Encryption Schemes with Post-Challenge Auxiliary Inputs

Tsz Hon Yuen[1], Ye Zhang[2], and Siu Ming Yiu[1]

[1] The University of Hong Kong, Hong Kong
{thyuen,smyiu}@cs.hku.hk
[2] Pennsylvania State University, USA
yxz169@cse.psu.edu

Abstract. 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 and the encryptor, in the auxiliary input model. Existing models usually do not allow adversaries to query more leakage information *after* seeing the challenge ciphertext of the security games. On one hand, side-channel attacks on the random factor (selected by the encryptor) are already shown to be feasible. Leakage from the encryptor should not be overlooked. On the other hand, the technical challenge for allowing queries from the adversary after he sees the ciphertext is to avoid a trivial attack to the system since he can then embed the decryption function as the leakage function (note that we consider the auxiliary input model in which the leakage is modeled as computationally hard-to-invert functions). 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. Thus the adversary cannot embed the decryption function as a leakage function after seeing the challenge ciphertext while is allowed to make challenge-dependent queries. This model is able to capture a wider class of real-world side-channel 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 identity-based encryption (IBE). Furthermore, we extend Canetti *et al.*'s technique, that converts CPA-secure IBE to CCA-secure PKE, into the leakage-resilient setting. More precisely, we construct a CCA-secure PKE in the post-challenge auxiliary input model, by using strong one-time signatures and strong extractor with hard-to-invert auxiliary inputs, together with a CPA-secure IBE in the auxiliary input model. Moreover, we extend our results to signatures, to obtain fully leakage-resilient signatures with auxiliary inputs using standard signatures and strong extractor with hard-to-invert auxiliary inputs. It is more efficient than the existing fully leakage-resilient signature schemes.

Keywords: leakage-resilient, auxiliary inputs, randomness

1 Introduction

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 ℓ bits such that ℓ must be less than $|\mathsf{sk}|$ [1]. Naor and Segev [17] considered the entropy of sk and required that the decrease in entropy of the sk is at most ℓ bits upon observing $f(\mathsf{sk})$. Dodis *et al.* [10] further generalized the leakage functions and proposed the model of *auxiliary input* which only requires the leakage functions to be computationally hard to compute sk given $f(\mathsf{sk})$.

1.1 Motivation for Post-Challenge Auxiliary Inputs

Post-challenge leakage query for PKE: The auxiliary input model is general enough to capture a large class of side-channel leakages. However, there are still shortcomings. For example, in the CCA security model for PKE, the adversary $\mathcal{A}$ is allowed to ask for the decryption of arbitrary ciphertexts *before and after* receiving the challenge ciphertext C^*, in order to maximize the ability of $\mathcal{A}$[3]. But for most leakage-resilient PKE, the adversary $\mathcal{A}$ can only specify and query the leakage function $f(\mathsf{sk})$ *before* getting C^*. In real situations, this is not true. The adversary should be able to obtain more information even after the attack target is known. The main reason for not being able to have *post-challenge leakage queries* (queries from the adversary after the challenge ciphertext is given) is as follows. If we allow $\mathcal{A}$ to specify the leakage function after getting C^*, he can easily embed the decryption of C^* as the leakage function, which will lead to a trivial break to the security game. So, the issue is to come up with a model with minimal restriction needed to allow post-challenge leakage query after getting the challenge ciphertext, while avoiding the above trivial attack. Comparing with the existing leakage-resilient PKE, the objective is to increase the ability of the adversary to make the model more realistic and capture a larger class of side-channel attacks.

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

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

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

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

In both scenarios, we should avoid the adversary $\mathcal{A}$ submitting a leakage function as the decryption of C^* in the security game (in case of leakage from secret key owner) or to submit a leakage function to reveal the information for the encryption randomness r for a trivial attack (in case of leakage from encryptor). A possible direction is to ask $\mathcal{A}$ to submit a set of functions $\mathcal{F}_0$ before seeing the public key or C^*. After seeing the challenge ciphertext, $\mathcal{A}$ can only ask for the leakage of arbitrary function $f' \in \mathcal{F}_0$. Therefore, f' cannot be the decryption of C^* and cannot lead to a trivial attack for the case of encryption randomness. This restriction is reasonable in the real world since most side-channel attacks apply to the physical implementation rather than the algorithm used (e.g. the leakage method of the power or timing attacks are the same, no matter RSA or ElGamal encryption are applied; 512-bit or 1024-bit keys are used.). Similar restriction was proposed by Yuen *et al.* [19] for leakage-resilient signatures in the auxiliary input model[4]. However, directly applying this idea to PKE, by simply allowing both pre-challenge and post-challenge leakages on sk, is not meaningful. Specifically, as the possible choice of leakage function f' is chosen before seeing the challenge ciphertext C^*, the post-challenge leakage $f'(\mathsf{sk})$ can simply be asked before seeing C^*, as a pre-challenge leakage. Therefore this kind of post-challenge leakage can be captured by slightly modifying the original auxiliary input model and does not strengthen our security model for PKE. Hence, we propose the leakage $f'(r)$ on the encryption randomness of C^* as the post-challenge leakage query. This kind of post-challenge leakage cannot be captured by the existing models. Since we focus on the auxiliary input model in this paper, we call our new model as the *post-challenge auxiliary input* model.

Practical Threats to Randomness. Finally, we want to stress that information leakage caused by poor implementation of pseudorandom number generator (PRNG) is practical. Argyros and Kiayias [2] outlined the flaws of PRNG in PHP. Lenstra *et al.* [14] inspected millions of public keys and found that some of the weak keys could be a result of poorly seeded PRNGs. Michaelis *et al.* [15] uncovered signicant weaknesses of PRNG of some java runtime libraries, including Android. These practical attacks demonstrate the potential weakness of the encryption randomness when using PRNG in practice.

1.2 Our Contributions

In this paper, we propose the post-challenge auxiliary input model for public key encryption. The significance of our post-challenge auxiliary input model is twofold. Firstly, it allows the leakage *after* seeing the challenge ciphertext. Secondly, it considers the leakage of two different parties: the secret key owner and the encryptor. In most leakage-resilient PKE schemes, they only consider the leakage of the secret key. However, the randomness used by the encryptor may also suffer from side-channel attacks. There are some encryption schemes which only consider the leakage on randomness, but not the secret key. Bellare *et al.* [3] only allows randomness leakage before receiving the public key. Namiki *et al.* [16] only allows randomness

[4] Yuen *et al.* [19] named their model as the selective auxiliary input model, due to similarity to the selective-ID model in identity-based encryption.

leakage before the challenge phase. Therefore our post-challenge auxiliary input model also improves this line of research on randomness leakage. To the best of the authors' knowledge, no existing leakage-resilient PKE schemes consider the leakage of secret key and randomness at the same time. Therefore, our post-challenge auxiliary input model is the *first* model to consider the leakage from both the secret key owner and the encryptor. This model captures a wider class of side-channel attacks than the previous models in the literature. We allow for leakage on the values being computed on, which will be a function of both the encryption random r and the public key pk. Specifically, we allows for $g(\mathsf{pk}, f(r))$ where g is any polynomial-time function and f is any computationally hard-to-invert function. We put the restriction on $f(r)$ to avoid trivial attacks on our security model.

