

# Identity-based Encryption with Post-Challenge Auxiliary Inputs for Secure Cloud Applications and Sensor Networks

Tsz Hon Yuen<sup>1\*</sup>, Ye Zhang<sup>3</sup>, Siu Ming Yiu<sup>2</sup>, and Joseph K. Liu<sup>4</sup>

<sup>1</sup> Huawei, Singapore

yuen.tsz.hon@huawei.com

<sup>2</sup> The University of Hong Kong, Hong Kong

smyiu@cs.hku.hk

<sup>3</sup> Pennsylvania State University, USA

yxz169@cse.psu.edu

<sup>4</sup> Infocomm Security Department, Institute for Infocomm Research, Singapore

ksliu@i2r.a-star.edu.sg

**Abstract.** Identity-based encryption (IBE) is useful for providing end-to-end access control and data protection in many scenarios such as cloud applications and wireless sensor networks. However, there are some practical threats for the data owner or the sensor, who encrypts raw data; and the data user or the control centre, who decrypts the ciphertext and recovers the raw data.

In this paper, we tackle the open problem of proposing a leakage-resilience encryption model that can capture leakage from both the secret key owner (the data user or control centre) and the encryptor (the data owner or sensor), in the auxiliary input model. Existing models only allow the leakage of the secret key and do not allow adversaries to query more leakage information *after* seeing the challenge ciphertext of the security games. We solve this problem by defining the *post-challenge auxiliary input* model in which the family of leakage functions must be defined before the adversary is given the public key. The post-challenge query will return the leakage of the encryption randomness used by the encryptor. This model is able to capture a wider class of real-world attacks.

To realize our model, we propose a generic transformation from the auxiliary input model to our new post-challenge auxiliary input model for both public key encryption (PKE) and IBE. Furthermore, we extend Canetti *et al.*'s technique, that converts CPA-secure IBE to CCA-secure PKE, into the leakage-resilient setting.

## 1 Introduction

In autonomic provisioning of applications hosted in cloud server, all customer requirements and constraints must be fulfilled, as specified in Service Level Agree-

---

\* The research was mainly conducted when the first author was affiliated with the University of Hong Kong.

ment. Confidentiality, integrity and access control are important issues for security and privacy of such open infrastructure [10]. Access control is classified as one to the top 10 challenges in big data security by the Cloud Security Alliance (CSA) [6]. Sensitive data must be protected by using cryptographically secure algorithm and suitable access control. It is challenging to adapt unpredictable changes of access control policy in autonomic computing.

It was reported that a Google employee monitored some teenagers' communication records in Google Voice, stalked them by recovering their names and phone numbers [12]. Therefore, end-to-end access control is more resilient to attacks from the system administrator of the cloud server. Moreover, the data owner has more control on who is allowed to access by using end-to-end access control. The CSA suggested the use of identity-based encryption (IBE) as one of the possible cryptographic approach to enforce access control in big data applications [6]. In IBE, the data owner (encryptor) can encrypt a message by using the recipient's (decryptor) identity as his public key. The decryptor has to request a third party called Private Key Generator (PKG) to issue an identity-based secret key to him. By using this secret key, the decryptor recovers the original message. Access control is enforced by the data owner since he can choose the identity of the recipient of his choice.

The case of sensor networks is similar. After getting raw data by the sensor, it encrypts these sensitive information by using the identity of the control centre and sends the ciphertext back to it for further analysis. IBE eliminates the costly certificate verification process and thus it is preferred in sensor networks.

### 1.1 Practical Threats of Using IBE for Access Control

Although IBE can protect the users from malicious cloud server administrator, there are still a number of practical threats for both the encryptor and the decryptor.

**Side Channel Attacks to the Decryptor.** Real world attackers can obtain partial information about the secret key of the decryptor. Side-channel attacks explore the physical weakness of the implementation of cryptosystems. For example, the hamming weight of the secret key can be leaked by observing the running time of the decryption process, or the power consumption used.

**Weak Randomness Used by the Encryptor.** The randomness used in the encryption process may be leaked by poor implementation of pseudorandom number generator (PRNG). Argyros and Kiayias [2] outlined the flaws of PRNG in PHP. Lenstra *et al.* [13] inspected millions of public keys and found that some of the weak keys could be a result of poorly seeded PRNGs. Michaelis *et al.* [14] uncovered significant weaknesses of PRNG of some java runtime libraries, including Android. The NIST standard for pseudorandom number generation has fallen into dispute after the discovery of back door algorithm in [18] and received great attention through the Snowden disclosures [17]. These practical attacks demonstrate the potential weakness of the encryption randomness when using PRNG in practice. In big data applications, data are usually generated

by some devices with limited computational power. It is possible that the data are encrypted using such weak randomness from java runtime libraries. This situation is particular serious in the case of wireless sensors as they are usually exposed in the open air but contain only very limited computation power. Attackers may easily guess the randomness they are using for generating the ciphertext.

## 1.2 Motivation for Post-Challenge Auxiliary Inputs

We need to provide leakage-resilient protection for users of the cloud applications, including the encryptor and the decryptor. It motivates the new Post-Challenge Auxiliary Inputs model for IBE setting. We first review the background of leakage-resilient cryptography.

**Protecting the Decryptor: Leakage-Resilient Cryptography.** In modern cryptography, we use a security model to capture the abilities of a potential attacker (the adversary). For example, in the chosen-ciphertext attack (CCA) model for public key encryption (PKE), the adversary is allowed to ask for the decryption of arbitrary ciphertexts, except for the one that he intends to attack. This models the real-world scenario that the adversary may obtain some pairs of messages and ciphertexts from the secret key owner. Under a given model, a cryptographic scheme is said to be *proven 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, 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* on the physical implementation of cryptographic schemes, obtaining partial information of the secret key becomes feasible and relatively easier. 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, 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  $f$  such that the adversary should not be able to recover  $sk$  completely and to win the security game trivially. One approach is to restrict the output size of  $f$  to be at most  $\ell$  bits such that  $\ell$  must be less than  $|sk|$  [1]. Naor and Segev [16] considered the entropy of  $sk$  and required that the decrease in entropy of the  $sk$  is at most  $\ell$  bits upon observing  $f(sk)$ . Dodis *et al.* [8] 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  $f(sk)$ .

