The Pennsylvania State University
The Graduate School
College of Engineering

THE NEW SECURITY BOUNDS AND LEAKAGE-RESILIENT MODELS FOR ENCRYPTION SCHEMES

A Dissertation in
Computer Science and Engineering
by
Ye Zhang

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2015

The dissertation of Ye Zhang was reviewed and approved* by the following:

Adam Smith
Associate Professor of Computer Science and Engineering
Dissertation Advisor
Chair of Committee

Martin Fürer
Professor of Computer Science and Engineering

Trent Jaeger
Professor of Computer Science and Engineering

Jason Morton
Assistant Professor of Mathematics

Lee Coraor
Associate Professor of Computer Science and Engineering
Director of Academic Affairs

*Signatures are on file in the Graduate School.

Abstract

One of the earliest and most important tasks in cryptography is to encode messages in a way that only authorized parties can read them. Encryption is the process of doing so. In an encryption scheme, an encryption key and a decryption key are first generated. Given the encryption key and a message m, the encryption scheme generates the encoded message: the ciphertext C. Given decryption key and a ciphertext C, the encryption scheme can recover the original message.

In this thesis, we study some problems about security bounds and leakage-resilient models for encryption schemes, by showing tighter security bounds for encryption schemes, devising efficient constructions in the existing security models and proposing more powerful security models:

PKCS #1 v1.5 is a long-standing and a widely used standard that defines a set of encryption schemes. Ciphertext Indistinguishability under Chosen Plaintext Attack (IND-CPA) is one of the most popular security definitions for public key encryption. In this thesis, first, we show the encryption scheme in PKCS #1 v1.5 is IND-CPA secure for messages of length roughly 8 times as long as in the previous work.

Second, we consider securely updating encryption keys in a security co-processor where information could be leaked to an attacker periodically. We devise a leakage-resilient key evolution scheme to address the problem. Our construction can update keys in a near-linear time in n, where n is the length of key. Previous work on this problem updates keys in time $\Theta(n^2)$. Our security analysis uses new results on the connectivity of random graphs.

Third, we consider the *auxiliary input model*, which captures settings where the attacker has some information about the decryption key. Previous work only considered public-key encryption (PKE) in this model. In this thesis, we devise the first secure identity-based encryption (IBE) construction in this model. IBE is more flexible than PKE in the choice of public keys. This makes it much more useful, but harder to achieve. We also extend the auxiliary model to a stronger model by allowing the attackers to have some information about the randomness that is used to generate ciphertexts. This new model is important as in some cases (e.g., cloud computing), the randomness used by encryptors (e.g., data owners) is weak. We devise secure IBE and PKE constructions in this new model.

Table of Contents

List of Figures

Acknowledgments

The author would like to thank Prof. Adam Smith for providing valuable guidance when the author works toward his PhD thesis. The author also would like to thank Dr. Wai-Kit Wong, Prof. Siu-Ming Yiu, Prof. Nikos Mamoulis, Prof. David W. Cheung, Prof. Chun Jason Xue, Prof. Duncan S. Wong, Dr. Tsz Hon Yuen, Prof. Sherman S.M. Chow, Dr. Juan Garay, Dr. Payman Mohassel and Dr. Joseph Liu for valuable discussions and being a part of the work discussed in this thesis.

Dedication

To my parents.

Chapter 1
Introduction

One of the earliest and most important tasks in cryptography is encryption, the process of encoding messages in a way that only authorized parties can read them. Though it has been studied for hundreds of years, some very important problems about encryption remain unsolved. First, for many efficient encryption schemes that are widely used today, we do not know any attacks to them nor do we have rigorous analysis showing they are secure. To prove their security against realistic adversaries under any well-established assumptions is an interesting and important problem. It also helps us to understand more fundamental questions – the tradeoffs between security and efficiency. Second, many encryption schemes are proved to be secure using traditional security definitions (e.g., IND-CPA security, which will be discussed later). However, those traditional security definitions do not capture all real-world attacks, especially, a class of attacks called *side-channel attacks* where part of the secret key is leaked to an attacker. To devise secure and efficient encryption schemes that resist those attacks becomes an interesting and important problem.

Let $\lambda \in \mathbb{N}^+$ be an integer, called the security parameter. An encryption scheme consists of three (randomized) algorithms that run in polynomial time that grows slowly (at most polynomial) in λ. Specifically, given λ, a key generation algorithm generates a pair of keys (strings in $\{0,1\}^*$), one for encryption and one for decryption. The encryption key may or may not be identical to the decryption key. In some cases, e.g., in public-key encryption, it should be computationally hard to derive one from the other. The encryption algorithm that takes an encryption key ek and a message m outputs an encoded message: the ciphertext C. Given a decryption key dk, a deterministic polynomial-time "decryption" algorithm takes a ciphertext and outputs a message. Without a proper decryption key, it should be computationally hard to extract any information (other than the length) about the message from its ciphertext C. The security parameter λ measures the "security" of the scheme.

Roughly, the running time of all attacks that break the scheme should be at least 2^λ. For example, the security parameters in RSA-2048 and AES-128 encryption schemes are roughly 80 (bits).

We can classify encryption schemes into public-key encryption (PKE) and symmetric-key encryption. As the name suggests, in symmetric-key encryption, the encryption key is identical to the decryption key. All parties involved in the communication share the same secret key, which cannot be made public. Examples of symmetric-key encryption schemes include DES (Date Encryption Standard) [DES], 3DES [3DE] and AES (Advance Encryption Standard) [AES]. In public-key encryption, there is a public key (as the encryption key) and a private key (as the decryption key). To derive the private key from the public key is computationally hard. Given a public key, one can encrypt a message and given a private key, one can decrypt a ciphertext. However, given the public key only, one cannot decrypt ciphertexts. Examples of public-key encryption schemes include RSA [RSA78a], ElGamal [ElG85], elliptic curve encryption [Kob87, Mil85] and identity-based encryption [BF01, Coc01] .

In this thesis, we study security of some encryption schemes in the RSA family. We also show how to protect a symmetric (encryption) key when the key is computed inside a security co-processor where a certain amount information can be leaked to the outside with a limited rate (e.g. 20 bits per second). Before giving a detailed explanation of our contributions, we provide some background first.

1.1 Background

In modern cryptography, we use a security model to capture the abilities of a potential attacker. IND-CPA (ciphertext indistinguishability under chosen plaintext attack) security [KL07] captures the intuition that no probabilistic polynomial-time (PPT) adversary should extract any information about the original message other than its length from the ciphertext with noticeable probability. This problem can be reduced in polynomial time to distinguish the ciphertexts of one message m_0 from ciphertexts of another message m_1, which is a single bit information (0 or 1). More specifically, two equal-length messages m_0 and m_1 ($m_0 \neq m_1$) are chosen by the adversary $\mathcal{A}$. Then, a ciphertext C^* (challenge ciphertext) will be given to $\mathcal{A}$. C^* is generated by either encrypting m_0 or m_1 with equal probability. $\mathcal{A}$ needs to decide if C^* corresponds to m_0 or m_1 with probability significantly higher than 1/2. Even though $\mathcal{A}$ knows both m_0 and m_1, it does not know the random string used to generate C^*.

A good encryption scheme shall use the randomness to hide this single bit of information computationally. We can show that hiding this single bit information is computationally equivalent to showing that no algorithm can extract any information other than the length of the original message. It is easy to prove security against IND-CPA. Note that IND-CPA security also captures the scenarios where people use public-key encryption. Therefore, IND-CPA becomes the standard notation of security for public key encryption (PKE). A formal definition of IND-CPA security will be given in the next chapter.

IND-CCA2 (ciphertext indistinguishability under adaptively chosen ciphertext attack) [KL07] is identical to IND-CPA except that the adversary is given an additional oracle $\mathcal{O}(.) := \mathtt{Dec}(\mathtt{dk}, .)$. $\mathcal{O}(.)$ can access the challenge ciphertext C^*. Given a string $s \in \{0,1\}^*$ (that may or may not form a valid ciphertext), the decryption oracle $\mathcal{O}(.)$ first checks if s is identical to C^* or not. If it is not, the oracle returns $\mathtt{Dec}(\mathtt{dk}, s)$. It is necessary to do this check, otherwise a trivial attack can be launched by calling $\mathcal{O}(C^*) = m_b$. IND-CCA2 was largely considered to be a theoretical concern until 1998, when Daniel Bleichenbacher [Ble98] showed a practical IND-CCA2 attack on the PKCS #1 v1 scheme.

Under a given model (e.g., IND-CPA security), an encryption scheme is said to be provably secure if the scheme is capable of withstanding the attacks from adversaries with the abilities captured by the model. But if the adversary has some extra abilities, the security of the scheme is no longer guaranteed. In most traditional security models (e.g., IND-CPA, IND-CCA1, IND-CCA2 etc.), it is assumed that the adversary does not have the ability to obtain any information (even one single bit) about the secret key. However, due to the advancement of a large class of side-channel attacks (e.g., [Koc96, BS97, MDS99, KJJ99, HSHea08, GPT14]) on the physical implementation of cryptographic schemes, obtaining partial information of the secret key becomes feasible and easier.

For example, let (N, d) be the private key of an RSA encryption scheme. Given a ciphertext C, RSA decryption outputs $C^d \bmod N$. A timing attack can be launched as follows. An attacker can monitor the execution time of decryption. Since the execution time for the square-and-multiply algorithm used in the exponentiation depends linearly on the number of 1 bits in the private key, the attacker is able to find different timing patterns on bit 0 and bit 1. This enables it to recover d and to break IND-CPA security. Thus, the assumption for absolute secrecy of the secret key may not hold. In recent years, a number of works have been done in leakage-resilient cryptography to formalize these attacks in the security model.

Leakage-resilient cryptography models various side-channel attacks by allowing the adversary to specify an arbitrary and efficiently computable function f, and to obtain the

output of f (representing the information leaked) applied to the secret key sk. Clearly, we must have some restrictions on f so that the adversary should not be able to recover sk completely. A common model is to bound the number of leaked bits (such as with "relative leakage", e.g. [AGV09] or "bounded retrieval", e.g. [DP08, ADN+10, CDRW10]). For example, in [AGV09], the output size of f is at most ℓ bits such that ℓ must be less than $|\mathsf{sk}|$. More general models bound only computational difficulty of guessing the secret given the leakage (e.g. [DGK+10, YCZY12b]). For example, Naor and Segev [NS09] considered the entropy of sk and required that the decrease in entropy of sk is at most ℓ bits upon observing $f(\mathsf{sk})$. Dodis *et al.* [DKL09] further generalized the leakage functions and proposed the model of auxiliary input which only requires the leakage functions to be computationally hard to compute sk given $f(\mathsf{sk})$. Subsequent work investigated leakage that occurs continually over many time steps (e.g. [DHLAW10, BKKV10, LLW11]). Schemes with a deterministic update are vulnerable to leakage on future keys [DP08, YSPY]; this leads naturally to the models of restricted leakage mentioned above, including the DKW model [DKW].

1.2 Our Contributions

In this thesis, we study the problems about security bounds and leakage resilient models of encryption schemes. Specifically, we provide the following results.

New Security Bounds on PKCS #1 and Simultaneously Hardcore Bits. PKCS #1 (Public Key Cryptography Standard #1: RSA Cryptography Standard) is proposed by RSA Security Inc, which has 4 public versions right now (v1.5, 2.0, 2.1 and 2.2) that define a set of encryption schemes that are variant to the RSA encryption scheme. For example, PKCS #1 v1.5 applies a random padding r to the original message m and then applies the RSA function $(m||r)^e \bmod N$ (where "||" denotes concatenation) to generate the ciphertext, where (e, N) is the public key. Until very recently [LOS13], there was no IND-CPA security proof for PKCS #1 v1.5 under any well-understood assumption, even though the parameters can be chosen arbitrarily. Lewko et al. in [LOS13] show that if the length of message m is less than $\frac{logN}{32}$ (recall that $N = pq$) [1] , the encryption scheme in PKCS #1 v1.5 is IND-CPA secure under Φ-Hiding assumption. Their results are based on the latest estimation to Gauss sums [HBK00a, BGK06].

In this thesis, we also show that the encryption scheme in PKCS #1 v1.5 is IND-CPA secure under Φ-Hiding Assumption. However, we could now support the length of message

[1] Logarithms are to the base 2.

m to be less than $\frac{logN}{4}$. Theoretically, this is a 8-fold improvement. Concretely, we show in the setting where N is of length 8192 bits and the security level is 80 bits, results from this thesis support 1735-bit messages, which is a 13-fold improvement compared with Lewko et al.'s results (128 bit at the same setting).

Simultaneously hardcore bits for the RSA problem can be described as follows. Let x be a random number chosen uniformly from $\mathbb{Z}_N$ ($N = pq$). Given N, e and $x^e \bmod N$, the question is which part of x still looks random computationally? Assuming that RSA is hard to invert, only λ bits (that is $O(\sqrt[3]{\log N})$) are simultaneously hardcore, where 2^λ is the time to invert. Lewko et al. [LOS13] showed $\log e - O(\log \frac{1}{\epsilon})$ bits of RSA are simultaneously hardcore. However in this thesis, we show their results are incorrect, as one key lemma in their paper is incorrect. In fact, we prove a weaker version of the claim which is nonetheless sufficient for most, though not all, of their applications. For example, we show that the most (or least) $\log e - 2\log \frac{1}{\epsilon} - 2$ (that is $O(\log N)$) significant bits of RSA are simultaneously hardcore.

Leakage-Resilient Key Evolution Schemes. A key evolution scheme $y_{i+1} = \mathcal{KE}(y_i)$ given the i-th round key y_i, outputs the $(i+1)$-th round key y_{i+1}. It is required that y_{i+1} needs to be computationally indistinguishable from a random string (pseudorandom), even if some information about on y_i can be leaked (e.g, via side-channel attacks) during the i-th round. We consider a side-channel model where the key evolution scheme is carried inside a secure co-processor. Initially, a random key y_0 is securely downloaded in the co-processor. Side-channel attacks are modeled as a small adversary inside the co-processor, that has limited storage s (bits) and limited external communication c (bits) for each period of time. The adversary inside the co-processor is restricted to run in polynomial time. The key has to be updated periodically (at the end of each round) to prevent it from being compromised.

In this thesis, we show a secure key evolution scheme in this attack model under the random oracle model. A random oracle model assumes that a deterministic function that takes an input string and outputs a truly random string exists. In order to analyze the time complexity and security properties easily, we use a graph for describing the key evolution scheme. We also use the technique of pebbling games (e.g., see [DNW]) to connect the complexity of coloring a certain set of vertices in the graph, with the security properties in the random oracle model.

The existing work [DKW] to this problem updates keys in time $O(n^2)$ where n is the key length. Their construction is based on a simple $n \times n$ grid graph. In this thesis, we construct a scheme that has a quasilinear time complexity in n. The results rely on stacks

of n-superconcentrators [Pip77, LT82]. It is non-trivial to show the scheme (this thesis) has a quasilinear time complexity and is provable secure. The n-superconcentrator must have some combinatorial properties. For example, the n-superconcentrator is built from an ϵ-local bipartite graph that is also $(4n/5, A > 1)$-vertex expander [Vad]. The definitions of ϵ localness and vertex expanders can be found in Chapter 4.

Auxiliary Input Models and Leakage-Resilient Encryption Schemes. We also consider the side-channel attacks as any one-way functions. Let x be chosen uniformly at random. Loosely speaking, f is a one-way function if given $f(x)$, it is computationally hard to find x' such that $f(x') = f(x)$. This is called auxiliary input model in the previous work [DGK$^+$10], but [DGK$^+$10] only considered this model in the public key encryption setting.

In this thesis, we consider the auxiliary input models for identity-based encryption (the new model combines IND-ID-CPA [BF01] and auxiliary input is called IND-ID-AI-CPA security). Our method is based on dual-system encryption that is proposed by Waters [Wat09]. In an identity-based encryption (IBE), a public key can be chosen arbitrarily. Given a public key and the master secret key (generated during setup), a key generation algorithm can generate a private key for the public key.

In addition to auxiliary-input models, we also propose the post-auxiliary-input (pAI) model that works as follows. The adversary can query any auxiliary information (modeled as one-way functions) on the secret key before seeing the challenge ciphertext (this is identical to auxiliary input model). After seeing the challenge ciphertext, the adversary is allowed to query a set of restricted functions (but still one-way) over r^* where r^* is the encryption randomness for the challenge ciphertext. Combining it with the traditional IND-CPA and IND-CCA2 models, in the thesis, we propose public key encryption (PKE) schemes that are both IND-pAI-CPA and IND-pAI-CCA2 secure in the post-auxiliary models. We also devise an IND-ID-pAI-CPA secure IBE for the post-auxiliary models. Our construction is generic. It converts any IND-AI-CPA (e.g., [DGK$^+$10]) and IND-ID-AI-CPA (e.g., [YCZY12a]) constructions into their post auxiliary-input secure (pAI) versions. Technically, it is based on a primitive called strong extractor with hard-to-invert auxiliary inputs that is independently proposed in this thesis. The definition of this strong extractor can be found in Chapter 5.

1.3 Discussion

The results presented in this thesis are based on pseudorandom objects (for a survey, see [Vad]). They are very useful in designing algorithms, cryptographic schemes etc. For ex-

ample, we can apply them to de-randomize algorithms; we can also apply them to extract randomness from a source with enough entropy. Pseudorandom objects include expander graphs, list-decodable codes, randomness extractors and pseudorandom generators among others.

In this thesis, the results from PKCS #1 v1.5 can be interpreted as a deterministic extractor; the key evolution scheme is based on n-superconcentrators and vertex expander graphs; the encryption schemes in the (post) auxiliary input model are derived using a strong extractor with auxiliary input. This can be viewed as an explanation of how techniques in this thesis are connected with each other.

1.4 Organization

The rest of this thesis is organized as follows. Chapter 2 will discuss the preliminaries that are essential to the thesis. Chapter 3 will discuss the IND-CPA security of PKCS #1 v1.5. Chapter 4 will discuss secure leakage-resilient key evolution schemes and Chapter 5 will discuss the leakage-resilient encryption schemes with auxiliary input models.

Chapter 2
Preliminaries

We denote by $SD(A; B)$ the statistical distance between the distributions of random variables A and B taking values in the same set. We write $A \approx_\epsilon B$ as shorthand for $SD(A; B) \leq \epsilon$.

Given an integer $I \in \mathbb{Z}^+$, we write $[I]$ for the set $\{0, 1, 2, \ldots, I-1\}$. Thus, an arithmetic progression ("AP") of length K can be written $P = \sigma[K] + \tau$ for some $\sigma, \tau \in \mathbb{Z}$.

We consider adversaries that are restricted to probabilistic polynomial time (PPT), and let $\mathsf{negl}(k)$ be a negligible function in k, that is, one that decreases faster than the reciprocal of any polynomial. We write $A \leftarrow\!\!\$\, B$ to indicate that the random variable A is generated by running (randomized) algorithm B using fresh random coins, if B is an algorithm, or that A is distributed uniformly in B, if B is a finite set.

2.1 Computational Assumptions

It should note that for most of cryptographic tasks, we cannot prove security without assuming that there exist some hard problems that cannot be solved in probabilistic polynomial time. We call those assumptions of hard problems as computational assumptions, or simply hard assumptions. In this thesis, we need the Φ-Hiding Assumptions to show the results in Chapter 3.

Let θ be an even integer and $c \in (0, 1)$ be a constant. We define two alternate parameter generation algorithms for RSA keys:

Algorithm $RSA^{\text{inj}}_{c,\theta}(1^k)$:	**Algorithm** $RSA^{\text{loss}}_{c,\theta}(1^k)$
$e \leftarrow\!\!\$\ \mathsf{Primes}_{ck}$	$e \leftarrow\!\!\$\ \mathsf{Primes}_{ck}$
$(N, p, q) \leftarrow\!\!\$\ \mathsf{RSA}_k$	$p \leftarrow\!\!\$\ \mathsf{Primes}_{\frac{k}{2}-\frac{\theta}{2}}[p = 1 \bmod e]$
Return (N, e)	$q \leftarrow\!\!\$\ \mathsf{Primes}_{\frac{k}{2}+\frac{\theta}{2}}$
	Return (pq, e)

Definition 1 ((c,θ)-Φ-Hiding Assumption (ΦA))**.** Let θ, c be parameters that are functions of the modulus length k, where $\theta \in \mathbb{Z}^+$ is even and $c \in (0, 1)$. For any probabilistic polynomial-time distinguisher $\mathcal{D}$,

$$\mathbf{Adv}^{\Phi A}_{c,\theta,\mathcal{D}}(k) = \left|\Pr[\mathcal{D}(RSA^{\text{inj}}_{c,\theta}(1^k)) = 1] - \Pr[\mathcal{D}(RSA^{\text{loss}}_{c,\theta}(1^k)) = 1]\right| \leq \mathsf{negl}(k).$$

where $\mathsf{negl}(k)$ is a negligible function in k.

2.2 Random Oracle Models

Random Oracle (RO) models are introduced by Bellare and Rogaway [BR93a]. They assume that there exist oracles (functions) $\mathcal{O}(\cdot)$ that are accessible by the adversary whose outputs are drawn uniformly random. For example, we have a_1, a_2, a_3 where $a_1 = a_3$ and $a_1 \neq a_2$. By calling $\mathcal{O}(a_1), \mathcal{O}(a_2), \mathcal{O}(a_3)$, the value $\mathcal{O}(a_1) = \mathcal{O}(a_3)$ (as random oracles are functions); however, the values of $\mathcal{O}(a_1)$ and $\mathcal{O}(a_2)$ should be drawn uniformly random.

It turns out that random oracle modes are very useful in cryptography. For example, we can let the challenger first pick up a random number y^* and maintain a table. Once the adversary asks for $\mathcal{O}(x)$, the challenger will lookup its table. If x is recorded, it returns its output in the table; if $x = x^*$ (x^* is a magic number decided by the challenger), it returns y^*; otherwise, the challenger draw an uniformly random number as the output. In this thesis, we will see how to reduce the problem of coloring a graph to the problem of pebbling game, via random oracle models. Then, in order to show security of our graph-based solution, we only need to show there are some certain properties on the graph, which is relatively easy and straightforward.

2.3 IND-CPA Security Definitions

Let λ be a security parameter. Denote the message space as $\mathcal{M}$. A public-key encryption scheme Π consists of three PPT algorithms:

- $\mathtt{Gen}(1^\lambda)$: On input the security parameter λ, output a public key pk and a secret key sk.
- $\mathtt{Enc}(\mathsf{pk}, M)$: Denote the message space as $\mathcal{M}$. On input a message $M \in \mathcal{M}$ and pk, output a ciphertext C.
- $\mathtt{Dec}(\mathsf{sk}, C)$: On input sk and C, output the message M or $\perp$ for invalid ciphertext.

For *correctness*, we require $\mathtt{Dec}(\mathsf{sk}, \mathtt{Enc}(\mathsf{pk}, M)) = M$ for all $M \in \mathcal{M}$ and $(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathtt{Gen}(1^\lambda)$.

Let λ be the security parameter and $\Pi = (\mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec})$ be a public-key encryption scheme. Let $\mathcal{A}$ (the adversary) be any algorithms with the running time $<= t(\lambda)$. Let $\mathcal{C}$ (the challenge) be a PPT algorithm. The (t, ϵ)-IND-CPA (Ciphertext Indistinguishability under Chosen Plaintext Attacks) security game can be defined as follows.

1. Initially, $\mathcal{C}$ runs $\mathtt{Gen}(1^\lambda)$ to generate sk and pk. pk is given to $\mathcal{A}$.
2. $\mathcal{A}$ submits two message $m_0, m_1 \in \mathcal{M}$ to $\mathcal{C}$ where $m_0 \neq m_1$ and $|m_0| = |m_1|$.
3. $\mathcal{C}$ flips a random coin $b \leftarrow\!\!\$\ \{0, 1\}$ and generates the challenge ciphertext $C^* \leftarrow\!\!\$\ \mathtt{Enc}(\mathsf{pk}, m_b)$. C^* is given to $\mathcal{A}$.
4. $\mathcal{A}$ outputs its guess bit $b' \in \{0, 1\}$.

The advantage against the above IND-CPA game and $\mathcal{A}$ is defined as

$$\mathbf{Adv}^{\text{IND-CPA}}_{\mathcal{A}}(1^\lambda) = \left| \Pr[b = b'] - \frac{1}{2} \right|.$$

Definition 2 ((t, ϵ)-IND-CPA Security)**.** Let $\Pi = (\mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec})$ be a public-key encryption. Π is (t, ϵ)-IND-CPA secure if for any $\lambda \in \mathbb{N}^+$ and any adversary $\mathcal{A}$ with running time $\leq t(\lambda)$:

$$\mathbf{Adv}^{\text{IND-CPA}}_{\mathcal{A}}(1^\lambda) < \epsilon(\lambda).$$

Note that the above definition, two functions $t(\lambda)$ and $\epsilon(\lambda)$ are required to be concrete. If we interest on IND-CPA security itself, we can simplify the above definition by assuming $t(\lambda)$ is the class of probabilist polynomial time and $\epsilon(\lambda)$ is a negligible function in λ:

Definition 3 (IND-CPA Security)**.** Let $\Pi = (\mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec})$ be a public-key encryption. Π is IND-CPA secure if for any $\lambda \in \mathbb{N}^+$ and any probabilistic polynomial time adversary $\mathcal{A}$,

there exists a negligible function $\mathsf{negl}(\lambda)$, such that

$$\mathbf{Adv}_{\mathcal{A}}^{\text{IND-CPA}}(1^{\lambda}) < \mathsf{negl}(\lambda).$$

Chapter 3
Improved Security Bounds on Padding-Based Encryption

Cryptographic schemes based on the RSA trapdoor permutation [RSA78b] are ubiquitous in practice. Many of the schemes, are simple, natural and highly efficient. Unfortunately, their security is often understood only in the random oracle model [BR93b], if at all.[1] When can the security of natural constructions be proven under well-defined and thoroughly studied assumptions? For example, consider the "simple embedding" RSA-based encryption scheme (of which RSA PKCS #1 v1.5, which is still in wide use, is a variant): given a plaintext x, encrypt it as $(x\|R)^e \bmod N$, where R is a random string of appropriate length and '$\|$' denotes string concatenation. Until recently [LOS13], there was no proof of security for this scheme under a well-understood assumption. The security of this scheme under chosen plaintext attacks is closely related to another fundamental question, namely, whether many physical bits of RSA are simultaneously hardcore [ACGS88, AGS03].

Indistinguishability of RSA on Arithmetic Progressions. Both of these questions are related to the hardness of a basic computational problem, which we dub *RSA-AP*. Consider a game in which a distinguisher is first given an RSA public key (N, e) and a number K. The distinguisher then selects the description of an arithmetic progression (abbreviated "AP") $\mathcal{P} = \{\sigma i + \tau \mid i = 0, \ldots, K-1\}$ of length K. Finally, the distinguisher gets a number $Y \in \mathbb{Z}_N$, and must guess whether Y was generated as $Y = X^e \bmod N$, where X is uniform in the AP P, or Y was drawn uniformly from $\mathbb{Z}_N$. We say RSA-AP is hard for length K (where K may depend on the security parameter) if no polynomial-time distinguisher can win this game with probability significantly better than it could by random guessing.

[1] There are many RSA-based constructions without random oracles, e.g., [BG85, HK09, HW09], but they are less efficient and not currently widely used.

Hardness statements for the RSA-AP problem have important implications. For example, in the "simple embedding" scheme above, the input to the RSA permutation is $x\|R$, which is distributed uniformly over the AP $\{x2^\rho + i \mid i = 0 \ldots, 2^\rho - 1\}$ where ρ is the bit length of R. If RSA-AP is hard for length 2^ρ, then $(x\|R)^e \bmod N$ is indistinguishable from uniform for all messages x and so simple embedding is CPA secure.

In this thesis, we show that RSA-AP is hard under well-studied assumptions, for much shorter lengths K than was previously known. From this, we draw conclusions about classic problems (the CPA security of PKCS #1 v1.5 and the simultaneous hardcoreness of many physical bits of RSA) that were either previously unknown, or for which previous proofs were incorrect.

Φ-Hiding, Lossiness and Regularity. The Φ-Hiding assumption, due to [CMS99], states that it is computationally hard to to distinguish standard RSA keys—that is, pairs (N, e) for which $\gcd(e, \Phi(N)) = 1$—from *lossy* keys (N, e) for which $e \mid \Phi(N)$. Under a lossy key, the map $x \mapsto x^e$ is not a permutation: if $N = pq$ where p, q are prime, e divides $p - 1$ and $\gcd(e, q - 1) = 1$, then $x \mapsto x^e$ is e-to-1 on $\mathbb{Z}_N^*$. The Φ-Hiding assumption has proven useful since under it, statements about *computational* indistinguishability in the real world (with regular keys) may be proven by showing the *statistical* indistinguishability of the corresponding distributions in the "lossy world" (where $e \mid \Phi(N)$) [KOS10, KK12, LOS13].

Specifically, [LOS13] showed that under Φ-Hiding, the hardness of RSA-AP for length K is implied by the approximate *regularity* of the map $x \mapsto x^e$ on arithmetic progressions when $e \mid \phi(N)$. Recall that a function is regular if it as the same number of preimages for each point in the image. For positive integers e, N and K, let $Reg(N, e, K, \ell_1)$ denote the maximum, over arithmetic progressions P of length K, of the statistical difference between $X^e \bmod N$, where $X \leftarrow_\$ P$, and a uniform e-th residue in $\mathbb{Z}_N$. That is,

$$Reg(N, e, K, \ell_1) \stackrel{def}{=} \max \left\{ SD(X^e \bmod N; U^e \bmod N) \;\middle|\; \begin{array}{l} \sigma \in \mathbb{Z}_N^*, \tau \in \mathbb{Z}_N, \\ X \leftarrow_\$ \{\sigma i + \tau \mid i = 0, \ldots, K-1\}, \\ U \leftarrow_\$ \mathbb{Z}_N \end{array} \right\}$$

Note that the maximum is taken over the choice of the AP parameters σ and τ. We can restrict our attention, w.l.o.g., to the case where $\sigma = 1$ (see Chapter 2); the maximum is thus really over the choice of τ.