To illustrate the feasibility of the model, we propose a generic construction of CPA-secure PKE in our new post-challenge auxiliary input model (pAI-CPA PKE). It is a generic transformation from the CPA-secure PKE in the auxiliary input model (AI-CPA PKE, e.g. [8]) 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. [18]).

Furthermore, we extend the generic transformation for CPA-secure IBE to CCA-secure PKE by Canetti *et al.* [7] into the leakage-resilient setting. The original transformation by Canetti *et al.* [7] 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 §4.1). Therefore, we have to link the randomness used in the IBE and the strong one-time signatures. We propose to use strong extractor with hard-to-invert auxiliary inputs as the linkage. It is because the strong extractor allows us to compute the randomness of IBE and the strong one-time signature from the same source, and yet remains indistinguishable from uniform distribution. It helps to simulate the leakage of the randomness in the security proof. Our contributions on encryption can be summarized in Fig. 1.

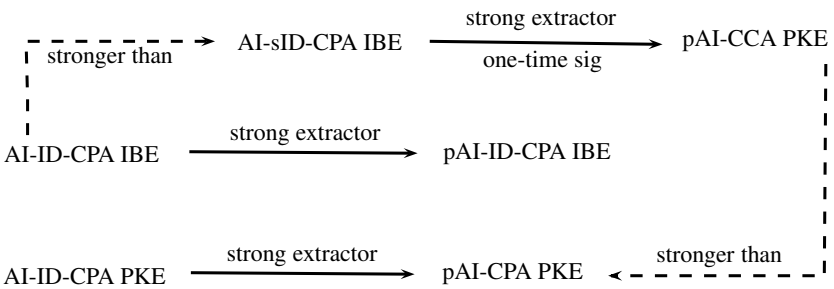

Fig. 1. Our Contributions on Encryption

Finally, we also observe that the strong extractor with hard-to-invert auxiliary inputs also helps us to construct fully leakage-resilient signatures with auxiliary inputs. We consider

the *selective auxiliary input model* by Yuen *et al.* [19], which allows the attacker to obtain (hard-to-invert) leakage on all randomness used by the secret key owner. Similar to our post-challenge auxiliary input model, a set of possible leakage functions must be submitted at the beginning of the security game, in order to avoid trivial attack (such as generating a forgery of a signature directly). We find a generic construction of such signature scheme by using a standard signature and the strong extractor with hard-to-invert auxiliary inputs. It greatly simplifies the construction of most generic leakage-resilient signatures (either in bounded leakage or hard-to-invert leakage; the leakage function either takes the secret key as input only or takes all intermediate randomness as input), which usually involves a number of primitives such as simulation sound NIZK, lossy encryption, admissible hash functions, etc [13, 6, 9, 5, 11, 19]. As a result, our scheme is the most efficient fully leakage-resilient signatures.

Related Works. Dodis *et al.* [10] introduced the model of *auxiliary inputs* leakage functions. PKE secure in the auxiliary input model was proposed in [8]. Signature schemes secure in the auxiliary input model were independently proposed by Yuen *et al.* [19] and Faust *et al.* [11], 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 [17] that

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

Halevi and Lin [12] 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 [12] 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 sk and encryption randomness r (together), such that leaking L bits on (sk, r) reveals no more than L bits on the underlying encrypted message.

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

2 Security Model of Post-Challenge Auxiliary Inputs

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

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 Π consists of three Probabilistic Polynomial Time (PPT) algorithms:

- $\mathtt{Gen}(1^\lambda)$: On input the security parameter λ, output a public key pk and a secret key sk.

- $\mathtt{Enc}(\mathsf{pk}, M)$: Denote the message space as $\mathcal{M}$. On input a message $M \in \mathcal{M}$ and pk, output a ciphertext C.
- $\mathtt{Dec}(\mathsf{sk}, C)$: On input sk and C, output the message M or $\perp$ for invalid ciphertext.

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

As introduced in §1.1, 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 §1.1. Since $\mathcal{A}$ is a PPT algorithm, we consider that $m := |\mathcal{F}_0|$ is polynomial in the security parameter λ.

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

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

1. The adversary $\mathcal{A}$ submits a set of leakage functions $\mathcal{F}_0$ to the challenger $\mathcal{C}$ with $m := |\mathcal{F}_0|$ is polynomial in λ.
2. $\mathcal{C}$ runs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{Gen}(1^\lambda)$ and outputs pk to $\mathcal{A}$.
3. $\mathcal{A}$ may adaptively query the (pre-challenge) leakage oracle:
   - $\mathcal{LO}_s(f_i)$ with f_i. $\mathcal{LO}_s(f_i)$ returns $f_i(\mathsf{sk}, \mathsf{pk})$ to $\mathcal{A}$.
4. $\mathcal{A}$ submits two messages $m_0, m_1 \in \mathcal{M}$ of the same length to $\mathcal{C}$. $\mathcal{C}$ samples $b \leftarrow \{0,1\}$ and the randomness of encryption $r' \leftarrow \{0,1\}^*$. It returns $C^* \leftarrow \mathtt{Enc}(\mathsf{pk}, m_b; r')$ to $\mathcal{A}$.
5. $\mathcal{A}$ may adaptively query the (post-challenge) leakage oracle and the decryption oracle:
   - $\mathcal{LO}_r(f'_i)$ with $f'_i \in \mathcal{F}_0$. It returns $f'_i(r')$ to $\mathcal{A}$.
   - $\mathcal{DEC}(C)$ with $C \neq C^*$. It returns $\mathtt{Dec}(\mathsf{sk}, C)$ to $\mathcal{A}$.
6. $\mathcal{A}$ outputs its guess $b' \in \{0,1\}$. The advantage of $\mathcal{A}$ is $Adv^{\mathrm{pAI-CCA}}_{\mathcal{A}}(\Pi) = |\Pr[b = b'] - \frac{1}{2}|$.

