博士論文

New Tools for Factoring-based Cryptography
(素因数分解に基づく暗号における新たな手法)

山川高志

Copyright © 2017, Takashi Yamakawa.

Abstract

In security proofs of cryptographic schemes, various kinds of assumptions have been used. Among those assumptions, the hardness of the factoring problem, called the factoring assumption, is one of the most reliable assumptions. In this dissertation, we study constructions of factoring-based cryptographic schemes. First we study a cryptographic primitive called a self-bilinear map. We define a relaxed version of a self-bilinear map, which we call a self-bilinear maps with auxiliary information (AI-SBM). We construct an AI-SBM based on the factoring assumption and the existence of indistinguishability obfuscation (iO). We also show that AI-SBM can be used for constructing cryptographic schemes including multiparty key exchange, broadcast encryption, attribute-based encryption and homomorphic signatures. As a side result, we construct a somewhat homomorphic encryption for NC_1 circuits based on the Φ-hiding assumption and the existence of iO. Second we study a cryptographic primitive called lossy trapdoor function (LTDF). We define a relaxed version of an LTDF which we call an adversary-dependent lossy trapdoor functions (ad-LTDF). Then we show that in many applications of LTDFs, we can replace LTDFs with ad-LTDFs. Moreover, we give a construction of ad-LTDFs based on the factoring assumption w.r.t. semi-smooth RSA subgroup moduli (SS moduli), which is a special type of RSA moduli introduced by Groth. As a result, we almost automatically obtain new constructions of a collision resistant hash function, CPA secure PKE scheme and DPKE scheme based on the same assumption. Especially, our DPKE scheme is the first scheme that satisfies the security notion called the PRIV security for block sources proposed by Boldyreva et al. solely based on the factoring assumption. Besides direct applications of ad-LTDFs, we construct a CCA secure PKE scheme based on the factoring assumption w.r.t. SS moduli whose ciphertext overhead is the shortest among schemes based on the same assumption.

Contents

Chapter 1

Introduction

1.1 Background

Cryptography is indispensable to secure information transmissions. While the history of cryptography dates back to more than hundreds years ago, the concept of public key encryption (PKE), proposed by Diffie and Hellman [DH76] in 1976, has brought about a revolution in cryptography. In “classical” cryptography before that, it is assumed that a sender and a receiver share a common key, which is used both for encryption and decryption. On the other hand, in a PKE scheme, there are two types of keys called an encryption key and a decryption key. An important feature of PKE is that the security is ensured even if an encryption key is made public. By this feature, anyone can encrypt a massage to generate a ciphertext without any secret information, while only a party who knows a decryption key can decrypt the ciphertext to recover the message. This enable us to securely communicate with many and unspecified parties on the Internet. Moreover, it turns out that the idea of PKE can be used for constructing many other cryptographic primitives such as digital signatures, identification, commitments, etc.

Security proof and hardness assumptions. For ensuring security of cryptographic schemes, a commonly used approach is to give security proofs for those schemes. That is, we prove that breaking the scheme is as hard as solving a certain mathematical problem that is known to be hard to solve. As a result, we ensure the security of the scheme under the assumption that the underlying mathematical problem is really hard to solve. This approach gives us a strong evidence of security because the security is proven in a mathematical way as long as the underlying assumption is true. On the other hand, if the underlying assumption is false (i.e., the underlying mathematical problem is easy to solve), the security of the scheme is no longer ensured. Therefore it is important to reduce the security of a cryptographic scheme to a reliable assumption.

Nowadays, various kinds of assumptions are used for security proofs including the factoring [RSA78, Rab79], discrete-logarithm [DH76], paring-based [Jou00, SOK01, BF01], lattice-based [Ajt96, Reg05] assumptions etc. Among them, the factoring assumption, which claims the hardness of factorizing large composite number, is one of the most reliable assumptions because no efficient algorithm to solve the integer factoring problem is known so far though intensive researches on that topic have been done over several decades [Pol74, Pol75, Len87, CP05]. Therefore if the security of a cryptographic scheme is proven under the factoring assumption, then we obtain a strong evidence on the security of the scheme. Thus in this dissertation, we study constructions of factoring-based cryptographic schemes.

Here, we remark the definition of the term “factoring-based” used in this dissertation. When we say “factoring-based scheme”, we do not always mean that the security of the scheme is reduced to the factoring assumption, but we mean the scheme is reduced to any assumption which is related to the factoring assumption. These “factoring-related assumptions” include the RSA [RSA78], strong RSA [BP97], quadratic residuosity (QR) [GM82], decision composite residuosity(DCR) [Pai99], Φ-hiding [CMS99] assumptions etc. Though there is no known algorithm to break these assumptions without breaking the factoring assumption, there is no rigorous reduction from any of these assumptions to the factoring assumption. Therefore we distinguish factoring-based constructions and those reduced to the factoring assumption.

1.2 Cryptographic primitives

Here, we review current status of various cryptographic primitives, mainly focusing on factoring-based constructions. We note that since the theme of this dissertation is factoring-based cryptography, we omit some important results that is not related to factoring.

Public key encryption. The first PKE scheme was proposed by Rivest, Shamir and Adleman in 1978 [RSA78], which is called the RSA scheme. The description of the RSA scheme is as follows. A public key consists of a product $N = PQ$ of two primes P and Q and a public exponent $e \in \mathbb{Z}$. A secret key is a secret exponent d, such that $ed \equiv 1 \mod (P-1)(Q-1)$. For encrypting a message $M \in \mathbb{Z}_N$, we compute $C = M^e \mod N$ and output C as a ciphertext. The decryption is done as $M = C^d$. The correctness of the scheme follows from the fact that $M^{ed} = M$ holds for any

$M \in \mathbb{Z}_N$, which can be seen by the Fermat's little theorem. The security is based on the RSA assumption, which claims that any probabilistic polynomial time (PPT) algorithm cannot compute M with non-negligible probability given N, e and $C = M^e$.

Though the RSA scheme was a significant breakthrough, there are still two problems regarding to the security. The first one is that since the RSA assumption is not reduced to the factoring assumption so far, the security of the scheme cannot be reduced to the factoring assumption. Hopefully, we want to obtain a PKE scheme whose security can be reduced to the factoring assumption. The second one is that though any PPT adversary cannot compute a message M from a ciphertext C, it may be possible that C reveals a partial information of M. Hopefully we want to ensure that C does not leak any information of M.

The first problem is solved by Rabin [Rab79] in 1979, who proposed the first PKE scheme whose security can be reduced to the factoring assumption. The Rabin scheme is as follows. A public key is $N = PQ$ as in the RSA scheme, and the secret key is P and Q. The encryption of a message M is done as $C = M^2 \mod N$ to generate a ciphertext C, and decryption is done by finding M that satisfies $C = M^2 \mod N$ *[1]. The scheme can be seen as a variant of the RSA scheme for $e = 2$, but what is important is that the security of the Rabin scheme is rigorously reduced to the factoring assumption.

The second problem is solved by Goldwasser and Mical [GM82] in 1982. They formalize the notion of semantic security for PKE, which means that any information of message is not leaked from a ciphertext and a public key. Then they construct a PKE scheme that satisfies the semantic security under the QR assumption.

Blum and Goldwasser [BG84] proposed the first PKE scheme that solves the these two problems simultaneously: The semantic security of the scheme can be reduced to the factoring assumption.

All of the above works only consider the security against chosen plaintext attacks (CPA) where an adversary only observes public keys and ciphertexts. On the other hand, there is a stronger security notion called the chosen ciphertext attack (CCA) security where an adversary in addition to observing public keys and ciphertexts accesses to a decryption oracle that returns decryption of an arbitrary ciphertext except the "target ciphertext" that adversary tries to break. Though the CCA security had been considered only of theoretical interest for a long time, in 1998, Bleichenbacher [Ble98] showed that a (partial) chosen ciphertext attack is possible in the real

*[1] In fact, there are 4 possible messages M that satisfies $M^2 = C$, therefore additional two bits should be included in a ciphertext to specify a message.

world by demonstrating an attack against PKCS #1 which was a widely used PKE scheme in the real. Therefore nowadays, the CCA security is considered as a desirable security of a PKE scheme. Naor and Yung [NY89] and Dolev, Dwork and Naor [DDN91] proposed a generic conversion from a PKE scheme with the CPA security to a one with the CCA security by using non-interactive zero-knowledge proof systems (NIZKs), but their schemes are too inefficient and far from practical. Fujisaki and Okamoto [FO99] proposed a practical conversion, but their security analysis relies on the random oracle model, in which a hash function is modeled as a completely random function. In 2009, Hofheinz and Kiltz [HK09b] proposed the first practical PKE scheme whose CCA security can be reduced to the factoring assumption in the standard model. Some variants of the Hofheinz-Kiltz scheme have been proposed [MLLJ11, LLML11, LLML12, LLL13, YYN^{+}14].

Broadcast encryption. Broadcast encryption (BE) [FN93] is a variant of PKE where a sender can arbitrary decide a set of receivers, and only designated receivers can decrypt the ciphertext to obtain a message. Since there is a trivial construction of a BE scheme whose ciphertext size is proportional to the number of receivers, BE is meaningful only if a ciphertext size is sublinear to the number of receivers. In 2000, Naor and Pinkas [NP00] proposed a revocation scheme, which can be seen as a BE scheme whose ciphertext size is proportional to the number of revoked users. That is, when encrypting a message to $n - t$ receivers out of n potential receivers, the ciphertext size of the scheme is $O(t)$ instead of $O(n)$. Wee [Wee11] constructed a factoring-based variant of the Naor-Pinkas scheme. Though these schemes are efficient when the receiver set is large, when that is small, namely $t \approx n$, the ciphertext size is almost the same as that of the trivial construction. Boneh, Gentry and Waters [BGW05] constructed a BE scheme whose ciphertext size is $O(\sqrt{n})$ regardless of the number of revoked users based on a pairing, and Boneh and Silverberg [BS02] (instantiated by [GGH13a]) proposed a BE scheme whose ciphertext is $O(1)$ based on a multilinear map. A factoring-based BE scheme with the similar property has been unknown so far.

Identity-based encryption. Identity-based encryption (IBE) [Sha84] is a variant of PKE where an arbitrary string can be used as a public key. Though the concept of IBE was proposed in 1984, the first constructions were proposed in 2000 and 2001 by Sakai, Ohgishi and Kasahara [SOK00] and Boneh and Franklin [BF01] independently. Their schemes are based on a pairing, and proven secure in the random oracle model. Boneh and Boyen [BB04] proposed an IBE scheme proven selectively secure in the standard model, and Waters [Wat05] proposed an IBE scheme proven adaptively secure in the standard model.

As a scheme related to the factoring problem, Cocks [Coc01] proposed an IBE scheme which is proven secure under the QR assumption in the random oracle model. Boneh, Gentry and Hamburg [BGH07] proposed an efficient variant of the above scheme, which is also proven secure under the QR assumption in the random oracle model. We note that there is no known IBE scheme which is proven secure under a factoring related assumption in the standard model, or proven secure under the factoring assumption (even in the random oracle model).

Attribute-based encryption. Attribute-based encryption (ABE) is an extension of PKE which enable us an arbitrary access control depending on attributes assigned to each receiver. The concept of ABE was proposed by Sahai and Waters [SW05], and there are many constructions of an ABE scheme based on bilinear pairings [BSW07, GPSW06]. Garg et al. [GGH$^+$13c] constructed the first ABE scheme for general circuits based on multilinear maps. Boneh et al. [BGG$^+$14] proposed a variant of the scheme with a compact ciphertext. Gorbunov et al. [GVW13] constructed an ABE scheme for general circuits based on the standard learning with errors (LWE) assumption. There is no known factoring-based construction of an ABE scheme.

Functional encryption. Functional encryption is an extension of ABE where a secret key sk_f is associated with a function f, and an encryption of a message m is decrypted to $f(x)$ by the secret key sk_f. The concept of FE is proposed by Boneh, Sahai and Waters [BSW11]. Garg, [GGH$^+$13b] constructed the first (selectively secure) FE scheme for all circuits based on an indistinguishability obfuscation. Waters [Wat15] and Ananth et al. [ABSV15] proposed fully secure FE schemes for all circuits. As an FE scheme for functions of more restricted class, Abdalla et al. [ABCP15] constructed selectively FE schemes for innner product based on a PKE scheme with a special structure, which can be constructed based on various kind of assumptions including DDH and LWE assumptions. Agrawal, Libert and Stehlé [ALS16] constructed fully secure FE schemes for inner products based on the DDH, LWE and DCR assumptions. There is no known factoring-based construction of an FE scheme for general circuits, or an FE scheme for inner products whose security.can be reduced to the factoring assumption.

Homomorphic encryption. Homomorphic encryption is PKE with the homomorphic property such that computation over encrypted messages can be done publicly. In fact, the first PKE, the RSA encryption, is already multiplicative homomorphic. That is, given a ciphertexts $C_1 = M_1^e$ and $C_2 = M_2^e$ for messages M_1 and M_2 respectively, we can compute a new ciphertext $C_1 \cdot C_2 = (M_1 \cdot M_2)^e$, which corresponds to the message $M_1 \cdot M_2$. The PKE scheme proposed by Goldwasser and Micali [GM82] is additively homomorphic over a message space $\mathbb{Z}_2$, and there has been proposed some additively homomorphic encryption scheme whose message space is larger based on factoring-related assumption [Cla94, NS98, Pai99, Gro05, JL13]. However, there is no known additively homomorphic encryption scheme which can be reduced to the factoring assumption.

In 2009, Gentry [Gen09] proposed the first fully homomorphic encryption (FHE) scheme based on an ideal lattice in which arbitrary computation can be done over encrypted messages. Thereafter, many FHE schemes have been proposed based on lattice-based [BV11, BGV12, Bra12, GSW13, CGGI16] or integer-based assumptions [vDGHV10, CMNT11, CNT12, CCK$^+$13, CLT14]. On the other hand, there is still no known FHE scheme based on a standard factoring-related assumption.

Deterministic public key encryption. Deterministic public key encryption (DPKE) is PKE whose encryption algorithm is deterministic. Though a DPKE scheme cannot satisfy the semantic security, Bellare et al. [BBO07] defined the PRIV security as the best possible security of a DPKE scheme. They constructed a DPKE scheme that satisfies the PRIV security in the random oracle model. Boldyreva et al. [BFO08] weakened the PRIV security to define the security notion which they call the PRIV security for block-sources, and constructed DPKE schemes with this security in the standard model based on lossy trapdoor functions (LTDFs). Bellare et al. [BFOR08] constructed DPKE scheme with a further weaker security notion (where messages are uniformly random) can be constructed from any one-way trapdoor permutation. There is no known construction of a DPKE scheme that can be proven to satisfy the PRIV security for block sources under the factoring assumption.

Non-interactive key exchange. Non-interactive key exchange (NIKE) is a cryptographic protocol where many parties share a common key without any interaction except publishing each user's public key. Though Diffie and Hellman [DH76] proposed the first 2-party NIKE scheme in 1976, it was not until 2013 that a formal definition and security models were given by [FHKP13]. In [FHKP13], they proposed a NIKE schemes that satisfies the strongest security definition they define under a pairing-based assumption in the standard model, or the factoring assumption in the random oracle model. There is no known construction of 2-party NIKE scheme that satisfies their security requirement under the factoring (or a factoring-related) assumption in the standard model.

For the case of more than 2 parties, Joux [Jou00] proposed a 3-party NIKE scheme based on a pairing, and Boneh and Silverberg [BS02] (instantiated by [GGH13a]) proposed a multi-party NIKE scheme based on a multilinear map. There is no known NIKE scheme for more than 2 parties based on a factoring-related assumption.

Digital signatures. A digital signature is a digital analogue of a handwrite signature. The concept of digital signatures was proposed by Diffie and Hellman [DH76], and the first digital signature scheme was proposed by Rivest, Shamir and Adleman [RSA78] at the same time as the first PKE scheme was proposed. Actually, their signature scheme uses the same mechanism as the RSA encryption: The verification key consists of a multiple $N = PQ$ of two primes P and Q and a public exponent e, and a signing key is a secret exponent d such that $ed \equiv 1 \mod (P-1)(Q-1)$. A signature σ for a message M is generated as $\sigma := M^d \mod N$, and a signature is verified by checking whether $\sigma^e = M \mod N$ holds. Intuitively, the security of the scheme follows from the assumption that it is hard to find a e-th root modulo N. However, there is a vulnerability in their scheme that even without a signing key, one can first generate a signature σ and then generate a message $M := \sigma^e \mod N$ so that (M, σ) is accepted by the verification algorithm.

Goldwasser, Micali and Rivest [GMR88] gave a strong security definition called existential unforgeability against chosen message attack (EUF-CMA). Rompel [Rom90] constructed EUF-CMA secure digital signatures solely based on a one-way function. However, their construction is very complicated and not practical. As a more direct and efficient construction of digital signatures based on factoring-related assumptions, There have been proposed some EUF-CMA secure digital signatures based on the strong RSA assumption, [GHR99, CS00, Fis03, HK08]. Hohenberger and Waters [HW09] constructed EUF-CMA secure digital signatures based on the RSA assumption for the first time. Some variants of the scheme have been proposed [HJK11, YHK12, BHJ$^+$13].

Homomorphic signatures Homomorphic signatures are digital signatures with a homomorphic property that anyone can publicly evaluate a function on signatures to generate a new signature for the function value on original messages. Gennaro et al. [GKKR10] constructed a linearly homomorphic signature scheme based on the RSA assumption. Boneh and Freeman [BF11] were the first to propose homomorphic signatures that can handle a wider class of functions than linear functions. Their scheme can handle arbitrary polynomial and security is proven in the random oracle based on the hardness of the short integer solution (SIS) problem. Catalano et al. [CFW14] proposed such a scheme in the standard model based on a multilinear map. Gorbunov et al. [GVW15] constructed a (leveled) fully homomorphic signature, which can handle any polynomial size function based on the learning with errors (LWE) assumption. Xie et al. [XX14] proposed (bounded) fully homomorphic signatures based on $i\mathcal{O}$. There is no known factoring-based homomorphic signatures that can handle a wider class of functions than linear functions.

Pseudorandom generator. Pseudorandom generator (PRG) is a cryptographic primitives that is given a random seed to generate a longer pseudorandom string that cannot be distinguished from a uniformly random string. There is a generic construction of PRG based on a one-way function [HILL99]. As a direct construction from the factoring assumption, Blum, Blum and Shub [BBS86] constructed an efficient PRG.

Pseudorandom function. Pseudorandom function (PRF) is a efficiently computable keyed function which is indistinguishable from a truly random function by a black-box access. There is a generic construction of PRF based on a PRG [GGM84]. As a direct construction from the factoring assumption, Naor and Reingold [NR04] constructed an efficient PRF, and the scheme is further improved by Naor, Reingold and Rosen [NRR02].

1.3 Limitations of Factoring-based Cryptography.

As summarized in the previous section, there have been wide variety of progresses in factoring-based cryptography until now. However, there are still many open problems in this area. Especially, we focus on the following two problems in this dissertation.

1. There is no known construction of cryptographic primitives with high functionality including ABE, FE, FHE etc. though they are constructed based on other assumptions related to pairings or lattices. It is an important problem to consider whether it is possible to construct these primitives based on the factoring or factoring-related assumptions.
2. There are some cryptographic primitives that can be constructed based on a factoring-related assumption, but cannot be reduced to the factoring assumption including DPKE, additively homomorphic encryption etc. It is an important problem to consider whether these primitives can be constructed based on the factoring assumption.

In this dissertation, we make progresses toward solving these problems.

1.4 Summary of Contributions

Our contributions can be divided into the following two parts.

1. In the first part, we study a cryptographic primitive called self-bilinear map. A self-bilinear map is a special type of a bilinear map where target and domain groups are identical. Cheon and Lee [CL09] showed that a self-bilinear map implies a multilinear map, which is known to imply various kinds of cryptographic primitives including multiparty non-interactive key exchange (NIKE) [BS02], broadcast encryption [BS02], attribute-based encryption [GGH$^+$13c], homomorphic signatures [CFW14] etc. On the other hand, they also showed that an impossibility result on the existence of a self-bilinear map on known prime order group. Namely, they showed that if there exists an efficiently computable self-bilinear map on a group of known prime order, then the computational Diffie-Hellman (CDH) assumption cannot hold on the group. In this dissertation we consider unknown order group instead of prime order group to avoid the above impossibility result. We define a weaker variant of self-bilinear map, which we call self-bilinear map with auxiliary information (AI-SBM). We construct an AI-SBM based on the factoring assumption and the existence of indistinguishability obfuscation (iO), which is another cryptographic primitive that makes a circuit totally unintelligible while keeping its functionality. We show that we can replace a multilinear map with AI-SBM in many applications including multiparty NIKE, BE, ABE and homomorphic signatures. Moreover, our construction of multiparty NIKE and broadcast encryption is the first construction that admits unbounded number of users. As a side result, we construct a somewhat homomorphic encryption for log-depth arithmetic circuits based on the Φ-hiding assumption and the existence of iO.
   Since we assume the existence of iO in addition to the factoring assumption, the construction is not fully factoring-based one. However, our result shows a connection between those advanced cryptographic primitives and factoring-based cryptography, and we believe that it may be useful for factoring-based construction of those primitives in the future.
   Contents of this part are based on [YYHK14, YHK16, YYHK].
2. In the second part, we study a cryptographic primitive called a lossy trapdoor function (LTDF) introduced by Peikert and Waters [PW08]. It is known that LTDFs imply various kinds of cryptographic primitives including collision

resistant hash function, oblivious transfer, CPA/CCA secure PKE [PW08], DPKE [BFO08], selective opening attack secure PKE [BHY09], universally composable commitment [NFT09], etc. It is also known that LTDFs can be constructed based on various kinds of assumptions including decisional Diffie-Hellman (DDH), learning with errors (LWE), QR, DCR assumptions etc. Though there are some "factoring-based construction"of LTDFs, there is no known construction of LTDFs whose security is rigorously reduced to the factoring assumption. In this part, we first relax the definition of an LTDF to define an adversary-dependent lossy trapdoor functions (ad-LTDF). Then we show that in many applications of LTDFs, we can replace LTDFs with ad-LTDFs. Moreover, we give a construction of an ad-LTDF based on the factoring assumption w.r.t. semi-smooth RSA subgroup moduli (SS moduli), which is a special type of RSA moduli introduced by Groth [Gro05]. As a result, we almost automatically obtain a new constructions of collision resistant hash function, CPA/CCA secure PKE schemes and DPKE scheme based on the same assumption. Especially, our DPKE scheme is the first scheme that satisfies the security notion called the PRIV security for block sources proposed by Boldyreva et al. [BFO08] solely based on the factoring assumption.

Besides direct applications of ad-LTDFs, we construct a CCA secure PKE scheme based on the factoring assumption w.r.t. SS moduli whose ciphertext overhead is the shortest among schemes based on the same assumption.

Contents of this part are based on [YYHK16].

1.5 Organization

In Chapter 2, we review notations and definitions of cryptographic primitives that are used throughout this dissertation. In Chapter 3, we study constructions and applications of AI-SBMs. In Chapter 4, we study constructions and applications of ad-LTDFs. In Chapter 5, we give concluding remarks and open problems.

Chapter 2

Preliminaries

2.1 Notations

We use $\mathbb{N}$ to denote the set of all natural numbers, and $[n]$ to denote the set $\{1, \ldots n\}$ for $n \in \mathbb{N}$. If S is a finite set, then we use $x \stackrel{\$}{\leftarrow} S$ to denote that x is chosen uniformly at random from S. If $\mathcal{A}$ is an algorithm, we use $x \leftarrow \mathcal{A}(y; r)$ to denote that x is output by $\mathcal{A}$ whose input is y and randomness is r. We often omit r. For a finite set S, $|S|$ denotes the cardinality of S. For a real number x, $\lceil x \rceil$ denotes the smallest integer not smaller than x and $\lfloor x \rfloor$ denotes the largest integer not larger than x. For a bit string a, ℓ_a denotes the length of a. For a function f in λ, we often denote f to mean $f(\lambda)$ for notational simplicity. We say that a function $f(\cdot) : \mathbb{N} \to [0, 1]$ is negligible if for all positive polynomials $p(\cdot)$ and all sufficiently large $\lambda \in \mathbb{N}$, we have $f(\lambda) < 1/p(\lambda)$. We say f is overwhelming if $1 - f$ is negligible. We say that a function $f(\cdot) : \mathbb{N} \to [0, 1]$ is noticeable if there exists a polynomial p such that for all sufficiently large λ, we have $f(\lambda) > |1/p(\lambda)|$. We say that an algorithm $\mathcal{A}$ is probabilistic polynomial time (PPT) if there exists a polynomial p such that the running time of $\mathcal{A}$ with input length λ is less than $p(\lambda)$. We use $a|b$ to mean that a is divisor of b. For a set S and a random variable x over S, we say that x is almost random on S if the statistical distance between the distribution of x and the uniform distribution on S is negligible. For a natural number N, $\Phi(N)$ denote the number of natural numbers smaller than N that are coprime to N. For random variables X and Y, $\Delta(X, Y)$ denote the statistical distance between them. We use the fact that for any (probabilistic) function f, $\Delta(f(X), f(Y)) \leq \Delta(X, Y)$ holds, and that $\Delta((X_1, Z), (Y_1, Z)) = \mathbb{E}_Z[\Delta(X_1, Y_1)]$ where $\mathbb{E}$ denotes the expected value. For random variables X and Y, we define min-entropy of X as $H_\infty(X) := -\log(\max_x \Pr[X = x])$ and average min-entropy of X given Y as $\tilde{H}_\infty(X|Y) := -\log(\Sigma_y \Pr[Y = y] \max_x \Pr[X = x|Y = y])$. When we treat a circuit, we use the similar notation as in [BHR12, GGH$^+$13c]. For a circuit f with input length n which has v wires, We identify the set of wires with $[v]$ and input wires with $[n]$, and we label the output wire by v. For a wire w which is an output wire of a gate, we denote the first input incoming wire of the gate by $A(w)$ and the second incoming wire of the gate by $B(w)$. We use λ to denote the security parameter

2.2 Syntax and Security Notions

Here, we review definitions of cryptographic primitives.

Pairwise independent hash function. We say that a family $\mathcal{H}$ of hash functions from $\{0,1\}^n$ to $\{0,1\}^m$ is pairwise independent if for any $x_1 \neq x_2 \in \{0,1\}^n$ and $y_1, y_2 \in \{0,1\}^m$, $\Pr[H(x_1) = y_1 \wedge H(x_2) = y_2 : H \overset{\$}{\leftarrow} \mathcal{H}] = 2^{-2m}$ holds.

Collision resistant hash function. We formalize a collision resistant hash function as a pair of PPT algorithms $\Pi = (\mathsf{Gen}, \mathsf{Eval})$.

$\mathsf{Gen}(1^\lambda)$ It takes the security parameter 1^λ as input and outputs a function description h.

$\mathsf{Eval}(h, x)$ It is a deterministic algorithm that takes a function description h and x as input and outputs $h(x)$.

We require that for any PPT adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{CR}}_{\mathcal{A},\Pi}(\lambda) := \Pr[h(x) = h(x'), x \neq x' : h \leftarrow \mathsf{Gen}(1^\lambda); (x, x') \leftarrow \mathcal{A}(h)]$ is negligible.

Public key encryption. A PKE scheme consists of three algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

$\mathsf{Gen}(1^\lambda)$: It takes the security parameter 1^λ as input and outputs (PK, SK), where PK is a public key and SK is a secret key.

$\mathsf{Enc}(PK.msg)$: It takes a public key PK and a message msg as input and outputs a ciphertext C.

$\mathsf{Dec}(\mathsf{SK}, \mathsf{C})$: It takes a secret key SK and a ciphertext C as input and outputs a massage msg.

We require that for all (PK, SK) output by Gen, all msg and all C output by $\mathsf{Enc}(PK, msg)$, we have $\mathsf{Dec}(SK, C) = msg$.

For ATK $\in \{\mathrm{CPA}, \mathrm{CCA}\}$, a public key encryption scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is ATK secure if for all PPT adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, $\mathsf{Adv}^{\mathsf{ATK}}_{\mathcal{A},\mathsf{PKE}}(\lambda) := |\Pr[b =$

$b' : (PK, SK) \leftarrow \mathsf{Gen}(1^\lambda); (msg_0, msg_1, st) \leftarrow \mathcal{A}_1^{\mathcal{O}_1}(PK); b \xleftarrow{\$} \{0,1\}; C^* \leftarrow \mathsf{Enc}(PK, msg_b); b \leftarrow \mathcal{A}_2^{\mathcal{O}_2}(PK, C^*, st)] - 1/2|$ is negligible where if ATK=CPA, then $\mathcal{O}_i$ $(i = 1, 2)$ is an oracle that always returns $\perp$, and if ATK=CCA, then $\mathcal{O}_i$ $(i = 1, 2)$ is a decryption oracle that is given a ciphertext C and returns $\mathsf{Dec}(SK, C)$ if $i = 1$ or $C \neq C^*$ and otherwise $\perp$.

Key encapsulation mechanism. Here, we review the definition of key encapsulation mechanism (KEM) and its security. It is shown that a CCA secure PKE scheme is obtained by combining a constrained CCA (CCCA) secure KEM and a CCA secure authenticated symmetric key encryption scheme [HK07]. In the following, we recall the definitions of KEM and its CCCA security.

A KEM consists of three algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

$\mathsf{Gen}(1^\lambda)$: It takes a security parameter 1^λ as input and outputs (PK, SK), where PK is a public key and SK is a secret key.

$\mathsf{Enc}(PK)$: It takes a public key PK as input and outputs (C, K), where C is a ciphertext and K is a symmetric key.

$\mathsf{Dec(SK, C)}$: takes a secret key SK and a ciphertext C as input and outputs a key K with length ℓ_K or $\perp$.

We require that for all (PK, SK) output by Gen and all (C, K) output by $\mathsf{Enc}(PK)$, we have $\mathsf{Dec}(SK, C) = K$.

To define the CCCA security of $\mathsf{KEM} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, we consider the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. First, $\mathcal{C}$ generates $(PK, SK) \leftarrow \mathsf{Gen}(1^\lambda)$ and $(C^*, K) \leftarrow \mathsf{Enc}(PK)$, chooses a random bit $b \xleftarrow{\$} \{0,1\}$, and sets $K^* := K$ if $b = 1$ and otherwise $K^* \xleftarrow{\$} \{0,1\}^{\ell_K}$. Then (PK, C^*, K^*) is given to the adversary $\mathcal{A}$. In the game, $\mathcal{A}$ can query pairs of ciphertexts and predicates any number of times. When $\mathcal{A}$ queries (C, pred), $\mathcal{C}$ computes $K \leftarrow \mathsf{Dec}(SK, C)$ and returns K to $\mathcal{A}$ if $C \neq C^*$ and $\mathsf{pred}(K) = 1$, and otherwise $\perp$. Finally, $\mathcal{A}$ outputs a bit b'. We define the CCCA advantage of $\mathcal{A}$ as $\mathsf{Adv}^{\mathsf{CCCA}}_{\mathcal{A},\mathsf{KEM}}(\lambda) := |\Pr[b = b'] - 1/2|$. We say that KEM is CCCA secure if $\mathsf{Adv}^{\mathsf{CCCA}}_{\mathcal{A},\mathsf{KEM}}(\lambda)$ is negligible for any PPT *valid* adversary $\mathcal{A}$, where “valid” is defined below.

Before defining “valid” , we prepare two definitions. We say that a predicate pred is *non-trivial* if $\Pr[\mathsf{pred}(K) = 1 : K \xleftarrow{\$} \{0,1\}^{\ell_K}]$ is negligible. We say that an algorithm $\mathcal{C}'$ is an *alternative challenger* if it has the same syntax as the real challenger $\mathcal{C}$. We say that an adversary $\mathcal{A}$ is valid if for any PPT alternative challenger $\mathcal{C}'$, all predicates pred queried by $\mathcal{A}$ in the game between $\mathcal{A}$ and $\mathcal{C}'$ are non-trivial.

Though the above definition of the CCCA security slightly differs from the original definition given in [HK07], we can easily prove that our definition still yields the "hybrid encryption theorem" that a CCA secure PKE scheme can be obtained by a CCCA secure KEM and authenticated symmetric key encryption.

Deterministic public key encryption. Here, we define deterministic public key encryption (DPKE). A DPKE scheme consists of three algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

$\mathsf{Gen}(1^\lambda)$ It takes a security parameter 1^λ as input and outputs (PK, SK), where PK is a public key and SK is a secret key.

$\mathsf{Enc}(PK, msg)$: It is a deterministic algorithm that takes a public key PK and a message msg as input and outputs a ciphertext C.

$\mathsf{Dec}(\mathsf{SK}, \mathsf{C})$: It takes a secret key SK and a ciphertext C as input and outputs a message msg or $\perp$.

For correctness, we require that for all msg, (PK, SK) output by Gen and C output by $\mathsf{Enc}(PK, msg)$, we have $\mathsf{Dec}(SK, C) = msg$.

We recall security notions for deterministic encryption following [BFO08]. In [BFO08], the authors considered three security notions called PRIV, PRIV1 and PRIV1-IND, and proved all of them are equivalent. Therefore we consider only the simplest security definition PRIV1-IND in this dissertation. A random variable X over $\{0,1\}^n$ is called a (u,n)-source if $H_\infty(X) \geq u$. For ATK $\in$ {CPA, CCA}, a deterministic encryption scheme $\mathsf{DE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ for ℓ-bit message is PRIV1-IND-ATK secure for (t,n)-sources if for any (t,n)-sources M_0 and M_1 and all PPT adversaries $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{PRIV1-IND-ATK}}_{\mathcal{A},M_0,M_1,\mathsf{DE}}(\lambda) := |\Pr[b = b' : (PK, SK) \leftarrow \mathsf{Gen}(1^\lambda); b \overset{\$}{\leftarrow} \{0,1\}; msg^* \overset{\$}{\leftarrow} M_b; C^* \leftarrow \mathsf{Enc}(PK, msg^*); b' \leftarrow \mathcal{A}^{\mathcal{O}}(PK, C^*)] - 1/2|$ is negligible where if ATK=CPA, then $\mathcal{O}$ is an oracle that always returns $\perp$, and if ATK=CCA, then $\mathcal{O}$ is an decryption oracle that is given a ciphertext C and returns $\mathsf{Dec}(SK, C)$ if $C \neq C^*$ and otherwise $\perp$.

Distributed broadcast encryption. Here, we define distributed broadcast encryption following [BZ14]. A distributed broadcast encryption scheme consists of four algorithms $(\mathsf{Setup}, \mathsf{Join}, \mathsf{Enc}, \mathsf{Dec})$.

$\mathsf{Setup}(1^\lambda)$: It takes the security parameter 1^λ as input and outputs public parameters PP.

$\mathsf{Join}(PP)$: It takes public parameters PP as input and outputs a public key pk and a secret key sk.

$\mathsf{Enc}(PP, pk_1, \ldots, pk_n, msg)$: It takes public parameters PP, a message msg, and a public keys $pk_1, \ldots, pk_n$ of designated receivers and outputs a ciphertext CT.

$\mathsf{Dec}(PP, sk, pk_1, \ldots, pk_n, CT)$: It takes public parameters PP, a secret key sk, public keys of designated receivers and a ciphertext CT and outputs a message msg.

As correctness, we require that for any security parameter λ, an integer n and a message msg, we have $\mathsf{Dec}(PP, sk, pk_1, \ldots, pk_n, CT) = msg$, where $PP \leftarrow \mathsf{Setup}(1^\lambda)$, $(pk_1, sk_1), \ldots, (pk_n, sk_n) \leftarrow \mathsf{Join}(PP)$, $CT \leftarrow \mathsf{Enc}(PP, pk_1, \ldots, pk_n, msg)$.

We define the security notion. We consider the following experiment. A challenger runs $\mathsf{Setup}(1^\lambda)$ to generate public parameters PP and runs $\mathsf{Join}(PP)$ n times to generate $(pk_1, sk_1), \ldots, (pk_n, sk_n)$. It gives $(PP, pk_1, \ldots, pk_n)$ to $\mathcal{A}$. $\mathcal{A}$ chooses two messages msg_0 and msg_1 to submit them to $\mathcal{C}$. $\mathcal{C}$ uniformly choose $b \overset{\$}{\leftarrow} \{0, 1\}$ and runs $\mathsf{Enc}(PP, pk_1, \ldots, pk_n, msg_b)$ to generate CT, and gives CT to $\mathcal{A}$. Finally $\mathcal{A}$ outputs b'. We say that $\mathcal{A}$ wins if $b = b'$ holds.

We say that a distributed broadcast encryption scheme is statically secure if for any n (polynomially bounded in λ) and any PPT adversary $\mathcal{A}$, $|\Pr[\mathcal{A} \text{ wins}] - 1/2|$ is negligible.

Remark 1. *At first glance, the above security notion seems weaker than the usual static security of broadcast encryption because we do not allow an adversary to corrupt receivers who are out of the target set. However, in distributed setting, secret and public keys of such receivers can be simulated efficiently by using the public parameters. Therefore we still capture the setting where the adversary may corrupt some receivers (as long as the set of corrupted users is determined at the beginning of the experiment).*

Remark 2. *There is stronger security notion for broadcast encryption called adaptive security, where an adversary can determine which receiver to corrupt adaptively. In this dissertation we only consider the static security and does not consider the adaptive security.*

Attribute-based encryption for circuits. Here, we define attribute-based encryption (ABE) and its security. An ABE scheme consists of four algorithms $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$.

$\mathsf{Setup}(1^\lambda)$: It takes the security parameter 1^λ, the length n of the index as input and upper bound d of circuit depth, and outputs the public parameters PP and a master secret key MSK.

$\mathsf{Enc}(PP, x, M)$: It takes the public parameters PP, an index $x \in \{0, 1\}^n$ and a message M as input, and outputs a ciphertext CT.

$\mathsf{KeyGen}(MSK, f)$: It takes a master secret key MSK and a circuit f with a single output gate, and outputs a secret key SK.

$\mathsf{Dec}(SK, CT)$: It takes a secret key SK and a ciphertext CT as input, and outputs a message M or $\perp$.

For correctness, we require that for all M, $x \in \{0,1\}^n$ and f with depth lower than d such that $f(x) = 1$, $\mathsf{Dec}(SK, CT) = M$ always holds, where $(PP, MSK) \leftarrow \mathsf{Setup}(1^\lambda, n, d)$, $SK = \mathsf{KeyGen}(MSK, f)$ and $CT = \mathsf{Enc}(PP, x, M)$.

Next, we define the security of ABE. Here, we only define the selective security since we only consider it in this dissertation. For an adversary $\mathcal{A}$, we consider the following game between $\mathcal{A}$ and a challenger. $\mathcal{A}$ first declares the target index x^*. Then the challenger computes $(PP, MSK) \leftarrow \mathsf{Setup}(1^\lambda)$ and gives PP to $\mathcal{A}$. Then $\mathcal{A}$ declares M_0 and M_1. The challenger chooses $b \xleftarrow{\$} \{0,1\}$ and computes $CT \leftarrow \mathsf{Enc}(PP, x^*, M_b)$. Then it gives CT to $\mathcal{A}$. In the game, $\mathcal{A}$ can query a circuit f such that $f(x^*) = 0$ for key generation oracle, and the oracle returns $\mathsf{KeyGen}(MSK, f)$ to $\mathcal{A}$. Finally, $\mathcal{A}$ outputs b'. We say that $\mathcal{A}$ wins if $b' = b$. We say that an ABE scheme is selectively secure if for any efficient adversary $\mathcal{A}$, tha probability that $\mathcal{A}$ wins is negligibly close to $1/2$.

It is known that any general Boolean circuit can be converted to an equivalent monotone layered Boolean circuit [GGH$^+$13c]. Therefore we only consider ABE for monotone layered circuits. Here, monotone circuit is a circuit where all gates are either AND or OR gates of two inputs, and layered circuit is a circuit where a gate at depth j receive both of its inputs from wires at depth $j - 1$.

Homomorphic encryption Here, we recall some definitions for homomorphic encryption. A homomorphic encryption scheme HE consists of the four algorithms $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$.

$\mathsf{KeyGen}(1^\lambda)$: It takes the security parameter 1^λ as input and outputs a public key pk and a secret key sk.

$\mathsf{Enc}(pk, m)$ It takes a public key pk and a massage $m \in \{0,1\}$ as input, and outputs a ciphertext c.

$\mathsf{Eval}(pk, f, \ell, c_1, \ldots, c_\ell)$ It takes a public key pk, a circuit f with input length ℓ and a set of ℓ ciphertexts $c_1, \ldots, c_\ell$ as input, and outputs a ciphertext c_f.

$\mathsf{Dec}(sk, c)$: IT takes a secret key sk and a ciphertext c as input, and outputs a message m.

For correctness of the scheme, we require that for all security parameters λ, all

$(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and all $m \in \{0,1\}$, we have $\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m$ with overwhelming probability.

Next, we define some properties of homomorphic encryption such as the CPA security, $\mathcal{C}$-homomorphism, and compactness.