[LOS13] observed that if $Reg(N, e, K, \ell_1)$ is negligible for the lossy keys (N, e), then Φ-Hiding implies that RSA-AP is hard for length K. Motivated by this, they studied the regularity of lossy exponentiation on arithmetic progressions. They claimed two types of

bounds: average-case bounds, where the starting point τ of the AP is selected uniformly at random, and much weaker *worst-case* bounds, where τ is chosen adversarially based on the key (N, e).

3.0.1 Our Contributions

We provide new, *worst-case* bounds on the regularity of lossy exponentiation over $\mathbb{Z}_N$. These lead directly to new results on the hardness of RSA-AP, the CPA-security of simple padding-based encryption schemes, and the simultaneous hardcoreness of physical RSA bits. In addition, we provide a corrected version of the incorrect bound from [LOS13] which allows us to recover some, though not all, of their claimed results.

Notice that in order to get any non-trivial regularity for exponentiation, we must have $K \geq N/e$, since there are at least N/e images. If the e-th powers of different elements were distributed uniformly and independently in $\mathbb{Z}_N$, then in fact we would expect statistical distance bounds of the form $\sqrt{\frac{N}{eK}}$. The e-th powers are of course not randomly scattered, yet we recover this type of distance bound under a few different conditions.

Our contributions can be broken into three categories:

Worst-case bounds (Section 3.2). We provide a new worst-case bound on the regularity of exponentiation for integers with an unbalanced factorization, where $q > p$. We show that

$$Reg(N, e, K, \ell_1) = O\Big(\tfrac{p}{q} + \sqrt{\tfrac{N}{eK}}\Big). \tag{3.1}$$

When q is much larger than p, our bound scales as $\sqrt{\frac{N}{eK}}$. This bound is much stronger than the analogous worst-case bound from [LOS13], which is $\tilde{O}\Big(\sqrt{\frac{N}{eK}} \cdot \sqrt{\frac{N}{K}} \cdot \sqrt[8]{pe}\Big)$ (where $\tilde{O}(\cdot)$ hides polylogarithmic factors in N).[2] In particular, we get much tighter bounds on the security of padding-based schemes than [LOS13] (see "Applications", below).

Applying our new bounds requires one to assume a version of the Φ-Hiding assumption in which the "lossy" keys are generated in such a way that $q \gg p$ (roughly, $\log(q) \geq \log(p) + \lambda$ for security parameter λ). We dub this variant the *unbalanced* Φ-hiding assumption.

Average-case bounds (Section 3.3). We can remove the assumption that lossy keys have different-length factors if we settle for an average-case bound, where the average is

[2]The bound of [LOS13] relies on number-theoretic estimates of *Gauss sums*. Under the best known estimates [HBK00b], the bound has the form above. Even under the most optimistic number-theoretic conjecture on Gauss sums (the "MVW conjecture" [MVW95]), the bounds of [LOS13] have the form $\tilde{O}(\sqrt{\frac{N}{eK}} \cdot \sqrt{\frac{N}{K}})$ and are consequently quite weak in the typical setting where $K \ll N$.

taken over random translations of an arithmetic progression of a given length. We show that if X is uniform over an AP of length K, then

$$\mathop{\mathbb{E}}_{c \leftarrow\!\$\, \mathbb{Z}_N} \Big(SD\Big((c+X)^e \bmod N \;;\; U^e \bmod N \Big) \Big) = O\Big(\sqrt{\tfrac{N}{eK}} + \tfrac{p+q}{N} \Big),$$

where U is uniform in $\mathbb{Z}_N^*$. The expectation above can also be written as the distance between the pairs $(C, (C+X)^e \bmod N)$ and $(C, U^e \bmod N)$, where $C \leftarrow\!\$\, \mathbb{Z}_N$. This average-case bound is sufficient for our application to simultaneous hardcore bits.

This result was claimed in [LOS13] for *arbitrary* random variables X that are uniform over a set of size K. The claim is false in general (to see why, notice that exponentiation by e does not lose any information when working modulo q, and so $X \bmod q$ needs to be close to uniform in $\mathbb{Z}_q$). However, the techniques from our worst-case result can be used to prove the lemma for arithmetic progressions (and, more generally, for distributions X which are high in min-entropy and are distributed uniformly modulo q).

Applications (Section 3.4). Our bounds imply that, under Φ-Hiding, the RSA-AP problem is hard roughly as long as $K > \frac{N}{e}$. This, in turn, leads to new results on the security of RSA-based cryptographic constructions.

1. Simple encryption schemes that pad the message with a random string before exponentiating (including PKCS #1 v1.5) are semantically secure under unbalanced Φ-hiding as long as the random string is more than $\log(N) - \log(e)$ bits long (and hence the message is roughly $\log(e)$ bits). In contrast, the results of [LOS13] only apply when the message has length at most $\frac{\log(e)}{16}$.[3]

   Known attacks on Φ-Hiding fail as long as $e \ll \sqrt{p}$ (see "Related Work", below). Thus, we can get security for messages of length up to $\frac{\log(N)}{4}$, as opposed to $\frac{\log(N)}{32}$. For example, when N is 8192 bits long, our analysis supports messages of 1735 bits with 80-bit security, as opposed to 128 bits [LOS13].

2. Under Φ-hiding, the $\log(e)$ most (or least) significant input bits of RSA are simultaneously hardcore. This result follows from both types of bounds we prove (average- and worst-case). If we assume only that RSA is hard to invert, then the best known reductions show security only for a number of bits proportional to the security parameter (e.g., [AGS03]), which is at most $\sqrt[3]{\log N}$.

[3] Even under the MVW conjecture (see footnote 2), one gets security for messages of at most $\frac{\log(e)}{2}$ bits.

[LOS13] claimed a proof that *any* contiguous block of about $\log(e)$ physical bits of RSA is simultaneously hardcore. Our corrected version of their result applies on to the most or least significant bits, however. Proving security of other natural candidate hardcore functions remains an interesting open problem.

3.0.2 Techniques

The main idea behind our new worst-case bounds is to lift an average-case bound over the smaller ring $\mathbb{Z}_p$ to a worst-case bound on the larger ring $\mathbb{Z}_N$. First, note that we can exploit the product structure of $\mathbb{Z}_N \equiv \mathbb{Z}_q \times \mathbb{Z}_p$ to decompose the problem into mod p and mod q components. The "random translations" lemma of [LOS13] *does* work over $\mathbb{Z}_p$ (for p prime), even though it is false over $\mathbb{Z}_N$. The key observation is that, when the source X is drawn from a long arithmetic progression, the mod q component (which is close to uniform) acts as a random translation on the mod p component of X.

More specifically, let $V = [X \bmod q]$ denote the mod q component of X (drawn from an arithmetic progression of length much greater than q) and, for each value $v \in \mathbb{Z}_q$, let X_v denote the conditional distribution of X given $V = v$. Then

$$X_v \approx X_0 + v\,.$$

That is, X_v is statistically close to a translation of the shorter but sparser AP X_0 (namely, elements of the original AP which equal 0 modulo q). In the product ring $\mathbb{Z}_q \times \mathbb{Z}_p$, the random variable X is thus approximated by the pair

$$(\underbrace{V}_{\in \mathbb{Z}_q}, \underbrace{X_0 + V}_{\in \mathbb{Z}_p})\,.$$

Since V is essentially uniform in $\mathbb{Z}_q$, its reduction modulo p is also close to uniform in $\mathbb{Z}_p$ when $q \gg p$. This allows us to employ the random translations lemma in $\mathbb{Z}_p$ [LOS13] to show that $X^e \bmod N$ is close to $U^e \bmod N$.

Discussion. Our worst-case bounds can be viewed as stating that multiplicative homomorphisms in $\mathbb{Z}_N$ (all of which correspond to exponentiation by a divisor of $\phi(N)$) are deterministic extractors for the class of sources that are uniform on arithmetic progressions of length roughly the number of images of the homomorphism. This is in line with the growing body of work in additive combinatorics that seeks to understand how additive and multiplicative structure interact. Interestingly, our proofs are closely tied to the product

structure of $\mathbb{Z}_N$. The Gauss-sums-based results of [LOS13] remain the best known for analogous questions in $\mathbb{Z}_p$ when p is prime.

3.1 RSA-AP Problem and Φ-Hiding Assumption

Let Primes_t denote the uniform distribution of t-bit primes, and let $\mathsf{Primes}_t[\cdots]$ be shorthand the uniform distribution over t-bit primes that satisfy the condition in brackets. Let RSA_k denote the usual modulus generation algorithm for RSA which selects $p, q \leftarrow\!\!\$\ \mathsf{Primes}_{\frac{k}{2}}$ and outputs (N, p, q) where $N = pq$. Note that k is generally taken to be $\Omega(\lambda^3)$, where λ is the security parameter, so that known algorithms take 2^λ expected time to factor $N \leftarrow\!\!\$\ \mathsf{RSA}_k$.

The RSA-AP problem. The RSA-AP problem asks an attacker to distinguish $P^e \bmod N$ from $(\mathbb{Z}_N)^e \bmod N$. Formally, we allow the attacker to choose the arithmetic progression based on the public key (this is necessary for applications to CPA security). We define $\mathsf{RSA\text{-}AP}(1^k, K)$ to be the assumption that the two following distributions are computationally indistinguishable, for any PPT attacker $\mathcal{A}$:

Experiment $\mathsf{RSA\text{-}AP}(1^k, K)$:	**Experiment** $\mathsf{RSA\text{-}Unif}(1^k, K)$:
$(N, p, q) \leftarrow\!\!\$\ \mathsf{RSA}_k$	$(N, p, q) \leftarrow\!\!\$\ \mathsf{RSA}_k$
$(\sigma, \tau) \leftarrow \mathcal{A}(N, e)$ where $\sigma \in \mathbb{Z}_N^*$ and $\tau \in \mathbb{Z}$	$(\sigma, \tau) \leftarrow \mathcal{A}(N, e)$
$X \leftarrow\!\!\$\ \{\sigma i + \tau : i = 0, \ldots K-1\}$	$U \leftarrow\!\!\$\ \mathbb{Z}_N$
Return (N, e, X)	Return (N, e, U)

Note that without loss of generality, we may always take $\sigma = 1$ in the above experiments, since given the key (N, e) and the element $X^e \bmod N$ where X is uniform in $P = \{\sigma i + \tau : i = 0, \ldots K-1\}$, one can compute $(\sigma^{-1}X)^e \bmod N$ where σ^{-1} is an inverse of σ modulo N. The element $\sigma^{-1}X$ is uniform in $P' = \{i + \sigma^{-1}\tau : i = 0, \ldots K-1\}$, while the element $\sigma^{-1}U$ will still be uniform in $\mathbb{Z}_N$. Hence, a distinguisher for inputs drawn from P can be used to construct a distinguisher for elements drawn from P', and vice-versa.

Φ-Hiding Assumption. Let θ be an even integer and $c \in (0, 1)$ be a constant. We define two alternate parameter generation algorithms for RSA keys:

Algorithm $RSA^{\text{inj}}_{c,\theta}(1^k)$:	**Algorithm** $RSA^{\text{loss}}_{c,\theta}(1^k)$
$e \leftarrow\!\!\$\ \mathsf{Primes}_{ck}$	$e \leftarrow\!\!\$\ \mathsf{Primes}_{ck}$
$(N, p, q) \leftarrow\!\!\$\ \mathsf{RSA}_k$	$p \leftarrow\!\!\$\ \mathsf{Primes}_{\frac{k}{2}-\frac{\theta}{2}}[p = 1 \bmod e]$
Return (N, e)	$q \leftarrow\!\!\$\ \mathsf{Primes}_{\frac{k}{2}+\frac{\theta}{2}}$
	Return (pq, e)

Definition 4 ((c,θ)-Φ-Hiding Assumption (ΦA))**.** Let θ, c be parameters that are functions of the modulus length k, where $\theta \in \mathbb{Z}^+$ is even and $c \in (0,1)$. For any probabilistic polynomial-time distinguisher $\mathcal{D}$,

$$\mathbf{Adv}^{\Phi A}_{c,\theta,\mathcal{D}}(k) = \left|\Pr[\mathcal{D}(RSA^{\text{inj}}_{c,\theta}(1^k)) = 1] - \Pr[\mathcal{D}(RSA^{\text{loss}}_{c,\theta}(1^k)) = 1]\right| \leq \mathsf{negl}(k).$$

where $\mathsf{negl}(k)$ is a negligible function in k.

As mentioned in the introduction, the regularity of lossy exponentiation on AP's of length K implies, under Φ-hiding, then RSA-AP is hard:

Observation 1. *Suppose that $Reg(N, e, K, \ell_1) \leq \epsilon$ for a $1-\delta$ fraction of outputs of $RSA^{loss}_{c,\theta}(1^k)$. Then the advantage of an attacker $\mathcal{D}$ at distinguishing RSA-AP$(1^k, K)$ from* RSA-Unif(1^k) *is at most $\mathbf{Adv}^{\Phi A}_{c,\theta,\mathcal{D}}(k) + \epsilon + \delta$.*

Though the definitions above are stated in terms of asymptotic error, we state our main results directly in terms of a time-bounded distinguisher's advantage, to allow for a concrete security treatment.

3.2 Improved ℓ_1-Regularity Bounds for Arithmetic Progressions

Let $\mathcal{P} = \sigma[K] + \tau$ be an arithmetic progression where $K \in \mathbb{Z}^+$. In this section, we show that if X is uniformly distributed over an arithmetic progression, then $X^e \bmod N$ is statistically close to a uniformly random e-th residue in $\mathbb{Z}_N$. Specifically, we have the following main result:

Theorem 2. *Let $N = pq$ (p, q primes) and we assume $q > p$ and $\gcd(\sigma, N) = 1$. Let $\mathcal{P}$ be* AP *where $\mathcal{P} = \sigma[K] + \tau$ and assume that $K > q$. Let e be such that $e|p-1$ and $\gcd(e, q-1) = 1$. Then,*

$$SD(X^e \bmod N, U^e \bmod N) \leq \frac{3q}{K} + \frac{2p}{q-1} + \frac{2}{p-1} + \sqrt{\frac{N}{eK}}$$

where $X \leftarrow_{\$} \mathcal{P}$ and $U \leftarrow_{\$} \mathbb{Z}^*_N$.

Recall, from Section 2, that it suffices to prove the Theorem for $\sigma = 1$. The main idea behind the proof is as follows.

For any $v \in \mathbb{Z}_q$ and a set $\mathcal{P} \subset \mathbb{Z}_N$, we define $\mathcal{P}_v = \{x \in \mathcal{P} | x \bmod q = v\}$. First, we observe that $SD(X^e \bmod N, U^e \bmod N) \approx \mathbb{E}_{v \in \mathbb{Z}_q^*}(SD(X_v^e \bmod p, U_p^e \bmod p))$ (Lemma 3) where $U_p \leftarrow_\$ \mathbb{Z}_p^*$ and for any $v \in \mathbb{Z}_q^*$, $X_v \leftarrow_\$ \mathcal{P}_v$. Second, we show that $\mathcal{P}_v$ is almost identical to $\mathcal{P}_0 + v$ (that is the set $\mathcal{P}_0$ shifted by $v \in \mathbb{Z}_q$) (Lemma 4). Therefore, we can replace $\mathbb{E}_{v \leftarrow_\$ \mathbb{Z}_q^*}(SD(X_v^e \bmod p, U_p^e \bmod p))$ with $\mathbb{E}_{v \leftarrow_\$ \mathbb{Z}_q^*}(SD((Y+v')^e \bmod p, U_p^e \bmod p))$ where $Y \leftarrow_\$ \overline{\mathcal{P}}$. The last term can be bounded via hybrid arguments and a similar technique to [LOS13, Lemma 3] (our Lemma 6).

In order to prove this theorem, we need the following lemmas (whose proof will be given at the end of this section):

Lemma 3. *Let $N = pq$ (p, q primes). Let $\mathcal{P}$ be an* AP *where $\mathcal{P} = [K] + \tau$ and assume that $K > q$. Let e be such that $e|p-1$ and $\gcd(e, q-1) = 1$. Then,*

$$SD(X^e \bmod N, U^e \bmod N) \leq \frac{q}{K} + \mathop{\mathbb{E}}_{v \leftarrow_\$ \mathbb{Z}_q^*}(SD(X_v^e \bmod p, U_p^e \bmod p))$$

where $X \leftarrow_\$ \mathcal{P}$, $U \leftarrow_\$ \mathbb{Z}_N^*$, $U_p \leftarrow_\$ \mathbb{Z}_p^*$ and for any $v \in \mathbb{Z}_q^*$, $X_v \leftarrow_\$ \mathcal{P}_v$.

Lemma 4. *Let $N = pq$ (p, q primes). Let $\mathcal{P}$ be an* AP *where $\mathcal{P} = [K] + \tau$. For any $v \in \mathbb{Z}_q^*$, $|\mathsf{SymDiff}(\mathcal{P}_\mathsf{v}, (\mathcal{P}_\mathsf{0} + \mathsf{v}))| \leq 2$ where* SymDiff *denotes symmetric difference.*

Lemma 5. *Let $N = pq$ (p, q primes) and assume $q > p$. Let e be such that $e|p-1$ and $\gcd(e, q-1) = 1$. Let $\overline{\mathcal{K}} \subset \mathbb{Z}_N$ be an arbitrary subset (not necessarily an* AP*):*

$$\begin{aligned} &SD((C \bmod p, (C+R)^e \bmod p), (C \bmod p, U_p^e \bmod p)) \\ &\leq SD((V_p, (V_p+R)^e \bmod p), (V_p, U_p^e \bmod p)) + \frac{2p}{q-1}. \end{aligned}$$

where $C \leftarrow_\$ \mathbb{Z}_q^*, V_p, U_p \leftarrow_\$ \mathbb{Z}_p^*$ and $R \leftarrow_\$ \overline{\mathcal{K}}$.

Notice that in this lemma, the random variable C is chosen from $\mathbb{Z}_q^*$ but always appears reduced modulo p.

Roughly speaking, Lemma 5 says that if $[I]$ ($I \in \mathbb{Z}^+$; e.g., $I = q-1$) is large enough ($I > p$), we can replace $Q \bmod p$ with V_p, where $Q \leftarrow_\$ [I]$ and $V_p \leftarrow_\$ \mathbb{Z}_p^*$. Then, we can apply the random translations lemma [LOS13] over $\mathbb{Z}_p^*$ to show Lemma 6.

We should point out that the mistake in the proof of [LOS13] does not apply to Lemma 6. Specifically, the mistake in [LOS13] is due to the fact that $\omega - 1$ may not be invertible in $\mathbb{Z}_N$ where $N = pq$, $\omega^e = 1 \bmod N$ and $\omega \neq 1$ (refer Section 3.3 for more detailed explanation).

However, $\omega - 1$ is invertible in $\mathbb{Z}_p$, (since p is prime) which is the ring used in Lemma 6. Specifically, we apply the following corrected version of [LOS13, Lemma 3]:

Lemma 6 (Random Translations Lemma, adapted from [LOS13])**.** *Let $N = pq$ (p, q primes). Let $V_p, U_p \leftarrow_{\$} \mathbb{Z}_p^*$. Let $R \leftarrow_{\$} \overline{\mathcal{K}}$ where $\overline{\mathcal{K}} \subset \mathbb{Z}_N$ and $|\overline{\mathcal{K}}| = \overline{K}$.*

$$SD((V_p, (V_p + R)^e \bmod p), (V_p, U_p^e \bmod p)) \leq \frac{2}{p-1} + \sqrt{\frac{p-1}{e\overline{K}}}.$$

Proof. This proof is observed by [LOS13]. However, in [LOS13], $\omega - 1$ may not be invertible in $\mathbb{Z}_N$ (recall that $\omega \in \{x | x^e \bmod N = 1\}$) but $\omega - 1$ is invertible in $\mathbb{Z}_p$ as $e | p - 1$.

Let $\mathcal{Q}$ be the distribution of $(V, (V+X)^e \bmod p)$ and $\mathcal{T}$ be the distribution of $(V, U^e \bmod p)$. $\mathcal{Q}_0$ is identical to $\mathcal{Q}$ except that the event $(V + X)^e \bmod p = 0$ occurs; $\mathcal{T}_0$ is identical to $\mathcal{T}$ except that the event $U^e \bmod p = 0$ occurs. Similarly, $\mathcal{Q}_1$ is defined to be identical to $\mathcal{Q}$ except that $(V + X)^e \bmod p \neq 0$; $\mathcal{T}_1$ is identical to $\mathcal{T}$ except that $U^e \bmod p \neq 0$. Then, we have:

$$SD(\mathcal{Q}, \mathcal{T}) = SD(\mathcal{Q}_0, \mathcal{T}_0) + SD(\mathcal{Q}_1, \mathcal{T}_1).$$

$$\begin{aligned} SD(\mathcal{Q}_0, \mathcal{T}_0) &\leq < 1, \mathcal{Q}_0 > + < 1, \mathcal{T}_0 > \\ &\leq \frac{1}{p-1} + \frac{1}{p-1} \leq \frac{2}{p-1}. \end{aligned}$$

$$\begin{aligned} SD(\mathcal{Q}_1, \mathcal{T}_1) &\leq \sqrt{supp(\mathcal{Q}_1 - \mathcal{T}_1)||\mathcal{Q}_1||_2^2 - 1} \\ &\leq \sqrt{\frac{(p-1)^2}{e}||\mathcal{Q}_1||_2^2 - 1}. \end{aligned}$$

Where,

$$\begin{aligned} ||\mathcal{Q}_1||_2^2 &= \Pr[(V, (V+X)^e \bmod p) = (V', (V'+Y) \bmod p)] \\ &= \frac{1}{p-1} \Pr[(V+X)^e \bmod p = (V+Y)^e \bmod p] \\ &= \frac{1}{p-1} \sum_{\omega \in \{x | x^e \bmod p = 1\}} \Pr[(V+X) = \omega(V+Y) \bmod p] \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{p-1}(\Pr[X = Y \bmod p] + \sum_{\omega \neq 1} \Pr[V = (\omega-1)^{-1}(X - \omega Y) \bmod p]) \\
&= \frac{1}{p-1}(\Pr[X = Y \bmod p] + \frac{e-1}{p-1}) \\
&\leq \frac{1}{p-1}(\frac{1}{p} + \frac{1}{\overline{\overline{K}}} + \frac{e-1}{p-1}) \\
&\leq \frac{1}{p-1}(\frac{e}{p} + \frac{1}{\overline{\overline{K}}}).
\end{aligned}
$$

Therefore, we have:

$$
\begin{aligned}
&SD(\mathcal{Q}, \mathcal{T}) \\
&\leq \frac{2}{p-1} + \sqrt{\frac{p-1}{e}(e/p) - 1 + \frac{p-1}{e\overline{\overline{K}}}} \\
&\leq \frac{2}{p-1} + \sqrt{\frac{p-1}{e\overline{\overline{K}}}}.
\end{aligned}
$$

□

We can now prove our main result, Theorem 2:

Proof of Theorem 2. Let $X \leftarrow\!\!\$\, \mathcal{P}$, $U \leftarrow\!\!\$\, \mathbb{Z}_N^*$, $U_p \leftarrow\!\!\$\, \mathbb{Z}_p^*$. For any $v \in \mathbb{Z}_q$, let $X_v \leftarrow\!\!\$\, \mathcal{P}_v$ (recall $\mathcal{P}_v$ is a set $\{x \in \mathcal{P} | x \bmod q = v\}$). By Lemma 3, we have:

$$
SD(X^e \bmod N, U^e \bmod N) \leq \frac{q}{K} + \mathop{\mathbb{E}}_{v \leftarrow\!\!\$\, \mathbb{Z}_q^*}(SD(X_v^e \bmod p, U_p^e \bmod p)).
$$

Let $Y \leftarrow\!\!\$\, \mathcal{P}_0$. By the triangle inequality:

$$
\mathop{\mathbb{E}}_{v \leftarrow\!\!\$\, \mathbb{Z}_q^*} SD(X_v^e \bmod p, U_p^e \bmod p) \leq \mathop{\mathbb{E}}_{v \leftarrow\!\!\$\, \mathbb{Z}_q^*}(SD(X_v, Y + v) + SD((Y+v)^e \bmod p, U_p^e \bmod p)).
$$

Note that $SD(A^e \bmod p, B^e \bmod p) \leq SD(A, B)$ for any A and B. By Lemma 4, for every $v \in \mathbb{Z}_q^*$, we have $|\mathsf{SymDiff}(\mathcal{P}_\mathsf{v}, (\mathcal{P}_\mathsf{0} + \mathsf{v}))| \leq \mathsf{2}$. Therefore, we have $SD(X_v, Y + v) = \frac{|\mathsf{SymDiff}(\mathcal{P}_\mathsf{v}, (\mathcal{P}_\mathsf{0} + \mathsf{v}))|}{|\mathcal{P}_0|} \leq \frac{2}{|\mathcal{P}_0|} \leq \frac{2q}{K}$. Then,

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{v \leftarrow\!\!\$\, \mathbb{Z}_q^*} \Big(SD(X_v, Y + v) + SD((Y+v)^e \bmod p, U_p^e \bmod p)\Big) \\
&\leq \mathop{\mathbb{E}}_{v \leftarrow\!\!\$\, \mathbb{Z}_q^*} SD(X_v, Y + v) + \mathop{\mathbb{E}}_{v \leftarrow\!\!\$\, \mathbb{Z}_q^*} SD\left((Y+v)^e \bmod p, U_p^e \bmod p\right)
\end{aligned}
$$

$$\leq \frac{2q}{K} + \mathop{\mathbb{E}}_{v \leftarrow\$ \mathbb{Z}_q^*} SD\left((Y+v)^e \bmod p, U_p^e \bmod p\right).$$

First, note that only the reduced value of $v \bmod p$ affects the statistical distance $SD((Y + v)^e \bmod p, U_p^e \bmod p)$. so the expression above can be rewritten as:

$$\mathop{\mathbb{E}}_{v \leftarrow\$ \mathbb{Z}_q^*} SD(((Y+v)^e \bmod p, U_p^e \bmod p)) = \mathop{\mathbb{E}}_{v \leftarrow\$ \mathbb{Z}_q^*; w \leftarrow\$ v \bmod p} SD\left((Y+w)^e \bmod p, U_p^e \bmod p\right).$$

Let $U_q \leftarrow\$ \mathbb{Z}_q^*$. The expectation above can be written as the distance between two pairs:

$$\begin{aligned}&\mathop{\mathbb{E}}_{v \leftarrow\$ \mathbb{Z}_q^*; w \leftarrow\$ v \bmod p} SD((Y+w)^e \bmod p, U_p^e \bmod p)\\ &= SD\left(U_q \bmod p, (Y+U_q)^e \bmod p, (U_q \bmod p, U_p^e \bmod p)\right).\end{aligned}$$

By Lemma 3 and 4, we have $SD\left(U_q \bmod p, (R+U_q)^e \bmod p, (U_q \bmod p, U_p^e \bmod p)\right) < \frac{2p}{q-1} + \frac{2}{p-1} + \sqrt{\frac{p-1}{e|\overline{\mathcal{K}}|}}$ where $\overline{\mathcal{K}} \subset \mathbb{Z}_N$ and $R \leftarrow\$ \overline{\mathcal{K}}$. We apply with $\overline{\mathcal{K}} = \mathcal{P}_0$:

$$\begin{aligned}&SD((U_q \bmod p, (Y+U_q)^e \bmod p), (U_q \bmod p, U_p^e \bmod p))\\ &\leq \frac{2p}{q-1} + \frac{2}{p-1} + \sqrt{\frac{p-1}{e|\mathcal{P}_0|}} \leq \frac{2p}{q-1} + \frac{2}{p-1} + \sqrt{\frac{N}{eK}}.\end{aligned}$$

since $|\mathcal{P}_0| = \{\lfloor \frac{K}{q} \rfloor, \lceil \frac{K}{q} \rceil\}$. □

3.2.1 Proofs of Lemmas

We now prove the technical lemmas from previous section.

Proof of Lemma 3. The proof is done via hybrid arguments. By the Chinese Remainder Theorem, the mapping $a \mapsto (a \bmod p, a \bmod q)$ is an isomorphism from $\mathbb{Z}_N \to \mathbb{Z}_p \times \mathbb{Z}_q$. Therefore, we can rewrite $SD(X^e \bmod N, U^e \bmod N)$ as $SD((X^e \bmod p, X^e \bmod q), (U_p^e \bmod p, U_q^e \bmod q))$ where $U \leftarrow\$ \mathbb{Z}_N^*$, $U_p \leftarrow\$ \mathbb{Z}_p^*$ and $U_q \leftarrow\$ \mathbb{Z}_q^*$. Furthermore, as $\gcd(e, q-1) = 1$, $a \to a^e \bmod q$ is a 1-to-1 mapping over $\mathbb{Z}_q^*$. Therefore,

$$\begin{aligned}&SD((X^e \bmod p, X^e \bmod q), (U_p^e \bmod p, U_q^e \bmod q))\\ &= SD((X^e \bmod p, X \bmod q), (U_p^e \bmod p, U_q \bmod q).\end{aligned}$$

Now, we define $T_0 = (X \bmod q, X^e \bmod p)$, $T_1 = (U_q, X_{U_q}^e \bmod p)$ and $T_2 = (U_q, U_p^e \bmod$

p) where X_{U_q} is the random variable that chooses $v \leftarrow_\$ \mathbb{Z}_q^*$ and then $X_{U_q} \leftarrow_\$ \mathcal{P}_v$. By the triangle inequality (hybrid arguments),

$$SD(T_0, T_2) \leq SD(T_0, T_1) + SD(T_1, T_2)$$

where we have $SD(T_1, T_2) = \mathbb{E}_{v \in \mathbb{Z}_q^*} SD((X_v^e \bmod p, U_p^e \bmod p)$.