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

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

Denote the set of functions asked in the pre-challenge leakage oracle $\mathcal{LO}_s$ as $\mathcal{F}_s$. We have to define the families $(\mathcal{F}_s, \mathcal{F}_0)$ for the leakage functions asked in the oracles. We can define the family of length-bounded function by restricting the size of the function output as in [10] (Refer to [10] 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 $\mathcal{H}_{\mathsf{ow}}$, which is extended from the definition in [10]:

– Let $\mathcal{H}_{\mathsf{ow}}(\epsilon_r)$ be the class of all polynomial-time computable functions $h : \{0,1\}^{|r'|} \to \{0,1\}^*$, such that given $h(r')$ (for a randomly generated r'), no PPT algorithm can find r' with probability greater than ϵ_r[5]. The function $h(r')$ can be viewed as a composition of $q \in \mathbb{N}^+$ functions: $h(r') = (h_1(r'), \ldots, h_q(r'))$. Therefore $\{h_1, \ldots, h_q\} \in \mathcal{H}_{\mathsf{ow}}(\epsilon_r)$.

Also, we consider $\mathcal{F}_s$ as a family of one-way function $\mathcal{H}_{\mathsf{pk-ow}}$, which is extended from the definition in [10]:

– Let $\mathcal{H}_{\mathsf{pk-ow}}(\epsilon_s)$ be the class of all polynomial-time computable functions $h : \{0,1\}^{|\mathsf{sk}|+|\mathsf{pk}|} \to \{0,1\}^*$, such that given $(\mathsf{pk}, h(\mathsf{sk}, \mathsf{pk}))$ (for a randomly generated $(\mathsf{sk}, \mathsf{pk})$), no PPT algorithm can find sk with probability greater than ϵ_s[6]. The function $h(\mathsf{sk}, \mathsf{pk})$ can be viewed as a composition of q' functions: $h(\mathsf{sk}, \mathsf{pk}) = (h_1(\mathsf{sk}, \mathsf{pk}), \ldots, h_{q'}(\mathsf{sk}, \mathsf{pk}))$. Therefore $\{h_1, \ldots, h_{q'}\} \in \mathcal{H}_{\mathsf{pk-ow}}(\epsilon_s)$.

Definition 1. *We say that Π is pAI-CCA secure with respect to the families $(\mathcal{H}_{\mathsf{pk-ow}}(\epsilon_s), \mathcal{H}_{\mathsf{ow}}(\epsilon_r))$ if the advantage of any PPT adversary $\mathcal{A}$ in the above game is negligible.*

We can also define the security for chosen plaintext attack (CPA) similarly. By forbidding the decryption oracle query, we have the security model for pAI-CPA. If we further forbid the leakage of the encryption randomness, we get the original AI-CPA model in [10].

We also define the security model for identity-based encryption similarly in Appendix A.

3 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., [8]) and a *strong extractor with ϵ-hard-to-invert auxiliary inputs leakage.*

3.1 Strong Extractor with Hard-to-invert Auxiliary Inputs

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

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

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

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

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

[6] 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 [10], no PPT algorithm will output sk with ϵ_s probability given f_i, pk, as pk leaks some information about sk. Therefore, we also define that no PPT algorithm will output r' with ϵ_r probability given $f_i', C^*, \mathsf{pk}, m_0^*, m_1^*$. We omit these extra input parameters for simplicity in the rest of the paper.

An interesting property of the above definition is that such a strong extractor itself is ϵ-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 $f \in \mathcal{H}_{\mathsf{ow}}(\epsilon)$, given $(r, f(x))$ and $\mathsf{Ext}(r, x)$, no PPT adversary can find x with probability $\geq \epsilon$, provided that $\mathsf{Ext}(r, x)$ is a strong extractor with ϵ-hard-to-invert auxiliary inputs.*

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

Interestingly, we find that a strong extractor with ϵ-hard-to-invert auxiliary inputs can be constructed from the modified Goldreich-Levin theorem from [8]. Denote $\langle r, x\rangle = \sum_{i=1}^{l} r_i x_i$ as the inner product of $x = (x_1, \ldots x_l)$ and $r = (r_1, \ldots, r_l)$.

Theorem 1 ([8]). *Let q be a prime, and let $\bar{H}$ be an arbitrary subset of $GF(q)$. Let $f : \bar{H}^{\bar{n}} \to \{0,1\}^*$ be any (possibly randomized) function. s is chosen randomly from $\bar{H}^{\bar{n}}$, r is chosen randomly from $GF(q)^{\bar{n}}$ and u is chosen randomly from $GF(q)$. We also have $y = f(s)$. If there is a distinguisher D that runs in time t such that*

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

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

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

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

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

3.2 Construction of pAI-CPA Secure PKE

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

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

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

Proof. Denote the randomness used in the challenge ciphertext as x^*. Let $Game_0$ be the pAI-CPA security game with Π scheme. $Game_1$ is the same as $Game_0$ except that when encrypting the challenge ciphertext $c = \mathtt{Enc}'(\mathsf{pk}, m_b; y)$, we replace $y = \mathsf{Ext}(r, x^*)$ with y' which is chosen uniformly at random in $\{0,1\}^{m'}$. The leakage oracle outputs $f_i(x^*)$ for both games.

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

$$|Adv_{\mathcal{A}}^{Game_0}(\Pi) - Adv_{\mathcal{A}}^{Game_1}(\Pi)| \leq \mathsf{neg}(\lambda).$$

Assume that there exists an adversary $\mathcal{A}$ such that $|Adv_{\mathcal{A}}^{Game_0}(\Pi) - Adv_{\mathcal{A}}^{Game_1}(\Pi)| \geq \epsilon_A$ which is non-negligible.

The simulator $\mathcal{S}$ is given $(r, f_1(x^*), f_2(x^*), \ldots, f_q(x^*), T)$ where T is either $T_0 = \langle r, x^* \rangle$ or $T_1 = u$ which is a random number as in Definition 2. Given $f_1(x^*), \ldots, f_q(x^*)$, no PPT adversary can recover x^* with probability greater than ϵ_r by the definition of $\mathcal{H}_{\mathsf{ow}}(\epsilon_r)$. Then, the simulator generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{Gen}'(1^\lambda)$. It sets $\mathsf{SK} = \mathsf{sk}$ and gives the adversary $\mathsf{PK} = (\mathsf{pk}, r)$. The simulator can answer pre-challenge leakage oracle as it has PK and SK. The adversary submits two message m_0 and m_1 to the simulator where the simulator flips a coin b. It encrypts the challenge ciphertext $C^* = \mathtt{Enc}(\mathsf{pk}, m_b; T)$ and gives it to $\mathcal{A}$. $\mathcal{A}$ can ask $f_i(x)$ as the post-challenge leakage queries. $\mathcal{A}$ outputs its guess bit b' to the simulator. If $b = b'$, the simulator outputs 1; otherwise, it outputs 0.