Definition 1. *(CPA security) We say that a scheme* HE *is CPA secure if for any efficient adversary* $\mathcal{A}$*,*

$$|\Pr[1 \leftarrow \mathcal{A}(pk, \mathsf{Enc}(pk, 0))] - \Pr[1 \leftarrow \mathcal{A}(pk, \mathsf{Enc}(pk, 1))]|$$

is negligible, where $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$*.*

Definition 2. *($\mathcal{C}$-homomorphism) Let* $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda \in \mathbb{N}}$ *be a class of circuits. A scheme* HE *is* $\mathcal{C}$*-homomorphic if for any family of circuits* $\{f_\lambda\}_{\lambda \in \mathbb{N}}$ *such that* $f_\lambda \in \mathcal{C}$ *whose input length is* ℓ *and any messages* $m_1, \ldots, m_\ell \in \{0,1\}$*,*

$$\Pr[\mathsf{Dec}(sk, \mathsf{Eval}(pk, C, c_1, \ldots, c_\ell)) \neq C(m_1, \ldots, m_\ell)]$$

is negligible, where $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$ *and* $c_i \leftarrow \mathsf{Enc}(pk, m_i)$*.*

Remark 3. *We can also consider the additional property that an output of* Eval *can be used as input of another homomorphic evaluation. This is called "multi-hop" homomorphism, and many fully homomorphic encryption schemes have this property. However, our scheme does not.*

Definition 3. *(Compactness) A homomorphic encryption scheme* HE *is compact if there exists a polynomial* poly *such that the output length of* Eval *is at most* $\mathsf{poly}(\lambda)$*-bit.*

Multiparty non-interactive key exchange. First, we formally define multiparty non-interactive key exchange (NIKE) and its security following [BZ14]. A multiparty NIKE scheme consists of three algorithms (Setup,$\mathsf{Publish}$,KeyGen).

$\mathsf{Setup}(1^\lambda)$: This algorithm takes a security parameter 1^λ as input [*1]. It outputs public parameters params.

$\mathsf{Publish}(\mathsf{params})$: This algorithm takes public parameters params as input. It outputs a public key pk and a secret key sk.

$\mathsf{KeyGen}(\mathsf{params}, sk, pk_1, \ldots, pk_{n-1})$: This algorithm takes public parameter params, a secret key sk and public keys $pk_1, \ldots, pk_{n-1}$. It outputs a shared key k.

[*1] In our definition, the setup algorithm does not take the number of maximum users as input. This means that our scheme admits unbounded number of users.

As correctness, we require that for any n, $\mathsf{params} \leftarrow \mathsf{Setup}(1^\lambda)$, $(pk_i, sk_i) \leftarrow \mathsf{Publish}(\mathsf{params})$ for $i \in [n]$, for any $i_1, i_2 \in [n]$, we have

$$\begin{aligned}&\mathsf{KeyGen}(\mathsf{params}, sk_{i_1}, pk_1, \dots, pk_{i_1-1}, pk_{i_1+1}, \dots pk_n)\\ &= \mathsf{KeyGen}(\mathsf{params}, sk_{i_2}, pk_1, \dots, pk_{i_2-1}, pk_{i_2+1}, \dots pk_n).\end{aligned}$$

We define the security notion for NIKE. In this dissertation, we consider the minimum security notion against a passive adversary.

We say that a multiparty NIKE scheme is statically secure if for any integer n which is polynomial in the security parameter, for any efficient adversary $\mathcal{A}$, $|\Pr[b \xleftarrow{\$} \mathcal{A}(\mathsf{params}, pk_1, \dots, pk_n, K_b)] - 1/2|$ is negligible, where $\mathsf{params} \leftarrow \mathsf{Setup}(1^\lambda)$, $(pk_i, sk_i) \leftarrow \mathsf{Publish}(\mathsf{params})$ for $i = 1, \dots, n$, $K_1 := \mathsf{KeyGen}(\mathsf{params}, sk_1, pk_2, \dots, pk_n)$, $K_0 \xleftarrow{\$} \{0,1\}^{\ell_K}$ and $b \xleftarrow{\$} \{0,1\}$.

Remark 4. *In some existing works [BZ14, KRS15], they consider stronger security notion that considers adversaries actively generate malformed public keys. In this dissertation, we do not consider such an adversary, and we assume that an adversary only observe honestly generated public keys.*

Homomorphic signature. Here, we give the definition of a homomorphic signature. In this dissertation, we only consider a selectively secure single data homomorphic signature for simplicity. A single data homomorphic signature for a function class $\mathcal{C}$ consists of PPT algorithms $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Eval})$.

$\mathsf{KeyGen}(1^\lambda, 1^n) \to (vk, sk)$: This algorithm takes the security parameter 1^λ and a data size 1^n, and outputs a pair $(vk.sk)$ of a verification key and a signing key.

$\mathsf{Sign}(sk, i, m) \to \sigma$: This algorithm takes a signing key sk, an index $i \in [n]$ and a message $m \in \mathcal{M}$, and outpus a signature σ.

$\mathsf{Eval}(f, (m_1, \sigma_1), \dots, (m_n, \sigma_n)) \to \sigma^*$: This algorithm takes a function $f \in \mathcal{C}$ and pairs $(m_1, \sigma_1), \dots, (m_n, \sigma_n)$ of a message and a signature, and outputs a signature σ^* for the message $f(m_1, \dots, m_n)$.

$\mathsf{Verify}(vk, f, m, \sigma) \to 1/0$: This algorithm takes a verification key vk, a function f, a message m and a signature σ, and outputs 1 if accepts and 0 else

Remark 5. *In ordinary homomorphic signature schemes, signatures generated by* Eval *can be input to* Eval *again to evaluate another function. On the other hand, in our formulation, only signatures generated by* Sign *can be evaluated by* Eval.

We require a homomorphic signature to satisfy the following properties.

Correctness For any $(vk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda, 1^n)$, $(m_1, \ldots, m_n) \in \mathcal{M}^N$, $f \in \mathcal{C}$, if we let $\sigma_i \leftarrow \mathsf{Sign}(sk, i, m_i)$, $m^* := f(m_1, \ldots, m_n)$, and $\sigma^* := \mathsf{Eval}(f, (m_1, \sigma_1), \ldots, (m_n, \sigma_n))$, then we always have $\mathsf{Verify}(vk, f, m^*, \sigma^*) = 1$.

Remark 6. *In the above, we only require the correctness for signatures generated by* Eval*, but if* $\mathcal{C}$ *includes the identity function, then this implies the correctness for signatures generated by* Sign*.*

Indistinguishability obfuscator. Here, we recall the definition of an indistinguishability obfuscator [GGH+13b, SW14].

Let C_λ be the class of circuits of size at most λ. An efficient randomized algorithm $i\mathcal{O}$ is called an indistinguishability obfuscator for *P/poly* if the following conditions are satisfied:

- For all security parameters $\lambda \in \mathbb{N}$, for all $C \in \mathcal{C}_\lambda$, we have that

$$\Pr[\forall x \; C'(x) = C(x) : C' \leftarrow i\mathcal{O}(\lambda, C)] = 1.$$

- For any (not necessarily uniform) efficient algorithm $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, there exists a negligible function α such that the following holds: if $\mathcal{A}_1(1^\lambda)$ always outputs (C_0, C_1, σ) such that we have $C_0, C_1 \in \mathcal{C}_\lambda$ and $\forall x \; C_0(x) = C_1(x)$, then we have

$$\begin{aligned} &|\Pr[\mathcal{A}_2(\sigma, i\mathcal{O}(\lambda, C_0)) = 1 : (C_0, C_1, \sigma) \leftarrow \mathcal{A}_1(1^\lambda)] \\ &- \Pr[\mathcal{A}_2(\sigma, i\mathcal{O}(\lambda, C_1)) = 1 : (C_0, C_1, \sigma) \leftarrow \mathcal{A}_1(1^\lambda)]| \leq \alpha(\lambda) \end{aligned}$$

2.2.1 RSA Modulus and Factoring Assumption

An integer $N = PQ$ is called a RSA modulus if P and Q are distinct primes with the same length. We define the group of quadratic residues as $\mathbb{QR}_N := \{u^2 : u \in \mathbb{Z}_N{}^*\}$ and the group of signed quadratic resides as $\mathbb{QR}_N^+ := \{|x| : x \in \mathbb{QR}_N\}$. where $|x|$ is the absolute value of x when it is represented as an element of $\{-(N-1)/2, \ldots, (N-1)/2\}$. This is a group with multiplication defined as $x \circ y := |(xy \bmod N)|$ for $x, y \in \mathbb{QR}_N^+$.

Let RSAGen be an efficient algorithm that is given the security parameter 1^λ and outputs an ℓ_N-bit RSA modulus $N = PQ$ with its factorization (P, Q). We say that the factoring assumption holds w.r.t. RSAGen if for any PPT algorithm $\mathcal{A}$, $\Pr[\mathcal{A}(N) \in \{P, Q\} : (N, P, Q) \leftarrow \mathsf{RSAGen}(1^\lambda)]$ is negligible.

Since we consider different type of RSA moduli in each chapter, we refer more detailed structure of N to each chapter.

Chapter 3

Self-bilinear map

3.1 Introduction

In this chapter, we study a cryptographic primitive called a self-bilinear map. Though a self-bilinear map is a very useful primitive, there is a negative result on the existence of a self-bilinear map on known prime order group. We consider unknown order group instead of prime order group to avoid the above impossibility result. We define a weaker variant of self-bilinear map, which we call self-bilinear map with auxiliary information (AI-SBM). We construct an AI-SBM based on the factoring assumption and the existence of indistinguishability obfuscation (iO). We show that we can replace a multilinear map with AI-SBM in many applications including multiparty NIKE, BE, ABE and homomorphic signatures. Moreover, our construction of multiparty NIKE and broadcast encryption is the first construction that admits unbounded number of users. As a side result, we construct a somewhat homomorphic encryption for log-depth arithmetic circuits based on the Φ-hiding assumption and the existence of iO.

3.1.1 Background

A bilinear map is an important tool in constructions of various cryptographic primitives, such as identity-based encryption (IBE)[BF01, BB04, Wat05], attribute-based encryption (ABE) [SW05, BSW07, GPSW06], non-interactive zero-knowledge (NIZK) proof systems [GOS06, GS08] etc. Bilinear maps which are mainly used in cryptography, are constructed on elliptic curve groups. In these constructions, the target group is different from the domain groups.

This leads to the natural question: is it possible to construct a bilinear map where the domain and target groups are identical? Such a bilinear map is called a *self-bilinear map*, and has previously been studied by Cheon and Lee [CL09]. They showed that a self-bilinear map is useful to construct cryptographic primitives by highlighting that it can be used for constructing a multilinear map [BS02]. However, in contrast to this useful property, they also proved an impossibility result: the computational Diffie-Hellman (CDH) assumption cannot hold in a group G of *known prime order* if there exists an efficiently computable self-bilinear map on G. This is undesirable for cryptographic applications. The overview of the proof is as follows. Let $e : G \times G \to G$ be a self-bilinear map and g be a generator of G, then we have $e(g^x, g^y) = e(g, g)^{xy} = g^{cxy}$ where c is an integer such that $e(g, g) = g^c$. Then we can compute g^{xy} by computing c-th root of $e(g^x, g^y)$ since G is a prime and known order group.[*1] However, their impossibility result cannot be applied for the case that G is a *unknown* order group. This is the setting we focus on in this chapter.

3.1.2 Our Contribution

We first introduce a new cryptographic primitive which we call a *self-bilinear map with auxiliary information* (AI-SBM) which can be seen as a weaker variant of an ideal self-bilinear map. Then we construct an AI-SBM by using indistinguishability obfuscation (iO) [GGH$^+$13b]. Though our self-bilinear map with auxiliary information has a limited functionality compared with an ideal self-bilinear map, we show that it is still useful to construct various cryptographic primitives. Especially, it is sufficient to instantiate some multilinear-map-based cryptographic primitives such as multiparty non-interactive key exchange (NIKE), broadcast encryption, attribute-based encryption for circuits and homomorphic signatures. Our multiparty NIKE and distributed broadcast encryption schemes are the first schemes where the number of users is not fixed in the setup phase. We also show that our technique can be used for constructing a somewhat homomorphic encryption scheme for NC_1 circuits.

Self-bilinear map with auxiliary information. For ideal self-bilinear map, we require that we can publicly compute $e(g^x, g^y)$ efficiently given g^x and g^y (and a description of the map e). We relax the notion of self-bilinear map to define AI-SBM so that $e(g^x, g^y)$ can be computed efficiently given an auxiliary information τ_x or τ_{g^y} corresponding to g^x or g^y in addition to g^x and g^y. An auxiliary information τ_x can

[*1] Here, we consider only the case in which c is known. However, [CL09] proved that the CDH assumption does not hold even if c is unknown as long as G is a group of known prime order.

be generated efficiently from x . Though this seems a significant relaxation, AI-SBMs are still useful for instantiating many multilinear-map based constructions.

We introduce various hardness assumptions w.r.t. AI-SBMs which we call Multilinear Computational Diffie-Hellman with Auxiliary Information (AI-MCDH), Multilinear Hashed Diffie-Hellman with Auxiliary information (AI-MHDH) and Augmented Power Multilinear Diffie-Hellman with Auxiliary Information (AI-APMDH) assumption as a counter part of assumptions w.r.t. multilinear maps. Moreover, we introduce the Mulitilinear Generalized Diffie-Hellman with Auxiliary Information (AI-GMDH) assumption which can be seen as an "uber assumption" [Boy08] w.r.t. AI-SBMs.

We give two constructions of AI-SBM based on indistinguishability obfuscation (iO), which we call basic and extended constructions. We prove that the AI-MCDH assumption and AI-MHDH assumption for one-bit output hash function hold w.r.t. the basic construction. For extended construction, we prove that the AI-MHDH assumption for multiple-bit output hash functions. Moreover, we give a sufficient condition such that the AI-GMDH assumption holds. As a result, we prove that the AI-APMDH assumption holds.

Applications of AI-SBM. As applications of AI-SBMs, we construct a multiparty NIKE, distributed broadcast encryption, ABE for circuits and homomorphic signatures. The details follow.

- **Multiparty NIKE.** Multiparty NIKE is a cryptographic primitive which enable multiple users to share a common key without any interaction. We construct a multiparty NIKE scheme where the maximum number of users is not fixed in the setup phase. In particular, the size of both the public parameters and a public key generated by a user are independent of the number of users. The construction is a natural extension of the Diffie-Hellman key exchange by using our multilinear map [DH76, BS02].
- **Distributed broadcast encryption.** Broadcast encryption is PKE where a sender can arbitrary decide a set of receivers, and only designated receivers can decrypt the ciphertext to obtain a message. Distributed broadcast encryption is broadcast encryption with an additional property that a user can join the system by himself without the assistance of a trusted third party holding a master key. We construct a distributed broadcast encryption scheme where the maximum number of users is not fixed in the setup phase based on our multiparty NIKE scheme. In particular, the size of both the public parameters and a ciphertext overhead are independent of the number of users. We apply the generic conversion from multiparty NIKE to distributed broadcast encryption

given in [BZ14].

- **ABE for circuits.** ABE is an extension of PKE which enable us an arbitrary access control depending on attributes assigned to each receiver. We construct an ABE scheme for general circuits by using our multilinear map. The construction is a simple analogue of the scheme in [GGH$^+$13c], which constructs an ABE scheme based on multilinear maps.
- **Homomorphic signatures.** Homomorphic signatures is digital signatures with a homomorphic property that anyone can publicly evaluate a function on signatures to generate a new signature for the function value on original messages. We construct a homomorphic signatures for polynomial-degree polynomials. The construction is a simple analogue of the scheme in [CFW14], which constructs homomorphic signatures based on multilinear maps. We only consider selective security for single data set for simplicity.

The above results can be interpreted as an evidence that AI-SBMs can replace existing multilinear maps in some applications since all of the above constructions are simple analogues of known multilinear-map-based constructions.

Somewhat homomorphic encryption. Besides direct applications of our self-bilinear map with auxiliary information, we construct a somewhat homomorphic encryption scheme by using a similar technique. Our somewhat homomorphic encryption scheme is chosen plaintext (CPA) secure, NC_1 circuit homomorphic and compact under the Φ-hiding assumption and the existence of iO.

Note that all known candidate constructions of indistinguishability obfuscation are far from practical, and hence, the above constructions are mostly of theoretical interest.

3.1.3 Technical Overview

Here, we give a technical overview of our result. Our basic idea is to avoid the impossibility result of self-bilinear maps which is explained above by considering a group of *unknown order*. Note that even if we consider such a group, many decisional assumptions such as the decisional Diffie-Hellman (DDH) assumption cannot hold if there exists an efficiently computable self-bilinear map on the group. Therefore we consider only computational assumptions such as the CDH assumption. For a Blum integer N, we consider the group $\mathbb{QR}_N^+$ of signed quadratic residues [HK09a]. On this group, we consider a self-bilinear map $e : \mathbb{QR}_N^+ \times \mathbb{QR}_N^+ \to \mathbb{QR}_N^+$ which is defined as $e(g^x, g^y) := g^{2xy}$. The reason why we define it in this manner is that we want to ensure that the CDH assumption holds in $\mathbb{QR}_N^+$, even if e is efficiently computable. That is, even if we can compute $e(g^x, g^y) = g^{2xy}$, it is difficult to compute g^{xy} from it since the Rabin function is hard to invert under the factoring assumption. However, given only the group elements g^x and g^y, we do not know how to compute $e(g^x, g^y)$ efficiently. To address this, we introduce *auxiliary information* τ_y for each element $g^y \in \mathbb{QR}_N^+$ which enables us to compute a map $e(\cdot, g^y)$ efficiently. This leads to the notion of *self-bilinear map with auxiliary information* which we introduce in this chapter.

The problem is how to define auxiliary information τ_y which enables us to compute $e(\cdot, g^y)$ efficiently. The most direct approach is to define τ_y as a circuit that computes the $2y$-th power. However, if we define τ_y as a "natural" circuit that computes the $2y$-th power, then we can extract $2y$ from τ_y, and thus we can compute y. This clearly enables us to compute g^{xy}, which breaks the CDH assumption.

A cleverer way is to define τ_y as a circuit that computes the t_y-th power where $t_y = 2y \pm \mathrm{ord}(\mathbb{QR}_N^+)$.[*2] In this way, it seems that τ_y does not reveal y since t_y is a "masked" value of $2y$ by $\mathrm{ord}(\mathbb{QR}_N^+)$ which is an unknown odd number. This idea is already used by Seurin [Seu13] to construct a trapdoor DDH group. Actually, he proved that even if t_y is given in addition to g^x and g^y, it is still difficult to compute g^{xy}. In this way, it seems that we can construct a self-bilinear map with auxiliary information. However, this creates a problem: we do not have an efficient algorithm to compute t_y from y without knowing the factorization of N. If such an algorithm does not exist, then we cannot instantiate many bilinear map-based primitives using the resulting map such as the 3-party Diffie-Hellman key exchange [Jou00].

To overcome the above difficulty, we use *indistinguishability obfuscation.* An indistinguishability obfuscator ($i\mathcal{O}$) is an efficient randomized algorithm that makes circuits C_0 and C_1 computationally indistinguishable if they have exactly the same functionality.

We observe that a circuit that computes the $2y$-th power and a circuit that computes the t_y-th power for an element of $\mathbb{QR}_N^+$ have exactly the same functionality since we have $t_y = 2y \pm \mathrm{ord}(\mathbb{QR}_N^+)$. Therefore if we obfuscate these circuits by $i\mathcal{O}$, then the resulting circuits are computationally indistinguishable. Then we define auxiliary information τ_y as an obfuscation of a circuit that computes the $2y$-th power. With this definition, it is clear that τ_y can be computed from y efficiently, and the above

[*2] In the definition of t_y, whether $+$ or $-$ is used depends on y. See [Seu13] for more details.

mentioned problem is solved. Moreover, τ_y is computationally indistinguishable from an obfuscation of a circuit that computes the t_y-th power. Therefore it must still be difficult to compute g^{xy} even if τ_y is given in addition to g^x and g^y.

Thus we obtain a self-bilinear map with auxiliary information on $\mathbb{QR}_N^+$ while ensuring that the auxiliary information does not allow the CDH assumption to be broken. By extending the above, we prove that the AI-MCDH assumption holds w.r.t. our AI-SBM under the security of $i\mathcal{O}$ and the factoring assumption. Moreover, we slightly modify the construction as $e(g^x, g^y) = g^{2^k xy}$ for some integer k, and give a sufficient condition such that the AI-GMDH assumption holds.

3.1.4 Related Work

Bilinear maps. In cryptography, bilinear maps on elliptic curves were first used for breaking the discrete logarithm problem on certain curves [MOV93]. The first constructive cryptographic applications of a bilinear map are given in [Jou00, SOK00, BF01]. Since then, many constructions of cryptographic primitives based on a bilinear map have been proposed.

Multilinear maps Boneh and Silverberg [BS02] considered a multilinear map which is an extension of a bilinear map, and showed its usefulness for constructing cryptographic primitives though they did not give a concrete construction of multilinear maps. Garg et al. [GGH13a] proposed a candidate construction of multilinear maps based on ideal lattices for the first time, and then some other constructions have been proposed [CLT13, CLT15, GGH15] [*3]. We note that some cryptanalysis on these schemes have been discussed [CHL$^+$15, HJ16, CFL$^+$16, CLLT16].

Indistinguishability Obfuscation The notion of indistinguishability obfuscation was first proposed by Barak et al. [BGI$^+$01]. The first candidate construction of indistinguishability obfuscation was proposed by Garg et al. [GGH$^+$13b], followed by many works [PST14, BR14, BGK$^+$14, AGIS14, GLSW15, GMM$^+$16]. Since then, many applications of indistinguishability obfuscation have been proposed [SW14, BZ14, HSW14, GGG$^+$14, GGHR14, Hof14, PPS15]. Although some constructions are broken [MSZ16], there are some constructions that remain unbroken so far.

Multilinar map from $i\mathcal{O}$. There are some works that shows the relation between multilinear maps and $i\mathcal{O}$. Paneth and Sahai [PS15] constructed a polynomial jigsaw puzzle,

[*3] More precisely, they construct *graded encoding systems* which can be seen as an approximate version of multilinear maps.

which is a variant of a multilinear map, solely based on $i\mathcal{O}$. However, they does not provide any application of polynomial jigsaw puzzles and thus it is unclear how that is useful in constructions of cryptographic primitives. Albrecht et al. [AFH$^+$16] constructed a multilinear map based on $i\mathcal{O}$, non-interactive zero-knowledge proof system, and additive homomorphic encryption. Since the assumptions they rely on is incomparable to ours, their result is incomparable to ours. Moreover, their multilinear map does not provide a graded encoding system [GGH13a] and thus some applications of multilinear maps such as attribute based encryption [GGH$^+$13c] and homomorphic signatures [CFW14] cannot be instantiated.

Multiparty non-interactive key exchange and broadcast encryption. Boneh et al. [BS02] observed that if there exists a cryptographic multilinear map, then we can construct a multiparty NIKE scheme and very efficient broadcast encryption scheme. Garg et al. [GGH13a] gave the first instantiation for the construction by proposing a multilinear map.

Boneh and Zhandry [BZ14] constructed a multiparty NIKE based on iO and a one-way function. Their scheme achieve stronger security than ours, in which they consider an active adversary who generates public keys maliciously. On the other hand, in their construction, the maximum number of users is bounded at the setup phase unlike ours.

Subsequent to our work, Khurana et al. [KRS15] also constructed a multiparty NIKE scheme for unbounded parties. Their scheme also achieve stronger security than ours like [BZ14]. On the other hand, they assume a cryptographic primitive called somewhere statistically binding commitment in addition to iO and a one-way function.

3.2 RSA modulus and Group of Signed Quadratic Residues

Here we state structures of RSA moduli considered in this chapter. In this chapter, for an RSA modulus $N = PQ$ output by RSAGen, we assume that $P \equiv Q \equiv 3 \bmod 4$ holds and all prime factors of $\Phi(N)/4 = (P-1)(Q-1)$ are pairwise distinct and larger than $2^{\delta\ell_N}$ for some positive constant δ like in [HK09a][*4]. In this case, $\mathbb{QR}_N^+$ is a cyclic group of order $(P-1)(Q-1)/4$, and a uniformly random element of $\mathbb{QR}_N$ is a generator of the group with overwhelming probability. A remarkable property of $\mathbb{QR}_N^+$ is that the group is efficiently recognizable. That is, there exists an efficient algorithm that

[*4] For example, these conditions are satisfied if N is a strong RSA modulus, i.e., $P = 2p + 1$ and $Q = 2q + 1$ for some primes p and q

determines whether a given string is an element of $\mathbb{QR}_N^+$ or not [HK09a]. Throughout this chapter, we assume that the factoring assumption holds w.r.t. RSAGen that satisfies the above properties.

3.3 Self-bilinear Maps

In this section, we recall the definition of a self-bilinear map [CL09]. Next, we introduce the notion of *self-bilinear map with auxiliary information* (AI-SBM) which is a weaker variant of a self-bilinear map. Finally we define hardness assumptions with respect to a multilinear map which is constructed from a self-bilinear map.

3.3.1 Definition of a Self-bilinear Map

First, we recall the definition of a self-bilinear map. A self-bilinear map is a bilinear map where the domain and target groups are identical. The formal definition is as follows.

Definition 4. *(Self-bilinear Map [CL09]) For a cyclic group G, a self-bilinear map $e : G \times G \to G$ has the following properties.*

- *For all $g_1, g_2 \in G$ and $\alpha \in \mathbb{Z}$, it holds that*

$$e(g_1^\alpha, g_2) = e(g_1, g_2^\alpha) = e(g_1, g_2)^\alpha.$$

- *The map e is non-degenerate, i.e, if $g_1, g_2 \in G$ are generators of G, then $e(g_1, g_2)$ is a generator of G.*

As shown in [CL09], we can construct an n-multilinear map for any integer $n \geq 2$ from a self-bilinear map e. We denote this n-multilinear map by e_n. This can be seen by easy induction: suppose that an n-multilinear map e_n can be constructed from a self-bilinear map e, then we can construct an $(n+1)$-multilinear map e_{n+1} by defining

$$e_{n+1}(g_1, \ldots, g_n, g_{n+1}) := e(e_n(g_1, \ldots, g_n), g_{n+1}).$$

3.3.2 Self-bilinear Map with Auxiliary Information

We usually expect a self-bilinear map to be efficiently computable for cryptographic applications. However, here we relax this requirement so that the map is efficiently computable if "auxiliary information" is given. That is, when we compute $e(g^x, g^y)$, we require auxiliary information τ_x or τ_y corresponding to g^x or g^y, respectively. We call this relaxed notion a *self-bilinear map with auxiliary information* (AI-SBM). We formalize it as a set of algorithms $\mathcal{SBP} = (\mathsf{InstGen}, \mathsf{AIGen}, \mathsf{Map}, \mathsf{AIMult})$.

$\mathsf{InstGen}(1^\lambda) \to \mathsf{params} = (G, e, g)$: InstGen takes the security parameter 1^λ as input and outputs the public parameters params which consists of descriptions of an efficiently recognizable cyclic group G on which the group operation is efficiently computable, a self-bilinear map e on G and an element g of G. We require that g is a generator of G with overwhelming probability and that an approximation $\mathrm{Approx}(G)$ of $\mathrm{ord}(G)$ can be computed efficiently from params, which is negligibly close to $\mathrm{ord}(G)$. By using g and $\mathrm{Approx}(G)$, we can generate an almost uniform element h of G by taking $x \xleftarrow{\$} [\mathrm{Approx}(G)]$ and outputting $h := g^x$. With a slight abuse of notation, we often simply write $h \xleftarrow{\$} G$ to mean the above procedure. Additionally, params specifies sets T_X of auxiliary information for all $X \in G$. Since params is input for all algorithms below, we omit it for simplicity.

$\mathsf{AIGen}(x) \to \tau_x$: AIGen takes an integer x as input, and outputs an auxiliary information $\tau_x \in T_{g^x}$ that corresponds to g^x.

$\mathsf{Map}(g^x, \tau_y) \to e(g^x, g^y)$: Map takes $g^x \in G$ and $\tau_y \in T_{g^y}$ as input and outputs $e(g^x, g^y)$. By using this algorithm iteratively, we can compute $e_n(g^{x_1}, g^{x_2}, \ldots, g^{x_n})$ if we are given $g^{x_1}, g^{x_2}, \ldots, g^{x_n}$[*5].

$\mathsf{AIMult}(\tau_x, \tau_y) \to \tau_{\mathsf{Mult}}$: AIMult takes $\tau_x \in T_{g^x}$, $\tau_y \in T_{g^y}$ as input and outputs $\tau_{\mathsf{Mult}} \in T_{g^{x+y}}$. We require that $|\tau_{\mathsf{mult}}| \leq |\tau_x| + |\tau_y| + \mathsf{poly}(\lambda)$ holds.

$\mathsf{AIMap}(\tau_x, \tau_y) \to \tau_{\mathsf{Map}}$: AIMap takes $\tau_x \in T_{g^x}$, $\tau_y \in T_{g^y}$ as input and outputs $\tau_{\mathsf{Map}} \in T_{e(g^x, g^y)}$. We require that $|\tau_{\mathsf{Map}}| \leq |\tau_x| + |\tau_y| + \mathsf{poly}(\lambda)$ holds.

$\mathsf{AIExp}(\tau_x, \alpha) \to \tau_{\mathsf{Exp}}$: AIMap takes $\tau_x \in T_{g^x}$ and a integer α as input and outputs $\tau_{\mathsf{Exp}} \in T_{g^{\alpha x}}$. We require that $|\tau_{\mathsf{Exp}}| \leq |\tau_x| + \mathsf{poly}(\lambda, \log \alpha)$ holds.

$\mathsf{AIRand}(S, \tau_x) \to \tau'_x$: AIRand takes a natural number S and $\tau_x \in T_{g^x}$ such that $|\tau_x| \leq S$ as input and outputs $\tau'_x \in T_{g^x}$ such that $|\tau'_x| \leq \mathsf{poly}(S, \lambda)$.

We require for AIRand to satisfy the following property.

- **Indistinguishability of auxiliary information.** Intuitively, two auxiliary information corresponding to the same group element output by

[*5] Actually, e_n can be computable even if one of $\tau_{g^{x_1}}, \ldots, \tau_{g^{x_n}}$ is not given.

AIRand are computationally indistinguishable. More formally, for any $\mathsf{params} \leftarrow \mathsf{InstGen}(1^\lambda)$, $g^x \in G$, if $\tau_{x,i} \in T_{g^x}$ and $|\tau_{x,i}| \leq S$ hold and we set $\tau'_{x,i} \leftarrow \mathsf{AIRand}(S, \tau_{x,i})$ $(i = 0, 1)$, then $\tau'_{x,0}$ and $\tau'_{x,1}$ are computationally indistinguishable.

Remark 7. *As pointed out in [AFH⁺16], AI-SBMs have a significant drawback compared with ideal self-bilinear maps that the size of auxiliary information grows almost double in each computation of* AIMult *and* AIMap*. Thus if we apply these computations recursively, then the size grows exponentially in the number of computations. Thus we cannot compute polynomial depth circuit on auxiliary information. We remark, however, that we can compute logarithmic depth circuits on auxiliary information. (If randomization is required after each gate evaluation, then we can handle only constant depth circuits.)*

3.3.3 Definition of Hardness Assumptions

Here, we introduce some hardness assumptions with respect to AI-SBM. We first define Auxiliary Information Multilinear Computational Diffie-Hellman (AI-MCDH) assumption, Auxiliary Information Multilinear Hashed Diffie-Hellman (AI-MHDH) assumption and the Auxiliary Information Augmented Power Multilinear Diffie-Hellman (AI-APMDH) assumptions, which are counterparts of the similar assumptions defined for multi-linear maps respectively. Finally, we define the Auxiliary Information Generalized Multilinear Diffie-Hellman (AI-GMDH) assumption, which can be seen as a general class of Diffie-Hellman type search assumptions with respect to AI-SBMs including the AI-MCDH and AI-APMDH assumptions.

First, we define the AI-MCDH assumption, which is an analogue of the multilinear computational Diffie-Hellman (MCDH) assumption defined for multilinear maps.

Definition 5. *(n-AI-MCDH assumption) We say that the n-Auxiliary Information Multilinear Computational Diffie-Hellman (n-MCDHAI assumption) holds if there exists a polynomial $S(\lambda)$ such that for any efficient algorithm $\mathcal{A}$,*

$$\Pr[e_n(g, \ldots, g)^{\prod_{i=1}^{n} x_i} \leftarrow \mathcal{A}(\mathsf{params}, \{g^{x_i}\}_{i \in [n+1]}, \{\tau_{x_i}\}_{i \in [n+1]})]$$

is negligible, where $\mathsf{params} \xleftarrow{\$} \mathsf{InstGen}(1^\lambda)$, $x_1, \ldots, x_{n+1} \xleftarrow{\$} [\mathrm{Approx}(G)]$ *and* $\tau_{x_i} \leftarrow \mathsf{AIRand}(S, \mathsf{AIGen}(x_i))$ *for* $i \in [n+1]$.

Next, we define the AI-MHDH assumption which is the“hashed version” of the AI-MCDH assumption.

Definition 6. *(n-AI-MHDH assumption) We say that the n-Auxiliary Information Multilinear Hashed Diffie-Hellman (n-AI-MHDH assumption) holds with respect to a family $\mathcal{H} = \{H : G \to \{0,1\}^k\}$ of hash functions if there exists a polynomial $S(\lambda)$ such that for any efficient algorithm $\mathcal{A}$,*

$$\begin{aligned} &\Pr[1 \leftarrow \mathcal{A}(\mathsf{params}, \{g^{x_i}\}_{i\in[n]}, \{\tau_{x_i}\}_{i\in[n+1]}, H, T)|\beta = 1] \\ -&\Pr[1 \leftarrow \mathcal{A}(\mathsf{params}, \{g^{x_i}\}_{i\in[n]}, \{\tau_{x_i}\}_{i\in[n+1]}, H, T)|\beta = 0] \end{aligned}$$

is negligible, where $\mathsf{params} \xleftarrow{\$} \mathsf{InstGen}(1^\lambda)$, $x_1, \dots, x_{n+1} \xleftarrow{\$} [\mathrm{Approx}(G)]$, $\tau_{x_i} \leftarrow \mathsf{AIRand}(S, \mathsf{AIGen}(x_i))$ *for* $i \in [n+1]$, $H \xleftarrow{\$} \mathcal{H}$, *and* $T := H(e_n(g, \dots, g)^{\prod_{i=1}^{n+1} x_i})$ *if* $\beta = 1$, *and otherwise* $T \xleftarrow{\$} \{0,1\}^k$.

Definition 7. *((ℓ, M)-AI-APMDH assumption) We say that the (ℓ, M)-auxiliary information augmented power multilinear Diffie-Hellman((ℓ, M)-AI-APMDH) holds if there exists a polynomial $S(\lambda)$ such that for any PPT adversary $\mathcal{A}$,*

$$\Pr[(c^*, F^{*c^*}) \leftarrow \mathcal{A}(\mathsf{params}, \{F_i\}_{i\in[4]}, \{\tau_{f_i}\}_{i\in[4]}), c^* \neq 0, |c^*| \leq M]$$

is negligible where $\mathsf{params} \xleftarrow{\$} \mathsf{InstGen}(1^\lambda)$, $x_1, x_2, x_3 \xleftarrow{\$} [\mathrm{Approx}(G)]$, $F_1 := g^{x_2}$, $F_2 := g^{x_3}$, $F_3 := g^{x_1 x_2}$, $F_4 := g^{x_1 x_2 x_3}$, $F^* := e_\ell(g, \dots, g)^{x_1^{\ell-1}(x_2 x_3)^\ell}$, $\tau_{f_1} \leftarrow \mathsf{AIRand}(S, \mathsf{AIGen}(x_2))$, $\tau_{f_2} \leftarrow \mathsf{AIRand}(S, \mathsf{AIGen}(x_3))$, $\tau_{f_3} \leftarrow \mathsf{AIRand}(S, \mathsf{AIGen}(x_1 x_2))$, *and* $\tau_{f_4} \leftarrow \mathsf{AIRand}(S, \mathsf{AIGen}(x_1 x_2 x_3))$.

Remark 8. *In the original definition of the APMDH assumption, an adversary is also given g^{x_1} and $g^{x_1 x_3}$ additionally. In our application in Sec. 3.5.4, they are not needed and thus we omit them.*

Remark 9. *If G is a group of known prime order, then it is the same if we only consider the case of $c^* = 1$. However, since we consider a group of unknown order, we formulate the assumption as the above.*

Next, we define the AI-GMDH assumption which generalizes the above assumptions.

Definition 8. *($(\{f_i\}_{i\in[m]}, f^*, \ell^*, M)$-AI-GMDH assumption) Let $f_1, \dots, f_m$, f^* be n-variable polynomials and ℓ^* and M be natural numbers. Then we say that the Auxiliary Information Generalized Multilinear Diffie-Hellman (AI-GMDH) assumption holds if the following holds. There exists a polynomial $S(\lambda)$ such that for any PPT adversary $\mathcal{A}$,*

$$\Pr[(c^*, F^{*c^*}) \leftarrow \mathcal{A}(\mathsf{params}, \{F_i\}_{i\in[m]}, \{\tau_{f_i}\}_{i\in[m]}), c^* \neq 0, |c^*| \leq M]$$

is negligible, where $\mathsf{params} \xleftarrow{\$} \mathsf{InstGen}(1^\lambda)$, $x_1, \ldots, x_m \xleftarrow{\$} [\mathrm{Approx}(G)]$, $F_i := g^{f_i(x_1,\ldots,x_n)}$, $\tau_{f_i} \leftarrow \mathsf{AIRand}(S, \mathsf{AIGen}(f_i(x_1,\ldots,x_n)))$ *(for* $i \in [m]$*), and* $F^* := e_{\ell^*}(g,\ldots,g)^{f^*(x_1,\ldots,x_n)}$.

Example 1. *If we define* $f_i(x_1,\ldots,x_{n+1}) := x_i$ *for* $i \in [n+1]$*, and* $f^*(x_1,\ldots,x_{n+1}) := \prod_{i=1}^{n+1} x_i$*, then the* n*-AI-MCDH assumption is equivalent to* $(\{f_i\}_{i\in[n]}, f^*, n, 1)$*-AI-GMDH assumption.*

Example 2. *If we define* $f_1(x_1,x_2,x_3) := x_2$*,* $f_2(x_1,x_2,x_3) := x_3$*,* $f_3(x_1,x_2,x_3) := x_1x_2$*,* $f_4(x_1,x_2,x_3) := x_1x_2x_3$*, and* $f^*(x_1,x_2,x_3) := x_1^{\ell-1}(x_2x_3)^\ell$*, then the* (ℓ, M)*-AI-APMDH assumption is equivalent to the* $(\{f_i\}_{i\in[4]}, f^*, \ell, M)$*-AI-GMDH assumption.*

3.4 Constructions of AI-SBM

In this section, we give two constructions of AI-SBM. We call the first one the basic construction and the second one the extended construction. For the basic construction, we prove that the AI-MCDH assumption and the AI-MHDH assumption for a one-bit output hash function hold under the security of iO and the factoring assumption. For the extended construction, we show that the AI-MHDH assumption for a multiple-bit output hash function holds under the same assumption, and give a easily checkable sufficient condition for parameters such that the $((\{f_i\}_{i\in[m]}, f^*, \ell^*, M)$-AI-GMDH assumption holds.

3.4.1 Basic Construction

First we prepare some notations for circuits on $\mathbb{QR}_N^+$.

Notation for circuits on $\mathbb{QR}_N^+$. In the following, for an ℓ_N-bit RSA modulus N and an integer $x \in \mathbb{Z}$, $\mathcal{C}_{N,x}$ denotes a set of circuits $C_{N,x}$ that computes x-th power on the group $\mathbb{QR}_N^+$. If an input is not an element of $\mathbb{QR}_N^+$, $C_{N,x}$ outputs 0^{ℓ_N} (that is interpreted as $\perp$). We define the canonical circuit $\tilde{C}_{N,x}$ in $\mathcal{C}_{N,x}$ in a natural way [*6]. For circuits C_1, C_2 whose output can be interpreted as elements of $\mathbb{QR}_N^+$, $\mathsf{Mult}(C_1, C_2)$ denotes a circuit that takes a as input and outputs $C_1(a) \cdot C_2(a)$ where $\cdot$ denotes the multiplication on $\mathbb{QR}_N^+$. $C_1 \circ C_2$ denotes a circuit that takes a as input and outputs $C_1(C_2(a))$. The sizes of $\mathsf{Mult}(C_1, C_2)$ and $C_1 \circ C_2$ can be bounded by

[*6] There is flexibility for the definition of the canonical circuit. However, any definition works if the size of $\tilde{C}_{N,x}$ is polynomially bounded in λ and $|x|$.

$|C_1| + |C_2| + \mathsf{poly}(\log N)$.

Now we are ready to describe the construction. Let k be an arbitrary natural number. The construction is as follows.

$\mathsf{InstGen}(1^\lambda) \to \mathsf{params} = (N, e, g)$: Run $\mathsf{RSAGen}(1^\lambda)$ to obtain (N, P, Q), chooses $g \xleftarrow{\$} \mathbb{QR}_N^+$ and outputs $\mathsf{params} = (N, g)$. params defines the underlying group $G := \mathbb{QR}_N^+$, the self bilinear map $e(g^x, g^y) := g^{2xy}$ and $\mathrm{Approx}(G) := (N-1)/4$. For any element $X = g^x \in G$, the set T_X is defined as the set of all circuits that computes $2x$-th power on $\mathbb{QR}_N^+$ (and outputs $\perp$ for input out of $\mathbb{QR}_N^+$).

$\mathsf{AIGen}(x) \to \tau_x$: Take the canonical circuit $\tilde{C}_{N,2x} \in \mathcal{C}_{N,2x}$, set $\tau_x := \tilde{C}_{N,2x}$ and output τ_x.

$\mathsf{Map}(g^x, \tau_y) \to e(g^x, g^y)$: Compute $\tau_y(g^x)$ and output it. (Recall that τ_y is a circuit that computes the $2y$-th power for an element of $\mathbb{QR}_N^+$.)

$\mathsf{AIMult}(\tau_x, \tau_y) \to \tau_{\mathsf{Mult}}$: Compute $\tau_{\mathsf{Mult}} \leftarrow \mathsf{Mult}(\tau_x, \tau_y)$ and output it.

$\mathsf{AIMap}(\tau_x, \tau_y) \to \tau_{\mathsf{Map}}$: Compute $\tau_{\mathsf{Map}} \leftarrow \tau_x \circ \tau_y$ and output it.