Now, we bound $SD(T_0, T_1)$. Define $T_0' = (W \bmod q, X^e_{W \bmod q} \bmod p)$ where $W \leftarrow_\$ [K]$ (recall that $|\mathcal{P}| = K$). We claim that $SD(T_0, T_1) = SD(T_0', T_1)$. Specifically,

$$\begin{aligned}
SD(T_0, T_1) &= \frac{1}{2} \sum_{a \in \mathbb{Z}_q^*} \left| \Pr_{(\ell+\tau) \leftarrow_\$ \mathcal{K}}[\ell + \tau \bmod q = a] - \Pr_{x \leftarrow_\$ \mathbb{Z}_q^*}[x \bmod q = a] \right| \\
&= \frac{1}{2} \sum_{a \in \mathbb{Z}_q^*} \left| \Pr_{\ell \leftarrow_\$ [K]}[\ell \bmod q = (a - \tau) \bmod q] - \Pr_{x \leftarrow_\$ \mathbb{Z}_q^*}[x \bmod q = a] \right| \\
&= \frac{1}{2} \sum_{a \in \mathbb{Z}_q^*} \left| \Pr_{\ell \leftarrow_\$ [K]}[\ell \bmod q = (a - \tau) \bmod q] - \Pr_{x \leftarrow_\$ \mathbb{Z}_q^*}[x \bmod q = (a - \tau) \bmod q] \right| \\
&= SD(T_0', T_1).
\end{aligned}$$

Now, we bound $SD(T_0', T_1)$:

$$\begin{aligned}
SD(T_0', T_1) &= SD((W \bmod q, X^e_{W \bmod q} \bmod p), (U_q, X^e_{U_q} \bmod p)) \\
&\leq SD(W \bmod q, U_q).
\end{aligned}$$

Let $r = K \bmod q$. Then,

$$\begin{aligned}
SD(W \bmod q, U_q) &= \frac{1}{2} \sum_{a \in \mathbb{Z}_q^*} \left| \Pr_{x \leftarrow_\$ [K]}[x \bmod q = a] - \Pr_{x \leftarrow_\$ \mathbb{Z}_q^*}[x = a] \right| \\
&= r\left(\frac{(K - r)/q + 1}{K} - \frac{1}{q - 1}\right).
\end{aligned}$$

Note that $\frac{(K-r)/q+1}{K} \leq (1 + \frac{q-r}{K})\frac{1}{q-1}$ and we have:

$$r\left(\frac{(K - r)/q + 1}{K} - \frac{1}{q - 1}\right) \leq \frac{r}{(q-1)}\frac{q-r}{K} \leq \frac{q}{K}$$

as $0 \leq r \leq q-1$. To conclude,

$$\begin{aligned} SD(T_0, T_2) &\leq SD(T_0', T_1) + SD(T_1, T_2) \\ &\leq \frac{q}{K} + \mathop{\mathbb{E}}_{v \xleftarrow{\$} \mathbb{Z}_q^*} SD((X_v^e \bmod p, U_p^e \bmod p)). \end{aligned}$$

□

Proof of Lemma 4. Let $u \in \mathbb{Z}_q$, we have

$$\begin{aligned} \mathcal{P}_u &= \{x \in \mathcal{P} | x \bmod q = u\} \\ &= \{\ell + \tau | \ell \leq K \wedge \ell = u - \tau \bmod q\} \\ &= \{(u-\tau) \bmod q + qk + \tau | 0 \leq k \leq \frac{K - (u-\tau) \bmod q}{q}\}. \end{aligned}$$

Specifically, we have $\mathcal{P}_0 = \{qk - \tau \bmod q + \tau | 0 \leq k \leq \frac{K+\tau \bmod q}{q}\}$. Recall that $v < q$ ($v \in \mathbb{Z}_q^*$), we have:

$$\mathcal{P}_v = \begin{cases} \{qk - \tau \bmod q + \tau + q + v | 0 \leq k \leq \frac{K-v+\tau \bmod q)}{q} - 1\} & v < \tau \bmod q; \\ \{qk - \tau \bmod q + \tau + v | 0 \leq k \leq \frac{K-v+\tau \bmod q)}{q}\} & \text{otherwise.} \end{cases}$$

Therefore, for any $v \in \mathbb{Z}_q^*$, $|\mathsf{SymDiff}(\mathcal{P}_\mathsf{v}, (\mathcal{P}_\mathsf{0} + \mathsf{v}))| \leq 2$ where $\mathsf{SymDiff}$ denotes symmetric difference. □

Proof of Lemma 5. The proof is done via hybrid arguments. Let $T_0 = (C \bmod p, (C \bmod p + R)^e \bmod p)$, $T_1 = (V_p, (V_p + R)^e \bmod p)$ and $T_2 = (V_p, U_p^e \bmod p)$ and $T_3 = (C \bmod p, U_p^e \bmod p)$. Then,

$$SD(T_0, T_3) \leq SD(T_0, T_1) + SD(T_1, T_2) + SD(T_2, T_3).$$

Via the similar technique (to show $SD(W \bmod q, U_q)$) in Lemma 3, we have:

$$\begin{aligned} SD(T_0, T_1) = SD(T_2, T_3) &= SD(C \bmod p, U_p) \\ &\leq \frac{p}{|C|} = \frac{p}{q-1}. \end{aligned}$$

□

3.3 Average-case Bounds over Random Translations

In this section, we point out a mistake in the proof of Lemma 4 from [LOS13]. We give a counter example to the lemma, explain the error in the proof and prove a corrected version of the lemma which still implies the main conclusions from [LOS13]. First, we restate their lemma:

Incorrect Claim 1 (Lemma 4 [LOS13])**.** *Let $N = pq$ and e be such that $e|p-1$ and $gcd(e, q-1) = 1$. Let $\mathcal{K} \subset \mathbb{Z}_N$ such that $|\mathcal{K}| \geq \frac{4N}{e\alpha^2}$ for some $\alpha \geq \frac{4(p+q-1)}{N}$. Then,*

$$SD((C, (C+X)^e \bmod N), (C, U^e \bmod N) \leq \alpha$$

where $C, U \leftarrow_\$ \mathbb{Z}_N$ and $X \leftarrow_\$ \mathcal{K}$.

3.3.1 Counterexample to Lemma 4 in LOS

The problem with this lemma, as stated, is that raising numbers to the e-th power is a permutation in $\mathbb{Z}_q$, and so exponentiation does not erase any information (statistically) about the value of the input mod q. (It may be that information is lost computationally when p, q are secret, but the claim is about statistical distance.)

Adding a publicly available random offset does not help, since the composition of translation and exponentiation is still a permutation of $\mathbb{Z}_q$. Hence, if $X \bmod q$ is not close to uniform, then $(C, (C+X)^e \bmod q)$ is not close to uniform in $\mathbb{Z}_N \times \mathbb{Z}_q$, and so $(C, (C+X)^e \bmod N)$ is not close to uniform in $\mathbb{Z}_N^2$.

To get a counterexample to the claimed lemma, let $\mathcal{K} = \left\{x \in \mathbb{Z}_N : x \bmod q \in \{0, ..., \frac{q-1}{2}\}\right\}$ (the subset of $\mathbb{Z}_N$ with mod q component less than $q/2$). $\mathcal{K}$ is very large (size about $N/2$) but the pair $C, (X+C)^e \bmod q$ will never be close to uniform when $X \leftarrow_\$ \mathcal{K}$.

The above attack was motivated by the discovery of a mistake in the proof of Lemma 4 from [LOS13]. Specifically, the authors analyze the probability that $(C+X)^e = (C+Y)^e$ by decomposing the event into events of the form $(C+X) = \omega(C+Y)$ where ω is an e-th root of unity. The problem arises because

$$\Pr[(C+X) = \omega(C+Y)] \neq \Pr[C = (\omega-1)^{-1}(\omega Y - X)]$$

since $\omega - 1$ is not invertible in $\mathbb{Z}_N^*$ (it is 0 mod q).

3.3.2 Corrected Translation Lemma

It turns out that distinguishability mod q is the only obstacle to the random translation lemma. We obtain the following corrected version:

Lemma 7. *Let $N = pq$ and e be such that $e|p-1$ and $\gcd(e, q-1) = 1$. Let $\mathcal{K} \subset \mathbb{Z}_N$ be an arithmetic progression. Specifically, let $\mathcal{K} = \sigma[K] + \tau$ with $K > q$. Then,*

$$\begin{aligned} SD\Big((C, (C+X)^e \bmod N), (C, U^e \bmod N)\Big) &\leq \frac{1}{p} + \frac{2}{p-1} + \sqrt{\frac{N}{eK}} + SD(X \bmod q, U \bmod q) \\ &\leq \frac{3}{p-1} + \sqrt{\frac{N}{eK}} + \frac{q}{K}. \end{aligned}$$

where $C, U \leftarrow_\$ \mathbb{Z}_N$ and $X \leftarrow_\$ \mathcal{K}$.

Proof. Applying the same idea in Lemma 3, let $U_p \leftarrow_\$ \mathbb{Z}_p, U_q \leftarrow_\$ \mathbb{Z}_q$, we have:

$$\begin{aligned} &SD\Big((C, (C+X)^e \bmod N), (C, U^e \bmod N)\Big) \\ &= \mathop{\mathbb{E}}_{c \leftarrow_\$ \mathbb{Z}_N} \left(SD((c+X)^e \bmod N, U^e \bmod N)\right) \\ &= \mathop{\mathbb{E}}_{c \leftarrow_\$ \mathbb{Z}_N} \Big(SD(((c+X)^e \bmod p, (c+X) \bmod q), (U_p^e \bmod p, U_q))\Big). \end{aligned}$$

Notice that the mod q components are not raised to the e-th power. This is because exponentiation is a permutation of $\mathbb{Z}_q^*$ as $\gcd(e, q-1) = 1$. For any $c \in \mathbb{Z}_N$, let $T_0(c) = ((c+X)^e \bmod p, (c+X) \bmod q), T_1(c) = ((c+X)^e_{U_q} \bmod p, U_q), T_2 = (U_p^e \bmod p, U_q)$. Then, we can rewrite $\mathbb{E}_{c \leftarrow_\$ \mathbb{Z}_N} \Big(SD(((c+X)^e \bmod p, (c+X) \bmod q), (U_p^e \bmod p, U_q))\Big)$ as $\mathbb{E}_{c \leftarrow_\$ \mathbb{Z}_N} SD\left(T_0(c), T_2\right)$. By the triangle inequality, we have:

$$\mathop{\mathbb{E}}_{c \leftarrow_\$ \mathbb{Z}_N} SD(T_0(c), T_2) \leq \mathop{\mathbb{E}}_{c \leftarrow_\$ \mathbb{Z}_N} (SD(T_0(c), T_1(c)) + SD(T_1(c), T_2)).$$

For each $c \in \mathbb{Z}_N$:

$$\begin{aligned} &SD(T_0(c), T_1) = SD\Big(((c+X)^e \bmod p, (c+X) \bmod q), ((c+X)^e_{U_q} \bmod p, U_q)\Big) \\ &\leq SD((c+X) \bmod q, U_q) \leq SD(X \bmod q, U_q). \end{aligned}$$

The last equality holds because translation by c is a permutation of $\mathbb{Z}_q$. We have:

$$SD(T_1(c), T_2) = SD\Big(((c+X)^e_{U_q} \bmod p, U_q), (U_p^e \bmod p, U_q)\Big)$$

$$= \mathop{\mathbb{E}}_{v \leftarrow\$ \mathbb{Z}_q} SD((X+c)_v^e \bmod p, U_p^e \bmod p).$$

Recall that for any $v \in \mathbb{Z}_q$, $(c+X)_v$ denotes the random variable $c+X$ conditioned on the event that $c+X \bmod q = v$. To sum up,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{c \leftarrow\$ \mathbb{Z}_N} ((c+X)^e \bmod N, U^e \bmod N) \\
&\leq SD(X \bmod q, U_q) + \mathop{\mathbb{E}}_{v \leftarrow\$ \mathbb{Z}_q} \mathop{\mathbb{E}}_{c \leftarrow\$ \mathbb{Z}_N} SD\left((X+c)_v^e \bmod p, U_p^e \bmod p\right).
\end{aligned}$$

Note that only the value of $c \bmod p$ affects $SD((X+c)_v^e \bmod p, U_p^e \bmod p)$. We can replace $c \leftarrow\$ \mathbb{Z}_N$ with $V_p \leftarrow\$ \mathbb{Z}_p^*$. Specifically, let **BAD** be the event that $\gcd(c,p) \neq 1$. As $c \leftarrow\$ \mathbb{Z}_N$, we have $\Pr[\mathbf{BAD}] = \Pr_{c \leftarrow\$ \mathbb{Z}_N}[\gcd(c,p) \neq 1] = \frac{1}{p}$. Therefore, for any $v \in \mathbb{Z}_q$,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{c \leftarrow\$ \mathbb{Z}_N} SD((X+c)_v^e \bmod p, U_p^e \bmod p) \\
&\leq \Pr[\mathbf{BAD}] \cdot 1 + 1 \cdot \mathop{\mathbb{E}}_{c \leftarrow\$ \mathbb{Z}_p^*} SD((X+c)_v^e \bmod p, U_p^e \bmod p) \\
&\leq \frac{1}{p} + \mathop{\mathbb{E}}_{V_p \leftarrow\$ \mathbb{Z}_p^*} SD((X+V_p)_v^e \bmod p, U_p^e \bmod p)
\end{aligned}$$

as $\Pr[\mathbf{BAD}] < 1$ and statistical distance $SD(\cdot,\cdot) < 1$.

By Lemma 6, we have $\mathbb{E}_{V_p \leftarrow\$ \mathbb{Z}_p^*}\left((X+V_p)_v^e \bmod p, U_p^e \bmod p\right) \leq \frac{2}{p-1} + \sqrt{\frac{N}{eK}}$. Thus,

$$\begin{aligned}
&SD_{C \leftarrow\$ \mathbb{Z}_N}((C, (C+X)^e \bmod N), (C, U^e \bmod N)) \\
&\leq \frac{1}{p} + \frac{2}{p-1} + \sqrt{\frac{N}{eK}} + SD(X \bmod q, U_q) \leq \frac{1}{p} + \frac{2}{p-1} + \sqrt{\frac{N}{eK}} + \frac{q}{K}.
\end{aligned}$$

□

3.4 Applications

In this section, we apply the above results to understanding the IND-CPA security of PKCS #1 v1.5 and to showing that the most/least $\log e - 3\log\frac{1}{\epsilon} + o(1)$ significant RSA bits are simultaneously hardcore. To illustrate our results, we show that our bounds imply improvements to the concrete security parameters from [LOS13].

3.4.1 IND-CPA Security of PKCS #1 v1.5

Below, a_{16} denotes the 16-bit binary representation of a two-symbol hexadecimal number $a \in \{00, ..., FF\}$. The encryption scheme in PKCS #1 v1.5 can be defined: let $PKCS(x; r) = x||00_{16}||r$ and r is chosen uniformly random from $\{0,1\}^\rho$. The ciphertext for message x with encryption randomness r then is $(00_{16}||02_{16}||PKCS(x;r))^e \bmod N$ [4].

Theorem 8 (CPA security of PKCS #1 v1.5)**.** *Let λ be the security parameter, $\overline{k} = k(\lambda) \in \mathbb{Z}^+$ and $\epsilon(\lambda), c(\lambda) > 0$. Suppose ΦA holds for c and $\theta \geq 4 + \log \frac{1}{\epsilon}$. Let Π_{PKCS} be the PKCS #1 v1.5 encryption scheme. Assume that $\rho \geq \log N - \log e + 2\log(1/\epsilon) + 4$, then for any IND-CPA adversary $\mathcal{A}$ against Π_{PKCS}, there exists a distinguisher $\mathcal{D}$ for Φ-Hiding with $time(\mathcal{D}) \leq time(\mathcal{A}) + O(\overline{k}^3)$ such that for all $\lambda \in \mathbb{N}$:*

$$\mathbf{Adv}^{ind-cpa}_{\Pi_{PKCS},\mathcal{A}}(\lambda) \leq \mathbf{Adv}^{\Phi A}_{c,\theta,\mathcal{D}}(\lambda) + \epsilon(\lambda).$$

Proof. Define $\mathbf{Game}_0$ be the original IND-CPA security game with the adversary $\mathcal{A}$. Let $\mathbf{Game}_1$ be identical to $\mathbf{Game}_0$ except that (N, e) is generated via lossy RSA key generation (Section 2, Φ-Hiding Assumption), such that $e|p-1$ and $gcd(e, q-1) = 1$. $\mathbf{Game}_2$ is identical to $\mathbf{Game}_1$ except that the challenge ciphertext $c^* = (00_{16}||02_{16}||PKCS(x^*, r^*))^e \bmod N$ is replaced with $U^e \bmod N$ where $U \leftarrow_\$ \mathbb{Z}_N^*$.

An adversary who performs differently in $\mathbf{Game}_0$ and $\mathbf{Game}_1$ can be used to attack the ΦA assumption; the increase in running time is the time it takes to generate a challenge ciphertext, which is at most $O(\overline{k}^3)$. The difference between $\mathbf{Game}_1$ and $\mathbf{Game}_2$ is $SD((00_{16}||02_{16}||PKCS(x^*, r^*))^e \bmod N, (\mathbb{Z}_N^*)^e \bmod N)$ (information theoretically) where x^* is the challenge plaintext and r^* is the encryption randomness. Specifically, given the challenge plaintext x^* that may depend on $\mathsf{pk} = (N, e)$, $00_{16}||02_{16}||PKCS(x^*, \cdot) = \{r + x^* 2^{\rho+8} + 2^{\rho+8+|x|} | r \in \{0,1\}^\rho\}$ is an arithmetic progression with length 2^ρ. By Theorem 2,

$$\begin{aligned} SD(0002_{16}||PKCS(x^*, r^*)^e \bmod N, (\mathbb{Z}_N^*)^e \bmod N) &\leq \frac{1}{p-1} + \frac{2p}{q-1} + \frac{3q}{2^{\rho+1}} + \sqrt{\frac{N}{e2^\rho}} \\ &\leq 2\left(\frac{2p}{q-1} + \sqrt{\frac{N}{e2^\rho}}\right) < \epsilon \end{aligned}$$

where we have $\frac{2p}{q-1} < \frac{\epsilon}{4}$ when $\theta \geq 4 + \log \frac{1}{\epsilon}$, and $\sqrt{\frac{N}{e2^\rho}} < \frac{\epsilon}{4}$ when $\rho \geq \log N - \log e - 2\log \epsilon + 4$.

[4] RFC2313, http://tools.ietf.org/html/rfc2313

Note that the advantage of $\mathcal{A}$ in $\mathbf{Game}_2$ is 0. $\square$

Achievable Parameters. To get a sense of the parameters for which our analysis applies, recall the best known attack on Φ-Hiding (using Coppersmith's algorithm) has a tradeoff of time to success probability of at least 2^λ when $p < q$ and $\log(e) = \frac{\log(p)}{2} - \lambda$. We therefore select this value of e (that is, $e = \sqrt{p}/2^\lambda$) for security parameter λ.

For a message of length m, PKCS #1 v1.5 uses a random string of length $\rho = \log N - m - 48$ (since six bytes of the padded string are fixed constants). To apply Theorem 8, we need two conditions. First, we need $\rho \geq \log N - \log e + 2\log(1/\epsilon) + 4$; for this, it suffices have a message of length $m \leq \log(e) - 2\log(1/\epsilon) - 52$. Second, we need $\theta = \log q - \log p \geq 4 + \log(1/\epsilon)$. Setting p to have length $\log(p) = \frac{\log(N)}{2} - \frac{\log(1/\epsilon)+4}{2}$ satisfies this condition.

Using the value of e based on Coppersmith's attack, and setting $\epsilon = 2^{-\lambda}$ in Theorem 8, we get CPA security for messages of length up to

$$m = \tfrac{1}{4}\log N - \tfrac{13}{4}\lambda - 53\,. \tag{3.2}$$

with security parameter λ.

In contrast, the analysis of [LOS13] proves security for messages of length only $m = \frac{\log N}{32} - \Theta(\lambda)$. Even under the most optimistic number-theoretic conjecture (the MVW conjecture on Gauss sums), their analysis applies to messages of length only $m = \frac{\log N}{4} - \Theta(\lambda)$. Their proof methodology cannot go beyond that bound. Our results therefore present a significant improvement over the previous work.

Concrete Parameters: Take the modulus length $\overline{k} = \log N = 8192$ as an example. We will aim for $\lambda = 80$-bit security. We get CPA security for messages of length up to

$$m = \frac{\log N}{4} - \frac{13}{4}\lambda - 53 = 1735 \text{ (bits)}.$$

This is improves over the 128 bit messages supported by the analysis of [LOS13] by a factor of 13. (That said, we do not claim to offer guidance for setting parameters in practice, since our results require an exponent e much larger than the ones generally employed.)

3.4.2 (Most/Least Significant) Simultaneously Hardcore Bits for RSA

Let λ be the security parameter and let $\overline{k} = \log N$ be the modulus length. For $1 \leq i < j \leq \overline{k}$, we want to show that the two distributions $(N, e, x^e \bmod N, x[i,j])$ and $(N, e, x^e \bmod N, r)$

are computationally indistinguishable, where $x \leftarrow_\$ \mathbb{Z}_N^*$, $r \leftarrow_\$ \{0,1\}^{j-i-1}$, and $x[i:j]$ denotes bits i through j of the binary representation of x.

In this section, we apply Theorem 2 to show the most and least $\log e - O(\log \frac{1}{\epsilon})$ significant bits of RSA functions are simultaneously hardcore (Theorem 9). We should note that we can apply the corrected random translations lemma (our Lemma 7) to this problem, which yields an essentially identical result. For brevity, we omit its proof.

Theorem 9. *Let λ be the security parameter, $\overline{k} = k(\lambda) \in \mathbb{Z}^+$ and $\epsilon(\lambda), c(\lambda) > 0$. Suppose ΦA holds for c and $\theta > 4 + \log \frac{1}{\epsilon}$. Then, the most (or least) $\log e - 2\log \frac{1}{\epsilon} - 2$ significant bits of RSA are simultaneously hardcore. Specifically, for any distinguisher $\mathcal{D}$, there exists a distinguisher $\overline{\mathcal{D}}$ running in time time$(\mathcal{D}) + O(\overline{k}^3)$ such that*

$$\left|\Pr[\mathcal{D}(N, e, x^e \bmod N, x[i:j]) = 1] - \Pr[\mathcal{D}(N, e, x^e \bmod N, r[i:j]) = 1]\right| \leq \mathbf{Adv}^{\Phi A}_{c,\theta,\overline{\mathcal{D}}}(\lambda) + 2\epsilon.$$

where $r \leftarrow_\$ \mathbb{Z}_N$, $|j - i| \leq \log e - 2\log \frac{1}{\epsilon} - 2$ and either $i = 1$ or $j = \overline{k}$. Furthermore, the distribution of $r[i;j]$ is $2^{\overline{k}-j}$-far from uniform on $\{0,1\}^{j-i+1}$.

It's important to note that the theorem is stated in terms of the distinguishability between bits i through j of the RSA input, and bits i through j of a random element r of $\mathbb{Z}_N$. The string $r[i:j]$ is not exactly uniform – indeed, when $j = \overline{k}$, it is easily distinguishable from uniform unless N happens to be very close to a power of 2.

Depending on the application, it may be important to have $x[i:j]$ indistinguishable from a truly uniform string. In that case, one may either set $i = 1$ (use the least significant bits) or, in the case $j = \overline{k}$, ignore the top $\log(1/\epsilon)$ bits of $r[i;\overline{k}]$ (effectively reducing the number of hardcore bits to about $\log(e) - 3\log(1/\epsilon)$ bits).

Proof of Theorem 9. We define two games. Let $U \leftarrow_\$ \mathbb{Z}_N^*$. $\mathbf{Game}_0$ is to distinguish $(N, e, x^e \bmod N, x[i,j])$ and $(N, e, U^e \bmod N, x[i,j])$; $\mathbf{Game}_1$ is to distinguish $(N, e, U^e \bmod N, x[i,j])$ and $(N, e, U^e \bmod N, r)$. Since x is chosen uniform randomly from $\mathbb{Z}_N^*$, the advantage in $\mathbf{Game}_1$ is at most $2^{j-\overline{k}}$ (since $\overline{k}$ is the bit length). Let $\mathcal{D}$ be any distinguisher, and let $\overline{\mathcal{D}}$ be the distinguisher for the Φ-Hiding game that prepares inputs to $\mathcal{D}$ using a challenge public key and uses $\mathcal{D}$'s output as its own. We have

$$\begin{aligned}\mathbf{Adv}^{\mathbf{Game}_0}_{\mathcal{D}}(1^\lambda) &= |\Pr[\mathcal{D}(N, e, x^e \bmod N, x[i,j]) = 1] - \Pr[\mathcal{D}(N, e, (\mathbb{Z}_N^*)^e \bmod N, x[i,j]) = 1]| \\ &\leq \mathbf{Adv}^{\Phi A}_{c,\theta,\overline{\mathcal{D}}}(\lambda) + SD(\mathcal{P}^e \bmod N, U^e \bmod N)\end{aligned}$$

where $\mathcal{P}$ is the set of integers with bits i through j set to $x[i:j]$.

The structure of $\mathcal{P}$ depends on the integers i and j. In general, when $j < \overline{k}$ and $i > 1$, $\mathcal{P}$ may not be well-approximated by an arithmetic progression. However, if $j = \overline{k}$, then $\mathcal{P}$ is the arithmetic progression $\mathcal{P} = \{x[i,j] \cdot 2^{i-1} + a \mid a = 0, \ldots, 2^{i-1} - 1\}$. If $i = 1$, then the set $\mathcal{P}$ is more complicated, but it is closely approximated by an AP. Specifically, let $\mathcal{P}' = \{x[i,j] + b \cdot 2^j \mid b = 0, ..., N_j\}$, where $N_j \stackrel{def}{=} N \text{ div } 2^j$ is the integer obtained by consider only bits $j+1$ through $\overline{k}$ of the binary representation of the modulus N. Then the uniform distribution on $\mathcal{P}$ is at most $2^{\overline{k}-j}$-far from the uniform distribution on $\mathcal{P}'$.

As Theorem 2 applies to arithmetic progressions, we can apply it in the cases $i = 1$ and $j = \overline{k}$. By Theorem 2,

$$\mathbf{Adv}_{\mathcal{D}}^{\mathbf{Game}_0}(1^\lambda) \leq 2\left(\tfrac{2p}{q-1} + \sqrt{\tfrac{N}{e2^{\overline{k}-|j-i|}}}\right) < 2\epsilon\,.$$

The last inequality uses the hypotheses that $\theta = \log q - \log p \geq 4 + \log \frac{1}{\epsilon}$ and $|j - i| < \log e - 2\log \frac{1}{\epsilon} - 2$. □

Concrete Parameters: Let λ denote the security parameter. As in the calculations for PKCS in the previous section, we require $\log(e) \leq \frac{\log p}{2} - \lambda$ (for Coppersmith's attack to be ineffective) and $\epsilon = 2^{-\lambda}$. To apply Theorem 9, we require that $\theta \geq 4 + \log \frac{1}{\epsilon} = 4 + \lambda$, and therefore $\log e \leq \frac{\overline{k}-\theta}{4} - \lambda \leq \frac{\overline{k}-5\lambda}{4} - 1$. Theorem 9 then proves security for a run of bits with length $\log e - 2\lambda - 2 = \frac{1}{4}\overline{k} - \frac{13}{4}\lambda - 3$. For example, for a modulus of length $\overline{k} = 2048$ bits and security parameter $\lambda = 80$, we get that the 249 least significant bits are simultaneously hardcore. Alternatively, our analysis shows that the 169 bits in positions $\overline{k} - 249$ through $\overline{k} - 169$ are simultaneously hardcore (see the discussion immediately after Theorem 9).

Chapter 4
Leakage-Resilient Key Evolution Schemes

Side-channel attacks have led the cryptographic community to develop tools for reasoning about the security of computations on partially compromised devices. Recently, Dziembowski, Kazana and Wichs [DKW] (henceforth "DKW") proposed a model for leakage-resilient updating of a stored secret key that protects against an internal attacker who can continuously leak a bounded amount of information to the outside world and even tamper with the device's internal computations. In this thesis, we develop tools for reasoning about computations in the DKW model. These tools allow us to provide a highly efficient key evolution scheme as well as stronger connections to complexity theory, namely to the theory of pebbling games and expander graphs. Our new scheme tolerates a linear amount of leakage and runs in time quasilinear in the key length, improving significantly on the quadratic-time scheme of [DKW].

Key Evolution Schemes and the DKW Model.