Since the difference of advantage of $\mathcal{A}$ between $Game_0$ and $Game_1$ is ϵ_A, then

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

which is non-negligible if ϵ_A is non-negligible. It contradicts the definition of strong extractor in Definition 2. Therefore, no PPT adversary can distinguish $Game_0$ from $Game_1$ with non-negligible probability.

Next, we want to show that

$$Adv_{\mathcal{A}}^{Game_1}(\Pi) = \mathsf{neg}(\lambda).$$

Note that the challenge ciphertext now is $c = \mathtt{Enc}'(\mathsf{pk}, M; y')$ where y' is chosen uniformly at random in $\{0,1\}^{m'}$. Therefore the output of the leakage oracle $f_i(x^*)$ will not reveal any information related to c. Then $Game_1$ is the same as the AI-CPA game with Π'. As Π is based on Π' which is AI-CPA secure, we have that $Adv_{\mathcal{A}}^{Game_1}(\Pi)$ is negligible. □

Extension to IBE. We can use the same technique to construct pAI-ID-CPA secure IBE. Let $\Sigma' = (\mathsf{Setup}', \mathsf{Extract}', \mathtt{Enc}', \mathtt{Dec}')$ be an AI-ID-CPA secure IBE (e.g. [18]) where the encryption randomness is in $\{0,1\}^{m'}$, $\mathsf{Ext} : \{0,1\}^{l_1} \times \{0,1\}^{l_2} \to \{0,1\}^{m'}$ is a strong extractor with ϵ_r-hard-to-invert auxiliary inputs leakage, then construct a pAI-ID-CPA secure IBE scheme Σ as follows.

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

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

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

Corollary 1. *Instantiating with the strong extractor construction in §3.1 and the identity-based encryption scheme in [18], the identity-based encryption construction Σ' is pAI-ID-CPA secure.*

4 CCA Public Key Encryption from CPA Identity-Based Encryption

In this section, we show that auxiliary-inputs (selective-ID) CPA secure IBE and strong one-time signatures imply post-challenge auxiliary-inputs CCA secure PKE. Canetti *et al.* [7] 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 [7], we use the strong one-time signature to prevent the PKE adversaries asking for decrypting ciphertexts of ID^* in the post stage as the IBE adversaries are not allowed to ask $\mathsf{Extract}(\mathsf{ID}^*)$. However, we cannot apply the technique in [7] directly.

4.1 Intuition

Let $(\mathtt{Gen}_s, \mathtt{Sign}, \mathtt{Verify})$ be a strong one-time signature scheme. Let $(\mathsf{Setup}', \mathsf{Extract}', \mathtt{Enc}', \mathtt{Dec}')$ be an auxiliary-inputs CPA secure IBE scheme (refer to the definition in Appendix A, by dropping the post-challenge query). The construction directly following Canetti *et al.*'s transformation [7] is as follows.

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

Problems in the Post-Challenge Auxiliary Input Model. At first glance it seems that Canetti *et al.*'s transformation [7] also works in our pAI-CCA model for PKE, if we simply change the underlying IBE to be secure in the corresponding post-challenge auxiliary input model. However, we find that this is not true. The main challenge of pAI-CCA secure PKE is how to handle the leakage of the randomness used in the challenge ciphertext. It includes the randomness used in $\mathtt{Gen}_s$, $\mathtt{Sign}$ and $\mathtt{Enc}'$, denoted as r_{sig_1}, r_{sig_2} and r_{enc} respectively. Specifically, we have $(\mathsf{vk}, \mathsf{sk}_s) \leftarrow \mathtt{Gen}_s(1^\lambda; r_{\mathsf{sig}_1})$, $\sigma \leftarrow \mathtt{Sign}(\mathsf{sk}_s, c; r_{\mathsf{sig}_2})$ and $c \leftarrow \mathtt{Enc}'(\mathsf{mpk}, \mathsf{vk}, m_b; r_{\mathsf{enc}})$.

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

- $f_1(r') = r_{\mathsf{enc}}$, such that f_1 is still hard-to-invert upon r'. In this case, $\mathcal{A}$ can test $c^* \stackrel{?}{=} \mathtt{Enc}'(\mathsf{mpk}, \mathsf{vk}, m_0; r_{\mathsf{enc}})$ to win the pAI-CCA game; or
- $f_2(r') = (r_{\mathsf{sig}_1}, r_{\mathsf{sig}_2})$, such that f_2 is still hard-to-invert upon r'. In this case, given r_{sig_1}, $\mathcal{A}$ can generate $(\mathsf{vk}, \mathsf{sk}_s) = \mathtt{Gen}_s(1^\lambda; r_{\mathsf{sig}_1})$ which causes $\Pr[\mathsf{Forge}]$ defined in [7] 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 [7] fail in our model.

Our Solution: To get rid of this problem, we set both $r_{\mathsf{sig}_1}, r_{\mathsf{sig}_2}$ and r_{enc} are generated from the same source of randomness $x \in \{0,1\}^{l_2}$. Suppose $r_{\mathsf{sig}_1}||r_{\mathsf{sig}_2}$ and r_{enc} are bit-strings of length n'. Suppose $\mathsf{Ext} : \{0,1\}^{l_1} \times \{0,1\}^{l_2} \to \{0,1\}^{n'}$ is a strong extractor with ϵ_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 pk. Then the *randomness* used in the IBE and the one-time signature can be calculated by $r_{\mathsf{enc}} = \mathsf{Ext}(r_1, x)$ and $(r_{\mathsf{sig}_1}||r_{\mathsf{sig}_2}) = \mathsf{Ext}(r_2, x)$ 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 Π'. The simulator of the pAI-CCA game has to simulate the post-challenge leakage oracle without knowing the encryption randomness x of the challenge ciphertext, which was produced by the challenger of Π'. We solve this problem by proving that it is indistinguishable by replacing $r^*_{enc} = \mathsf{Ext}(r_1, x^*)$ and $r^*_{sig_1}||r^*_{sig_2} = \mathsf{Ext}(r_2, x^*)$ with random numbers. Therefore, the post-challenge leakages on x^* will be independent with r^*_{enc} and $r^*_{sig_1}||r^*_{sig_2}$ which are used to produce the real challenge ciphertext. Then, the simulator can randomly choose x^* and simulate the post-challenge oracles by it own. However, when we show to replace $r^*_{sig_1}||r^*_{sig_2}$ with a random number, the simulator needs to compute $r_{enc^*} = \mathsf{Ext}(r_1, x^*)$. One way to solve it is to include $\mathsf{Ext}(r_1, x^*)$ as a post-challenge leakage query in the pAI-CCA game. As we will see later (by Lemma 1), including $\mathsf{Ext}(r_1, x^*)$ in leakage queries is still ϵ_r-hard-to-invert.

Following [7], 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*.

