Functional Encryption against Probabilistic Queries: Definition, Construction and Applications

Geng Wang, Shi-Feng Sun, Zhedong Wang and Dawu Gu

School of Electronic Information and Electrical Engineering
Shanghai Jiao Tong University, 200240, P.R.China
{wanggxx,shifeng.sun,wzdstill,dwgu}@sjtu.edu.cn

Abstract. Functional encryption (FE for short) can be used to calculate a function output of a message, without revealing other information about the message. There are mainly two types of security definitions for FE, exactly simulation-based security (SIM-security) and indistinguishability-based security (IND-security). Both of them have some limitations: FE with SIM-security supporting all circuits cannot be constructed for unbounded number of ciphertext and/or key queries, while IND-security is sometimes not enough: there are examples where an FE scheme is IND-secure but not intuitively secure. In this paper, we present a new security definition which can avoid the drawbacks of both SIM-security and IND-security, called indistinguishability-based security against probabilistic queries (pIND-security for short), and we give an FE construction for all circuits which is secure for unbounded key/ciphertext queries under this new security definition. We prove that this new security definition is strictly between SIM-security and IND-security, and provide new applications for FE which were not known to be constructed from IND-secure or SIM-secure FE.

Keywords: functional encryption, probabilistic queries, indistinguishability-based security, provable security

1 Introduction

Functional encryption (FE) was first introduced by Boneh et al in 2011 [BSW11], which can calculate the function output $f(m)$ given the encrypted message $\mathsf{Enc}(m)$, and leaks nothing else about the message m. Functional encryption is a mighty cryptographic primitive, and can be considered as a generalization of attribute-based encryption, predicate encryption and inner product encryption.

Functional encryption is also an important method for computing on encrypted data, especially for cloud computing [KLM⁺18,RSG⁺19,MSH⁺19]. Using functional encryption, the cloud server can take ciphertexts as input, and outputs the required computation result as plaintext. This is different from homomorphic encryption, where the result is a ciphertext that requires additional decryption procedure and may not be suitable for some applications.

Informally, a functional encryption scheme consists of four algorithms: despite the normally defined algorithms $\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec}$ as in public key encryption, there is another algorithm KeyGen in functional encryption, which takes the master secret key and a function $f \in \mathcal{F}$ as input, and outputs a function key sk_f. In the decryption algorithm, function key sk_f instead of the master secret key is used, and the function value $f(m)$ instead of the message m itself is returned. (See Section 2 for the formal definition.)

There are mainly two types of security definitions for functional encryption: indistinguishability-based security (IND-security) and simulation-based security (SIM-security). However, both of them have their own drawbacks. We first briefly introduce the two types of security notions, then show why it is necessary to define a new type of security notions between them.

1.1 Overview of Security Notions for FE

The standard IND-security is equivalent to the natural notion of semantic security in public key encryption, and is also defined for many other cryptographic primitives, such as identity-based and attribute-based encryption. But for functional encryption, it has been pointed out that IND-security is not the strongest security definition. We first informally recall the definition of IND-security for FE:

An adversary $\mathcal{A}$ cannot distinguish between a ciphertext for m_0 and a ciphertext for m_1, even if allowed to query secret keys $\{sk_f\}$ for polynomial many different functions $\{f \in \mathcal{F}\}$, providing that $f(m_0) = f(m_1)$. (We say that $\mathcal{A}$ is "admissible" if it only makes queries such that $f(m_0) = f(m_1)$.)

