Weak-Key Leakage Resilient Cryptography

Zuoxia Yu [*1], Qiuliang Xu [†1], Yongbin Zhou [‡2], Chengyu Hu [§1], Rupeng Yang [¶1], and Guangjun Fan [‖3]

[1]School of Computer Science and Technology, Shandong University, Jinan, 250101, China
[2]State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, 100093, China
[3]Trusted Computing and Information Assurance Laboratory,Institute of Software,Chinese Academy of Sciences,Beijing,100190,China

Abstract

In traditional cryptography, the standard way of examining the security of a scheme is to analyze it in a black box manner which does not capture the side channel attacks. Such attacks can exploit various forms of unintended information leakage and threaten the practical security. One way to protect against such attacks is to extend the traditional models to capture them. Early models rely on the assumption that *only computation leaks information*, and can not capture memory attacks such as cold boot attacks. Thus, Akavia et al. (TCC '09) formalize the model of key-leakage attacks to cover them. However, as we will mention below, most key-leakage attacks in reality may be weak key-leakage attacks which can be viewed as a non-adaptive version of the key-leakage attacks. And the existing construction of cryptographic schemes in models that can capture adaptive key-leakage attacks has some drawbacks. We mainly consider the models that cover weak key-leakage attacks and the corresponding constructions in them.

In this paper, we extend the transformation paradigm presented by Naor and Segev that can transform from any chosen-plaintext secure public key encryption (PKE) scheme into a chosen-plaintext weak key-leakage secure PKE scheme. Our extensions are mainly in two manners. On one hand, we extend the paradigm into chosen-ciphertext attack scenarios and prove that the properties of the paradigm still hold when we consider chosen-ciphertext attacks. We also give an instantiation based on DDH assumption in this setting for concrete. On the other hand, we extend the paradigm to cover more powerful side channel attacks. We do this by relaxing the restrictions on leakage functions. We further consider attacks that require the secret key still has enough min-entropy after leaking and prove the original paradigm is still applicable in this case with chosen-ciphertext attacks. We also consider attacks that require the secret key is computationally infeasible to recover given the leakage information and formalize the informal discusses by Naor and Segev in (Crypto' 09) on how to adapt the original paradigm in this new models.

Keywords: weak key-leakage attacks, public-key encryption scheme, chosen-ciphertext security

[*]yuzuoxia1990@gmail.com
[†]Corresponding author: xql@sdu.edu.cn
[‡]zhouyongbin@iie.ac.cn
[§]hcy@sdu.edu.cn
[¶]orbbyrp@gmail.com
[‖]guangjunfan@163.com

1 Introduction

In traditional cryptography, the standard way of examining the security of a scheme is to analyze it in a black box manner. In such abstract models, the legal involved parties are viewed as interactive Turing machines, and each party has its own local secrets. On the other hand, the adversaries in these models can only access the secrets in some restricted ways. For example, in the scenario of digital signature, the adversary may only be able to learn knowledge about the secret which is the signing key by querying the signature of messages that he chooses. However, this assumption may not hold in reality since the adversary can use some unintended ways to fetch some information about the secrets. One important class of such ways are called side channel attacks. Furthermore, they have made great contributions to the breaking down of many classical cryptographic systems that are believed to be secure in the black box models(e.g.[17, 16]).

To protect against side channel attacks, there are mainly two complementary approaches to our best knowledge. One approach is to diminish the damage of specific attacks by designing appropriate hardwares. For example, one can imprison a card in a Faraday cage to block the electromagnetic field radiation [22]. However, working on the hardware level is rather expensive, inefficient and is even impossible in some cases. So the other approach is always more appealing. In this strategy, one uses abstract notions of computation to depict side channel attacks, modifies the black box models by considering these attacks and constructs secure schemes in these new models. Note that, our work in this paper belongs to the latter approach.

The method of protecting against side channel attacks by treating them in an abstract way was first proposed by Micali and Reyzin in their pioneering work [18]. However, their model relies on the assumption that *only computation leaks information* and thus can not capture memory attacks which can obtain secret information that is not being used in computation such as cold boot attacks introduced by [13].

Inspired by the cold boot attacks presented by Halderman et al. [13], Akavia et al. [1] formalized a general framework for modeling memory attacks in which the adversaries are allowed to access the secrets by querying a series of efficiently computable functions of the secrets adaptively as long as the total amount of output bits of these functions is bounded. A scheme is said secure in this model if it can maintain its original security (in the black box model without leakage) even under attacks of such adversaries. They also present a weaker notion of security that demands the adversaries to determine the leakage functions non-adaptively which means that the adversaries can only choose the leakage functions in the beginning and thus these functions are independent with the public parameters of the target schemes. We call attacks considered in these two models *key-leakage attacks* and *weak key-leakage attacks* respectively.

Although weak key-leakage attacks seem much more restricted compared with the key-leakage attacks, they are enough to cover realistic attacks in most insensitive demotic scenarios since most memory attacks in reality are actually non-adaptive to our best knowledge. For example, the cold boot attack as well as most viruses, worms and malware attacks can be viewed as weak key-leakage attacks, and they occupy a large proportion of the attacks mostly used in reality. Also, it is quite expensive to exploit adaptive leakage attacks because such attacks should be designed differently for each application and thus are not economic in common insensitive scenarios.

Another reason why it is meaningful to consider models with weak key-leakage attacks is because there are some drawbacks when consider key-leakage attacks in constructing secure schemes. Take public-key encryption (PKE) schemes resilient to chosen-ciphertext attacks as an example. It is

not easy to construct PKE schemes which are secure under chosen-ciphertext key-leakage attacks since the adversary that can learn information about the secret key adaptively may break the sophisticated structures of the PKE schemes. For example, a negative result about constructing public-key encryption scheme that is secure under key-leakage attacks with lossy trapdoor functions has been presented by Qin, Liu et al. in [21]. In addition, the existing PKE schemes secure under key-leakage attacks as well as chosen-ciphertext attacks are either extremely inefficient (e.g. constructing with simulation sound NIZK proof system in [19]), quite complicated (e.g. constructing with hash proof system and lossy filter in [20]), or with a very low leakage rate [1] (e.g. constructing with hash proof system in [19]). In contrast, it is straightforward (as presented in this paper) to transform from any chosen-ciphertext secure (without leakage) PKE scheme into a PKE scheme that is secure under chosen-ciphertext weak key-leakage attacks with leakage rate close to 1, and the new PKE scheme is nearly as efficient as the original one.