**Restriction of the Auxiliary Input Model.** 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 and IBE,

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  $\mathcal{A}$ <sup>5</sup>. But for most leakage-resilient PKE or IBE, the adversary  $\mathcal{A}$  can only specify and query the leakage function  $f(\text{sk})$  *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, 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 and IBE, 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.

**Protecting the Encryptor: Leakage-Resilient from the Encryptor’s Randomness.** Another direction for considering post-challenge leakage query is to model the leakage of encryptor (data owner). In the previous section, we showed the practical threats of using weak PRNG as the source of encryption randomness. This random value is 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  $(\text{Enc}, \text{Dec})$  a leakage-resilient PKE scheme in the auxiliary input model and one-time pad to form a new encryption scheme:

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

The randomness used in  $\text{Enc}'$  by the encryptor is  $P$  and the randomness in  $\text{Enc}$ . 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

<sup>5</sup> 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.

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.* [20] for leakage-resilient signatures in the auxiliary input model<sup>6</sup>. However, directly applying this idea to PKE, by simply allowing both pre-challenge and post-challenge leakages on  $\text{sk}$ , is not meaningful. Specifically, as the possible choice of leakage function  $f'$  is chosen before seeing the challenge ciphertext  $C^*$ , the post-challenge leakage  $f'(\text{sk})$  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 and IBE. 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 paper, we call our new model as the *post-challenge auxiliary input* model.

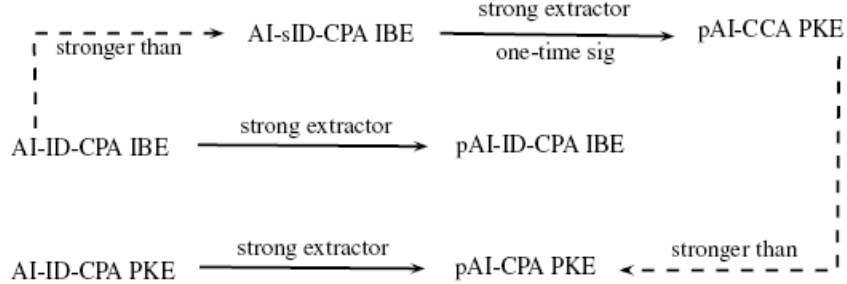
### 1.3 Our Contributions

In this paper, we propose the post-challenge auxiliary input model for public key and identity-based 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 and IBE 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.* [3] only allows randomness leakage before receiving the public key. Namiki *et al.* [15] 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 or IBE schemes consider the leakage of secret key and randomness at the same time. Therefore, our post-challenge 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  $\text{pk}$ . Specifically, we allow for  $g(\text{pk}, f(r))$  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.

<sup>6</sup> Yuen *et al.* [20] named their model as the selective auxiliary input model, due to similarity to the selective-ID model in identity-based encryption.

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. [7]) 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. [19]).

Furthermore, we extend the generic transformation for CPA-secure IBE to CCA-secure PKE by Canetti *et al.* [5] into the leakage-resilient setting. The original transformation by Canetti *et al.* [5] 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 §5.1). 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 Fig. 1.



**Fig. 1.** Our Contributions on Encryption

**Related Work.** Dodis *et al.* [8] introduced the model of *auxiliary inputs* leakage functions. PKE secure in the auxiliary input model was proposed in [7]. Signature schemes secure in the auxiliary input model were independently proposed by

Yuen *et al.* [20] and Faust *et al.* [9], 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 [16] that

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

Halevi and Lin [11] 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 [11] 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.* [4] showed that any non-committing encryption scheme is tolerant to leakage on both the secret key  $\text{sk}$  and encryption randomness  $r$  (together), such that leaking  $L$  bits on  $(\text{sk}, r)$  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 reveal *any bit* on the underlying encrypted message.

## 2 Security Model of Post-Challenge Auxiliary Inputs

We denote the security parameter by  $\lambda$ . We use the notation  $\text{neg}(\lambda)$  to refer to some negligible function of  $\lambda$ , and  $\text{poly}(\lambda)$  to refer to some polynomial function of  $\lambda$ .

We give the new post-challenge auxiliary input model for (probabilistic) public key encryption. Denote the message space as  $\mathcal{M}$ . A public-key encryption scheme  $\Pi$  consists of three Probabilistic Polynomial Time (PPT) algorithms:

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

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

As introduced in §1.2, 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 §1.2. Since  $\mathcal{A}$  is a PPT algorithm, we consider that  $m := |\mathcal{F}_0|$  is polynomial in the security parameter  $\lambda$ .

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  $\epsilon_r$ .  $\mathcal{A}$  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  $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$  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  $\lambda$ .

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  $\lambda$ .
2.  $\mathcal{C}$  runs  $(\text{pk}, \text{sk}) \leftarrow \text{Gen}(1^\lambda)$  and outputs  $\text{pk}$  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  $f_i(\text{sk}, \text{pk})$  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  $C^* \leftarrow \text{Enc}(\text{pk}, m_b; r')$  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  $\text{Dec}(\text{sk}, C)$  to  $\mathcal{A}$ .
6.  $\mathcal{A}$  outputs its guess  $b' \in \{0, 1\}$ . The advantage of  $\mathcal{A}$  is  $\text{Adv}_{\mathcal{A}}^{\text{pAI-CCA}}(\Pi) = |\Pr[b = b'] - \frac{1}{2}|$ .