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

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

Theorem 5. *Assuming that Π' is a AI-sID-CPA secure IBE scheme with respect to family $\mathcal{H}_{\mathsf{pk-ow}}(\epsilon_s)$, Π_s is a strong one-time signature and Ext is ϵ_r hard-to-invert strong extractor, then there exists a PKE scheme Π which is pAI-CCA secure with respect to families $(\mathcal{H}_{\mathsf{pk-ow}}(\epsilon_s), \mathcal{H}_{\mathsf{ow}}(\epsilon_r))$.*

Proof. We prove the security by a number of security games. Let Game_0 be the original pAI-CCA game for the PKE scheme Π. Specifically for the challenge ciphertext, the simulator picks a random number x^* to compute $r^*_{\mathsf{enc}} = \mathsf{Ext}(r_1, x^*)$ and $r^*_{\mathsf{sig}_1}||r^*_{\mathsf{sig}_2} = \mathsf{Ext}(r_2, x^*)$. Let Game_1 be the same as Game_0, except that $r^*_{\mathsf{sig}_1}||r^*_{\mathsf{sig}_2}$ is randomly chosen from $\{0,1\}^{n'}$. Let Game_2 be the same as Game_1, except that r^*_{enc} is randomly chosen from $\{0,1\}^{n'}$.

Lemma 2. *For any PPT adversary $\mathcal{A}$, Game_0 is indistinguishable from Game_1 if Ext is a strong extractor with ϵ_r-hard-to-invert auxiliary inputs.*

Lemma 3. *For any PPT adversary $\mathcal{A}$, Game_1 is indistinguishable from Game_2 if Ext is a strong extractor with ϵ_r-hard-to-invert auxiliary inputs.*

Lemma 4. *For any PPT adversary $\mathcal{A}$, the advantage in Game_2 is negligible if Π' is a AI-sID-CPA secure IBE scheme with respect to family $\mathcal{H}_{\mathsf{pk-ow}}(\epsilon_s)$ and Π_s is a strong one-time signature.*

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

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

$$|Adv^{Game_0}_{\mathcal{A}}(\Pi) - Adv^{Game_1}_{\mathcal{A}}(\Pi)| \leq \mathsf{neg}(\lambda).$$

Assume that there exists an adversary $\mathcal{A}$ such that $|Adv^{Game_0}_{\mathcal{A}}(\Pi) - Adv^{Game_1}_{\mathcal{A}}(\Pi)| \geq \epsilon_A$ which is non-negligible.

The simulator $\mathcal{S}$ picks a random $r_1, r_2 \in \{0,1\}^{l_1}$. The simulator is given $(r_2, f_1(x^*), \ldots, f_q(x^*), f_{q+1}(x^*), T)$ where $f_1, \ldots, f_q \in \mathcal{F}_0$, $f_{q+1}(x^*) = \mathsf{Ext}(r_1, x^*)$, and T is either $T_0 = \langle r_2, x^* \rangle$ or $T_1 = u$ (a random number as in Definition 2). Given $f_1(x^*), \ldots, f_q(x^*)$, no PPT adversary

can recover x^* with probability greater than ϵ_r by the definition of $\mathcal{H}_{\mathsf{ow}}(\epsilon_r)$ (We will later show that including $\mathsf{Ext}(r_1, \cdot)$ is also ϵ_r-hard-to-invert).

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

Since the difference of advantage of $\mathcal{A}$ between $Game_0$ and $Game_1$ is ϵ_A, then

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

which is non-negligible if ϵ_A is non-negligible. It contradicts the definition of strong extractor in Definition 2. Therefore, no PPT adversary can distinguish $Game_0$ from $Game_1$ with non-negligible probability.

Finally, we need to show that including $\mathsf{Ext}(r_1, \cdot)$ is also ϵ_r-hard-to-invert. This follows directly from Lemma 1 if we set $f = (f_1(x^*), \ldots, f_q(x^*)) \in \mathcal{H}_{ow}(\epsilon_r)$. □

Proof (Lemma 3). The post-challenge query functions $(f_1, \ldots, f_q) \in \mathcal{F}_0$ are ϵ_r-hard-to-invert by definition. Fix any auxiliary-input function $f_1, \ldots, f_q$, $\langle r_1, f_1(x^*), \ldots, f_q(x^*), \mathsf{Ext}(r_1, x^*) \rangle$ is indistinguishable with $\langle r_1, f_1(x^*), \ldots, f_q(x^*), u \rangle$ where u is randomly chosen from $\{0,1\}^{n'}$, by the definition of strong extractor. Hence Game_1 is indistinguishable from Game_2. The reduction is similar to the previous proof. □

Proof (Lemma 4). Let $\mathcal{A}$ be an adversary to Π on Game_2 and we construct an AI-sID-CPA adversary $\mathcal{A}'$ to Π' that runs $\mathcal{A}$ as a subroutine. Initially, $\mathcal{A}$ submits a set of leakage functions $\mathcal{F}_0$ that he would like to ask in the Game_2 to $\mathcal{A}'$. $\mathcal{A}'$ picks $r_{\mathsf{sig}_1}||r_{\mathsf{sig}_2}$ uniformly random from $\{0,1\}^{n'}$ and computes $(\mathsf{vk}^*, \mathsf{sk}_s^*) = \mathtt{Gen}_s(1^\lambda; r_{\mathsf{sig}_1})$. $\mathcal{A}'$ submits the challenge identity vk^* to the AI-sID-CPA challenger $\mathcal{C}$, and $\mathcal{C}$ returns mpk to $\mathcal{A}'$. Then $\mathcal{A}'$ picks r_1 and r_2 which are independent and uniformly chosen from $\{0,1\}^{l_1}$. $\mathcal{A}'$ gives $\mathsf{pk} = (\mathsf{mpk}, r_1, r_2)$ to $\mathcal{A}$.

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

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

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

Finally $\mathcal{A}$ outputs its guess b' and $\mathcal{A}'$ forwards it to $\mathcal{C}$ as its guess bit. Therefore, if $\mathcal{A}$ wins the Game_2 with a non-negligible probability, then $\mathcal{A}'$ will win the AI-sID-CPA game also with a non-negligible probability, which contradicts that Π' is AI-sID-CPA secure.

To show that the probability that $\mathcal{A}$ asks for the decryption of a valid ciphertext with identity vk^* is negligible, let $\mathcal{C}'$ be the challenger of the strong one-time signature scheme. We construct an algorithm $\mathcal{B}$ to break the strong one-time signature scheme by running $\mathcal{A}$ as a subroutine. Initially, $\mathcal{A}$ submits its post-challenge leakage class $\mathcal{F}_0$ to $\mathcal{B}$. $\mathcal{C}'$ gives vk^* to $\mathcal{B}$. $\mathcal{B}$ runs $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}'(1^\lambda)$ and picks r_1 and r_2 which are independent and uniformly chosen from $\{0,1\}^{l_1}$. $\mathcal{B}$ returns $\mathsf{pk} = (\mathsf{mpk}, r_1, r_2)$ to $\mathcal{A}$.