$\mathsf{AIExp}(\tau_x, \alpha) \to \tau_{\mathsf{Exp}}$: Take the canonical circuit $\tilde{C}_{N,\alpha} \in \mathcal{C}_{N,\alpha}$, compute $\tau_x \circ \tilde{C}_{N,\alpha}$ and output it.

$\mathsf{AIRand}(S, \tau_x) \to \tau'_x$: Compute $\tau'_x \leftarrow i\mathcal{O}(S, \tau_x)$ and output it.

The indistinguishability of auxiliary information easily follows from the definition of indistinguishability obfuscation.

Hardness Assumptions In this paragraph, we show that various hardness assumptions hold w.r.t. our construction of AI-SBM. We first prove that the AI-MCDH assumption holds with respect to our AI-SBM if $i\mathcal{O}$ is a secure indistinguishability obfuscator for $P/poly$ and the factoring assumption holds. Then we observe that if we use the Goldreich-Levin hardcore bit function [GL89] as $\mathcal{H}$, the AI-MHDH assumption also holds. Finally, by extending the above proof technique, we show that more general class of assumptions also holds with respect to our AI-SBM. In particular, we give a sufficient condition such that the AI-GMDH assumption holds. As a corollary of the theorem, we prove that the AI-APMDH assumption holds w.r.t. the AI-SBM.

First, we prove that the AI-MCDH assumption holds if $i\mathcal{O}$ is a secure indistinguishability obfuscator for $P/poly$ and the factoring assumption holds. We note that similar idea to our proof can be found in [HK09a], where it is proven that Strong Diffie-Hellman (SDH) assumption on $\mathbb{QR}_N^+$ holds under the factoring assumption, and in [Seu13], where trapdoor DDH group is constructed on $\mathbb{QR}_N^+$.

Theorem 1. *For any integer n (polynomially bounded by the security parameter), the n-AI-MCDH assumption holds w.r.t. the above AI-SBM if the factoring assumption holds w.r.t.* RSAGen *and $i\mathcal{O}$ is an indistinguishability obfuscator for P/poly.*

Proof. For an algorithm $\mathcal{A}$ and an integer n (which is polynomially bounded by the security parameter), we consider the following games.

Game 1. This game is the original n-MCDHAI game. More precisely, it is as follows.

$(N, P, Q) \leftarrow \mathsf{RSAGen}(1^\lambda)$
$g \xleftarrow{\$} \mathbb{QR}_N^+$
$x_1, \ldots, x_{n+1} \xleftarrow{\$} [(N-1)/4]$
$\tau_{x_i} \leftarrow i\mathcal{O}(S, \tilde{C}_{N,2x_i})$ for $i \in [n+1]$
$U \leftarrow \mathcal{A}(N, g, \{g^{x_i}\}_{i\in[n+1]}, \{\tau_{x_i}\}_{i\in[n+1]})$

Game $1'$ This game is the same as Game 1 except that $x_1, \ldots, x_{n+1}$ are uniformly chosen from $[\mathrm{ord}(\mathbb{QR}_N^+)]$.

Game $2'$ This game is the same as Game $1'$ except that g, $x_1, \ldots, x_{n+1}$, $\tau_{x_1}, \ldots, \tau_{x_{n+1}}$ are set differently. More precisely, they are set as follows.

$(N, P, Q) \leftarrow \mathsf{RSAGen}(1^\lambda)$
$h \xleftarrow{\$} \mathbb{QR}_N^+$
$g := h^2$
$x'_0, \ldots, x'_n \xleftarrow{\$} [\mathrm{ord}(\mathbb{QR}_N^+)]$
$g^{x_i} := g^{x'_i} h$ for $i \in [n+1]$
(This implicitly defines $x_i \equiv x'_i + 1/2 \bmod \mathrm{ord}(\mathbb{QR}_N^+)$).
$\tau_{x_i} \leftarrow i\mathcal{O}(S, \tilde{C}_{N,2x'_i+1})$ for $i \in [n+1]$
$U \leftarrow \mathcal{A}(N, g, \{g^{x_i}\}_{i\in[n+1]}, \{\tau_{x_i}\}_{i\in[n+1]})$

Game 2 This game is the same as Game $2'$ except that $x'_0, \ldots, x'_{n+1}$ are uniformly chosen from $[(N-1)/4]$.

We say that $\mathcal{A}$ wins if it outputs $U = e_n(g, \ldots, g)^{\Pi_{i=1}^{n+1} x_i}$. For $i = 1, 2$, we let T_i and T'_i be the events that $\mathcal{A}$ wins in Game i and Game i', respectively. What we want to prove is that $\Pr[T_1]$ is negligible. We prove it by the following lemmas.

Lemma 1. $|\Pr[T_i] - \Pr[T'_i]|$ *is negligible for $i = 1, 2$*

Proof. This follows since the statistical distance of uniform distributions on $[(N-1)/4]$ and $[\mathrm{ord}(\mathbb{QR}_N^+)]$ are negligible. □

Lemma 2. $|\Pr[T'_1] - \Pr[T'_2]|$ *is negligible if* $i\mathcal{O}$ *is an indistinguishability obfuscator for P/poly.*

Proof. We consider hybrid games $H_0, \ldots H_{n+1}$. A hybrid game H_i is the same as $\mathsf{Game}\ 1'$ except that the first i auxiliary information (i.e, $\tau_{x_1}, \ldots, \tau_{x_i}$) are generated as in $\mathsf{Game}\ 2'$. It is clear that H_0 is identical to $\mathsf{Game}\ 1'$ and H_n is identical to $\mathsf{Game}\ 2'$. Let S_i be the event that $\mathcal{A}$ wins in $\mathsf{Game}\ H_i$. It suffices to show that $|\Pr[S_i] - \Pr[S_{i-1}]|$ is negligible. We construct an algorithm $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ that breaks the security of $i\mathcal{O}$ by using $\mathcal{A}$ that distinguishes H_i and H_{i-1}.

$\mathcal{B}_1(1^\lambda)$: $\mathcal{B}_1$ runs $(N, P, Q) \leftarrow \mathsf{RSAGen}(1^\lambda)$, chooses $h \xleftarrow{\$} \mathbb{QR}_N^+$ and $x_1, \ldots, x_{n+1} \xleftarrow{\$} [\mathrm{ord}(\mathbb{QR}_N^+)]$ and sets $g := h^2$. $\mathcal{B}_1$ computes $x'_0, \ldots, x'_{n+1} \in \mathrm{ord}(\mathbb{QR}_N^+)$ such that $x_j \equiv x'_j + 1/2 \bmod \mathrm{ord}(\mathbb{QR}_N^+)$ for $j \in [n+1]$. (This can be computed since $\mathcal{B}_1$ knows the factorization of N.) Then $\mathcal{B}_1$ sets $C_0 := \tilde{C}_{N,2x_i}$, $C_1 := \tilde{C}_{N,2x'_i+1}$ and $\sigma := (N, P, Q, h, g, x_1, \ldots, x_n, x'_1, \ldots, x'_n)$ and outputs (C_0, C_1, σ).

$\mathcal{B}_2(\sigma, C^*)$: $\mathcal{B}_2$ sets

$$\tau_{x_j} \leftarrow \begin{cases} i\mathcal{O}(S, \tilde{C}_{N,2x'_j+1}) & \text{if } j = 0, \ldots, i-1 \\ C^* & \text{if } j = i \\ i\mathcal{O}(S, \tilde{C}_{N,2x_j}) & \text{if } j = i+1, \ldots, n+1. \end{cases}$$

Then $\mathcal{B}_2$ runs $\mathcal{A}(N, g, \{g^{x_i}\}_{i\in[n+1]}, \{\tau_{x_i}\}_{i\in[n+1]})$ to obtain U. If we have $U = e_n(g, \ldots, g)^{\Pi_{i=0}^{n+1} x_i}$, then $\mathcal{B}_2$ outputs 1, and otherwise outputs 0.

The above completes the description of $\mathcal{B}$. First, we show C_0 and C_1 output by $\mathcal{B}_1$ has completely the same functionality. Since we have $x_j \equiv x'_j + 1/2 \bmod \mathrm{ord}(\mathbb{QR}_N^+)$, we have $2x_j \equiv 2x'_j + 1 \bmod \mathrm{ord}(\mathbb{QR}_N^+)$. Therefore $2x_j$-th power and $(2x'_j + 1)$-th power return exactly the same value on the group $\mathbb{QR}_N^+$ and thus C_0 and C_1 have exactly the same functionality. We note that each of g^{x_j} $(j = 0, \ldots, n)$ is distributed in $\mathbb{QR}_N^+$ independently of each other in all hybrid games H_i for $i = 0, \ldots, n$. Therefore $\mathcal{B}$ generates them in exactly the same way as those are generated in the hybrids H_{i-1} and H_i. Then we can see that $\mathcal{B}$ perfectly simulates H_{i-1} if $C^* \leftarrow i\mathcal{O}(S, C_0)$ and H_i if $C^* \leftarrow i\mathcal{O}(S, C_1)$ from the view of $\mathcal{A}$. If the difference between the probability that $\mathcal{A}$ wins in H_{i-1} and that in H_i is non-negligible, then $\mathcal{B}$ succeeds in distinguish whether C^* is computed as $C^* \leftarrow i\mathcal{O}(S, C_0)$ or $C^* \leftarrow i\mathcal{O}(S, C_1)$, with non-negligible advantage, and thus breaks the security of $i\mathcal{O}$.

□

Lemma 3. $\Pr[T_2]$ *is negligible if the factoring assumption holds.*

Proof. Assuming that $\mathcal{A}$ wins in Game 2 with non-negligible probability, we construct an algorithm $\mathcal{B}$ that factorizes N. This part is very similar to techniques used in [HK09a, Seu13]. The construction of $\mathcal{B}$ is as follows.

$\mathcal{B}(N)$: $\mathcal{B}$ chooses $h' \xleftarrow{\$} \mathbb{Z}_N^* \setminus \mathbb{QR}_N^+$, sets $h := |h'^2 \mod N| \in \mathbb{QR}_N^+$ and $g := h^2$ and chooses $x_1', \ldots, x_n' \xleftarrow{\$} [(N-1)/4]$. Then $\mathcal{B}$ sets $g^{x_i} := g^{x_i'}h$ and $\tau_{x_i} \leftarrow i\mathcal{O}(M_1, \tilde{C}_{N,2x_i'+1})$ for all $i \in [n+1]$. Then $\mathcal{B}$ runs $\mathcal{A}(N, g, \{g^{x_i}\}_{i\in[n+1]}, \{\tau_{x_i}\}_{i\in[n+1]})$. Let U be the output of $\mathcal{A}$. Then $\mathcal{B}$ computes $X := \Pi_{i=1}^{n+1}(2x_i'+1)$ and $v = Ug^{-(x_0'X+(X-1)/2)}$. (Note that X is odd and therefore $(X-1)/2$ is an integer.) Then it outputs $\gcd(h', V)$.

Since $\mathcal{B}$ perfectly simulates Game 2 from the view of $\mathcal{A}$, $\mathcal{A}$ outputs $e_n(g, \ldots, g)^{\Pi_{i=1}^{n+1} x_i}$ with non-negligible probability. If it occurs, then we have

$$\begin{aligned} U = e_n(g, \ldots, g)^{\Pi_{i=1}^{n+1} x_i} &= g^{2^{n-1}\Pi_{i=1}^{n+1} x_i} = h^{2^n \Pi_{i=1}^{n+1} x_i} = h^{x_1 \Pi_{i=2}^{n+1} 2x_i} \\ &= h^{(x_1'+1/2)\Pi_{i=1}^{n+1}(2x_i'+1)} = h^{x_1'X+X/2} = h^{x_1'X+(X-1)/2+1/2} \end{aligned}$$

where we used that $x_i \equiv x_i' + 1 \mod \operatorname{ord}(\mathbb{QR}_N)$ holds for $i \in [n+1]$. Therefore we have $V = h^{1/2}$. Since $V \in \mathbb{QR}_N^+$, h' and V are distinct square roots of h in $\mathbb{QR}_N^+$. Therefore $\gcd(h', V)$ is a non-trivial factor of N. □

Theorem 1 is proven by the above lemmas. □

The following is immediate from Theorem 1 and the Goldreich-Levin theorem [GL89].

Theorem 2. *The AI-MHDH assumption holds w.r.t. the above AI-SBM and the Goldreich-Levin hardcore bit function if the factoring assumption holds with respect to* RSAGen *and* $i\mathcal{O}$ *is an indistinguishability obfuscator for P/poly.*

3.4.2 Extended Construction

In this section, we construct a variant of the previous construction, for which various kinds of assumptions holds. Specifically, we first prove that the AI-MHDH assumption holds for a hash function with multi-bit output. We also show a sufficient condition such that the AI-GMDH assumption holds. As a corollary, we show that the AI-APMDH assumption holds w.r.t. the construction.

The construction is actually very similar to the basic construction. The difference from the previous construction is that we define $e(g^x, g^y) = g^{2^k xy}$ for some integer k instead of $e(g^x, g^y) = g^{2xy}$.

Let k be an arbitrary natural number. The construction is as follows.

$\mathsf{InstGen}(1^\lambda) \to \mathsf{params} = (N, e, g)$: Run $\mathsf{RSAGen}(1^\lambda)$ to obtain (N, P, Q), chooses $g \xleftarrow{\$} \mathbb{QR}_N^+$ and outputs $\mathsf{params} = (N, g)$. params defines the underlying group $G := \mathbb{QR}_N^+$, the self bilinear map $e(g^x, g^y) := g^{2^k xy}$ and $\mathrm{Approx}(G) := (N-1)/4$. For any element $X = g^x \in G$, the set T_X is defined as the set of all circuits that computes $2^k x$-th power on $\mathbb{QR}_N^+$ (and outputs $\perp$ for input out of $\mathbb{QR}_N^+$).

$\mathsf{AIGen}(x) \to \tau_x$: Take the canonical circuit $\tilde{C}_{N,2^k x} \in \mathcal{C}_{N,2^k x}$, set $\tau_x := \tilde{C}_{N,2^k x}$ and output τ_x.

$\mathsf{Map}(g^x, \tau_y) \to e(g^x, g^y)$: Compute $\tau_y(g^x)$ and output it. (Recall that τ_y is a circuit that computes the $2^k y$-th power for an element of $\mathbb{QR}_N^+$.)

$\mathsf{AIMult}(\tau_x, \tau_y) \to \tau_{\mathsf{Mult}}$: Compute $\tau_{\mathsf{Mult}} \leftarrow \mathsf{Mult}(\tau_x, \tau_y)$ and output it.

$\mathsf{AIMap}(\tau_x, \tau_y) \to \tau_{\mathsf{Map}}$: Compute $\tau_{\mathsf{Map}} \leftarrow \tau_x \circ \tau_y$ and output it.

$\mathsf{AIExp}(\tau_x, \alpha) \to \tau_{\mathsf{Exp}}$: Take the canonical circuit $\tilde{C}_{N,\alpha} \in \mathcal{C}_{N,\alpha}$, compute $\tau_{\mathsf{Exp}} \leftarrow \tau_x \circ \tilde{C}_{N,\alpha}$ and output it.

$\mathsf{AIRand}(S, \tau_x) \to \tau'_x$: Compute $\tau'_x \leftarrow i\mathcal{O}(S, \tau_x)$ and output it.

The indistinguishability of auxiliary information easily follows from the definition of indistinguishability obfuscation.

Hardness assumptions We prove that the AI-MHDH assumption holds w.r.t. the construction.

We first define the BBS generator which we will use as a hardcore function.

Definition 9. *For ℓ_N-bit Blum integer N, $g \in \mathbb{QR}_N^+$ and $r \in \{0,1\}^{\ell_N}$, we define the BBS generator as*

$$BBS_r(g) := (\mathsf{GL}_r(g), \ldots, \mathsf{GL}_r(g^{k-1}))$$

where GL denote the Goldreich-Levin hardcore bit function [GL89]. That is, $\mathsf{GL}_r(x) := \bigoplus_{i=1}^{\ell_N} r_i x_i$ where r_i and x_i are i-th bit of r and x which is represented as an integer in $\{1, \ldots, (N-1)/2\}$. We write $\mathcal{BBS}$ to denote the family of functions $\{BBS_r\}_{r\in\{0,1\}^{\ell_N}}$.

Then we show the following theorem.

Theorem 3. *The AI-MHDH assumption holds w.r.t. the above AI-SBM and $\mathcal{BBS}$ if* *the factoring assumption holds for* RSAGen *and* $i\mathcal{O}$ *is an indistinguishability obfuscator for* $P/poly$.

Proof. For an algorithm $\mathcal{A}$, we consider the following games.

Game 1. This game is the original n-MHDH game. More precisely, it is as follows.

$$\begin{aligned}
&(N,P,Q) \leftarrow \mathsf{RSAGen}(1^\lambda)\\
&g \xleftarrow{\$} \mathbb{QR}_N^+\\
&r \xleftarrow{\$} \{0,1\}^{\ell_N}\\
&x_1,\dots,x_{n+1} \xleftarrow{\$} [(N-1)/4]\\
&\tau_{x_i} \leftarrow i\mathcal{O}(M_{\ell_i}, \tilde{C}_{N,2^k x_i}) \text{ for } i \in [n+1]\\
&T := BBS_r(g^{2^{k(n-1)}\Pi_{i=1}^{n+1}x_i})\\
&b \leftarrow \mathcal{A}(N, g, g^{x_1},\dots,g^{x_{n+1}}, \tau_{x_1}\dots,\tau_{x_{n+1}}, r, T)
\end{aligned}$$

Game $1'$ This game is the same as Game 1 except that $x_1,\dots,x_n$ are chosen from $[\mathrm{ord}(\mathbb{QR}_N^+)]$.

Game $2'$. This game is the same as Game 1 except that g, $x_1,\dots,x_{n+1}$, $\tau_{x_1},\dots,\tau_{x_{n+1}}$ are set differently. More precisely, it is as follows.

$$\begin{aligned}
&(N,P,Q) \leftarrow \mathsf{RSAGen}(1^\lambda)\\
&h \xleftarrow{\$} \mathbb{QR}_N^+\\
&g := h^{2^k}\\
&x'_1,\dots,x'_{n+1} \xleftarrow{\$} [\mathrm{ord}(\mathbb{QR}_N^+)]\\
&g^{x_i} := g^{x'_i}h \text{ for } i \in [n+1]\\
&(\text{This implicitly defines } x_i \equiv x'_i + 1/2^k \bmod \mathrm{ord}(\mathbb{QR}_N^+))\\
&\tau_{x_i} \leftarrow i\mathcal{O}(M_{\ell_i}, \tilde{C}_{N,2^k x'_i+1}) \text{ for } i \in [n+1]\\
&T := BBS_r(g^{2^{k(n-1)}\Pi_{i=1}^{n+1}x_i})\\
&b \leftarrow \mathcal{A}(N, g, g^{x_1},\dots,g^{x_{n+1}}, \tau_{x_1}\dots,\tau_{x_{n+1}}, r, T)
\end{aligned}$$

Game 2 This game is the same as Game $2'$ except that $x'_1,\dots,x'_{n+1}$ are chosen from $[(N-1)/4]$.

Game 3. This game is the same as Game 2 except that T is set as a random k-bit string.

Let T_i be the event that $\mathcal{A}$ outputs 1 in Game i and T'_i be the event that $\mathcal{A}$ outputs 1 in Game i'. What we want to prove is $|\Pr[T_1] - \Pr[T_3]|$ is negligible. We prove this by the following lemmas.

Lemma 4. $|\Pr[T_i] - \Pr[T'_i]|$ *is negligible for* $i = 1, 2$

Proof. This follows since the statistical distance between the uniform distributions of $[(N-1)/4]$ and $[\mathrm{ord}(\mathbb{QR}_N^+)]$ are negligible. □

Lemma 5. $|\Pr[T'_1] - \Pr[T'_2]|$ *is negligible if* $i\mathcal{O}$ *is an indistinguishability obfuscator for P/poly.*

Proof. We define hybrid games $H_{1,0}, \dots H_{1,n+1}$. A hybrid game $H_{1,i}$ is the same as $\mathsf{Game}\ 1'$ except that the first i auxiliary information (i.e, $\tau_{x_1}, \dots, \tau_{x_i}$) are generated as in $\mathsf{Game}\ 2'$. Let $T_{1,i}$ be the event that $\mathcal{A}$ outputs 1 in the hybrid $H_{1,i}$. It is clear that $H_{1,0}$ is $\mathsf{Game}\ 1'$ and $H_{1,n}$ is $\mathsf{Game}\ 2'$. Let $T_{1,i}$ be the event that $\mathcal{A}$ wins in Game $H_{1,i}$. Since we have $x_i \equiv x'_i + 1/2^k \bmod \mathrm{ord}(\mathbb{QR}_N^+)$, $C_{N,2^k x'_i+1}$ computes exactly the same as $C_{N,2^k x_i}$ for any input for $i = 1, \dots n+1$. (Recall that these circuits computes the exponentiation only for an element of $\mathbb{QR}_N^+$.) Then we can see that $|\Pr[T_{1,i}] - \Pr[T_{1,i-1}]$ is negligible for $i \in [n+1]$ from the security of $i\mathcal{O}$. (Note that a reduction algorithm knows the factorization of N.) □

Lemma 6. $|\Pr[T_2] - \Pr[T_3]|$ *is negligible if the factoring assumption holds for* RSAGen *and* $i\mathcal{O}$ *is an indistinguishability obfuscator for P/poly.*

Proof. We define hybrid games $H_{2,0}, \dots H_{2,k}$. For $i = 0, 1, \dots, k$, a hybrid game $H_{2,i}$ is the same as $\mathsf{Game}\ 2$ except that the first i-bit of T are set as in $\mathsf{Game}\ 2$ and other bits are set as in $\mathsf{Game}\ 3$, i.e, $T := U_1 || \dots || U_i || \mathsf{GL}_r(g^{2^{k(n-1)+i} \Pi_{j=1}^{n+1} x_j}) || \dots || \mathsf{GL}_r(g^{2^{kn-1} \Pi_{j=1}^{n+1} x_j})$, where $U_1 \dots U_i \xleftarrow{\$} \{0,1\}$. In the following, we write $\mathsf{GL}(r, i)$ to denote $\mathsf{GL}_r(g^{2^{k(n-1)+i} \Pi_{j=1}^{n+1} x_j})$ for notational simplicity. It is clear that $H_{2,0}$ is the same as $\mathsf{Game}\ 2$ and $H_{2,k}$ is the same as $\mathsf{Game}\ 3$. Let $T_{2,i}$ be the event that $\mathcal{A}$ outputs 1 in the hybrid $H_{2,i}$. We prove that $|\Pr[T_{2,i-1}] - \Pr[T_{2,i}]|$ is negligible for all $i \in [k]$. To do so, we assume that there exists an algorithm $\mathcal{A}$ that distinguishes $H_{2,i}$ and $H_{2,i-1}$, and construct a factoring algorithm by using $\mathcal{A}$. Without loss of generality, we can assume that there exists a negligible function ϵ such that $\Pr[T_{2,i-1}] - \Pr[T_{2,i}] > \epsilon$. This is because given $\mathcal{A}$, the sign of $\Pr[T_{2,i-1}] - \Pr[T_{2,i}]$ can be checked efficiently, and if $\Pr[T_{2,i-1}] - \Pr[T_{2,i}] < 0$ then we can modify $\mathcal{A}$ to output inverse of the original output so that $\Pr[T_{2,i-1}] - \Pr[T_{2,i}] > 0$. In the following, we use a similar argument as in [MLLJ11].

Hardcore Predictor $\mathcal{P}$. First, we construct an algorithm $\mathcal{P}$ that predicts $\mathsf{GL}(r, i-1)$ with non-negligible advantage when it is given $(r, N, g, g^{x_1}, \dots, g^{x_{n+1}},$

$\tau_{x_1}\ldots,\tau_{x_{n+1}},g^{2^{k(n-1)+i}\Pi_{j=1}^{n+1}x_j})$ where r, N, g, $x_1,\ldots,x_{n+1}$ and $\tau_{x_1}\ldots,\tau_{x_{n+1}}$ are defined as in Game 2. The construction of $\mathcal{P}$ is as follows.

$\mathcal{P}(N,g,g^{x_1},\ldots,g^{x_n+1},\tau_{x_1}\ldots,\tau_{x_{n+1}},g^{2^{k(n-1)+i}\Pi_{j=1}^{n+1}x_j},r)$: $\mathcal{D}'$ picks $b \xleftarrow{\$} \{0,1\}$, sets $T := U_1||\ldots||U_{i-1}||b||\mathsf{GL}(r,i)||\ldots||\mathsf{GL}(r,k-1)$ and runs $\mathcal{A}(N,g,g^{x_1},\ldots,g^{x_{n+1}},\tau_{x_1}\ldots,\tau_{x_{n+1}},r,T)$. Note that $\mathcal{D}$' can generate $\mathsf{GL}(r,i),\ldots,\mathsf{GL}(r,k-1)$ since it knows $g^{2^{k(n-1)+i}\Pi_{j=1}^{n+1}x_j}$. If $\mathcal{A}$ outputs 1, then $\mathcal{P}$ outputs b, and otherwise it picks an independently random bit $b' \xleftarrow{\$} \{0,1\}$ and outputs it.

We define $Y' := (N,g,g^{x_1},\ldots,g^{x_{n+1}},\tau_{x_1}\ldots,\tau_{x_{n+1}},g^{2^{k(n-1)+i}\Pi_{i=1}^{n+1}x_i})$, i.e., Y' denotes input of $\mathcal{P}$ except r and define $Y := (N,g,g^{x_1},\ldots,g^{x_{n+1}},\tau_{x_1}\ldots,\tau_{x_{n+1}})$, i.e., Y denotes input of $\mathcal{P}$ except r and T. We prove that with the probability at least $\epsilon/2$ over the choice of Y', $\mathcal{P}$ predicts $\mathsf{GL}(r,i-1)$ with advantage $\epsilon/4$. By the standard averaging argument, with at least $\epsilon/2$ fraction of the choice of Y, we have

$$\begin{aligned}&\Pr[1\leftarrow\mathcal{A}(Y,r,U_1||\ldots||U_{i-1}||\mathsf{GL}(r,i-1)||\ldots||\mathsf{GL}(r,k-1))]\\ &-\Pr[1\leftarrow\mathcal{A}(Y,r,U_1||\ldots||U_i||\mathsf{GL}(r,i)||\ldots||\mathsf{GL}(r,k-1))]>\epsilon/2.\end{aligned}$$

over the choice of r and randomness of $\mathcal{A}$. Conditioned on such Y is fixed, we have

$$\begin{aligned}&\Pr\left[\mathsf{GL}(r,i-1)\leftarrow\mathcal{P}(Y',r)\right]\\ &=\Pr[\mathsf{GL}(r,i-1)\leftarrow\mathcal{P}(Y',r)|b=\mathsf{GL}(r,i-1)]\Pr[b=\mathsf{GL}(r,i-1)]\\ &\quad+\Pr[\mathsf{GL}(r,i-1)\leftarrow\mathcal{P}(Y',r)|b\neq\mathsf{GL}(r,i-1)]\Pr[b\neq\mathsf{GL}(r,i-1)]\\ &=1/2+1/2\cdot(\Pr[1\leftarrow\mathcal{A}(Y,r,U_1||\ldots||U_{i-1}||\mathsf{GL}(r,i-1)||\ldots||\mathsf{GL}(r,k-1)]\\ &\quad-\Pr[1\leftarrow\mathcal{A}(Y,r,U_1||\ldots||U_{i-1}||1-\mathsf{GL}(r,i-1)||\mathsf{GL}(r,i)||\ldots||\mathsf{GL}(r,k-1))])\\ &>1/2+\epsilon/4\end{aligned}$$

Reconstruction Algorithm. We obtained an algorithm $\mathcal{P}$ that distinguishes $\mathsf{GL}(r,i-1)=\mathsf{GL}_r(g^{2^{k(n-1)+i-1}\Pi_{j=1}^{n+1}x_j})$ from a random bit with the advantage larger than ϵ when it is given Y',r for at least $\epsilon/2$ fraction of Y'. Here, we use the Goldreich-Levin theorem.

Theorem 4. *(Goldreich-Levin Theorem [GL89]) Let x be an n-bit string. If there exists a PPT algorithm $\mathcal{P}$ such that*

$$|\Pr[\mathsf{GL}_r(x)\leftarrow\mathcal{P}(r,z)]-1/2|$$

is non-negligible where $r \xleftarrow{\$} \{0,1\}^n$, then there exists a PPT algorithm $\mathcal{R}$ such that

$$\Pr[x\leftarrow\mathcal{R}(z)]$$

is non-negligible.

By using this theorem, we obtain an algorithm $\mathcal{R}$ that computes $g^{2^{k(n-1)+i-1}\Pi_{j=1}^{n+1}x_j}$ when it is given $(N, g, g^{x_1}, \dots, g^{x_{n+1}}, \tau_{x_1} \dots, \tau_{x_{n+1}}, g^{2^{k(n-1)+i}\Pi_{j=1}^{n+1}x_j})$ with non-negligible probability for non-negligible fraction of its input.

Factoring Algorithm Then we construct an algorithm $\mathcal{B}$ that factorizes an RSA modulus N. The construction of $\mathcal{B}$ is as follows.

$\mathcal{B}(N)$: $\mathcal{B}$ chooses $h' \overset{\$}{\leftarrow} \mathbb{Z}_N^* \setminus \mathbb{QR}_N^+$ sets $h := |h'^2 \mod N| \in \mathbb{QR}_N^+$, $g := h^{2^k}$, chooses $x'_1, \dots, x'_{n+1} \overset{\$}{\leftarrow} [(N-1)/4]$, sets $g^{x_1} := g^{x'_1}h^{2^{k-i}}$, $\tau_{x_1} \leftarrow i\mathcal{O}(M'_1, C_{N,2^k x'_1+2^{k-i}})$, $g^{x_i} := g^{x'_i}h$, $\tau_{x_i} \leftarrow i\mathcal{O}(M'_1, C_{N,2^k x'_i+1})$ for $i = 2, \cdots, n+1$. Then $\mathcal{B}$ can compute $g^{2^{k(n-1)+i}\Pi_{j=1}^{n+1}x_j} = h^{2^{kn+i}\Pi_{j=1}^{n+1}x_j} = h^{(2^i x'_1+1)\Pi_{j=2}^{n+1}(2^k x'_j+1)}$. $\mathcal{B}$ runs $\mathcal{R}(N, g, g^{x_1}, \dots, g^{x_{n+1}}, \tau_{x_1}, \dots, \tau_{x_{n+1}}, g^{2^{k(n-1)+i}\Pi_{j=1}^{n+1}x_j})))$. Let U be the output of $\mathcal{R}$. Then $\mathcal{B}$ computes $X := \Pi_{j=2}^{n+1}(2^k x'_j + 1)$ and computes $V = Uh^{-(2^{i-1}x'_1 X+(X-1)/2)}$. (Note that X is odd and therefore $(X-1)/2$ is an integer.) Then it outputs $\gcd(h', V)$.

First, we consider the distribution of input for $\mathcal{R}$. Clearly, all components except g^{x_1} and τ_{x_1} are distributed as in Game 2. In the above algorithm, g^{x_1} is distributed almost uniformly on $\mathbb{QR}_N^+$ as in Game 2 and therefore this difference causes a negligible difference on the behavior of $\mathcal{R}$. τ_{x_1} is set as an obfuscation of a circuit that computes $2^k x_1$-th power both in the above algorithm and in Game 2, and this causes a negligible difference by the property of indistinguishability obfuscation. Therefore $\mathcal{R}$ outputs $g^{2^{k(n-1)+i-1}\Pi_{j=1}^{n+1}x_j}$ with non-negligible probability for non-negligible fraction of its input. In this case, we have

$$\begin{aligned} U = g^{2^{k(n-1)+i-1}\Pi_{j=1}^{n}x_j} &= h^{2^{kn+i-1}(x'_1+1/2^i)\Pi_{j=2}^{n+1}(x'_j+1/2^k)} \\ &= h^{(2^{i-1}x'_1+1/2)\Pi_{j=2}^{n+1}(2^k x'_j+1)} = h^{2^{i-1}x'_1 X+(X-1)/2+1/2}. \end{aligned}$$

Therefore we have $V = h^{1/2}$. Thus h' and V are distinct square roots of h in $\mathbb{Z}_N^*$ and therefore $\gcd(h', V)$ is a non-trivial factor of N. □

Theorem 3 is proven by the above lemmas. □

Next, we show a sufficient condition such that the AI-GMDH assumption holds. Before stating our result, we prepare a notation.

Definition 10. *For a monic monomial f defined by $f(x_1, \dots, x_n) = \prod_{i=1}^n x_i^{t_i}$, we define its corresponding polynomial $\bar{f}$ by $\bar{f}(x_1, \dots, x_n) := \sum_{i=1}^n t_i x_i$.*

Then our result is the following.

Theorem 5. *Let $f_1, \dots, f_m$ and f^* be functions of the form as in Def. 10. If there exists $(a_1, \dots, a_n) \in \mathbb{Z}^n$ such that $\bar{f}_i(a_1, \dots, a_n) \geq -1$ for all $i \in [m]$ and $\ell^* + \bar{f}^*(a_1, \dots, a_n) \leq -1$ hold. Then if $i\mathcal{O}$ is an indistinguishability obfuscation and the factoring assumption holds w.r.t.* RSAGen*, then $(\{f_i\}_{i \in [m]}, f^*, \ell^*, 2^{k-1})$-AI-GMDH assumption holds w.r.t. the above AI-SBM.*

Proof. Assume that there exists a PPT adversary $\mathcal{A}$ that breaks the $(\{f_i\}_{i \in [m]}, f^*, \ell^*, 2^{k-1})$-AI-GMDH assumption. We construct a PPT algorithm $\mathcal{B}$ that computes the square root of a random element of $\mathbb{QR}_N^+$ with non-negligible probability. (such an algorithm yields a PPT algorithm that breaks the factoring assumption.) The description of $\mathcal{B}$ is as follows.

$\mathcal{B}(N, h)$: Let $g := h^{2^k}$ and $\mathsf{params} := (N, g)$. Pick $x'_i \xleftarrow{\$} [(N-1)/4]$ and implicitly define $x_i := 2^{ka_i}(2x'_i + 1) \mod \mathrm{ord}(\mathbb{QR}_N^+)$. (Since $\mathcal{B}$ do not know $\mathrm{ord}(\mathbb{QR}_N^+)$, it cannot compute x_i. It defines as above only in mind.) Then for all $i \in [m]$, we have

$$f_i(x_1, \dots, x_n) \equiv 2^{k\bar{f}_i(a_1, \dots, a_n)}\mathrm{odd_i} \mod \mathrm{ord}(\mathbb{QR}_\mathrm{N}^+)$$

$$f^*(x_1, \dots, x_n) \equiv 2^{k\bar{f}^*(a_1, \dots, a_n)}\mathrm{odd}^* \mod \mathrm{ord}(\mathbb{QR}_\mathrm{N}^+)$$

where $\mathrm{odd_i}$ and odd^* are odd numbers efficiently computable from $\{x'_i\}_{i \in [n]}$. Here, we let $A_i := 2^{k(\bar{f}_i(a_1, \dots, a_n)+1)}\mathrm{odd_i}$, $F_i := h^{A_i}$, and $\tau_{f_i} := i\mathcal{O}(S, \tilde{C}_{N, A_i})$ where $\tilde{C}_{N, A_i}$ is the canonical circuit that computes A_i-th power on $\mathbb{QR}_N^+$. Then $\mathcal{B}$ runs $(c^*, T) \leftarrow \mathcal{A}(\mathsf{params}, \{F_i\}_{i \in [m]}, \{\tau_{f_i}\}_{i \in [m]})$. We can express c^* as $c^* = 2^v \mathrm{odd_{c^*}}$ where $\mathrm{odd_{c^*}}$ is the odd part of c^*. Then we have $v \leq k - 1$ since we have $|c^*| \leq M \leq 2^{k-1}$. If $\mathcal{A}$ succeeds, then we have

$$\begin{aligned} T &= e_{\ell^*}(g, \dots, g)^{c^* f^*(x_1, \dots, x_n)} \\ &= g^{2^{k(\ell^*-1)} c^* f^*(x_1, \dots, x_n)} \\ &= h^{2^{k\ell^*} c^* 2^{k\bar{f}^*(a_1, \dots, a_n)} \mathrm{odd}^*} \\ &= h^{2^{k(\ell^* + \bar{f}^*(a_1, \dots, a_n)) + v} \mathrm{odd}'} \end{aligned}$$

where we define $\mathrm{odd}' := \mathrm{odd}^* \cdot \mathrm{odd_{c^*}}$. Here, since we have $\ell^* + \bar{f}^*(a_1, \dots, a_n) \leq -1$ by the assumption and $v \leq k-1$, we have $k(\ell^* + \bar{f}^*(a_1, \dots, a_n)) + v \leq -1$. . Then if we define a natural number α by $\alpha := -(k(\ell^* + \bar{f}^*(a_1, \dots, a_n)) + v)$, then we have $T = h^{2^{-\alpha}\mathrm{odd}'}$. Therefore we have $T^{\alpha - 1} = h^{2^{-1}\mathrm{odd}'}$. Then if we let odd$'$:= 2even$'$ + 1 , then we have $T^{\alpha-1} = h^{\text{even}'+1/2}$. Therefore $\mathcal{B}$ can compute $h^{1/2}$ by computing $T^{\alpha-1}h^{-\text{even}'}$.

This completes the description of $\mathcal{B}$. In the above description, we already show that if $\mathcal{A}$ succeeds, then $\mathcal{B}$ also succeeds. What is left is to prove the distribution of $\mathcal{A}$'s input in the above algorithm is computationally indistinguishable from that in the AI-GMDH assumption. N is generated in the same way as in the AI-GMDH assumption (it is generated as $N \leftarrow \mathsf{RSAGen}(1^\lambda)$). g is uniformly distributed on $\mathbb{QR}_N^+$ as in the AI-GMDH assumption since h is uniformly distributed on $\mathbb{QR}_N^+$ and 2^k is coprime to $\mathrm{ord}(\mathbb{QR}_N^+)$. Since $\{x_i'\}_{i\in[n]}$ are almost uniformly distributed on $[\mathrm{ord}(\mathbb{QR}_N^+)]$ and 2 is coprime to $\mathrm{ord}(\mathbb{QR}_N^+)$, $\{x_i\}_{i\in[n]}$ are also almost uniformly distributed on $[\mathrm{ord}(\mathbb{QR}_N^+)]$. Since we have $F_i = h^{A_i}$ and $A_i \equiv 2^k f_i(x_1,\ldots,x_n) \mod \mathrm{ord}(\mathbb{QR}_N^+)$, we have $h^{A_i} = h^{2^k f_i(x_1,\ldots,x_n)} = g^{f_i(x_1,\ldots,x_n)}$ for $i \in [m]$. Thus we can see that $g^{f_i(x_1,\ldots,x_n)}$ is simulated correctly. What is left is to prove that the distribution of τ_{f_i} $(i \in [m])$ simulated by $\mathcal{B}$ is computationally indistinguishable from the real distribution in the AI-GMDH assumption conditioned on any fixed params, $\{F_i\}_{i\in[m]}$. τ_{f_i} is generated as $\tau_{f_i} := i\mathcal{O}(S, \tilde{C}_{N,A_i})$ in the simulation by $\mathcal{B}$, and $\tau_{f_i} := i\mathcal{O}(S, \tilde{C}_{N,2^k f_i(x_1,\ldots,x_n)})$ in the AI-GMDH assumption. Here, since we have $A_i \equiv 2^k f_i(x_1,\ldots,x_n) \mod \mathrm{ord}(\mathbb{QR}_N^+)$, $\tilde{C}_{N,2^k f_i(x_1,\ldots,x_n)}$ and $\tilde{C}_{N,A_i}$ have the completely the same functionality. Therefore if S is larger than the sizes od these circuits, then the above two are computationally indistinguishable by the property of the indistinguishability obfuscation. □

3.5 Applications of AI-SBM

In Sec. 3.4, we constructed an AI-SBM. In this section, we construct multiparty NIKE, distributed broadcast encryption, ABE for circuits and homomorphic signatures schemes by using an AI-SBM.

3.5.1 Multiparty NIKE.

Here, we construct a multiparty NIKE scheme. The idea of our construction is to use our multilinear map (with auxiliary information) for the“multiparty” Diffie-Hellman key exchange [DH76, BS02].

Construction. Our construction of multiparty NIKE scheme is as follows. Let $\mathcal{H}$ be a family of hash functions.

$\mathsf{Setup}_{\mathsf{NIKE}}(1^\lambda)$: $\mathsf{Setup}_{\mathsf{NIKE}}$ runs $\mathsf{params} = (G, e, g) \leftarrow \mathsf{InstGen}(1^\lambda)$ and chooses $H \overset{\$}{\leftarrow} \mathcal{H}$. It outputs $\mathsf{PP} = (\mathsf{params}, H)$ as the public parameter.

$\mathsf{Publish}_{\mathsf{NIKE}}(\mathsf{PP})$: It chooses $x \leftarrow [\mathrm{Approx}(G)]$ and sets $\tau_x \leftarrow \mathsf{AIRand}(S, \mathsf{AIGen}(x))$. It sets $pk := (g^x, \tau_x)$ and $sk := x$, and outputs (pk, sk).

$\mathsf{KeyGen}_{\mathsf{NIKE}}(\mathsf{PP}, sk, \{pk_j\}_{j=1,\ldots,n-1})$: Let $pk_j = (g^{x_j}, \tau_{g^{x_j}})$ for $j \in [n-1]$. It first sets $k_1 := g^{x_1}$. For $j = 2, \ldots, n-1$, it computes $k_j = \mathsf{Map}(\mathsf{params}, k_{j-1}, \tau_{x_j})$. Finally, it computes $k_n = k_{n-1}^{sk}$ and $K = h(k_n)$, and output K. Finally, it outputs $K := H(k_n)$ as its derived key.