Note that in the pre-challenge leakage stage,  $\mathcal{A}$  may choose  $f_i(\text{sk}, \text{pk})$  to encode  $\text{Dec}(\text{sk}, \cdot)$  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  $g(\text{pk}, m_0, f(r^*))$  and  $g(\text{pk}, m_1, f(r^*))$ , where  $f$  is any hard-to-invert function. Since the adversary knows  $\text{pk}$ ,  $m_0$  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 [8] (Refer to [8] for the definition of such family). In this paper, 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  $\mathcal{H}_{\text{ow}}$ , which is extended from the definition in [8]:

- Let  $\mathcal{H}_{\text{ow}}(\epsilon_r)$  be the class of all polynomial-time computable functions  $h : \{0, 1\}^{|r'|} \rightarrow \{0, 1\}^*$ , such that given  $h(r')$  (for a randomly generated  $r'$ ), no PPT algorithm can find  $r'$  with probability greater than  $\epsilon_r$ <sup>7</sup>. The function  $h(r')$  can be viewed as a composition of  $q \in \mathbb{N}^+$  functions:  $h(r') = (h_1(r'), \dots, h_q(r'))$ . Therefore  $\{h_1, \dots, h_q\} \in \mathcal{H}_{\text{ow}}(\epsilon_r)$ .

Also, we consider  $\mathcal{F}_s$  as a family of one-way function  $\mathcal{H}_{\text{pk-ow}}$ :

- Let  $\mathcal{H}_{\text{pk-ow}}(\epsilon_s)$  be the class of all polynomial-time computable functions  $h : \{0, 1\}^{|\text{sk}|+|\text{pk}|} \rightarrow \{0, 1\}^*$ , such that given  $(\text{pk}, h(\text{sk}, \text{pk}))$  (for a randomly generated  $(\text{sk}, \text{pk})$ ), no PPT algorithm can find  $\text{sk}$  with probability greater than  $\epsilon_s$ <sup>8</sup>. The function  $h(\text{sk}, \text{pk})$  can be viewed as a composition of  $q'$  functions:  $h(\text{sk}, \text{pk}) = (h_1(\text{sk}, \text{pk}), \dots, h_{q'}(\text{sk}, \text{pk}))$ . Therefore  $\{h_1, \dots, h_{q'}\} \in \mathcal{H}_{\text{pk-ow}}(\epsilon_s)$ .

**Definition 1.** *We say that  $\Pi$  is pAI-CCA secure with respect to the families  $(\mathcal{H}_{\text{pk-ow}}(\epsilon_s), \mathcal{H}_{\text{ow}}(\epsilon_r))$  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 [8].

We also define the security model for identity-based encryption in the full version of the paper [21].

### 3 Strong Extractor with Hard-to-invert Auxiliary Inputs

**Definition 2 (( $(\epsilon, \delta)$ -Strong extractor with auxiliary inputs).** *Let  $\text{Ext} : \{0, 1\}^{l_1} \times \{0, 1\}^{l_2} \rightarrow \{0, 1\}^{m'}$ , where  $l_1, l_2$  and  $m'$  are polynomial in  $\lambda$ .  $\text{Ext}$  is said to be a  $(\epsilon, \delta)$ -strong extractor with auxiliary inputs, if for every PPT adversary  $\mathcal{A}$ , and for all pairs  $(x, f)$  such that  $x \in \{0, 1\}^{l_2}$  and  $f \in \mathcal{H}_{\text{ow}}(\epsilon)$ , we have:*

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

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

<sup>7</sup> Otherwise, for example,  $\mathcal{A}$  can choose an identity mapping  $f$ . Then,  $\mathcal{A}$  can learn  $r' = f(r')$  and test if  $C^* = \text{Enc}(\text{pk}, m_0^*; r')$  to determine  $b$  and win the game.

<sup>8</sup> 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 [8], no PPT algorithm will output  $\text{sk}$  with  $\epsilon_s$  probability given  $f_i, \text{pk}$ , as  $\text{pk}$  leaks some information about  $\text{sk}$ . Therefore, we also define that no PPT algorithm will output  $r'$  with  $\epsilon_r$  probability given  $f'_i, C^*, \text{pk}, m_0^*, m_1^*$ . We omit these extra input parameters for simplicity in the rest of the paper.

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

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

*Proof.* Suppose on the contrary,  $x$  can be recovered with probability  $2\delta$  when knowing  $r, f(x)$  and  $\text{Ext}(r, x)$ . However by the definition of strong extractor, fix any auxiliary-input function  $f \in \mathcal{H}_{\text{ow}}(\epsilon)$ ,  $\langle r, f(x), \text{Ext}(r, x) \rangle$  is  $\delta$ -indistinguishable with  $\langle r, f(x), u \rangle$ . It leads to a contradiction, since if  $x$  can be recovered with probability  $2\delta$ , the attacker of  $\text{Ext}$  can compare that: (1) if  $f(x)$  value is correct, then it receives  $\text{Ext}(r, x)$ ; (2) else it receives  $u$  instead. It breaks the strong extractor with probability  $\delta$ , which is a contradiction.  $\square$

Interestingly, we find that a  $(\epsilon, \delta)$ -strong extractor with auxiliary inputs can be constructed from the modified Goldreich-Levin theorem from [7]. Denote  $\langle r, x \rangle = \sum_{i=1}^l r_i x_i$  as the inner product of  $x = (x_1, \dots, x_l)$  and  $r = (r_1, \dots, r_l)$ .

**Theorem 1 ([7]).** *Let  $q$  be a prime, and let  $\bar{H}$  be an arbitrary subset of  $GF(q)$ . Let  $f : \bar{H}^{\bar{n}} \rightarrow \{0, 1\}^*$  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  $t' = t \cdot \text{poly}(\bar{n}, |\bar{H}|, \frac{1}{\delta})$  such that  $\Pr[\mathcal{A}(y) = s] \geq \frac{\delta^3}{512\bar{n}q^2}$ .*

Now we are ready to show that strong extractor with  $\epsilon$ -hard-to-invert auxiliary inputs can be instantiated using inner product.