Consider a cryptographic key y that is stored on a device about which an attacker may be able to learn information gradually over time. To stop the entire key from being leaked, it is periodically updated via a deterministic key evolution scheme $\mathcal{KE}$ to obtain a sequence of keys $y_0, y_1, y_2, \ldots$ where $y_{i+1} = \mathcal{KE}(y_i)$. Determinism allows keys to be updated independently on separate devices. The hope is that the update operator $\mathcal{KE}$ effectively erases the effect of previously learned information. If we limit only the *amount* of leakage between updates, no deterministic scheme is secure since the attacker may directly ask for bits of a future key (for example, if the key is $\overline{k}$ bits long, the attacker could ask for single bit of $y_{\overline{k}} = \mathcal{KE}^{(\overline{k})}(y_0)$ at each of the first $\overline{k}$ steps, effectively leaking the entire key $y_{\overline{k}}$). To get around this, researchers have studied restricted classes of functions that can be leaked at each step. In other words,

we think of a highly constrained "small" adversary *inside* the device who leaks information to an outside "big" adversary. For example, in addition to restricting the length of the leaked information, one might restrict the leakage to operate independently on separate parts of the memory (e.g., the "wire-probe" [ISW03] and "split-state" models [DDV10]), or to operate only on parts of memory that are explicitly touched by a computation [MR04], or to be from a computationally simple circuit class such as AC_0 [FRR+10].

The DKW model [DKW] assumes instead that the "small" adversary (the corrupted device) is limited in space. They do *not* assume that the computations inside the device are performed "honestly", but they do assume that they "look" right to the outside world, meaning that the correct round key is available when an update occurs. More specifically, they consider a two-part attacker $(\mathcal{A}_\mathrm{s}, \mathcal{A}_\mathrm{b})$, where $\mathcal{A}_\mathrm{s}$ has access to the initial key y_0. $\mathcal{A}_\mathrm{b}$ is unlimited in communication or storage, but $\mathcal{A}_\mathrm{s}$ is significantly restricted:

- When the i-th update occurs, $\mathcal{A}_\mathrm{s}$ holds the correct key $y_i = \mathcal{KE}^{(i)}(y_0)$ in memory.
- $\mathcal{A}_\mathrm{s}$ can send up to c bits to $\mathcal{A}_\mathrm{b}$ during each "round" (that is, between any two updates). To avoid a trivial attack, c must be less than $|y|$.
- $\mathcal{A}_\mathrm{s}$ may use any algorithm that works with total *space* at most s bits. We denote by $s_\mathrm{extra} = s - s_\mathrm{honest}$ the difference between s and the space s_honest used by a honest implementation of $\mathcal{KE}$. For the model to make sense, s_extra must be nonnegative.
- $\mathcal{A}_\mathrm{s}$ and $\mathcal{A}_\mathrm{b}$ are limited to reasonable computation. We use the random oracle model (as do [DKW]); the number of oracle calls made by the adversary is bounded by the parameter q.

The key evolution scheme is secure roughly if the leakage on the first i updates lets $\mathcal{A}_\mathrm{b}$ learn nothing about the later keys $y_{i+1}, y_{i+2}, \ldots$. Formalizing this is delicate (see "Our Contributions", below). Intuitively though, any key evolution scheme must somehow prevent $\mathcal{A}_\mathrm{s}$ from making a copy of the key (since then it could compute future keys and leak information about them while still keeping the current key around), so s must be less than $|y| + s_\mathrm{honest} \approx 2|y|$ for a secure key evolution scheme to be possible. An ideal scheme would allow the honest evaluation algorithm to use time and space as close to $|y|$ as possible, while tolerating $c \approx |y|$ leakage and $s_\mathrm{extra} \approx |y|$ extra space.

At first glance, it seems that a simple solution to this problem is to use a sufficiently complicated hash function to update the key at each step. If we use the random oracle model and assume that the oracle maps $\{0,1\}^{|y|}$ to $\{0,1\}^{|y|}$ bits, then the scheme can indeed

be proven secure when $s < 2|y| - \log q$ and $c < |y| - \log q$. But such a naïve version of the random oracle abstraction hides too much in this setting: a hash function may be computable from a tiny summary of its input (hash functions based on the Merkle-Damgård paradigm, for example, are insecure in the DKW model; see [DKW] for details).

Instead, we seek to design schemes that provably prevent an untrusted device from computing future keys while keeping the current key available. We use the random oracle to model a "small" hash function H that maps $\{0,1\}^{dw}$ to $\{0,1\}^w$ for a small constant d and fixed word length w. The update function $\mathcal{KE}$ operates on longer keys and tolerates leakage far above the length of the hash function's inputs and outputs.

DKW [DKW] proposed an elegant key evolution scheme that is secure in the random oracle model as long as $4c + s_{\text{extra}}$ is significantly less than $|y|/2$. The result is remarkable in that the key length $|y| = nw$ can be very long even when the hash function output length w is short, yet the leakage and adversarial work space can both be linear in $|y|$. Their update algorithm requires only $s_{\text{honest}} = |y| + w$ bits, but it requires at least $\frac{3}{2}n^2$ hash function calls, which is quadratic in the key length.

4.0.3 Our Contributions

1. We give a new key evolution scheme, also in the random oracle model, for which the key update step can be done in quasilinear time $O(n \log n \log q)$ for keys of length $|y| = nw$ (assuming hash evaluations take constant time). This improves dramatically over the n^2 running time in DKW. As with DKW, the extra space s_{extra} and leakage c in our scheme can both be linear in $|y|$. Specifically, the update algorithm can be run in space $s_{\text{honest}} = (1 + \delta)|y|$ for an arbitrarily small constant $\delta > 0$ and it is secure roughly as long as $4c + s_{\text{extra}} < |y|/8$ (slightly worse than the $|y|/2$ tolerance of DKW).

2. We strengthen the connections between the DKW model and pebbling theory, showing that random-oracle-based key evolution schemes are secure as long as the "graph" of the update function's calls to the oracle has appropriate combinatorial properties. This builds on a connection between pebbling and the random oracle model first established by Dwork, Naor and Wee [DNW] and built on by DKW [DKW11, DKW]. Our scheme's efficiency relies on the existence (which we show) of families of δ-*local* bipartite expander graphs of constant degree.

3. We provide a precise formalization in the standard model for the problem described by DKW. Their definition of security was highly specific to a particular class of schemes

in the random oracle model. We provide a stand-alone security definition and prove that the simplified security definition in [DKW] implies our definition.

4.0.4 Background and Further Related Work

Pebbling and Random Oracles. Pebbling games were first investigated as a way to prove time/space tradeoffs in circuit complexity [Val75]. Given a directed acyclic graph G, the *inputs* (or sources) are vertices of in-degree 0, and *outputs* (or sinks) are vertices of out-degree 0. A pebbling strategy begins with special markers ("pebbles") on the input vertices and seeks to cover all the outputs with pebbles, under the restriction that a pebble can only be placed on a vertex when all of its immediate predecessors have been covered.

Pebbling games were first used in cryptography by Dwork, Naor and Wee [DNW] in the context of proofs of work. They observed a strong connection between pebbling games and the random oracle model: Given a graph G and a hash function H, they design a boolean function whose computational complexity subject to a space constraint (in the RO model) is given by number of moves needed to pebble G subject to a constraint on the number of available pebbles. More specifically, each vertex of the graph is assigned a label of length w (the output size of the hash function). Input vertices are labeled using the function's inputs, and other vertices are labeled by the hash of the labels of their predecessors. The output of the function is the concatenation of the labels of output vertices. This connection between pebbling and the RO model was developed further by DKW to design "one-time" pseudorandom functions [DKW11] and leakage-resilient key evaluation schemes [DKW]. Assuming the availability of a random oracle, DKW showed that the computations in their model could be made to correspond to a variant of pebbling (see Section 4.4). We develop new techniques for analyzing such pebbling games in this work.

Superconcentrators. A class of graphs called *superconcentrators* [Pip77] emerged as important for pebbling lower bounds. "Stacks" of superconcentrators (that is, graphs constructed by placing many superconcentrators in sequence) require exponentially many moves to pebble using a sublinear number of pebbles [LT82]. We use such a stack in our construction. While we cannot use the results from previous work directly (since we use a variant of standard pebbling games), we modify a key lemma from Lengauer and Tarjan [LT82] (the "basic lower bound") for our analysis.

A line of research focused on construction superconcentrators with as sparse as possible, eventually obtaining constructions $O(n)$ edges for n inputs and outputs [Pip77, AC]. Our

constructions use sparse superconcentrators, but we require graphs with several additional properties and so again we cannot use the results from the literature directly.

4.0.5 Overview of Our Construction and Techniques

The scheme of DKW defined a key update function $\mathcal{KE}$ via a very simple graph (using random oracle to get a boolean function as in Dwork et al. [DNW]). The graph has n inputs and n outputs (and thus the key length is nw). In between, there are $\frac{3}{2}n$ layers, each with n vertices. Each vertex j on a given layer i has edges *to* vertices j and $j+1 (\mathrm{mod} n)$ on layer $i+1$, and edges *from* vertices $j-1(\mathrm{mod}\, n)$ and j on layer $i-1$. It is possible to pebble this graph using only $n+2$ pebbles, and hence it is possible evaluate their update function using space $(n+2)w = |y| + 2w$. However, the graph has $\frac{3}{2}n^2$ vertices and thus requires $O(n^2)$ time to evaluate.

A key observation is that one can think of the DKW graph as being "generated" by a much simpler graph, the cycle C_n. Namely, if we identify vertices on two adjacent layers i and $i+1$, we get a graph on n vertices where each vertex j is connected to vertices $j-1$ and $j+1$ (as well as to itself). In a nutshell, our construction generates a key evaluation scheme from a more complex graph. This allows us to prove security using far fewer layers ($O(\log n \log q)$ layers, instead of n).

The graph of our key update scheme consists of a stack of $O(\log n \log q)$ copies of a "base" bipartite graph of constant degree. We need two apparently contradictory properties from the graphs: *locality* and *vertex expansion*.

Locality is the property that, if we order left and right vertices from 1 to n, then edges only exist between vertices that are "nearby" in the ordering (at most δn indices apart, for a small constant δ). If the bipartite graphs are local, then our graph can be pebbled using just $(1+\delta)n$ pebbles, and so the update function can be evaluated in space $s_{\mathrm{honest}} = (1+\delta)nw$.

Vertex expansion is the property that small sets of vertices on the left are connected to "many" vertices on the right. We show that key evolution schemes based on expander graphs are secure even against attackers with time exponential in the height of the stack used to define the update function. The proof has two components: first, we show that stacks of $\log n$ expanders are superconcentrators, and hence that learning anything about future keys requires $\mathcal{A}_{\mathrm{s}}$ to "sacrifice" a large amount of memory that cannot be used to compute the current round key. The second component is to show that $\mathcal{A}_{\mathrm{s}}$ also needs a large amount of memory to be able to eventually compute the current round key. This argument uses

expansion more directly (that is, it does not go through superconcentrators), and is much more technically involved. Taken together, the two components show that any successful attack requires $s_{\text{honest}} + \Omega(|y|)$ memory, even when $\Omega(|y|)$ leakage is possible.

Finally, we show that for every $\delta > 0$, one can find constant-degree, *local* vertex expanders, completing the construction.

4.1 Graph-based Key Evolution Schemes with Random Oracles

We work with a specific class of key evolution schemes, following the approach of Dwork et al. [DNW]. Let $G = (V, E)$ be a graph. The input set $I(G)$ of G is the set of vertices with in-degree of 0. Similarly, the output set $O(G)$ is the set of all vertices with out-degree 0. We denote by $V(G) = V$ the set of vertices of G and by $E(G) = E$ is the set of edges.

Definition 5 (Key Evolution Scheme as a Graph)**.** Let $H : \{0,1\}^{dw} \to \{0,1\}^w$ be a random oracle. Let $G_{\mathcal{KE}}$ be a directed graph with n inputs $I_1, \ldots, I_n$ and n outputs $O_1, \ldots, O_n$ such that each vertex in $G_{\mathcal{KE}}$ has indegree either 0 or d. We define a key evolution scheme $\mathcal{KE} : \{0,1\}^{nw} \to \{0,1\}^{nw}$ as follows: Let y_i denote the i-th round key and write $y_i = (r_{i1}, r_{i2}, \ldots, r_{in})$ where each r_{ij} has w bits. Assign a w-bit value (called a *label*) $r(v)$ to each vertex v in $G_{\mathcal{KE}}$. For inputs $I_1, \ldots, I_n$, set $r(I_j) = r_{ij}$ $(j = 1, \ldots, n)$. For $v \notin I(G_{\mathcal{KE}})$, let $t_1, \ldots, t_d$ be d vertices connected to v and set $r(v) = H(r(t_1), \ldots, r(t_d))$. Then the output y_{i+1} is defined as $(r(O_1), r(O_2), \ldots, r(O_n))$ (the concatenation of labels of the outputs).

The key evolution scheme in [DKW] can be described as a grid graph that has $\frac{3}{2}n$ layers and n vertices in each layer. Specifically, the j-th vertex at the i-th layer $v_{i,j}$ is connected to $v_{i+1,j}$ and $v_{i+1,(j+1) \bmod n}$.

4.2 Security Models

We consider two security models. The first, more general one is given in the standard model, and does not assume anything about the structure of the key evolution scheme or its analysis. The second model, due to DKW [DKW], is specific to graph-based key evolution schemes in the random oracle model. We show that the specific definition (Definition 8) implies the general one (Theorem 10).

We denote by *round* the period between two successive updates.

4.2.1 Security Definition in the Standard Model

Definition 6. Fix a round $u \geq 0$. Consider a security game $\mathbf{Game}_6$ between a challenger $\mathcal{C}$ and an adversary $\mathcal{A} = (\mathcal{A}_s, \mathcal{A}_b)$. $\mathcal{C}$ picks y_0 and T_0 uniformly at random. $\mathcal{C}$ also computes $T_1 = y_{u+1}$ via the key evolution scheme $\mathcal{K}$ and flips a random coin b uniformly. Initially, $\mathcal{A}_s$ is given y_0 and $\mathcal{A}_b$ is given T_b. At the end of each u'-th round where $u' < u$, $\mathcal{A}_s$ outputs $\tilde{y}_{u'}$ to $\mathcal{C}$. At the end of the u-th round, $\mathcal{A}_b$ outputs its guess bit b' to $\mathcal{C}$. Finally, $\mathcal{A}_s$ outputs $\tilde{y}_u$ to $\mathcal{C}$. It is important that $\tilde{y_u}$ is output after b' is received. Let E be the event that $\tilde{y}_i = y_i$ for all $i = 1, \ldots, u$. The advantage of $\mathcal{A} = (\mathcal{A}_s, \mathcal{A}_b)$ at round u is defined as:

$$\mathbf{Adv}_{\mathcal{A}}^{\mathbf{Game}_6} = |\Pr[(b' = b) \wedge E] - \frac{1}{2}\Pr[E]|.$$

We say that a key evolution scheme is ϵ-secure (against $\mathbf{Game}_6$) if for every adversary $A = (\mathcal{A}_s, \mathcal{A}_b)$, the advantage $Adv_{\mathcal{A}}^{\mathbf{Game}_6} \leq \epsilon$ for all rounds u.

Note that if $\Pr[E] = 1$, $\mathbf{Adv}_{\mathcal{A}}^{\mathbf{Game}_6} = |\Pr[b' = b] - \frac{1}{2}|$. We also note that, if $\mathcal{A}_s$ and $\mathcal{A}_b$ can communicate arbitrarily, $\mathcal{A}_s$ can send y_0 to $\mathcal{A}_b$ and $\mathcal{A}_b$ evaluates y_{u+1} itself. In this case, no key evolution scheme is secure for $\epsilon < \frac{1}{2}$. On the other hand, [DKW] shows that if c and s are with some restrictions, there do exist secure key evolution schemes in the random oracle model.

4.2.2 Security Definition with Graph-based Key Evolution Schemes in the Random Oracle Model

Definition 7 ((c, s, q) adversary)**.** An adversary $\mathcal{A} = (\mathcal{A}_s, \mathcal{A}_b)$ is a (c, s, q) adversary in the random oracle model if:

1. $\mathcal{A}_s$ can store at most s bits at any time,
2. $\mathcal{A}_s$ can send at most c bits of communication to $\mathcal{A}_b$ in each round,
3. $\mathcal{A}_b$ has unlimited storage space and can send unlimited communication to $\mathcal{A}_s$, and
4. $\mathcal{A}_s$ and $\mathcal{A}_b$ call the random oracle at most q times overall.

Given a vertex v, we say that the adversary *evaluates* $r(v)$ *via the random oracle* if it calls the oracle H on the correct labels $r(t_1), \ldots, r(t_d)$ of v's predecessors.

Definition 8 ((c, s, q, ϵ)-Security [DKW])**.** Consider a graph-based key evolution with graph G and oracle $H : \{0,1\}^{dw} \to \{0,1\}^w$. Fix a round $u > 0$. Consider the following security game $\mathbf{Game}_8$ between a challenger $\mathcal{C}$ and an adversary $\mathcal{A} = (\mathcal{A}_s, \mathcal{A}_b)$:

- $\mathcal{C}$ picks y_0 uniformly at random in $\{0,1\}^{nw}$ and gives y_0 to $\mathcal{A}_s$.
- For $i = 1, ..., u$, $\mathcal{A}_s$ outputs a string $\tilde{y}_i \in \{0,1\}^w$. Round i ends when $\mathcal{A}_s$ begins to output $\tilde{y}_i$.

The event E occurs if $\tilde{y}_i = y_i$ for all $i = 1, \ldots, u$. The event E_1 occurs if, *before the end of round* u, either $\mathcal{A}_s$ or $\mathcal{A}_b$ evaluates the label of any vertex $r_{u+1,j}$ (for $j \in \{1, \ldots, n\}$) defining the $(u+1)$-th key y_{u+1}.

The advantage of $\mathcal{A} = (\mathcal{A}_s, \mathcal{A}_b)$ at round u is defined as:

$$\mathbf{Adv}_{\mathcal{A}}^{\mathbf{Game}_8} = \begin{cases} 0 & \text{if } \Pr[E] = 0; \\ \Pr[E_1|E] & \text{otherwise.} \end{cases}$$

We say that a key evolution scheme is ϵ-secure (against (c, s, q)-adversaries in $\mathbf{Game}_8$) if for every (c, s, q) adversary $\mathcal{A} = (\mathcal{A}_s, \mathcal{A}_b)$, the advantage $\mathbf{Adv}_{\mathcal{A}}^{\mathbf{Game}_8}$ is at most ϵ for all rounds u.

Theorem 10. *Let $\mathcal{A} = (\mathcal{A}_s, \mathcal{A}_b)$ be an adversary in the random oracle model. Then,*

$$\mathbf{Adv}_{\mathcal{A}}^{\mathit{Game}_6} \leq \frac{3}{2}\mathbf{Adv}_{\mathcal{A}}^{\mathit{Game}_8}.$$

Proof. When $\Pr[E] = 0$, $\mathbf{Adv}_{\mathcal{A}}^{\mathbf{Game}_6} = |\Pr[(b' = b) \wedge E] - \frac{1}{2}\Pr[E]| = 0$. The claim is true. Otherwise,

$$\begin{aligned} \mathbf{Adv}_{\mathcal{A}}^{\mathbf{Game}_6} &= |\Pr[(b' = b) \wedge E] - \frac{1}{2}\Pr[E]| \\ &\leq |\Pr[b' = b|E] - \frac{1}{2}| \\ &\leq \Pr[E_1|E] + |\Pr[b' = b|E \wedge \overline{E_1}]\Pr[\overline{E_1}|E] - \frac{1}{2}|. \end{aligned}$$

Let $\delta = \Pr[b' = b|E \wedge \overline{E_1}] - \frac{1}{2}$. Then, we have:

$$|\Pr[b' = b|E \wedge \overline{E_1}]\Pr[\overline{E_1}|E] - \frac{1}{2}| = |(\delta + \frac{1}{2})\Pr[\overline{E_1}|E] - \frac{1}{2}|$$

$$
\begin{aligned}
&= |\Pr[\overline{E_1}|E]\delta - \frac{1}{2}(1 - \Pr[\overline{E_1}|E])| \\
&\leq |\Pr[\overline{E_1}|E]\delta - \frac{1}{2}\Pr[E_1|E]| \\
&\leq |\delta| + \frac{1}{2}\Pr[E_1|E].
\end{aligned}
$$

Therefore, $\mathbf{Adv}_{\mathcal{A}}^{\mathbf{Game}_6} \leq \frac{3}{2}\Pr[E_1|E] + |\delta| = \frac{3}{2}\mathbf{Adv}_{\mathcal{A}}^{\mathbf{Game}_8} + |\delta|$. On the other hand,

$$
\begin{aligned}
|\delta| &= |\Pr[b' = b|E \wedge \overline{E_1}] - \frac{1}{2}| \\
&= 0.
\end{aligned}
$$

Specifically, if the event E_1 does not occur, neither $\mathcal{A}_s$ nor $\mathcal{A}_b$ evaluates $r_{(u+1),j}$ for any $j \in \{1, \ldots, n\}$ via the random oracle calls. Therefore, the value $y_{u+1} = (r_{(u+1),1}, \ldots, r_{(u+1),n})$ is uniformly random to both $\mathcal{A}_s$ and $\mathcal{A}_b$ information theoretically. Then, the probability that $\mathcal{A}_b$ guesses correctly on $b' = b$ is exactly $\frac{1}{2}$. □

4.3 Quasilinear-time Key Evolution Schemes

The graph of our scheme consists of a stack of copies of a bipartite graph B, which itself is built from a "base" graph G with special properties: vertex expansion and locality. In the following, we will explain these two properties and our main construction. Then, we will show that our construction can be updated in quasilinear time (Theorem 15) and is secure (Theorem 16).

Definition 9. An undirected graph $G = (V, E)$ is a (k, A) vertex expander if for every $S \subset V$ and $|S| \leq k$, $|\Gamma(S)| \geq A|S|$, where $\Gamma(S) = \{v | u \in S \wedge (u, v) \in E\}$.

Definition 10. An undirected graph $G = (V, E)$ on n vertices $V = \{1, 2, \ldots, n\}$ is δ-*local* if $|i - j| \leq \delta n$ for every $(i, j) \in E$.

The DKW scheme is based on the cycle graph, which is δ-local with $\delta = 1/n$. Unfortunately, the cycle is a poor expander. We show that by letting δ be a small constant, we can in fact get asymptotically good expanders.

4.3.1 Existence of δ-local Vertex Expanders

Theorem 11 (Existence)**.** *For every $\delta > 0$, there exists a constant d such that, for all sufficiently large n, there exists a d-regular δ-local graph G that is a $(\frac{4n}{5}, A = 1 + \frac{\delta}{2})$ vertex expander.*

To prove the theorem, we show that the probability that a *random* d-regular δ-local graph is a $(\frac{4n}{5}, 1 + \frac{\delta}{2})$ vertex expander is close to 1, when d is a sufficiently large constant. In order to prove it, we need some lemmas (which will be proved later).

Definition 11. For $\forall k \in \mathbb{N}^+, \delta > 0$, let $S \subset [1, n]$ with size $|S| = k$; $T \subset [1, n]$ of size Ak. Define $p_i(S, T) = \frac{|T \cap \Gamma^*(v_i)|}{|\Gamma^*(v_i)|} \leq 1$ (p_i for short) where v_i is the i-th element in S and $\Gamma^*(v_i) = [v_i - \delta n, v_i + \delta n]$.

Lemma 12. *For $\forall 1 > \beta \geq 0$, If $\frac{1}{k}\sum_{i=1}^{k} p_i \leq 1 - \beta$, then there are at least $(1 - \sqrt{1-\beta})k$ elements in S whose $p_i \leq \sqrt{1-\beta}$.*

Proof. Suppose that $> \sqrt{1-\beta}k$ elements in S whose $p_i > \sqrt{1-\beta}$. Then, we have $\frac{1}{k}\sum_{i=1}^{k} p_i > \sqrt{1-\beta}^2 = 1 - \beta$, which shows a contradiction. □

Lemma 13. *Let $A = 1 + \frac{\delta}{2}$. For any sets S, T with $|S| = k$ and $|T| = Ak$, where $\frac{4n}{5} \geq k \geq \frac{2\delta n+1}{A+1}$ and assuming that n is large enough, then*

$$\max_{S,T} \frac{1}{k}\sum_{i=1}^{k} p_i \leq 1 - \frac{5\delta}{16}$$

.

Lemma 14. *For $\frac{2\delta n-1}{A+1} \geq k \geq 1$, assuming n is large enough, we have $\frac{1}{k}\sum p_i \leq \frac{Ak}{2\delta n} < 1$.*

Proof of Theorem 11. We prove this theorem by considering a random d-regular graph. We show that such a graph is *good* with probability greater than 0. Therefore, there will exist such a *good* expander graph.

Specifically, given any $1 \leq k \leq \frac{4n}{5}$, any sets S, T such that $|S| = k, |T| = Ak$, we consider the probability that $p(S, T) = Pr[\Gamma(S) \subset T]$ where we define $\Gamma(S) = \{u \in T | \exists v \in S : (v, u) \in E\}$ that is the set of vertices in T which are reachable from S. Specifically, let $S = \{v_1, \ldots, v_k\} \subset [1, n]$. Then, we have $p(S, T) = \prod_{v_i \in S} {p_i}^d$, as the graph that we consider is d-regular. Let $q(k)$ be the probability that there exist some sets S and T ($|S| = k$ and

$|T| = Ak$) where vertex expansion is not satisfied (then, the graph is not a (K, A) vertex expander). Specifically, we have

$$\begin{aligned} q(k) &= \binom{n}{k}\binom{n}{Ak}p(S,T) \\ &\leq (\frac{en}{k})^k(\frac{en}{Ak})^{Ak}p(S,T). \end{aligned}$$

There are two cases. First, if $k < \frac{2\delta n-1}{A+1}$, according to Lemma 14, we have $\frac{1}{k}\sum p_i \leq \frac{Ak}{2\delta n} < 1$. Therefore, we can pick up a constant c ($c > 1$) such that $\frac{Ak}{2\delta n} < \frac{1}{c} < 1$. Then, we have $1 - \sqrt{\frac{Ak}{2\delta n}} \geq 1 - \sqrt{\frac{1}{c}}$. Due to Lemma 12, we have:

$$\begin{aligned} q(k) &\leq (\frac{en}{Ak})^{Ak}(\frac{en}{k})^k(\sqrt{\frac{Ak}{2\delta n}})^{(1-\sqrt{\frac{1}{c}})dk} \\ &\leq \frac{e^{(A+1)k}}{A^{Ak}}(\frac{2\delta}{A})^{(A+1)k}(\frac{Ak}{2\delta n})^{\frac{1}{2}(1-\sqrt{\frac{1}{c}})dk-(A+1)k} \\ &\leq 2^{-100k} \end{aligned}$$

as $\frac{Ak}{2\delta n} < 1$ and when d is a large enough constant (depending only on δ).

In the second case, when $\frac{4n}{5} \geq k \geq \frac{1.9\delta n}{A+1}$ (note that $\frac{2\delta n-1}{A+1} > \frac{1.9\delta n}{A+1}$ when n is large), we have

$$\frac{en}{k} \leq \frac{(A+1)e}{1.9\delta}.$$

Due to Lemma 13, we know that $\frac{1}{k}\sum p_i < 1 - \frac{5\delta}{16}$. Therefore, according to Lemma 12,

$$\begin{aligned} q(k) &\leq (\frac{(A+1)e}{1.9\delta})^k(\frac{(A+1)Ae}{1.9\delta})^{Ak}\sqrt{(1-\frac{5\delta}{16})}^{(1-\sqrt{1-\frac{5\delta}{16}})dk} \\ &\leq 2^{-100k} \end{aligned}$$

when d is a large enough constant (depending only on δ). Then, we have $\sum_{k=1}^{\frac{4n}{5}} 2^{-100k} < 1$. □

Proofs of Lemmas.

Proof of Lemma 13. Consider the set S that is an interval. We compute $\frac{1}{k}\sum p_i$ by counting on each element in T. Specifically, there are $k - 2\delta n + 1$ elements that each connects with

exact $2\delta n$ elements of S. Moreover, there are 2 elements connecting with exact $2\delta n - 1$ elements; 2 elements connecting with $2\delta n - 2$ elements of S and so on.

Therefore, we can come up with a greedy strategy that works as follow. We first select those elements that connect with exact $2\delta n$ elements in S. If there are still openings in Ak elements, then we choose the two elements that connect $2\delta n - 1$ elements of S into T and so on. We claim that our strategy is indeed optimal because otherwise if there exists an optimized solution T' that is different in at least one element in T with ours. Then, we can sort the elements in T' according to how many elements in S connect with it. Then, we will find that the one in T' that is different with ours will have a lower rank. Now, we can switch this element with the element in our solution and the resulting set T'' has a larger $\frac{1}{k}\sum p_i$ than T'. This shows that the set T produced by our strategy is optimal.

Therefore, assuming $t = ak + \delta n$ where $a = \frac{A-1}{2}$, we have:

$$\begin{aligned}
\frac{1}{k}\sum_{i=1}^{k} p_i &= \frac{2\delta n(k - 2\delta n + 1) + 2(2\delta n - 1) + \ldots + 2(2\delta - t)}{2\delta nk} \\
&= \frac{\delta n - \delta^2 n^2 + 2ak\delta n - a^2k^2 - ak + 2\delta nk}{2\delta nk} \\
&= 1 + a - \frac{a}{2\delta n} - \frac{\delta n - 1}{2k} - \frac{a^2 k}{2\delta n} \\
&\leq 1 + \frac{2ak - \delta n + 1}{2k}.
\end{aligned}$$

Moreover, if $k \leq \frac{4n}{5}$,

$$\begin{aligned}
2ak - \delta n + 1 &\leq 2 \cdot \frac{\delta}{4}\frac{4n}{5} - \delta n + 1 \\
&\leq -\frac{3\delta n}{5} + 1 \leq -\frac{\delta n}{2}
\end{aligned}$$

when $n \geq \frac{10}{\delta}$.

Therefore, we have $1 + \frac{2ak - \delta n + 1}{2k} \leq 1 - \frac{\delta n}{4k} \leq 1 - \frac{5\delta}{16}$.