To protect against weak key-leakage attacks, Naor and Segev have presented a general transformation from any chosen-plaintext secure PKE to a chosen-plaintext weak key-leakage secure PKE[19]. And we extend this paradigm to adapt more scenarios in this paper.

1.1 Our Results

In this paper, we extend the paradigm presented by Naor and Segev in [19] which can transform from any chosen-plaintext secure PKE scheme to a chosen-plaintext weak key-leakage secure PKE scheme to adapt more scenarios in different directions.

Our first attempt is to extend the paradigm into chosen-ciphertext attack scenarios. More precisely, we prove that the properties of the paradigm still hold when we consider chosen-ciphertext attacks. That is to say, we can transform from any chosen-ciphertext secure PKE scheme to a PKE scheme that is secure under chosen-ciphertext weak key-leakage attacks. Also, the new PKE scheme is nearly as efficient as the original one, and it can maintain its security even when the secret key is nearly totally revealed (the leakage rate is close to 1). We also give an instantiation based on DDH assumption in this setting for concrete.

Another approach to extend the paradigm is to cover more powerful side channel attacks. The original weak key-leakage attacks acctually means the weak λ-key-leakage attacks, which request the leakage information that the adversary obtains is no more than λ bits. Other restrictions on the leakage functions include one that requests the conditional mutual information [2] of the secret key and the leakage information given public key is at most λ, and one that only requests the secret key is computationally infeasible to recover given the leakage information. We call weak key-leakage attacks with above restrictions on the leakage functions *weak noisy leakage attacks* and *weak auxiliary input attacks* respectively. In this paper, we also consider PKE schemes that are secure under chosen-ciphertext weak noisy leakage attacks and secure under chosen-ciphertext weak auxiliary input attacks. More precisely, we prove the original paradigm is also applicable in weak noisy leakage attacks scenarios and modify the paradigm a bit to adapt in weak auxiliary input attacks scenarios.

[1] Leakage rate is the ratio of allowed leakage amount to the length of secret key.

[2] This is about how much information one variable tells about another in average case.

1.2 Related Work

Key-leakage attacks The model considering key-leakage attacks was first formalized by Akavia, Goldwasser and Vaikuntanathan in [1]. They also proved that the Regev encryption scheme in [23] is secure under chosen-plaintext key-leakage attacks. However their scheme can only tolerate a very tiny amount of leakage information, which is about $O(N/\log N)$ where N is the length of the secret key. To construct PKE schemes that can tolerate more leakage amount, Naor and Segev [19] improved the chosen-plaintext secure PKE scheme based on hash proof system (HPS) in [6] to be secure under chosen-plaintext key-leakage attacks with leakage rate about $\frac{1}{2}$. They also extend the above scheme to be secure under chosen-ciphertext key-leakage attacks either by simulation sound NIZK proof system or by $universal_2$ HPS. However the new scheme based on simulation sound NIZK proof system is extremely inefficient because NIZK proof systems are inherently inefficient. Also the new scheme based on HPS has a quite low leakage rate (at most $\frac{1}{6}$) since they have a bigger secret key. To construct an efficient scheme with a leakage rate that is not too low, Qin and Liu [20] exploit one-time lossy filter to extend the original chosen-plaintext key-leakage secure PKE scheme in [19] and the new scheme is compareable to the chosen-ciphertext key-leakage secure PKE scheme based on $universal_2$ HPS in efficiency with a leakage rate close to $\frac{1}{2}$. But the construction is quite complicated and so it is not easy to analyze and implement it. There are also various other cryptographic primitives that are secure against key-leakage attacks, e.g. the signature schemes [15, 4], the identity based encryption schemes [2] and so on.

A natural extension of the original notion of key-leakage attacks proposed in [1] is to relax the restriction on the leakage functions and only requires the conditional mutual information of the secret key and the leakage information given public key is at most λ. The new notion was first proposed by Naor and Segev in [19] and most existing schemes that are secure under original key-leakage attacks are also secure in the new models since the two notions are of great resemblance.

One can further relax the restriction on the leakage functions and only requires the secret key is computationally hard to recover given the leakage information. Models considering such attacks are called auxiliary input models. This model was first proposed by Dodis et al. in [9] and they also present secure symmetric-key encryption schemes in this model in both chosen-plaintext attacks scenarios and chosen-ciphertext attacks scenarios. Public key schemes that are secure in the auxiliary input models are then constructed by Dodis et al.[7] based on either DDH assumption and LWE assumption.

Weak key-leakage attacks Weak key-leakage attacks was first proposed by Akavia et al.[1] as a weaker version of key-leakage attacks. However, as mentioned above, for most common attacks in reality, it is enough to consider this weaker notion of key-leakage attacks. Constructing secure schemes under weak key-leakage attacks is not hard. Naor and Segev presented an efficient generic transformation from any chosen-plaintext secure PKE to a chosen-plaintext weak key-leakage secure PKE in [19] whose leakage rate is $(1 - o(1))$. Our work is based on this transformation paradigm and extend it to much richer scenarios. In another work, Qin et al.[21] presented a construction of chosen-ciphertext weak key-leakage secure PKE scheme from leakage-resilient lossy trapdoor functions. They consider a more generic model in which partial information of the public key can be used to choose leakage functions by the adversary. However, this generation may not capture much more attacks scenarios, and their construction relies on a concrete underlying primitive, while we provide a generic transformation from any chosen-ciphertext secure public-key encryption

scheme.

2 Priminaries

In this section, we recall some basic notions, terminology and computational assumption.

2.1 Basic Notions

A function f mapping non-negative integers to non-negative reals is *negligible* if for every polynomial $\mathrm{p}(\cdot)$, there exists an integer $n_0 \geq 0$ such that for all integers $n > n_0$, $f(n) < \frac{1}{p(n)}$.