**Theorem 2.** *Let  $\lambda$  be the security parameter. Let  $x$  be chosen uniformly random from  $\{0, 1\}^{l(\lambda)}$  where  $l(\lambda) = \text{poly}(\lambda)$ . Similarly, we choose  $r$  uniformly random from  $GF(q)^{l(\lambda)}$  and  $u$  uniformly random from  $GF(q)$ . Then, given  $f \in \mathcal{H}_{\text{ow}}(\epsilon)$ , 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  $t = \text{poly}_1(\lambda)$  with probability  $\epsilon'$ . Then, there exists an inverter  $\mathcal{A}$  that runs in time  $t \cdot \text{poly}(l(\lambda), 2, \frac{1}{\epsilon}) = \text{poly}'(\lambda)$  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  $f \in \mathcal{H}_{\text{ow}}(\epsilon)$ .  $\square$

## 4 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., [7]) and a *strong extractor with  $\epsilon$ -hard-to-invert auxiliary inputs leakage*.

### 4.1 Construction of pAI-CPA Secure PKE

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

1.  $\text{Gen}(1^\lambda)$ : It runs  $(\text{pk}, \text{sk}) \leftarrow \text{Gen}'(1^\lambda)$  and chooses  $r$  uniformly random from  $\{0, 1\}^{l_1}$ . Then, we set the public key  $\text{PK} = (\text{pk}, r)$  and the secret key  $\text{SK} = \text{sk}$ .
2.  $\text{Enc}(\text{PK}, M)$ : It picks  $x$  uniformly random from  $\{0, 1\}^{l_2}$ . Then, it computes  $y = \text{Ext}(r, x)$ . The ciphertext is  $c = \text{Enc}'(\text{pk}, M; y)$ .
3.  $\text{Dec}(\text{SK}, c)$ : It returns  $\text{Dec}'(\text{sk}, c)$ .

**Theorem 3.** *If  $\Pi'$  is an AI-CPA secure encryption with respect to family  $\mathcal{H}_{\text{pk-ow}}(\epsilon_s)$  and  $\text{Ext}$  is a  $(\epsilon_r, \text{neg}(\lambda))$ -strong extractor with auxiliary inputs, then  $\Pi$  is pAI-CPA secure with respect to families  $(\mathcal{H}_{\text{pk-ow}}(\epsilon_s), \mathcal{H}_{\text{ow}}(\epsilon_r))$ .*

*Proof.* Denote the randomness used in the challenge ciphertext as  $x^*$ . Let  $\text{Game}_0$  be the pAI-CPA security game with  $\Pi$  scheme.  $\text{Game}_1$  is the same as  $\text{Game}_0$  except that when encrypting the challenge ciphertext  $c = \text{Enc}'(\text{pk}, m_b; y)$ , we replace  $y = \text{Ext}(r, x^*)$  with  $y'$  which is chosen uniformly at random in  $\{0, 1\}^{m'}$ . The leakage oracle outputs  $f_i(x^*)$  for both games.

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

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

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

The simulator  $\mathcal{S}$  is given  $(r, f_1(x^*), f_2(x^*), \dots, f_q(x^*), T)$  where  $T$  is either  $T_0 = \langle r, x^* \rangle$  or  $T_1 = u$  which is a random number as in Definition 2. Given  $f_1(x^*), \dots, f_q(x^*)$ , no PPT adversary can recover  $x^*$  with probability greater than  $\epsilon_r$  by the definition of  $\mathcal{H}_{\text{ow}}(\epsilon_r)$ . Then, the simulator generates  $(\text{pk}, \text{sk}) \leftarrow \text{Gen}'(1^\lambda)$ . It sets  $\text{SK} = \text{sk}$  and gives the adversary  $\text{PK} = (\text{pk}, r)$ . The simulator can answer pre-challenge leakage oracle as it has  $\text{PK}$  and  $\text{SK}$ . 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  $C^* = \text{Enc}(\text{pk}, m_b; T)$  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  $\text{Game}_0$  and  $\text{Game}_1$  is  $\epsilon_A$ , then

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

which is non-negligible if  $\epsilon_A$  is non-negligible. It contradicts the definition of strong extractor in Definition 2. Therefore, no PPT adversary can distinguish  $\text{Game}_0$  from  $\text{Game}_1$  with non-negligible probability.

Next, we want to show that

$$Adv_{\mathcal{A}}^{\text{Game}_1}(\Pi) = \text{neg}(\lambda).$$

Note that the challenge ciphertext now is  $c = \text{Enc}'(\text{pk}, M; y')$  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  $\text{Game}_1$  is the same as the AI-CPA game with  $\Pi'$ . As  $\Pi$  is based on  $\Pi'$  which is AI-CPA secure, we have that  $Adv_{\mathcal{A}}^{\text{Game}_1}(\Pi)$  is negligible.  $\square$

**Extension to IBE.** We can use the same technique to construct pAI-ID-CPA secure IBE. Let  $\Sigma' = (\text{Setup}', \text{Extract}', \text{Enc}', \text{Dec}')$  be an AI-ID-CPA secure IBE (e.g. [19]) where the encryption randomness is in  $\{0, 1\}^{m'}$ ,  $\text{Ext} : \{0, 1\}^{l_1} \times \{0, 1\}^{l_2} \rightarrow \{0, 1\}^{m'}$  is a  $(\epsilon_r, \text{neg}(\lambda))$ -strong extractor with auxiliary inputs, then construct a pAI-ID-CPA secure IBE scheme  $\Sigma$  as follows.