The above argument is applicable when $k \geq 2\delta n$. For $\frac{2\delta n}{A+1} \leq k < 2\delta n$, we have $\frac{1}{k}\sum p_i \leq \frac{\delta}{4} + \frac{3-A}{4\delta n}$. When, $n > \frac{(3-A)}{(1-9\delta/16)4\delta}$ is large enough, we have $\frac{\delta}{4} + \frac{3-A}{4\delta n} < 1 - \frac{5\delta}{16}$.

Now, we consider that S' is not an interval. Let S' be any set of size k but S' is not an interval. We claim that the value of $\max_{S',T} \frac{1}{k}\sum p_i$ is lower than $\max_{S,T} \frac{1}{k}\sum p_i$. We

process S' from right v_1 to left v_k. There are two cases. First we assume that the distance $d(v_1, v_2) = |v_1 - v_2|$ (as $v_i \in [1, n]$) between v_1 and v_2 is $> 2\delta n$. Then, we move elements $\{v_2, v_3, \ldots, v_k\}$ in S towards v_1 with $d(v_1, v_2) - 1$ steps. Moreover, we move $T \cap [-\infty, v_2 + \delta n]$ with $d(v_1, v_2) - 1$ steps. Since the distance $d(v_1, v_2) > 2\delta n$, we don't remove any elements in T that are originally connected with v_1. Moreover, since we move $[v_2, v_k]$ and $T \cap [-\infty, v_2 + \delta n]$ with the same amount steps, this does not change any connection from T to $[v_2, v_k]$. The only change is that when we move elements in T towards v_1, some elements in T are now connected with v_1 and this will increase $\frac{1}{k}\sum p_i$.

For the second case, if $d(v_1, v_2) \leq 2\delta n$, we do the same steps as in the first case. Therefore, those elements in T that connect with $[v_2, v_k]$ will still connect with them. However, we need to carefully count on elements in T that connects with v_1. We first note that when we move elements in T as $d(v_1, v_2) < 2\delta n$, those elements $T \cap [v_1 - \delta n, v_2 + \delta n]$ still are $\subset [v_1 - \delta n, v_1 + \delta n]$. On the other hand, they may now overwrite some elements that does not move (because they are $\in [v_2 + \delta n + 1, v_1 + \delta n]$), then we have some openings. But consider that, those elements are only related with v_1 (i.e., no elements in S except v_1 connect with them before the movement), we can do the greedy strategy to assign the openings. As a summary, for each of the elements moved, they still connect with the same elements in $[v_1, v_k]$; for those elements that are not moved, we assign them to new positions such that they can connect with even more elements of $S = [v_1, v_k]$. Inductively, we show that after we move all elements in S and elements in T, we will have the value of $\frac{1}{k}\sum p_i$ which is not lower than before. This shows that the case S that is an interval maximizes $\frac{1}{k}\sum p_i$. □

Proof of Lemma 14. Consider $|S| = k$ and $|T| = Ak$ as above. For each element in T, it connects with at most $2\delta n$ elements. Therefore, some elements in T may connect all the elements in S as $|S| = k$ and $k < 2\delta n$. There are at most $2\delta n - k + 1$ such elements. As $\frac{2\delta n - 1}{A+1} \geq k$, we know that $Ak \leq 2\delta n + k - 1$. Therefore, we can ask that all elements in T connect all elements of S where $|S| = k$. Then, in this case, we have:

$$\begin{aligned} \frac{1}{k}\sum p_i &= \frac{Ak \cdot k}{2k\delta n} \\ &= \frac{Ak}{2\delta n} < 1 \end{aligned}$$

when n is large enough. □

4.3.2 Our Construction and its Efficiency

Definition 12 (Double Cover Graph)**.** Let $G = (V, E)$ be an undirected graph on n vertices (without loss of generality, let $V = \{1, \ldots, n\}$). Its double cover graph $B(G)$ is a directed bipartite graph $(L \cup R, E')$ such that $L = R = V$. Edges in E' are directed from L to R and therefore $E' \subset L \times R$. For each $(u, v) \in E$, we add (u, v) and (v, u) to E'. Furthermore, we add (i, i) to E' for $i = 1, \ldots, n$.

Note that the input and output sets of $B(G)$ are L and R. We say $B(G)$ is δ-local if G is δ-local.

Definition 13. Let G be an undirected graph on n vertices. Given $h \in \mathbb{N}^+$, a *stack of h copies of G*, denoted $\Gamma(G, h)$, is a layered DAG defined as follows: Let $B_1, \ldots, B_h$ be h copies of the double cover graph $B(G)$, and identify the outputs of B_i with the inputs of B_{i+1} for $1 \leq i < h$, so $\Gamma(G, h)$ has $h + 1$ layers and $n(h + 1)$ vertices.

Now, we are ready to describe our construction:

Definition 14 (Our Construction)**.** The graph $G_{\mathcal{KE}}$ of our scheme is defined by two ingredients: (a) an undirected graph G, which is assumed to be a d-regular, δ-local, $(\frac{4n}{5}, A = 1 + \frac{\delta}{2})$-vertex expander for constants $d \in \mathbb{N}$ and $\delta > 0$; (b) an integer parameter t. Then, $G_{\mathcal{KE}}$ is $\Gamma(G, M)$, where $M = th$, $h = c_A \log n$ and c_A is a constant depending on A.

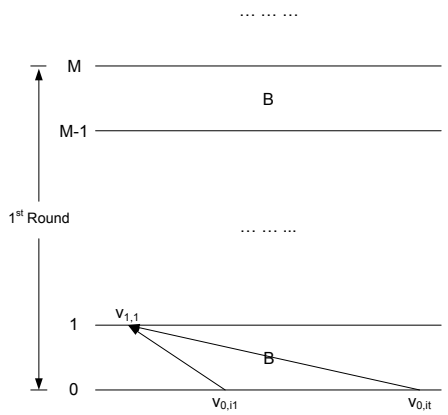

Figure 4.1. Key Evolution Scheme as a Graph $G_{\mathcal{KE}} = \Gamma(G, M)$.

Locality ensures that $G_{\mathcal{KE}}$ can be updated in quasilinear time:

Theorem 15 (Efficiency)**.** *The update function $\mathcal{KE}$ defined by our construction can be computed in time $O(tn \log n)$ (assuming constant-time oracle calls) and $(1 + 2\delta)nw$ space.*

Proof. To evaluate $\mathcal{KE}$, it suffices to evaluate the labels of the outputs of $\Gamma(G, M)$. Let $V_0, \ldots, V_M$ be the M layers of $\Gamma(G, M)$. At the beginning, assign n vertices at V_0 their own values, using the input. To evaluate the j-th vertex at V_1, as $B(G)$ satisfies δ locality, we need to know the values of at most $2\delta n$ vertices (e.g., those in $[j - \delta n, j + \delta n]$). Suppose that we have evaluated the j-th vertex and we now want to evaluate the $(j + 1)$-st vertex in V_1. We need to keep values for at most the inputs in $[j + 1 - \delta n, j + 1 + \delta n]$. Therefore, we can forget the value of the $(j - \delta n)$-th vertex at V_0 and evaluate the $(j + 1)$-th vertex in V_1. Continuing in this manner, we need to keep values of at most $2\delta n + n$ vertices in $\Gamma(G, M)$ in memory. For every k, given that all vertices at V_k are evaluated, the number of random oracle calls to evaluate all vertices at V_{k+1} is $O(n)$. The total number of oracle calls to evaluate $\Gamma(G, M)$ is $O(nM) = O(tn \log n)$ as $M = tc_A \log n$ where c_A is a constant depending on A. □

4.3.3 Security

The parameter t is set based on the class of adversaries that we consider. More specifically, if q is the number of random oracle calls made by an adversary $\mathcal{A}$, then our key evolution scheme can be updated in time $O(n \log n \log q)$, which is quasilinear in n:

Theorem 16 (Security)**.** *For all $w, q, \lambda \in \mathbb{N}$, $c, s > 0$, $0.06 \geq \delta > 0$ and large enough n, if $\frac{4c+s+2\lambda}{w-\log q} \leq 1.12n$ and $t > \frac{\log q}{\log 1.01} + 3$, then the key evolution scheme $\mathcal{KE}$ is $(\frac{q}{2^w} + 2^{1-\lambda})$-secure against (c, s, q) adversaries in the random oracle model.*

Proof. This is a corollary to Theorem 18. When $(1.01)^{t-3} > q$ and n is large enough, we have:

$$q < (1.01)^{t-3} < \frac{n}{2d+1}(\frac{2.1n}{2n+4.1})^{t-3}.$$

Therefore, we can set $t > \frac{\log q}{\log 1.01} + 3$. □

Note that our key evolution scheme is meaningful when $s > (1 + 2\delta)nw$, where δ can be arbitrarily small. The DKW scheme is $(\frac{q}{2^w} + 2^{-\lambda})$-secure when $\frac{4c+s+2\lambda}{w-\log q} < 1.5n$, which exhibits a better leading constant than our bound, $\frac{4c+s+2\lambda}{w-\log q} < 1.12n$. However, their scheme needs $O(n^2)$ time to update while our scheme requires a quasilinear time only. In summary, if $4c + s \leq 1.12|y|$, our scheme does exist, is secure according to Definition 8 and can be updated in a quasilinear time.

4.4 Pebbling Games and Random Oracle Models

We apply graph theory, specifically pebbling games [DNW], to show the security in the random oracle model. Specifically, we can translate any computation involving random oracle calls into a pebbling game where we can pebble colors on a graph. For example, considering $r = H(r_1, r_2)$ where H is a random oracle, if the adversary $\mathcal{A}$ knows the value of r, intuitively, $\mathcal{A}$ should also know both r_1 and r_2. Otherwise, the probability that $\mathcal{A}$ can guess r correctly is negligible. To capture this, we can construct a graph (V, E) such that $V = \{v, v_1, v_2\}$ and $E = \{(v_1, v), (v_2, v)\}$. The values associated with v, v_1, v_2 are $r(v) = r, r(v_1) = r_1, r(v_2) = r_2$. Then, we define a pebbling strategy by placing a color (i.e., a pebble) on $u \in V$. For $u \in V$ with indegree 0, we can place a pebble if we know the value $r(u)$; for $u \in V$ with indegree $\neq 0$, if all predecessors of u have been pebbled, we can place a pebble on u. In our example, v can be pebbled, if v_1 and v_2 get pebbled first, because both (v_1, v) and (v_2, v) are in E. The above rules to pebble a vertex v capture the computation process how we evaluate $r(v)$ via H. More generally, we can define a pebbling game as follows with two colors: black and red (with respect to $\mathcal{A}_s$ and $\mathcal{A}_b$).

Definition 15 (Pebbling Rules)**.**

1. A red pebble can be placed on any vertex already containing a black pebble.
2. If all predecessors of a vertex v are pebbled (each may contain different colors), a black pebble can be placed on v.
3. A black pebble can be removed from any vertex.

Definition 16. Let $G_{\mathcal{KE}}$ be a key evolution scheme (as a graph). $\overline{G}$ is defined as an infinite stack of copies of $G_{\mathcal{KE}}$. For $k \in \mathbb{N}^+ \cup \{0\}$, we define V_k to be the set of vertices on the k-th layer of $\overline{G}$. Moreover, $V_{\geq k} = \cup_{i=k}^{+\infty} V_i$.

Considering a (c, s, q) adversary $\mathcal{A} = (\mathcal{A}_s, \mathcal{A}_b)$ that plays with $\mathbf{Game}_8$, we will have a transcript on when $\mathcal{A}$ calls the random oracle H and with which inputs. The input values are associated with vertices in $\overline{G}$. As we have discussed, we can convert this transcript into a pebbling strategy consisting of a series of pebbling moves. The resulting pebbling strategy is called ex-post-facto [DNW, DKW]. In fact, we can show that, for all values that $\mathcal{A}$ evaluates via the random oracle, the associated vertices will be pebbled as well in the ex-post-facto strategy in $\overline{G}$, with exactly the same order of random oracle calls [DNW, DKW]. Moreover, given c and s, we can bound the number of pebbles used by the ex-post-facto strategy:

Definition 17 (Δ-bounded)**.** Let B_u be the maximum number of black pebbles on $V_{\geq uM}$ in the u-th round. Let R_u be the number of times that rule 1 (Definition 15) is applied in the u-th round. Given $\Delta \in \mathbb{R}$, we say that a pebbling strategy Ψ is Δ-bounded if for each round $u \geq 0$:

$$2R_u + B_u \leq \Delta.$$

The following lemma follows from [DKW]. In this thesis, we explicitly investigate the relationship between the number of random oracle calls q (made by $\mathcal{A}$) and the number of moves in the resulting ex-post-facto strategy:

Lemma 17. *Consider the key evolution scheme as a graph $G_{\mathcal{KE}}$ with a constant degree d and $H : \{0,1\}^{dw} \to \{0,1\}^w$ as a random oracle. Let $\mathcal{A} = (\mathcal{A}_s, \mathcal{A}_b)$ be a (c, s, q) adversary in the random oracle model. Let Ψ be its ex-post-facto strategy. Then, for any $\lambda > 0$, Ψ is valid consisting of at most $(2d+1)q$ moves and is $\frac{4c+s+2\lambda}{w-\log q}$-bounded with probability at least $1 - \frac{2}{2^\lambda} - \frac{q}{2^w}$. The probability is over the choice of the random oracle H and the 0-th round key y_0.*

Proof. This is a corollary of Theorem 4.10 [DKW]. Specifically, given each oracle call made by $\mathcal{A}$, the reduction algorithm will generate at most two moves (i.e., placing a red pebble and then removing a black pebble) for each inputs of the call. Moreover, it produces one move for each output. As the random oracle $H : \{0,1\}^{dw} \to \{0,1\}^w$ takes d inputs and there are at most q oracle calls made by $\mathcal{A}$, the reduction algorithm can produce at most $(2d+1)q$ moves. □

4.5 Security Analysis

Theorem 18. *For all $w, t, d \in \mathbb{N}$, $\lambda > 0$, let $H : \{0,1\}^{dw} \to \{0,1\}^w$ be a random oracle. If $\frac{4c+s+2\lambda}{w-\log q} \leq 1.12n$ and $q \leq \frac{n}{2d+1}(\frac{2.1n}{2n+4.1})^{t-3}$, then our key evolution scheme $G_{\mathcal{KE}}$ is $(2^{1-\lambda} + \frac{q}{2^w})$-secure against any (c, s, q) adversaries in the random oracle model (Definition 8).*

To prove Theorem 18, we notice that, based on Lemma 17, the transcript for any adversary can produce a valid Δ-bounded pebbling strategy with high probability. Therefore, if we can show that, for any Δ-bounded pebbling game, no one can win as Definition 8, then we proved Theorem 18. More specifically, in term of pebbling game, at the end of any round u, Definition 8 asks that no one can pebble any vertices on $V_{(u+1)M}$ and then outputs the u-th round key.

To prove this, we show two lower bound results. The first lower bound shows that if the adversary can pebble any vertex on $V_{(u+1)M}$, at some point t, there do exist many pebbles between $V_{(u+1)M}$ and V_{uM}. This follows from the fact that, at each round u, $G_{\mathcal{KE}}$ is a stack of t copies of a n-superconcentrator $\Gamma(G, h)$. The second lower bound shows that if the adversary can pebble all vertices at V_{uM} at the end of u-th round, at any time, there must be sufficient pebbles on or below V_{uM}. This is based on the fact that G is a vertex expander and h is large enough. However, based on these two lower bound results, at the point t, the number of pebbles on $\overline{G}$ exceeds the Δ-bounded assumption for a given Δ, which shows a contradiction.

Proof of Theorem 18. When $\Pr[E] = 0$, by Definition 8, $\mathbf{Adv}_{\mathcal{A}}^{\mathbf{Game}_8} = 0$. Therefore, we can assume that $\Pr[E] \neq 0$. Let E' be the event that the ex-post-facto strategy is valid, is $(\frac{4c+s+2\lambda}{w-\log q})$-bounded and consists of at most $n(\frac{2.1n}{2n+4.1})^{t-3}$ moves.

$$\begin{aligned}\mathbf{Adv}_{\mathcal{A}}^{\mathbf{Game}_8} &= \Pr[E_1|E] = \Pr[E_1|E' \wedge E]\Pr[E'|E] + \Pr[E_1|\overline{E'} \wedge E]\Pr[\overline{E'}|E] \\ &\leq \Pr[E_1|E' \wedge E] + \Pr[\overline{E'}|E] \\ &\leq 0 + 2^{1-\lambda} + \frac{q}{2^w}.\end{aligned}$$

If $\mathcal{A}$ evaluates the value of any vertex v in $V_{\geq(u+1)M}$ in the u-th round, v will also be pebbled in the ex-post-facto strategy. Moreover, since $\mathcal{A}$ makes at most q queries where $q \leq \frac{n}{2d+1}(\frac{2.1n}{2n+4.1})^{t-3}$, the generated ex-post-facto strategy consists of at most $n(\frac{2.1n}{2n+4.1})^{t-3}$ moves. Therefore, if the ex-post-facto graph is valid with $1.12n$-bounded and with at most $n(\frac{2.1n}{2n+4.1})^{t-3}$ moves, according to Theorem 24 part (3), we will not pebble any vertex in $V_{\geq(u+1)M}$. Therefore, the probability $\Pr[E_1|E' \wedge E] = 0$. $\Pr[\overline{E'}|E] = 2^{1-\lambda} + \frac{q}{2^w}$ is due to Lemma 17. □

In the following sections, we first prove that $\Gamma(G, h)$ is a n-superconcentrator. Then, we show the first and the second lower bound results.

4.5.1 n-Superconcentrator

Definition 18 (n-superconcentrator)**.** A graph G with input set I and output set O ($|I| = |O| = n$) is a n-superconcentrator provided that given any $S \subset I$ and $T \subset O$ with $|S| = |T| = k \leq n$, there are k vertex disjoint paths connected with S and T.

We seek a n-superconcentrator construction that is a layered graph such that to pebble vertices in the i-th layer requires vertices in the $(i-1)$-th layer only. Moreover, it is built upon good expanders with $A > 1$ and therefore the number of layers can be logarithmic in n. The following lemma follows from the max-flow-min-cut theorem, whose proof can be found in [Val75].

Lemma 19. *Let G be a d-regular layered graph with the height h. Let I and O be its input set and output set such that $|I| = |O| = n$. Specifically, I is the 0-th layer and O is the $(h+1)$-th layer. For any $k \leq n$, for every $S \subset I$ and $T \subset O$ with $|S| = |T| = k$: there exist k vertex-disjoint paths from S to T if and only if removing any set of k-1 vertices from $V(G) - I - O$ cannot disconnect S from T.*

Lemma 20. *There exists $c > 0$, $\forall A > 1$, if $h \geq c \log_A n$, $\Gamma(G, h)$ is a n-superconcentrator when G is a $(\frac{4n}{5}, A)$ vertex expander.*

Proof. Due to Lemma 19, we have to prove that for any $k \leq n$, the set of U (where $|U| < k$) cannot disconnect S from T. Specifically, let $U_i \subset U$ is the set of vertices of U at the i-th layer. Let $A_0 = S$ and $B_0 = T$. $A_{i+1} = \Gamma(A_i) - U_{i+1}$ where $\Gamma(A_i)$ is the set of vertices at the $(i+1)$-th layer that are reachable from A_i. Similarly, we define $B_{i+1} = \Gamma(B_i) - U_{i+1}$. Moreover, we let $a_i = |A_i|$, $b_i = |B_i|$ and $u_i = |U_i|$. Then, for all i such that $a_i \leq \frac{4n}{5}$, we have $a_{i+1} \geq Aa_i - u_{i+1}$ because G is $(\frac{4n}{5}, A)$ vertex expander.

Let $t_i = a_i + b_{h-i}$. We want to show that $\frac{1}{h}\sum_{i=1}^{h} t_i > n$. Then, there exists $i^* \in [1, h]$ such that $t_{i^*} > n$. Otherwise, we have for all i: $t_i < n$ and therefore $\frac{1}{h}\sum t_i < n$ which leads a contradiction. On the other hand, $a_{i^*} + b_{h-i^*} > n$ implies that $|A_{i^*} \cap B_{h-i^*}| \neq \emptyset$ because each layer i^* contains n vertices. To show $\frac{1}{h}\sum_{i=1}^{h} t_i > n$, we can simply show that both $\frac{1}{h}\sum_{i=1}^{h} a_i > \frac{n}{2}$ and $\frac{1}{h}\sum_{i=1}^{h} b_i > \frac{n}{2}$.

Now we prove $\frac{1}{h}\sum_{i=1}^{h} a_i > \frac{n}{2}$ (the proof to $\frac{1}{h}\sum_{i=1}^{h} b_i > \frac{n}{2}$ can be done similarly.) We first assume that $a_0 = k \geq \frac{4n}{5}$. Otherwise, we can use expansion property to grow it up to $\frac{4n}{5}$ in $\Theta(\log n)$ layers. Specifically, for any t, if $a_{t-1} < \frac{4n}{5}$: we have $a_t > Aa_{t-1} - u_t$. Iteratively,

$$\begin{aligned} a_t &> A^t a_0 - \sum_{j=1}^{t} A^{t-j} u_j \\ &> A^t a_0 - A^{t-1} \sum_{j=1}^{t} u_j \\ &> A^t a_0 - A^{t-1}(k-1) > A^{t-1}. \end{aligned}$$

If we set $t \geq \log_A(\frac{4n}{5}) + 1$, $a_t > A^{t-1} \geq \frac{4n}{5}$.

Now, let's assume $a_0 > \frac{4n}{5} > \frac{3n}{4} > \frac{n}{2}$. Then, we investigate $a_1, a_2, \ldots$. We say that a_i is *bad* if $a_i < \frac{3n}{4}$. Once we encounter a *bad* a_i, we will grow it up to $\frac{4n}{5}$ again by the above analysis.

We claim that there are at most 20 bad a_i(s). This is true because each time it takes at least $\frac{4n}{5} - \frac{3n}{4} = \frac{n}{20}$ vertices in U to bring $a_i > \frac{4n}{5}$ to some $a_j < \frac{3n}{4}$. Therefore, this can happen at most $\frac{k}{n/20} \leq \frac{n}{n/20} = 20$ times.

Let $h = c \log_A n$ for some constant c that will decide soon. We want $\frac{1}{h}\sum_{i=1}^{h} a_i > \frac{n}{2}$. Specifically, we need

$$\begin{aligned}\frac{1}{h}\sum_{i=1}^{h} a_i &\geq \frac{[h - 20(t+1)]\frac{3n}{4}}{h} \\ &> \frac{[c\log_A n - 20(\log_A n + 2)]\frac{3n}{4}}{c\log_A n} > \frac{n}{2}\end{aligned}$$

when $c > 60$ and n is large enough. □

4.5.2 Lower Bound Results and Security Proof

In this section, we show the first lower bound via a modified Basic Lower Bound Argument (BLBA) lemma. Then, we introduce necessary tools (e.g., optimal width) to prove for the second lower bound (Theorem 24) and consequently the main security theorem (Theorem 18).

Lemma 21 ((modified) BLBA Lemma [LT82])**.** *In order to pebble $S_b + S_e + 1$ outputs of a n-superconcentrator, starting with a configuration of at most S_b black and red pebbles and finishing with a configuration of at most S_e black and red pebbles on the graph, at least $N - S_b - S_e$ different inputs of the graph have to be pebbled and unpebbled.*

Proof. We prove it by contradiction. Suppose the claim is not true, then $\geq S_b + S_e + 1$ different inputs are not both pebbled and unpebbled. On the other hand, since $S_b + S_e + 1$ outputs of a n-superconcentrator are pebbled, there are $S_b + S_e + 1$ vertex-disjoint paths connecting those $S_b + S_e + 1$ inputs and outputs. Those vertex-disjoint paths have to be pebbled at some point. Initially there are at most S_b black and red pebbles and at the end there are at most S_e black and red pebbles, therefore, at least one of the path does not receive pebbles at the beginning point and at the ending point. Therefore, the input on this path has to be pebbled and unpebbled, which is contradiction with our assumption. □

Lemma 22 (First Lower Bound)**.** *Let S be the maximum number of pebbles on $Y_{u+1} = \bigcup_{i=uM+1}^{(u+1)M} V_i$ at any configuration of the u-th round. Let $T(n, M, S)$ be the minimum number of moves needed to pebble one output of $V_{(u+1)M}$ during the u-th round. Assuming that at the initial configuration of the u-th round, there is no pebble on or above V_{uM}, then,*

$$T(n, M, S) > n(\frac{n-2S}{2S+1})^{t-3}.$$

Proof. We know that at the initial configuration of the u-th round, there are no pebbles on or above V_{uM}. In order to pebble any one output of $V_{(u+1)M}$, we have to pebble all its predecessors. Specifically, Y_{u+1} consists of t n-superconcentrators $C_1, \ldots, C_t$. V_{uM} is the input set of C_1 and $V_{(u+1)M}$ is the output set of C_t. Furthermore, C_t is empty initially. Therefore, we have to pebble all n inputs of C_t. On the other hand, the input set of C_t is identical to the output set of C_{t-1}. By applying (modified) BLBA, we need to pebble and unpebble $\lfloor \frac{n}{2S+1} \rfloor (n-2S)$ inputs of C_{t-1}. Moreover, we know that the output set of C_{t-2} is identical to the input set of C_{t-1}. In order to pebble $\lfloor \frac{n}{2S+1} \rfloor (n-2S)$ outputs of C_{t-2}, starting with any configuration on the graph, we can apply (modified) BLBA on C_{t-2}. We have $\lfloor \frac{n}{2S+1} \frac{n-2S}{2S+1} \rfloor (n-2S)$ of the inputs of C_{t-2} have to be pebbled and unpebbled. Iterating this BLBA argument to $C_{t-3}, \ldots, C_1$, we have

$$T(n, M, S) \geq \lfloor n(\frac{n-2S}{2S+1})^{t-3} \rfloor.$$

□

Definition 19 (Optimistic Width)**.** Let $\Gamma(G, h)$ be the n-superconcentrator (cf. Definition 13) that is built from a $(\frac{4n}{5}, A)$-vertex expander. Then, the optimistic width of a list of integers $(a_0, \ldots, a_{k-1})$ w.r.t. $\Gamma(G, h)$ is, $OptWidth(a_0, \ldots, a_{k-1}) = (b_0, \ldots, b_{k-1})$ where:

$$b_i = \begin{cases} a_0 & i = 0; \\ \min\{n, a_i + \frac{b_{i-1}}{A}\} & i > 0 \text{ and } b_{i-1} < \frac{4nA}{5}; \\ \min\{n, a_i + b_{i-1}\} & i > 0 \text{ and } b_{i-1} \geq \frac{4nA}{5}. \end{cases}$$

Definition 20. Let G be a layered graph with k layers. Then, the projection of $V' \subset V(G)$: $proj(V') = (|V' \cap V_0|, \ldots, |V' \cap V_{k-1}|)$ (V_i is the set of vertices at the i-th layer of G).

Definition 21. Let $G = (V, E)$ be a graph. $\forall S \subset V$, the closure of S, denoted $[S]$, is defined recursively: (i) if $v \in S$, $v \in [S]$; (ii) if all children of v (i.e., $\{u|(u, v) \in E\}$) are in

$[S]$, $v \in [S]$.

Intuitively, $[S]$ includes all possible pebbles that can be derived from S. For any $k \in \mathbb{N}^+$, Let $\vec{v} = (v_0, \ldots, v_{k-1}) \in \mathbb{R}^k$ and $\vec{u} = (u_0, \ldots, u_{k-1}) \in \mathbb{R}^k$ be any two vectors. We say $\vec{v} \geq \vec{u}$ if and only if $v_i \geq u_i$ for all $i = 0, \ldots, k-1$.

Lemma 23. *The optimistic width w.r.t. $\Gamma(G, h)$ satisfies the following two properties.*

1. ***Upper Bound:*** *For any $U \subset V(\Gamma(G, h))$, $OptWidth(proj(U)) \geq proj([U])$.*
2. ***Addition:*** *Let $U, W \subset V(\Gamma(G, h))$. Let $(s_0, \ldots, s_{k-1}) = OptWidth(proj(U))$ and Let $(t_0, \ldots, t_{k-1}) = OptWidth(proj(U \cup W))$. Then, $0 \leq t_i - s_i \leq |W|$ for $i = 0, \ldots, k-1$.*

Proof. First, we prove for the upper bound property. Let $(s_0, \ldots, s_{k-1}) = OptWidth(proj(U))$ and $(t_0, \ldots, t_{k-1}) = proj([U])$. We have to show $s_i \geq t_i$ for all $i = 0, \ldots, k-1$. We prove this by introduction. To do so, we assume $(\alpha_0, \ldots, \alpha_{k-1}) = proj(U)$ and $(\beta_0, \ldots, \beta_{k-1}) = proj([U])$. For the base case $i = 0$, we see that $s_0 = \alpha_0 = \beta_0 = t_0$.

Suppose that the claim is true for $i-1$, we have $s_{i-1} \geq t_{i-1}$ (i.e., s_{i-1} is a upper bound on the number of pebbles can be derived at the $(i-1)$-th layer of U). We want to show a upper bound on the number of pebbles that can be derived at the i-th layer. Let S_0 be the set of pebbles at the i-th layer that can be derived from the $(i-1)$-th layer. On the other hand, there are a_i pebbles at the i-th layer of U. Let S_1 be the set of a_i pebbles at the i-th layer. At the worst case, $S_0 \cap S_1 = \emptyset$. Therefore, $t_i \leq \min\{a_i + |S_0|, n\}$. Now, we have to bound on $|S_0|$.