We show the correctness. Let $\mathsf{PP} \leftarrow \mathsf{Setup}_{\mathsf{NIKE}}(1^\lambda)$ and $(g^{x_j}, \tau_{g^{x_j}}) \leftarrow \mathsf{Publish}_{\mathsf{NIKE}}(PP)$ for $j \in [n]$. Then for all $j^* \in [n]$, it is easy to show that $K = h(e_{n-1}(g, \ldots, g)^{\Pi_{j=1}^n x_j})$. Therefore the correctness holds.

The security of our NIKE scheme can be stated as follows.

Theorem 6. *This multiparty NIKE scheme is statically secure if the AI-MHDH assumption holds with respect to the underlying AI-SBM and $\mathcal{H}$.*

This is immediate from the definition of the security of NIKE and the AI-MHDH assumption.

3.5.2 Distributed broadcast encryption.

Here, we give a construction of distributed broadcast encryption scheme based on a multiparty NIKE scheme. This is based on the conversion proposed in [BZ14]. Let ($\mathsf{Setup}_{\mathsf{NIKE}}$,$\mathsf{Publish}_{\mathsf{NIKE}}$,$\mathsf{KeyGen}_{\mathsf{NIKE}}$) be a multiparty NIKE scheme. Then we construct a distributed broadcast encryption scheme ($\mathsf{Setup}_{\mathsf{BE}}$,$\mathsf{Join}_{\mathsf{BE}}$,$\mathsf{Enc}_{\mathsf{BE}}$, $\mathsf{Dec}_{\mathsf{BE}}$) as follows.

$\mathsf{Setup}_{\mathsf{BE}}(1^\lambda)$: It runs $\mathsf{Setup}_{\mathsf{NIKE}}(1^\lambda)$ to obtain public parameters PP and outputs PP as its own public parameters.

$\mathsf{Join}_{\mathsf{BE}}(PP)$: It runs $\mathsf{Publish}_{\mathsf{NIKE}}(PP)$ to obtain a public key pk and a secret key sk, and outputs (pk, sk).

$\mathsf{Enc}(PP, pk_1, \ldots, pk_n, msg)$: It runs $\mathsf{Publish}_{\mathsf{NIKE}}(PP)$ to obtain (pk^*, sk^*). Then it runs $\mathsf{KeyGen}_{\mathsf{NIKE}}(\mathsf{PP}, sk, \{pk_j\}_{j=1,\ldots,n})$ to obtain K. It computes $\Psi := K \oplus M$ and outputs a ciphertext $CT = (pk^*, \Psi)$.

$\mathsf{Dec}(PP, sk, pk_1, \ldots, pk_n, CT)$: It parses CT as pk^*, Ψ. It finds i such that sk is a corresponding secret key of pk_i. Then it runs $K \leftarrow \mathsf{KeyGen}_{\mathsf{NIKE}}(\mathsf{PP}, sk, pk^*, pk_1, \ldots, pk_{i-1}, pk_{i+1}, \ldots, pk_n)$ and outputs $msg := K \oplus \Psi$.

The security of the above scheme is immediate from the security of the underlying multiparty NIKE scheme.

Theorem 7. *If the multiparty NIKE scheme* $(\mathsf{Setup}_{\mathsf{NIKE}}$,$\mathsf{Publish}_{\mathsf{NIKE}}$ *is statically secure, then the distributed broadcast encryption scheme* $(\mathsf{Setup}_{\mathsf{BE}}, \mathsf{Join}_{\mathsf{BE}}, \mathsf{Enc}_{\mathsf{BE}}, \mathsf{Dec}_{\mathsf{BE}})$ *is statically secure.*

3.5.3 Attribute Based Encryption for Circuits

Here, we construct an attribute based encryption scheme for general circuits. Our construction is an analogue of [GGH$^+$13c].

Construction. Then we give our construction of an ABE scheme. Let $\mathcal{H}$ be a family of hash functions $H : G \to \{0,1\}^{\ell_H}$ and S and S_1 be sufficiently large integers.

$\mathsf{Setup}(1^\lambda, n, d)$: It runs $\mathsf{params} = (G, e, g) \leftarrow \mathsf{InstGen}(1^\lambda)$ and chooses $\alpha \leftarrow [\text{Approx}(G)]$, $h_1, \ldots, h_n \overset{\$}{\leftarrow} G$ and $H \overset{\$}{\leftarrow} \mathcal{H}$. Then it outputs $PP := (\mathsf{params}, H, e_{d+1}(g, \ldots, g)^\alpha, h_1, \ldots, h_n)$ and $MSK := (\alpha, PP)$.

$\mathsf{Enc}(\mathsf{PP}, x \in \{0,1\}^n, M \in \{0,1\}^{\ell_H})$: It chooses $s \leftarrow [\text{Approx}(G)]$, sets $\tau_s \leftarrow \mathsf{AIRand}(S, \mathsf{AIGen}(s))$ and

$$CT := (M \oplus H((e_{d+1}(g, \ldots, g)^\alpha)^s), \tau_s, \{h_i^s\}_{i \in X})$$

where X is the set of all $i \in [n]$ such that $x_i = 1$. It outputs CT.

$\mathsf{KeyGen}(MSK, f)$: It chooses $r_1, \ldots, r_v \overset{\$}{\leftarrow} [\text{Approx}(G)]$ and sets

$$K_H := e_d(g, \ldots, g)^{\alpha - r_v}$$

where v is the number of wires of f. Next, it generates key component for each wire of f as follows.

- *Input Wire*: If w is an input wire (i.e., depth is 1), then it chooses $z_w \overset{\$}{\leftarrow} [\text{Approx}(G)]$ and sets

$$K_w := g^{r_w} h_w^{-z_w}$$
$$\tau_{z_w} \leftarrow \mathsf{AIRand}(S_1, \mathsf{AIGen}(z_w)).$$

  We let (K_w, τ_{z_w}) be the key component for wire w.

- *OR Gate*: If w is an output wire of an OR gate with depth j, then it chooses $a_w, b_w \xleftarrow{\$} [\text{Approx}(G)]$ and sets

$$\begin{aligned} K_{w,1} &:= e_j(g, \ldots, g)^{r_w - a_w r_{A(w)}} \\ K_{w,2} &:= e_j(g, \ldots, g)^{r_w - b_w r_{B(w)}} \\ \tau_{a_w} &\leftarrow \mathsf{AIRand}(S_1, \mathsf{AIGen}(a_w)) \\ \tau_{b_w} &\leftarrow \mathsf{AIRand}(S_1, \mathsf{AIGen}(b_w)). \end{aligned}$$

  We let $(K_{w,1}, K_{w,2}, \tau_{a_w}, \tau_{b_w})$ be the key component for the wire w.
- *AND Gate*: If w is an output wire of an AND gate with depth j, then it chooses $a_w, b_w \xleftarrow{\$} [\text{Approx}(G)]$ and sets

$$\begin{aligned} K_w &:= e_j(g, \ldots, g)^{r_w - a_w r_{A(w)} - b_w r_{B(w)}} \\ \tau_{a_w} &\leftarrow \mathsf{AIRand}(S_1, \mathsf{AIGen}(a_w)) \\ \tau_{b_w} &\leftarrow \mathsf{AIRand}(S_1, \mathsf{AIGen}(b_w)). \end{aligned}$$

  We let $(K_w, \tau_{a_w}, \tau_{b_w})$ be the key component for wire w.

It outputs SK which consists of description of f, K_H and key components for each wire.

$\mathsf{Dec}(PP, SK, CT)$: Let SK be a secret key corresponding to f and CT be a ciphertext corresponding to x. We can correctly decrypt it if $f(x) = 1$. First, it computes $E' := e(g^s, K_H) = e_{d+1}(g, \ldots, g)^{s(\alpha - r_v)}$ by using τ_s. Then it computes as follows for all wires from wires with lower depth.

- *Input Wire*: Let w be an input wire. If x_w=1 holds, then it computes

$$\begin{aligned} E_w &:= e(g^s, K_w) e(g^{z_w}, h_w^s) \\ &= e(g^s, g^{r_w} h_w^{-z_w}) e(g^{z_w}, h_w^s) \\ &= e(g, g)^{s r_w} \end{aligned}$$

  by using τ_s and τ_{z_w}.
- *OR gate*: Let w be an output wire of an OR gate with depth j. If $f_w(x) = 1$ holds, then it works as follows. In this case we have $f_{A(w)} = 1$ or $f_{B(w)} = 1$. If we have $f_{A(w)} = 1$, then it computes

$$\begin{aligned} E_w &:= e(g^{a_w}, E_{A(w)}) e(g^s, K_{w,1}) \\ &= e(g^{a_w}, e_j(g, \ldots, g)^{s r_{A(w)}}) e(g^s, e_j(g, \ldots, g)^{r_w - a_w r_{A(w)}}) \\ &= e_{j+1}(g, \ldots, g)^{s r_w} \end{aligned}$$

by using τ_{a_w} and τ_s. If we have $f_{A(w)} \neq 1$, (in this case, we have $f_{B(w)} = 1$,) then it computes

$$\begin{aligned} E_w &:= e(g^{b_w}, E_{B(w)})e(g^s, K_{w,2}) \\ &= e(g^{b_w}, e_j(g,\ldots,g)^{sr_{B(w)}})e(g^s, e_j(g,\ldots,g)^{r_w - b_w r_{B(w)}}) \\ &= e_{j+1}(g,\ldots,g)^{sr_w} \end{aligned}$$

by using τ_{b_w} and τ_s.

- *AND gate*: Let w be an output gate of an AND gate with depth j. If $f_w(x) = 1$ holds, i.e, we have $f_{A(w)} = 1$ and $f_{B(w)} = 1$, then it works as follows. It computes

$$\begin{aligned} E_w &:= e(E_{A(w)}, g^{a_w})e(E_{B(w)}, g^{b_w})e(g^s, K_w) \\ &= e(e_j(g,\ldots,g)^{sr_{A(w)}}, g^{a_w})e(e_j(g,\ldots,g)^{sr_{B(w)}}, g^{b_w}) \\ &\quad\ e(g^s, e_j(g,\ldots,g)^{r_w - a_w r_{A(w)} - b_w r_{B(w)}}) \\ &= e_{j+1}(g,\ldots,g)^{sr_w} \end{aligned}$$

by using τ_{a_w}, τ_{b_w} and τ_s.

If $f(x) = 1$ holds, then for output wire v with depth d, it can compute

$$E_v = e_{d+1}(g,\ldots,g)^{sr_v}.$$

Next, it computes

$$\begin{aligned} E'' &:= E' E_v \\ &= e_{d+1}(g,\ldots,g)^{s(\alpha - r_v)} e_{d+1}(g,\ldots,g)^{sr_v} \\ &= e_{d+1}(g,\ldots,g)^{s\alpha}. \end{aligned}$$

Finally, it outputs $M := C_M \oplus H(E'')$.

The correctness of the scheme is already checked in the above description. The security of the scheme is as follows.

Theorem 8. *Our ABE scheme is selectively secure if the AI-MHDH assumption holds with respect to the underlying multilinear map and $\mathcal{H}$.*

Proof. We construct an algorithm $\mathcal{D}$ that breaks the $(d+1)$-AI-MHDH assumption by using $\mathcal{A}$ that breaks the ABE scheme. The construction of $\mathcal{D}$ is as follows. We note that the label of an instance of the AI-MHDH problem is different from that in the definition for notational convenience.

$\mathcal{D}(\mathsf{params}, g^s, g^{c_1}, \ldots, g^{c_{d+1}}, \tau_s, \tau_{c_1}, \ldots, \tau_{c_{d+1}}, H, T)$:

Setup. Let $x^* = (x_1^*, \ldots, x_n^*)$ be a target input declared by $\mathcal{A}$. $\mathcal{D}$ sets $\alpha := \Pi_{i=1}^{d+1} c_i$. Then it can computes $e_{d+1}(g, \ldots, g)^\alpha$ by iterated usage of Map by using $\tau_{c_1}, \ldots, \tau_{c_{d+1}}$. It chooses $y_i \xleftarrow{\$} [\mathrm{Approx}(G)]$ for $i = 1, \ldots, n$, and sets

$$h_i := \begin{cases} g^{y_i} & \text{if } x_i^* = 1 \\ g^{y_i + c_1} & \text{if } x_i^* = 0 \end{cases}$$

and $PP := (\mathsf{params}, H, e_{d+1}(g, \ldots, g)^\alpha, h_1, \ldots, h_n)$. Then it gives PP to $\mathcal{A}$.

Challenge Ciphertext. For messages M_0, M_1 which are declared by $\mathcal{A}$, $\mathcal{D}$ chooses $b \xleftarrow{\$} \{0, 1\}$, and sets $CT := (M_b \oplus T, \tau_s, \{(g^s)^{y_i}\}_{i \in X^*})$ where X^* is the set of all $i \in [n]$ such that $x_i^* = 1$. It gives CT to $\mathcal{A}$ as a challenge ciphertext.

Key Generation. For $\mathcal{A}$'s key query f such that $f(x^*) = 0$, $\mathcal{D}$ computes as follows for all wires from wires with lower depth.

- *Input Wire* : Let w be an input wire. If $x_w^* = 1$, then $\mathcal{D}$ chooses $z_w, r_w \xleftarrow{\$} [\mathrm{Approx}(G)]$, and computes

$$\begin{aligned} K_w &:= g^{r_w} h_w^{-z_w} \\ \tau_{z_w} &\leftarrow \mathsf{AIRand}(S_1, \mathsf{AIGen}(z_w)). \end{aligned}$$

  If $x_w^* = 0$, then it chooses $\eta_w, \nu_w \xleftarrow{\$} [\mathrm{Approx}(G)]$, generates $\tau_{\nu_w} \leftarrow \mathsf{AIGen}(\nu_w)$ sets $z_w := c_2 + \nu_w$ and $r_w := c_1 c_2 + \eta_w$, and computes

$$\begin{aligned} K_w &:= g^{-c_2 y_w + \eta_w - (y_w + c_1)\nu_w} \\ \tau_{z_w} &:= \mathsf{AIRand}(S_1, \mathsf{AIMult}(\tau_{c_2}, \tau_{\nu_w})). \end{aligned}$$

- *OR Gate*: Let w be an output wire of an OR gate with depth j. If $f_w(x^*) = 1$, then $\mathcal{D}$ chooses $a_w, b_w \xleftarrow{\$} [\mathrm{Approx}(G)]$ and computes

$$\begin{aligned} K_{w,1} &:= e_j(g, \ldots, g)^{r_w - a_w r_{A(w)}} \\ K_{w,2} &:= e_j(g, \ldots, g)^{r_w - b_w r_{B(w)}} \\ \tau_{a_w} &:= \mathsf{AIRand}(S', \mathsf{AIGen}(a_w)) \\ \tau_{b_w} &:= \mathsf{AIRand}(S', \mathsf{AIGen}(b_w)). \end{aligned}$$

  If $f_w(x^*) = 0$, then it chooses $\psi_w, \phi_w, \eta_w \xleftarrow{\$} [\mathrm{Approx}(G)]$, generates $\tau_{\psi_w} \leftarrow$

$\mathsf{AIGen}(\psi_w)$ and $\tau_{\phi_w} \leftarrow \mathsf{AIGen}(\phi_w)$, sets $a_w := c_{j+1} + \psi_w$, $b_w := c_{j+1} + \phi_w$ and $r_w := \Pi_{i=1}^{j+1} c_i + \eta_w$, and computes

$$\begin{aligned}
K_{w,1} &:= e_j(g, \ldots, g)^{\eta_w - \psi_w \eta_{A(w)} - c_{j+1}\eta_{A(w)} - \psi_w \Pi_{i=1}^{j} c_i} \\
K_{w,2} &:= e_j(g, \ldots, g)^{\eta_w - \phi_w \eta_{B(w)} - c_{j+1}\eta_{B(w)} - \phi_w \Pi_{i=1}^{j} c_i} \\
\tau_{a_w} &:= \mathsf{AIRand}(S', \mathsf{AIMult}(\tau_{c_{j+1}}, \tau_{\psi_w})) \\
\tau_{b_w} &:= \mathsf{AIRand}(S', \mathsf{AIMult}(\tau_{c_{j+1}}, \tau\phi_w)).
\end{aligned}$$

- *AND Gate*: Let w be an output wire of an AND gate with depth j. If $f_w(x^*) = 1$, then $\mathcal{D}$ chooses $a_w, b_w \xleftarrow{\$} [\text{Approx}(G)]$, computes

  $$\begin{aligned}
  K_w &:= e_j(g, \ldots, g)^{r_w - a_w r_{A(w)} - b_w r_{B(w)}} \\
  \tau_{a_w} &\leftarrow \mathsf{AIRand}(S', \mathsf{AIGen}(a_w)) \\
  \tau_{b_w} &\leftarrow \mathsf{AIRand}(S', \mathsf{AIGen}(b_w)).
  \end{aligned}$$

  If $f_w(x^*) = 0$, then it works as follows. If $f_{A(w)}(x^*) = 0$, then it chooses $\psi_w, \phi_w, \eta_w \xleftarrow{\$} [\text{Approx}(G)]$, generates $\tau_{\psi_w} \leftarrow \mathsf{AIGen}(\psi_w)$ and $\tau_{\phi_w} \leftarrow \mathsf{AIGen}(\phi_w)$, sets $a_w := c_{j+1} + \psi_w$, $b_w := \phi_w$ and $r_w := \Pi_{i=1}^{j+1} c_i + \eta_w$, and computes

  $$\begin{aligned}
  K_w &:= e_j(g, \ldots, g)^{\eta_w - \psi_w \eta_{A(w)} - \phi_w r_{B(w)} - c_{j+1}\eta_{A(w)} - \psi_w \Pi_{i=1}^{j} c_i} \\
  \tau_{a_w} &:= \mathsf{AIRand}(S', \mathsf{AIMult}(\tau_{c_{j+1}}, \tau_{\psi_w})) \\
  \tau_{b_w} &:= \mathsf{AIRand}(S', \tau_{\phi_w}).
  \end{aligned}$$

  If $f_{A(w)}(x^*) = 1$ and $f_{B(w)}(x^*) = 0$, it works symmetric to what is above, with the roles of a_w and b_w reversed.

Remark 10. *$\mathcal{D}$ can actually simulate the key generation oracle as the above since $e_j(g, \ldots, g)^{\Pi_{i=1}^{j} c_i}$ can be computed by iterated usage of Map by using $\tau_{c_1}, \ldots, \tau_{c_j}$. Note that it need not compute $e_j(g, \ldots, g)^{\Pi_{i=1}^{j+1} c_i}$ thanks to the cancellation technique.*

Since we have $f(x^*) = 0$, for the output wire v, r_v is defined as $\Pi_{i=1}^{d+1} c_i + \eta_v$. Therefore it can generate

$$K_H = e_d(g, \ldots, g)^{\alpha - r_v} = e_d(g, \ldots, g)^{-\eta_v}.$$

Thus, it can simulate the key generation oracle.

Guess. Finally, when $\mathcal{A}$ outputs b', $\mathcal{D}$ outputs 1 if $b = b'$, and otherwise 0.

The above completes the description of $\mathcal{D}$. We can easily see that if $\beta = 1$, then the CPA game and the simulated environment are computationally indistinguishable from the view of $\mathcal{A}$ by the indistinguishablity of auxiliary information. (Recall that β is a random coin that determines T is random or not.) On the other hand, if $\beta = 0$, then information of b is completely hidden and therefore the probability that $\mathcal{A}$ predicts b is equal to 1/2. Therefore $|\Pr[1 \leftarrow \mathcal{D}|\beta = 1] - \Pr[1 \leftarrow \mathcal{D}|\beta = 0]|$ is non-negligible if $\mathcal{A}$ breaks the CPA security of the scheme. □

3.5.4 Homomorphic Signature

Here, we construct a selectively secure single data homomorphic signature scheme for the class of all polynomials. Our scheme is based on the idea of [CFW14]. Namely, our scheme is almost automatically obtained by replacing multilinear maps by AI-SBP in the scheme of [CFW14].

Notation. In the following, we abuse the notation so that τ_X denotes an auxiliary information corresponding to X, $\tau_X \cdot \tau_Y$ denotes $\mathsf{AIMult}(\tau_X, \tau_Y)$, $e(\tau_X, \tau_Y)$ denotes $\mathsf{AIMap}(\tau_X, \tau_Y)$ and τ_X^α denotes $\mathsf{AIExp}(\tau_X, \alpha)$.

Our construction is as follows. We let $[M]$ be the message space.

$\mathsf{KeyGen}(1^\lambda, 1^n) \rightarrow (vk, sk)$: Generate $(G, e, g) = \mathsf{params} \leftarrow \mathsf{ParamGen}(1^\lambda)$, choose $r_i \xleftarrow{\$} [\mathrm{Approx}(G)]$ $(i = 1, \ldots, n)$ and $x_1, x_2, x_3 \xleftarrow{\$} [\mathrm{Approx}(G)]$, and set

$$\begin{aligned}
&R_i := g^{r_i} \text{ (for } i = 1, \ldots, n), A := g^{x_2}, B := g^{x_3}, C := g^{x_1 x_2}, U := g^{x_1 x_2 x_3},\\
&\tau_{R_i} := \mathsf{AIRand}(S', \mathsf{AIGen}(r_i)) \text{ (for } i = 1, \ldots, n),\\
&\tau_A := \mathsf{AIRand}(S, \mathsf{AIGen}(x_2)), \tau_B := \mathsf{AIRand}(S, \mathsf{AIGen}(x_3)),\\
&\tau_C := \mathsf{AIRand}(S, \mathsf{AIGen}(x_1 x_2)), \tau_U := \mathsf{AIRand}(S, \mathsf{AIGen}(x_1 x_2 x_3)),
\end{aligned}$$

where S' can be set as an arbitrary integer larger than the maximal size of auxiliary information that is used as a second input of AIRand when generating R_i through the real scheme and the security proof. Then set

$$vk := (\mathsf{params}, \{R_i\}_{i \in [n]}, A, B, C, U, \{\tau_{R_i}\}_{i \in [n]}, \tau_A, \tau_B, \tau_C, \tau_U), sk := (x_1, x_2, vk)$$

and output $(vk.sk)$.

$\mathsf{Sign}(sk, i, m) \rightarrow \sigma$:

Compute

$$\Lambda := (R_i B^{-m})^{x_2}, \Gamma := \Lambda^{x_1},$$
$$\tau_\Lambda := \mathsf{AIRand}(S'', (\tau_{R_i}\tau_B^{-m})^{x_2}), \tau_\Gamma := \mathsf{AIRand}(S'', (\tau_{R_i}\tau_B^{-m})^{x_1 x_2})$$

and output $\sigma := (\Lambda, \Gamma, \tau_\Lambda, \tau_\Gamma)$, where S' can be set as an arbitrary integer larger than the maximal size of auxiliary information that is used as a second input of AIRand when generating τ_Γ through the real scheme and the security proof.

$\mathsf{Eval}(f, (m_1, \sigma_1), \ldots, (m_n, \sigma_n)) \to \sigma^*$: Let f be a polynomial of degree d. Then f can be seen as an arithmetic circuit of depth $O(\log(d))$. We let $\sigma_i = (\Lambda_i, \Gamma_i, \tau_{\Lambda_i}, \tau_{\Gamma_i})$. We label the i-th input wire of f by $(1, m_i, \sigma_i)$. For all $i \in [d]$, compute

$$U_i := e_i(U, \ldots, U), \tau_{U_i} := e_i(\tau_U, \ldots, \tau_U).$$

For each gate of f, compute the following.

Addition: Assume that the input wires for this gate is labeled by $(i, m^{(1)}, (\Lambda^{(1)}, \Gamma^{(1)}, \tau_{\Lambda^{(1)}}, \tau_{\Gamma^{(1)}}))$, $(j, m^{(2)}, (\Lambda^{(2)}, \Gamma^{(2)}, \tau_{\Lambda^{(2)}}, \tau_{\Gamma^{(2)}}))$. Without loss of generality, we assume that $i \geq j$. First, adjust the "degree" of each value. That is, set

$$\Lambda'^{(2)} := e_{i-j+1}(\Lambda^{(2)}, g, \ldots, g), \Gamma'^{(2)} := e_{i-j+1}(\Gamma^{(2)}, g, \ldots, g),$$
$$\tau_{\Lambda'^{(2)}} := e_{i-j+1}(\tau_{\Lambda^{(2)}}, \tau_g, \ldots, \tau_g), \tau_{\Gamma'^{(2)}} := e_{i-j+1}(\tau_{\Gamma^{(2)}}, \tau_g, \ldots, \tau_g).$$

Then set

$$m^* := m_1 + m_2, \Lambda^* := \Lambda^{(1)} \cdot \Lambda'^{(2)}, \Gamma^* := \Gamma^{(1)} \cdot \Gamma'^{(2)},$$
$$\tau_{\Lambda^*} := \tau_{\Lambda^{(1)}} \cdot \tau_{\Lambda'^{(2)}}, \tau_{\Gamma^*} := \tau_{\Gamma^{(1)}} \cdot \tau_{\Gamma'^{(2)}},$$

and assign $(i, m^*, (\Lambda^*, \Gamma^*, \tau_{\Lambda^*}, \tau_{\Gamma^*}))$ to the output wire of this gate.

Multiplication by constant c: Let $(i, m, (\Lambda, \Gamma, \tau_\Lambda, \tau_\Gamma))$ be the value labeled to the input wire of this gate. Then compute

$$m^* := c \cdot m, \quad \Lambda^* := \Lambda^c, \quad \Gamma^* := \Gamma^c, \quad \tau_\Lambda := \tau_\Lambda^c, \quad \tau_\Gamma := \tau_\Gamma^c$$

and assign $(i, m^*, (\Lambda^*, \Gamma^*, \tau_{\Lambda^*}, \tau_{\Gamma^*}))$ to the output wire of this gate.

Multiplication: Assume that the input wires for this gate is labeled by $(i, m^{(1)}, (\Lambda^{(1)}, \Gamma^{(1)}, \tau_{\Lambda^{(1)}}, \tau_{\Gamma^{(1)}}))$, $(j, m^{(2)}, (\Lambda^{(2)}, \Gamma^{(2)}, \tau_{\Lambda^{(2)}}, \tau_{\Gamma^{(2)}}))$. Then compute

$$\begin{aligned}
m^* &:= m_A \cdot m_B,\\
\Lambda^* &:= e(\Lambda^{(1)}, \Gamma^{(2)}) \cdot e(\Lambda^{(1)}, U_j^{m_2}) \cdot e(U_i^{m_i}, \Lambda^{(2)}),\\
\Gamma^* &:= e(\Gamma^{(1)}, \Gamma^{(2)}) \cdot e(\Gamma^{(1)}, U_j^{m_2}) \cdot e(U_i^{m_i}, \Gamma^{(2)}),\\
\tau_{\Lambda^*} &:= e(\tau_{\Lambda^{(1)}}, \tau_{\Gamma^{(2)}}) \cdot e(\tau_{\Lambda^{(1)}}, \tau_{U_j}^{m_2}) \cdot e(\tau_{U_i}^{m_1}, \tau_{\Lambda^{(2)}}),\\
\tau_{\Gamma^*} &:= e(\tau_{\Gamma^{(1)}}, \tau_{\Gamma^{(2)}}) \cdot e(\tau_{\Gamma^{(1)}}, \tau_{U_j}^{m_2}) \cdot e(\tau_{U_i}^{m_1}, \tau_{\Gamma^{(2)}}),
\end{aligned}$$

and assign $(i + j, m^*, (\Lambda^*, \Gamma^*, \tau_{\Lambda^*}, \tau_{\Gamma^*}))$ to the output wire of this gate.

We can see that the output wire of f is labeled by $(d, f(m_1, \ldots, m_n), (\Lambda_{\mathsf{out}}, \Gamma_{\mathsf{out}}, \tau_{\Lambda_{\mathsf{out}}}, \tau_{\Gamma_{\mathsf{out}}}))$ for some $(\Lambda_{\mathsf{out}}, \Gamma_{\mathsf{out}}, \tau_{\Lambda_{\mathsf{out}}}, \tau_{\Gamma_{\mathsf{out}}})$. Then Eval outputs Λ_{out}.

$\mathsf{Verify}(vk, f, m, \sigma = \Lambda) \to 1/0$: Set $g_d := e_d(g, \ldots, g)$ and compute $R = g_d^{f(r_1,\ldots,r_N)}$ and its corresponding auxiliary information τ_R. This can be computed by evaluating f on the values $R_1, \ldots, R_n$. Namely, replace an addition in f by a multiplication in G and a multiplication by an evaluation of e. Let $B_d := e_d(B, \ldots, B)$ and verify $e(R \cdot B_d^{-m}, g_d^{x_1^{d-1} x_2^d}) = e(\Lambda_d, g_d)$. If this equation holds, then output 1, and otherwise output 0. Here, required values for the verification can be computed as

$$\begin{aligned}
\tau_{R \cdot B_d^{-m}} &:= \tau_R \cdot e(\tau_B, \ldots, \tau_B)^{-m},\\
g_d^{x_1^{d-1} x_2^d} &:= e_d(A, C, \ldots, C),\\
\tau_{g_d} &:= e_d(\tau_g, \ldots, \tau_g).
\end{aligned}$$

Security

Theorem 9. *If the (dM)-AI-APMDH assumption hold for all polynomially bounded d then the above scheme is selectively secure.*

Proof. We construct a PPT adversary $\mathcal{B}$ that breaks the (ℓ, M)-AI-APMDH assumption by using a PPT adversary $\mathcal{A}$ that breaks the above homomorphic signature scheme. Here, if f^* output by $\mathcal{A}$ as a part of forgery is d, thenwe let $\ell := d$. The construction of $\mathcal{B}$ is as follows.

$\mathcal{B}(\mathsf{params}, \{F_i\}_{i\in[4]}, \{\tau_{F_i}\}_{i\in F_4})$: First, runs $\mathcal{A}$ to obtain a signing query $(m_1, \ldots, m_n)$. Set $A := F_1$, $B := F_2$, $C := F_3$, $U := F_4$, $\tau_A := \tau_{F_1}$, $\tau_B := \tau_{F_2}$, $\tau_C := \tau_{F_3}$ and $\tau_U := \tau_{F_4}$. Choose $y_i \xleftarrow{\$} [\mathrm{Approx}[G]]$ for $i \in [n]$ and set $R_i := g^{y_i} B^{m_i}$, $\tau_{R_i} := \mathsf{AIRand}(S', \mathsf{AIGen}(y_i) \cdot \tau_B^{m_i})$ and $vk := \begin{pmatrix} \mathsf{params}, \{R_i\}_{i\in[N]}, A, B, C, U, \\ \{\tau_{R_i}\}_{i\in[N]}, \tau_A, \tau_B, \tau_C, \tau_U \end{pmatrix}$.

Set $\Lambda_i := A^{y_i}$, $\Gamma_i := C^{y_i}$, $\tau_{\Lambda_i} := \mathsf{AIRand}(S'', \tau_A^{y_i})$, $\tau_{\Gamma_i} := \mathsf{AIRand}(S'', \tau_C^{y_i})$ and $\sigma_i := (\Lambda_i, \Gamma_i, \tau_{\Lambda_i}, \tau_{\Gamma_i})$ for $i \in [n]$. Give $(vk, \{\sigma_i\}_{i \in [n]})$ to $\mathcal{A}$. Let (f^*, m^*, σ^*) be a forgery output by $\mathcal{A}$. Set $\Lambda^* := \sigma^*$ and compute $\Lambda'^* \leftarrow \mathsf{Eval}(vk, f^*, \{(m_i, \sigma_i)\}_{i \in [n]})$. Set $c^* := f^*(m_1, \ldots, m_n) - m^*$ and output $(c^*, \Lambda^* \cdot {\Lambda'^*}^{-1})$.

We show that the above algorithm works well. First, we show that the distribution of vk and σ_i for $i \in [n]$ given to $\mathcal{A}$ in the $\mathcal{B}$'s simulation is computationally indistinguishable from that in the game of the selective security. Each component of vk except R_i and τ_{R_i} is generated in exactly the same way as in the real scheme.It is easy to see that R_i is distributed almost uniformly on G both in $\mathcal{B}$'s simulation and the real scheme. τ_{R_i} is generated by inputting an element of T_{R_i} to AIRand in both the simulation and the real scheme, thus they are computationally indistinguishable due to the indistinguishability of auxiliary information. We can see that Λ_i and Γ_i are simulated correctly since we have $(R_i B^{-m_i})^{x_2} = (g^{y_i})^{x_2} = A^{y_i}$ and $(R_i B^{-m_i})^{x_1 x_2} = (g^{y_i})^{x_1 x_2} = C^{y_i}$. Moreover, the distribution of τ_{Λ_i} and τ_{Γ_i} generated by $\mathcal{B}$ is computationally indistinguishable from the real one since they are generated by inputting an element of T_{Λ_i} and T_{Γ_i} to AIRand, respectively. Therefore, the distribution of $\mathcal{A}$'s input simulated by $\mathcal{B}$ is computationally indistinguishable from the real one and thus $\mathcal{A}$ succeeds to output a forgery with non-negligible probability in $\mathcal{B}$'s simulation. If $\mathcal{A}$ succeeds to output a forgery, then we have $f^*(m_1, \ldots, m_n) \neq m^*$ and $e(R \cdot B_d^{-m^*}, g_d^{x_1^{d-1} x_2^d}) = e(\Lambda^*, g_d)$. Therefore we have $\Lambda^* = (R \cdot B_d^{-m^*})^{x_1^{d-1} x_2^d}$. On the other hand, by the correctness of the scheme, we have $e(R \cdot B_d^{-f^*(m_1, \ldots, m_n)}, g_d^{x_1^{d-1} x_2^d}) = e(\Lambda'^*, g_d)$. That is, we have $\Lambda'^* = (R \cdot B_d^{-f^*(m_1, \ldots, m_n)})^{x_1^{d-1} x_2^d}$. Therefore we have $\Lambda^* \cdot {\Lambda'^*}^{-1} = B_d^{f^*(m_1, \ldots, m_n) - m^*) x_1^{d-1} x_2^d} = g^{c^* x_1^{d-1} x_2^d x_3^d}$, where $f^*(m_1, \ldots, m_n) \neq m^*$ and thus we have $c^* \neq 0$ and $|c^*| = |(f^*(m_1, \ldots, m_n) - m^*| < M$. Therefore $\mathcal{B}$ succeeds to break the (d, M)-AI-GMDH assumption. $\square$

3.6 Homomorphic Encryption

In this section, we construct a somewhat homomorphic encryption scheme by using an indistinguishability obfuscator. This is not a direct application of our self-bilinear map. However, the idea behind the construction is similar.

3.6.1 Φ-hiding Assumption

Here, we give the definition of the Φ-hiding assumption [KOS10] as follows. Let $\mathsf{RSA}[p \equiv 1 \bmod e]$ be an efficient algorithm which takes the security parameter 1^λ as input and outputs (N, P, Q) where $N = PQ$ is an ℓ_N-bit Blum integer such that $P \equiv 1 \bmod e$ and $\mathbb{QR}_N^+$ is cyclic. Let $\mathcal{P}_\ell$ be the set of all ℓ-bit primes.

Definition 11. *For a constant c, we consider the following distributions.*

$$\mathcal{R} = \{(e, N) : e, e' \xleftarrow{R} \mathcal{P}_{c\ell_N}; N \leftarrow \mathsf{RSA}[p \equiv 1 \bmod e'](1^\lambda)\}$$

$$\mathcal{L} = \{(e, N) : e \xleftarrow{R} \mathcal{P}_{c\ell_N}; N \leftarrow \mathsf{RSA}[p \equiv 1 \bmod e](1^\lambda)\}$$

We say that the Φ-hiding assumption holds with respect to RSA *if for any efficient adversary $\mathcal{A}$, $|\Pr[1 \leftarrow \mathcal{A}(\mathcal{L})] - \Pr[1 \leftarrow \mathcal{A}(\mathcal{R})]|$ is negligible.*

Parameters. According to [KOS10], N can be factorized in time $O(N^\epsilon)$ where $e \xleftarrow{R} \mathcal{P}_{c\ell_N}; N \leftarrow \mathsf{RSA}[p \equiv 1 \bmod e](1^k)$ and $c = 1/4 - \epsilon$. In our scheme, we set c to be the value such that $c\ell_N = \lambda$. This setting avoids the above mentioned attack in a usual parameter setting (e.g., $\ell_N = 1024$ for 80-bit security).

3.6.2 Our Construction

Here, we construct a somewhat homomorphic encryption scheme by using indistinguishability obfuscation. We use the notation for circuits on $\mathbb{QR}_N^+$ which is given in Sec. 3.4. In addition to that, here, we use the following notation. For circuits C_1 and C_2 such that an output of C_1 can be interpreted as input for C_2, $C_1 \circ C_2$ denotes the composition of C_1 and C_2, i.e, $C_1 \circ C_2$ is a circuit that computes $C_2(C_1(x))$ for input x. The construction of our homomorphic encryption $\mathsf{HE}_{\mathsf{Ours}} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ is as follows.

$\mathsf{KeyGen}(1^\lambda)$: Choose $e \xleftarrow{\$} \mathcal{P}_\lambda$ and $(N, P, Q) \leftarrow \mathsf{RSA}[p \equiv 1 \bmod e](1^\lambda)$. Choose $g \xleftarrow{\$} \mathbb{QR}_N^+$ and compute an integer ρ such that $\rho \equiv 0 \bmod \mathrm{ord}(\mathbb{QR}_N^+)/e$ and $\rho \equiv 1 \bmod e$. It outputs a public key $pk = (N, e, g)$ and a secret key $sk = (\rho, pk)$.

$\mathsf{Enc}(pk, m \in \{0,1\})$: Choose $r \xleftarrow{\$} [(N-1)/4]$, set $c \leftarrow i\mathcal{O}(\mathsf{Max}, \tilde{C}_{N,m+re})$ and output c, where Max is defined as an integer larger than $\max_{m\in\{0,1\}, r\in[(N-1)/4]}\{|\tilde{C}_{N,m+re}|\}$.

$\mathsf{Eval}(pk, f, c_1, \dots, c_\ell)$: Work only if $c_1, \dots, c_\ell$ are circuits (i.e., generated by Enc). Convert f into an arithmetic circuit f' on $\mathbb{Z}_e$. (That is, each gate of f' is addition, multiplication or negation on $\mathbb{Z}_e$.)[*7] Compute as follows for all wires of f' from wires with lower depth.

- *Input*: Let w be the i-th input wire. Then c_i is assigned to this wire.
- *Addition*: Let w be an output wire of an addition gate. Set $c_w := \mathsf{Mult}(c_{A(w)}, c_{B(w)})$.
- *Multiplication*: Let w be an output wire of a multiplication gate. Set $c_w := c_{A(w)} \circ c_{B(w)}$.
- *Negation*: Let w be an output wire of a negation gate. Set $c_w := C_{N,inv} \circ c_{A_(w)}$ where $C_{N,inv}$ is a circuit that computes an inverse on $\mathbb{QR}_N^+$.

Let v be the output wire. Compute $c_{\mathsf{eval}} = c_v(g)$ and output it. Note that it is a group element and not a circuit. Therefore we cannot evaluate it again.

$\mathsf{Dec}(sk, c)$: Work differently depending on whether c is an output of Enc or Eval. If c is an output of Enc, then compute $M = c(g)$. If $M^\rho = 1$, then output 0, and otherwise output 1. If c is an output of Eval, then output 0 if $c^\rho = 1$, and otherwise output 1.

First, we prove the correctness of the scheme. We have $e|\mathrm{ord}(\mathbb{QR}_N^+)$ by the choice of N. Therefore, there exists a subgroup G_e^+ of order e of $\mathbb{QR}_N^+$. We can see that for any element $h \in \mathbb{QR}_N^+$, h^ρ is the G_e^+ component of h. In the decryption, we have $M = i\mathcal{O}(\mathsf{Max}, C_{N,m+re})(g) = g^{m+re}$. Therefore M^ρ is the G_e^+ component of g^m. We can see that G_e^+ component of g is not 1 with overwhelming probability since e is a λ-bit prime. Therefore $M^\rho = 1$ is equivalent to $m = 0$ and $M^\rho \neq 1$ is equivalent to $m = 1$ with overwhelming probability. Thus the correctness follows.

The security of $\mathsf{HE}_{\mathsf{Ours}}$ relies on the Φ-hiding assumption. Specifically, it satisfies the following property.

Theorem 10. $\mathsf{HE}_{\mathsf{Ours}}$ *is NC^1-homomorphic, compact and CPA secure if the Φ-hiding assumption holds with respect to* RSA *and $i\mathcal{O}$ is an indistinguishability obfuscator for P/poly.*

Proof.

NC^1-homomorphism. We show that $\mathsf{HE}_{\mathsf{Ours}}$ is NC^1-homomorphic. Here, NC^1 is the class of circuits with depth $O(\log(\lambda))$. Note that if f is in NC^1, then the depth of the corresponding arithmetic circuit f' is also $O(\log(\lambda))$. First, we show the correctness of Eval. We can see that output c_{eval} of Eval satisfies $c_{\mathsf{eval}} = g^m$ by an easy induction where m is a corresponding message. Therefore we can prove that c_{eval} is correctly

[*7] This can be done since we have $a \wedge b = a \cdot b \bmod e$ and $a \vee b = a + b - a \cdot b \bmod e$ if $a, b \in \{0, 1\}$.

decrypted similarly as the above. Next, we show that Eval is computed efficiently if the depth of f' is $O(\log(\lambda))$. We can easily see that Eval is computed efficiently if $|c_v|$ is polynomially bounded in the security parameter. Let M_j be the maximum value of $|c_w|$ for a wire w with depth j. M_1 is constant in $|f'|$. For $j \geq 2$, we have $M_j \leq 2M_{j-1} + \max\{|C_{N,mult}|, |C_{N,inv}|\}$. Therefore if the depth of f' is $O(\log \lambda)$, then $|c_v|$ is polynomially bounded by $|f'|$ (and therefore λ). Therefore Eval is efficiently computable.