1.  $\text{Setup}(1^\lambda)$ : It runs  $(\text{mpk}, \text{msk}) \leftarrow \text{Setup}'(1^\lambda)$  and chooses  $r$  uniformly random from  $\{0, 1\}^{l_1}$ . Then, we set the master public key  $\text{MPK} = (\text{mpk}, r)$  and the master secret key  $\text{MSK} = \text{msk}$ .
2.  $\text{Extract}(\text{MSK}, \text{ID})$ : It returns  $\text{sk}_{\text{ID}} \leftarrow \text{Extract}'(\text{MSK}, \text{ID})$ .
3.  $\text{Enc}(\text{MPK}, \text{ID}, M)$ : It chooses  $x$  uniformly random from  $\{0, 1\}^{l_2}$ . Then, it computes  $y = \text{Ext}(r, x)$ . The ciphertext is  $c = \text{Enc}'(\text{mpk}, \text{ID}, M; y)$ .
4.  $\text{Dec}(\text{sk}_{\text{ID}}, c)$ : It returns  $\text{Dec}'(\text{sk}_{\text{ID}}, c)$ .

**Theorem 4.** *If  $\Sigma'$  is an AI-ID-CPA secure IBE with respect to family  $\mathcal{H}_{\text{pk-ow}}(\epsilon_s)$  and  $\text{Ext}$  is a  $(\epsilon_r, \text{neg}(\lambda))$ -strong extractor with auxiliary inputs, then  $\Sigma$  is pAI-ID-CPA secure with respect to families  $(\mathcal{H}_{\text{pk-ow}}(\epsilon_s), \mathcal{H}_{\text{ow}}(\epsilon_r))$ .*

The proof is similar to the proof of Theorem 3 and hence is omitted.

**Corollary 1.** *Instantiating with the strong extractor construction in §3 and the identity-based encryption scheme in [19], the identity-based encryption construction  $\Sigma'$  is pAI-ID-CPA secure.*

## 5 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.* [5] 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 [5], 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  $\text{Extract}(ID^*)$ . However, we cannot apply the technique in [5] directly.

### 5.1 Intuition

Let  $(\text{Gen}_s, \text{Sign}, \text{Verify})$  be a strong one-time signature scheme. Let  $(\text{Setup}', \text{Extract}', \text{Enc}', \text{Dec}')$  be an auxiliary-inputs CPA secure IBE scheme (refer to the definition in the full version of the paper [21], by dropping the post-challenge query). The construction directly following Canetti *et al.*'s transformation [5] is as follows.

1.  $\text{Gen}(1^\lambda)$ : Run  $(\text{mpk}, \text{msk}) \leftarrow \text{Setup}'(1^\lambda)$ . Set the public key  $\text{pk} = \text{mpk}$  and the secret key  $\text{sk} = \text{msk}$ .
2.  $\text{Enc}(\text{pk}, M)$ : Run  $(\text{vk}, \text{sk}_s) \leftarrow \text{Gen}_s(1^\lambda)$ . Calculate  $c \leftarrow \text{Enc}'(\text{pk}, \text{vk}, M)$  and  $\sigma \leftarrow \text{Sign}(\text{sk}_s, c)$ . Then, the ciphertext is  $C = (c, \sigma, \text{vk})$ .
3.  $\text{Dec}(\text{sk}, C)$ : First, test  $\text{Verify}(\text{vk}, c, \sigma) \stackrel{?}{=} 1$ . If it is "1", compute  $\text{sk}_{\text{vk}} = \text{Extract}'(\text{sk}, \text{vk})$  and return  $\text{Dec}'(\text{sk}_{\text{vk}}, c)$ . Otherwise, return  $\perp$ .

**Problems in the Post-Challenge Auxiliary Input Model.** At first glance it seems that Canetti *et al.*'s transformation [5] 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  $\text{Gen}_s$ ,  $\text{Sign}$  and  $\text{Enc}'$ , denoted as  $r_{\text{sig}_1}$ ,  $r_{\text{sig}_2}$  and  $r_{\text{enc}}$  respectively. Specifically, we have  $(\text{vk}, \text{sk}_s) \leftarrow \text{Gen}_s(1^\lambda; r_{\text{sig}_1})$ ,  $\sigma \leftarrow \text{Sign}(\text{sk}_s, c; r_{\text{sig}_2})$  and  $c \leftarrow \text{Enc}'(\text{mpk}, \text{vk}, m_b; r_{\text{enc}})$ .

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  $C^* = (c^*, \sigma^*, \text{vk}^*)$ ,  $\mathcal{A}$  can ask the leakage  $f(r')$  where  $r' = (r_{\text{enc}}, r_{\text{sig}_1}, r_{\text{sig}_2})$  is the randomness used to produce  $C^*$ . To some extreme,  $\mathcal{A}$  may ask:

- $f_1(r') = r_{\text{enc}}$ , such that  $f_1$  is still hard-to-invert upon  $r'$ . In this case,  $\mathcal{A}$  can test  $c^* \stackrel{?}{=} \text{Enc}'(\text{mpk}, \text{vk}, m_0; r_{\text{enc}})$  to win the pAI-CCA game; or

- $f_2(r') = (r_{\text{sig}_1}, r_{\text{sig}_2})$ , such that  $f_2$  is still hard-to-invert upon  $r'$ . In this case, given  $r_{\text{sig}_1}$ ,  $\mathcal{A}$  can generate  $(\text{vk}, \text{sk}_s) = \text{Gen}_s(1^\lambda; r_{\text{sig}_1})$  which causes  $\Pr[\text{Forge}]$  defined in [5] 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 [5] fail in our model.