This can be done on $\Gamma(G, h)$ where G is a $(\frac{4n}{5}, A)$ vertex expander. Recall that the graph between the i-th (for any i) and the $(i+1)$-th layers is a bipartite graph $B(G)$. Let $L(B(G))$ be the left side of $B(G)$ (i.e., that corresponds to the $(i-1)$-th layer) and $R(B(G))$ be the right side of $B(G)$ (i.e., that corresponds to the i-th layer). Specifically, for any set $V \subset L(B(G))$ and $|V| \leq \frac{4n}{5}$, we need at least $A|V|$ pebbles (in $R(B(G))$) to derive it. In this case, we have $|S_0| \leq \frac{s_{i-1}}{A}$ provided that $s_{i-1} < \frac{4An}{5}$. On the other hand, if $|V| \geq \frac{4n}{5}$, we know that s_{i-1} pebbles can derive at most s_{i-1} pebbles, which shows the upper bound property.

Now we prove for the addition property. We prove it via adding elements of W into U one by one. Specifically, suppose that the added element e is in the j-th layer. Let $(s_0, \ldots, s_{k-1}) = OptWidth(proj(U))$ and $(s'_0, \ldots, s'_{k-1}) = OptWidth(proj(U \cup \{e\}))$. Moreover, let $(a_0, \ldots, a_{k-1}) = proj(U)$. Then, for $i < j$, $s'_i = s_i$. For $i = j$, without loose of generality, we

assume $s'_{j-1} < \frac{4nA}{5}$, we have

$$\begin{aligned} s'_j &\leq \frac{s'_{j-1}}{A} + a_j + 1 \\ &\leq \frac{s_{j-1}}{A} + a_j + 1 \\ &= s_j + 1 \end{aligned}$$

as $A > 1$ and $s'_{j-1} = s_{j-1}$. Then applying it to $i = j + 1$ and, without loose of generality, assuming $s'_j \leq \frac{4n}{5}$, we have:

$$\begin{aligned} s'_{j+1} &\leq a_{j+1} + \frac{s'_j}{A} \\ &\leq a_{j+1} + \frac{(s_j + 1)}{A} \\ &\leq s_{j+1} + 1 \end{aligned}$$

as $A > 1$. Applying the same argument to $i = j + 2, \ldots, k - 1$, we complete the proof. □

Definition 22 (Fully Covering Assumption)**.** Let r^*_u be the last configuration of the u-th round ($u \geq 0$). Then, all vertices on V_{uM} have to be pebbled at r^*_u.

Definition 23. A vertex is **heavy** if it contains a black pebble or a red pebble derived using the pebbling rule 1.

Theorem 24. *For $u \geq 0$, let r^*_u be the last configuration of the u-th round. Let Heavy be the set of heavy pebbles. Define Q_u to be the set of pebbles at r^*_u except black pebbles at the uM-th layer. Let $Y_u = \cup_{i=(u-1)M+1}^{uM} V_i$. Under the fully covering assumption and Δ-bounded assumption ($1.12n > \Delta > n$), for every round u and every configuration r at the u-th round, we have:*

1. *$P_1(u)$: $OptWidth(proj(Q_u \cap Heavy)) < (2(\Delta - n), \ldots, 2(\Delta - n), \Delta - n, \ldots, \Delta - n, 1, \ldots, 1)$.*
2. *$P_2(u)$: (Second Lower Bound) Let $P(u, r)$ be the set of heavy red pebbles in Y_u derived in the u-th round and black pebbles in Y_u at the configuration r. Then,*

$$|P(u, r)| > 2n - \Delta.$$

3. $P_3(u)$: *For any adversary $\mathcal{B}$ that makes at most T moves, at the end of round u ($u \geq 0$), there are no pebbles on or above $V_{(u+1)M}$, provided that $T \leq n(\frac{2.1n}{2n+4.1})^{t-3}$ (recall that $M = th$).*

4. $P_4(u)$: *Let S_u be the number of red pebbles on V_{uM} of the configuration r_u^*. Let S_u^* be the number of heavy red pebbles in Y_u derived in the u-th round. Then,*

$$S_u \leq S_u^*.$$

Proof. We prove it by induction on round number u. For $u = 0$, we have $|P(u, r)| = n > 2n - \Delta$ if $\Delta > n$. Moreover, $OptWidth(proj(Q_u \cap Heavy)) = OptWidth(proj(\emptyset)) < (2(\Delta - n), \ldots, 2(\Delta - n), \Delta - n, \ldots, \Delta - n, 1, \ldots, 1)$. Initially, there are n pebbles on V_0 at the end of 0-th round. Therefore, $P_3(0)$ and $P_4(0)$ are trivially true. We assume the claim is true for $u \leq k$ and we prove the claim still holds for $(k+1)$-th round. Specifically, we prove the following lemmas hold:

1. $P_1(k) \Rightarrow P_2(k+1)$;
2. $P_3(k) \bigwedge P_2(k+1) \Rightarrow P_3(k+1)$;
3. $P_4(k) \bigwedge P_2(k+1) \Rightarrow P_4(k+1)$;
4. $P_1(k) \bigwedge P_3(k+1) \bigwedge P_4(k+1) \Rightarrow P_1(k+1)$.

□

$P_1(k) \Rightarrow P_2(k+1)$. We have $proj([(Q_k \cup P(k+1, r)) \cap Heavy]) = proj([Q_k \cup P(k+1, r)])$ because all non-heavy pebbles are derived by heavy red ones. By the fully covering assumption, at the end of $(k+1)$-th round, $V_{(k+1)M}$ will be fully covered (by n pebbles). Therefore, the $(k+1)M$-th entry: $proj([Q_k \cup P(k+1, r)])_{(k+1)M} = n$. Hence, we have $proj([(Q_k \cup P(k+1, r)) \cap Heavy])_{(k+1)M} = n$. On the other hand, $(Q_k \cup P(k+1, r)) \cap Heavy = (Q_k \cap Heavy) \cup P(k+1, r)$ because $P(k+1, r)$ contains heavy pebbles only. By Lemma 23 part (1), we have

$$\begin{aligned}
&OptWidth(proj((Q_k \cap Heavy) \cup P(k+1, r)))_{(k+1)M} \\
&= OptWidth(proj((Q_k \cup P(k+1, r)) \cap Heavy))_{(k+1)M} \\
&\geq proj([(Q_k \cup P(k+1, r)) \cap Heavy])_{(k+1)M} = n.
\end{aligned}$$

However, by $P_2(k)$, we have $OptWidth(proj(Q_k \cap Heavy))_{(k+1)M} < \Delta - n$. Therefore, $|P(k+1,r)| > n - (\Delta - n) = 2n - \Delta$ by Lemma 23 part (2). □

$P_3(k) \bigwedge P_2(k+1) \Rightarrow P_3(k+1)$. We prove it by contradiction. Assume that there exists an adversary $\mathcal{B}$ that makes at most $n(\frac{2.1n}{2n+4.1})^{t-3}$ moves can pebble some vertex in $V_{\geq(u+1)M}$. Since $P_3(k)$ is true, at the beginning of the $(k+1)$-th round, there are no pebbles on or above $V_{(k+1)M}$. Therefore, by the second lower bound lemma (Lemma 22), there exists a configuration r in the $(k+1)$-th round, such that $\mathcal{B}$ needs $> S = \frac{n}{4.1}$ space in Y_{k+2}. On the other hand, By $P_2(k+1)$, in the configuration r, there are $|P(k+1,r)| > 2n - \Delta$ pebbles in Y_{k+1}. $Y_{k+1} \cap Y_{k+2} = \emptyset$. Therefore, the space needed in that configuration r is $> 2n - \Delta + \frac{n}{4.1} \geq \Delta$ when $\Delta \leq 1.12n$, which shows a contradiction, given the Δ-bounded assumption. □

$P_4(k) \bigwedge P_2(k+1) \Rightarrow P_4(k+1)$. First, we prove that for any configuration r of the $(k+1)$-th round, there is no layer with $\geq \frac{4n}{5}$ red pebbles between the $(kM+1)$-th and the $(k+1)M$-th layer. We prove it by contradiction. Suppose that there exists a layer V_m such that there are $\geq \frac{4n}{5}$ red pebbles on it. Let T_0 be the set of heavy red pebbles between V_{kM+1} and V_m and T_1 be the set of heavy red pebbles on V_{kM}. By the property of n-superconcentrator, $|T_0|+|T_1| \geq \frac{4n}{5}$. Therefore, either $|T_0| \geq \frac{4n}{5} - \frac{\Delta}{2}$ or $|T_1| \geq \frac{\Delta}{2}$. We prove that both $|T_0| < \frac{4n}{5} - \frac{\Delta}{2}$ and $|T_1| < \frac{\Delta}{2}$ when $\Delta \leq 1.12n$. We prove both of them by contradiction. First, we prove $|T_0| < \frac{4n}{5} - \frac{\Delta}{2}$ given Δ-bounded assumption and $P_2(k+1)$. We assume that $|T_0| \geq \frac{4n}{5} - \frac{\Delta}{2}$. By definition, we have $|T_0| \leq R_{k+1}$. By Δ-bounded assumption, we have $2R_{k+1} + B_{k+1} < \Delta$. Therefore, $R_{k+1} + B_{k+1} < \Delta - (\frac{4n}{5} - \frac{\Delta}{2}) = \frac{3X}{2} - \frac{4n}{5}$. On the other hand, according to $P_2(k+1)$: $|P(k+1,r)| > 2n - \Delta$. As $|P(k+1,r)| \leq R_{k+1} + B_{k+1} < \frac{3X}{2} - \frac{4n}{5}$, this leads to a contradiction because $2n - \Delta \geq \frac{3X}{2} - \frac{4n}{5}$ when $\Delta \leq 1.12n$.

Then, we prove for $|T_1| < \frac{\Delta}{2}$ by induction hypothesis $P_4(k)$ and Δ-bounded assumption. On the contrary, we assume that $|T_1| \geq \frac{\Delta}{2}$. First, by $P_4(k)$, we have $S_k \leq S_k^*$. By definitions, we have $|T_1| \leq S_k$ and $S_k^* \leq R_k$. Therefore, we have $R_k \geq \frac{\Delta}{2}$. On the other hand, by Δ-bounded assumption, we have $R_k < \frac{\Delta}{2}$. This leads to a contradiction.

Now, let t_i be the number of heavy pebbles on $V_{(k+1)M-i}$. We also assume that $S_{k+1} > S_{k+1}^*$. Since there is no layer with $\geq \frac{4n}{5}$ red pebbles between $V_{(k+1)M}$ and V_{kM+1}, we can use the expansion property. Specifically, we know that there are $S_{k+1} - t_0$ non-heavy red pebbles on $V_{(k+1)M}$. They are derived by $A(S_u - t_0) - t_1$ non-heavy red pebbles on $V_{((k+1)M-1}$. Apply

the same argument till V_{kM+1}, we have:

$$S_k \geq A^{M-1}S_k - \sum_{i=0}^{M-1} A^{M-1-i}t_i.$$

On the other hand, we have $\sum_{i=0}^{M-1} A^{M-1-i}t_i \leq A^{M-1}\sum t_i \leq A^{M-1}S^*_{k+1}$. Therefore, we have:

$$S_k \geq A^{M-1}(S_{k+1} - S^*_{k+1}) > A^{M-1} \geq \frac{4n}{5}$$

when $M \geq \log_A \frac{4n}{5} + 1$.

However, by $P_4(k)$, we have $S^*_k \geq S_k \geq \frac{4n}{5}$. On the other hand, by Δ-bounded assumption (for $\Delta \leq 1.12n$), $0.56n \geq \frac{\Delta}{2} > S^*_k$ which implies a contradiction. Therefore, $S^*_{k+1} \geq S_{k+1}$. □

$P_1(k) \bigwedge P_3(k+1) \bigwedge P_4(k+1) \Rightarrow P_1(k+1)$. Let T be the set of heavy pebbles of r^*_{k+1} ; T^- is identical to T except black pebbles on $V_{(k+1)M}$. Then, we have:

$$\begin{aligned} OptWidth(Q_{k+1} \cap Heavy) &= OptWidth((Q_k \cup T^-) \cap Heavy) \\ &= OptWidth((Q_k \cap Heavy) \cup (T^- \cap Heavy)) \\ &= OptWidth((Q_k \cap Heavy) \cup T^-). \end{aligned}$$

On the other hand, we show that $|T^-| < \Delta - n$. By the fully covering assumption, we have $(n - S_{k+1})$ black pebbles on the $(k+1)M$-th layer. By the Δ-bounded assumption, we have:

$$|T^-| + (n - S_{k+1}) < R_{k+1} + B_{k+1}.$$

On the other hand, we have $S_{k+1} \leq S^*_{k+1} \leq R_{k+1}$ by $P_4(k+1)$ and the definition of R_{k+1} (recall that R_{k+1} is the number of heavy red pebbles derived in the $(k+1)$-th round). Therefore, $|T^-| < 2R_{k+1} + B_{k+1} - n < \Delta - n$. Moreover, by $P_3(k+1)$, we know that T^- does not contain any pebbles on or above $V_{(k+2)M}$. By Lemma 23 and $P_1(k)$, we have:

$$\begin{aligned} &OptWidth(OptWidth(Q_{k+1} \cap Heavy) = OptWidth((Q_k \cap Heavy) \cup T^-) \\ &< (2(\Delta - n), \ldots, 2(\Delta - n), 2(\Delta - n), \ldots, 2(\Delta - n), \Delta - n, \ldots, (\Delta - n), 1, \ldots, 1). \end{aligned}$$

□

Chapter 5
(Post-Challenge) Auxiliary Inputs Model for Encryption

Auxiliary input model is proposed by [DGK+10] that models side-channel attacks as one-way functions. It captures the intuition that given any side-channel information, it should be computationally hard to recovery the secret key; otherwise, no security can be guaranteed. [DGK+10] combines IND-CPA security with auxiliary input model to propose the IND-AI-CPA security. They also devise the first public-key encryption construction that is IND-AI-CPA secure.

5.0.3 Motivation for Post-Challenge Auxiliary Inputs

Post-challenge leakage query for PKE: The auxiliary input model is general enough to capture a large class of side-channel leakages. However, there are still shortcomings. For example, in the CCA security model for PKE, the adversary $\mathcal{A}$ is allowed to ask for the decryption of arbitrary ciphertexts *before and after* receiving the challenge ciphertext C^*, in order to maximize the ability of $\mathcal{A}$[1]. But for most leakage-resilient PKE, the adversary $\mathcal{A}$ can only specify and query the leakage function $f(\mathsf{sk})$ *before* getting C^*. In real situations, this is not true. The adversary should be able to obtain more information even after the attack target is known. The main reason for not being able to have *post-challenge leakage queries* (queries from the adversary after the challenge ciphertext is given) is as follows. If we allow $\mathcal{A}$ to specify the leakage function after getting C^*, he can easily embed the decryption of C^* as the leakage function, which will lead to a trivial break to the security game. So,

[1]Sometimes this is known as the CCA2 security, in contrast with the CCA1 security, where the adversary is only allowed to ask the decryption oracle before getting the challenge ciphertext.

the issue is to come up with a model with minimal restriction needed to allow post-challenge leakage query after getting the challenge ciphertext, while avoiding the above trivial attack. Comparing with the existing leakage-resilient PKE, the objective is to increase the ability of the adversary to make the model more realistic and capture a larger class of side-channel attacks.

Leakage from the Encryptor: Another reason for considering post-challenge leakage query is to model the leakage of encryptor. In generating the ciphertext, besides the encryption key, the encryptor requires to pick a random value r in probabilistic encryption schemes. This random value is also critical. If the adversary $\mathcal{A}$ can obtain the entire r, it can encrypt the two challenge messages m_0 and m_1 by itself using r and compare if they are equal to the challenge ciphertext, thus wins the game easily. Therefore, the leakage of this randomness should not be overlooked. We demonstrate the impact of leaking encryption randomness in the following artificial encryption scheme. We use $(\mathtt{Enc}, \mathtt{Dec})$ a leakage-resilient PKE scheme in the auxiliary input model and one-time pad to form a new encryption scheme:

- $\mathtt{Enc}'$: On input a message M and a public key pk, pick a random one-time pad P for M and calculate $C_1 = \mathtt{Enc}(\mathsf{pk}, P), C_2 = P \oplus M$, where $\oplus$ is the bit-wise XOR. Return the ciphertext $C = (C_1, C_2)$.

- $\mathtt{Dec}'$: On input a secret key sk and a ciphertext $C = (C_1, C_2)$, calculate $P' = \mathtt{Dec}(\mathsf{sk}, C_1)$ and output $M = C_2 \oplus P'$.

The randomness used in $\mathtt{Enc}'$ by the encryptor is P and the randomness in $\mathtt{Enc}$. However, leaking the first bit of P will lead to the leakage of the first bit in M. Therefore, leakage from the encryptor helps the adversary to recover the message. Without post-challenge leakage query, the side-channel attacks to the encryption randomness cannot be modeled easily.

In both scenarios, we should avoid the adversary $\mathcal{A}$ submitting a leakage function as the decryption of C^* in the security game (in case of leakage from secret key owner) or to submit a leakage function to reveal the information for the encryption randomness r for a trivial attack (in case of leakage from encryptor). A possible direction is to ask $\mathcal{A}$ to submit a set of functions $\mathcal{F}_0$ before seeing the public key or C^*. After seeing the challenge ciphertext, $\mathcal{A}$ can only ask for the leakage of arbitrary function $f' \in \mathcal{F}_0$. Therefore, f' cannot be the decryption of C^* and cannot lead to a trivial attack for the case of encryption randomness. This restriction is reasonable in the real world since most side-channel attacks apply to the physical implementation rather than the algorithm used (e.g. the leakage method of the power or timing attacks are the same, no matter RSA or ElGamal encryption are applied;

512-bit or 1024-bit keys are used.). Similar restriction was proposed by Yuen *et al.* [YYH12] for leakage-resilient signatures in the auxiliary input model[2]. However, directly applying this idea to PKE, by simply allowing both pre-challenge and post-challenge leakages on sk, is not meaningful. Specifically, as the possible choice of leakage function f' is chosen before seeing the challenge ciphertext C^*, the post-challenge leakage $f'(\mathsf{sk})$ can simply be asked before seeing C^*, as a pre-challenge leakage. Therefore this kind of post-challenge leakage can be captured by slightly modifying the original auxiliary input model and does not strengthen our security model for PKE. Hence, we propose the leakage $f'(r)$ on the encryption randomness of C^* as the post-challenge leakage query. This kind of post-challenge leakage cannot be captured by the existing models. Since we focus on the auxiliary input model in this thesis, we call our new model as the *post-challenge auxiliary input* model.

Practical Threats to Randomness. Finally, we want to stress that information leakage caused by poor implementation of pseudorandom number generator (PRNG) is practical. Argyros and Kiayias [AK12] outlined the flaws of PRNG in PHP. Lenstra *et al.* [LHA+12] inspected millions of public keys and found that some of the weak keys could be a result of poorly seeded PRNGs. Michaelis *et al.* [MMS13] uncovered significant weaknesses of PRNG of some java runtime libraries, including Android. These practical attacks demonstrate the potential weakness of the encryption randomness when using PRNG in practice.

5.0.4 Our Contributions

In this thesis, we propose the post-challenge auxiliary input model for public key encryption. The significance of our post-challenge auxiliary input model is twofold. Firstly, it allows the leakage *after* seeing the challenge ciphertext. Secondly, it considers the leakage of two different parties: the secret key owner and the encryptor. In most leakage-resilient PKE schemes, they only consider the leakage of the secret key. However, the randomness used by the encryptor may also suffer from side-channel attacks. There are some encryption schemes which only consider the leakage on randomness, but not the secret key. Bellare *et al.* [BBN+09] only allows randomness leakage before receiving the public key. Namiki *et al.* [NTY11] only allows randomness leakage before the challenge phase. Therefore our post-challenge auxiliary input model also improves this line of research on randomness leakage. To the best of the authors' knowledge, no existing leakage-resilient PKE schemes consider the leakage of secret key and randomness at the same time. Therefore, our post-challenge

[2]Yuen *et al.* [YYH12] named their model as the selective auxiliary input model, due to similarity to the selective-ID model in identity-based encryption.

auxiliary input model is the *first* model to consider the leakage from both the secret key owner and the encryptor. This model captures a wider class of side-channel attacks than the previous models in the literature. We allow for leakage on the values being computed on, which will be a function of both the encryption random r and the public key pk. Specifically, we allows for $g(\mathsf{pk}, f(r))$ where g is any polynomial-time function and f is any computationally hard-to-invert function. We put the restriction on $f(r)$ to avoid trivial attacks on our security model.

To illustrate the feasibility of the model, we propose a generic construction of CPA-secure PKE in our new post-challenge auxiliary input model (pAI-CPA PKE). It is a generic transformation from the CPA-secure PKE in the auxiliary input model (AI-CPA PKE, e.g. [DGK+10]) and a new primitive called the *strong extractor with hard-to-invert auxiliary inputs*. The strong extractor is used to ensure that given the partial leakage of the encryption randomness, the ciphertext is indistinguishable from uniform distribution. As an independent technical contribution, we instantiate the strong extractor using the extended Goldreich-Levin theorem. Similar transformation can also be applied to identity-based encryption (IBE). Therefore we are able to construct pAI-ID-CPA IBE from AI-ID-CPA IBE (e.g. [YCZY12a]).

Furthermore, we extend the generic transformation for CPA-secure IBE to CCA-secure PKE by Canetti *et al.* [CHK04] into the leakage-resilient setting. The original transformation by Canetti *et al.* [CHK04] only requires the use of strong one-time signatures. However, the encryption randomness of the PKE now includes both the encryption randomness used in IBE and the randomness used in the strong one-time signatures. Leaking either one of them will not violate our post-challenge auxiliary input model, but will lead to a trivial attack (details are explained in Section 5.4.1). Therefore, we have to link the randomness used in the IBE and the strong one-time signatures. We propose to use strong extractor with hard-to-invert auxiliary inputs as the linkage. It is because the strong extractor allows us to compute the randomness of IBE and the strong one-time signature from the same source, and yet remains indistinguishable from uniform distribution. It helps to simulate the leakage of the randomness in the security proof. Our contributions on encryption can be summarized in Figure 5.1.

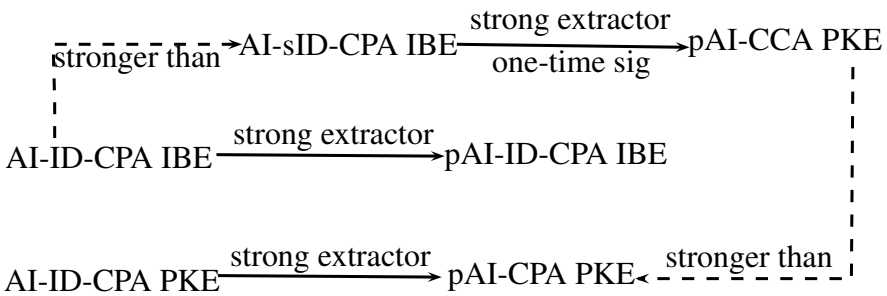

Figure 5.1. Our Contributions on Encryption

5.0.5 Related Works

Dodis *et al.* [DKL09] introduced the model of *auxiliary inputs* leakage functions. PKE secure in the auxiliary input model was proposed in [DGK+10]. Signature schemes secure in the auxiliary input model were independently proposed by Yuen *et al.* [YYH12] and Faust *et al.* [FHN+12], under different restrictions to the security model. All of these works only consider the leakage from the owner of the secret key.

For leakage-resilient PKE, Naor and Segev wrote in [NS09] that

> "It will be very interesting to find an appropriate framework that allows a certain form of challenge-dependent leakage."

Halevi and Lin [HL11] proposed the model for *after-the-fact leakage* which also considered leakage that occurs after the challenge ciphertext is generated. In their entropic leakage-resilient PKE, even if the adversary designs its leakage function according to the challenge ciphertext, if it only leaks k bits then it cannot *amplify* them to learn more than k bits about the plaintext. Halevi and Lin [HL11] mentioned that

> "Our notion only captures leakage at the receiver side (i.e., from the secret key) and not at the sender side (i.e., from the encryption randomness). It is interesting to find ways of simultaneously addressing leakage at both ends."

Recently, Bitansky *et al.* [BCH12] showed that any non-committing encryption scheme is tolerant to leakage on both the secret key sk and encryption randomness r (together), such that leaking L bits on (sk, r) reveals no more than L bits on the underlying encrypted message.

We solve the open problem of allowing simultaneous leakage from sender and encryptor by our *post-challenge auxiliary input model*, which allows hard-to-invert leakage and does not reveals *any bit* on the underlying encrypted message.

5.1 Security Model of Post-Challenge Auxiliary Inputs

We give the new post-challenge auxiliary input model for (probabilistic) public key encryption. As introduced in Section 5.0.3, the basic setting of our new security model is similar to the classic IND-CCA model and the auxiliary input model for public key encryption. Our improvement is to require the adversary $\mathcal{A}$ to submit a set of possible leakages $\mathcal{F}_0$ that may be asked later in the security game, in order to avoid the trivial attacks mentioned in Section 5.0.3. Since $\mathcal{A}$ is a PPT algorithm, we consider that $m := |\mathcal{F}_0|$ is polynomial in the security parameter λ.

During the security game, $\mathcal{A}$ is only allowed to ask for at most q queries $f'_1, \dots f'_q \in \mathcal{F}_0$ to the post-challenge leakage oracle and obtains $f'_1(r'), \dots f'_q(r')$, where r' is the encryption randomness of the challenge ciphertext, but $\mathcal{A}$ cannot recover r' with probability better than ϵ_r. $\mathcal{A}$ can make these choices adaptively after seeing the challenge ciphertext. Hence, the post-challenge leakage query is meaningful. Denote the number of pre-challenge leakage oracle queries as q'.

We are now ready to give the formal definition of the model below. Let $\Pi = (\mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec})$ be a public-key encryption scheme. The security against post-challenge auxiliary inputs and adaptive chosen-ciphertext attacks is defined as the following game pAI-CCA, with respect to the security parameter λ.

1. The adversary $\mathcal{A}$ submits a set of leakage functions $\mathcal{F}_0$ to the challenger $\mathcal{C}$ with $m := |\mathcal{F}_0|$ is polynomial in λ.

2. $\mathcal{C}$ runs $(\mathsf{pk}, \mathsf{sk}) \leftarrow_\$ \mathtt{Gen}(1^\lambda)$ and outputs pk to $\mathcal{A}$.

3. $\mathcal{A}$ may adaptively query the (pre-challenge) leakage oracle:

   - $\mathcal{LO}_s(f_i)$ with f_i. $\mathcal{LO}_s(f_i)$ returns $f_i(\mathsf{sk}, \mathsf{pk})$ to $\mathcal{A}$.

4. $\mathcal{A}$ submits two messages $m_0, m_1 \in \mathcal{M}$ of the same length to $\mathcal{C}$. $\mathcal{C}$ samples $b \leftarrow_\$ \{0,1\}$ and the randomness of encryption $r' \leftarrow_\$ \{0,1\}^*$. It returns $C^* \leftarrow_\$ \mathtt{Enc}(\mathsf{pk}, m_b; r')$ to $\mathcal{A}$.

5. $\mathcal{A}$ may adaptively query the (post-challenge) leakage oracle and the decryption oracle:

- $\mathcal{LO}_r(f_i')$ with $f_i' \in \mathcal{F}_0$. It returns $f_i'(r')$ to $\mathcal{A}$.
- $\mathcal{DEC}(C)$ with $C \neq C^*$. It returns $\mathtt{Dec}(\mathsf{sk}, C)$ to $\mathcal{A}$.

6. $\mathcal{A}$ outputs its guess $b' \in \{0,1\}$. The advantage of $\mathcal{A}$ is $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{pAI-CCA}}(\Pi) = |\Pr[b = b'] - \frac{1}{2}|$.

Note that in the pre-challenge leakage stage, $\mathcal{A}$ may choose $f_i(\mathsf{sk}, \mathsf{pk})$ to encode $\mathtt{Dec}(\mathsf{sk}, \cdot)$ to query the pre-challenge leakage oracle $\mathcal{LO}_s$. Recall that we do not restrict f_i to be in $\mathcal{F}_0$. Therefore to provide an explicit decryption oracle is superfluous.

Furthermore, our model implicitly allows the adversary to obtain some leakage g on intermediate values during the encryption process, in the form of $g(\mathsf{pk}, m_0, f(r^*))$ and $g(\mathsf{pk}, m_1, f(r^*))$, where f is any hard-to-invert function. Since the adversary knows pk, m_0 and m_1, it can compute this kind of leakage for any polynomial time function g given the knowledge of $f(r^*)$.

Denote the set of functions asked in the pre-challenge leakage oracle $\mathcal{LO}_s$ as $\mathcal{F}_s$. We have to define the families $(\mathcal{F}_s, \mathcal{F}_0)$ for the leakage functions asked in the oracles. We can define the family of length-bounded function by restricting the size of the function output as in [DKL09] (Refer to [DKL09] for the definition of such family). In this thesis, we consider the families of one-way function for auxiliary input model. We usually consider $\mathcal{F}_0$ as a family of one-way function $\mathcal{H}_{\mathsf{ow}}$, which is extended from the definition in [DKL09]:

- Let $\mathcal{H}_{\mathsf{ow}}(\epsilon_r)$ be the class of all polynomial-time computable functions $h : \{0,1\}^{|r'|} \to \{0,1\}^*$, such that given $h(r')$ (for a randomly generated r'), no PPT algorithm can find r' with probability greater than ϵ_r[3]. The function $h(r')$ can be viewed as a composition of $q \in \mathbb{N}^+$ functions: $h(r') = (h_1(r'), \ldots, h_q(r'))$. Therefore $\{h_1, \ldots, h_q\} \in \mathcal{H}_{\mathsf{ow}}(\epsilon_r)$.