Compactness. $\mathsf{HE}_{\mathsf{Ours}}$ is compact since output of Eval is always an element of $\mathbb{QR}_N^+$.

Security. To prove the security, we consider the following sequence of games.

Game 1: This game is the original CPA game. More formally, it is as follows.

$e \xleftarrow{\$} \mathcal{P}_\lambda$
$(N, P, Q) \leftarrow \mathsf{RSA}[p \equiv 1 \bmod e](1^\lambda)$
$g \xleftarrow{\$} \mathbb{Z}_N^*$
$b \xleftarrow{\$} \{0, 1\}$
$r \xleftarrow{\$} [(N-1)/4]$
$c \leftarrow i\mathcal{O}(\mathsf{Max}, \tilde{C}_{N,b+re})$
$b' \xleftarrow{\$} \mathcal{A}(N, e, g, c)$

Game 2: This is the same game as Game 1 except that N and e are set differently as follows.

$e, e' \xleftarrow{\$} \mathcal{P}_\lambda$
$(N, P, Q) \leftarrow \mathsf{RSA}[p \equiv 1 \bmod e'](1^\lambda)$

Game 3: This is the same game as Game 2 except that c is set differently as follows.

$c \leftarrow i\mathcal{O}(\mathsf{Max}, \tilde{C}_{N,(b+re \bmod \mathrm{ord}(\mathbb{QR}_N^+))})$

Game 4: This is the same game as Game 3 except that c is set differently as follows.

$r' \xleftarrow{\$} [\mathrm{ord}(\mathbb{QR}_N^+)]$
$c \leftarrow i\mathcal{O}(\mathsf{Max}, C_{N,r'})$

Let T_i be the event that $b' = b$ in Game i. What we want to prove is $|\Pr[T_1] - 1/2|$ is negligible. We prove it by the following lemmas.

Lemma 7. $\Pr[T_1] - \Pr[T_2]$ *is negligible if the* Φ*-hiding assumption holds.*

Proof. It is easy to see that an adversary that distinguishes Game 1 and Game

2 is reduced to an adversary that breaks the Φ-hiding assumption. □

Lemma 8. $\Pr[T_2]-\Pr[T_3]$ *is negligible if $i\mathcal{O}$ is an indistinguishability obfuscator for P/poly.*

Proof. $\tilde{C}_{N,b+re}$ and $\tilde{C}_{N,(b+re \bmod \mathrm{ord}(\mathbb{QR}_N^+))}$ compute identically for all input. Therefore the lemma follows from the property of $i\mathcal{O}$. □

Lemma 9. $\Pr[T_3] - \Pr[T_4]$ *is negligible.*

Proof. In Game 3, $\mathrm{ord}(\mathbb{QR}_N^+)$ is coprime to e with overwhelming probability. Therefore the distribution of $r \bmod \mathrm{ord}(\mathbb{QR}_N^+)$ where $r \xleftarrow{\$} [(N-1)/4]$ is negligibly close to the uniform distribution on $\mathbb{Z}_{\mathrm{ord}(\mathbb{QR}_N^+)}$ since $(N-1)/4$ is negligibly close to $\mathrm{ord}(\mathbb{QR}_N^+)$. □

Lemma 10. $\Pr[T_4] = 1/2$

Proof. In Game 4, $\mathcal{A}$ obtain no information of b, therefore the probability that $\mathcal{A}$ predicts b is $1/2$. □

□

Chapter 4

Adversary-dependent Lossy Trapdoor Function

4.1 Introduction

In this chapter, we study a cryptographic primitive called a lossy trapdoor function (LTDF). Though there are some “factoring-based construction”of LTDFs, there is no known construction of LTDFs whose security is rigorously reduced to the factoring assumption. In this part, we first relax the definition of an LTDF to define an adversary-dependent lossy trapdoor functions (ad-LTDF). Then we show that in many applications of LTDFs, we can replace LTDFs with ad-LTDFs. Moreover, we give a construction of an ad-LTDF based on the factoring assumption w.r.t. semi-smooth RSA subgroup moduli (SS moduli), which is a special type of RSA moduli introduced by Groth [Gro05]. As a result, we almost automatically obtain a new constructions of collision resistant hash function, CPA/CCA secure PKE schemes and DPKE scheme based on the same assumption. Especially, our DPKE scheme is the first scheme that satisfies the security notion called the PRIV security for block sources proposed by Boldyreva et al. [BFO08] solely based on the factoring assumption. Besides direct applications of ad-LTDFs, we construct a CCA secure PKE scheme based on the factoring assumption w.r.t. SS moduli whose ciphertext overhead is the shortest among schemes based on the same assumption.

4.1.1 Background

In modern cryptography, constructing provably secure cryptographic primitives is an important research topic. In this line of researches, Peikert and Waters [PW08] proposed *lossy trapdoor functions* (LTDFs) and constructed a number of cryptographic primitives such as a collision resistant hash function, a chosen plaintext (CPA) and chosen ciphertext (CCA) secure public key encryption (PKE) schemes and an oblivious transfer scheme based on LTDFs. Following the work, it is also shown that LTDFs can be used for constructing a deterministic encryption (DE) scheme [BFO08], a selective opening attack (SOA) secure PKE scheme [BHY09], universally composable commitment [NFT09] etc. As seen above, LTDFs have many applications, and therefore it is important to research concrete constructions of LTDFs.

As concrete constructions of LTDFs, Peikert and Waters [PW08] constructed schemes based on the decisional Diffie-Hellman (DDH) and learning with errors (LWE) assumptions. After that, many constructions of LTDFs have been proposed thus far. Among them, LTDFs related to the factoring are based on the quadratic residuosity (QR) [FGK$^+$10], decisional composite residuosity (DCR) [FGK$^+$10], Φ-hiding [KOS10], or general class of subgroup decision assumptions [XLL$^+$13], all of which are decision assumptions. On the other hand, there is no known construction of an LTDF based on the factoring assumption or a factoring-related search assumption. In general, search assumptions are rather weaker than decision assumptions. Thus it is important to research the possibility of constructing LTDFs based on a search assumption.

4.1.2 Our Result

In this chapter, though we do not construct LTDFs based on the factoring assumption, we construct an *adversary dependent lossy trapdoor function* (ad-LTDF), which is a new notion we introduce, based on the factoring assumption w.r.t. semi-smooth RSA subgroup (SS) moduli, which are RSA moduli of a special form [Gro05]. Then we show that ad-LTDFs can replace LTDFs in many applications. As a result, we immediately obtain factoring-based cryptographic primitives including a hash function, PKE scheme and DPKE scheme. Besides direct applications of ad-LTDFs, by using similar technique, we construct CCA secure PKE scheme with compact ciphertext based on the factoring assumption w.r.t. SS moduli. More details are given in the following.

Adversary-dependent lossy trapdoor function. We first reconsider the definition of LTDFs, and introduce a notion of an ad-LTDF, which is a weaker variant of an LTDF. Intuitively, an LTDF is a computationally indistinguishable pair of an injective and lossy functions. Here, the description of lossy functions should be fixed by the scheme. On the other hand, for ad-LTDFs, we allow a description of lossy function to depend on an adversary. That is, we only require that for any efficient adversary $\mathcal{A}$ there exists a lossy function that $\mathcal{A}$ cannot distinguish from an injective function. We observe that this significant relaxation does not harm the security of many LTDF-based cryptographic constructions. This is because in many LTDF-based schemes, lossy functions are used only in security proofs and they do not appear in the real scheme. This means that even if lossy functions depend on an adversary, we can still prove the security of the scheme. By this observation, we can see that ad-LTDFs can replace LTDFs in many applications.

Moreover, we construct an ad-LTDF based on the factoring assumption w.r.t. SS moduli, which is introduced by Groth [Gro05]. As a result, we can instantiate many LTDF-based constructions based on the factoring assumption w.r.t. SS moduli. The intuition of the construction of the ad-LTDF is given in Sec. 4.1.3.

Applications of ad-LTDFs. As stated above, ad-LTDFs can replace LTDFs in many applications, and we give a construction of an ad-LTDF under the factoring assumption w.r.t. SS moduli. Thus we immediately obtain new factoring-based constructions of many cryptographic primitives such as a collision resistant hash function, CPA/CCA secure PKE scheme and a DPKE scheme. Among them, the DPKE scheme obtained by this way is the first factoring-based scheme that satisfies the PRIV security for block-sources, which is defined in [BFO08], without relying on any decision assumption.

Table 4.1. Comparison among CCA secure PKE schemes based on the factoring assumption: ℓ_N is the bit-length of an underlying composite number N, ℓ_{MAC} denotes the bit-length of a message authentication code, Factoring SS denotes the factoring assumption w.r.t. SS moduli, and we assume that an exponentiation with an exponent of length ℓ can be computed by 1.5ℓ multiplications.

Schemes	Ciphertext overhead (bit)	Public key size (bit)	Computational cost for encryption (mult)	Computational cost for decryption (mult)	Assumption
[HK09b]	$2\ell_N$	$3\ell_N$	$3\ell_N + 3.5\lambda$	$1.5\ell_N + 10.5\lambda$	Factoring
[MLLJ11]	$2\ell_N$	$3\ell_N$	18.5λ	18λ	Factoring SS
Ours	$\ell_N + \ell_{MAC}$	$O(\lambda^2 \ell_N / \log \lambda)$	$O(\lambda \ell_N^2 / \log \lambda)$	$O(\lambda \ell_N^2 / \log \lambda)$	Factoring SS

CCA secure PKE with short ciphertext. Besides direct applications of ad-LTDFs, we construct a CCA secure PKE scheme whose ciphertext overhead is the shortest among schemes based on the factoring assumption w.r.t. SS moduli. Table 4.1 shows the efficiency of CCA secure PKE schemes based on the factoring assumption. Among existing schemes, the scheme proposed by Hofheinz and Kiltz [HK09b] is one of the best in regard to the ciphertext overhead, which consists of 2 elements of $\mathbb{Z}_N^*$. Mei et al. [MLLJ11] improved the efficiency of the Hofheinz-Kiltz scheme [HK09b] in regard to encryption and decryption costs by using SS moduli. However, they did not improve the ciphertext overhead. In contrast, the ciphertext overhead of our scheme consists of only 1 element of $\mathbb{Z}_N^*$ and a message authentication code (MAC), whose bit-length can be much smaller than that of N. By giving a concrete parameter, the ciphertext overhead of our scheme is 1360-bit for 80-bit security whereas that of [HK09b] is 2048-bit. On the other hand, the public key size of our scheme is much larger than that of [HK09b], and an encryption and decryption are much less efficient than those in [HK09b]. We note that the reduction from the CCA security of our scheme to the factoring assumption w.r.t. SS moduli is quite loose, but all known CCA secure PKE scheme based on the factoring assumption (including [HK09b, MLLJ11]) also require loose reductions because they require Blum-Blum-Shub pseudo-random number generator [BBS86].

We note that there is a strong negative result for a CCA secure PKE scheme whose ciphertext overhead is less than 2 group elements in a prime order setting [HMS12]. Even in a composite order setting, there are only a few CCA secure PKE schemes whose ciphetext overhead is less than 2 group elements, all of which rely on a subgroup decision assumption [HK07, KPSY09, HK09a] or an interactive assumption [KMO10] stronger than the factoring assumption. Ours is the first scheme to overcome this bound based solely on the factoring assumption (though our assumption is the factoring assumption w.r.t. SS moduli, which may not be considered standard).

4.1.3 Our Technique

Difficulty of constructiing LTDFs based on a search assumption. Before explaining our technique, we first explain why it is difficult to construct LTDFs based on a search assumption. Recall that an LTDF is a computationally indistinguishable pair of injective and lossy functions. Apparently, the definition of LTDFs itself requires the hardness of a decision problem. Thus for constructing LTDFs based on a search assumption, we have to rely on some "search-to-decision" reduction. As a general technique for such a reduction, there is the Goldreich-Levin hardcore theorem [GL89], which enables us to extract “pseudorandomness” from hardness of any search problem. However, the Goldreich-Levin hardcore bit destroys algebraic structures of original problems. On the other hand, considering existing constructions of LTDFs, algebraic structures of underlying problems are crucial for constructing LTDFs. Thus, for constructing LTDFs based on search assumptions, we have to establish another “search-to-decision” reduction technique that does not hurt underlying algebraic structures. In the context of lattice problems, this has been already done. Namely, it is shown that search-LWE and decision-LWE assumptions are equivalent [Reg05]. Thus LTDFs can be constructed based on the search-LWE assumption. However, there is no known such a reduction in the context of the factoring problem. Namely, we have no reduction from decision assumptions such as QR, DCR, or more general subgroup decision assumptions to the factoring assumption.

New search-to-decision reduction technique. The core of this work is to give a new search-to-decision reduction technique in the context of factoring w.r.t. SS moduli. Namely, we introduce a new decision assumption that we call the adversary-dependent decisional RSA subgroup (ad-DRSA) assumption, and reduce the ad-DRSA assumption to the factoring assumption w.r.t. SS moduli. In the following, we explain the technique in more detail.

We say that a composite number N is an SS modulus if it can be written as $N = PQ = (2pp' + 1)(2qq' + 1)$, where P and Q are primes with the same length, p and q are “smooth” numbers (i.e., products of distinct small primes) and p' and q' are relatively large primes. Then the group of quadratic residues $\mathbb{QR}_N$ is a cyclic group of order $pqp'q'$, and has many subgroups since pq is smooth. With respect to SS moduli, Groth [Gro05] proposed the decisional RSA subgroup (DRSA) assumption , which claims that any PPT adversary cannot distinguish a random element of G from that of $\mathbb{QR}_N$ where G is the unique subgroup of $\mathbb{QR}_N$ of order $p'q'$.

Our first observation is that if there exists an algorithm that breaks the DRSA assumption, then one can find at least one small prime that divides $\Phi(N)$. This can be seen by the following argument: Assume that all prime factors of pq are of ℓ_B-bit length. (Since pq is smooth, ℓ_B is relatively small. Especially, we set $\ell_B = O(\log \lambda)$.) Recall that the DRSA assumption claims that any PPT algorithm cannot distinguish a random element of G from that of $\mathbb{QR}_N$. This is equivalent to that the distributions of $g^{p_1 \cdots p_M}$ and g are indistinguishable where $g \xleftarrow{\$} \mathbb{QR}_N$ and $p_1, \ldots, p_M$ are the all ℓ_B-bit primes (and thus M is the number of the all ℓ_B-bit primes). If there exists an algorithm $\mathcal{A}$ that breaks the DRSA assumption, then it distinguishes these two distributions. Thus, by the hybrid argument, there exists $j \in [M]$ such that $\mathcal{A}$ distinguishes the distribution of $g^{p_1 \cdots p_{j-1}}$ from $g^{p_1, \cdots p_j}$. By using $\mathcal{A}$, one can find this p_j by the exhaustive search since M is polynomial in the security parameter in our parameter setting. (See Sec. 4.3 for more detail.) For this p_j, we have $p_j | \Phi(N)$ (with overwhelming probability) since otherwise p_j-th power on $\mathbb{QR}_N$ is a permutation on the group and thus distributions of $g^{p_1 \cdots p_{j-1}}$ and $g^{p_1, \cdots p_j}$ are completely identical. The above argument proves that if there exists an algorithm that breaks the DRSA assumption, then one can find at least one small prime that divides $\Phi(N)$. However, this fact states nothing about the reduction from the DRSA assumption to the factoring assumption since even if one can find one small prime p that divides $\Phi(N)$, we do not know how to factorize N.

Here, we relax the DRSA assumption to define the *adversary-dependent decisional RSA subgroup* (ad-DRSA) assumption. Intuitively, the ad-DRSA assumption claims that for any PPT adversary $\mathcal{A}$, there exists a subgroup $S_{\mathcal{A}}$ of $\mathbb{QR}_N$ such that $\mathcal{A}$ does not distinguish a random element of $S_{\mathcal{A}}$ from that of $\mathbb{QR}_N$. More precisely, the ad-DRSA assumption is parametrized by an integer $m \leq M$, and m-ad-DRSA assumption claims that for any PPT algorithm $\mathcal{A}$, there exists at least one choice of $p_1, \ldots p_m$ out of all ℓ_B-bit primes such that $\mathcal{A}$ cannot distinguish $g^{p_1 \cdots p_m}$ from g where $g \xleftarrow{\$} \mathbb{QR}_N$. By this definition, if there exists a PPT algorithm $\mathcal{A}$ that breaks the m-ad-DRSA assumption, then $\mathcal{A}$ distinguishes $g^{p_1 \cdots p_m}$ from g for *all* choices of $p_1, \ldots, p_m$. If m is sufficiently smaller than M, then there exists "many" choices of $p_1, \ldots, p_m$ and thus one can find "many" primes that divides $\Phi(N)$: One can find at least one such prime for each choice of $p_1, \ldots, p_m$ by the similar method as in the case of the DRSA assumption. Then the product of these primes is a large divisor of $\Phi(N)$ and thus one can factorize N by using the Coppersmith theorem [Cop96], which claims that if one is given a "large" divisor of $\Phi(N)$, then one can factorize N efficiently. Thus, the m-ad-DRSA assumption is reduced to the factoring assumption.

Remark 11. *We remark that if m is so small that there exists a choice of $p_1, \ldots, p_m$, all of which are coprime to $\Phi(N)$, then the m-ad-DRSA assumption is trivial since in that case g and $g^{p_1 \cdots p_m}$ are distributed identically. We show that there exists a parameter choice such that m-ad-DRSA assumption is non-trivial and it can be reduced to the factoring assumption simultaneously.*

How to use the ad-DRSA assumption. As explained above, we show a reduction from the ad-DRSA assumption, which is a certain type of a subgroup assumption, to the factoring assumption. However, the ad-DRSA assumption is not an ordinary subgroup decision assumption: Roughly speaking, it only claims that for any PPT adversary $\mathcal{A}$, there exists a subgroup $S_{\mathcal{A}} \in \mathbb{QR}_N$ such that $\mathcal{A}$ cannot distinguish random elements of $S_{\mathcal{A}}$ from $\mathbb{QR}_N$. Therefore, it cannot be used for constructions where elements of a subgroup are used in the real descriptions of the scheme. On the other hand, if elements of a subgroup are used only in the security proof, the ad-DRSA assumption suffices. We give two examples of such cases.

One is ad-LTDFs. As explained in Sec. 4.1.2, ad-LTDFs is a relaxation of LTDFs such that descriptions of lossy functions can depend on an adversary. For constructing ad-LTDFs based on the ad-DRSA assumption, we simply imitate the construction by Xue et al. [XLL$^+$13], who constructed LTDFs based on the (standard) DRSA assumption. We observe that in their construction, the descriptions of injective functions consist only of elements of $\mathbb{QR}_N$, and elements of its subgroup are used only in the descriptions of lossy functions. Therefore even if we replace the DRSA assumption with the ad-DRSA assumption, only lossy functions depend on an adversary. This meets the definition of the ad-LTDFs.

The other is the hash-proof system-based CCA secure public key encryption. Hofheniz and Kiltz [HK07] introduced the concept of constraind CCA (CCCA) security, and showed efficient constructions of CCA secure public key encryption schemes based on a hash proof system, which can be constructed from any subgroup decision assumption [CS02]. Though elements of a subgroup are used in the real protocol of their original construction, it is easy to see that even if elements of a subgroup are replaced with those of a larger group, the scheme is still secure because they are indistinguishable by the assumption. Thus that scheme can be instantiated based on the ad-DRSA assumption.

4.1.4 Discussion

Plausibility of the factoring assumption w.r.t. SS moduli. Here, we discuss the plausibility of the assumption we used. SS moduli was first introduced by Groth [Gro05] in 2005 and they have been used in some works [MLLJ11, XLL$^+$13, YYN$^+$14]. All of these works assume the factoring assumption w.r.t. SS moduli (or more stronger assumptions). On the other hand, in 2011, Coron et.al. [CJM$^+$11] gave a cryptanalysis against the Groth's work [Gro05]. However, they did not improve attacks against SS moduli. Thus, we can say that SS moduli has attracted a certain amount of attention in the sense of both constructions and cryptanalysis, but no fatal attack is found thus far. Therefore we believe that the hardness of factoring SS moduli is rather reliable.

Interpretation of our result. In this chapter, we constructed a weaker variant of LTDF (ad-LTDF) based on the factoring assumption w.r.t. SS moduli. One may wonder how meaningful our result is since an SS modulus is not an RSA modulus of a standard form. We believe that our result is meaningful in terms of that we constructed an "LTDF-like primitive" (ad-LTDF), which can replace LTDFs in many applications, based on a search assumption (factoring w.r.t. SS moduli) rather than a decision assumption. Although the application given in this chapter is limited to the case of SS moduli, we hope that our new search-to-decision reduction technique can be extended to other general settings.

Limitation of ad-LTDFs. Though ad-LTDFs can replace LTDFs in many cases, there exist some LTDF-based primitives that cannot be obtained from ad-LTDFs. A typical example is the oblivious transfer protocol proposed by Peikert and Waters [PW08]. The reason why we cannot construct the scheme based on ad-LTDFs is that in the scheme, a lossy function is explicitly required. Specifically, a receiver sends a pair of injective and lossy functions to a sender. Since we cannot specify a lossy function before fixing an adversary, we cannot instantiate this scheme based on ad-LTDFs.

4.1.5 Related Work

Factoring based CCA secure PKE schemes. In 2009, Hofheinz and Kiltz [HK09b] proposed the first practical CCA secure PKE scheme under the factoring assumption in the standard model. After that, many variants of the scheme are proposed thus far [MLLJ11, LLML12, LLL13, YYN$^+$14]. However, none of them improve the ciphertext overhead of the scheme. On the other hand, the ciphretext overhead of our proposed scheme is shorter than those of them.

4.2 Preliminaries

4.2.1 Known Lemmas

Here, we review known lemmas used in this chapter. First, we review a simple variant of the Hoeffding inequality [Hoe63].

Lemma 11. *(Hoeffding inequality) Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be probability distributions over*

$\{0,1\}$. *Let* $X_1, \ldots, X_K$ *be* K *independent random variables with the distribution* $\mathcal{D}_1$ *and* $Y_1, \ldots, Y_K$ *be* K *independent random variables with the distribution* $\mathcal{D}_2$. *If we define* $\epsilon := |\Pr[X = 1 : X \xleftarrow{\$} \mathcal{D}_1] - \Pr[Y = 1 : Y \xleftarrow{\$} \mathcal{D}_2]|$, *then* $\Pr[|\frac{|\Sigma_{k=1}^{K} X_k - \Sigma_{i=k}^{K} Y_k|}{K} - \epsilon| \geq \delta] \leq 4e^{-\delta^2 K/2}$ *holds.*

The following is the generalized leftover hash lemma [DORS08].

Lemma 12. *(Generalized leftover hash lemma) Let* $X \in \{0,1\}^{n_1}$ *and* Y *be random variables. Let* $\mathcal{H}$ *be a family of pairwise independent hash function from* $\{0,1\}^{n_1}$ *to* $\{0,1\}^{n_2}$. *Then we have* $\Delta((H(X), H, Y), (U, H, Y)) \leq \delta$ *where* $H \xleftarrow{\$} \mathcal{H}$ *as long as* $\tilde{H}_\infty(X|Y) \geq n_2 + 2\log(1/\delta)$.

The following is the "crooked version" of the above lemma proven by Boldyreva et al. [BFO08].

Lemma 13. *(Generalized crooked leftover hash lemma [BFO08, Lemma7.1]) Let* $X \in \{0,1\}^n$ *and* Y *be random variables. Let* $\mathcal{H}$ *be a family of pairwise independent hash function from* $\{0,1\}^n$ *to* R *and* f *be a function from* R *to* S. *Then for* $H \xleftarrow{\$} \mathcal{H}$, *we have* $\Delta((f(H(X)), H, Y), (f(U), H, Y)) \leq \delta$ *as long as* $\tilde{H}_\infty(X|Y) \geq \log|S| + 2\log(1/\delta) - 2$.

Finally, we review the Coppersmith theorem about bivariate integer equations. The following lemma is a special case of [Cop96, Theorem 3].

Lemma 14. *Let* $p(x, y) = a + bx + cy$ *be a polynomial over* $\mathbb{Z}$. *For positive integers* X, Y *and* $W = \max\{a, bX, cY\}$, *if* $XY < 2^{-8} \cdot W$ *holds, then one can find all solutions* (x_0, y_0) *such that* $p(x_0, y_0) = 0$, $|x_0| < X$ *and* $|y_0| < Y$ *in time polynomial in* $\log_2 W$.

4.3 Semi-smooth RSA subgroup modulus

Here, we define a semi-smooth RSA subgroup modulus (SS modulus) and state its properties. For integers ℓ_B, t_p and t_q, We say that $N = PQ = (2pp' + 1)(2qq' + 1)$ is an (ℓ_B, t_p, t_q)-semi-smooth RSA subgroup ((ℓ_B, t_p, t_q)-SS) modulus if the following conditions hold.

- P and Q are distinct prime numbers with the same length that satisfy $\gcd(P - 1, Q - 1) = 2$.
- p' and q' are distinct primes larger than 2^{ℓ_B}.
- p and q are products of t_p and t_q distinct ℓ_B-bit primes. Here, an ℓ_B-bit prime means a prime number between $2^{\ell_B - 1}$ and 2^{ℓ_B}. We note that we have $\gcd(p, q) = 1$ since we have $\gcd(P - 1, Q - 1) = 2$.

We define $t := t_p + t_q$. Let $\mathcal{P}_{\ell_B}$ be the set of all ℓ_B-bit primes, and $M_{\ell_B} := |\mathcal{P}_{\ell_B}|$. We define the group of quadratic residues as $\mathbb{QR}_N := \{u^2 : u \in \mathbb{Z}_N^*\}$. This is a subgroup of $\mathbb{Z}_N^*$, and a cyclic group of order $pqp'q'$. Then there exists unique subgroups of order $p'q'$ and pq, and we denote them by G and $G^{\perp}$ respectively. Then we have $\mathbb{QR}_N = G \times G^{\perp}$. That is, for any element $g \in \mathbb{QR}_N$, we can uniquely represent $g = g(G)g(G^{\perp})$ by using $g(G) \in G$ and $g(G^{\perp}) \in G^{\perp}$. Moreover, if the factorization of N is given, then we can compute $g(G)$ and $g(G^{\perp})$ from g efficiently.

When N is an SS modulus, we cannot say that a random element g of $\mathbb{QR}_N$ is a generator (i.e., $\mathrm{ord}(g) = pqp'q'$) with overwhelming probability. However, we can prove that g has an order larger than a certain value with overwhelming probability.

Lemma 15. *([Gro05, Lemma2]) Let N be an (ℓ_B, t_p, t_q)-SS modulus. For any integer $d < t$ if $\frac{(t2^{1-\ell_B})^{d+1}}{(1-t2^{1-\ell_B})(d+1)!}$ is negligible, then $\Pr[\mathrm{ord}(g) \geq p'q'2^{(t-d)(\ell_B-1)} : g \xleftarrow{\$} \mathbb{QR}_N]$ is overwhelming. Especially, $\Pr[\mathrm{ord}(g(G^{\perp})) \geq 2^{(t-d)(\ell_B-1)} : g \xleftarrow{\$} \mathbb{QR}_N]$ is overwhelming.*

When ℓ_B is small, $\mathrm{ord}(G^{\perp})$ is smooth, and therefore the discrete logarithm on the group can be solved efficiently by the Pohlig-Hellman algorithm [PH78].

Lemma 16. *([Gro05]) If $\ell_B = O(\log \lambda)$, then the discrete logarithm problem on $G^{\perp}$ can be solved efficiently. More precisely, there exists a PPT algorithm that, given an (ℓ_B, t_p, t_q)-SS modulus N, $g \in G^{\perp}$ and g^x, outputs $x \mod \mathrm{ord}(g)$.*

By combining the above lemmas, we obtain the following lemma.

Lemma 17. *Let N be an (ℓ_B, t_p, t_q)-SS modulus and we assume $\ell_B = O(\log(\lambda))$. If $\frac{(t2^{1-\ell_B})^{d+1}}{(1-t2^{1-\ell_B})(d+1)!}$ is negligible and $x \leq 2^{(t-d)(\ell_B-1)}$ holds, then there exists a PPT algorithm PLog that, given P,Q, g, g^x, outputs x with overwhelming probability where $g \xleftarrow{\$} \mathbb{QR}_N$.*

Hardness assumptions. Here, we give definitions of two hardness assumptions. Let IGen be an algorithm that is given the security parameter 1^{λ} and outputs an (ℓ_B, t_p, t_q)-SS modulus with its factorization. We first define the factoring assumption.

Definition 12. *We say that the factoring assumption holds with respect to IGen if for any PPT algorithm $\mathcal{A}$, $\Pr[\mathcal{A}(N) \in \{P, Q\} : (N, P, Q) \leftarrow \mathsf{IGen}(1^{\lambda})]$ is negligible.*

Next, we define the *decisional RSA subgroup* (DRSA) assumption proposed by Groth [Gro05]. This assumption claims that any PPT algorithm cannot distinguish a random element of G from that of $\mathbb{QR}_N$. We note that actually we do not use this assumption in this dissertation. We include this only for the information of the reader.

Definition 13. *We say that the decisional RSA subgroup (DRSA) assumption holds with respect to* IGen *if for any PPT algorithm* $\mathcal{A}$, $|\Pr[1 \leftarrow \mathcal{A}(N,g) : (N,P,Q) \leftarrow \mathsf{IGen}(1^\lambda); g \xleftarrow{\$} \mathbb{QR}_N] - \Pr[1 \leftarrow \mathcal{A}(N,g) : (N,P,Q) \leftarrow \mathsf{IGen}(1^\lambda); g \xleftarrow{\$} G]|$ *is negligible.*

Attacks. We review factoring attacks against SS moduli as discussed in [Gro05]. As shown in [Gro05], by using Pollard's ρ-method [Pol75], we can factorize an SS modulus in time $\tilde{O}(\min(\sqrt{p'}, \sqrt{q'}))$. As another method, by using Naccache et al.'s algorithm [NS98], if a divisor of $P-1$ or $Q-1$ larger than $N^{1/4}$ is given, then N can be factorized efficiently. Thus ℓ_B should be large enough so that it is difficult to guess a significant portion of factors of p or q. In 2011, Coron et al. [CJM+11] proposed a new factoring algorithm for a certain class of RSA moduli that includes SS moduli. For the case of SS moduli, their algorithm work in time $\tilde{O}(\min(\sqrt{p'}, \sqrt{q'}))$, which matches the time complexity of Pollard's ρ-method. As observed in [Gro05], other methods such as the baby-step giant-step algorithm [Sha71], Pollard's λ-method [Pol78] or Pollard's $p-1$ method [Pol74] require $O(\min(p', q'))$ time.

The above attacks use the structure of SS moduli. On the other hand, there are algorithms such as the elliptic curve method [Len87] or the general number field sieve [CP05], which can be applied to general RSA moduli. Among these algorithms, general number field sieve is asymptotically the most efficient and its heuristic running time is $\exp((1.92 + o(1)) \ln(N)^{1/3} \ln\ln(N)^{2/3})$.

Parameter settings. Here, we discuss parameter settings of SS moduli. We have to set parameters to avoid the above attacks. We first give an asymptotic parameter setting. We set $\ell_{p'} = \ell_{q'} = O(\lambda)$, $\ell_B = \lfloor 4 \log \lambda \rfloor$ and $t_p = t_q = O(\lambda^3/\log\lambda)$ (then we have $\ell_N \approx \ell_{p'} + \ell_{q'} + t\ell_B = O(\lambda^3)$). In this setting, we have $M_{\ell_B} = O(\lambda^4/\log\lambda)$ by the prime number theorem and thus there exists exponentially many choices of p and q. If we set $d := \lfloor t/4 \rfloor$, then $\frac{(t2^{1-\ell_B})^{d+1}}{(1-t2^{1-\ell_B})(d+1)!}$ is negligible*1. We use the fact that in this parameter setting, given N, $g \in \mathbb{QR}_N$ and $p_1, \ldots, p_m$ for $m \leq M_{\ell_B}$, $g^{p_1 \cdots p_m}$ can be computed in polynomial time in λ. This is because we have $m \leq M_{\ell_B} = O(\lambda^4/\log(\lambda))$ and $p_1 \ldots p_m \leq 2^{\ell_B M_{\ell_B}} = 2^{O(\lambda^4)}$, and thus $p_1 \ldots p_m$-th power can be computed by $O(\lambda^4)$ multiplications. We use this asymptotic parameter setting throughout the chapter. As a concrete parameter, Groth [Gro05] proposed to set $\ell'_p = \ell'_q = 160$, $\ell_B = 15$, $t_p = t_q = 32$ and $d = 7$ for 80-bit security (then we have $\ell_N = 160 \cdot 2 + 15 \cdot 2 \cdot 32 = 1280$). We use this parameter for the construction

*1 In fact, d can be set as $d := \lfloor ct \rfloor$ for any small enough constant c.

of CCA secure PKE scheme with compact ciphertext (Section 4.8). However, this parameter does not give us enough lossiness in the construction of ad-LTDFs. Thus we propose to set $\ell'_p = \ell'_q = 160$, $\ell_B = 15$, $t_p = t_q = 70$ and $d = 8$ (then we have $\ell_N = 160 \cdot 2 + 15 \cdot 2 \cdot 70 = 2420$) for 80-bit security for other applications (Sec. 4.5 to 4.7). We note that the number of $\ell_B = 15$-bit primes is 1612. Therefore the possible choice of $t = 64$ or 140 primes out of them is much larger than 2^{80} and thus it is hard to guess the significant portion of their factors.

4.4 Adversary-dependent Decisional RSA Subgroup Assumption

In this section, we generalize the DRSA assumption. Specifically, we define the m-adversary-dependent decisional RSA subgroup (m-ad-DRSA) assumption for any integer $m \leq M_{\ell_B}$ with respect to (ℓ_B, t_p, t_q)-SS moduli. Intuitively, this assumption claims that for any PPT algorithm $\mathcal{A}$, there exist distinct ℓ_B-bit primes $p_1, \ldots, p_m$ such that $\mathcal{A}$ does not distinguish g from $g^{p_1 \cdots p_m}$ where g is a random element of $\mathbb{QR}_N$. We prove that under a certain condition, the m-ad-DRSA assumption holds under the factoring assumption.

First we give the precise definition of the m-ad-DRSA assumption.

Definition 14. *Let* IGen *be a PPT algorithm that generates an* (ℓ_B, t_p, t_q)*-SS RSA modulus. We say that for any integer* $m \leq M_{\ell_B}$*, the* m*-adversary-dependent decisional RSA subgroup (*m*-ad-DRSA) assumption holds with respect to* IGen *if for any noticeable function* ϵ *and PPT algorithm* $\mathcal{A}$*, there exists a PPT algorithm* $\mathcal{S}_{\mathcal{A},\epsilon}$ *that is given* (ℓ_B, t_p, t_q)*-SS RSA modulus* N *and outputs distinct* ℓ_B*-bit primes* $p_1, \ldots, p_m$*, such that the following is satisfied. If we let*

$$P_0 := \Pr\left[1 \leftarrow \mathcal{A}(N, g) : \begin{array}{c} (N, P, Q) \leftarrow \mathsf{IGen}(1^\lambda) \\ g \xleftarrow{\$} \mathbb{QR}_N \end{array}\right]$$

$$P_1 := \Pr\left[1 \leftarrow \mathcal{A}(N, g^{p_1 \cdots p_m}) : \begin{array}{c} (N, P, Q) \leftarrow \mathsf{IGen}(1^\lambda) \\ g \xleftarrow{\$} \mathbb{QR}_N \\ \{p_1, \ldots, p_m\} \leftarrow \mathcal{S}_{\mathcal{A},\epsilon}(N) \end{array}\right]$$

then we have $|P_0 - P_1| \leq \epsilon(\lambda)$ *for sufficiently large* λ*.*

Remark 12. *One may think that the assumption defined above cannot be used for proving security of any cryptographic scheme since* ϵ *is noticeable. However, an important remark here is that* ϵ *can be an arbitrary noticeable function. Thus, in security* *proofs, we can set ϵ depending on an adversary $\mathcal{A}$'s advantage against the scheme that we want to prove secure, such that ϵ is smaller than the advantage of $\mathcal{A}$ (for infinitely many security parameters). This can be done if $\mathcal{A}$ breaks the security of the scheme since in these cases, the advantage of $\mathcal{A}$ should be non-negligible. See security proofs in Sec. 4.7 and 4.8 to see this argument indeed works.*

Remark 13. *In the above definition, if m is so small that there exists a choice of $p_1, \ldots, p_m$, all of which are coprime to $\Phi(N)$, then $g^{p_1 \cdots p_m}$ is distributed uniformly on $\mathbb{QR}_N$. In this case, m-ad-DRSA assumption is trivial. This occurs if and only if we have $M_{\ell_B} - m \geq t$. In this chapter, we set m to be relatively large so that m-ad-DRSA assumption is non-trivial. (See Remark 14.)*

The following theorem claims that the m-ad-DRSA assumption holds under the factoring assumption if m is small enough.

Theorem 11. *Let* IGen *be a PPT algorithm that generates an (ℓ_B, t_p, t_q)-SS RSA modulus where $\ell_B = O(\log \lambda)$. If the factoring assumption holds with respect to* IGen *and there exists a constant c such that $(M_{\ell_B} - m + 1)(\ell_B - 1) \geq (1/2 + c)\ell_N$ holds, then the m-ad-DRSA assumption holds with respect to* IGen.

Remark 14. *If we set $m := \lfloor M_{\ell_B} + 1 - (1/2 + c)\ell_N/(\ell_B - 1) \rfloor$ for sufficiently small c, then by the above theorem, the m-ad-DRSA assumption holds under the factoring assumption. Moreover, by setting the parameter as given in 4.3, we have $M_{\ell_B} - m \approx (1/2 + c)\ell_N/(\ell_B - 1) \approx (1/2 + c)(\ell_{p'} + \ell_{q'} + t\ell_B)/\ell_B \leq t$ for sufficiently large λ if $c < 1/2$ since $t = O(\lambda^3/\log \lambda)$ and $\ell_{p'} = \ell_{q'} = O(\lambda)$. Thus the m-ad-DRSA assumption is non-trivial.*

Before proving the theorem, we prepare a lemma related to the Coppersmith attack. Though a heuristic proof appeared in [NS98], to the best of our knowledge, this has not been proven rigorously in the literature.

Lemma 18. *Let P and Q be primes with the same length and $N = PQ$. Let e be a divisor of $\Phi(N) = (P - 1)(Q - 1)$. If there exists a positive constant c such that $e > N^{1/2+c}$ holds, then there exists a polynomial time algorithm that is given N and e, and factorizes N.*

Proof. We define e_1 and e_2 such that $e = e_1 e_2$, $e_1 | P - 1$ and $e_2 | Q - 1$. (Note that we cannot always compute e_1 and e_2 from e.) Then we can write $P = e_1 k_1 + 1$ and $Q = e_2 k_2 + 1$ by using integers k_1 and k_2. Then we have $N = PQ = (e_1 k_1 + 1)(e_2 k_2 + 1) = e k_1 k_2 + e_1 k_1 + e_2 k_2 + 1$. Therefore if we define $p(x, y) = N + ex + y$, then

$p(x,y)=0$ has a solution $(x_0,y_0)=(-k_1k_2,-(e_1k_1+e_2k_2+1))$. Let $X:=N^{1/2-c}$, $Y:=3N^{1/2}$ and $W:=\max(N,eX,Y)$. One can see that $|x_0|<X$, $|y_0|<Y$ and $XY=3N^{1-c}<2^{-8}\cdot N\leq 2^{-8}\cdot W$ hold (for sufficiently large N). Therefore one can compute the solution $(x_0,y_0)=(-k_1k_2,-(e_1k_1+e_2k_2+1))$ in polynomial time in $\log N$ by Lemma 14. Then one can compute $P+Q=e_1k_1+e_2k_2+2=-y_0+1$ and factorize N. □

Intuition for the proof of Theorem 11. Here, we give an intuition for the proof of Theorem 11. We remark that the following argument is not a rigorous one. What we have to do is to construct a PPT algorithm $\mathcal{S}_{\mathcal{A},\epsilon}$ that is given N and outputs $\{p_1,\ldots,p_m\}$ such that $\mathcal{A}$'s advantage to distinguish g from $g^{p_1\cdots p_m}$ is smaller than ϵ where $g\xleftarrow{\$}\mathbb{QR}_N$. Let list $L:=\mathcal{P}_{\ell_B}$, which is the set of all ℓ_B-bit primes. First, $\mathcal{S}_{\mathcal{A},\epsilon}$ randomly chooses m distinct primes $\{p_1,\ldots,p_m\}$ from L and test whether $\mathcal{A}$'s advantage to distinguish g from $g^{p_1\cdots p_m}$ is smaller than ϵ or not. More precisely, $\mathcal{S}_{\mathcal{A},\epsilon}$ approximates $\mathcal{A}$'s advantage by iterating the execution of $\mathcal{A}(g)$ and $\mathcal{A}(g^{p_1\cdots p_m})$ for independently random $g\xleftarrow{\$}\mathbb{QR}_N$ a number of times and counting the number that each of them outputs 1. We denote the approximated advantage by ϵ'. Due to the Hoeffding inequality [Hoe63], the approximation error can be made smaller than $\epsilon/4$ by polynomial times iterations since ϵ is noticeable. If $\epsilon'<\epsilon/2$, then $\mathcal{A}$'s real advantage is smaller than $3\epsilon/4<\epsilon$ and thus $\mathcal{S}_{\mathcal{A},\epsilon}$ outputs $\{p_1,\ldots,p_m\}$ and halts. Otherwise, $\mathcal{A}$'s advantage to distinguish g from $g^{p_1\cdots p_m}$ is larger than $\epsilon/4$. Then there exists p_j such that $\mathcal{A}$'s advantage to distinguish $g^{p_1\cdots p_{j-1}}$ from $g^{p_1\cdots p_j}$ is larger than $\epsilon/(4m)$ by the hybrid argument. $\mathcal{S}_{\mathcal{A},\epsilon}$ can find this p_j in polynomial time since $\epsilon/(4m)$ is noticeable. We remark that we have $p_j|\Phi(N)$. This is because, otherwise $\mathcal{A}$'s advantage to distinguish $g^{p_1\cdots p_{j-1}}$ from $g^{p_1\cdots p_j}$ is 0 since their distributions are completely identical and thus ϵ' should be smaller than $\epsilon/2$. Then $\mathcal{S}_{\mathcal{A},\epsilon}$ removes p_j from L. Then it randomly chooses m distinct primes $\{p_1,\ldots,p_m\}$ from L again, and do the same as the above. Then it outputs $\{p_1,\ldots,p_m\}$ and halts if approximated $\mathcal{A}$'s advantage to distinguish g from $g^{p_1\cdots p_m}$ is smaller than $\epsilon/2$, or otherwise removes some $p_{j'}|\Phi(N)$ from L. $\mathcal{S}_{\mathcal{A},\epsilon}$ repeat this procedure many times. Assume that $\mathcal{S}_{\mathcal{A},\epsilon}$ does not halts by the time it cannot choose m distinct primes from L. By that time, $M_{\ell_B}-m+1$ distinct ℓ_B-bit primes that divide $\Phi(N)$ are removed from L. Let e be the product of them. Then we have $e|\Phi(N)$ and $e\geq 2^{(\ell_B-1)(M_{\ell_B}-m+1)}\geq N^{1/2+c}$. Therefore if e is given, then we can factorize N efficiently by Lemma 18. Thus under the factoring assumption, $\mathcal{S}_{\mathcal{A},\epsilon}$ must output some $\{p_1,\ldots,p_m\}$ before $|L|$ becomes smaller than m with overwhelming probability, and $\mathcal{A}$'s advantage to distinguish g from $g^{p_1\cdots p_m}$ is smaller than $3\epsilon/4<\epsilon$.