We denote by U_m a random variable with uniform distribution over $\{0,1\}^m$ and write $[n]$ to denote the set $\{1, 2, 3, \ldots, n\}$. What's more, for a finite set S, we denote by $|S|$ the number of elements in S. In addition, all logarithms in this paper are with base 2.

2.2 One-Way Function and Goldreich-Levin Hard-core Bits

Assume v, h, m be functions mapping the security parameter of a public-key encryption scheme denoted as n to a non-negative integers, and for simplicity, we omit the security parameter n. Let sk be the secret key of a public-key encryption scheme whose length is v bits.

Definition 2.1 ([7]). *For any polynomial time computable function $f : \{0,1\}^v \to \{0,1\}^*$. We say the hardness parameter of f is $\lambda(v)$ means that given* f(SK)*, there is no probabilistic polynomial time algorithm can find sk with probability greater than $\frac{1}{2^{\lambda(v)}}$.*

Let $\mathcal{F}_{ow}(\lambda(v))$ be the collection of all polynomial time computable function $f : \{0,1\}^v \to \{0,1\}^*$ whose hardness parameter is $\lambda(v)$.

Let *GL(sk,h)* be the Goldreich-Levin hard-core bits [11]. And the output of *GL(sk,h)* is computationally indistinguishable from uniform even given the value of *f(sk)* for some function f as long as f is hard to invert. More precisely, if GL is a Goldreich-Levin hard-core bits from $\{0,1\}^v \times \{0,1\}^t$ to $\{0,1\}^m$, the advantage to distinguish $(GL(sk,h),\ f(sk),\ h)$ and $(y,\ f(k),\ h)$ where y is from U_m is negligible in v for any efficient adversary $\mathcal{A}$, as long as f is from $\mathcal{F}_{ow}(m)$ and m is a polynomial of v.

2.3 Randomness Extractor

We recall some basic notions relating to randomness extractors here.

Definition 2.2. *Let X and Y be two random variables in a finite set U. The statistical distance between X and Y is defined as*

$$SD(X,\ Y) = \frac{1}{2} \sum_{u \in U} |Pr[X = u] - Pr[Y = u]|$$

We say that two variables are $\varepsilon - close$ if their statistical distance is at most ε.

Definition 2.3 ([10]). *Let X be a random variable. Then the min-entropy of X, denoted $H_\infty(X)$, is defined as*

$$H_\infty(X) = -\log(max_x Pr[X = x])$$

Definition 2.4 ([10]). *Let X and Y be two random variables. Then the average min-entropy of X conditioned on Y, denoted $\tilde{H}_\infty(X|Y)$, is defined as*

$$\begin{aligned}\tilde{H}_\infty(X|Y) &= -\log(\mathbb{E}_{y\leftarrow Y}[\underset{x\leftarrow X}{max} Pr[X=x|Y=y]]) \\ &= -\log(\mathbb{E}_{y\leftarrow Y}[2^{-H_\infty(X|Y=y)}])\end{aligned}$$

Lemma 2.5. *Let X, Y and Z be random variables. If Y has at most 2^λ possible values, then $\tilde{H}_\infty(X|(Y,\ Z)) \geq \tilde{H}_\infty((X,\ Y)|Z) - \lambda \geq \tilde{H}_\infty(X|Z) - \lambda$.*

The proof of this lemma can be found in [10].

Definition 2.6. *Let U be a finite set. A function $Ext : U \times \{0,1\}^t \to \{0,1\}^m$ is an average-case $(v,\ \epsilon)$-strong extractor if for all pairs of random variables $(X,\ I)$ such that the range of X is U and $\tilde{H}_\infty(X|I) \geq v$, it holds that*

$$SD((Ext(X,\ R),\ R,\ I), (U_m,\ R,\ I)) \leq \epsilon$$

where R is uniform on $\{0,1\}^t$.

We remark that as we can easily encode elements in U into binary strings in our settings, we just write Ext as a function from $U \times \{0,1\}^t$ to $\{0,1\}^m$ for the given U for simplicity.

For the existence of average-case randomness extractors, Dodis et al [10]. proved that from any family of universal hash functions we can get an average-case strong extractor. More precisely, we have:

Lemma 2.7. *Fix an output length m, for any $v \geq 0$ and $\epsilon \geq 0$ we have an average-case $(v,\ \epsilon)$-strong extractor from $U \times \{0,1\}^t$ to $\{0,1\}^m$ as long as $m \leq v - 2\log(\frac{1}{\epsilon}) + 2$.*

2.4 Computational Assumption

Let *GroupGen* be a probabilistic polynomial-time algorithm that takes 1^n as input and outputs a tuple $(G,\ q,\ g)$, where q is a prime whose length is a polynomial of n, G is a cyclic group of order q and g is generator of G.

Informally speaking, the Decisional Diffie-Hellman (DDH) assumption is based on the hardness of the discrete logarithm problem. Furthermore, DDH assumption means that the relationship between power exponents are hiden.

Definition 2.8. *The Decisional Diffie-Hellman problem is hard relative to GroupGen if for any probabilistic polynomial-time algorithm $\mathcal{A}$ it holds that*

$$|Pr[\mathcal{A}(G,\ g,\ g^{r_1},\ g^{r_2},\ g^{r_1 r_2})] - Pr[\mathcal{A}(G,\ g,\ g^{r_1},\ g^{r_2},\ g^{r_3})]|$$

is negligible in n where $(q,\ G,\ g) \leftarrow GroupGen(1^n)$ and r_1, r_2, r_3 are chosen uniformly at random from Z_q. Here the probability is defined by choosing r_1, r_2 and r_3 at random, as well as the internal coin tosses of GroupGen and $\mathcal{A}$.

The DDH assumption is the assumption that the DDH problem is hard for some algorithm *GroupGen*.