Also, we consider $\mathcal{F}_s$ as a family of one-way function $\mathcal{H}_{\mathsf{pk-ow}}$, which is extended from the definition in [DKL09]:

- Let $\mathcal{H}_{\mathsf{pk-ow}}(\epsilon_s)$ be the class of all polynomial-time computable functions $h : \{0,1\}^{|\mathsf{sk}|+|\mathsf{pk}|} \to \{0,1\}^*$, such that given $(\mathsf{pk}, h(\mathsf{sk}, \mathsf{pk}))$ (for a randomly generated $(\mathsf{sk}, \mathsf{pk})$), no PPT algorithm can find sk with probability greater than ϵ_s[4]. The function $h(\mathsf{sk}, \mathsf{pk})$ can be

[3]Otherwise, for example, $\mathcal{A}$ can choose an identity mapping f. Then, $\mathcal{A}$ can learn $r' = f(r')$ and test if $C^* = \mathtt{Enc}(\mathsf{pk}, m_0^*; r')$ to determine b and win the game.

[4]Note that we consider the probability of hard-to-invert function given the public key, the public parameters and other related parameters in the security game. Similar to the weak-AI-CPA model in [DKL09], no PPT algorithm will output sk with ϵ_s probability given f_i, pk, as pk leaks some information about sk. Therefore, we also define that no PPT algorithm will output r' with ϵ_r probability given $f_i', C^*, \mathsf{pk}, m_0^*, m_1^*$. We omit these extra input parameters for simplicity in the rest of the thesis.

viewed as a composition of q' functions: $h(\mathsf{sk}, \mathsf{pk}) = (h_1(\mathsf{sk}, \mathsf{pk}), \ldots, h_{q'}(\mathsf{sk}, \mathsf{pk}))$. Therefore $\{h_1, \ldots, h_{q'}\} \in \mathcal{H}_{\mathsf{pk-ow}}(\epsilon_s)$.

Definition 24. We say that Π is pAI-CCA secure with respect to the families $(\mathcal{H}_{\mathsf{pk-ow}}(\epsilon_s), \mathcal{H}_{\mathsf{ow}}(\epsilon_r))$ if the advantage of any PPT adversary $\mathcal{A}$ in the above game is negligible.

We can also define the security for chosen plaintext attack (CPA) similarly. By forbidding the decryption oracle query, we have the security model for pAI-CPA. If we further forbid the leakage of the encryption randomness, we get the original AI-CPA model in [DKL09].

We also define the security model for identity-based encryption similarly. An identity-based encryption scheme Π consists of four PPT algorithms:

- $\mathsf{Setup}(1^\lambda)$: On input the security parameter λ, output a master public key mpk and a master secret key msk. Denote the message space as $\mathcal{M}$ and the identity space as $\mathcal{I}$.
- $\mathsf{Extract}(\mathsf{msk}, \mathsf{ID})$: On input msk and an identity $\mathsf{ID} \in \mathcal{I}$, output the identity-based secret key $\mathsf{sk}_{\mathsf{ID}}$.
- $\mathtt{Enc}(\mathsf{mpk}, \mathsf{ID}, M)$: On input mpk, $\mathsf{ID} \in \mathcal{I}$ and a message $M \in \mathcal{M}$, output a ciphertext C.
- $\mathtt{Dec}(\mathsf{sk}_{\mathsf{ID}}, C)$: On input $\mathsf{sk}_{\mathsf{ID}}$ and C, output the message M or $\perp$ for invalid ciphertext.

We require $\mathtt{Dec}(\mathsf{sk}_{\mathsf{ID}}, \mathtt{Enc}(\mathsf{mpk}, \mathsf{ID}, M)) = M$ for all $M \in \mathcal{M}$, $\mathsf{ID} \in \mathcal{I}$, $(\mathsf{mpk}, \mathsf{msk}) \leftarrow_\$ \mathsf{Setup}(1^\lambda)$ and $\mathsf{sk}_{\mathsf{ID}} \leftarrow_\$ \mathsf{Extract}(\mathsf{msk}, \mathsf{ID})$.

We are now ready to give the formal definition of the model below. Let $\Pi = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ be an identity-based encryption scheme. The security against post-challenge auxiliary inputs and adaptive chosen-identity, chosen-plaintext attacks is defined as the following game pAI-ID-CPA, with respect to the security parameter λ.

1. The adversary $\mathcal{A}$ submits a set of leakage functions $\mathcal{F}_0$ to the challenger $\mathcal{C}$ with $m := |\mathcal{F}_0|$ is polynomial in λ.
2. $\mathcal{C}$ runs $(\mathsf{mpk}, \mathsf{msk}) \leftarrow_\$ \mathsf{Setup}(1^\lambda)$ and outputs mpk to $\mathcal{A}$. $\mathcal{C}$ also samples the randomness of encryption $r' \leftarrow_\$ \{0,1\}^*$.
3. $\mathcal{A}$ may adaptively query the (pre-challenge) leakage oracles:
   - $\mathcal{LO}_s(f_i)$ with f_i. $\mathcal{LO}_s(f_i)$ returns $f_i(\mathsf{msk}, \mathsf{mpk})$ to $\mathcal{A}$.

4. $\mathcal{A}$ submits its challenge identity $\mathsf{ID}^* \in \mathcal{I}$ along with two messages $m_0, m_1 \in \mathcal{M}$ of the same length to $\mathcal{C}$. $\mathcal{C}$ samples $b \leftarrow_{\$} \{0,1\}$. It returns $C^* \leftarrow_{\$} \mathsf{Enc}(\mathsf{mpk}, \mathsf{ID}^*, m_b; r')$ to $\mathcal{A}$.

5. $\mathcal{A}$ may adaptively query the (post-challenge) leakage oracle

   - $\mathcal{LO}_r(f_i')$ with $f_i' \in \mathcal{F}_0$. $\mathcal{LO}_r(f_i')$ returns $f_i'(r')$ to $\mathcal{A}$.
   - $\mathcal{EO}(\mathsf{ID})$ for $\mathsf{ID} \neq \mathsf{ID}^* \in \mathcal{I}$. The extraction oracle returns $\mathsf{sk}_{\mathsf{ID}} \leftarrow_{\$} \mathsf{Extract}(\mathsf{msk}, \mathsf{ID})$.

6. $\mathcal{A}$ outputs its guess $b' \in \{0,1\}$. The advantage of $\mathcal{A}$ is $Adv_{\mathcal{A}}^{\mathrm{pAI-ID-CPA}}(\Pi) = |\Pr[b = b'] - \frac{1}{2}|$.

Note that in the pre-challenge leakage stage, $\mathcal{A}$ may choose $f_i(\cdot, \mathsf{mpk})$ to encode $\mathsf{Extract}(\cdot, \mathsf{ID})$ to query the pre-challenge leakage oracle $\mathcal{LO}_s$. Recall that we do not restrict f_i to be in $\mathcal{F}_0$. Therefore to provide an explicit extraction oracle is superfluous.

Similar to the model for PKE, we define the families $(\mathcal{F}_s, \mathcal{F}_0)$ for the leakage functions asked in the oracles.

Definition 25. We say that Π is pAI-ID-CPA secure with respect to the families $(\mathcal{F}_s, \mathcal{F}_0)$ if the advantage of any PPT adversary $\mathcal{A}$ in the above game is negligible.

Similar to the standard security models for IBE, we can define CCA security if the adversary can ask the decryption oracle for arbitrary ciphertext except the challenge ciphertext. We can also define the selective identity (sID) model, where the adversary has to submit ID^* in step 1 of the security game.

5.2 CPA Secure PKE Construction Against Post-Challenge Auxiliary Inputs

In this section, we give the construction of a public key encryption which is pAI-CPA secure. We show that it can be constructed from an AI-CPA secure encryption (e.g., [DGK+10]) and a *strong extractor with ϵ-hard-to-invert auxiliary inputs leakage*.

5.2.1 Strong Extractor with Hard-to-invert Auxiliary Inputs

We first give the definition of the strong extractor with ϵ-hard-to-invert auxiliary inputs leakage as follows.

Definition 26 ((ϵ, δ)-Strong extractor with auxiliary inputs)**.** Let $\mathsf{Ext} : \{0,1\}^{l_1} \times \{0,1\}^{l_2} \to \{0,1\}^{m'}$. Ext is said to be a (ϵ, δ)-strong extractor with auxiliary inputs, if for every PPT adversary $\mathcal{A}$, and for all $f \in \mathcal{H}_{\mathsf{ow}}(\epsilon)$, we have:

$$|\Pr[\mathcal{A}(r, f(x), \mathsf{Ext}(r, x)) = 1] - \Pr[\mathcal{A}(r, f(x), u) = 1]| < \delta.$$

where $r \in \{0,1\}^{l_1}$, $x \in \{0,1\}^{l_2}$ and $u \in \{0,1\}^{m'}$ are chosen uniformly random.

An interesting property of the above definition is that such a strong extractor itself is 2δ-hard-to-invert. This property is useful when we prove pAI-CCA encryption security.

Lemma 25. *Let $r \in \{0,1\}^{l_1}, x \in \{0,1\}^{l_2}$ be chosen uniformly random. For any $f \in \mathcal{H}_{\mathsf{ow}}(\epsilon)$, given $r, f(x)$ and $\mathsf{Ext}(r, x)$, no PPT adversary can find x with probability $\geq 2\delta$, provided that $\mathsf{Ext}(r, x)$ is a (ϵ, δ)-strong extractor with auxiliary inputs.*

Proof. Suppose on the contrary, there exists an adversary $\mathcal{A}$, given $r, f(x)$ and $\mathsf{Ext}(r, x)$, can recover x with probability $\geq 2\delta$. Then, we show an adversary $\mathcal{B}$ that can distinguish $\mathsf{Ext}(\mathsf{r}, \mathsf{x})$ from a uniformly random u with high probability. Specifically, $\mathcal{B}$ is given $(r, f(x), T)$ where T is either $\mathsf{Ext}(r, x)$ or u with half and half probabilities. $\mathcal{B}$ first calls $\mathcal{A}$ on input $(r, f(x), T)$ where $\mathcal{A}$ will output a result x'. $\mathcal{B}$ will test $\mathsf{Ext}(r, x') = T$ or not. If equal, $\mathcal{B}$ outputs 1, and 0 otherwise.

$$\begin{aligned}\Pr[\mathcal{B} \text{ wins}] &= \frac{1}{2}\Pr[\mathcal{B}\text{wins}|T = \mathsf{Ext}(r, x)] + \frac{1}{2}\Pr[\mathcal{B} \text{ wins}|T = u] \\ &\geq \frac{1}{2} \times 2\delta = \delta,\end{aligned}$$

which shows a contradiction with the fact that Ext is a (ϵ, δ)-strong extractor with auxiliary inputs.

□

Interestingly, we find that a (ϵ, δ)-strong extractor with auxiliary inputs (where $\delta = O(\epsilon^{1/3})$) can be constructed from the modified Goldreich-Levin theorem from [DGK+10]. Denote $\langle r, x\rangle = \sum_{i=1}^{l} r_i x_i$ as the inner product of $x = (x_1, \ldots x_l)$ and $r = (r_1, \ldots, r_l)$.

Theorem 26 ([DGK+10])**.** *Let q be a prime, and let $\bar{H}$ be an arbitrary subset of $GF(q)$. Let $f : \bar{H}^{\bar{n}} \to \{0,1\}^*$ be any (possibly randomized) function. s is chosen randomly from $\bar{H}^{\bar{n}}$, r is chosen randomly from $GF(q)^{\bar{n}}$ and u is chosen randomly from $GF(q)$. We also have*

$y = f(s)$. *If there is a distinguisher* D *that runs in time* t *such that*

$$|\Pr[D(r, y, \langle r, s\rangle) = 1] - \Pr[D(r, y, u)] = 1| = \delta,$$

then there is an inverter $\mathcal{A}$ *that runs in time* $t' = t \cdot \mathsf{poly}(\bar{n}, |\bar{H}|, \frac{1}{\delta})$ *such that* $\Pr[\mathcal{A}(y) = s] \geq \frac{\delta^3}{512\bar{n}q^2}$.

Theorem 27. *Let* λ *be the security parameter. Let* x *be chosen uniformly random from* $\{0,1\}^{l(\lambda)}$ *where* $l(\lambda) = \mathsf{poly}(\lambda)$. *Similarly, we choose* r *uniformly random from* $GF(q)^{l(\lambda)}$ *and* u *uniformly random from* $GF(q)$. *Then, given* $f \in \mathcal{H}_{\mathsf{ow}}(\epsilon)$, *no PPT algorithm* $\mathcal{A}'$ *can distinguish* $(r, f(x), \langle r, x\rangle)$ *from* $(r, f(x), u)$ *with probability* $\epsilon' \geq (512l(\lambda)q^2\epsilon)^{1/3}$.

Proof. Now, we let $\bar{H} = \{0,1\} \subset GF(q)$, $\bar{n} = l(\lambda)$. Suppose there is an algorithm that can distinguish $(r, f(x), \langle r, x\rangle)$ and $(r, f(x), u)$ in time $t = \mathsf{poly}_1(\lambda)$ with probability ϵ'. Then, there exists an inverter $\mathcal{A}$ that runs in time $t \cdot \mathsf{poly}(l(\lambda), 2, \frac{1}{\epsilon}) = \mathsf{poly}'(\lambda)$ such that $\Pr[\mathcal{A}(f(x)) = x] \geq \frac{\epsilon'^3}{512l(\lambda)q^2} \geq \epsilon$ if $\epsilon' \geq (512l(\lambda)q^2\epsilon)^{1/3}$. It contradicts that $f \in \mathcal{H}_{\mathsf{ow}}(\epsilon)$. □

5.3 Construction of pAI-CPA Secure PKE

Let $\Pi' = (\mathtt{Gen}', \mathtt{Enc}', \mathtt{Dec}')$ be an AI-CPA secure encryption (with respect to family $\mathcal{H}_{\mathsf{pk-ow}}(\epsilon_s)$) where the encryption randomness is in $\{0,1\}^{m'}$, $\mathsf{Ext} : \{0,1\}^{l_1} \times \{0,1\}^{l_2} \to \{0,1\}^{m'}$ is a strong extractor with ϵ_r-hard-to-invert auxiliary inputs leakage, then a pAI-CPA secure (with respect to families $(\mathcal{H}_{\mathsf{pk-ow}}(\epsilon_s), \mathcal{H}_{\mathsf{ow}}(\epsilon_r))$) encryption scheme Π can be constructed as follows.

1. $\mathtt{Gen}(1^\lambda)$: It runs $(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\, \mathtt{Gen}'(1^\lambda)$ and chooses r uniformly random from $\{0,1\}^{l_1}$. Then, we set the public key $\mathsf{PK} = (\mathsf{pk}, r)$ and the secret key $\mathsf{SK} = \mathsf{sk}$.

2. $\mathtt{Enc}(\mathsf{PK}, M)$: It picks x uniformly random from $\{0,1\}^{l_2}$. Then, it computes $y = \mathsf{Ext}(r, x)$. The ciphertext is $c = \mathtt{Enc}'(\mathsf{pk}, M; y)$.

3. $\mathtt{Dec}(\mathsf{SK}, C)$: It returns $\mathtt{Dec}'(\mathsf{sk}, C)$.

Theorem 28. *If* Π' *is an AI-CPA secure encryption with respect to family* $\mathcal{H}_{\mathsf{pk-ow}}(\epsilon_s)$ *and* Ext *is a* $(\epsilon_r, \mathsf{negl}(\lambda))$*-strong extractor with auxiliary inputs, then* Π *is pAI-CPA secure with respect to families* $(\mathcal{H}_{\mathsf{pk-ow}}(\epsilon_s), \mathcal{H}_{\mathsf{ow}}(\epsilon_r))$.

Proof. Denote the randomness used in the challenge ciphertext as x^*. Let $\mathbf{Game}_0$ be the pAI-CPA security game with Π scheme. $\mathbf{Game}_1$ is the same as $\mathbf{Game}_0$ except that when

encrypting the challenge ciphertext $c = \mathsf{Enc}'(\mathsf{pk}, m_b; y)$, we replace $y = \mathsf{Ext}(r, x^*)$ with y' which is chosen uniformly at random in $\{0,1\}^{m'}$. The leakage oracle outputs $f_i(x^*)$ for both games.

Let $\mathbf{Adv}_{\mathcal{A}}^{\mathbf{Game}_i}(\Pi)$ be the advantage that the adversary $\mathcal{A}$ wins in $\mathbf{Game}_i$ with Π scheme. Now, we need to show for any PPT adversary $\mathcal{A}$:

$$|\mathbf{Adv}_{\mathcal{A}}^{\mathbf{Game}_0}(\Pi) - \mathbf{Adv}_{\mathcal{A}}^{\mathbf{Game}_1}(\Pi)| \leq \mathsf{negl}(\lambda).$$

Assume that there exists an adversary $\mathcal{A}$ such that $|\mathbf{Adv}_{\mathcal{A}}^{\mathbf{Game}_0}(\Pi) - \mathbf{Adv}_{\mathcal{A}}^{\mathbf{Game}_1}(\Pi)| \geq \epsilon_A$ which is non-negligible.

The simulator $\mathcal{S}$ is given $(r, f_1(x^*), f_2(x^*), \ldots, f_q(x^*), T)$ where T is either $T_0 = \mathsf{Ext}(r, x^*)$ or $T_1 = u$ which is a random number as in Definition 26. Given $f_1(x^*), \ldots, f_q(x^*)$, no PPT adversary can recover x^* with probability greater than ϵ_r by the definition of $\mathcal{H}_{\mathsf{ow}}(\epsilon_r)$. Then, the simulator generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow_{\$} \mathsf{Gen}'(1^\lambda)$. It sets $\mathsf{SK} = \mathsf{sk}$ and gives the adversary $\mathsf{PK} = (\mathsf{pk}, r)$. The simulator can answer pre-challenge leakage oracle as it has PK and SK. The adversary submits two message m_0 and m_1 to the simulator where the simulator flips a coin b. It encrypts the challenge ciphertext $C^* = \mathsf{Enc}(\mathsf{pk}, m_b; T)$ and gives it to $\mathcal{A}$. $\mathcal{A}$ can ask $f_i(x)$ as the post-challenge leakage queries. $\mathcal{A}$ outputs its guess bit b' to the simulator. If $b = b'$, the simulator outputs 1; otherwise, it outputs 0.

Since the difference of advantage of $\mathcal{A}$ between $\mathbf{Game}_0$ and $\mathbf{Game}_1$ is ϵ_A, then

$$\begin{aligned}
\mathbf{Adv}_{\mathcal{S}} &= \left|\frac{1}{2}\Pr[\mathcal{S} \text{ outputs } 1|T_1] + \frac{1}{2}\Pr[\mathcal{S} \text{ outputs } 0|T_0] - \frac{1}{2}\right| \\
&= \left|\frac{1}{2}\Pr[\mathcal{S} \text{ outputs } 1|T_1] + \frac{1}{2}(1 - \Pr[\mathcal{S} \text{ outputs } 1|T_0]) - \frac{1}{2}\right| \\
&= \frac{1}{2}(|\Pr[b = b'|T_1] - \Pr[b = b'|T_0]|) \geq \frac{\epsilon_A}{2}.
\end{aligned}$$

which is non-negligible if ϵ_A is non-negligible. It contradicts the definition of strong extractor in Definition 26. Therefore, no PPT adversary can distinguish $\mathbf{Game}_0$ from $\mathbf{Game}_1$ with non-negligible probability.

Next, we want to show that

$$\mathbf{Adv}_{\mathcal{A}}^{\mathbf{Game}_1}(\Pi) = \mathsf{negl}(\lambda).$$

Note that the challenge ciphertext now is $C = \mathsf{Enc}'(\mathsf{pk}, M; y')$ where y' is chosen uniformly at random in $\{0,1\}^{m'}$. Therefore the output of the leakage oracle $f_i(x^*)$ will not reveal any

information related to C. Then $\mathbf{Game}_1$ is the same as the AI-CPA game with Π'. As Π is based on Π' which is AI-CPA secure, we have that $\mathbf{Adv}_{\mathcal{A}}^{\mathbf{Game}_1}(\Pi)$ is negligible. □

5.3.1 Extension to IBE

We can use the same technique to construct pAI-ID-CPA secure IBE. Let $\Sigma' = (\mathsf{Setup}', \mathsf{Extract}', \mathtt{Enc}', \mathtt{Dec}')$ be an AI-ID-CPA secure IBE (e.g. [YCZY12a]) where the encryption randomness is in $\{0,1\}^{m'}$, $\mathsf{Ext} : \{0,1\}^{l_1} \times \{0,1\}^{l_2} \to \{0,1\}^{m'}$ is a $(\epsilon_r, \mathsf{negl}(\lambda))$-strong extractor with auxiliary inputs, then construct a pAI-ID-CPA secure IBE scheme Σ as follows.

1. $\mathsf{Setup}(1^\lambda)$: It runs $(\mathsf{mpk}, \mathsf{msk}) \leftarrow_\$ \mathsf{Setup}'(1^\lambda)$ and chooses r uniformly random from $\{0,1\}^{l_1}$. Then, we set the master public key $\mathsf{MPK} = (\mathsf{mpk}, r)$ and the master secret key $\mathsf{MSK} = \mathsf{msk}$.

2. $\mathsf{Extract}(\mathsf{MSK}, \mathsf{ID})$: It returns $\mathsf{sk}_{\mathsf{ID}} \leftarrow_\$ \mathsf{Extract}(\mathsf{MSK}, \mathsf{ID})$.

3. $\mathtt{Enc}(\mathsf{MPK}, \mathsf{ID}, M)$: It chooses x uniformly random from $\{0,1\}^{l_2}$. Then, it computes $y = \mathsf{Ext}(r, x)$. The ciphertext is $C = \mathtt{Enc}'(\mathsf{mpk}, \mathsf{ID}, M; y)$.

4. $\mathtt{Dec}(\mathsf{sk}_{\mathsf{ID}}, C)$: It returns $\mathtt{Dec}'(\mathsf{sk}_{\mathsf{ID}}, C)$.

Theorem 29. *If Σ' is an AI-ID-CPA secure IBE with respect to family $\mathcal{H}_{\mathsf{pk-ow}}(\epsilon_s)$ and Ext is a $(\epsilon_r, \mathsf{negl}(\lambda))$-strong extractor with auxiliary inputs, then Σ is pAI-ID-CPA secure with respect to families $(\mathcal{H}_{\mathsf{pk-ow}}(\epsilon_s), \mathcal{H}_{\mathsf{ow}}(\epsilon_r))$.*

The proof is similar to the proof of Theorem 28 and hence is omitted.

Corollary 30. *Instantiating with the strong extractor construction in Section 5.2.1 and the identity-based encryption scheme in [YCZY12a], the identity-based encryption construction Σ' is pAI-ID-CPA secure.*

5.4 CCA Public Key Encryption from CPA Identity-Based Encryption

In this section, we show that auxiliary-inputs (selective-ID) CPA secure IBE and strong one-time signatures imply post-challenge auxiliary-inputs CCA secure PKE. Canetti *et al.* [CHK04] showed that a CCA secure encryption can be constructed from a (selective-ID) CPA

secure IBE and a strong one-time signatures. We would like to show that this transformation can also be applied to the auxiliary input model after some modifications. As in [CHK04], we use the strong one-time signature to prevent the PKE adversaries asking for decrypting ciphertexts of ID^* in the post stage as the IBE adversaries are not allowed to ask $\mathsf{Extract}(\mathsf{ID}^*)$. However, we cannot apply the technique in [CHK04] directly.

5.4.1 Intuition

Let $(\mathtt{Gen}_s, \mathtt{Sign}, \mathtt{Verify})$ be a strong one-time signature scheme. Let $(\mathsf{Setup}', \mathsf{Extract}', \mathtt{Enc}', \mathtt{Dec}')$ be an auxiliary-inputs CPA secure IBE scheme (refer to the definition in Section **??**, by dropping the post-challenge query). The construction directly following Canetti *et al.*'s transformation [CHK04] is as follows.

1. $\mathtt{Gen}(1^\lambda)$: Run $(\mathsf{mpk}, \mathsf{msk}) \leftarrow_\$ \mathsf{Setup}'(1^\lambda)$. Set the public key $\mathsf{pk} = \mathsf{mpk}$ and the secret key $\mathsf{sk} = \mathsf{msk}$.

2. $\mathtt{Enc}(\mathsf{pk}, M)$: Run $(\mathsf{vk}, \mathsf{sk}_s) \leftarrow_\$ \mathtt{Gen}_s(1^\lambda)$. Calculate $c \leftarrow_\$ \mathtt{Enc}'(\mathsf{pk}, \mathsf{vk}, M)$ and $\sigma \leftarrow_\$ \mathtt{Sign}(\mathsf{sk}_s, c)$. Then, the ciphertext is $C = (c, \sigma, \mathsf{vk})$.

3. $\mathtt{Dec}(\mathsf{sk}, C)$: First, test $\mathtt{Verify}(\mathsf{vk}, c, \sigma) \stackrel{?}{=} 1$. If it is "1", compute $\mathsf{sk}_{\mathsf{vk}} = \mathsf{Extract}'(\mathsf{sk}, \mathsf{vk})$ and return $\mathtt{Dec}'(\mathsf{sk}_{\mathsf{vk}}, c)$. Otherwise, return $\perp$.

5.4.2 Problems in the Post-Challenge Auxiliary Input Model

At first glance it seems that Canetti *et al.*'s transformation [CHK04] also works in our pAI-CCA model for PKE, if we simply change the underlying IBE to be secure in the corresponding post-challenge auxiliary input model. However, we find that this is not true. The main challenge of pAI-CCA secure PKE is how to handle the leakage of the randomness used in the challenge ciphertext. It includes the randomness used in $\mathtt{Gen}_s$, $\mathtt{Sign}$ and $\mathtt{Enc}'$, denoted as r_{sig_1}, r_{sig_2} and r_{enc} respectively. Specifically, we have $(\mathsf{vk}, \mathsf{sk}_s) \leftarrow_\$ \mathtt{Gen}_s(1^\lambda; r_{\mathsf{sig}_1})$, $\sigma \leftarrow_\$ \mathtt{Sign}(\mathsf{sk}_s, c; r_{\mathsf{sig}_2})$ and $c \leftarrow_\$ \mathtt{Enc}'(\mathsf{mpk}, \mathsf{vk}, m_b; r_{\mathsf{enc}})$.

Let $\mathcal{A}$ be a pAI-CCA adversary of the PKE. Let f be (one of) the post-challenge leakage function submitted by $\mathcal{A}$ before seeing the public key. Then, after receiving the challenge ciphertext $C^* = (c^*, \sigma^*, \mathsf{vk}^*)$, $\mathcal{A}$ can ask the leakage $f(r')$ where $r' = (r_{\mathsf{enc}}, r_{\mathsf{sig}_1}, r_{\mathsf{sig}_2})$ is the randomness used to produce C^*. To some extreme, $\mathcal{A}$ may ask:

- $f_1(r') = r_{\mathsf{enc}}$, such that f_1 is still hard-to-invert upon r'. In this case, $\mathcal{A}$ can test $c^* \stackrel{?}{=} \mathtt{Enc}'(\mathsf{mpk}, \mathsf{vk}, m_0; r_{\mathsf{enc}})$ to win the pAI-CCA game; or
- $f_2(r') = (r_{\mathsf{sig}_1}, r_{\mathsf{sig}_2})$, such that f_2 is still hard-to-invert upon r'. In this case, given r_{sig_1}, $\mathcal{A}$ can generate $(\mathsf{vk}, \mathsf{sk}_s) = \mathtt{Gen}_s(1^\lambda; r_{\mathsf{sig}_1})$ which causes $\Pr[\mathsf{Forge}]$ defined in [CHK04] to be non-negligible ("Forge" is the event that $\mathcal{A}$ wins the game by outputting a forged strong one-time signature).

Therefore, leaking part of the randomness in r' will make the proof of [CHK04] fail in our model.

5.4.3 Our Solution

To get rid of this problem, we set both $r_{\mathsf{sig}_1}, r_{\mathsf{sig}_2}$ and r_{enc} are generated from the same source of randomness $x \in \{0,1\}^{l_2}$. Suppose $r_{\mathsf{sig}_1}||r_{\mathsf{sig}_2}$ and r_{enc} are bit-strings of length n'. Suppose $\mathsf{Ext} : \{0,1\}^{l_1} \times \{0,1\}^{l_2} \to \{0,1\}^{n'}$ is a strong extractor with auxiliary inputs; r_1 and r_2 are independent and uniformly chosen from $\{0,1\}^{l_1}$ which are also included in the public key pk. Then the *randomness* used in the IBE and the one-time signature can be calculated by $r_{\mathsf{enc}} = \mathsf{Ext}_1(r_1, x)$ and $(r_{\mathsf{sig}_1}||r_{\mathsf{sig}_2}) = \mathsf{Ext}_2(r_2, x)$ respectively. In the security proof, the pAI-CCA adversary $\mathcal{A}$ can ask for the leakage of $f(x)$, where f is any *hard-to-invert* function.

The main part of the security proof is to use the pAI-CCA adversary $\mathcal{A}$ to break the AI-ID-CPA security of the underlying IBE scheme Π'. The simulator of the pAI-CCA game has to simulate the post-challenge leakage oracle without knowing the encryption randomness x of the challenge ciphertext, which was produced by the challenger of Π'. We solve this problem by proving that it is indistinguishable by replacing $r^*_{enc} = \mathsf{Ext}_1(r_1, x^*)$ and $r^*_{sig_1}||r^*_{sig_2} = \mathsf{Ext}_2(r_2, x^*)$ with random numbers. Therefore, the post-challenge leakages on x^* will be independent with r^*_{enc} and $r^*_{sig_1}||r^*_{sig_2}$ which are used to produce the real challenge ciphertext. Then, the simulator can randomly choose x^* and simulate the post-challenge oracles by it own. However, when we show to replace $r^*_{sig_1}||r^*_{sig_2}$ with a random number, the simulator needs to compute $r_{enc^*} = \mathsf{Ext}_1(r_1, x^*)$. One way to solve it is to include $\mathsf{Ext}_1(r_1, x^*)$ as a post-challenge leakage query in the pAI-CCA game. As we will see later (by Lemma 25), including $\mathsf{Ext}_1(r_1, x^*)$ in leakage queries is still $\mathsf{negl}(\lambda)$-hard-to-invert.