Now we give the full proof of Theorem 11

Proof. (of Theorem 11) First, we prove the following two claims.

Claim 1. *For any PPT algorithm $\mathcal{A}$ and a noticeable function δ, there exists a PPT algorithm* $\mathrm{Approx}_{\mathcal{A},\delta}$ *that satisfies the following. Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be descriptions of distributions that are samplable in polynomial time in λ, and $\epsilon := |\Pr[1 \leftarrow \mathcal{A}(X) : X \xleftarrow{\$} \mathcal{D}_1] - \Pr[1 \leftarrow \mathcal{A}(X) : X \xleftarrow{\$} \mathcal{D}_2]|$. Then* $\mathrm{Approx}_{\mathcal{A},\delta}(1^\lambda, \mathcal{D}_1, \mathcal{D}_2)$ *outputs ϵ' such that $|\epsilon' - \epsilon| \leq \delta(\lambda)$ with overwhelming probability. (We say that* $\mathrm{Approx}_{\mathcal{A},\delta}$ *succeeds if it outputs such ϵ'.)*

Proof. The construction of $\mathrm{Approx}_{\mathcal{A},\delta}$ is as follows.

$\mathrm{Approx}_{\mathcal{A},\delta}(1^\lambda, \mathcal{D}_1, \mathcal{D}_2)$: For $i = 1$ to K where $K := \lambda/\delta(\lambda)^2$, choose X_i and Y_i according to $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively, and run $\mathcal{A}(X_i)$ and $\mathcal{A}(Y_i)$ for each i. Let k_1 be the number of the event that $\mathcal{A}(X_i)$ outputs 1 and k_2 be the number of the event that $\mathcal{A}(Y_i)$ outputs 1. Output $|k_1 - k_2|/K$.

Since δ is noticeable, K is polynomial in λ and therefore $\mathrm{Approx}_{\mathcal{A},\delta}$ is a PPT algorithm. It can be seen by Lemma 11 that $\mathrm{Approx}_{\mathcal{A},\delta}$ satisfies the desired property. □

Claim 2. *For any PPT algorithm $\mathcal{A}$ and a noticeable function ϵ, there exists a PPT algorithm* $\mathsf{Find}_{\mathcal{A},\epsilon}$ *that satisfies the following. For any (ℓ_B, t_p, t_q)-SS RSA modulus N and a set $I = \{p_1, \dots, p_m\}$ of distinct ℓ_B-bit primes, if $|\Pr[1 \leftarrow \mathcal{A}(N, g) : g \xleftarrow{\$} \mathbb{QR}_N] - \Pr[1 \leftarrow \mathcal{A}(N, g^{p_1 \cdots p_m}) : g \xleftarrow{\$} \mathbb{QR}_N]| > \epsilon(\lambda)$ holds, then* $\mathsf{Find}_{\mathcal{A},\epsilon}(N, I)$ *outputs $p_j \in I$ that divides $\Phi(N)$ with overwhelming probability. (We say that* $\mathsf{Find}_{\mathcal{A},\epsilon}$ *succeeds if it outputs such p_j or the inequality assumed is false.)*

Proof. The construction of $\mathsf{Find}_{\mathcal{A},\epsilon}$ is as follows.

$\mathsf{Find}_{\mathcal{A},\epsilon}(N, I = \{p_1, \dots, p_m\})$: Define distributions $\mathcal{D}_0 := \{(N, g) : g \xleftarrow{\$} \mathbb{QR}_N\}$ and $\mathcal{D}_j := \{(N, g^{p_1 \cdots p_j}) : g \xleftarrow{\$} \mathbb{QR}_N\}$ $(j = 1, \dots, m)$. For $j := 1$ to m, repeat the following.

```
    Compute $\epsilon' \leftarrow \mathrm{Approx}_{\mathcal{A},\epsilon/(2m)}(1^\lambda, \mathcal{D}_{j-1}, \mathcal{D}_j)$.
    If $\epsilon' > \epsilon/(2m)$, then output $p_j$ and halt.
If it does not halt by the time the above loop is finished, then output $\perp$.
```

First, we show $\mathsf{Find}_{\mathcal{A},\epsilon}$ is a PPT algorithm. Since $m \leq M_{\ell_B}$ is polynomial in λ and thus $\epsilon/(2m)$ is noticeable, $\mathrm{Approx}_{\mathcal{A},\epsilon/(2m)}$ is a PPT algorithm. Therefore $\mathsf{Find}_{\mathcal{A},\epsilon}$ is a PPT algorithm. We prove that $\mathsf{Find}_{\mathcal{A},\epsilon}$ satisfies the desired property. First, we assume that all executions of $\mathrm{Approx}_{\mathcal{A},\epsilon/(2m)}$ called by $\mathsf{Find}_{\mathcal{A},\epsilon}$ succeed. The probability that this assumption is satisfied is overwhelming since the number of executions of $\mathrm{Approx}_{\mathcal{A},\epsilon/(2m)}$ is polynomial in λ and each execution succeeds with overwhelming probability. First, we prove that $\mathsf{Find}_{\mathcal{A},\epsilon}$ outputs any prime $p_j \in I$ if we have $|\Pr[1 \leftarrow \mathcal{A}(N, g) : g \overset{\$}{\leftarrow} \mathbb{QR}_N] - \Pr[1 \leftarrow \mathcal{A}(N, g^{p_1 \cdots p_m}) : g \overset{\$}{\leftarrow} \mathbb{QR}_N]| > \epsilon$. By the hybrid argument, there exists $j \in [m]$ such that $|\Pr[1 \leftarrow \mathcal{A}(X) : X \overset{\$}{\leftarrow} \mathcal{D}_{j-1}] - \Pr[1 \leftarrow \mathcal{A}(X) : X \overset{\$}{\leftarrow} \mathcal{D}_j]| > \epsilon/m$ holds. For such j, if we let $\epsilon' := \mathrm{Approx}_{\mathcal{A},\epsilon/(2m)}(\mathcal{D}_{j-1}, \mathcal{D}_j)$, then we have $\epsilon' > \epsilon/m - \epsilon/(2m) = \epsilon/(2m)$ and thus p_j is output. Then we prove that if p_j is output by $\mathsf{Find}_{\mathcal{A},\epsilon}$, then $p_j | \Phi(N)$ holds. If p_j does not divide $\Phi(N)$, then p_j is coprime to $\mathrm{ord}(\mathbb{QR}_N)$, and especially p_j-th power is a permutation on the group $\{g^{p_1 \cdots p_{j-1}} : g \in \mathbb{QR}_N\}$. Therefore $\mathcal{D}_{j-1}$ and $\mathcal{D}_j$ are completely the identical distributions. Therefore we have $|\Pr[1 \leftarrow \mathcal{A}(X) : X \overset{\$}{\leftarrow} \mathcal{D}_{j-1}] - \Pr[1 \leftarrow \mathcal{A}(X) : X \overset{\$}{\leftarrow} \mathcal{D}_j]| = 0$. Thus if we let $\epsilon' := \mathrm{Approx}_{\mathcal{A},\epsilon/(2m)}(\mathcal{D}_{j-1}, \mathcal{D}_j)$, then we have $\epsilon' < \epsilon/(2m)$, and thus such p_j cannot be output. □

Then we go back to the proof of Theorem 11. For any PPT algorithm $\mathcal{A}$ and a noticeable function ϵ, we construct a PPT algorithm $\mathcal{S}_{\mathcal{A},\epsilon}$ such that $\Pr[1 \leftarrow \mathcal{A}(N, g) : (N, P, Q) \leftarrow \mathsf{IGen}(1^\lambda); g \overset{\$}{\leftarrow} \mathbb{QR}_N] - \Pr[1 \leftarrow \mathcal{A}(N, g^{p_1 \cdots p_m}) : (N, P, Q) \leftarrow \mathsf{IGen}(1^\lambda); g \overset{\$}{\leftarrow} \mathbb{QR}_N; \{p_1, \ldots, p_m\} \leftarrow \mathcal{S}_{\mathcal{A},\epsilon}(N)] \leq \epsilon(\lambda)$ holds for sufficiently large λ. The construction of $\mathcal{S}_{\mathcal{A},\epsilon}$ is as follows.

```
$\mathcal{S}_{\mathcal{A},\epsilon}(N)$ : Let $L := \mathcal{P}_{\ell_B}$. (Recall that $\mathcal{P}_{\ell_B}$ is the set of all $\ell_B$-bit primes.)
    While $|L| \geq m$, repeat the following.
        Choose distinct $\ell_B$-bit primes $p_1, \ldots, p_m$ from $L$ randomly, and let $I := \{p_1, p_2, \ldots, p_m\}$, $\mathcal{D}_0 := \{(N, g) : g \overset{\$}{\leftarrow} \mathbb{QR}_N\}$ and $\mathcal{D}_m := \{(N, g^{p_1 \cdots p_m}) : g \overset{\$}{\leftarrow} \mathbb{QR}_N\}$. Compute $\epsilon' \leftarrow \mathsf{Approx}_{\mathcal{A},\epsilon/4}(1^\lambda, \mathcal{D}_0, \mathcal{D}_m)$. If $\epsilon' < \epsilon/2$, then output $I$ and halts. Otherwise run $\tilde{p} \leftarrow \mathsf{Find}_{\mathcal{A},\epsilon/4}(N, I)$. If $\tilde{p} \in L$ then remove $\tilde{p}$ from $L$, otherwise remove a random element from $L$.
    If it does not halt by the time the above loop finishes, then it outputs $\perp$.
```

First, we prove that $\mathcal{S}_{\mathcal{A},\epsilon}(N)$ is a PPT algorithm. Since ϵ is noticeable, $\mathsf{Approx}_{\mathcal{A},\epsilon/4}$ and $\mathsf{Find}_{\mathcal{A},\epsilon/4}$ are PPT algorithms. Moreover the number of repeat is at most $M_{\ell_B} - m + 1 \leq M_{\ell_B}$, which is polynomial in λ. Therefore $\mathcal{S}_{\mathcal{A},\epsilon}(N)$ is a PPT algorithm.

Then we prove that $\mathcal{S}_{\mathcal{A},\epsilon}(N)$ satisfies the desired property. In the following, we assume that all executions of $\mathsf{Approx}_{\mathcal{A},\epsilon/4}$ and $\mathsf{Find}_{\mathcal{A},\epsilon/4}$ called by $\mathcal{S}_{\mathcal{A},\epsilon}(N)$ succeed. The probability that the above assumption holds is overwhelming since the number of executions is polynomial and each execution succeeds with overwhelming probability. If $\mathcal{S}_{\mathcal{A},\epsilon}$ outputs some $I = \{p_1, \ldots, p_m\}$, then we have $\Pr[1 \leftarrow \mathcal{A}(N, g) : g \overset{\$}{\leftarrow} \mathbb{QR}_N] -$

$\Pr[1 \leftarrow \mathcal{A}(N, g^{p_1 \cdots p_m}) : g \xleftarrow{\$} \mathbb{QR}_N]| < \epsilon' + \epsilon/4 < \epsilon/2 + \epsilon/4 = 3\epsilon/4$. Next, we prove that for overwhelming fraction of N generated by IGen, the probability that $\mathcal{S}_{\mathcal{A},\epsilon}(N)$ outputs $\perp$ is negligible. First, we prove that in each repeat, $\tilde{p}$ that is removed from L divides $\Phi(N)$. We let $\epsilon' \leftarrow \mathsf{Approx}_{\mathcal{A},\epsilon/4}(1^\lambda, \mathcal{D}_0, \mathcal{D}_m)$. If $\epsilon' \geq \epsilon/2$, then we have $|\Pr[1 \leftarrow \mathcal{A}(N, g) : g \xleftarrow{\$} \mathbb{QR}_N] - \Pr[1 \leftarrow \mathcal{A}(N, g^{p_1 \cdots p_m}) : g \xleftarrow{\$} \mathbb{QR}_N]| > \epsilon' - \epsilon/4 \geq \epsilon/2 - \epsilon/4 = \epsilon/4$. Therefore $\mathsf{Find}_{\mathcal{A},\epsilon/4}(N, I)$ outputs $p_j \in I$ that divides $\Phi(N)$ since it succeeds. Thus if $\mathcal{S}_{\mathcal{A},\epsilon}(N)$ outputs $\perp$, then one ℓ_B-bit prime factor of $\Phi(N)$ is removed from L in each repeat, and the repeat is done $M_{\ell_B} - m + 1$ times. Therefore throughout the execution of $\mathcal{S}_{\mathcal{A},\epsilon}(N)$, $M_{\ell_B} - m + 1$ distinct ℓ_B-bit prime factors of $\Phi(N)$ are removed from L. If we let e be the product of these primes, then we have $e > (2^{\ell_B - 1})^{M_{\ell_B} - m + 1} \geq 2^{(1/2+c)\ell_N} > N^{1/2+c}$ and $e | \Phi(N)$. By Lemma 18, we can factorize N efficiently by using e. Therefore for overwhelming fraction of N generated by IGen, the probability that $\mathcal{S}_{\mathcal{A},\epsilon}(N)$ outputs $\perp$ is negligible under the factoring assumption. Therefore for overwhelming fraction of N generated by IGen, we have $\Pr[1 \leftarrow \mathcal{A}(N, g) : g \xleftarrow{\$} \mathbb{QR}_N] - \Pr[1 \leftarrow \mathcal{A}(N, g^{p_1 \cdots p_m}) : \{p_1, \ldots, p_m\} \leftarrow \mathcal{S}_{\mathcal{A},\epsilon}(N); g \xleftarrow{\$} \mathbb{QR}_N]| < 3\epsilon/4$ with overwhelming probability over the randomness of $\mathcal{S}_{\mathcal{A},\epsilon}$. Since ϵ is noticeable, by the averaging argument, $\Pr[1 \leftarrow \mathcal{A}(N, g) : (N, P, Q) \leftarrow \mathsf{IGen}(1^\lambda); g \xleftarrow{\$} \mathbb{QR}_N] - \Pr[1 \leftarrow \mathcal{A}(N, g^{p_1 \cdots p_m}) : (N, P, Q) \leftarrow \mathsf{IGen}(1^\lambda); g \xleftarrow{\$} \mathbb{QR}_N; \{p_1, \ldots, p_m\} \leftarrow \mathcal{S}_{\mathcal{A},\epsilon}(N)] \leq \epsilon(\lambda)$ holds for sufficiently large λ. □

4.5 Adversary-dependent Lossy Trapdoor Function

In this section, we define ad-LTDFs. Then we give a construction of an ad-LTDF based on the m-ad-DRSA assumption, which can be reduced to the factoring assumption by Theorem 11.

4.5.1 Definition

Here we define ad-LTDFs. Intuitively, ad-LTDFs are defined by weakening LTDFs so that descriptions of lossy functions that cannot be distinguished from those of injective functions may depend on a specific distinguisher. Namely, the algorithm that generates lossy functions takes a "lossy function index" I as well as a public parameter as input, and we require that for any PPT algorithm $\mathcal{A}$, there exists at least one I such that $\mathcal{A}$ does not distinguish lossy functions generated with index I from injective functions. Moreover, we require that such I can be efficiently computed given $\mathcal{A}$. The precise definition is as follows. For integers n and k such that $0 < k < n$, an (n, k)-ad LTDF consists of 5 algorithms ($\mathsf{ParamsGen}, \mathsf{SampleInj}, \mathsf{SampleLossy}, \mathsf{Evaluation}, \mathsf{Inversion}$) with a family $\{\mathcal{I}(\lambda)\}_{\lambda \in \mathbb{N}}$ of lossy function index sets.

$\mathsf{ParamsGen}(1^\lambda) \to (PP, SP)$: It takes a security parameter 1^λ as input, and outputs a public parameter PP and a secret parameter SP.

$\mathsf{SampleInj}(PP) \to \sigma$: It takes a public parameter PP as input, and outputs a function description σ, which specifies an injective function f_σ over the domain $\{0,1\}^n$.

$\mathsf{SampleLossy}(PP, I) \to \sigma$: It takes a public parameter PP and a lossy function index $I \in \mathcal{I}(\lambda)$ as input, and outputs a function index σ, which specifies a "lossy" function f_σ over the domain $\{0,1\}^n$.

$\mathsf{Evaluation}(PP, \sigma, x) \to f_\sigma(x)$: It takes a public parameter PP, function description σ and $x \in \{0,1\}^n$ as input, and outputs $f_\sigma(x)$

$\mathsf{Inversion}(SP, \sigma, y) \to f_\sigma^{-1}(y)$: It takes a secret parameter SP, a function description σ and y and outputs $f_\sigma^{-1}(y)$.

We require ad-LTDFs to satisfy the following three properties.

Correctness: For all $x \in \{0,1\}^n$, we have $\mathsf{Inversion}(SP, \sigma, \mathsf{Evaluation}(PP, \sigma, x)) = x$ with overwhelming probability where $(PP, SP) \leftarrow \mathsf{ParamsGen}(1^\lambda)$ and $\sigma \leftarrow \mathsf{SampleInj}(PP)$.

Lossiness: For all $\lambda \in \mathbb{N}$, $(PP, SP) \leftarrow \mathsf{ParamsGen}(1^\lambda)$, $I \in \mathcal{I}(\lambda)$ and $\sigma \leftarrow \mathsf{SampleLossy}(PP, I)$, the image of f_σ has size at most 2^{n-k}.

Indistinguishability between injective and lossy functions. Intuitively, we require that for any PPT adversary $\mathcal{A}$, there exists at least one lossy function index $I \in \mathcal{I}(\lambda)$ such that $\mathcal{A}$ cannot distinguish an injective function from a lossy function with the lossy function index I.

The more precise definition is as follows. For any PPT adversary $\mathcal{A}$ and noticeable function $\epsilon(\lambda)$, there exists a PPT algorithm $\mathcal{S}_{\mathcal{A},\epsilon}$ that takes a public parameter PP as input and outputs $I \in \mathcal{I}(\lambda)$ such that the following is satisfied. If we let

$$P_{\mathsf{inj}} := \Pr\left[1 \leftarrow \mathcal{A}(PP, \sigma) : \begin{array}{c} (PP, SP) \leftarrow \mathsf{ParamsGen}(1^\lambda) \\ \sigma \leftarrow \mathsf{SampleInj}(PP) \end{array}\right]$$

$$P_{\mathsf{lossy}} := \Pr\left[1 \leftarrow \mathcal{A}(PP, \sigma) : \begin{array}{c} (PP, SP) \leftarrow \mathsf{ParamsGen}(1^\lambda) \\ I \leftarrow \mathcal{S}_{\mathcal{A},\epsilon}(PP) \\ \sigma \leftarrow \mathsf{SampleLossy}(PP, I) \end{array}\right]$$

then we have $|P_{\mathsf{inj}} - P_{\mathsf{lossy}}| \leq \epsilon(\lambda)$ for sufficiently large λ.

As mentioned in Remark 12, though ϵ must be noticeable in the above definition, ad-LTDFs can be used for many cryptographic applications. This is because ϵ can be set depending on the advantage of an adversary in security reductions.

Remark 15. *Besides what is explained above, there is a minor difference between the definition of ad-LTDFs and that of LTDFs. In the definition of ad-LTDFs,* ParamsGen *is explicitly separated from* SampleInj *or* SampleLossy, *whereas there is no separation between them in the definition of LTDFs [PW08]. This is only for simplifying the presentation, and there is no significant difference here.*

4.5.2 Construction

We construct an ad-LTDF based on the m-ad-DRSA assumption. Let IGen be an algorithm that generates an ℓ_N-bit (ℓ_B, t_p, t_q)-SS RSA modulus with the parameter given in Sec. 4.3 and $n := (t - d)(\ell_B - 1)$.

Definition of $\mathcal{I}(\lambda)$: $\mathcal{I}(\lambda)$ is defined as the set of all m-tuple of distinct primes of length ℓ_B. That is, we define $\mathcal{I}(\lambda) := \{\{p_1, \ldots, p_m\} : p_1, \ldots, p_m$ are distinct ℓ_B bit primes$\}$.

$\mathsf{ParamsGen}(1^\lambda) \rightarrow (PP, SP)$: Generate $(N, P, Q) \leftarrow \mathsf{IGen}(1^\lambda)$, set $PP := N$ and $SP := (P, Q)$, and output (PP, SP).

$\mathsf{SampleInj}(PP = N) \rightarrow \sigma$: Choose $g \xleftarrow{\$} \mathbb{QR}_N$ and output $\sigma := g$.

$\mathsf{SampleLossy}(PP = N, I = \{p_1, \ldots, p_m\}) \rightarrow \sigma$: Choose $g \xleftarrow{\$} \mathbb{QR}_N$ and output $\sigma := g^{p_1 \cdots p_m}$.

$\mathsf{Evaluation}(PP = N, \sigma = g, x \in \{0,1\}^n) \rightarrow f_\sigma(x)$: Interpret x as an element of $[2^n]$ and output g^x.

$\mathsf{Inversion}(SP = (P, Q), \sigma = g, y) \rightarrow f_\sigma^{-1}(y)$: Compute $x = \mathsf{PLog}(P, Q, g, y)$ and output x where PLog is the algorithm given in Lemma 17.

Theorem 12. *If the m-ad-DRSA assumption holds with respect to* IGen*, then the above scheme is an* $(n, n - (\ell_{p'} + \ell_{q'} + (M_{\ell_B} - m)\ell_B))$*-ad-LTDF.*

Then the following corollary follows by combining the above theorem and Theorem 11.

Corollary 1. *If the factoring assumption holds with respect to* IGen *for the parameter setting given in Sec. 4.3, then there exists an ad-LTDF.*

Proof. (of Corollary 1.) Recall that we set $\ell_{p'} = \ell_{q'} = O(\lambda)$, $\ell_B = \lfloor 4 \log \lambda \rfloor$, $t_p = t_q = O(\lambda^3 / \log \lambda)$ (then we have $\ell_N \approx \ell_{p'} + \ell_{q'} + t\ell_B = O(\lambda^3)$) and $d := t/4$. We let $m := \lfloor M_{\ell_B} + 1 - (1/2 + c)\frac{\ell_N}{(\ell_B - 1)} \rfloor$ for a constant $c < 1/4$. Then we have $(M_{\ell_B} - m + 1)(\ell_B - 1) \geq (1/2 + c)\ell_N$ and therefore the m-ad-DRSA assumption holds under the factoring assumption by Theorem 11. Then we prove that the above ad-LTDF for this m is non-trivial, i.e., we have $n - (\ell_{p'} + \ell_{q'} + (M_{\ell_B} - m)\ell_B) > 0$. Since we have $m \approx M_{\ell_B} - (1/2 + c)\frac{\ell_N}{(\ell_B - 1)}$, we have $n - (\ell_{p'} + \ell_{q'} + (M_{\ell_B} - m)\ell_B) \approx (t - d)\ell_B - (\ell_{p'} + \ell_{q'} + (1/2 + c)\frac{\ell_N \ell_B}{(\ell_B - 1)}) \approx (1/4 - c)t\ell_B - (3/2 + c)(\ell_{p'} + \ell_{q'}) > 0$ for sufficiently large λ since $t\ell_B = O(\lambda^3)$ and $\ell_{p'} + \ell_{q'} = O(\lambda)$. Thus the obtained ad-LTDF for this m is non-trivial. □

Remark 16. *If we set $\ell_{p'} = \ell_{q'} = 160$, $\ell_B = 15$, $t = 64$, $d = 7$ and $\ell_N = 2420$ as given in Sec. 4.3, and $c = 1/20$ then by setting $m := \lfloor M_{\ell_B} + 1 - (1/2 + c)\frac{\ell_N}{(\ell_B - 1)} \rfloor$, the obtained scheme is a $(1848, 103)$-ad-LTDF. If better lossiness is required, then one may set t larger (as long as factorizing N is hard).*

Then we prove Theorem 12.

Proof. (of Theorem 12)
Correctness. If g is generated by $\mathsf{SampleInj}$, then it is a random element of $\mathbb{QR}_N$. Thus $\mathsf{Inversion}((P, Q), g, \mathsf{Evaluation}(N, \sigma, x)) = \mathsf{Inversion}((P, Q), g, g^x) = x$ holds by the correctness of PLog given in Lemma 17.

Lossiness. Next, we prove that the above construction satisfies $(n, n - (\ell_{p'} + \ell_{q'} + (M_{\ell_B} - m)\ell_B))$-lossiness. Let σ be a function description generated by $\mathsf{SampleLossy}(N, I = \{p_1, \ldots, p_m\})$. What we should prove is that the image size of f_σ is at most $2^{\ell_{p'} + \ell_{q'} + (M_{\ell_B} - m)\ell_B}$. There exists $g \in \mathbb{QR}_N$ such that $\sigma = g^{p_1 \cdots p_m}$, and thus any output of f_σ is an element of the group $S := \{h^{p_1 \cdots p_m} : h \in \mathbb{QR}_N\}$. We consider the order of S. S is a subgroup of $\mathbb{QR}_N = G \times G^\perp$ and $p_1 \ldots p_m$ is coprime to $\mathrm{ord}(G) = p'q'$. Therefore there exists a subgroup $S^\perp$ of $G^\perp$ such that $S = G \times S^\perp$. We can see that $\mathrm{ord}(S^\perp)$ is the product of some distinct ℓ_B-bit primes and coprime to $p_1 \dots p_m$ by the definition. Therefore that is the product of at most $M_{\ell_B} - m$ such primes, and can be bounded by $2^{(M_{\ell_B} - m)\ell_B}$. Therefore the order of S is at most $2^{\ell_{p'} + \ell_{q'} + (M_{\ell_B} - m)\ell_B}$.

Indistinguishability between injective and lossy functions. This immediately follows from the m-ad-DRSA assumption. Indeed, clearly we have $P_{\mathsf{inj}} = P_0$ and $P_{\mathsf{lossy}} = P_1$ where P_0 and P_1 are defined in Def. 14, and the m-ad-DRSA assumption requires $|P_0 - P_1| < \epsilon(\lambda)$ for sufficiently large λ.

□

4.6 Adversary-dependent All-but-one Lossy Trapdoor Function.

In this section, we define adversary-dependent all-but-one lossy trapdoor functions (ad-ABO) and construct it based on ad-LTDFs. Moreover we give more efficient construction of ad-ABO based on the ad-DRSA assumption.

4.6.1 Definition

For integers n and k such that $0 < k < n$, an (n, k)-adversary-dependent all-but-one lossy trapdoor function (ad-ABO) consists of 5 algorithms ($\mathsf{ParamsGen}, \mathsf{SampleInj}$, $\mathsf{SampleABO}, \mathsf{Evaluation}, \mathsf{Inversion}$) and a family $\{\mathcal{I}(\lambda)\}_{\lambda \in \mathbb{N}}$ of lossy function index sets.

$\mathsf{ParamsGen}(1^\lambda) \to (PP, SP)$: It takes a security parameter 1^λ as input, and outputs a public parameter PP and a secret parameter SP.

$\mathsf{SampleInj}(PP) \to \sigma$: It takes a public parameter PP as input, and outputs a function description σ, which specifies an injective function f_σ over the domain $\{0,1\}^n \times \{0,1\}^{\ell_b}$.

$\mathsf{SampleABO}(PP, b^*, I) \to \sigma$: It takes a public parameter PP, a lossy branch $b \in \{0,1\}^{\ell_b}$ and an all-but-one function index $I \in \mathcal{I}(\lambda)$ as input, and outputs a function index σ, which specifies a "all-but-one" function f_σ over the domain $\{0,1\}^n \times \{0,1\}^{\ell_b}$.

$\mathsf{Evaluation}(PP, \sigma, b, x) \to f_\sigma(x, b)$: It takes a public parameter PP, function description σ, a branch b and $x \in \{0,1\}^n$ as input, and outputs $f_\sigma(x, b)$

$\mathsf{Inversion}(SP, \sigma, b^*, b, y) \to x$: It takes a secret parameter SP, a function description σ,, $b^* \in \{0,1\}^{\ell_b} \cup \perp$, b and y and outputs the "inversion" x.

We require ad-LTDFs to satisfy the following three properties.
Correctness:

1. For all $x \in \{0,1\}^n$ and $b \in \{0,1\}^{\ell_b}$, we have $\mathsf{Inversion}(SP, \sigma, \perp, b, \mathsf{Evaluation}(PP, \sigma, b, x)) = x$ with overwhelming probability where $(PP, SP) \leftarrow \mathsf{ParamsGen}(1^\lambda)$ and $\sigma \leftarrow \mathsf{SampleInj}(PP)$.

2. For all $x \in \{0,1\}^n$, $b^*, b \in \{0,1\}^{\ell_b}$ with $b \neq b^*$ and $I \in \mathcal{I}(\lambda)$, we have $\mathsf{Inversion}(SP, \sigma, b^*, b, \mathsf{Evaluation}(PP, \sigma, b, x)) = x$ with overwhelming probability where $(PP, SP) \leftarrow \mathsf{ParamsGen}(1^\lambda)$ and $\sigma \leftarrow \mathsf{SampleABO}(PP, b^*, I)$.

All-but-one lossiness: For all $\lambda \in \mathbb{N}$, $(PP, SP) \leftarrow \mathsf{ParamsGen}(1^\lambda)$, $b^* \in B(\lambda)$, $I \in \mathcal{I}(\lambda)$ and $\sigma \leftarrow \mathsf{SampleABO}(PP, b^*, I)$, the image of $f_\sigma(\cdot, b^*)$ has size at most 2^{n-k}.

Indistinguishability between injective and ABO functions. Intuitively, we require that for any PPT adversary $\mathcal{A}$, there exists at least one lossy function index $I \in \mathcal{I}(\lambda)$ such that for all $b^* \in B(\lambda)$, $\mathcal{A}$ cannot distinguish an injective function from an ABO function with the lossy branch b^* and the lossy function index I.

The more precise definition is as follows. For any PPT adversary $\mathcal{A}$ and noticeable function $\epsilon(\lambda)$, there exists a PPT algorithm $\mathcal{S}^{\mathsf{abo}}_{\mathcal{A},\epsilon}$ that takes a public parameter PP and a lossy branch b^* as input and outputs $I \in \mathcal{I}(\lambda)$ such that the following is satisfied. For all $b^* \in B(\lambda)$, if we let

$$P_{\mathsf{inj}} := \Pr\left[1 \leftarrow \mathcal{A}(PP, \sigma) : \begin{array}{c} (PP, SP) \leftarrow \mathsf{ParamsGen}(1^\lambda) \\ \sigma \leftarrow \mathsf{SampleInj}(PP) \end{array}\right]$$

$$P_{\mathsf{abo},b^*} := \Pr\left[1 \leftarrow \mathcal{A}(PP, \sigma) : \begin{array}{c} (PP, SP) \leftarrow \mathsf{ParamsGen}(1^\lambda) \\ I \leftarrow \mathcal{S}_{\mathcal{A},\epsilon,b^*}(PP) \\ \sigma \leftarrow \mathsf{SampleLossy}(PP, b^*, I) \end{array}\right]$$

then we have $|P_{\mathsf{inj}} - P_{\mathsf{abo},b^*}| \leq \epsilon(\lambda)$ for sufficiently large λ.

4.6.2 Generic Construction

Here, We construct an ad-ABO based on an ad-LTDF. Let $(\mathsf{ParamsGen}_{\mathsf{gltdf}}, \mathsf{SampleInj}_{\mathsf{gltdf}}, \mathsf{SampleLossy}_{\mathsf{gltdf}}, \mathsf{Evaluation}_{\mathsf{gltdf}}, \mathsf{Inversion}_{\mathsf{gltdf}})$ be an (n, k)- ad-LTDF

Definition of $\mathcal{I}(\lambda)$: $\mathcal{I}(\lambda)$ is the same as that of the underlying ad-LTDF.
$\mathsf{ParamsGen}_{\mathsf{gabo}}(1^\lambda) \rightarrow (PP, SP)$: Run $(PP, SP) \leftarrow \mathsf{ParamsGen}_{\mathsf{gltdf}}(1^\lambda)$ and output

(PP, SP).

$\mathsf{SampleInj}_{\mathsf{gabo}}(PP) \to \sigma$: For all $i \in [\ell_b]$, generate $\sigma_{i,0}, \sigma_{i,1} \leftarrow \mathsf{SampleInj}_{\mathsf{gltdf}}(PP)$, and output $\sigma := \{\sigma_{i,j}\}_{i\in[\ell_b], j\in\{0,1\}}$.

$\mathsf{SampleABO}_{\mathsf{gabo}}(PP, b^*, I) \to \sigma$: For all $i \in [\ell_b]$, generate $\sigma_{i,b_i^*} \leftarrow \mathsf{SampleLossy}_{\mathsf{gltdf}}$ (PP, I) and $\sigma_{i,1-b_i^*} \leftarrow \mathsf{SampleInj}_{\mathsf{gltdf}}(PP)$, and output $\sigma := \{\sigma_{i,j}\}_{i\in[\ell_b], j\in\{0,1\}}$.

$\mathsf{Evaluation}_{\mathsf{gabo}}(PP, \sigma, b, x) \to f_\sigma(x, b)$: For all $i \in [\ell_b]$, compute $y_i := \mathsf{Evaluation}_{\mathsf{gltdf}}$ (PP, σ_{i,b_i}, x), and output $y := \{y_i\}_{i\in[\ell_b]}$.

$\mathsf{Inversion}_{\mathsf{gabo}}(SP, \sigma, b^*, b, y) \to f_\sigma^{-1}(y)$: Find $i \in [\ell_b]$ such that $b_i^* \neq b_i$, compute $x := \mathsf{Inversion}_{\mathsf{ltdf}}(SP, \sigma_{i,b_i}, y_i)$ and output x.

Theorem 13. *If the underlying scheme is an $(n, n-r)$-ad-LTDF, then the above scheme is an $(n, n - r\ell_B)$-ad-ABO.*

The proof is almost the same as the proof of generic construction of all-but-one lossy trapdoor function from a lossy trapdoor function in [PW08]. Therefore we omit it.

4.6.3 Direct Construction

Here, we construct an ad-ABO based on the m-ad-DRSA assumption directly. Let IGen be an algorithm that generates an ℓ_N-bit (ℓ_B, t_p, t_q)-SS RSA modulus with the parameter given in Sec. 4.3 and $n := \frac{(t-d)(\ell_B-1)-\ell_b}{2}$.

Definition of $\mathcal{I}(\lambda)$: $\mathcal{I}(\lambda)$ is defined as the set of all m-tuple of distinct primes of length ℓ_B. That is, we define $\mathcal{I}(\lambda) := \{\{p_1, \ldots, p_m\} : p_1, \ldots, p_m$ are distinct ℓ_B bit primes$\}$.

$\mathsf{ParamsGen}(1^\lambda) \to (PP, SP)$: Generate $(N, P, Q) \leftarrow \mathsf{IGen}(1^\lambda)$, set $PP := N$ and $SP := (P, Q)$, and output (PP, SP).

$\mathsf{SampleInj}(PP = N) \to \sigma$: Choose $g, h \xleftarrow{\$} \mathbb{QR}_N$ and output $\sigma := (g, h)$.

$\mathsf{SampleABO}(PP = N, b^*, I = \{p_1, \ldots, p_m\}) \to \sigma$: Choose $g, g' \xleftarrow{\$} \mathbb{QR}_N$, set $h := g^{-b^*} g'^{p_1 \cdots p_m}$ and output $\sigma := (g, h)$.

$\mathsf{Evaluation}(PP = N, \sigma = (g, h), b, x \in \{0,1\}^n) \to f_\sigma(x)$: Interpret x as an element of $[2^n]$ and output $(g^b h)^x$.

$\mathsf{Inversion}(SP = (P, Q), \sigma = (g, h), b^*, by) \to f_\sigma^{-1}(y)$: Compute $x = \mathsf{PLog}(P, Q, g^b h, y)$ and output x where PLog is the algorithm given in Lemma 17.

Theorem 14. *If the m-ad-DRSA assumption holds with respect to* IGen*, then the above scheme is an $(n, n - (\ell_{p'} + \ell_{q'} + (M_{\ell_B} - m)\ell_B))$-ad-ABO.*

Proof.

Correctness.

1. If $\sigma = (g,h)$ is generated by $\mathsf{SampleInj}$, then they are independently uniform elements of $\mathbb{QR}_N$. In particular, for any $b \in \{0,1\}^{\ell_b}$, $g^b h$ is a uniform element of $\mathbb{QR}_N$. Thus $\mathsf{Inversion}((P,Q),(g,h),\perp,b,\mathsf{Evaluation}(N,\sigma,b,x)) = \mathsf{Inversion}((P,Q),g^b h,(g^b h)^x) = x$ holds by the correctness of PLog given in Lemma 17.

2. Let $(N,(P,Q)) \leftarrow \mathsf{ParamsGen}$ and $\sigma = (g,h) \leftarrow \mathsf{SampleABO}(N,b^*,I = \{p_1,\ldots,p_m\})$ for some $b^* \in \{0,1\}^{\ell_b}$. Then g is a uniform element of $\mathbb{QR}_N$ and $h = g^{-b^*} g'^{p_1\cdots p_m}$ for $g' \xleftarrow{\$} \mathbb{QR}_N$. Then for any $b \neq b^*$, we have $g^b h = g^{b-b^*} g'^{p_1\cdots p_m}$. By Lemma 15, $\mathrm{ord}(g)$ has at least $t-d$ distinct ℓ_b-bit prime divisors. The number of these prime divisors that are not coprime to $b-b^*$ is at most $\log_{2^{\ell_B-1}}(b-b^*) \leq \ell_b/(\ell_B-1)$. We write $p'_1,\ldots,p'_s$ to denote the all ℓ_b-bit divisors of $\mathrm{ord}(g)$ that are coprime to $b-b^*$. Then we have $s \geq t-d-\ell_b/(\ell_B-1)$. For each p'_i, the probability that $\mathrm{ord}(g^b h)$ is comprime to p'_i is $1/p'_i \leq 2^{-\ell_B+1}$ and they are all independent. Then one can see that there exists more than $s/2$ p'_i such that $p'_i|\mathrm{ord}(g^b h)$ by similar analysis as in [Gro05, Lemma3]. Therefore one can recover x from $(g^b h)^x$ as long as $x \leq 2^{\frac{(t-d)(\ell_B-1)-\ell_b}{2}} \leq (2^{\ell_B-1})^{s/2}$.

All-but-one lossiness. Next, we prove that the above construction satisfies the all-but-one lossiness property. Let $\sigma = (g,h)$ be a function description generated by $\mathsf{SampleLossy}(N, I = \{p_1,\ldots,p_m\})$. Then there exists $g' \in \mathbb{QR}_N$ such that $h = g^{-b^*} g'^{p_1\cdots p_m}$. In particular, we have $g^{b^*} h = g'^{p_1\cdots p_m}$. Then any output of $f_\sigma(\cdot,b^*)$ is an element of the group $S := \{h'^{p_1\cdots p_m} : h' \in \mathbb{QR}_N\}$. We consider the order of S. S is a subgroup of $\mathbb{QR}_N = G \times G^\perp$ and $p_1 \ldots p_m$ is coprime to $\mathrm{ord}(G) = p'q'$. Therefore there exists a subgroup $S^\perp$ of $G^\perp$ such that $S = G \times S^\perp$. We can see that $\mathrm{ord}(S^\perp)$ is the product of some distinct ℓ_B-bit primes and coprime to $p_1 \ldots p_m$ by the definition. Therefore that is the product of at most $M_{\ell_B} - m$ such primes, and can be bounded by $2^{(M_{\ell_B}-m)\ell_B}$. Therefore the order of S is at most $2^{(\ell_{p'}+\ell_{q'})(M_{\ell_B}-m)\ell_B}$.