3 Modeling Weak Key-Leakage Attacks

Inspired by the work of Akavia et al.[1], in this section, we extend the model of Akavia et al. [1] to the setting of adaptive chosen-ciphertext attacks, which almost covers all scenarios of weak key-leakage attacks. In the setting of adaptive chosen-ciphertext attacks, we take into account three types of leakage functions which are in non-adaptive key-leakage attacks, and they are: leakage function whose output length is bounded, leakage function whose conditional mutual information is limited, and leakage function who is hard to invert. Additionally, the adversary is allowed to access the decryption oracle $\mathrm{Dec}(SK, \cdot)$ that takes a ciphertext as input and output the corresponding decryption of the ciphertext. We define $\mathrm{Dec}_{\neq C^*}(SK, \cdot)$, a decryption oracle that can decrypt any ciphertext except ciphertext C^*. The following three models will use one of these leakage functions respectively.

3.1 Chosen-Ciphertext Weak λ-Key-Leakage Attacks

In this section, we consider a model of chosen-ciphertext weak λ-key-leakage attacks whose leakage function is with bounded output length denoted as λ. As mentioned before, the leakage function is given to the adversary before he knows any knowledge of the public-key.

Definition 3.1. *(wlr-CCA security) A public-key encryption scheme Π = (KeyGen,Enc,Dec) is chosen-ciphertext secure under weak λ-key leakage attacks if for any probabilistic polynomial-time λ-key leakage adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and for any ensemble $\mathcal{F} = \{f_n : SK_n \to \{0,1\}^{\lambda(n)}\}_{n \in N}$ it holds that*

$$Adv^{wlrCCA}_{\Pi,\mathcal{A}}(n) = |Pr[Expt^{wlrCCA}_{\Pi,\mathcal{A},\mathcal{F}}(n,0) = 1] - Pr[Expt^{wlrCCA}_{\Pi,\mathcal{A},\mathcal{F}}(n,1) = 1]|$$

is negligible in n, where $Expt^{wlrCCA}_{\Pi,\mathcal{A},\mathcal{F}}(n,b)$ is defined as follows:

Experiment $Expt^{wlrCCA}_{\Pi,\mathcal{A},\mathcal{F}}(n,b)$:

1. $(SK, PK) \leftarrow KeyGen(1^n)$.
2. $(M_0, M_1, state) \leftarrow \mathcal{A}_1^{Dec(SK,\cdot)}(PK, f_n(SK))$ s.t. $|M_0| = |M_1|$.
3. $b \leftarrow \{0,1\}$.
4. $C^* \leftarrow Enc(M_b)$.
5. $b' \leftarrow \mathcal{A}_2^{Dec_{\neq C^*}(SK,\cdot)}(C^*, state)$.
6. *output* b'.

3.2 Chosen-Ciphertext Weak Noisy Leakage Attacks

Instead of limiting the output length of leakage function, we consider a more general scenario. In [13], Halderman et al. presented that the adversary can learn a noisy version of all the memory. Then, inspired by the work of Naor and Segev [19], we can make a generalization of the weak noisy leakage attacks, that on one hand the length of leakage is not bounded, but on the other hand it is

conditioned that the secret key is still unpredictable even given the noisy leakage. Note that, as in weak noisy leakage attacks, the noisy leakage is chosen by the adversary ahead of time before he knows anything about the public key. We denote the leakage information by any random variable W, and only require that the conditional mutual information of the secret key and W given public key is at most λ, which we can write as following:

$$\widetilde{I}_{\infty}(SK; W|PK) = \widetilde{H}_{\infty}(SK|PK) - \widetilde{H}_{\infty}(SK|PK, W) \leq \lambda$$

Definition 3.2. *(weak noisy-leakage attacks) A public-key encryption scheme* Π *= (KeyGen,Enc,Dec) is chosen-ciphertext secure under weak noisy leakage attacks if for any probabilistic polynomial-time noisy leakage adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *and for any ensemble* $\mathcal{W}$ *of random variables* W *for which the conditional mutual information of the secret key and* W *given public key is at most* λ*, it holds that*

$$Adv^{wnlrCCA}_{\Pi,\mathcal{A}}(n) = |Pr[Expt^{wnlrCCA}_{\Pi,\mathcal{A},\mathcal{W}}(n,0) = 1] - Pr[Expt^{wnlrCCA}_{\Pi,\mathcal{A},\mathcal{W}}(n,1) = 1]|$$

is negligible in n*, where* $Expt^{wnlrCCA}_{\Pi,\mathcal{A},\mathcal{W}}(n,b)$ *is defined as follows:*

Experiment $Expt^{wnlrCCA}_{\Pi,\mathcal{A},\mathcal{W}}(n,b)$:

1. $(SK, PK) \leftarrow KeyGen(1^n)$.
2. $(M_0, M_1, state) \leftarrow \mathcal{A}_1^{Dec(SK,\cdot)}(PK, W)$ s.t. $|M_0| = |M_1|$.
3. $b \leftarrow \{0,1\}$.
4. $C^* \leftarrow Enc(M_b)$.
5. $b' \leftarrow \mathcal{A}_2^{Dec_{\neq C^*}}(C^*, state)$.
6. *output* b'.

3.3 Chosen-Ciphertext Weak Auxiliary Input Attacks

Next, we introduce another model of weak key leakage attacks which we call weak auxiliary input attacks. Recall the approach presented by Dodis et al.[9], the security analysis of the symmetric-key schemes is based on the presence of leakage of the form *f(SK)*, where SK is the secret key and f is any exponentially-hard one-way function. Without any restriction on the average min-entropy of the secret key, they only require that leakage function *f(SK)* is extremely hard to invert. Let $\lambda(v)$ be the hardness parameter of *f(SK)*, where v is the length of SK. We extend the above approach to weak auxiliary input attacks, which means that a extremely hard to invert leakage function $f(SK) \in \mathcal{F}_{ow}(\lambda(v))$ is chosen ahead of time by the adversary who knows nothing about the public key at that moment.