Following [CHK04], the transformation also works for the weaker selective identity (sID) model. As a result, we only need a AI-sID-CPA secure IBE. To sum up, we need three primitives to construct a pAI-CCA secure PKE: *strong extractor with auxiliary inputs*, *strong one-time signatures* and *AI-sID-CPA secure IBE*.

5.4.4 Post-Challenge Auxiliary Inputs CCA secure PKE

We are now ready to describe our post-challenge auxiliary inputs CCA secure PKE. Denote a AI-sID-CPA secure IBE scheme $\Pi' = (\mathsf{Setup}', \mathsf{Extract}', \mathtt{Enc}', \mathtt{Dec}')$, a strong one-time signature scheme $\Pi_s = (\mathtt{Gen}_s, \mathtt{Sign}, \mathtt{Verify})$ and a strong extractor with ϵ_r-hard-to-invert auxiliary input $\mathsf{Ext} : \{0,1\}^{l_1} \times \{0,1\}^{l_2} \to \{0,1\}^{n'}$, where the size of r_{enc} and $r_{\mathsf{sig}_1}||r_{\mathsf{sig}_2}$ are both $\{0,1\}^{n'}$; and the verification key space of Π_s is the same as the identity space of Π'. We construct a PKE scheme $\Pi = (\mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec})$ as follows.

1. $\mathtt{Gen}(1^\lambda)$: Run $(\mathsf{mpk}, \mathsf{msk}) \leftarrow_\$ \mathsf{Setup}'(1^\lambda)$. Choose r_1, r_2 uniformly random from $\{0,1\}^{l_1}$. Set the public key $\mathsf{pk} = (\mathsf{mpk}, r_1, r_2)$ and the secret key $\mathsf{sk} = \mathsf{msk}$.

2. $\mathtt{Enc}(\mathsf{pk}, m)$: Randomly sample $x \in \{0,1\}^{l_2}$, calculate $r_{\mathsf{enc}} = \mathsf{Ext}_1(r_1, x)$ and $r_{\mathsf{sig}_1}||r_{\mathsf{sig}_2} = \mathsf{Ext}_2(r_2, x)$. Run $(\mathsf{vk}, \mathsf{sk}_s) = \mathtt{Gen}_s(1^\lambda; r_{\mathsf{sig}_1})$. Let $c = \mathtt{Enc}'(\mathsf{pk}, \mathsf{vk}, m; r_{\mathsf{enc}})$; $\sigma = \mathtt{Sign}(\mathsf{sk}_s, c; r_{\mathsf{sig}_2})$. Then, the ciphertext is $C = (c, \sigma, \mathsf{vk})$.

3. $\mathtt{Dec}(\mathsf{sk}, C)$: First, test $\mathtt{Verify}(\mathsf{vk}, c, \sigma) \stackrel{?}{=} 1$. If it is "1", compute $\mathsf{sk}_{\mathsf{vk}} = \mathsf{Extract}(\mathsf{sk}, \mathsf{vk})$ and return $\mathtt{Dec}'(\mathsf{sk}_{\mathsf{vk}}, c)$. Otherwise, return $\perp$.

Theorem 31. *Assuming that Π' is a AI-sID-CPA secure IBE scheme with respect to family $\mathcal{H}_{\mathsf{pk-ow}}(\epsilon_s)$, Π_s is a strong one-time signature, Ext_1 is a $(\epsilon_r, \mathsf{negl}_1)$-strong extractor with auxiliary inputs and Ext_2 is a $(2\mathsf{negl}_1, \mathsf{negl}_2)$-strong extractor with auxiliary inputs, then there exists a PKE scheme Π which is pAI-CCA secure with respect to families $(\mathcal{H}_{\mathsf{pk-ow}}(\epsilon_s), \mathcal{H}_{\mathsf{ow}}(\epsilon_r))$, where $\mathsf{negl}_1, \mathsf{negl}_2$ are some negligible functions.*

Proof. We prove the security by a number of security games. Let $\mathbf{Game}_0$ be the original pAI-CCA game for the PKE scheme Π. Specifically for the challenge ciphertext, the simulator picks a random number x^* to compute $r^*_{\mathsf{enc}} = \mathsf{Ext}_1(r_1, x^*)$ and $r^*_{\mathsf{sig}_1}||r^*_{\mathsf{sig}_2} = \mathsf{Ext}_2(r_2, x^*)$. Let $\mathbf{Game}_1$ be the same as $\mathbf{Game}_0$, except that $r^*_{\mathsf{sig}_1}||r^*_{\mathsf{sig}_2}$ is randomly chosen from $\{0,1\}^{n'}$. Let $\mathbf{Game}_2$ be the same as $\mathbf{Game}_1$, except that r^*_{enc} is randomly chosen from $\{0,1\}^{n'}$.

Lemma 32. *For any PPT adversary $\mathcal{A}$, $\boldsymbol{Game}_0$ is indistinguishable from $\boldsymbol{Game}_1$ if Ext_1 is a $(\epsilon_r, \mathsf{negl}_1)$-strong extractor and with auxiliary inputs and Ext_2 is a $(2\mathsf{negl}_1, \mathsf{negl}_2)$-strong extractor with auxiliary inputs.*

Lemma 33. *For any PPT adversary $\mathcal{A}$, $\boldsymbol{Game}_1$ is indistinguishable from $\boldsymbol{Game}_2$ if Ext_1 is a $(\epsilon_r, \mathsf{negl}_1)$-strong extractor with auxiliary inputs.*

Lemma 34. *For any PPT adversary $\mathcal{A}$, the advantage in $\boldsymbol{Game}_2$ is negligible if Π' is a AI-sID-CPA secure IBE scheme with respect to family $\mathcal{H}_{\mathsf{pk-ow}}(\epsilon_s)$ and Π_s is a strong one-time signature.*

Using the above three lemmas, we have proved the theorem. □

5.4.5 Proofs of Lemmas

Proof of Lemma 32. Let $\mathbf{Adv}_{\mathcal{A}}^{\mathbf{Game}_i}(\Pi)$ be the advantage that the adversary $\mathcal{A}$ wins in $\mathbf{Game}_i$ with Π scheme. Now, we need to show for any PPT adversary $\mathcal{A}$:

$$|\mathbf{Adv}_{\mathcal{A}}^{\mathbf{Game}_0}(\Pi) - \mathbf{Adv}_{\mathcal{A}}^{\mathbf{Game}_1}(\Pi)| \leq \mathsf{negl}(\lambda).$$

Assume that there exists an adversary $\mathcal{A}$ such that $|\mathbf{Adv}_{\mathcal{A}}^{\mathbf{Game}_0}(\Pi) - \mathbf{Adv}_{\mathcal{A}}^{\mathbf{Game}_1}(\Pi)| \geq \epsilon_A$ which is non-negligible.

The simulator $\mathcal{S}$ picks a random $r_1, r_2 \in \{0,1\}^{l_1}$. $\mathcal{S}$ is given $(r_2, f_1(x^*), \ldots, f_q(x^*), f_{q+1}(x^*), T)$ where $f_1, \ldots, f_q \in \mathcal{F}_0$, $f_{q+1}(x^*) = \mathsf{Ext}_1(r_1, x^*)$, and T is either $T_0 = \mathsf{Ext}_2(r_2, x^*)$ or $T_1 = u$ (a random number as in Definition 26). Given $f_1(x^*), \ldots, f_q(x^*)$, no PPT adversary can recover x^* with probability greater than ϵ_r by the definition of $\mathcal{H}_{\mathsf{ow}}(\epsilon_r)$ (We will later show that including $f_{q+1}(x^*) = \mathsf{Ext}_1(r_1, x^*)$ is also $2\mathsf{negl}_1(\lambda)$-hard-to-invert).

Then, the simulator generates $(\mathsf{mpk}, \mathsf{msk}) \leftarrow_{\$} \mathsf{Setup}'(1^\lambda)$. It sets $\mathsf{sk} = \mathsf{msk}$ and gives the adversary $\mathsf{pk} = (\mathsf{mpk}, r_1, r_2)$. $\mathcal{S}$ can answer pre-challenge leakage oracle as it has pk and sk. The adversary submits two messages m_0 and m_1 to $\mathcal{S}$ where the simulator flips a coin b. It sets $r_{\mathsf{sig}_1} || r_{\mathsf{sig}_2} = T$, runs $(\mathsf{vk}, \mathsf{sk}_s) \leftarrow_{\$} \mathsf{Gen}_s(1^\lambda; r_{\mathsf{sig}_1})$, $c = \mathsf{Enc}'(\mathsf{pk}, \mathsf{vk}, m_b; f_{q+1}(x^*))$ and $\sigma = \mathtt{Sign}(\mathsf{sk}_s, c; r_{\mathsf{sig}_2})$. It returns the challenge ciphertext $C^* = (c, \sigma, \mathsf{vk})$ to $\mathcal{A}$. $\mathcal{A}$ can ask $f_i(x^*)$ as the post-challenge leakage queries. $\mathcal{A}$ outputs its guess bit b' to the simulator. If $b = b'$, the simulator outputs 1; otherwise, it outputs 0.

Since the difference of advantage of $\mathcal{A}$ between $\mathbf{Game}_0$ and $\mathbf{Game}_1$ is ϵ_A, then

$$\begin{aligned}\mathbf{Adv}_{\mathcal{S}} &= \left|\frac{1}{2}\Pr[\mathcal{S} \text{ outputs } 1|T_1] + \frac{1}{2}\Pr[\mathcal{S} \text{ outputs } 0|T_0] - \frac{1}{2}\right| \\ &= \frac{1}{2}(|\Pr[b = b'|T_1] - \Pr[b = b'|T_0]|) \geq \frac{\epsilon_A}{2}.\end{aligned}$$

which is non-negligible if ϵ_A is non-negligible. It contradicts the fact that Ext_2 is a $(2\mathsf{negl}_1, 2\mathsf{negl}_2)$-strong extractor with auxiliary inputs. Therefore, no PPT adversary can distinguish $\mathbf{Game}_0$ from $\mathbf{Game}_1$ with non-negligible probability.

Finally, we need to show that including $\mathsf{Ext}_1(r_1, \cdot)$ is also $2\mathsf{negl}_1(\lambda)$-hard-to-invert, provided that Ext_1 itself is a $(\epsilon_r, 2\mathsf{negl}_1)$-strong extractor with auxiliary inputs. This follows directly from Lemma 25 if we set $f = (f_1(x^*), \ldots, f_q(x^*)) \in \mathcal{H}_{ow}(\epsilon_r)$. □

Proof of Lemma 33. The post-challenge query functions $(f_1, \ldots, f_q) \in \mathcal{F}_0$ are ϵ_r-hard-to-invert by definition. Fix any auxiliary-input function $f_1, \ldots, f_q$, $\langle r_1, f_1(x^*), \ldots, f_q(x^*), \mathsf{Ext}_1(r_1, x^*)\rangle$ is indistinguishable with $\langle r_1, f_1(x^*), \ldots, f_q(x^*), u\rangle$ where u is randomly chosen from $\{0,1\}^{n'}$, by the definition of strong extractor. Hence $\mathbf{Game}_1$ is indistinguishable from $\mathbf{Game}_2$. The reduction is similar to the previous proof. □

Proof of Lemma 34. Let $\mathcal{A}$ be an adversary to Π on $\mathbf{Game}_2$ and we construct an AI-sID-CPA adversary $\mathcal{A}'$ to Π' that runs $\mathcal{A}$ as a subroutine. Initially, $\mathcal{A}$ submits a set of leakage functions $\mathcal{F}_0$ that he would like to ask in the $\mathbf{Game}_2$ to $\mathcal{A}'$. $\mathcal{A}'$ picks $r_{\mathsf{sig}_1}||r_{\mathsf{sig}_2}$ uniformly random from $\{0,1\}^{n'}$ and computes $(\mathsf{vk}^*, \mathsf{sk}_s^*) = \mathtt{Gen}_s(1^\lambda; r_{\mathsf{sig}_1})$. $\mathcal{A}'$ submits the challenge identity vk^* to the AI-sID-CPA challenger $\mathcal{C}$, and $\mathcal{C}$ returns mpk to $\mathcal{A}'$. Then $\mathcal{A}'$ picks r_1 and r_2 which are independent and uniformly chosen from $\{0,1\}^{l_1}$. $\mathcal{A}'$ gives $\mathsf{pk} = (\mathsf{mpk}, r_1, r_2)$ to $\mathcal{A}$.

In the pre-challenge query phase, $\mathcal{A}$ can adaptively query $f_i(\mathsf{pk}, \mathsf{msk})$. $\mathcal{A}'$ records and forwards all the queries to $\mathcal{C}$; and uses the output by $\mathcal{C}$ to answer $\mathcal{A}$.

In the challenge phase, $\mathcal{A}$ submits m_0, m_1 to $\mathcal{A}'$, and $\mathcal{A}'$ forwards m_0, m_1 as the challenge message to $\mathcal{C}$. $\mathcal{C}$ returns $c^* = \mathtt{Enc}'(\mathsf{mpk}, \mathsf{vk}^*, m_b; r_{\mathsf{enc}})$ to $\mathcal{A}'$ for some random bit b and randomness r_{enc}. Then $\mathcal{A}'$ computes $\sigma^* = \mathtt{Sign}(\mathsf{sk}_s^*, c^*; r_{\mathsf{sig}_2})$. $\mathcal{A}'$ sends $C^* = (c^*, \sigma^*, \mathsf{vk}^*)$ to $\mathcal{A}$ as its challenge ciphertext. $\mathcal{A}'$ picks a random $x^* \in \{0,1\}^{l_2}$.

In the post-challenge query phase, $\mathcal{A}'$ can answer the adaptive query f_i' on the randomness x^* asked by $\mathcal{A}$. $\mathcal{A}$ may also adaptively query $\mathcal{DEC}(c, \sigma, \mathsf{vk})$. $\mathcal{A}'$ returns $\perp$ if $\mathtt{Verify}(\mathsf{vk}, c, \sigma) \neq 1$. Otherwise, there are two cases. If $\mathsf{vk} = \mathsf{vk}^*$, it means $(c, \sigma) \neq (c^*, \sigma^*)$. However, it implies that $\mathcal{A}$ forges the one-time signature. This happens with only a negligible probability. Else, $\mathsf{vk} \neq \mathsf{vk}^*$, $\mathcal{A}'$ asks the extraction oracle $\mathcal{EO}(\mathsf{vk})$ to $\mathcal{C}$ and uses $\mathsf{sk}_{\mathsf{vk}}$ to decrypt c.

Finally $\mathcal{A}$ outputs its guess b' and $\mathcal{A}'$ forwards it to $\mathcal{C}$ as its guess bit. Therefore, if $\mathcal{A}$ wins the $\mathbf{Game}_2$ with a non-negligible probability, then $\mathcal{A}'$ will win the AI-sID-CPA game also with a non-negligible probability, which contradicts that Π' is AI-sID-CPA secure.

To show that the probability that $\mathcal{A}$ asks for the decryption of a valid ciphertext with identity vk^* is negligible, let $\mathcal{C}'$ be the challenger of the strong one-time signature scheme. We construct an algorithm $\mathcal{B}$ to break the strong one-time signature scheme by running $\mathcal{A}$ as a subroutine. Initially, $\mathcal{A}$ submits its post-challenge leakage class $\mathcal{F}_0$ to $\mathcal{B}$. $\mathcal{C}'$ gives vk^* to $\mathcal{B}$.

$\mathcal{B}$ runs $(\mathsf{mpk}, \mathsf{msk}) \leftarrow_{\$} \mathsf{Setup}'(1^\lambda)$ and picks r_1 and r_2 which are independent and uniformly chosen from $\{0,1\}^{l_1}$. $\mathcal{B}$ returns $\mathsf{pk} = (\mathsf{mpk}, r_1, r_2)$ to $\mathcal{A}$.

In the pre-challenge query phase, $\mathcal{A}$ can adaptively query $f_i(\mathsf{pk}, \mathsf{msk})$ and $\mathcal{B}$ can answer them by itself.

In the challenge phase, $\mathcal{A}$ submits m_0, m_1 to $\mathcal{B}$. $\mathcal{B}$ picks r_{enc} uniformly random from $\{0,1\}^{n'}$. $\mathcal{B}$ picks a random bit b and calculates $c^* = \mathtt{Enc}'(\mathsf{mpk}, \mathsf{vk}^*, m_b; r_{\mathsf{enc}})$. Then $\mathcal{B}$ asks $\mathcal{C}'$ to sign on c^* and obtains the signature σ^*. $\mathcal{B}$ gives the challenge ciphertext $C^* = (c^*, \sigma^*, \mathsf{vk}^*)$ to $\mathcal{A}$. $\mathcal{B}$ picks a random $x^* \in \{0,1\}^{l_2}$.

In the post query phase, $\mathcal{A}$ can adaptively ask the post-challenge leakage $f'_i \in \mathcal{F}_0$ to $\mathcal{B}$ and $\mathcal{B}$ can answer it with x^*. $\mathcal{A}$ may also ask for the decryption oracle. Decryption of ciphertext involving $\mathsf{vk} \neq \mathsf{vk}^*$ can be answered by using msk. However, if $\mathcal{A}$ asks for the decryption of a valid ciphertext $(c, \sigma, \mathsf{vk}^*)$ that is not identical to $(c^*, \sigma^*, \mathsf{vk}^*)$, $\mathcal{B}$ returns (c, σ) to $\mathcal{C}'$. Therefore, the probability that $\mathcal{A}$ can output a forged signature is negligible provided that Π_s is a strong one-time signature, which completes the proof. $\square$

Bibliography

[3DE] Nist special publication 800-67 revision 1.

[AC] N. Alon and M. Capalbo. Smaller explicit superconcentrators. In *SODA 2003*.

[ACGS88] W. Alexi, B. Chor, O. Goldreich, and C. Schnorr. RSA and Rabin functions: Certain parts are as hard as the whole. *SIAM Journal on Computing*, 17(2):194–209, April 1988.

[ADN+10] J. Alwen, Y. Dodis, M. Naor, G. Segev, S. Walfish, and D. Wichs. Public-key encryption in the bounded-retrieval model. In *EUROCRYPT*, 2010.

[AES] Fips publication 197.

[AGS03] Adi Akavia, Shafi Goldwasser, and Shmuel Safra. Proving hard-core predicates using list decoding. In *44th Annual Symposium on Foundations of Computer Science*, pages 146–159. IEEE Computer Society Press, October 2003.

[AGV09] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In *TCC*, 2009.

[AK12] George Argyros and Aggelos Kiayias. I forgot your password: randomness attacks against php applications. In *Proceedings of the 21st USENIX conference on Security symposium*, Security'12, pages 6–6, Berkeley, CA, USA, 2012. USENIX Association.

[BBN+09] Mihir Bellare, Zvika Brakerski, Moni Naor, Thomas Ristenpart, Gil Segev, Hovav Shacham, and Scott Yilek. Hedged public-key encryption: How to protect against bad randomness. In Mitsuru Matsui, editor, *ASIACRYPT*, volume 5912 of *LNCS*, pages 232–249. Springer, 2009.

[BCH12] Nir Bitansky, Ran Canetti, and Shai Halevi. Leakage-tolerant interactive protocols. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 266–284. Springer, 2012.

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

[BG85] Manuel Blum and Shafi Goldwasser. An efficient probabilistic public-key encryption scheme which hides all partial information. In G. R. Blakley and David Chaum, editors, *Advances in Cryptology – CRYPTO'84*, volume 196 of *Lecture Notes in Computer Science*, pages 289–302. Springer, August 1985.

[BGK06] J. Bourgain, A. A. Glibichuk, and S. V. Konyagin. Estimates for the number of sums and products and for exponential sums in fields of prime order. *J. London Math Soc*, 2006.

[BKKV10] Z. Brakerski, Y. T. Kalai, J. Katz, and V. Vaikuntanathan. Overcoming the hole in the bucket: Public-key cryptography resilient to continual memory leakage. In *FOCS*, 2010.

[Ble98] D. Bleichenbacher. Chosen ciphertext attacks against protocols based on the rsa encryption standard pkcs#1. In *CRYPTO*, 1998.

[BR93a] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *CCS*, 1993.

[BR93b] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93: 1st Conference on Computer and Communications Security*, pages 62–73. ACM Press, November 1993.

[BS97] Eli Biham and Adi Shamir. Differential fault analysis of secret key cryptosystems. In *CRYPTO*, 1997.

[CDRW10] S. S.M. Chow, Y. Dodis, Y. Rouselakis, and B. Waters. Priactical leakage-resilient identity-based encryption from simple assumption. In *CCS*, 2010.

[CHK04] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–222. Springer, 2004.

[CMS99] Christian Cachin, Silvio Micali, and Markus Stadler. Computationally private information retrieval with polylogarithmic communication. In Jacques Stern, editor, *Advances in Cryptology – EUROCRYPT'99*, volume 1592 of *Lecture Notes in Computer Science*, pages 402–414. Springer, May 1999.

[Coc01] Clifford Cocks. An identity based encryption scheme based on quadratic residues. In *the 8th IMA International Conference on Cryptography and Coding*, 2001.

[DDV10] Francesco Davì, Stefan Dziembowski, and Daniele Venturi. Leakage-resilient storage. In Juan A. Garay and Roberto De Prisco, editors, *SCN*, volume 6280 of *Lecture Notes in Computer Science*, pages 121–137. Springer, 2010.

[DES] Fips publication 46.

[DGK+10] Y. Dodis, S. Goldwasser, Y. T. Kalai, C. Peikert, and V. Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In *TCC*, 2010.

[DHLAW10] Y. Dodis, K. Haralambiev, A. Lo'pez-Alt, and D. Wichs. Cryptography against continuous memory attacks. In *FOCS*, 2010.

[DKL09] Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett. On cryptography with auxiliary input. In Michael Mitzenmacher, editor, *STOC 2009*, pages 621–630. ACM, 2009.

[DKW] Stefan Dziembowski, Tomasz Kazana, and Daniel Wichs. Key-evolution schemes resilient to space-bounded leakage. In *CRYPTO 2011*.

[DKW11] Stefan Dziembowski, Tomasz Kazana, and Daniel Wichs. One-time computable self-erasing functions. In Yuval Ishai, editor, *TCC*, volume 6597 of *Lecture Notes in Computer Science*, pages 125–143. Springer, 2011.

[DNW] Cynthia Dwork, Moni Naor, and Hoeteck Wee. Pebbling and proofs of work. In *CRYPTO 2005*.

[DP08] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *FOCS*, 2008.

[ElG85] Taher ElGamal. A public-key cryptosystem and a signature scheme based on discrete logarithms. In *IEEE Transcations on Information Theory*, 1985.

[FHN+12] Sebastian Faust, Carmit Hazay, Jesper Buus Nielsen, Peter Sebastian Nordholt, and Angela Zottarel. Signature schemes secure against hard-to-invert leakage. Cryptology ePrint Archive, Report 2012/045, 2012. To appear in Asiacrypt 2012.

[FRR+10] Sebastian Faust, Tal Rabin, Leonid Reyzin, Eran Tromer, and Vinod Vaikuntanathan. Protecting circuits from leakage: the computationally-bounded and noisy cases. In Henri Gilbert, editor, *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 135–156. Springer, 2010.

[GPT14] Daniel Genkin, Itamar Pipman, and Eran Tromer. Get your hands off my laptop: Physical side-channel key-extraction attacks on pcs. In *CHES*, 2014.

[HBK00a] D. R. Heath-Brown and S. Konyagin. New bounds for gauss sums derived from kth powers, and for heilbronn's exponential sum. *Q. J. Math*, 2000.

[HBK00b] D.R. Heath-Brown and S. Konyagin. New bounds for gauss sums derived from kth powers, and for heilbronn's exponential sum. *The Quarterly Journal of Mathematics*, 51(2):221–235, 2000.

[HK09] Dennis Hofheinz and Eike Kiltz. Practical chosen ciphertext secure encryption from factoring. In Antoine Joux, editor, *Advances in Cryptology – EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 313–332. Springer, April 2009.

[HL11] Shai Halevi and Huijia Lin. After-the-fact leakage in public-key encryption. In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 107–124. Springer, 2011.

[HSHea08] J. Alex Halderman, Seth D. Schoen, Nadia Heninger, and et al. Lest we remember: Cold boot attacks on encryption keys. In *17th USENIX Security Symposium*, 2008.

[HW09] Susan Hohenberger and Brent Waters. Short and stateless signatures from the RSA assumption. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 654–670. Springer, August 2009.

[ISW03] Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In Dan Boneh, editor, *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 463–481. Springer, 2003.

[KJJ99] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *CRYPTO*, 1999.

[KK12] Saqib A. Kakvi and Eike Kiltz. Optimal security proofs for full domain hash, revisited. In *EUROCRYPT*, pages 537–553, 2012.

[KL07] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography: Principles and Protocols*. Chapman & Hall / CRC Press, 2007.

[Kob87] N. Koblitz. Elliptic curve cryptosytems. *Mathematics of Computation*, 1987.

[Koc96] Paul Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *CRYPTO*, 1996.

[KOS10] Elke Kiltz, Adam O'Neill, and Adam Smith. Instantiability of RSA-OAEP under chosen-plaintext attack. In *CRYPTO*, 2010.

[LHA+12] Arjen K. Lenstra, James P. Hughes, Maxime Augier, Joppe W. Bos, Thorsten Kleinjung, and Christophe Wachter. Public keys. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO*, volume 7417 of *LNCS*, pages 626–642. Springer, 2012.

[LLW11] A. B. Lewko, M. Lewko, and B. Waters. How to leak on key updates. In *STOC*, 2011.

[LOS13] Mark Lewko, Adam O'Neill, and Adam Smith. Regularity of lossy RSA on subdomains and its applications. In *EUROCRYPT*, 2013.

[LT82] Thomas Lengauer and Robert E. Tarjan. Asymptotically tight bounds on time-space trade-offs in a pebble game. *Journal of ACM*, 29:1087 – 1130, 1982.

[MDS99] Thomas S. Messerges, Ezzy A. Dabbish, and Robert H. Sloan. Investigations of power analysis attacks on smartcards. In *USENIX Workshop on Smartcard Technology*, 1999.

[Mil85] V. Miller. Use of elliptic curves in cryptography. In *CRYPTO*, 1985.

[MMS13] Kai Michaelis, Christopher Meyer, and Jörg Schwenk. Randomly failed! the state of randomness in current java implementations. In Ed Dawson, editor, *CT-RSA 2013*, volume 7779 of *LNCS*, pages 129–144. Springer, 2013.

[MR04] Silvio Micali and Leonid Reyzin. Physically observable cryptography (extended abstract). In Moni Naor, editor, *TCC*, volume 2951 of *Lecture Notes in Computer Science*, pages 278–296. Springer, 2004.

[MVW95] H.L. Montgomery, R.C. Vaughan, and T.D. Wooley. Some remarks on gauss sums associated with kth powers. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 118, pages 21–33. Cambridge Univ Press, 1995.

[NS09] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 18–35. Springer, 2009.

[NTY11] Hitoshi Namiki, Keisuke Tanaka, and Kenji Yasunaga. Randomness leakage in the kem/dem framework. In Xavier Boyen and Xiaofeng Chen, editors, *ProvSec*, volume 6980 of *LNCS*, pages 309–323. Springer, 2011.

[Pip77] N. Pippenger. Superconcentrators. *SIAM Journal on Computing*, 6:298 – 304, 1977.

[RSA78a] R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. In *Communications of the ACM*, 1978.

[RSA78b] 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.

[Vad] Salil Vadhan. Pseudorandomness.

[Val75] Leslie G. Valiant. On non-linear lower bounds in computational complexity. In *STOC*, 75.

[Wat09] Brent Waters. Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 619–636. Springer, 2009.

[YCZY12a] Tsz Hon Yuen, Sherman S. M. Chow, Ye Zhang, and Siu Ming Yiu. Identity-based encryption resilient to continual auxiliary leakage. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT*, volume 7237 of *LNCS*, pages 117–134. Springer, 2012.

[YCZY12b] Tsz Hon Yuen, Sherman S. M. Chow, Ye Zhang, and S.M. Yiu. Identity-based encryption resilient to continual auxilary leakage. In *EUROCRYPT*, 2012.

[YSPY] Yu Yu, Francois-Xavier Standaert, Olivier Pereira, and Moti Yung. Practical leakage-resilient pseudorandom generators. In *CCS 2010*.

[YYH12] Tsz Hon Yuen, Siu Ming Yiu, and Lucas Chi Kwong Hui. Fully leakage-resilient signatures with auxiliary inputs. In Willy Susilo, Yi Mu, and Jennifer Seberry, editors, *ACISP 2012*, volume 7372 of *LNCS*, pages 294–307. Springer, 2012.

Vita

Ye Zhang

Ye Zhang is a PhD candidate in the Department of Computer Science and Engineering at the Pennsylvania State University. Before joining Penn State, he received MPhil degree in the University of Hong Kong. His research interests include cryptography, database security and theoretical computer science. His recent work focus on psuedorandomness (e.g., expander graphs, randomness extractor and etc). His research articles have been published in conferences such as EUROCRYPT, VLDB and TCC. His recent work has won the best paper award at ESORICS'2014.