Indistinguishability between injective and all-but-one lossy functions. This follows from the m-ad-DRSA assumption. Actually, for any PPT adversary $\mathcal{A}'$ that tries to distinguish these functions and any $b^* \in \{0,1\}^{\ell_b}$, we consider the following adversary $\mathcal{A}'_{b^*}$ against the ad-DRSA assumption.

$\mathcal{A}'_{b^*}(N,\bar{g})$: Generate $g \xleftarrow{\$} \mathbb{QR}_N$, set $h := g^{b^*}\bar{g}$, run $\mathcal{A}(N,(g,h))$ and output as $\mathcal{A}$

outputs.

For any noticeable ϵ, let $\mathcal{S}^{\mathsf{gdrsa}}_{\mathcal{A}_{b^*},\epsilon}$ be the algorithm that is assumed to exist in Definition 14. If we let

$$P_0 := \Pr\left[1 \leftarrow \mathcal{A}(N,g) : \begin{array}{c}(N,P,Q) \leftarrow \mathsf{IGen}(1^\lambda)\\ g \xleftarrow{\$} \mathbb{QR}_N\end{array}\right]$$

$$P_1 := \Pr\left[1 \leftarrow \mathcal{A}(N,g^{p_1 \cdots p_m}) : \begin{array}{c}(N,P,Q) \leftarrow \mathsf{IGen}(1^\lambda)\\ g \xleftarrow{\$} \mathbb{QR}_N\\ \{p_1,\dots,p_m\} \leftarrow \mathcal{S}^{\mathsf{gdrsa}}_{\mathcal{A}_{b^*},\epsilon}(N)\end{array}\right]$$

then we have $|P_0 - P_1| \leq \epsilon(\lambda)$ for sufficiently large λ by the definition of the ad-DRSA assumption.

We let $\mathcal{S}^{\mathsf{abo}}_{\mathcal{A},\epsilon,b^*} := \mathcal{S}^{\mathsf{gdrsa}}_{\mathcal{A}^*_b,\epsilon}$. In the following, we show that it works well. If $\bar{g}$ is a uniform element of $\mathbb{QR}_N$, then h is a uniform on $\mathbb{QR}_N$ and independent of g. Thus in this case, $\mathcal{A}'_{b^*}$ simulates the environment for $\mathcal{A}$ where it is given an injective function. On the other hand, if $\bar{g}$ is generated as $\bar{g} := \bar{g}'^{p_1 \cdots p_m}$ where $\{p_1,\dots,p_m\} \leftarrow \mathcal{S}^{\mathsf{gdrsa}}_{\mathcal{A}^*_b,\epsilon}$, then $\mathcal{A}'_{b^*}$ simulates the environment for $\mathcal{A}$ where it is given an all-but-one function with lossy branch b^* and index $I := \{p_1,\dots,p_m\}$.

Therefore if we let

$$P_{\mathsf{inj}} := \Pr\left[1 \leftarrow \mathcal{A}(PP,\sigma) : \begin{array}{c}(PP,SP) \leftarrow \mathsf{ParamsGen}(1^\lambda)\\ \sigma \leftarrow \mathsf{SampleInj}(PP)\end{array}\right]$$

$$P_{\mathsf{abo},b^*} := \Pr\left[1 \leftarrow \mathcal{A}(PP,\sigma) : \begin{array}{c}(PP,SP) \leftarrow \mathsf{ParamsGen}(1^\lambda)\\ I \leftarrow \mathcal{S}_{\mathcal{A},\epsilon,b^*}(PP)\\ \sigma \leftarrow \mathsf{SampleLossy}(PP,b^*,I)\end{array}\right]$$

then we have $P_{\mathsf{inj}} = P_0$ and $P_{\mathsf{abo},b^*} = P_1$. Therefore we have $|P_{\mathsf{inj}} - P_{\mathsf{abo},b^*}| \leq \epsilon(\lambda)$ for sufficiently large λ as required. □

4.7 Applications

Here we discuss applications of ad-LTDFs. As mentioned before, ad-LTDFs can replace LTDFs in many applications. Informally, ad-LTDFs can replace LTDFs if a lossy function is used only in the security proof and not used in the real protocol. In such cases, a lossy function may depend on an adversary that tries to distinguish it from an injective function since an adversary is firstly fixed in security proofs. As a result, we can immediately obtain a collision resistant hash function [PW08], a CPA secure PKE scheme [PW08] and a DPKE scheme [BFO08] based on ad-LTDFs by simply replacing LTDFs by ad-LTDFs. Among them, by using our ad-LTDF based on the factoring assumption given in Sec. 4.5, we obtain the first DPKE scheme that satisfies the PRIV security for block-sources defined in [BFO08] under the factoring assumption.

4.7.1 Collision Resistant Hash Function

Here, we give an analogue of the collision resistant hash function in [PW08] based on ad-LTDFs. In fact, our scheme is obtained by simply replacing LTDFs in the scheme in [PW08] by ad-LTDFs. The concrete construction is as follows. Let $(\mathsf{ParamsGen}, \mathsf{SampleInj}, \mathsf{SampleLossy}, \mathsf{Evaluation}, \mathsf{Inversion})$ be an (n,k)-ad-LTDF and $\mathcal{H}$ be a family of pairwise independent hash functions from $\{0,1\}^k$ to $\{0,1\}^{\kappa n}$ where $\kappa := 2\rho + \delta$, $\rho < 1/2$ is a constant that satisfies $n - k \leq \rho n$ and δ is some constant in $(0, 1-2\rho)$,

$\mathsf{Gen}_{\mathsf{crh}}(1^\lambda)$: Run $(PP, SP) \leftarrow \mathsf{ParamsGen}(1^\lambda)$ and $\sigma \leftarrow \mathsf{SampleInj}(PP)$, and choose $H \overset{\$}{\leftarrow} \mathcal{H}$. Output a function description $h := (H, PP, \sigma)$.

$\mathsf{Eval}_{\mathsf{crh}}((H, PP, \sigma), x)$: Compute $H(\mathsf{Evaluation}(PP, \sigma, x))$ and output it.

Theorem 15. *The above hash function is collision resistant.*

We omit the proof since this can be proven by modifying the proof in [PW08] in a similar way as in Sec. 4.7.3.

4.7.2 CPA Secure Public Key Encryption

Here, we give an analogue of the CPA secure PKE scheme in [PW08] based on ad-LTDFs. In fact, our scheme is obtained by simply replacing LTDFs in the scheme in [PW08] by LTDFs. The concrete construction is as follows.

Let $(\mathsf{ParamsGen}, \mathsf{SampleInj}, \mathsf{SampleLossy}, \mathsf{Evaluation}, \mathsf{Inversion})$ be an (n,k)-ad-LTDF and $\mathcal{H}$ be a family of pairwise independent hash functions from $\{0,1\}^n$ to $\{0,1\}^\ell$, where $\ell \leq k - 2\log(1/\delta)$ for some negligible δ. The construction of our scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is as follows.

Key generation : $\mathsf{Gen}(1^\lambda)$ generates $(PP, SP) \leftarrow \mathsf{ParamsGen}(1^\lambda)$ and $\sigma \leftarrow \mathsf{SampleInj}(PP)$. It also chooses a hash function $H \overset{\$}{\leftarrow} \mathcal{H}$. It outputs a public key $PK = (PP, \sigma, H)$ and a secret key $SK = (SP, H)$.

Encryption : Enc takes as input a public key $PK = (PP, \sigma, H)$ and a message $msg \in \{0,1\}^{\ell}$. It chooses $x \xleftarrow{\$} \{0,1\}^n$, sets $C_1 := \mathsf{Evaluation}(PP, \sigma, x)$ and $C_2 := msg \oplus H(x)$ and outputs $C = (C_1, C_2)$

Decryption : Dec takes as input a secret key $SK = (SP, H)$ and a ciphertext $C = (C_1, C_2)$, computes $x := \mathsf{Inversion}(SP, \sigma, C_1)$ and $msg := C_2 \oplus H(x)$, and outputs msg.

Theorem 16. *The above scheme is CPA secure.*

Proof. Assume that the scheme is not CPA secure. Then there exists a PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ such that $\mathsf{Adv}^{\mathsf{CPA}}_{\mathcal{A},\mathsf{PKE}}(\lambda)$ is non-negligible. Then there exists a polynomial poly such that for infinitely many λ, $\mathsf{Adv}^{\mathsf{CPA}}_{\mathcal{A},\mathsf{PKE}}(\lambda) > 1/\mathsf{poly}(\lambda)$ holds. We consider the following sequence of games.

Game 1 This is the original CPA game between $\mathcal{A}$ and the challenger $\mathcal{C}$. That is, $\mathcal{C}$ generates $(PP, SP) \leftarrow \mathsf{ParamsGen}(1^{\lambda})$ and $\sigma \leftarrow \mathsf{SampleInj}(PP)$, and chooses $H \xleftarrow{\$} \mathcal{H}$. Then $\mathcal{C}$ gives $PK := (PP, \sigma, H)$, and $\mathcal{A}_1$ outputs (msg_0, msg_1, st). Then $\mathcal{C}$ chooses $b \xleftarrow{\$} \{0,1\}$ and $x \xleftarrow{\$} \{0,1\}^n$, computes $C_1 \leftarrow \mathsf{Evaluation}(PP, \sigma, x)$ and $C_2 := msg_b \oplus H(x)$ and gives $C := (C_1, C_2)$ to $\mathcal{A}_2$. Then $\mathcal{A}_2(C, st)$ outputs b'.

Game 2 This game is the same as the previous game except that σ is generated by $\mathsf{SampleLossy}(PP, I)$, where intuitively, I is an index such that “it is difficult to distinguish an injective function from a lossy function with index I for $\mathcal{A}$”. To describe this precisely, we consider the following PPT algorithm $\mathcal{B}$.

$\mathcal{B}(PP, \sigma)$: Choose $H \xleftarrow{\$} \mathcal{H}$, run $(msg_0, msg_1, st) \leftarrow \mathcal{A}_1(PP, \sigma, H)$, choose $b \xleftarrow{\$} \{0,1\}$ and $x \xleftarrow{\$} \{0,1\}^n$, compute $C_1 \leftarrow \mathsf{Evaluation}(PP, \sigma, x)$ and $C_2 := msg_b \oplus H(x)$, run $b' \leftarrow \mathcal{A}_2((C_1, C_2), st)$. If $b = b'$, then output 1, and otherwise 0.

Let $\mathcal{S}_{\mathcal{B},1/(2\mathsf{poly})}$ be the algorithm that is assumed to exist in the definition of ad-LTDFs. (Note that $\mathcal{B}$ is a PPT algorithm and $1/(2\mathsf{poly})$ is noticeable.) In this game, we let $I \leftarrow \mathcal{S}_{\mathcal{B},1/(2\mathsf{poly})}(PP)$ and $\sigma \leftarrow \mathsf{SampleLossy}(PP, I)$.

Game 3 This game is the same as the previous game except that C_2 is set as $C_2 \leftarrow msg_b \oplus U$ where $U \xleftarrow{\$} \{0,1\}^{\ell}$.

Game 4 This game is the same as the previous game except that C_2 is set as $C_2 \xleftarrow{\$} \{0,1\}^{\ell}$.

Let T_i be the event that $b = b'$ holds in Game i. By the definition, we have $|\Pr[T_1] - 1/2| = \mathsf{Adv}^{\mathsf{CPA}}_{\mathcal{A},\mathsf{PKE}}(\lambda)$. Then we prove the following lemmas.

Lemma 19. *For sufficiently large any λ, we have* $|\Pr[T_2] - \Pr[T_1]| \leq 1/(2\mathsf{poly}(\lambda))$.

Proof. By the definition of ad-LTDFs, if we let

$$P_{\mathsf{inj}} := \Pr\left[1 \leftarrow \mathcal{B}(PP, \sigma) : \begin{array}{c} (PP, SP) \leftarrow \mathsf{ParamsGen}(1^{\lambda}) \\ \sigma \leftarrow \mathsf{SampleInj}(PP) \end{array}\right]$$

$$P_{\mathsf{lossy}} := \Pr\left[1 \leftarrow \mathcal{B}(PP, \sigma) : \begin{array}{c} (PP, SP) \leftarrow \mathsf{ParamsGen}(1^{\lambda}) \\ I \leftarrow \mathcal{S}_{\mathcal{B}, 1/(2\mathsf{poly})}(PP) \\ \sigma \leftarrow \mathsf{SampleLossy}(PP, I) \end{array}\right]$$

then we have $|P_{\mathsf{inj}} - P_{\mathsf{lossy}}| \leq 1/(2\mathsf{poly}(\lambda))$ for sufficiently large λ. It is clear that $P_{\mathsf{inj}} = \Pr[T_1]$ and $P_{\mathsf{lossy}} = \Pr[T_2]$ hold. Therefore the lemma follows. □

Lemma 20. *We have* $|\Pr[T_3] - \Pr[T_2]| \leq \delta$.

In Lemma 12, we let $X := x$ and $Y := (PP, \sigma, f_{\sigma}(X))$. Then we have $\tilde{H}_{\infty}(X|Y) \geq \tilde{H}_{\infty}(X|PP, \sigma) - (n-k) = k \geq \ell + 2\log(1/\delta)$ since the size of the image of f_{σ} is at most 2^{n-k}. Then by Lemma 12, we have $\Delta((H(X), H, Y), (U, H, Y)) \leq \delta$ where $U \xleftarrow{\$} \{0,1\}^{\ell}$. Thus the lemma follows.

Lemma 21. *We have* $\Pr[T_4] = \Pr[T_3]$.

Proof. This is clear since U is independently random string. □

Lemma 22. *We have* $\Pr[T_4] = 1/2$.

Proof. In Game 3, $\mathcal{A}$ is given any information about b. Therefore the probability that $\mathcal{A}$ can correctly guess b is $1/2$. □

By combining these lemmas, for all sufficiently large λ, we have $|\Pr[T_1] - 1/2| \leq 1/(2\mathsf{poly}(\lambda)) + \delta$. That is, we have $\mathsf{Adv}^{\mathsf{CPA}}_{\mathcal{A},\mathsf{PKE}}(\lambda) \leq 1/(2\mathsf{poly}(\lambda)) + \delta$. Since we assumed for infinitely many λ, $\mathsf{Adv}^{\mathsf{CPA}}_{\mathcal{A},\mathsf{PKE}}(\lambda) > 1/\mathsf{poly}(\lambda)$, for infinitely many λ, we have $1/(2\mathsf{poly}(\lambda)) < \delta$. This contradicts to that δ is negligible. Therefore there does not exist a PPT adversary that breaks the scheme. □

Remark 17. *If we use ad-ABO, we can construct CCA secure PKE scheme similarly as in [PW08].*

4.7.3 Deterministic Public Key Encryption

Here, we construct a DPKE scheme based on ad-LTDFs. **Construction.** The construction is a simple analogue of the scheme in [BFO08] based on LTDFs. Indeed, our scheme is obtained by simply replacing LTDFs by ad-LTDFs in their scheme. The concrete construction is as follows. Let $(\mathsf{ParamsGen}, \mathsf{SampleInj}, \mathsf{SampleLossy}, \mathsf{Evaluation}, \mathsf{Inversion})$ be an (n,k)-ad-LTDF and $\mathcal{H}$ be a family of pairwise independent permutations on $\{0,1\}^n$, where $u \geq n-k+2\log(1/\delta)-2$ holds for some negligible δ. The construction of our scheme $\mathsf{DE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is as follows.

$\mathsf{Gen}(1^\lambda)$: Generate $(PP, SP) \leftarrow \mathsf{ParamsGen}(1^\lambda)$ and $\sigma \leftarrow \mathsf{SampleInj}(PP)$ and choose $H \xleftarrow{\$} \mathcal{H}$. Output a public key $PK = (PP, \sigma, H)$ and a secret key $SK = (SP, \sigma)$.

$\mathsf{Enc}(PK = (PP, \sigma, H), msg)$: Compute $C \leftarrow \mathsf{Evaluation}(PP, \sigma, H(msg))$ and output C.

$\mathsf{Dec}(SK, C)$: Compute $msg' \leftarrow \mathsf{Inversion}(SP, \sigma, C)$ and $msg := H^{-1}(msg')$ and output msg.

Theorem 17. *The above scheme is PRIV1-IND-CPA secure deterministic encryption for (u,n)-sources.*

Proof. Assume that the above scheme is not PRIV1-IND-CPA secure. There exists (u,n)-sources M_0, M_1 and a PPT adversary $\mathcal{A}$ such that $\mathsf{Adv}^{\mathsf{PRIV-IND-CPA}}_{\mathcal{A},\mathsf{DE}}(\lambda)$ is non-negligible. Then there exist a polynomial poly such that for infinitely many λ, $\mathsf{Adv}^{\mathsf{PRIV-IND-CPA}}_{\mathcal{A},\mathsf{DE}}(\lambda) > 1/\mathsf{poly}(\lambda)$ holds. We consider the following sequence of games.

Game 1 : This game is the original PRIV1-IND-CPA game with respect to M_0, M_1 and $\mathcal{A}$. That is, a challenger computes $(PP, SP) \leftarrow \mathsf{ParamsGen}(1^\lambda)$ and $\sigma \leftarrow \mathsf{SampleInj}(PP)$, chooses $H \leftarrow \mathcal{H}$, sets $PK := (PP, \sigma, H)$, chooses $b \xleftarrow{\$} \{0,1\}$, $msg^* \xleftarrow{\$} M_b$ and computes $C^* \leftarrow \mathsf{Evaluation}(PP, \sigma, H(msg^*))$. $\mathcal{A}$ is given (PK, C^*) and outputs b'.

Game 2 : This game is the same as the previous game except that σ is generated by $\mathsf{SampleLossy}(PP, I)$, where intuitively, I is an index such that "it is difficult to distinguish an injective function from a lossy function with index I for $\mathcal{A}$". To describe this precisely, we consider the following PPT algorithm $\mathcal{B}$.

$\mathcal{B}(PP, \sigma)$: Choose $H \xleftarrow{\$} \mathcal{H}$, $b \xleftarrow{\$} \{0,1\}$, $msg^* \xleftarrow{\$} M_b$, set $PK := (PP, \sigma, H)$, compute $C^* \leftarrow \mathsf{Evaluation}(PP, \sigma, H(msg^*))$, run $b' \leftarrow \mathcal{A}(PK, C^*)$ and output 1 if $b = b'$, and otherwise 0.

Let $\mathcal{S}_{\mathcal{B},1/(2\mathsf{poly})}$ be the algorithm that is assumed to exists in the definition of ad-LTDFs. (Note that $\mathcal{B}$ is a PPT algorithm and $1/(2\mathsf{poly})$ is noticeable.) In this game, we let $I \leftarrow \mathcal{S}_{\mathcal{B},1/(2\mathsf{poly})}(PP)$ and $\sigma \leftarrow \mathsf{SampleLossy}(PP, I)$.

Game 3 : This game is the same as the previous game except that a challenge ciphertext is set as $C^* \leftarrow \mathsf{Evaluation}(PP, \sigma, H(U))$ where $U \in \{0,1\}^n$ is a uniformly random string.

Let T_i be the event that $b = b'$ in Game i. Clearly we have $|\Pr[T_1] - 1/2| = \mathsf{Adv}^{\mathsf{PRIV-IND-CPA}}_{\mathcal{A},\mathsf{DE}}(\lambda)$. Then we prove the following lemmas.

Lemma 23. *For sufficiently large any λ, we have $|\Pr[T_2] - \Pr[T_1]| \leq 1/(2\mathsf{poly}(\lambda))$.*

Proof. By the definition of an adversary-dependent lossy trapdoor function, if we let

$$P_{\mathsf{inj}} := \Pr\left[1 \leftarrow \mathcal{B}(PP, \sigma) : \begin{array}{c} (PP, SP) \leftarrow \mathsf{ParamsGen}(1^\lambda) \\ \sigma \leftarrow \mathsf{SampleInj}(PP) \end{array}\right]$$

$$P_{\mathsf{lossy}} := \Pr\left[1 \leftarrow \mathcal{B}(PP, \sigma) : \begin{array}{c} (PP, SP) \leftarrow \mathsf{ParamsGen}(1^\lambda) \\ I \leftarrow \mathcal{S}_{\mathcal{B},1/(2\mathsf{poly})}(PP) \\ \sigma \leftarrow \mathsf{SampleLossy}(PP, I) \end{array}\right]$$

then we have $|P_{\mathsf{inj}} - P_{\mathsf{lossy}}| \leq 1/(2\mathsf{poly})$ for sufficiently large λ. It is clear that $P_{\mathsf{inj}} = \Pr[T_1]$ and $P_{\mathsf{lossy}} = \Pr[T_2]$ holds. Therefore the lemma follows. □

Lemma 24. *We have $|\Pr[T_3] - \Pr[T_2]| \leq \delta$.*

In Lemma 13, we let $f := \mathsf{Evaluation}(PP, \sigma, \cdot)$, $X := msg^*$ and $Y := (PP, \sigma)$. Then by the lossiness, $|S| \leq 2^{n-k}$ holds where S is the range of f. By the definition of (u, n)-sources, we have $\tilde{H}_\infty(X|Y) \geq u$ and $u \geq n - k + 2\log(1/\delta) - 2 \geq |S| + 2\log(1/\delta) - 2$. By Lemma 13, the statistical distance between $(C^*, H, (PP, \sigma))$ in Game 2 and that in Game 3 is at most δ. Thus the lemma follows.

Lemma 25. *We have $\Pr[T_3] = 1/2$.*

Proof. In Game 3, $\mathcal{A}$ is given no information about b. Therefore the probability that $\mathcal{A}$ can correctly guess b is $1/2$. □

By combining these lemmas, for all sufficiently large λ, we have $|\Pr[T_1] - 1/2| \leq 1/(2\mathsf{poly}(\lambda)) + \delta$, equivalently, $\mathsf{Adv}^{\mathsf{PRIV-IND-CPA}}_{\mathcal{A},\mathsf{DE}}(\lambda) \leq 1/(2\mathsf{poly}(\lambda)) + \delta$. On the other hand, we assumed, $\mathsf{Adv}^{\mathsf{PRIV-IND-CPA}}_{\mathcal{A},\mathsf{DE}}(\lambda) > 1/\mathsf{poly}(\lambda)$ for infinitely many λ. Combining these two inequalities, we have $1/(2\mathsf{poly}(\lambda)) < \delta$ for infinitely many λ, which contradicts to that δ is negligible. Therefore there does not exist a PPT adversary that breaks the scheme. □

Remark 18. *If we use ad-ABO given in Sec. 4.6, we can construct PRIV1-IND-CCA secure DPKE scheme similarly as in [BFO08].*

4.8 CCA Secure PKE with Short Ciphertext

In this section, we construct a CCCA secure KEM under the m-ad-DRSA assumption. By Theorem 11, under certain condition, this scheme is CCCA secure under the factoring assumption w.r.t. SS moduli. By setting a parameter appropriately, we obtain a PKE scheme whose ciphertext overhead is minimum among schemes that are CCA secure under the factoring assumption by combining our KEM and an authenticated symmetric key encryption scheme.

4.8.1 Construction

Idea of our construction. Since the m-ad-DRSA assumption is a type of subgroup decision assumptions, we can consider an "adversary-dependent version" of hash proof systems as in [CS02], where it is shown that a hash proof system can be constructed based on any subgroup decision assumption. Then we construct a KEM similarly as in [HK07], where the authors constructed a CCCA secure KEM based on a hash proof system. Though our construction is based on the above idea, for clarity, we give a direct construction of our KEM rather than defining the "adversary-dependent version" of hash proof systems.

The construction of our scheme $\mathsf{KEM}_{\mathsf{CCCA}}$ is as follows. Let IGen be a PPT algorithm that generates (ℓ_B, t_p, t_q)-SS RSA modulus, $\mathcal{H}$ be a family of pairwise independent hash functions from $(\mathbb{Z}_N^*)^n$ to $\{0,1\}^\lambda$ where $n := \lceil \frac{(2\ell_N+1)\lambda}{\ell_B-1} \rceil$, and $h : G \to \{0,1\}^\lambda$ be a target collision resistant hash function. For simplicity, we assume that the KEM key length is equal to the security parameter λ.

$\mathsf{Gen}(1^\lambda)$: Generate $(N, P, Q) \leftarrow \mathsf{IGen}(1^\lambda)$. Choose $H \xleftarrow{\$} \mathcal{H}$, $g \xleftarrow{\$} \mathbb{QR}_N$ and $x_{i,j}^{(k)} \xleftarrow{\$} [(N-1)/4]$ and set $X_{i,j}^{(k)} := g^{x_{i,j}^{(k)}}$ for $i = 1, \ldots, \lambda$, $j = 1, \ldots, n$ and $k = 0, 1$. Output $PK := (N, h, H, \{X_{i,j}^{(k)}\}_{i\in[\lambda], j\in[n], k\in\{0,1\}})$ and $SK := (\{x_{i,j}^{(k)}\}_{i\in[\lambda], j\in[n], k\in\{0,1\}}, PK)$.

$\mathsf{Enc}(PK)$: Choose $r \xleftarrow{\$} [(N-1)/4]$, compute $C := g^r$, $t := h(C)$ and $K := H((\prod_{i=1}^{\lambda} X_{i,1}^{(t_i)})^r, \ldots, (\prod_{i=1}^{\lambda} X_{i,n}^{(t_i)})^r)$ where t_i denotes the i-th bit of t. Output (C, K).

$\mathsf{Dec}(SK, C)$: Compute $t := h(C)$ and $K := H(C^{\sum_{i=1}^{\lambda} x_{i,1}^{(t_i)}}, \ldots, C^{\sum_{i=1}^{\lambda} x_{i,n}^{(t_i)}})$ where t_i denotes the i-th bit of t, and output K.

4.8.2 Security

Theorem 18. *If m-ad-DRSA assumption holds with respect to* IGen *and* $(\ell_B - 1)(t_p + t_q + m - M_{\ell_B}) \geq \lambda$ *holds, then* $\mathsf{KEM}_{\mathsf{CCCA}}$ *is CCCA secure.*

Corollary 2. *If the factoring assumption holds with respect to* IGen *for the parameter setting given in 4.3, then* $\mathsf{KEM}_{\mathsf{CCCA}}$ *is CCCA secure for* $n = O(\lambda^4/\log(\lambda))$.

Proof. (of Corollary 2) Let $m := \lfloor M_{\ell_B} + 1 - (1/2 + c)\frac{\ell_N}{(\ell_B - 1)} \rfloor$ for a constant $c < 1/4$. Then we have $(M_{\ell_B} - m + 1)(\ell_B - 1) \geq (1/2 + c)\ell_N$ and therefore the m-ad-DRSA assumption holds under the factoring assumption by Theorem 11. Moreover, we have $(\ell_B - 1)(t_p + t_q + m - M_{\ell_B}) = O(\lambda^3)$ if we use the parameter setting given in Sec. 4.3. Thus the obtained scheme is CCCA secure under the factoring assumption.

□

Theorem 18 can be proven almost similarly as the security of the CCCA secure KEM based on a hash proof system in [HK07]. However, for a technical reason, we need the following variant of the leftover hash lemma unlike in [HK07]. Specifically, in the leftover hash lemma (Lemma 12), a random variable X should be independent from H. On the other hand, in our proof, we need a variant in which a random variable X may depend on H. The following lemma states that this is possible with the loss of the number of possible random variables X. We note that this idea is already used in some existing works [TV00, RSV13]. This lemma is necessary because in our proof, we set X to be a decryption query, which is chosen by an adversary after seeing a public key which includes a pairwise independent hash function H.

Lemma 26. *Let $\mathcal{X}$ be a set of random variables X on $\{0,1\}^{n_1}$ such that $H_\infty(X) \geq n_2 + 2\log(1/\delta)$, and $\mathcal{H}$ be a family of pairwise independent hash functions from $\{0,1\}^{n_1}$ to $\{0,1\}^{n_2}$. Then for any computationally unbounded algorithm $\mathcal{F}$, which is given $H \in \mathcal{H}$ and outputs a description of a distribution $X \in \mathcal{X}$, we have $\Delta((H(X), H), (U, H)) \leq |\mathcal{X}|\delta$ where $H \xleftarrow{\$} \mathcal{H}$ and $X \leftarrow \mathcal{F}(H)$.*

Proof. We have

$$\begin{aligned}
&\Delta_{H \xleftarrow{\$} \mathcal{H}, X \leftarrow \mathcal{F}(H)}((H(X), H), (U, H)) \\
&= \mathbb{E}_{H \xleftarrow{\$} \mathcal{H}}[\Delta_{X \leftarrow \mathcal{F}(H)}(H(X), U)] \\
&\leq \mathbb{E}_{H \xleftarrow{\$} \mathcal{H}}[\sum_{X \in \mathcal{X}} \Delta(H(X), U)] \\
&= \sum_{X \in \mathcal{X}} \mathbb{E}_{H \xleftarrow{\$} \mathcal{H}}[\Delta(H(X), U)] \\
&= \sum_{X \in \mathcal{X}} \Delta_{H \xleftarrow{\$} \mathcal{H}}((H(X), H), (U, H)) \leq |\mathcal{X}|\delta
\end{aligned}$$

where the last inequality follows from Lemma 12. □

Then we give the proof of Theorem 18.

Proof. (of Theorem 18) Assume that there exists a valid PPT adversary $\mathcal{A}$ that breaks the CCCA security of the above scheme. Then there exists a polynomial poly such that $\mathsf{Adv}^{\mathsf{CCCA}}_{\mathcal{A},\mathsf{KEM}_{\mathsf{CCCA}}}(\lambda) > 1/\mathsf{poly}(\lambda)$ for infinitely many λ. We consider the following sequence of games.

Game 1 : This game is the original CCCA game of $\mathsf{KEM}_{\mathsf{CCCA}}$ for $\mathcal{A}$. That is, a challenger $\mathcal{C}$ generates $(N, P, Q) \leftarrow \mathsf{IGen}(1^\lambda)$, chooses $H \xleftarrow{\$} \mathcal{H}$, $g \xleftarrow{\$} \mathbb{QR}_N$ and $x^{(k)}_{i,j} \xleftarrow{\$} [(N-1)/4]$ and sets $X^{(k)}_{i,j} := g^{x^{(k)}_{i,j}}$ for $i = 1, \ldots, \lambda$, $j = 1, \ldots, n$ and $k = 0, 1$ and sets $PK := (N, h, H, \{X^{(k)}_{i,j}\}_{i \in [\lambda], j \in [n], k \in \{0,1\}})$. Then it chooses $b \xleftarrow{\$} \{0, 1\}$ and $r^* \xleftarrow{\$} [(N-1)/4]$, and computes $C^* := g^{r^*}$, $t^* := h(C^*)$ and $K^* := H((\prod_{i=1}^{\lambda} X^{(t^*_i)}_{i,1})^{r^*}, \ldots, (\prod_{i=1}^{\lambda} X^{(t^*_i)}_{i,n})^{r^*})$ where t^*_i denotes the i-th bit of t^* if $b = 1$ and $K^* \xleftarrow{\$} \{0, 1\}^\lambda$ otherwise. Then it gives (PK, C^*, K^*) to $\mathcal{A}$. In the game, $\mathcal{A}$ can query pairs of ciphertexts and predicates to an oracle $\mathcal{O}_{\mathsf{Dec}}$. When $\mathcal{A}$ queries (C, pred), $\mathcal{O}_{\mathsf{Dec}}$ computes $K \leftarrow \mathsf{Dec}(SK, C)$ and returns K to $\mathcal{A}$ if $C \neq C^*$ and $\mathsf{pred}(K) = 1$, and otherwise $\perp$. Finally, $\mathcal{A}$ outputs a bit b'.

Game 2 : This game is the same as the previous game except that K^* is set differently if $b = 1$. Specifically, it is set as $K^* := H(C^{*\sum_{i=1}^{\lambda} x^{(t^*_i)}_{i,1}}, \ldots, C^{*\sum_{i=1}^{\lambda} x^{(t^*_i)}_{i,n}})$ if $b = 1$.

Game 3 : This game is the same as the previous game except that C^* is set differently. Specifically, it is uniformly chosen from $\mathbb{QR}_N$.

Game 4 : This game is the same as the previous game except that g is uniformly chosen from a subgroup S of $\mathbb{QR}_N$, which is defined as follows. First, we define a PPT algorithm $\mathcal{B}$ as follows.

$\mathcal{B}(N, g)$: Choose $H \xleftarrow{\$} \mathcal{H}$, $x_{i,j}^{(k)} \xleftarrow{\$} [(N-1)/4]$ and set $X_{i,j}^{(k)} := g^{x_{i,j}^{(k)}}$ for $i \in [\lambda]$, $j \in [n]$ and $k = 0, 1$ and $PK := (N, h, H, \{X_{i,j}^{(k)}\}_{i \in [\lambda], j \in [n], k \in \{0,1\}})$, choose $C^* \xleftarrow{\$} \mathbb{QR}_N$ and $b \xleftarrow{\$} \{0,1\}$, and set $K^* := H(C^{*\sum_{i=1}^{\lambda} x_{i,1}^{(t_i^*)}}, \ldots, C^{*\sum_{i=1}^{\lambda} x_{i,n}^{(t_i^*)}})$ where $t^* := h(C^*)$ and t_i^* is the i-th bit of t^* if $b = 1$, and $K^* \xleftarrow{\$} \{0,1\}^\ell$ otherwise. Run $b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Dec}}}(PK, C^*, K^*)$ and output b'. We note that $\mathcal{B}$ can simulate $\mathcal{O}_{\mathsf{Dec}}$ for $\mathcal{A}$ since it knows $SK = (\{x_{i,j}^{(k)}\}_{i \in [\lambda], j \in [n], k \in \{0,1\}}, PK)$.

Let $\mathcal{S}_{\mathcal{B},\mathsf{poly}/2}$ be the algorithm that is assumed to exist in the definition of m-ad-DRSA assumption. Note that this algorithm actually exists since $\mathcal{B}$ is a PPT algorithm and $\mathsf{poly}/2$ is noticeable. Then we define the subgroup S as follows: We run $\{p_1, \ldots, p_m\} \leftarrow \mathcal{S}_{\mathcal{B},\mathsf{poly}/2}$ and define $S := \{h^{p_1,\cdots,p_m} : h \in \mathbb{QR}_N\}$.

Game 5 : This game is the same as the previous game except that the decryption oracle $\mathcal{O}_{\mathsf{Dec}}$ is replaced with an alternative decryption oracle $\mathcal{O}_{\mathsf{Dec}'}$ that works as follows: $\mathcal{O}_{\mathsf{Dec}'}$, given C and pred, computes $t := h(C)$ and returns $\perp$ if $t = t^*$. Otherwise it computes $K := H(C^{\sum_{i=1}^{\lambda} x_{i,1}^{(t_i)}}, \ldots, C^{\sum_{i=1}^{\lambda} x_{i,n}^{(t_i)}})$ and outputs K if $\mathsf{pred}(K) = 1$, and otherwise $\perp$.

Game 6 : This game is the same as the previous game except that $x_{i,j}^{(k)}$ is set differently. Specifically, it is uniformly chosen from $\mathrm{ord}(\mathbb{QR}_N)$ instead of from $[(N-1)/4]$ for $i = 1, \ldots, \lambda$, $j = 1, \ldots, n$ and $k = 0, 1$.

Game 7 : This game is the same as the previous game except that the decryption oracle $\mathcal{O}_{\mathsf{Dec}'}$ is replaced with an alternative decryption oracle $\mathcal{O}_{\mathsf{Dec}''}$ that works as follows: $\mathcal{O}_{\mathsf{Dec}''}$, given C and pred, computes $t := h(C)$ and returns $\perp$ if $t = t^*$ or $C \notin S$, where S is the group defined in Game 4. Otherwise it computes $K := H(C^{\sum_{i=1}^{\lambda} x_{i,1}^{(t_i)}}, \ldots, C^{\sum_{i=1}^{\lambda} x_{i,n}^{(t_i)}})$and outputs K if $\mathsf{pred}(K) = 1$, and otherwise $\perp$.

Game 8 : This game is the same as the previous game except that K^* is always an independently random string.

Let T_i be the event that $b = b^*$ holds in Game i. Then clearly we have $\mathsf{Adv}^{\mathsf{CCCA}}_{\mathcal{A},\mathsf{PKE}_{\mathsf{CCCA}}} = |\Pr[T_1] - 1/2|$. First, we prove that the group S defined in Game 4 is a proper subgroup of $\mathbb{QR}_N$. Moreover, we prove that $\mathrm{ord}(S)/\mathrm{ord}(\mathbb{QR}_N) \leq 2^{-\lambda}$ holds. By the definition of SS moduli, $\mathrm{ord}(\mathbb{QR}_N)$ has $t_p + t_q$ distinct prime factors $p'_1, \ldots, p'_{t_p+t_q}$ of ℓ_B-bit. Since the number of the all ℓ_B-bit primes is M_{ℓ_B}, there exist at least $t_p + t_q + m - M_{\ell_B}$ distinct primes contained in both $\{p'_1, \ldots, p'_{t_p+t_q}\}$ and $\{p_1, \ldots, p_m\}$. We denote those primes by $p''_1, \ldots p''_{t_p+t_q+m-M_{\ell_B}}$. Those primes cannot be a factor of $\mathrm{ord}(S)$ since $S = \{h^{p_1 \cdots p_m} : h \in \mathbb{QR}_N\}$ by the definition whereas they are a factor of $\mathrm{ord}(\mathbb{QR}_N)$.

Thus $\mathrm{ord}(S)/\mathrm{ord}(\mathbb{QR}_N) \leq \frac{1}{p''_1 \cdots p''_{t_p+t_q+m-M_{\ell_B}}}$
$\leq 2^{-(t_p+t_q+m-M_{\ell_B})(\ell_B-1)} \leq 2^{-\lambda}$. Then we prove the following lemmas.

Lemma 27. $\Pr[T_2] = \Pr[T_1]$ *holds.*

Proof. The modification between Game 1 and 2 is only conceptual. □

Lemma 28. $|\Pr[T_3] - \Pr[T_2]|$ *is negligible.*

Proof. This follows from the fact that the statistical distance between the uniform distributions on $[(N-1/4)]$ and $[\mathrm{ord}(\mathbb{QR}_N)]$ are negligible. □

Lemma 29. *We have* $|\Pr[T_4] - \Pr[T_3]| \leq 1/(2\mathsf{poly})$ *for sufficiently large* λ.

Proof. This follows immediately from the definition of m-ad-DRSA assumption. □

Lemma 30. *If h is collision resistant, then* $|\Pr[T_5] - \Pr[T_4]|$ *is negligible.*

Proof. From the view of $\mathcal{A}$, Game 4 and 5 may differ only if $\mathcal{A}$ makes a query (C, pred) such that $h(C) = t^*$. If $\mathcal{A}$ makes such a query, then this means that it finds a collision of h. □

Lemma 31. $|\Pr[T_6] - \Pr[T_5]|$ *is negligible.*

Proof. This follows from the fact that the statistical distance between the uniform distributions on $[(N-1/4)]$ and $[\mathrm{ord}(\mathbb{QR}_N)]$ are negligible. □

Lemma 32. $|\Pr[T_7] - \Pr[T_6]|$ *is negligible.*

Proof. Let q be an upper bound of the number of decryption queries $\mathcal{A}$ makes. We consider hybrids $\mathsf{H}_0, \ldots, \mathsf{H}_q$ that are defined as follows. A hybrid H_ℓ is the same as Game 6 except that the oracle to which $\mathcal{A}$ accesses works similarly as $\mathcal{O}_{\mathsf{Dec}''}$ for the first ℓ queries, and similarly as $\mathcal{O}_{\mathsf{Dec}'}$ for the rest of queries. Let $T_{6,\ell}$ be the event that $b = b'$ holds in the hybrid H_ℓ. Clearly, We have $\Pr[T_{6,0}] = \Pr[T_6]$ and $\Pr[T_{6,\ell}] = \Pr[T_7]$. Let F_ℓ be the event that $\mathcal{O}_{\mathsf{Dec}''}$ returns $\perp$ for $\mathcal{A}$'s ℓ-th query $(C_\ell, \mathsf{pred}_\ell)$ but $\mathcal{O}_{\mathsf{Dec}'}$ does not return $\perp$ for it. That is, F_ℓ is the event that $C_\ell \in \mathbb{QR}_N \setminus S$, $t \neq t^*$ and $\mathsf{pred}(K_\ell) = 1$ hold where $K_\ell := H(C_\ell^{\sum_{i=1}^{\lambda} x_{i,1}^{(t_i)}}, \ldots, C_\ell^{\sum_{i=1}^{\lambda} x_{i,n}^{(t_i)}})$, $t := h(C_\ell)$ and t_i denotes the i-th bit of t. Unless F_ℓ occurs, hybrids H_ℓ and $\mathsf{H}_{\ell-1}$ are exactly the same from the view of $\mathcal{A}$. Therefore we have $|\Pr[T_{6,\ell}] - \Pr[T_{6,\ell-1}]| \leq \Pr[F_\ell]$. Let view_ℓ be the view from $\mathcal{A}$ before it is given the response for its ℓ-th query. That is, view_ℓ consists of PK, C^*, K^*, C_ℓ and decryption queries and responses for them before the ℓ-th query. We prove the following claim.