**Our Solution:** We set both  $r_{\text{sig}_1}, r_{\text{sig}_2}$  and  $r_{\text{enc}}$  are generated from the same source of randomness  $x \in \{0, 1\}^{l_2}$ . Suppose  $r_{\text{sig}_1} || r_{\text{sig}_2}$  and  $r_{\text{enc}}$  are bit-strings of length  $n'$ . Suppose  $\text{Ext} : \{0, 1\}^{l_1} \times \{0, 1\}^{l_2} \rightarrow \{0, 1\}^{n'}$  is a strong extractor with  $\epsilon_r$ -hard-to-invert 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  $\text{pk}$ . Then the randomness used in the IBE and the one-time signature can be calculated by  $r_{\text{enc}} = \text{Ext}(r_1, x)$  and  $(r_{\text{sig}_1} || r_{\text{sig}_2}) = \text{Ext}(r_2, x)$  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  $\Pi'$ . 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  $\Pi'$ . We solve this problem by proving that it is indistinguishable by replacing  $r_{\text{enc}}^* = \text{Ext}(r_1, x^*)$  and  $r_{\text{sig}_1}^* || r_{\text{sig}_2}^* = \text{Ext}(r_2, x^*)$  with random numbers. Therefore, the post-challenge leakages on  $x^*$  will be independent with  $r_{\text{enc}}^*$  and  $r_{\text{sig}_1}^* || r_{\text{sig}_2}^*$  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  $r_{\text{sig}_1}^* || r_{\text{sig}_2}^*$  with a random number, the simulator needs to compute  $r_{\text{enc}}^* = \text{Ext}(r_1, x^*)$ . One way to solve it is to include  $\text{Ext}(r_1, x^*)$  as a post-challenge leakage query in the pAI-CCA game. As we will see later (by Lemma 1), including  $\text{Ext}(r_1, x^*)$  in leakage queries is still  $\text{neg}(\lambda)$ -hard-to-invert.

Following [5], 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*.

## 5.2 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  $\Pi' = (\text{Setup}', \text{Extract}', \text{Enc}', \text{Dec}')$ , a strong one-time signature scheme  $\Pi_s = (\text{Gen}_s, \text{Sign}, \text{Verify})$  and a strong extractor with  $\epsilon_r$ -hard-to-invert auxiliary input  $\text{Ext} : \{0, 1\}^{l_1} \times \{0, 1\}^{l_2} \rightarrow \{0, 1\}^{n'}$ , where the size of  $r_{\text{enc}}$  and  $r_{\text{sig}_1} || r_{\text{sig}_2}$  are both  $\{0, 1\}^{n'}$ ; and the verification key space of  $\Pi_s$  is the same as the identity space of  $\Pi'$ . We construct a PKE scheme  $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$  as follows.

1.  $\text{Gen}(1^\lambda)$ : Run  $(\text{mpk}, \text{msk}) \leftarrow \text{Setup}'(1^\lambda)$ . Choose  $r_1, r_2$  uniformly random from  $\{0, 1\}^{l_1}$ . Set the public key  $\text{pk} = (\text{mpk}, r_1, r_2)$  and the secret key  $\text{sk} = \text{msk}$ .
2.  $\text{Enc}(\text{pk}, m)$ : Randomly sample  $x \in \{0, 1\}^{l_2}$ , calculate  $r_{\text{enc}} = \text{Ext}_1(r_1, x)$  and  $r_{\text{sig}_1} || r_{\text{sig}_2} = \text{Ext}_2(r_2, x)$ . Run  $(\text{vk}, \text{sk}_s) = \text{Gen}_s(1^\lambda; r_{\text{sig}_1})$ . Let  $c = \text{Enc}'(\text{pk}, \text{vk}, m; r_{\text{enc}})$ ;  $\sigma = \text{Sign}(\text{sk}_s, c; r_{\text{sig}_2})$ . Then, the ciphertext is  $C = (c, \sigma, \text{vk})$ .
3.  $\text{Dec}(\text{sk}, C)$ : First, test  $\text{Verify}(\text{vk}, c, \sigma) \stackrel{?}{=} 1$ . If it is “1”, compute  $\text{sk}_{\text{vk}} = \text{Extract}(\text{sk}, \text{vk})$  and return  $\text{Dec}'(\text{sk}_{\text{vk}}, c)$ . Otherwise, return  $\perp$ .

**Theorem 5.** *Assuming that  $\Pi'$  is a AI-sID-CPA secure IBE scheme with respect to family  $\mathcal{H}_{\text{pk-ow}}(\epsilon_s)$ ,  $\Pi_s$  is a strong one-time signature, and  $\text{Ext}_1$  is  $(\epsilon_r, \text{neg}_1)$ -strong extractor with auxiliary inputs and  $\text{Ext}_2$  is  $(2\text{neg}_1, \text{neg}_2)$ -strong extractor with auxiliary inputs, then there exists a PKE scheme  $\Pi$  which is pAI-CCA secure with respect to families  $(\mathcal{H}_{\text{pk-ow}}(\epsilon_s), \mathcal{H}_{\text{ow}}(\epsilon_r))$ .*

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

**Lemma 2.** *For any PPT adversary  $\mathcal{A}$ ,  $\text{Game}_0$  is indistinguishable from  $\text{Game}_1$  if  $\text{Ext}_1$  is  $(\epsilon_r, \text{neg}_1)$ -strong extractor with auxiliary inputs and  $\text{Ext}_2$  is  $(2\text{neg}_1, \text{neg}_2)$ -strong extractor with auxiliary inputs.*

**Lemma 3.** *For any PPT adversary  $\mathcal{A}$ ,  $\text{Game}_1$  is indistinguishable from  $\text{Game}_2$  if  $\text{Ext}_1$  is  $(\epsilon_r, \text{neg}_1)$ -strong extractor with auxiliary inputs.*

**Lemma 4.** *For any PPT adversary  $\mathcal{A}$ , the advantage in  $\text{Game}_2$  is negligible if  $\Pi'$  is a AI-sID-CPA secure IBE scheme with respect to family  $\mathcal{H}_{\text{pk-ow}}(\epsilon_s)$  and  $\Pi_s$  is a strong one-time signature.*

□

*Proof (Lemma 2).* Let  $\text{Adv}_{\mathcal{A}}^{\text{Game}_i}(\Pi)$  be the advantage that the adversary  $\mathcal{A}$  wins in  $\text{Game}_i$  with  $\Pi$  scheme. Now, we need to show for any PPT adversary  $\mathcal{A}$ :