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

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

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

5 Fully Leakage Resilient Signatures with Selective Auxiliary Inputs

In the previous section, we used a strong extractor to generate the randomness of IBE and the strong one-time signatures, in order to achieve CCA security for PKE, in the post-challenge auxiliary input model. As shown in §3, using the strong extractor to generate the randomness of AI-CPA secure PKE (and IBE) can enhance the security to the post-challenge auxiliary input model. One natural question to ask is that if the strong extractor can be used to increase the security of signature schemes.

A signature scheme is *fully leakage-resilient* [13] if it is existentially unforgeable under an adaptive chosen-message attack even if an adversary may obtain leakage information (either bounded or hard-to-invert leakage) on all intermediate values that are used throughout the lifetime of the system. We find a new generic construction of fully leakage-resilient signatures from standard signatures and a strong extractor, in the selective auxiliary input model [19].

We review the selective auxiliary input model for fully leakage-resilient signatures in Appendix B.

5.1 Construction

Let $\Pi' = (\mathtt{KeyGen}', \mathtt{Sign}', \mathtt{Verify}')$ be an EUF-CMA secure standard signatures where the signing randomness is in $\{0,1\}^{m'}$, $\mathsf{Ext} : \{0,1\}^{l_1} \times \{0,1\}^{l_2} \to \{0,1\}^{m'}$ is a strong extractor with ϵ-hard-to-invert auxiliary inputs leakage, then construct an sAI-EUF-CMA secure fully leakage-resilient signatures (with respect to family $\mathcal{H}_{\mathsf{pk}-\sigma-\mathsf{ow}}(\epsilon)$) Π as follows.

1. $\mathtt{KeyGen}(1^\lambda)$: It picks r uniformly random from $\{0,1\}^{l_1}$ and x_0 uniformly random from $\{0,1\}^{l_2}$. Then, it computes $y_0 = \mathsf{Ext}(r, x_0)$. It runs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KeyGen}'(1^\lambda; y_0)$. Then, we set the public key $\mathsf{PK} = (\mathsf{pk}, r)$ and the secret key $\mathsf{SK} = \mathsf{sk}$.
2. $\mathtt{Sign}(\mathsf{SK}, M)$: It picks x uniformly random from $\{0,1\}^{l_2}$. Then, it computes $y = \mathsf{Ext}(r, x)$. The signature is $\sigma = \mathtt{Sign}'(\mathsf{sk}, M; y)$.

3. $\mathtt{Verify}(\mathsf{PK}, M, \sigma)$: It returns $\mathtt{Verify}'(\mathsf{pk}, M, \sigma)$.

Theorem 6. *If Π' is an EUF-CMA secure signature and* Ext *is a strong extractor with ϵ-hard-to-invert auxiliary inputs leakage, then Π is sAI-EUF-CMA secure fully leakage-resilient signatures with respect to family $\mathcal{H}_{\mathsf{pk}-\sigma-\mathsf{ow}}(\epsilon)$.*

The proof is given in Appendix C.

5.2 Comparison

We compare the our proposed scheme with existing leakage-resilient signatures in Table 1, based on the size of secret key used. For the schemes secure in auxiliary input models, the class of leakage function allowed is related to the key size ℓ (as well as other parameters in the concrete scheme). The verification time of our scheme is the same as the verification time of the underlying signature scheme, which is usually $O(1)$ exponentiation or pairing operations. Our proposed scheme is more efficient than the existing fully leakage-resilient signatures (i.e. allow leakage of the randomness) in many aspects, as shown in Table 1.

Scheme	Size of			Computation Time of			Types of Leakage	
	pk	sk	σ	KeyGen	Sign	Verify	Randomness	Model
[13] Scheme 1	$O(1)$	ℓ bits	$O(\ell)$	UOWHF hash of ℓ bits	$O(\ell)$ exp	$O(\ell)$ exp	$\times$	bounded leakage of $\ell - \ell^{\epsilon}$ bits, for any $\epsilon > 0$
[9]	$O(1)$	$O(\ell)$	$O(\ell)$	$O(\ell)$ exp	$O(\ell)$ exp	$O(\ell)$ pairing	$\times$	continual leakage of $\ell \log p - \lambda$ bits
[13] Scheme 2	$O(\ell)$	$O(\ell)$	$O(\ell)$	$O(\ell)$ UOWHF hash	O(1) evaluation of injective map	$O(\ell)$ UOWHF hash	$\surd$	t-time signatures, bounded leakage of $\Theta(\ell/t)$ bits
[5]	$O(1)$	$O(\ell)$	$O(\ell)$	$O(\ell)$ pairing	$O(\ell)$ exp	$O(\ell)$ pairing	$\surd$	bounded leakage of $\ell \log p - 2\lvert\mathbb{G}_T\rvert - \omega(\log \lambda)$ bits
[11]	$O(1)$	$O(\ell)$	$O(\ell)$	$O(\ell)$ pairing	$O(\ell)$ exp	$O(\ell)$ pairing	$\times$	auxiliary input model of EU-CMMA [5]
[19]	$O(1)$	$O(\ell)$	$O(\ell)$	$O(\ell)$ pairing	$O(\ell)$ exp	$O(\ell)$ pairing	$\surd$	selective auxiliary input model of EUF-CMA
This paper	$O(1)$	ℓ-bits	$O(1)$	dot product of ℓ-bit vectors	dot product of ℓ-bit vectors	$O(1)$ exp or pairing	$\surd$	selective auxiliary input model of EUF-CMA

Table 1. Comparison of leakage-resilient signatures, using the instantiation proposed in the original paper, with security parameter λ. The size are counted as the number of group elements, unless otherwise specified. Denote ℓ as the number of group element of sk in each scheme, and p as the order of the group. For auxiliary input models, ℓ affects the class of hard-to-invert function. The verification time of our scheme is the same as the underlying signature scheme, which is usually some $O(1)$ exponentiation or pairing.

6 Conclusion

In this paper, we solved the open problem of allowing leakage from *both the secret key owner and the encryptor*, and allowing leakage *after* seeing the challenge ciphertext. Specifically, we proposed the post-challenge auxiliary input model to capture these leakages. We showed that the post-challenge auxiliary inputs secure PKE (and IBE) can be constructed from auxiliary inputs secure PKE (and IBE) and strong extractors with auxiliary inputs. We extend the generic transformation of CCA security in [7] in the leakage-resilient setting, using strong one-time signature and strong extractors with auxiliary inputs. Finally, we give a new and simple generic construction of fully leakage-resilient signatures with selective auxiliary inputs.