Claim 3. *If $C_\ell \in \mathbb{QR}_N \setminus S$ and $t \neq t^*$, then K_ℓ is distributed almost uniformly on $\{0,1\}^\lambda$ from the view of $\mathcal{A}$ in the hybrids $\mathsf{H}_{\ell-1}$ and H_ℓ. More precisely, we have $\Delta((K_\ell, \mathsf{view}_\ell), (U, \mathsf{view}_\ell)) \leq 2^{-\lambda}$ where $U \overset{\$}{\leftarrow} \{0,1\}^\lambda$.*

Assume this claim is true. Then we prove that $\Pr[F_\ell]$ is negligible for any $\ell \in [q]$. Since $\mathcal{A}$ is valid, pred is non-trivial. That is, for independently uniform U, $\Pr[\mathsf{pred}_i(U) = 1]$ is negligible. By Claim 3, if $C_\ell \in \mathbb{QR}_N \setminus S$ and $t \neq t^*$, then we have $\Delta((K_\ell, \mathsf{view}), (U, \mathsf{view})) \leq 2^{-\lambda}$ where $U \overset{\$}{\leftarrow} \{0,1\}^\lambda$. Therefore $\Pr[\mathsf{pred}(K_\ell) = 1]$ is negligible and thus $\Pr[F_\ell]$ is negligible. Thus $|\Pr[T_7] - \Pr[T_6]|$ is negligible by the hybrid argument. What is left is to prove Claim 3.

Proof. (of Claim 3) Since we assumed $t \neq t^*$, there exists $i \in [\lambda]$ such that $t_i \neq t_i^*$. We denote minimum such i by i^*. Since $C \in \mathbb{QR}_N \setminus S$ and S is a proper subgroup of $\mathbb{QR}_N$, there exists an ℓ_B-bit prime $\bar{p}$ that divides $\mathrm{ord}(C)$ but does not divide $\mathrm{ord}(S)$. Here, we claim that the decryption oracle before ℓ-th query can be simulated by using $\{x_{i,j}^{(k)} \mod \mathrm{ord}(S)\}_{i\in[\lambda], j\in[n], k\in\{0,1\}}$ and PK. This can be seen by that the oracle immediately returns $\perp$ for a query (C, pred) such that $C \notin S$. If we define $\mathsf{view}'_\ell := (\overline{PK}, C^*, K^*, C_\ell, \{x_{i,j}^{(k)} \mod \mathrm{ord}(S)\}_{i\in[\lambda], j\in[n], k\in\{0,1\}})$ where $\overline{PK}$ denotes a public key except H, then we have $\Delta((K_\ell, \mathsf{view}_\ell), (U, \mathsf{view}_\ell)) \leq \Delta((K_\ell, H, \mathsf{view}'_\ell), (U, H, \mathsf{view}'_\ell))$. Thus it suffices to show that conditioned on any fixed value of view'_ℓ, $\Delta((K_\ell, H), (U_\ell, H)) \leq 2^{-\lambda}$ holds. One can see that view'_ℓ does not depend on $(x_{i^*,j}^{(t_{i^*})} \mod \bar{p})$ at all for $j \in [n]$: $\overline{PK}$ does not depend on $(x_{i^*,j}^{(t_{i^*})} \mod \bar{p})$ since $g \in S$ by the modification from Game 3 to 4. (C^*, K^*) does not depend on $(x_{i^*,j}^{(t_{i^*})} \mod \bar{p})$ since we assumed $t_{i^*} \neq t_{i^*}^*$ and thus $x_{i^*,j}^{(t_{i^*})}$ is not used for generating K^*. $\{x_{i,j}^{(k)} \mod \mathrm{ord}(S)\}_{i\in[\lambda], j\in[n], k\in\{0,1\}}$ does not depend on $(x_{i^*,j}^{(t_{i^*})} \mod \bar{p})$ since $\mathrm{ord}(S)$ is coprime to $\bar{p}$. Thus conditioned on any value of view'_ℓ, $(x_{i^*,j}^{(t_{i^*})} \mod \bar{p})$ is distributed uniformly for all $j \in [n]$. Therefore we have $H_\infty(C_\ell^{\sum_{i=1}^\lambda x_{i,1}^{(t_i)}}, \ldots, C_\ell^{\sum_{i=1}^\lambda x_{i,n}^{(t_i)}} | \mathsf{view}'_\ell) \geq n \log \bar{p} \geq n(\ell_B - 1) \geq \lambda + 2\ell_N\lambda$. Here, we use Lemma 26: We set $\mathcal{X} := \{X_C\}_{C \in \mathbb{QR}_N \setminus S}$ where X_C denotes a random variable that is distributed as $(C^{\sum_{i=1}^\lambda x_{i,1}^{(t_i)}}, \ldots, C^{\sum_{i=1}^\lambda x_{i,n}^{(t_i)}})$ conditioned on view'_ℓ, $\delta := 2^{-\ell_N\lambda}$, and $\mathcal{F}$ as an algorithm that simulates the game between $\mathcal{A}$ and the challenger and outputs X_{C_ℓ} where C_ℓ is $\mathcal{A}$'s ℓ-th decryption query. Then we have $\Delta((K_\ell, H), (U, H)) \leq |\mathbb{QR}_N \setminus S| 2^{-\ell_N\lambda} \leq 2^{-\lambda}$ where $U \overset{\$}{\leftarrow} \{0,1\}^\lambda$, conditioned on any fixed value of view'_ℓ. Thus the proof of Claim 3 is completed. □

This concludes the proof of Lemma 32.

□

Lemma 33. $|\Pr[T_8] - \Pr[T_7]|$ *is negligible.*

Proof. Since we have $\Pr[C^* \in S : C^* \xleftarrow{\$} \mathbb{QR}_N] \leq 2^{-\lambda}$, in the following, we assume $C^* \notin S$. Then there exists $\bar{p}$ that divides $\mathrm{ord}(C^*)$ but does not divide $\mathrm{ord}(S)$. Let view be the view from $\mathcal{A}$ in Game 8 except K^*, and $\mathsf{view}' := \{\overline{PK}, C^*, \{x_{i,j}^{(k)} \bmod \mathrm{ord}(S)\}_{i\in[\lambda],j\in[n],k\in\{0,1\}}\}$. By a similar argument as in the proof of Claim 3, we have $\Delta((K^*, \mathsf{view}), (U, \mathsf{view})) \leq \Delta((K^*, H, \mathsf{view}'), (U, H, \mathsf{view}'))$ and $\tilde{H}_\infty(C^{*\sum_{i=1}^{\lambda} x_{i,1}^{(t_i)}}, \ldots, C^{*\sum_{i=1}^{\lambda} x_{i,n}^{(t_i)}} | \mathsf{view}') \geq n \log \bar{p} \geq n(\ell_B - 1) \geq (2\ell_N + 1)\lambda$. If we let $X := (C^{*\sum_{i=1}^{\lambda} x_{i,1}^{(t_i)}}, \ldots, C^{*\sum_{i=1}^{\lambda} x_{i,n}^{(t_i)}})$, $Y := \mathsf{view}$ and $\delta := 2^{-\ell_N \lambda}$ in Lemma 12, then we have $\Delta((K^*, H, \mathsf{view}'), (U, H, \mathsf{view}')) \leq 2^{-\ell_N \lambda}$ where $K^* = H(C^{*x_1 + t^* y_1}, \ldots, C^{*x_n + t^* y_n})$ and $U \xleftarrow{\$} \{0,1\}^k$. Thus the lemma follows. □

By the above lemmas, we have $\mathsf{Adv}^{\mathsf{CCCA}}_{\mathcal{A},\mathsf{KEM}_{\mathsf{CCCA}}}(\lambda) = |\Pr[T_1] - \Pr[T_8]| \leq \mathsf{negl}(\lambda) + 1/(2\mathsf{poly}(\lambda))$ for sufficiently large λ where negl is some negligible function. On the other hand, we assumed that there are infinitely many λ such that $\mathsf{Adv}^{\mathsf{CCCA}}_{\mathcal{A},\mathsf{KEM}_{\mathsf{CCCA}}}(\lambda) > 1/\mathsf{poly}(\lambda)$. Therefore for infinitely many λ, we have $1/(2\mathsf{poly}(\lambda)) < \mathsf{negl}(\lambda)$, which contradicts to that $\mathsf{negl}(\lambda)$ is negligible. Thus there does not exist a valid PPT adversary that breaks the CCCA security of the scheme. □

Discussion. Here, we discuss the efficiency of the CCA secure PKE scheme that is obtained by combining the above KEM and an authenticated symmetric key encryption scheme. Table 4.1 shows the efficiency and hardness assumption of CCA secure PKE schemes based on the factoring in the standard model. Among existing schemes, the scheme proposed by Hofheinz and Kiltz [HK09b] is one of the best in regard to the ciphertext overhead, which consists of 2 elements of $\mathbb{Z}_N^*$. In contrast, the ciphertext overhead of our scheme consists of only 1 element of $\mathbb{Z}_N^*$ plus a MAC. By giving a concrete parameter ($\ell'_p = \ell'_q = 160$, $\ell_B = 15$, $t_p = t_q = 32$ and $\ell_N = 1280$), the ciphertext overhead of our scheme is 1360-bit for 80-bit security whereas that of [HK09b] is 2048-bit. On the other hand, the public key size of our scheme is much larger than that of [HK09b], and an encryption and decryption are much less efficient than those in [HK09b].

Chapter 5

Concluding Remarks and Open Problems

In the first part of this dissertation, we define a self-bilinear map with auxiliary information (AI-SBM) and construct it based on the factoring assumption and the existence of indistinguishability obfuscation (iO). There are following open problems regarding to this part.

- The first and most important open problem is to remove iO from our construction. Though we rely on the existence of iO, there is no known construction of iO based on the factoring assumption. Moreover, known constructions of iO are too inefficient and far from practical. It is necessary to remove iO for obtaining practical and solely factoring-based construction of AI-SBMs.
- The second problem is about constructing AI-SBMs with more useful properties. Though we show that an AI-SBM is useful in many applications, there are still some limitations. The most significant drawback is that the size expansion of auxiliary information. That is, given two auxiliary information τ_x and τ_y which correspond to g^x and g^y respectively, if we compute τ_{x+y} which corresponds to g^{x+y}, then the size of τ_{x+y} expands to be polynomial in the sizes of τ_x and τ_y. This unable us to apply group operations recursively more than constant times. It is an interesting open problem to construct an AI-SBM where there is no size expansion of auxiliary information, that would be much more useful than our scheme.
- The third problem is to construct an AI-SBM on a different group from $\mathbb{QR}_N^+$. It seems that the only property we use in the proof is that it is computationally hard to compute c-th root on an underlying group G where c is coprime

to $\mathrm{ord}(G)$. It would be interesting future work to prove the above intuition formally and find new instantiations.

In the second part of this dissertation, we define an adversary dependent lossy trapdoor function (ad-LTDF) and construct it based on the factoring assumption w.r.t. semi-smooth subgroup RSA moduli (SS moduli). There are following open problems regarding to this part.

- The first problem is to extend our technique to more general RSA moduli. In our result, we focus only on SS moduli, which have a special structure. We do not know if it is possible to obtain a similar result w.r.t. another type of RSA moduli. Since SS moduli is not a very standard form of RSA moduli, it would be better if we could construct an ad-LTDF based on the factoring assumption w.r.t. more general type of RSA moduli.
- The second problem is to analyze the hardness of factorizing an SS modulus. Though SS moduli have been considered for more than 10 years, study of efficient algorithm to factorize this type of RSA moduli is still not enough. Since we show the usefulness of the factoring assumption w.r.t. SS moduli, it would be important to analyze if factorizing SS moduli is really hard or not.

Acknowledgement

Firstly, I would like to thank my supervisor Associate Professor Noboru Kunihiro for his insightful suggestions. I would like to thank Professor Hirosuke Yamamoto and Assistant Professor Junya Honda for their advice and encouragement.

I would like to thank all current and former members of Yamamoto-Kunihiro Laboratory, Yao Lu, Masayuki Yoshino, Shota Yamada, Toru Akishita, Takayuki Kawai, Takahisa Suzuki, Yuto Sogo, Yuji Nagashima, Xiaofeng Wei, Masashi Ueda, Atsushi Takayasu, Kosei Endo, Takuma Koyama, Tomohiro Nakata, Yuka Kuwaori, Ko Sugimoto, Yuki Takahashi, Yuho Matsunaga, Weilun Liu, Yoshinao Uchide, Vannet Thomas Francis, Shuichi Katsumata, Ying Hwei Ming Jason, Yuki Tanigaki, Yuka Miyazaki, Yoshiki Machino, Ken Kaminakaya, Keisuke Kinoshita, Qiqiang Liu, Enze Ren, Shintaro Hara, Satoshi Furukawa, Kento Ohnishi, Sota Onozawa, Kazuho Kato, Yun Sheng, Sihui Chu, Tomotaka Hiraoka, Eisuke Moriwaki, Mengce Zheng for their friendship. I would like to thank the secretary Chiaki Sakakibara for her support.

I would like to thank all collaborators, Goichiro Hanaoka, Koji Nuida, Takahiro Matsuda, Hajime Watanabe (from AIST), Go Ohtake, Yuki Hironaka, Kenjiro Kai, Yosuke Endo (from NHK science and technology research laboratory), Nobuaki Kitajima, Takashi Nishide, Eiji Okamoto (from Tsukuba University), Yoshikazu Hanatani (from Toshiba), Professor Hideki Imai, Kohei Kasamatsu, Ikuma Fujiwara (from Chuo University) .

I would like to thank all members of the study group Shin-Akarui-Angou-Benkyou-Kai for useful discussions.

I would like to thank all members of NTT Secure Platform Laboratories for accepting me as an internship student. Especially, I thank Ryo Nishimaki, the mentor in NTT, for insightful suggestions and discussions during the summer intern.

I would like to thank my family for their understanding and encouragement.

Works presented in this dissertation are supported by Research Fellowships of Japan Society for the Promotion of Science for Young Scientists (JSPS).

Bibliography

[ABCP15] Michel Abdalla, Florian Bourse, Angelo De Caro, and David Pointcheval. Simple functional encryption schemes for inner products. In *PKC*, pages 733–751, 2015.

[ABSV15] Prabhanjan Ananth, Zvika Brakerski, Gil Segev, and Vinod Vaikuntanathan. From selective to adaptive security in functional encryption. In *CRYPTO Part II*, pages 657–677, 2015.

[AFH$^+$16] Martin R. Albrecht, Pooya Farshim, Dennis Hofheinz, Enrique Larraia, and Kenneth G. Paterson. Multilinear maps from obfuscation. In *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, pages 446–473, 2016.

[AGIS14] Prabhanjan Ananth, Divya Gupta, Yuval Ishai, and Amit Sahai. Optimizing obfuscation: Avoiding Barrington's theorem. Cryptology ePrint Archive, Report 2014/222, 2014. `http://eprint.iacr.org/`.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC*, pages 99–108, 1996.

[ALS16] Shweta Agrawal, Benoît Libert, and Damien Stehlé. Fully secure functional encryption for inner products, from standard assumptions. In *CRYPTO Part III*, pages 333–362, 2016.

[BB04] Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In *EUROCRYPT*, pages 223–238, 2004.

[BBO07] Mihir Bellare, Alexandra Boldyreva, and Adam O'Neill. Deterministic and efficiently searchable encryption. In *CRYPTO*, pages 535–552, 2007.

[BBS86] Lenore Blum, Manuel Blum, and Mike Shub. A simple unpredictable pseudo-random number generator. *SIAM J. Comput.*, 15(2):364–383, 1986.

[BF01] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO*, pages 213–229, 2001.

[BF11] Dan Boneh and David Mandell Freeman. Homomorphic signatures for polynomial functions. In *EUROCRYPT*, pages 149–168, 2011.

[BFO08] Alexandra Boldyreva, Serge Fehr, and Adam O'Neill. On notions of security for deterministic encryption, and efficient constructions without random oracles. In *CRYPTO*, pages 335–359, 2008.

[BFOR08] Mihir Bellare, Marc Fischlin, Adam O'Neill, and Thomas Ristenpart. Deterministic encryption: Definitional equivalences and constructions without random oracles. In *CRYPTO*, pages 360–378, 2008.

[BG84] Manuel Blum and Shafi Goldwasser. An efficient probabilistic public-key encryption scheme which hides all partial information. In *CRYPTO*, pages 289–302, 1984.

[BGG$^+$14] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In *EUROCRYPT 2014*, pages 533–556, 2014.

[BGH07] Dan Boneh, Craig Gentry, and Michael Hamburg. Space-efficient identity based encryption without pairings. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, pages 647–657, 2007.

[BGI$^+$01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In *CRYPTO*, pages 1–18, 2001.

[BGK$^+$14] Boaz Barak, Sanjam Garg, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. Protecting obfuscation against algebraic attacks. In *EUROCRYPT*, pages 221–238, 2014.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *ITCS*, pages 309–325, 2012.

[BGW05] Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *CRYPTO*, pages 258–275, 2005.

[BHJ$^+$13] Florian Böhl, Dennis Hofheinz, Tibor Jager, Jessica Koch, Jae Hong Seo, and Christoph Striecks. Practical signatures from standard assumptions. In *EUROCRYPT*, pages 461–485, 2013.

[BHR12] Mihir Bellare, Viet Tung Hoang, and Phillip Rogaway. Foundations of garbled circuits. In *the ACM Conference on Computer and Communica-*

tions Security, CCS'12, Raleigh, NC, USA, October 16-18, 2012, pages 784–796, 2012.

[BHY09] Mihir Bellare, Dennis Hofheinz, and Scott Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In *EUROCRYPT*, pages 1–35, 2009.

[Ble98] Daniel Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard pkcs #1. In *CRYPTO*, pages 1–12, 1998.

[Boy08] Xavier Boyen. The uber-assumption family. In *Pairing-Based Cryptography - Pairing 2008, Second International Conference, Egham, UK, September 1-3, 2008. Proceedings*, pages 39–56, 2008.

[BP97] Niko Barić and Birgit Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *EUROCRYPT*, pages 480–494, 1997.

[BR14] Zvika Brakerski and Guy N. Rothblum. Virtual black-box obfuscation for all circuits via generic graded encoding. In *TCC*, pages 1–25, 2014.

[Bra12] Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. In *CRYPTO*, pages 868–886, 2012.

[BS02] Dan Boneh and Alice Silverberg. Applications of multilinear forms to cryptography. *Contemporary Mathematics*, 324:71–90, 2002.

[BSW07] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy*, pages 321–334, 2007.

[BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings*, pages 253–273, 2011.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *FOCS*, pages 97–106, 2011.

[BZ14] Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In *CRYPTO*, 2014.

[CCK$^+$13] Jung Hee Cheon, Jean-Sébastien Coron, Jinsu Kim, Moon Sung Lee, Tancrède Lepoint, Mehdi Tibouchi, and Aaram Yun. Batch fully homomorphic encryption over the integers. In *EUROCRYPT*, pages 315–335, 2013.

[CFL$^+$16] Jung Hee Cheon, Pierre-Alain Fouque, Changmin Lee, Brice Minaud, and Hansol Ryu. Cryptanalysis of the new CLT multilinear map over the integers. In *EUROCRYPT 2016 Part I*, pages 509–536, 2016.

[CFW14] Dario Catalano, Dario Fiore, and Bogdan Warinschi. Homomorphic signatures with efficient verification for polynomial functions. In *CRYPTO*, pages 371–389, 2014.

[CGGI16] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In *ASIACRYPT Part I*, pages 3–33, 2016.

[CHL$^+$15] Jung Hee Cheon, Kyoohyung Han, Changmin Lee, Hansol Ryu, and Damien Stehlé. Cryptanalysis of the multilinear map over the integers. In *EUROCRYPT 2015 Part I*, pages 3–12, 2015.

[CJM$^+$11] Jean-Sébastien Coron, Antoine Joux, Avradip Mandal, David Naccache, and Mehdi Tibouchi. Cryptanalysis of the RSA subgroup assumption from TCC 2005. In *Public Key Cryptography*, pages 147–155, 2011.

[CL09] Jung Hee Cheon and Dong Hoon Lee. A note on self-bilinear maps. *Bulletin of the Korean Mathematical Society*, 46(2):303–309, 2009.

[Cla94] Josh Benaloh Clarkson. Dense probabilistic encryption. In *the Workshop on Selected Areas of Cryptography*, pages 120–128, 1994.

[CLLT16] Jean-Sébastien Coron, Moon Sung Lee, Tancrède Lepoint, and Mehdi Tibouchi. Cryptanalysis of GGH15 multilinear maps. In *CRYPTO 2016 Part II*, pages 607–628, 2016.

[CLT13] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In *CRYPTO (1)*, pages 476–493, 2013.

[CLT14] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Scale-invariant fully homomorphic encryption over the integers. In *PKC*, pages 311–328, 2014.

[CLT15] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. New multilinear maps over the integers. In *CRYPTO 2015 Part I*, pages 267–286, 2015.

[CMNT11] Jean-Sébastien Coron, Avradip Mandal, David Naccache, and Mehdi Tibouchi. Fully homomorphic encryption over the integers with shorter public keys. In *CRYPTO*, pages 487–504, 2011.

[CMS99] Christian Cachin, Silvio Micali, and Markus Stadler. Computationally private information retrieval with polylogarithmic communication. In

EUROCRYPT, pages 402–414, 1999.

[CNT12] Jean-Sébastien Coron, David Naccache, and Mehdi Tibouchi. Public key compression and modulus switching for fully homomorphic encryption over the integers. In *EUROCRYPT*, pages 446–464, 2012.

[Coc01] Clifford Cocks. An identity based encryption scheme based on quadratic residues. In *Cryptography and Coding, 8th IMA International Conference, Cirencester, UK, December 17-19, 2001, Proceedings*, pages 360–363, 2001.

[Cop96] Don Coppersmith. Finding a small root of a bivariate integer equation; factoring with high bits known. In *Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding*, pages 178–189, 1996.

[CP05] Richard Crandall and Carl Pomerance. *Prime Numbers: A Computational Perspective.* Springer, 2nd edition, 2005.

[CS00] Ronald Cramer and Victor Shoup. Signature schemes based on the strong RSA assumption. *ACM Trans. Inf. Syst. Secur.*, 3(3):161–185, 2000.

[CS02] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *EUROCRYPT*, pages 45–64, 2002.

[DDN91] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography (extended abstract). In *STOC*, pages 542–552, 1991.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[DORS08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008.

[FGK$^+$10] David Mandell Freeman, Oded Goldreich, Eike Kiltz, Alon Rosen, and Gil Segev. More constructions of lossy and correlation-secure trapdoor functions. In *Public Key Cryptography*, pages 279–295, 2010.

[FHKP13] Eduarda S. V. Freire, Dennis Hofheinz, Eike Kiltz, and Kenneth G. Paterson. Non-interactive key exchange. In *Public Key Cryptography*, pages 254–271, 2013.

[Fis03] Marc Fischlin. The cramer-shoup strong-rsasignature scheme revisited. In *PKC*, pages 116–129, 2003.

[FN93] Amos Fiat and Moni Naor. Broadcast encryption. In *Advances in Cryp-*

tology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings, pages 480–491, 1993.

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. How to enhance the security of public-key encryption at minimum cost. In *Public Key Cryptography*, pages 53–68, 1999.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178, 2009.

[GGG$^+$14] Shafi Goldwasser, S. Dov Gordon, Vipul Goyal, Abhishek Jain, Jonathan Katz, Feng-Hao Liu, Amit Sahai, Elaine Shi, and Hong-Sheng Zhou. Multi-input functional encryption. In *Eurocrypt*, 2014.

[GGH13a] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In *EUROCRYPT*, pages 1–17, 2013.

[GGH$^+$13b] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *FOCS*, pages 40–49, 2013.

[GGH$^+$13c] Sanjam Garg, Craig Gentry, Shai Halevi, Amit Sahai, and Brent Waters. Attribute-based encryption for circuits from multilinear maps. In *CRYPTO (2)*, pages 479–499, 2013.

[GGH15] Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In *TCC 2015 Part II*, pages 498–527, 2015.

[GGHR14] Sanjam Garg, Craig Gentry, Shai Halevi, and Mariana Raykova. Two-round secure MPC from indistinguishability obfuscation. In *TCC*, pages 74–94, 2014.

[GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions (extended abstract). In *FOCS*, pages 464–479, 1984.

[GHR99] Rosario Gennaro, Shai Halevi, and Tal Rabin. Secure hash-and-sign signatures without the random oracle. In *EUROCRYPT*, pages 123–139, 1999.

[GKKR10] Rosario Gennaro, Jonathan Katz, Hugo Krawczyk, and Tal Rabin. Secure network coding over the integers. In *PKC*, pages 142–160, 2010.

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *STOC*, pages 25–32, 1989.

[GLSW15] Craig Gentry, Allison Bishop Lewko, Amit Sahai, and Brent Waters. Indistinguishability obfuscation from the multilinear subgroup elimination assumption. In *FOCS 2015*, pages 151–170, 2015.

[GM82] Shafi Goldwasser and Silvio Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing, May 5-7, 1982, San Francisco, California, USA*, pages 365–377, 1982.

[GMM$^+$16] Sanjam Garg, Eric Miles, Pratyay Mukherjee, Amit Sahai, Akshayaram Srinivasan, and Mark Zhandry. Secure obfuscation in a weak multilinear map model. In *TCC 2016-B, Part II*, pages 241–268, 2016.

[GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2):281–308, 1988.

[GOS06] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for NP. In *EUROCRYPT*, pages 339–358, 2006.

[GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM Conference on Computer and Communications Security*, pages 89–98, 2006.

[Gro05] Jens Groth. Cryptography in subgroups of $\mathbb{Z}_n^*$. In *TCC*, pages 50–65, 2005.

[GS08] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In *EUROCRYPT*, pages 415–432, 2008.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO Part I*, pages 75–92, 2013.

[GVW13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In *STOC*, pages 545–554, 2013.

[GVW15] Sergey Gorbunov, Vinod Vaikuntanathan, and Daniel Wichs. Leveled fully homomorphic signatures from standard lattices. In *STOC 2015*, pages 469–477, 2015.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[HJ16] Yupu Hu and Huiwen Jia. Cryptanalysis of GGH map. In *EUROCRYPT 2016 Part I*, pages 537–565, 2016.

[HJK11] Dennis Hofheinz, Tibor Jager, and Eike Kiltz. Short signatures from weaker assumptions. In *ASIACRYPT*, pages 647–666, 2011.

[HK07] Dennis Hofheinz and Eike Kiltz. Secure hybrid encryption from weakened key encapsulation. In *CRYPTO*, pages 553–571, 2007.

[HK08] Dennis Hofheinz and Eike Kiltz. Programmable hash functions and their applications. In *CRYPTO*, pages 21–38, 2008.

[HK09a] Dennis Hofheinz and Eike Kiltz. The group of signed quadratic residues and applications. In *CRYPTO*, pages 637–653, 2009.

[HK09b] Dennis Hofheinz and Eike Kiltz. Practical chosen ciphertext secure encryption from factoring. In *EUROCRYPT*, pages 313–332, 2009.

[HMS12] Goichiro Hanaoka, Takahiro Matsuda, and Jacob C. N. Schuldt. On the impossibility of constructing efficient key encapsulation and programmable hash functions in prime order groups. In *CRYPTO*, pages 812–831, 2012.

[Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, March 1963.

[Hof14] Dennis Hofheinz. Fully secure constrained pseudorandom functions using random oracles. Cryptology ePrint Archive, Report 2014/372, 2014. `http://eprint.iacr.org/`.

[HSW14] Susan Hohenberger, Amit Sahai, and Brent Waters. Replacing a random oracle: Full domain hash from indistinguishability obfuscation. *Eurocrypt*, 2014.

[HW09] Susan Hohenberger and Brent Waters. Short and stateless signatures from the RSA assumption. In *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, pages 654–670, 2009.

[JL13] Marc Joye and Benoît Libert. Efficient cryptosystems from 2 ^{k} -th power residue symbols. In *EUROCRYPT*, pages 76–92, 2013.

[Jou00] Antoine Joux. A one round protocol for tripartite diffie-hellman. In *ANTS*, pages 385–394, 2000.

[KMO10] Eike Kiltz, Payman Mohassel, and Adam O'Neill. Adaptive trapdoor functions and chosen-ciphertext security. In *EUROCRYPT*, pages 673–692, 2010.

[KOS10] Eike Kiltz, Adam O'Neill, and Adam Smith. Instantiability of rsa-oaep under chosen-plaintext attack. In *CRYPTO*, pages 295–313, 2010.

[KPSY09] Eike Kiltz, Krzysztof Pietrzak, Martijn Stam, and Moti Yung. A new randomness extraction paradigm for hybrid encryption. In *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference* *on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, pages 590–609, 2009.

[KRS15] Dakshita Khurana, Vanishree Rao, and Amit Sahai. Multi-party key exchange for unbounded parties from indistinguishability obfuscation. In *ASIACRYPT 2015 I*, pages 52–75, 2015.

[Len87] Jr. Lenstra, H. W. Factoring integers with elliptic curves. *The Annals of Mathematics*, 126(3):pp. 649–673, 1987.

[LLL13] Xianhui Lu, Bao Li, and Yamin Liu. How to remove the exponent GCD in HK09. In *ProvSec*, pages 239–248, 2013.

[LLML11] Xianhui Lu, Bao Li, Qixiang Mei, and Yamin Liu. Improved tradeoff between encapsulation and decapsulation of hk09. In *Inscrypt*, pages 131–141, 2011.

[LLML12] Xianhui Lu, Bao Li, Qixiang Mei, and Yamin Liu. Improved efficiency of chosen ciphertext secure encryption from factoring. In *ISPEC*, pages 34–45, 2012.

[MLLJ11] Qixiang Mei, Bao Li, Xianhui Lu, and Dingding Jia. Chosen ciphertext secure encryption under factoring assumption revisited. In *Public Key Cryptography*, pages 210–227, 2011.

[MOV93] Alfred Menezes, Tatsuaki Okamoto, and Scott A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on Information Theory*, 39(5):1639–1646, 1993.

[MSZ16] Eric Miles, Amit Sahai, and Mark Zhandry. Annihilation attacks for multilinear maps: Cryptanalysis of indistinguishability obfuscation over GGH13. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, pages 629–658, 2016.

[NFT09] Ryo Nishimaki, Eiichiro Fujisaki, and Keisuke Tanaka. Efficient non-interactive universally composable string-commitment schemes. In *ProvSec*, pages 3–18, 2009.

[NP00] Moni Naor and Benny Pinkas. Efficient trace and revoke schemes. In *Financial Cryptography*, pages 1–20, 2000.

[NR04] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. *J. ACM*, 51(2):231–262, 2004.

[NRR02] Moni Naor, Omer Reingold, and Alon Rosen. Pseudorandom functions and factoring. *SIAM J. Comput.*, 31(5):1383–1404, 2002.

[NS98] David Naccache and Jacques Stern. A new public key cryptosystem based on higher residues. In *ACM Conference on Computer and Communications Security*, pages 59–66, 1998.

[NY89] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *STOC*, pages 33–43, 1989.

[Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, pages 223–238, 1999.

[PH78] Stephen C. Pohlig and Martin E. Hellman. An improved algorithm for computing logarithms over gf(p) and its cryptographic significance (corresp.). *IEEE Transactions on Information Theory*, 24(1):106–110, 1978.

[Pol74] John M. Pollard. Theorems of factorization and primality testing. In *Proceedings of the Cambridge Philosophical Society*, volume 76, pages 521–528, 1974.

[Pol75] John M. Pollard. A monte carlo method for factorization. In *BIT*, volume 15, pages 331–334, 1975.

[Pol78] John M. Pollard. Monte carlo methods for index computation (mod p). In *Math.Comp.*, volume 32, pages 918–924, 1978.

[PPS15] Omkant Pandey, Manoj Prabhakaran, and Amit Sahai. Obfuscation-based non-black-box simulation and four message concurrent zero knowledge for NP. In *TCC 2015 Part II*, pages 638–667, 2015.

[PS15] Omer Paneth and Amit Sahai. On the equivalence of obfuscation and multilinear maps. *IACR Cryptology ePrint Archive*, 2015:791, 2015.

[PST14] Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation from semantically-secure multilinear encodings. In *CRYPTO 2014 Part I*, pages 500–517, 2014.

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In *STOC*, pages 187–196, 2008.

[Rab79] Michael O. Rabin. Digital signatures and public key functions as intractable as factorization. Technical Report MIT/LCS/TR-212, Massachusetts Institute of Technology, January 1979., 1979.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC 2005*, pages 84–93, 2005.

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *STOC*, pages 387–394, 1990.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[RSV13] Ananth Raghunathan, Gil Segev, and Salil P. Vadhan. Deterministic public-key encryption for adaptively chosen plaintext distributions. In *EUROCRYPT*, pages 93–110, 2013.

[Seu13] Yannick Seurin. New constructions and applications of trapdoor DDH groups. In *Public Key Cryptography*, pages 443–460, 2013.

[Sha71] Daniel Shanks. Class number, a theory of factorization, and genera. In *1969 Number Theory Institute (Proc. Sympos. Pure Math., Vol. XX, State Univ. New York, Stony Brook, N.Y., 1969)*, pages 415–440. Providence, R.I., 1971.

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages 47–53, 1984.

[SOK00] Ryuichi Sakai, Kiyoshi Ohgishi, and Masao Kasahara. Cryptosystems based on pairing (in japanese). In *SCIS*, 2000.

[SOK01] Ryuichi Sakai, Kiyoshi Ohgishi, and Masao Kasahara. Cryptosystems based on pairing over elliptic curve (in japanese). In *SCIS*, 2001.

[SW05] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, pages 457–473, 2005.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. In *STOC*, 2014.

[TV00] Luca Trevisan and Salil P. Vadhan. Extracting randomness from samplable distributions. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pages 32–42, 2000.

[vDGHV10] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In *EUROCRYPT*, pages 24–43, 2010.

[Wat05] Brent Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT*, pages 114–127, 2005.

[Wat15] Brent Waters. CRYPTO part II. pages 678–697, 2015.

[Wee11] Hoeteck Wee. Threshold and revocation cryptosystems via extractable hash proofs. In *EUROCRYPT*, pages 589–609, 2011.

[XLL$^+$13] Haiyang Xue, Bao Li, Xianhui Lu, Dingding Jia, and Yamin Liu. Efficient lossy trapdoor functions based on subgroup membership assumptions. In

CANS, pages 235–250, 2013.

[XX14] Xiang Xie and Rui Xue. Bounded fully homomorphic signature schemes. *IACR Cryptology ePrint Archive*, 2014:420, 2014.

[YHK12] Shota Yamada, Goichiro Hanaoka, and Noboru Kunihiro. Space efficient signature schemes from the RSA assumption. In *PKC*, pages 102–119, 2012.

[YHK16] Takashi Yamakawa, Goichiro Hanaoka, and Noboru Kunihiro. Generalized hardness assumption for self-bilinear map with auxiliary information. In *ACISP*, pages 269–284, 2016.

[YYHK] Takashi Yamakawa, Shota Yamada, Goichiro Hanaoka, and Noboru Kunihiro. Self-bilinear map on unknown order groups from indistinguishability obfuscation and its applications (to appear). *Algorithmica*.

[YYHK14] Takashi Yamakawa, Shota Yamada, Goichiro Hanaoka, and Noboru Kunihiro. Self-bilinear map on unknown order groups from indistinguishability obfuscation and its applications. In *CRYPTO 2014 Part II*, pages 90–107, 2014.

[YYHK16] Takashi Yamakawa, Shota Yamada, Goichiro Hanaoka, and Noboru Kunihiro. Adversary-dependent lossy trapdoor function from hardness of factoring semi-smooth RSA subgroup moduli. In *CRYPTO Part II*, pages 3–32, 2016.

[YYN$^+$14] Takashi Yamakawa, Shota Yamada, Koji Nuida, Goichiro Hanaoka, and Noboru Kunihiro. Chosen ciphertext security on hard membership decision groups: The case of semi-smooth subgroups of quadratic residues. In *SCN*, pages 558–577, 2014.

List of Publications Related to the Dissertation

Journal Papers

1. Takashi Yamakawa, Shota Yamada, Goichiro Hanaoka, Noboru Kunihiro Self-bilinear map on unknown order groups from indistinguishability obfuscation and its applications. In *Algorithmica*, Springer, To appear.

Refereed Conference Papers (with Formal Proceedings)

1. Takashi Yamakawa, Shota Yamada, Goichiro Hanaoka, Noboru Kunihiro Adversary-dependent lossy trapdoor function from hardness of factoring semi-smooth RSA subgroup moduli. In *CRYPTO* (2), pages 3–32, 2016.
2. Takashi Yamakawa, Goichiro Hanaoka, Noboru Kunihiro Generalized hardness assumption for self-bilinear map with auxiliary information. In *ACISP*, pages 269–284, 2016.
3. Takashi Yamakawa, Shota Yamada, Goichiro Hanaoka, Noboru Kunihiro Self-bilinear map on unknown order groups from indistinguishability obfuscation and its applications. In *CRYPTO* (2), pages 90–107, 2014.

List of All Publications

Journal Papers

1. Takashi Yamakawa, Shota Yamada, Goichiro Hanaoka, Noboru Kunihiro Self-bilinear map on unknown order groups from indistinguishability obfuscation and its applications. In *Algorithmica*, Springer, To appear
2. Yoshikazu Hanatani, Goichiro Hanaoka, Takahiro Matsuda, Takashi Yamakawa Efficient key encapsulation mechanisms with tight security reductions to standard assumptions in the two security models. In *Security and Communication Networks* 9(12), pages 1676-1697.

Refereed Conference Papers (with Formal Proceedings)

1. Takashi Yamakawa, Shota Yamada, Goichiro Hanaoka, Noboru Kunihiro Adversary-dependent lossy trapdoor function from hardness of factoring semi-smooth RSA subgroup moduli. In *CRYPTO* (2), pages 3–32, 2016.
2. Takashi Yamakawa, Goichiro Hanaoka, Noboru Kunihiro Generalized hardness assumption for self-bilinear map with auxiliary information. In *ACISP*, pages 269–284, 2016.
3. Takashi Yamakawa, Nobuaki Kitajima, Takashi Nishide, Goichiro Hanaoka, Eiji Okamoto A short fail-stop signature scheme from factoring. In *ProvSec*, pages 309–316, 2014. (Short Paper)
4. Takashi Yamakawa, Shota Yamada, Koji Nuida, Goichiro Hanaoka, Noboru Kunihiro Chosen ciphertext security on haed membership decision groups: The case of semi-smooth subgroups of quadratic residues. In *SCN*, pages 558-577, 2014.
5. Takashi Yamakawa, Shota Yamada, Goichiro Hanaoka, Noboru Kunihiro Self-bilinear map on unknown order groups from indistinguishability obfuscation and its applications. In *CRYPTO* (2), pages 90–107, 2014. (辻井重男セキュリ

ティ学生論文賞受賞)

6. Takashi Yamakawa, Shota Yamada, Takahiro Matsuda, Goichiro Hanaoka, Noboru Kunihiro Reducing public key sizes in bounded CCA-secure KEMs with optimal ciphertext length. In *ISC*, pages 100–109, 2013. (Short Paper)
7. Go Ohtake, Yuki Hironaka, Kenjiro Kai, Yosuke Endo, Goichiro Hanaoka, Hajime Watanabe, Shota Yamada, Kohei Kasamatsu, Takashi Yamakawa, Hideki Imai Partially wildcarded attribute-based encryption and its efficient construction. In *SECRYPT*, pages 339–346, 2013
8. Takashi Yamakawa, Shota Yamada, Takahiro Matsuda, Goichiro Hanaoka, Noboru Kunihiro Efficient variants of the Naor-Yung and Dolev-Dwork-Naor transforms for CCA secure key encapsulation mechanism. In *AsiaPKC*, pages 23–32, 2013.

Non-Refereed Papers

1. 山川高志，花岡悟一郎，國廣昇，“識別不可性難読化に基づく絶対値平方剰余群上の自己双線型写像のさらなる応用” 2016年暗号と情報セキュリティシンポジウム(SCIS2016), 3E2-4, 熊本, 2016年1月.
2. 山川高志，山田翔太，花岡悟一郎，國廣昇，“Semi-smooth RSA数の素因数分解問題に基づく一般化Lossy Trapdoor関数” 2015年暗号と情報セキュリティシンポジウム(SCIS2015), 3E2-2, 小倉, 2015年1月.
3. 北島暢曜，山川高志，西出隆志，花岡悟一郎，岡本栄司，“素因数分解仮定に基づく署名長が短いFail-Stop署名” 2014年暗号と情報セキュリティシンポジウム(SCIS2014), 2D1-3, 鹿児島, 2014年1月.
4. 山川高志，山田翔太，花岡悟一郎，國廣昇，“セミスムース数を用いた損失落とし戸関数の構成” 2014年暗号と情報セキュリティシンポジウム(SCIS2014), 3E2-1, 鹿児島, 2014年1月.
5. 花谷嘉一，山川高志，花岡悟一郎，松田隆宏，“二つの安全性モデルにおいて標準的な仮定への緊密な帰着を持つ効率的な鍵カプセル化メカニズム” 2014年暗号と情報セキュリティシンポジウム(SCIS2014), 4E2-3, 鹿児島, 2014年1月.
6. 藤原育真，笠松宏平，山川高志，花岡 悟一郎，今井 秀樹，“CDH,DDH,GDH仮定の関係性について” 2013年暗号と情報セキュリティシンポジウム(SCIS2013), 3B2-1, 京都, 2013年1月.
7. 大竹剛，広中悠樹，加井謙二郎，遠藤洋介，花岡悟一郎，渡辺創，山田翔太，笠松宏平，山川高志，今井秀樹，“Wildcardを部分的に許す効率的な属性ベース暗号の構成に関する検討” 2013年暗号と情報セキュリティシンポジウム(SCIS2013),

3F4-3, 京都, 2013 年 1 月．

8. 山川高志，山田翔太，花岡悟一郎，國廣昇，“素因数分解問題に基づく Semi-smooth 部分群上の CCA 安全な公開鍵暗号の安全性証明について” 2013 年暗号と情報セキュリティシンポジウム (SCIS2013), 4B1-1, 京都, 2013 年 1 月．（SCIS 論文賞受賞）

Poster

1. Takashi Yamakawa, Shota Yamada, Goichiro Hanaoka, Noboru Kunihiro Bounded CCA-secure KEM from the computational bilinear Diffie-Hellman assumption In *IWSEC 2012* (Best Poster Award)