Definition 3.3. *(weak auxiliary input attacks) A public-key encryption scheme* Π *= (KeyGen,Enc,Dec) is chosen-ciphertext secure under weak auxiliary input attacks if for any probabilistic polynomial-time noisy leakage adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *and for any polynomial* λ*, it holds that*

$$Adv^{walr}_{\Pi,\mathcal{A}}(n) = |Pr[Expt^{walr}_{\Pi,\mathcal{A},\mathcal{F}_{ow}(\lambda)}(n,0) = 1] - Pr[Expt^{walr}_{\Pi,\mathcal{A},\mathcal{F}_{ow}(\lambda)}(n,1) = 1]|$$

is negligible in n, where $Expt^{walrCCA}_{\Pi,\mathcal{A},\mathcal{F}_{ow}(\lambda)}(n,b)$ *is defined as follows:*

Experiment $Expt^{walrCCA}_{\Pi,\mathcal{A},\mathcal{F}_{ow}(\lambda)}(n,b)$:

1. $(SK, PK) \leftarrow KeyGen(1^n)$.
2. $(M_0, M_1, state) \leftarrow \mathcal{A}_1^{Dec(SK,\cdot)}(PK, f(SK))$ s.t. $|M_0| = |M_1|$.
3. $b \leftarrow \{0,1\}$.
4. $C^* \leftarrow Enc(M_b)$.
5. $b' \leftarrow \mathcal{A}_2^{Dec_{\neq C^*}(SK,\cdot)}(C^*, state)$.
6. *Output* $1\ if\ b' = b$, *otherwise output* 0.

4 Generic Constructions of Weak Key-Leakage Resilient Cryptography Schemes

In this section, we propose several public-key encryption schemes which are respectively secure in the models mentioned in **Section 3** via the method presented in [19].

4.1 Overview

Weak key-leakage attacks as mentioned above means that different types of leakage functions are chosen by the adversary ahead of time without any knowledge of the public key. One approach to construct a generic scheme which is weak key-leakage resilient chosen-ciphertext secure is to transform from any chosen-ciphertext secure public-key encryption scheme. More precisely, we first use the internal coin tosses of the key generation algorithm as the secret key and generates public key with it. In addition, when decrypts a ciphertext, the receiver generates the secret key of the original scheme temporarily and decrypts using it. The remaining parts works identically to the underlying scheme. Thus, when the adversary gets the leakage about the secret key, the whole scheme can be viewed as a public-key encryption scheme with a "bad" randomness. We can then exploit different techniques that extract "good" randomness from "bad" randomness according to different scenarios. In this way, we obtain a weak key-leakage resilient chosen-ciphertext secure public-key encryption scheme.

Note that, the transformation techniques mentioned above are not trivial. On the contrary, it is very essential and efficient. More precisely, it is evident that not all chosen-ciphertext secure public-key encryption schemes are secure under the weak key-leakage attacks. For example, the cold boot attack can completely compromise the security of RSA which can be chosen-ciphertext secure without leakage.

4.2 Generic Construction in Chosen-Ciphertext Weak λ-Key-Leakage Model

Firstly, we consider the generic construction in the model of weak λ-key-leakage attacks. As mentioned below, the resulting encryption scheme of our generic construction is resilient to any leakage of $v(1 - o(1))$ bits under the chosen-ciphertext attack, where v is the length of secret key.

Let m, v, t, λ be functions mapping security parameter n to non-negative integers and ϵ be function mapping n to non-negative real. For simplicity, we omit the security parameter in the following. Assume that $\Pi =$ *(KeyGen,Enc,Dec)* be any public-key encryption scheme, and denote by m be the length of the random string used by *KeyGen*, λ is the leakage parameter. And assume $Ext : \{0,1\}^v \times \{0.1\}^t \to \{0,1\}^m$ to be an average-case$(v-\lambda, \epsilon)$-strong extractor for some negligible ϵ. Consider a public-key encryption scheme $\Pi_\lambda = (KeyGen_\lambda, Enc_\lambda, Dec_\lambda)$ defined as follows:

- Key generation: On input the security parameter 1^n, we choose $x \in \{0,1\}^v$ and $h \in \{0,1\}^t$ uniformly at random. Then compute $(pk, sk) \leftarrow KeyGen(Ext(x,h))$. Output $PK =$ *(pk,h)* and $SK=x$.
- Encryption: On input a message M and a public-key $PK =$ *(pk,h)*, compute the ciphertext $C \leftarrow Enc(pk, M)$, output C.
- Decryption: On input a ciphertext C and a secret key $SK = x$, compute $(pk, sk) = KeyGen(Ext(x,h))$ and output $Dec(\text{sk,C}) = M$.

Theorem 4.1. *Let $\Pi = (KeyGen, Enc, Dec)$ be a chosen-ciphertext secure public-key encryption scheme. Then for any polynomial λ , the scheme $\Pi_\lambda = (KeyGen_\lambda, Enc_\lambda, Dec_\lambda)$ is chosen-ciphertext secure against weak λ-key-leakage attacks.*

Proof. We prove this theorem via a series of games described as follows. We assume that the secret key is denoted as x, the public key is denoted as (pk, h). $\mathcal{A}$ is a chosen-ciphertext weak λ-key leakage adversary.
$\mathbf{Game}_1$ is the original weak λ-key leakage chosen-ciphertext attacks experiment, $Expt^{wlrCCA}_{\Pi,\mathcal{A},\mathcal{F}}(n,b)$. As described in the experiment, before knowing any knowledge of public key, $\mathcal{A}$ can get leakage function $f(x)$ whose output length is λ bits.
$\mathbf{Game}_2$ is identical to $\mathbf{Game}_1$ except that we replace the parameter of *KeyGen* with a truly random string $y \in \{0,1\}^m$.

Claim 4.2. ***Game***$_1$ *and* ***Game***$_2$ *are indistinguishable.*

Proof. To show that, we firstly fix the internal coin tosses of adversary $\mathcal{A}$ and variable b in $\mathbf{Game}_1$ and $\mathbf{Game}_2$ as they are chosen from the corresponding identical distributions in two games.