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

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

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

Then, the simulator generates  $(\text{mpk}, \text{msk}) \leftarrow \text{Setup}'(1^\lambda)$ . It sets  $\text{sk} = \text{msk}$  and gives the adversary  $\text{pk} = (\text{mpk}, r_1, r_2)$ . The simulator can answer pre-challenge leakage oracle as it has  $\text{pk}$  and  $\text{sk}$ . The adversary submits two messages  $m_0$  and  $m_1$  to the simulator where the simulator flips a coin  $b$ . It sets  $r_{\text{sig}_1} || r_{\text{sig}_2} = T$ , runs  $(\text{vk}, \text{sk}_s) \leftarrow \text{Gen}_s(1^\lambda; r_{\text{sig}_1})$ ,  $c = \text{Enc}'(\text{pk}, \text{vk}, m_b; f_{q+1}(x^*))$  and  $\sigma = \text{Sign}(\text{sk}_s, c; r_{\text{sig}_2})$ . It returns the challenge ciphertext  $C^* = (c, \sigma, \text{vk})$  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  $\text{Game}_0$  and  $\text{Game}_1$  is  $\epsilon_A$ , then

$$\text{Adv}_S = \frac{1}{2}(|\Pr[b = b'|T_1] - \Pr[b = b'|T_0]|) \geq \frac{\epsilon_A}{2}.$$

which is non-negligible if  $\epsilon_A$  is non-negligible. It contradicts the fact that  $\text{Ext}_2$  is  $(2\text{neg}_1, \text{neg}_2)$ -strong extractor with auxiliary inputs. Therefore, no PPT adversary can distinguish  $\text{Game}_0$  from  $\text{Game}_1$  with non-negligible probability.

Finally, we need to show that including  $\text{Ext}_1(r_1, \cdot)$  is also  $2\text{neg}_1(\lambda)$ -hard-to-invert, provided that  $\text{Ext}_1$  is a  $(\epsilon_r, \text{neg}_1)$ -strong extractor with auxiliary inputs. This follows directly from Lemma 1 if we set  $f = (f_1(x^*), \dots, f_q(x^*)) \in \mathcal{H}_{ow}(\epsilon_r)$ .  $\square$

*Proof (Lemma 3).* The post-challenge query functions  $(f_1, \dots, f_q) \in \mathcal{F}_0$  are  $\epsilon_r$ -hard-to-invert by definition. Fix any auxiliary-input function  $f_1, \dots, f_q$ ,  $\langle r_1, f_1(x^*), \dots, f_q(x^*), \text{Ext}(r_1, x^*) \rangle$  is indistinguishable with  $\langle r_1, f_1(x^*), \dots, f_q(x^*), u \rangle$  where  $u$  is randomly chosen from  $\{0, 1\}^{n'}$ , by the definition of strong extractor. Hence  $\text{Game}_1$  is indistinguishable from  $\text{Game}_2$ . The reduction is similar to the previous proof.  $\square$

*Proof (Lemma 4).* Let  $\mathcal{A}$  be an adversary to  $\Pi$  on  $\text{Game}_2$  and we construct an AI-sID-CPA adversary  $\mathcal{A}'$  to  $\Pi'$  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  $\text{Game}_2$  to  $\mathcal{A}'$ .  $\mathcal{A}'$  picks  $r_{\text{sig}_1} || r_{\text{sig}_2}$  uniformly random from  $\{0, 1\}^{n'}$  and computes  $(\text{vk}^*, \text{sk}_s^*) = \text{Gen}_s(1^\lambda; r_{\text{sig}_1})$ .  $\mathcal{A}'$  submits the challenge identity  $\text{vk}^*$  to the AI-sID-CPA challenger  $\mathcal{C}$ , and  $\mathcal{C}$  returns  $\text{mpk}$  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  $\text{pk} = (\text{mpk}, r_1, r_2)$  to  $\mathcal{A}$ .

In the pre-challenge query phase,  $\mathcal{A}$  can adaptively query  $f_i(\text{pk}, \text{msk})$ .  $\mathcal{A}'$  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  $c^* = \text{Enc}'(\text{mpk}, \text{vk}^*, m_b; r_{\text{enc}})$  to  $\mathcal{A}'$  for some random bit  $b$  and randomness  $r_{\text{enc}}$ . Then  $\mathcal{A}'$  computes  $\sigma^* = \text{Sign}(\text{sk}_s^*, c^*; r_{\text{sig}_2})$ .  $\mathcal{A}'$  sends  $C^* = (c^*, \sigma^*, \text{vk}^*)$  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  $\text{DEC}(c, \sigma, \text{vk})$ .  $\mathcal{A}'$  returns  $\perp$  if  $\text{Verify}(\text{vk}, c, \sigma) \neq 1$ . Otherwise, there are two cases. If  $\text{vk} = \text{vk}^*$ , 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,  $\text{vk} \neq \text{vk}^*$ ,  $\mathcal{A}'$  asks the extraction oracle  $\mathcal{EO}(\text{vk})$  to  $\mathcal{C}$  and uses  $\text{sk}_{\text{vk}}$  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  $\text{Game}_2$  with a non-negligible probability, then  $\mathcal{A}'$  will win the AI-sID-CPA game also with a non-negligible probability, which contradicts that  $\Pi'$  is AI-sID-CPA secure.

To show that the probability that  $\mathcal{A}$  asks for the decryption of a valid ciphertext with identity  $\text{vk}^*$  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  $\text{vk}^*$  to  $\mathcal{B}$ .  $\mathcal{B}$  runs  $(\text{mpk}, \text{msk}) \leftarrow \text{Setup}'(1^\lambda)$  and picks  $r_1$  and  $r_2$  which are independent and uniformly chosen from  $\{0, 1\}^{l_1}$ .  $\mathcal{B}$  returns  $\text{pk} = (\text{mpk}, r_1, r_2)$  to  $\mathcal{A}$ .

