-freeness of $\mathbb{Z}_N^\times$ with respect to the parametric distribution can be proven from some assumption other than the SRSA assumption and the RSA assumption. If such a parametric distribution is constructed, cryptographic schemes whose security is proven from the employed assumption can be yielded from the adaptive pseudo-freeness by using the constructed parametric distribution.

Finally, the third one is whether or not there exists an example of the adaptive pseudo-free group other than the RSA group $\mathbb{Z}_N^\times$. This question have been already considered by Catalano, Fiore and Warinschi [13]. If this question is resolved affirmatively, new cryptographic schemes will be obtained over such a group through the adaptive pseudo-freeness.

Bibliography

[1] M. Abe, J. Groth and M. Ohkubo, "Separating Short Structure-Preserving Signatures from Non-Interactive Assumptions," Proc. ASIACRYPT 2011, LNCS 7073, pp. 628–646, 2011.

[2] M. Abe, K. Haralambiev and M. Ohkubo, "Group to Group Commitments Do Not Shrink," Proc. EUROCRYPT 2012, LNCS 7237, pp. 301–317, 2012.

[3] M. Agrawal, N. Kayal and N. Saxena, "PRIMES Is in P," Annals of Mathematics, vol. 160, no. 2, pp. 781–793, 2004.

[4] M. Ajtai and C. Dwork, "A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence," In ACM Symp. on Theory of Computing, pp. 284–293, 1997.

[5] M. Anokhin, "Constructing a Pseudo-Free Family of Finite Computational Groups under the General Integer Factoring Intractability Assumption," Electronic Colloquium on Computational Complexity, report no. 114, 2012.

[6] N. Barić and B. Pfitzmann, "Collision-Free Accumulators and Fail-Stop Signature Schemes without Trees," Proc. EUROCRYPT 1997, LNCS 1233, pp. 480–494, 1997.

[7] M. Bellare, C. Namprempre, D. Pointcheval and M. Semanko, "The One-More-RSA-Inversion Problems and the Security of Chaum's Blind Signature Scheme," J. Cryptology, vol. 16, no. 3, pp. 185–215, 2008. (Conference version: Financial Cryptography 2001, LNCS 2339, pp. 319–338, 2002.)

[8] M. Bellare and P. Rogaway, "Optimal Asymmetric Encryption," Proc. EUROCRYPT 1994, LNCS 950, pp. 92–111, 1995.

[9] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing," Proc. CRYPTO 2001, LNCS 2139, pp. 213–229, 2001.

[10] E. Bresson, J. Monnerat and D. Vergnaud, "Separation Results on the 'One-More' Computational Problems," Proc. CT-RSA 2008, LNCS 4964, pp. 71–87, 2008.

[11] J. Camenisch and A. Lysyanskaya, "A Signature Scheme with Efficient Protocols," Proc. SCN 2003, LNCS 2576, pp. 268–289, 2003.

[12] D. Catalano and D. Fiore, "Vector Commitments and Their Applications," Proc. PKC 2013, LNCS 7778, pp. 55–72, 2013.

[13] D. Catalano, D. Fiore and B. Warinschi, "Adaptive Pseudo-Free Groups and Applications," Proc. EUROCRYPT 2011, LNCS 6632, pp. 207–223, 2011. (the full version is available at `https://eprint.iacr.org/2011/053.pdf`.)

[14] R. Cramer and V. Shoup, "Signature Schemes Based on the Strong RSA Assumption," Proc. ACM CCS 1999, pp. 46–51, 1999.

[15] R. Cramer and V. Shoup, "Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack," SIAM J. Comput., vol. 33, no. 1, pp. 167–226, 2003.

[16] W. Diffie and M. Hellman, "New Directions in Cryptography," IEEE Trans. on Information Theory, vol. 22, no. 6, pp. 644–654, 1976.

[17] G. Everest and T. Ward, *An Introduction to Number Theory*, 2005.

[18] M. Fischlin, "The Cramer-Shoup Strong-RSA Signature Scheme Revisited," Proc. PKC 2003, LNCS 2567, pp. 116–129, 2002.

[19] E. Fujisaki and T. Okamoto, "Statistical Zero Knowledge Protocols to Prove Modular Polynomial Relations," Proc. CRYPTO 1997, LNCS 1294, pp.16–30, 1997.

[20] E. Fujisaki, T. Okamoto, D. Pointcheval and J. Stern, "RSA-OAEP Is Secure under the RSA Assumption," J. Cryptology, vol. 17, no. 2, pp. 81–104, 2004.

[21] S. Garg, R. Bhaskar and S.V. Lokam, "Improved Bounds on Security Reductions for Discrete Log Based Signatures," Proc. CRYPTO 2008, LNCS 5157, pp. 93–107, 2008.

[22] R. Gennaro, S. Halevi and T. Rabin, "Secure Hash-and-Sign Signatures without the Random Oracle," Proc. EUROCRYPT 1999, LNCS 1592, pp. 123–139, 1999.

[23] S. Goldwasser, S. Micali and R.L. Rivest, "A Digital Signature Scheme Secure against Adaptive Chosen-Message Attacks," SIAM J. Comput., vol. 17, no. 2, pp. 281–308, 1988.

[24] G. Hanaoka, T. Matsuda and J.C.N. Schuldt, "On the Impossibility of Constructing Efficient Key Encapsulation and Programmable Hash Functions in Prime Order Groups," Proc. CRYPTO 2012, LNCS 7417, pp. 812–831, 2012.