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. E. Boyle, G. Segev, and D. Wichs. Fully leakage-resilient signatures. In K. G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 89–108. Springer, 2011.
6. Z. Brakerski, Y. T. Kalai, J. Katz, and V. Vaikuntanathan. Overcoming the hole in the bucket: Public-key cryptography resilient to continual memory leakage. In *FOCS 2010*, pages 501–510. IEEE Computer Society, 2010.
7. 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.
8. 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.
9. Y. Dodis, K. Haralambiev, A. López-Alt, and D. Wichs. Cryptography against continuous memory attacks. In *FOCS 2010*, pages 511–520. IEEE Computer Society, 2010.
10. Y. Dodis, Y. T. Kalai, and S. Lovett. On cryptography with auxiliary input. In M. Mitzenmacher, editor, *STOC 2009*, pages 621–630. ACM, 2009.
11. 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.
12. 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.
13. J. Katz and V. Vaikuntanathan. Signature schemes with bounded leakage resilience. In M. Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 703–720. Springer, 2009.
14. 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.
15. 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.
16. 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.
17. 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.
18. 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.
19. 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.

A Security Model for Identity-Based Encryption

We define post-challenge auxiliary inputs game for identity-based encryption against chosen plaintext attack. An identity-based encryption scheme Π consists of four PPT algorithms:

– $\mathsf{Setup}(1^\lambda)$: On input the security parameter λ, output a master public key mpk and a master secret key msk. Denote the message space as $\mathcal{M}$ and the identity space as $\mathcal{I}$.

- $\mathsf{Extract}(\mathsf{msk}, \mathsf{ID})$: On input msk and an identity $\mathsf{ID} \in \mathcal{I}$, output the identity-based secret key $\mathsf{sk}_{\mathsf{ID}}$.
- $\mathtt{Enc}(\mathsf{mpk}, \mathsf{ID}, M)$: On input mpk, $\mathsf{ID} \in \mathcal{I}$ and a message $M \in \mathcal{M}$, output a ciphertext C.
- $\mathtt{Dec}(\mathsf{sk}_{\mathsf{ID}}, C)$: On input $\mathsf{sk}_{\mathsf{ID}}$ and C, output the message M or $\perp$ for invalid ciphertext.

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

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

1. The adversary $\mathcal{A}$ submits a set of leakage functions $\mathcal{F}_0$ to the challenger $\mathcal{C}$ with $m := |\mathcal{F}_0|$ is polynomial in λ.
2. $\mathcal{C}$ runs $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and outputs mpk to $\mathcal{A}$. $\mathcal{C}$ also samples the randomness of encryption $r' \leftarrow \{0,1\}^*$.
3. $\mathcal{A}$ may adaptively query the (pre-challenge) leakage oracles:
   - $\mathcal{LO}_s(f_i)$ with f_i. $\mathcal{LO}_s(f_i)$ returns $f_i(\mathsf{msk}, \mathsf{mpk})$ to $\mathcal{A}$.
4. $\mathcal{A}$ submits its challenge identity $\mathsf{ID}^* \in \mathcal{I}$ along with two messages $m_0, m_1 \in \mathcal{M}$ of the same length to $\mathcal{C}$. $\mathcal{C}$ samples $b \leftarrow \{0,1\}$. It returns $C^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{ID}^*, m_b; r')$ to $\mathcal{A}$.
5. $\mathcal{A}$ may adaptively query the (post-challenge) leakage oracle
   - $\mathcal{LO}_r(f'_i)$ with $f'_i \in \mathcal{F}_0$. $\mathcal{LO}_r(f'_i)$ returns $f'_i(r')$ to $\mathcal{A}$.
   - $\mathcal{EO}(\mathsf{ID})$ for $\mathsf{ID} \neq \mathsf{ID}^* \in \mathcal{I}$. The extraction oracle returns $\mathsf{sk}_{\mathsf{ID}} \leftarrow \mathsf{Extract}(\mathsf{msk}, \mathsf{ID})$.
6. $\mathcal{A}$ outputs its guess $b' \in \{0,1\}$. The advantage of $\mathcal{A}$ is $Adv_{\mathcal{A}}^{\mathrm{pAI-ID-CPA}}(\Pi) = |\Pr[b = b'] - \frac{1}{2}|$.

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

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

Definition 3. *We say that Π is pAI-ID-CPA secure with respect to the families $(\mathcal{F}_s, \mathcal{F}_0)$ if the advantage of any PPT adversary $\mathcal{A}$ in the above game is negligible.*

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

B Selective Auxiliary Input Model for Unforgeability

We first review the selective auxiliary input model for fully leakage-resilient signatures [19], which is similar to our post-challenge auxiliary input model for encryption. We consider the following existential unforgeability game against chosen message attacks (EUF-CMA) for signatures, together with the leakage-resilient with selective auxiliary inputs [19].

Let $\Pi = (\mathtt{KeyGen}_s, \mathtt{Sign}, \mathtt{Verify})$ be a signature scheme. The existential unforgeability against selective auxiliary inputs and adaptive chosen-message attacks is defined in the following game (sAI-EUF-CMA) in the security parameter λ.

1. Select. Denote $\mathcal{F}$ as the space of leakage functions. $\mathcal{A}$ submits a set of leakage functions $\mathcal{F}_0 \subset \mathcal{F}$ to the challenger $\mathcal{C}$.
2. Setup. $\mathcal{C}$ samples $r_{\mathsf{sig}_1} \leftarrow \{0,1\}^*$ and runs $(\mathsf{vk}, \mathsf{sk}_s) \leftarrow \mathtt{KeyGen}_s(1^\lambda; r_{\mathsf{sig}_1})$. $\mathcal{C}$ gives vk to $\mathcal{A}$.
3. Query. Each of following oracles can be queried by $\mathcal{A}$:
   - Signing Oracle $\mathcal{SO}(m)$: On input a message m in the message space, it samples $r_{\mathsf{sig}_2} \leftarrow \{0,1\}^*$ and returns the signature $\sigma \leftarrow \mathtt{Sign}(\mathsf{sk}, m; r_{\mathsf{sig}_2})$.
   - Leak Oracle $\mathcal{LO}(f_i)$: On input a polynomial-time computable function $f_i \in \mathcal{F}_0$, denote R as all past randomness used, including r_{sig_1} and different r_{sig_2} used in signing oracles. It returns $f_i(R)$[7].
4. Output. $\mathcal{A}$ returns a message-signature pair (m^*, σ^*). $\mathcal{A}$ wins the game if $\mathtt{Verify}(\mathsf{vk}, m^*, \sigma^*) = 1$ and m^* was not the asked to $\mathcal{SO}$. The advantage of $\mathcal{A}$ is $Adv_{\mathcal{A}}^{\text{sAI}-\text{EUF}-\text{CMA}}(\Pi) = \Pr[\mathcal{A} \text{ wins}]$.