Conditioned on that, we further fix the random variable h used by extractor, the leakage function $f(x)$, and the input value of *KeyGen* denoted as r. It is evident that when $h, f(x)$ and r are fixed, the output of the two games are identical. Note that when A queries the decryption oracle, it use sk to decrypt which is also determined by r. Therefore, the results of the two games can be modeled as the output of a deterministic function whose input is r, h and $f(x)$. Since the two games are identical except the distribution of these three values, we can write as follows: $GameResult = R(r, h, f(x))$. As observed from above, in $\mathbf{Game}_1$, r is the output of extractor; in $\mathbf{Game}_2$, r is y. Then, $GameResult_1 = R(Ext(x,h), h, f(x))$, $GameResult_2 = R(y, h, f(x))$. Whereas, although in $\mathbf{Game}_1$, adversary A can get any leakage function $f(x)$ whose output length is $\lambda(n)$ bits. But by **Lemma2.5**, the following holds:

$$\tilde{H}_\infty(x|f(x)) \geq H_\infty(x) - \lambda \geq v - \lambda$$

It implies that the average min-entropy of the first input of extractor Ext is still bigger than $v - \lambda$. Then the property of Ext guarantees that the two tuples $(Ext(x,h), h, f(x))$ and $(y, h, f(x))$ are statistically indistinguishable in any conditional probability space conditioned on the internal coin tosses of adversary A and variable b. In this way, we can draw a conclusion that $\mathbf{Game}_1$ and $\mathbf{Game}_2$ are indistinguishable. □

Claim 4.3. *The advantage of A in* $\boldsymbol{Game}_2$ *is negligible.*

Proof. As mentioned above, in $\mathbf{Game}_2$, the internal coin tosses of algorithm *KeyGen* is a truly random string y, thus $f(x)$ is independent of y. Hence, Π_λ is the same as the underlying scheme Π. It is evident that the underlying public-key encryption scheme Π is CCA-secure which implies that in $\mathbf{Game}_2$ the advantage of A is negligible.

□

Now, this completes the proof.

□

As observed form above, to maintain the security of scheme Π_λ, the average min-entropy of the first input of extractor needs to be bigger than $v - \lambda$ which is denoted as variable p. Then the leakage rate can be written as follows:

$$\textit{Leakage-rate} = \frac{\lambda}{v} = \frac{v-p}{v} = 1 - \frac{p}{v}$$

Thus we can make the values of v, λ large enough but maintain the value of p. In this way, the leakage rate can reach $(1 - o(1))$ without decreasing the security of the scheme Π_λ.

4.3 Generic Construction in Chosen-Ciphertext Weak Noisy Leakage Model

Using nearly the same method as mentioned above, we present the generic construction of weak noisy leakage chosen-ciphertext secure public-key encryption scheme as follows.

Let Π = *(KeyGen,Enc,Dec)* be any public-key encryption scheme. Let m be the length of the random string used by *KeyGen*, W be any noisy leakage random variable under the requirement that the conditional mutual information of the secret key and W given public key is at most λ. Let $Ext : \{0,1\}^v \times \{0.1\}^t \to \{0,1\}^m$ be an average-case$(v - \lambda, \epsilon)$-strong extractor for some negligible ϵ. We present a public-key encryption scheme $\Pi_{noisy} = (KeyGen_{noisy}, Enc_{noisy}, Dec_{noisy})$ defined as follows:

- Key generation: On input parameter 1^n, choose $x \in \{0,1\}^v$ and $h \in \{0,1\}^t$ uniformly at random. Then compute $(pk, sk) \leftarrow KeyGen(Ext(x,h))$. Output PK = *(pk,h)* and $SK = x$.
- Encryption: On input a message M and a public-key PK=*(pk,h)*, compute the ciphertext $C \leftarrow$ *Enc(PK,M)*, output C.
- Decryption: On input a ciphertext C and a secret key $SK = x$, compute (pk, sk)=$KeyGen(Ext(x,h))$ and output *Dec(sk,C)*=M.

Theorem 4.4. *Let* Π *=(KeyGen,Enc,Dec) be a chosen-ciphertext secure public-key encryption scheme. Then for any noisy leakage random variable* W *under the requirement that the conditional mutual information of the secret key and* W *given public key is at most* λ*, the scheme* Π_{noisy} *=(*$KeyGen_{noisy}$*,* Enc_{noisy}*,* Dec_{noisy}*) is chosen-ciphertext secure against weak noisy leakage attacks.*

As the proof is identical to the the proof of **Theorem 4.1**, thus we omit it here.

4.4 Generic Construction in Chosen-Ciphertext Weak Auxiliary Input Model

In this section, we consider the generic construction in weak auxiliary input attacks model, where the secret key is computationally unpredictable which is different from the information-theoretically unpredictability in other two models.

Let m, v, t be functions mapping the security parameter n to non-negative integers, and m is also a polynomial of v. Assume the auxiliary input function *f(SK)* chosen by the adversary is from $\mathcal{F}_{ow}(m)$. Let *GL(SK,h)* be the Goldreich-Levin hard-core bits from $\{0,1\}^v \times \{0,1\}^t$ to $\{0,1\}^m$. Let Π = (*KeyGen,Enc,Dec*) be any public-key encryption scheme. Public-key encryption scheme $\Pi_{auxinput}$ = ($KeyGen_{auxinput}$, $Enc_{auxinput}$, $Dec_{auxinput}$) is defined as follows:

- Key generation : On input parameter 1^n, we choose $x \in \{0,1\}^v$ and $h \in \{0,1\}^t$ uniformly at random. Then compute $(pk, sk) \leftarrow$ *KeyGen(GL(x,h))*. Output *PK = (pk,h)* and *SK=x*.
- Encryption : On input a message M and a public-key *PK = (pk,h)*, compute the ciphertext $C \leftarrow$ *Enc(pk,M)*, output C.
- Decryption : On input a ciphertext C and a secret key $SK = x$, compute (pk, sk)= *KeyGen*$(GL(x, h))$ and output *Dec(sk,C)=M*.

Theorem 4.5. *Let* Π *=(KeyGen,Enc,Dec) be a chosen-ciphertext secure public-key encryption scheme. Then for any auxiliary input leakage function* $f(x)$ *mentioned above, the scheme* $\Pi_{auxinput}$ = ($KeyGen_{auxinput}$, $Enc_{auxinput}$, $Dec_{auxinput}$) *is chosen-ciphertext secure against weak auxiliary input attacks.*

The proof of **Theorem 4.5** is in **Appendix A**.