In the pre-challenge query phase,  $\mathcal{A}$  can adaptively query  $f_i(\text{pk}, \text{msk})$  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  $r_{\text{enc}}$  uniformly random from  $\{0, 1\}^{n'}$ .  $\mathcal{B}$  picks a random bit  $b$  and calculates  $c^* = \text{Enc}'(\text{mpk}, \text{vk}^*, m_b; r_{\text{enc}})$ . Then  $\mathcal{B}$  asks  $\mathcal{C}'$  to sign on  $c^*$  and obtains the signature  $\sigma^*$ .  $\mathcal{B}$  gives the challenge ciphertext  $C^* = (c^*, \sigma^*, \text{vk}^*)$  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  $\text{vk} \neq \text{vk}^*$  can be answered by using  $\text{msk}$ . However, if  $\mathcal{A}$  asks for the decryption of a valid ciphertext  $(c, \sigma, \text{vk}^*)$  that is not identical to  $(c^*, \sigma^*, \text{vk}^*)$ ,  $\mathcal{B}$  returns  $(c, \sigma)$  to  $\mathcal{C}'$ . Therefore, the probability that  $\mathcal{A}$  can output a forged signature is negligible provided that  $\Pi_s$  is a strong one-time signature, which completes the proof.  $\square$

## Acknowledgements

We thank anonymous reviewers for their helpful comments. Ye Zhang was supported by NSF award #0747294. Siu Ming Yiu was partially supported by an NSFC/RGC Joint Research Scheme N\_HKU 729/13. Joseph K. Liu was supported by A\*STAR project SPARK-1224104047.

## References

1. A. Akavia, S. Goldwasser, and V. Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In O. Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 474–495. Springer, 2009.
2. G. Argyros and A. Kiayias. I forgot your password: randomness attacks against php applications. In *USENIX, Security'12*, page 6. USENIX Association, 2012.
3. M. Bellare, Z. Brakerski, M. Naor, T. Ristenpart, G. Segev, H. Shacham, and S. Yilek. Hedged public-key encryption: How to protect against bad randomness. In M. Matsui, editor, *ASIACRYPT*, volume 5912 of *LNCS*, pages 232–249. Springer, 2009.

4. N. Bitansky, R. Canetti, and S. Halevi. Leakage-tolerant interactive protocols. In R. Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 266–284. Springer, 2012.
5. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–222. Springer, 2004.
6. Cloud Security Alliance. Expanded top ten big data security and privacy challenges, 2013.
7. Y. Dodis, S. Goldwasser, Y. T. Kalai, C. Peikert, and V. Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In D. Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 361–381. Springer, 2010.
8. Y. Dodis, Y. T. Kalai, and S. Lovett. On cryptography with auxiliary input. In M. Mitzenmacher, editor, *STOC 2009*, pages 621–630. ACM, 2009.
9. S. Faust, C. Hazay, J. B. Nielsen, P. S. Nordholt, and A. Zottarel. Signature schemes secure against hard-to-invert leakage. In X. Wang and K. Sako, editors, *ASIACRYPT*, volume 7658 of *LNCS*, pages 98–115. Springer, 2012.
10. C. Franke and P. Robinson. Autonomic provisioning of hosted applications with level of isolation terms. *2010 Seventh IEEE International Conference and Workshops on Engineering of Autonomic and Autonomous Systems*, 0:131–142, 2008.
11. S. Halevi and H. Lin. After-the-fact leakage in public-key encryption. In Y. Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 107–124. Springer, 2011.
12. A. Hough. Google engineer fired for privacy breach after “staking and harassing teenagers”. The Telegraph. Sept 15, 2010.
13. A. K. Lenstra, J. P. Hughes, M. Augier, J. W. Bos, T. Kleinjung, and C. Wachter. Public keys. In R. Safavi-Naini and R. Canetti, editors, *CRYPTO*, volume 7417 of *LNCS*, pages 626–642. Springer, 2012.
14. K. Michaelis, C. Meyer, and J. Schwenk. Randomly failed! the state of randomness in current java implementations. In E. Dawson, editor, *CT-RSA 2013*, volume 7779 of *LNCS*, pages 129–144. Springer, 2013.
15. H. Namiki, K. Tanaka, and K. Yasunaga. Randomness leakage in the kem/dem framework. In X. Boyen and X. Chen, editors, *ProvSec*, volume 6980 of *LNCS*, pages 309–323. Springer, 2011.
16. M. Naor and G. Segev. Public-key cryptosystems resilient to key leakage. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 18–35. Springer, 2009.
17. N. Perlroth, J. Larson, and S. Shane. N.S.A. able to foil basic safeguards of privacy on web. New York Times, September 5, 2013. <http://www.nytimes.com/2013/09/06/us/nsa-foils-much-internet-encryption.html>.
18. D. Shumow and N. Ferguson. On the possibility of a back door in the NIST SP800-90 dual ec prng. <http://rump2007.cr.yp.to/15-shumow.pdf>.
19. T. H. Yuen, S. S. M. Chow, Y. Zhang, and S. M. Yiu. Identity-based encryption resilient to continual auxiliary leakage. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT*, volume 7237 of *LNCS*, pages 117–134. Springer, 2012.
20. T. H. Yuen, S. M. Yiu, and L. C. K. Hui. Fully leakage-resilient signatures with auxiliary inputs. In W. Susilo, Y. Mu, and J. Seberry, editors, *ACISP 2012*, volume 7372 of *LNCS*, pages 294–307. Springer, 2012.
21. T. H. Yuen, Y. Zhang, S.-M. Yiu, and J. K. Liu. Encryption schemes with post-challenge auxiliary inputs. Cryptology ePrint Archive, Report 2013/323, 2013. <http://eprint.iacr.org/>.