Denote the number of leak oracle queries as q_ℓ and denote σ as the signing oracle outputs. We usually consider $\mathcal{F}_0$ as a family of one-way function $\mathcal{H}_{\mathsf{pk}-\sigma-\mathsf{ow}}$:

- Let $\mathcal{H}_{\mathsf{pk}-\sigma-\mathsf{ow}}(\epsilon)$ be the class of all polynomial-time computable functions $h : \{0,1\}^{|R|} \to \{0,1\}^*$, such that given vk, σ and $\{f_i(R)\}_{i\in[1,q_\ell]}$, (for $(r_{\mathsf{sig}_1}, r_{\mathsf{sig}_2})$ that is randomly generated), no PPT algorithm can find sk_s with probability greater than ϵ.

Definition 4. *The fully leakage-resilient signature scheme Π is unforgeable against selective auxiliary inputs and adaptive chosen-message attacks (sAI-EUF-CMA) with respect to the family $\mathcal{H}_{\mathsf{pk}-\sigma-\mathsf{ow}}(\epsilon)$ if the advantage of any PPT adversary $\mathcal{A}$ in the above game is negligible.*

C Proof of Theorem 6

Proof. Let $Game_0$ be the sAI-EUF-CMA security game with Π scheme. Denote the number of signing oracle queries and leak oracle queries as q_s and q respectively. For $i \in [1, q_s]$, define $Game_i$ is the same as $Game_0$ except that for the last i-th signing oracle queries, we replace $y = \mathsf{Ext}(r, x)$ with y' which is chosen uniformly at random in $\{0,1\}^{m'}$.

Let $Adv_{\mathcal{A}}^{Game_i}(\Pi)$ be the advantage that the adversary $\mathcal{A}$ wins in $Game_i$ with Π scheme. Now, we need to show for any PPT adversary $\mathcal{A}$ and for $i \in [1, q_s]$:

$$|Adv_{\mathcal{A}}^{Game_{i-1}}(\Pi) - Adv_{\mathcal{A}}^{Game_i}(\Pi)| \leq \mathsf{neg}(\lambda).$$

Assume that there exists an adversary $\mathcal{A}$ such that $|Adv_{\mathcal{A}}^{Game_i-1}(\Pi) - Adv_{\mathcal{A}}^{Game_i}(\Pi)| \geq \epsilon_A$ which is non-negligible.

The simulator $\mathcal{S}$ is given $(r, f_1(x^*), f_2(x^*), \ldots, f_q(x^*), T)$ where T is either $T_0 = \langle r, x^* \rangle$ or $T_1 = u$ which is a random number as in Definition 2. Given $f_1(x^*), \ldots, f_q(x^*)$, no PPT adversary can recover x^* with probability greater than ϵ by the definition of $\mathcal{H}_{\mathsf{pk}-\sigma-\mathsf{ow}}(\epsilon)$. Then, the simulator picks x_0 uniformly random from $\{0,1\}^{l_2}$ and generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KeyGen}'(1^\lambda; \mathsf{Ext}(r, x_0))$. It sets $\mathsf{SK} = \mathsf{sk}$ and gives the adversary $\mathsf{PK} = (\mathsf{pk}, r)$. The simulator can answer pre-challenge leakage oracle as it has PK and SK.

[7] We follow the fully security model in [5, 19] that only leaking the randomness in the $\mathtt{KeyGen}_s$, but not the secret key itself. As stated in [5], we do not need to explicitly add sk_s to R, since sk_s is a deterministic function of the initial randomness $\mathtt{KeyGen}_s(1^\lambda; r_{\mathsf{sig}_1})$ and $r_{\mathsf{sig}_1} \in R$. Therefore the leakage of sk_s is implied by the leakage of R.

For the first $(q_s - i)$-th signing oracle queries, $\mathcal{S}$ picks x uniformly random from $\{0,1\}^{l_2}$. Then, it computes $y = \mathsf{Ext}(r, x)$. $\mathcal{S}$ returns the signature $\sigma = \mathtt{Sign}'(\mathsf{sk}, M; y)$ and stores the randomness x. For the $(q_s - i + 1)$-th signing oracle query, $\mathcal{S}$ returns the signature $\sigma = \mathtt{Sign}'(\mathsf{sk}, M; T)$ and (implicitly) sets the randomness as x^*. For the remaining signing oracle queries, $\mathcal{S}$ picks some random number $x \in \{0,1\}^{l_2}, u \in \{0,1\}^{m'}$, returns the signature $\sigma = \mathtt{Sign}'(\mathsf{sk}, M; u)$ and stores the randomness x. Note that if $T = \langle r, x^* \rangle$, $\mathcal{S}$ simulates $Game_{i-1}$. Otherwise, $\mathcal{S}$ simulates $Game_i$.

For the leak oracle queries, the only signing randomness not known by $\mathcal{S}$ is x^*. However, the leakage of x^* can be simulated by $f_1(x^*), f_2(x^*), \ldots, f_q(x^*)$ since $(f_1, \ldots, f_q) \in \mathcal{F}_0$.

Therefore, upon Definition 2, no PPT adversary can distinguish $Game_{i-1}$ from $Game_i$ with non-negligible probability.

Let $Game_{kg}$ be the same as $Game_{q_s}$ except that in the $\mathtt{KeyGen}$ phase we replace $y_0 = \mathsf{Ext}(r, x_0)$ with y_0' which is chosen uniformly at random in $\{0,1\}^{m'}$. Similar to above, we can show that

$$|Adv_{\mathcal{A}}^{Game_{q_s}}(\Pi) - Adv_{\mathcal{A}}^{Game_{kg}}(\Pi)| \leq \mathsf{neg}(\lambda).$$

The leakage f_i on this randomness x_0 can be simulated by $f_i(x_0)$ given by the challenger of Ext, if f_i is ϵ-hard-to-invert. Leakage on all other randomness are known to the simulator.

Finally, we want to show that

$$Adv_{\mathcal{A}}^{Game_{kg}}(\Pi) = \mathsf{neg}(\lambda).$$

We note that since all the leaked randomness is independent with the real randomness used, the leakage will not help the adversary. $Game_{kg}$ is the same as the EUF-CMA game with Π'. As Π is based on Π' which is EUF-CMA secure, we have that $Adv_{\mathcal{A}}^{Game_{kg}}(\Pi)$ is negligible. □