5 Conclusions and Future Work

In this paper, we first summarize three models of chosen-ciphertext weak key-leakage attacks, which almost cover all possible scenarios of weak key-leakage attacks in the setting of chosen-ciphertext attacks. In addition, for each model considered above, we propose a generic construction of public-key encryption schemes that are secure respectively in it. What's more, the generic constructions can transform any chosen-ciphertext secure public-key encryption scheme into one that is secure in our models mentioned above. The resulting scheme is as efficient as the original one without relying on any additional computational assumption. Moreover, the generic construction of chosen-ciphertext weak λ-key-leakage attacks model can be resilient to any weak leakage of $v(1 - o(1))$ bits, where v is the length of the secret key.

Furthermore, it is interesting to investigate whether there is a generic construction of chosen-ciphertext key-leakage resilient secure public-key encryption scheme from any chosen-ciphertext secure public-key encryption scheme.

References

[1] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In *Theory of Cryptography*, pages 474–495. Springer, 2009.

[2] Joël Alwen, Yevgeniy Dodis, Moni Naor, Gil Segev, Shabsi Walfish, and Daniel Wichs. Public-key encryption in the bounded-retrieval model. In *Advances in Cryptology–EUROCRYPT 2010*, pages 113–134. Springer, 2010.

[3] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 103–112. ACM, 1988.

[4] Elette Boyle, Gil Segev, and Daniel Wichs. Fully leakage-resilient signatures. In *Advances in Cryptology–EUROCRYPT 2011*, pages 89–108. Springer, 2011.

[5] Zvika Brakerski, Yael Tauman Kalai, Jonathan Katz, and Vinod Vaikuntanathan. Overcoming the hole in the bucket: Public-key cryptography resilient to continual memory leakage. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 501–510. IEEE, 2010.

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

[7] Yevgeniy Dodis, Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In *Theory of Cryptography*, pages 361–381. Springer, 2010.

[8] Yevgeniy Dodis, Kristiyan Haralambiev, Adriana Lopez-Alt, and Daniel Wichs. Cryptography against continuous memory attacks. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 511–520. IEEE, 2010.

[9] Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett. On cryptography with auxiliary input. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 621–630. ACM, 2009.

[10] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Advances in cryptology-Eurocrypt 2004*, pages 523–540. Springer, 2004.

[11] Oded Goldreich and Leonid A Levin. A hard-core predicate for all one-way functions. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 25–32. ACM, 1989.

[12] Shafi Goldwasser, Yael Kalai, Chris Peikert, and Vinod Vaikuntanathan. Robustness of the learning with errors assumption. 2010.

[13] J Alex Halderman, Seth D Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph A Calandrino, Ariel J Feldman, Jacob Appelbaum, and Edward W Felten. Lest we remember: cold-boot attacks on encryption keys. *Communications of the ACM*, 52(5):91–98, 2009.

[14] Shai Halevi and Huijia Lin. After-the-fact leakage in public-key encryption. In *Theory of Cryptography*, pages 107–124. Springer, 2011.

[15] Jonathan Katz and Vinod Vaikuntanathan. Signature schemes with bounded leakage resilience. In *Advances in Cryptology–ASIACRYPT 2009*, pages 703–720. Springer, 2009.

[16] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *Advances in CryptologyCRYPTO99*, pages 388–397. Springer, 1999.

[17] Paul C Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *Advances in CryptologyCRYPTO96*, pages 104–113. Springer, 1996.

[18] Silvio Micali and Leonid Reyzin. Physically observable cryptography. In *Theory of Cryptography*, pages 278–296. Springer, 2004.

[19] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. *SIAM Journal on Computing*, 41(4):772–814, 2012.

[20] Baodong Qin and Shengli Liu. Leakage-resilient chosen-ciphertext secure public-key encryption from hash proof system and one-time lossy filter. 2013.

[21] Baodong Qin, Shengli Liu, Kefei Chen, and Manuel Charlemagne. Leakage-resilient lossy trapdoor functions and public-key encryption. In *Proceedings of the first ACM workshop on Asia public-key cryptography*, pages 3–12. ACM, 2013.

[22] Jean-Jacques Quisquater and David Samyde. Electromagnetic analysis (ema): Measures and counter-measures for smart cards. In *Smart Card Programming and Security*, pages 200–210. Springer, 2001.

[23] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):34, 2009.

[24] Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. *IACR Cryptology ePrint Archive*, 2004:332, 2004.

A Proof of Theorem 4.5

Proof. Just like the proof of **Theorem 4.1**, here we also use game sequence to complete the proof. Assume that the secret key is denoted as x, the public key is denoted as *(pk,h)*. $\mathcal{A}$ is a chosen-ciphertext weak auxiliary input leakage adversary.
$\mathbf{Game}_1$ is the original weak auxiliary input chosen-ciphertext attacks experiment, $Expt^{walrCCA}_{\Pi_{auxinput},\mathcal{A},\mathcal{F}_{ow}(m)}(n,b)$.
$\mathbf{Game}_2$ is identical to $\mathbf{Game}_1$ except that we replace the parameter of *KeyGen* with a truly random string $y \in \{0,1\}^m$. Next, we first show that $A's$ views in $\mathbf{Game}_1$ and $\mathbf{Game}_2$ are computationally indistinguishable. If not, we can construct a distinguisher D, which can distinguish *GL(x,h)* from the truly random string y. D works as follows:

Distinguisher D:
D is given an input string s with length m, a hard to invert one-way function *f(x)* and a random string $h \in \{0,1\}^t$, where s is either $GL(x,h)$ or $y \in \{0,1\}^m$. Then D does as follows:

1. Run $(pk, sk) \leftarrow KeyGen_{auxiliary}(s)$. Then public key is $PK = (pk, h)$, and gives PK and *f(x)* to $\mathcal{A}$.
2. When $\mathcal{A}$ queries the decryption oracle with a ciphertext C, answer this query by computing $Dec(sk, C)$, since it knows sk.
3. When $\mathcal{A}$ outputs message M_0, M_1 where $|M_0| = |M_1|$, choose a random bit $b \in \{0,1\}$ and then return the challenge ciphertext $Enc(pk, M_b)$ to $\mathcal{A}$.
4. Continue answering any decryption oracle queries of A as above except any query about the challenge ciphertext. Eventually, A outputs $b^{'}$. Output 1, if $b = b^{'}$ and output 0 otherwise.