[25] S. Hasegawa, S. Isobe, H. Shizuya and K. Tashiro, "On the Pseudo-Freeness and the CDH Assumption," International Journal of Information Security, vol. 8, no. 5, pp. 347–355, 2009.

[26] T. Hirano and K. Tanaka, "Variations on Pseudo-Free Groups," Research Reports, Series C: Computer Science, C-239, Tokyo Institute of Technology, 2007.

[27] D. Hofheinz and E. Kiltz, "Programmable Hash Functions and Their Applications," J. Cryptology, vol. 25, no. 3, pp. 484–527, 2012. (Conference version: CRYPTO 2008, LNCS 5157, pp. 21–38, 2008.)

[28] D. Hofheinz, T. Jager and E. Kiltz, "Short Signatures from Weaker Assumptions," Proc. ASIACRYPT 2011, LNCS 7073, pp. 647–666, 2011.

[29] S. Hohenberger, "The Cryptographic Impact of Groups with Infeasible Inversion," Master's thesis, EECS Dept. MIT, 2003.

[30] S. Hohenberger and B. Waters, "Short and Stateless Signatures from the RSA Assumption," Proc. CRYPTO 2009, LNCS 5677, pp. 654–670, 2009.

[31] M.P. Jhanwar and R. Barua, "Sampling from Signed Quadratic Residues: RSA Group Is Pseudofree," Proc. INDOCRYPT 2009, LNCS 5922, pp. 233–247, 2009.

[32] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, Chapman & Hall/CRC Press, 2007.

[33] S. Lim and H.S. Lee, "A Short and Efficient Redactable Signature Based on RSA," ETRI Journal, vol. 33, no. 4, pp. 621–628, 2011.

[34] D. Micciancio, "The RSA Group is Pseudo-Free," J. Cryptology, vol. 23, no. 2, pp. 169–186, 2010. (Conference version: EUROCRYPT 2005, LNCS 3494, pp. 387–403, 2005.)

[35] D. Naccache, D. Pointcheval and J. Stern, "Twin Signatures: An Alternative to the Hash-and-Sign Paradigm," Proc. ACM CCS 2001, pp. 20–27, 2001.

[36] R. Nishimaki, E. Fujisaki and K. Tanaka, "A Multi-Trapdoor Commitment Scheme from the RSA Assumption," IEICE Trans. Fundamentals, Special Section on Cryptography and Information Security, vol. E95-A, no. 1, pp. 176–184, 2012. (Conference version: Proc. ACISP 2010, LNCS 6168, pp. 182–199, 2010.)

[37] P. Paillier and D. Vergnaud, "Discrete-Log-Based Signatures May Not Be Equivalent to Discrete Log," Proc. ASIACRYPT 2005, LNCS 3788, pp. 1–20, 2005.

[38] C. Rackoff and D.R. Simon, "Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack," Proc. CRYPTO 1991, LNCS 576, pp. 433–444, 1992.

[39] R.L. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Communications of the ACM, vol. 21, no. 2, pp. 120–126, 1978.

[40] R.L. Rivest, "On the Notion of Pseudo-Free Groups," Proc. TCC 2004, LNCS 2951, pp. 505–521, 2004.

[41] S. Schäge and J. Schwenk, "A New RSA-Based Signature Scheme," Proc. AFRICACRYPT 2010, LNCS 6055, pp. 1–15, 2010.

[42] S. Schäge, "Strong Security from Probabilistic Signature Schemes," Proc. PKC 2012, LNCS 7293, pp. 84–101, 2012.

[43] Y. Seurin, "On the Exact Security of Schnorr-Type Signatures in the Random Oracle Model," Proc. EUROCRYPT 2012, LNCS 7237, pp. 554–571, 2012.

[44] A. Shamir, "On the Generation of Cryptographically Strong Pseudorandom Sequences," ACM Trans. on Computer Systems, vol. 1, no. 1, pp. 38–44, 1983.

[45] J.L. Villar, "Optimal Reductions of Some Decisional Problems to the Rank Problem," Proc. ASIACRYPT 2012, LNCS 7658, pp. 80–97, 2012.

[46] Z. Wang and W. Chen, "An ID-Based Online/Offline Signature Scheme without Random Oracles for Wireless Sensor Networks," Personal and Ubiquitous Computing, vol. 17, no. 5, pp. 837–841, 2013.

[47] S. Yamada, G. Hanaoka and N. Kunihiro, "Space Efficient Signature Schemes from the RSA Assumption," Proc. PKC 2012, LNCS 7293, pp. 102–119, 2012.

[48] H. Zhu, "New Digital Signature Scheme Attaining Immunity to Adaptive Chosen-Message Attack," Chinese Journal of Electronics, vol. 10, no. 4, pp. 484–486, 2001.

[49] H. Zhu, "A Formal Proof of Zhu's Signature Scheme," Cryptology ePrint Archive, Report 2003/155, http://eprint.iacr.org/, 2003.

List of Publications

[1] M. Fukumitsu, S. Hasegawa, S. Isobe, E. Koizumi and H. Shizuya, "Toward Separating the Strong Adaptive Pseudo-Freeness from the Strong RSA Assumption," 18th Australasian Conference on Information Security and Privacy (ACISP2013), Brisbane, LNCS 7959, pp. 72–87, July 2013.

[2] M. Fukumitsu, S. Hasegawa, S. Isobe and H. Shizuya, "The RSA Group Is Adaptive Pseudo-Free under the RSA Assumption," IEICE Trans. Fundamentals, Special Section on Cryptography and Information Security, vol. E97-A, no. 1, pp. 200–214, 2014.