It is evident that D perfectly simulate $\mathbf{Game}_1$ and $\mathbf{Game}_2$ depending on s being $GL(x,h)$ or y. More precisely, the output of D is identical to $Game_1$ if s is $GL(x,h)$, and identical to $Game_2$ otherwise. Then we can draw a conclusion that the distinguisher D can distinguish the value of $GL(x,h)$ from a truly random string y given $f(x)$ and h, which contradicts the fact that hard-core bits is computationally indistinguishable from random.

The next step of the proof is to show that the advantage of $\mathcal{A}$ is negligible. We can observe that in $\mathbf{Game}_2$, the auxiliary input function $f(x)$ is independent of y. As y is a truly random string, then scheme Π is CCA-secure. Therefore scheme $\Pi_{auxinput}$ is also CCA-secure. Thus, the advantage of adversary $\mathcal{A}$ in $\mathbf{Game}_2$ is negligible, in addition, we have shown $\mathbf{Game}_1$ and $\mathbf{Game}_2$ are computationally indistinguishable, combining the above observations together, we complete our proof. □

B Concrete Encryption Scheme Based on Diffie-Hellman Assumption in Weak Key-Leakage Model

In this section, we instantiate the generic construction in **Section 4.2** by using Cramer-Shoup CCA-secure public-key encryption scheme [6] as the underlying CCA-secure PKE scheme. Additionally, we present an instantiation of the above scheme based on the Decision Diffie-Hellman assumption. As the process of instantiating the generic constructions in our three models are similar, we omit the instantiations in the other two models.

Let (G, q, g) be the output of algorithm *GroupGen* on input 1^n for the security parameter n, where G is a group of prime order q and g is a generator of it. Let $v, t, l, \vartheta, \lambda$ be functions mapping security parameter n to non-negative integers and let ε be a function mapping security parameter n to a non-negative real. Let $Ext : \{0,1\}^v \times \{0.1\}^t \rightarrow \{0,1\}^l$ be an average-case$(v - \lambda, \epsilon)$-strong extractor for some negligible ϵ. Let Γ be an efficiently computable injective encoding function, $\Gamma : G \times G \times G \rightarrow Z_q^d$.

– **Key generation**

On input security parameter 1^n, choose $x \in \{0,1\}^v$ and $h \in \{0,1\}^t$ uniformly at random. Then compute $Ext(x,h) = r$, and use string r as the internal coin tosses of algorithm *KeyGen*. It works as follows:

1. Assume that algorithm *GroupGen* on input 1^n needs ϑ bits of internal coin tosses. Then get ϑ bits from the beginning of r denoted as r_1 and use it as the internal coin tosses of *GroupGen*. Namely, $GroupGen(1^n, r_1) = (G, q, g)$.
2. Get $2\log q$ bits from the $(\vartheta+1)th$ bit of r denoted as r_2. Use r_2 as the internal coin tosses of choosing g_0 and g_1 from G. By the means mentioned in [6], we can get a subset instance description $\Lambda(X, L, W, R)$ based on G, where $X = G \times G$ and L is the subgroup of X generated by $(g_0, g_1) \in X$. A witness for $(u_0, u_1) \in L$ is $w \in Z_q$ such that $(u_0, u_1) = (g_0^w, g_1^w)$.
3. As the same way mentioned above, get $(4+2d)\log q$ bits from r denoted as r_3. Use r_3 as the internal coin tosses of sampling $k_0, k_1, \tilde{k}_0, \tilde{k}_1$, and $\hat{k}_{i,j}$ for $i \in [d], j \in \{0,1\}$ from Z_q.
4. The last step of **key generation** is computing the public key.
   $s = g_0^{k_0} \cdot g_1^{k_1}$
   $\tilde{s} = g_0^{\tilde{k}_0} \cdot g_1^{\tilde{k}_1}$
   $\hat{s}_i = g_0^{\hat{k}_{i,0}} \cdot g_1^{\hat{k}_{i,1}}$, for $i \in [d]$.

The public key is $(h, g_0, g_1, s, \tilde{s}, \hat{s}_1, \hat{s}_2, ..., \hat{s}_d)$.

The secret key is x.

– **Encryption**

On input a message $M \in G$, one does the following.

Choose $w \in Z_q$ at random.

Compute $y_0 = g_0^w$ and $y_1 = g_1^w$.

Compute $\pi = s^w$.

Compute $e = M \cdot \pi$.

Compute $\pi' = \tilde{s}^w \cdot \prod_{i=1}^{d} \hat{s}_i^{w\gamma_i}$, where $(\gamma_1, \gamma_2, ..., \gamma_d) = \Gamma(y_0, y_1, e)$.

The ciphertext is $(y_0, y_1, e, \hat{\pi})$.

– **Decryption**

On input a ciphertext $(y_0, y_1, e, \hat{\pi})$, one does the following.

Compute $Ext(x, h) = r$.

Using the same method mentioned above, get the same bits string r_3 from r as introduced in key generation. Use r_3 as the internal coin tosses of sampling $k_0, k_1, \tilde{k}_0, \tilde{k}_1$, and $\hat{k}_{i,j}$ for $i \in [d], j \in \{0,1\}$ from Z_q.

Compute $\hat{\pi}' = y_0^{\tilde{k}_0 + \sum_{i=1}^{d} \gamma_i \hat{k}_{i,0}} \cdot y_1^{\tilde{k}_1 + \sum_{i=1}^{d} \gamma_i \hat{k}_{i,1}}$, where $(\gamma_1, \gamma_2, ..., \gamma_d) = \Gamma(y_0, y_1, e)$.

Check whether $\hat{\pi} = \hat{\pi}'$, if not, output reject and halt.

Otherwise, compute $\pi = y_0^{k_0} \cdot y_1^{k_1}$.

Compute $M = e \cdot \pi^{-1}$, and output M.

It is evident from [6] that our above PKE scheme is an instantiation of the generic construction in §4.2. Additionally, it is weak λ-key-leakage chosen-ciphertext secure as long as $\tilde{H}_\infty(x|f(x), PK) \geq v - \lambda$. Thus, the leakage rate is $v(1 - o(1))$ where v is the length of the secret key.