It seems to be natural for the restriction $f(m_0) = f(m_1)$, since $\mathcal{A}$ can trivially determine whether the ciphertext is for m_0 or m_1 otherwise. However, such a restriction leads to the counter-intuitive example given in [O'N10,BSW11] and refined in [AGVW13]:

Example 1.1 ([BSW11,AGVW13]). Let $\mathcal{F}$ be a family of one-way permutations. Suppose that $\mathsf{PKE} = (\mathsf{PKE.KeyGen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ is a secure public-key encryption scheme. Then the following FE construction for $\mathcal{F}$ is IND-secure:

- $\mathsf{Setup}(1^\lambda)$: Let $(\mathsf{PKE.pk}, \mathsf{PKE.sk}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$, and return $\mathsf{PK} = \mathsf{PKE.pk}, \mathsf{MSK} = \mathsf{PKE.sk}$.
- $\mathsf{Enc}(\mathsf{PK}, m)$: Return $\mathsf{PKE.Enc}(\mathsf{PK}, m)$.
- $\mathsf{KeyGen}(\mathsf{MSK}, f)$: Return (MSK, f).
- $\mathsf{Dec}(sk_f, ct_m)$: Let $sk_f = (\mathsf{MSK}, f)$, return $f(\mathsf{PKE.Dec}(\mathsf{MSK}, ct_m))$.

However, each sk_f totally leaks m, while $f(m)$ does not leak m (since f is one-way).

It is not difficult to understand why such a counter-example exists: since the adversary is only allowed to query on f such that $f(m_0) = f(m_1)$, it is not allowed to make any single key query if $\mathcal{F}$ is a family of one-way permutations.

In [BSW11], the authors defined a stronger security notion, called simulation-based security to handle such cases. Informally speaking, SIM-security implies that there exists a simulator that, given only the length of m and the function outputs $\{f(m)\}$, but not m itself, can simulate the role of the challenger in the real game. However, SIM-security is so strong that it suffers from the following impossible results:

(1) [BSW11]: SIM-secure FE for P/poly cannot be constructed for unbounded ciphertext queries before a single key query;

(2) [AGVW13]: SIM-secure FE for P/poly cannot be constructed for unbounded key queries before a single ciphertext query.

These impossible results hold even under the random oracle model [AKW18]. Indeed, there are already some constructions for simulation-based FE schemes, but they only work for either bounded ciphertext queries or bounded key queries (which means that the number of ciphertext/key queries must be pre-determined at the Setup phase) [GVW12,GJKS15,ALMT20]. However, for applications in the real world, we need to know how an FE scheme already proven to be SIM-secure performs when handling unbounded ciphertext and key queries. Therefore, a natural question is that: is there a new security notion between IND-security and SIM-security that overcomes the above drawbacks? Intuitively, the new security notion should satisfy the following properties:

- The new security notion must avoid the counter-intuitive example in Example 1.1;
- There must be a construction of FE for P/poly under the new security notion that supports both unbounded ciphertext and key queries;
- Any SIM-secure FE scheme should satisfy the new security notion, so that we are able to discuss the unbounded ciphertext/key security for existing SIM-secure schemes;
- The new security notion should be stronger than IND-security, so that the properties for IND-security also hold for this new security notion.

Next, we show how to define this new security notion by modifying the existing IND-security definition. We note that, the problem in the counter-example can be handled for IND-security, if we loose the restriction on the adversary, such that $\mathcal{A}$ is still allowed to make query f even if f is a one-way permutation. We start from the distributional indistinguishable security (DI-security), first introduced in [AM18], by letting the input of ciphertext queries be a pair of message distributions M_0, M_1, instead of a pair of messages m_0, m_1. For example, let M_0 be the uniform distribution of messages such that the first bit is 0, and M_1 be the uniform distribution of messages such that the first bit is 1. When the adversary submits M_0, M_1 to the challenger, the challenger first randomly chooses a bit b, and then samples $m \leftarrow M_b$.

Now we show that why the counter-example can no longer satisfy the DI-security definition which allows probabilistic ciphertexts. We only need to construct an adversary $\mathcal{A}$ which queries the challenger with a pair of message distributions, instead of a pair of messages, such that $\mathcal{A}$ can break the scheme in Example 1.1.

- Let b be a hardcore predicate of f, we let $\mathcal{A}$ submit two distributions: M_0 is uniform on all strings with $b(m) = 0$, M_1 is uniform on all strings with $b(m) = 1$. (More details can be found in Section 5.)
- Now $\mathcal{A}$ can make queries on f for the one-way permutation f, since $f(M_0)$ and $f(M_1)$ are computationally indistinguishable by the property of hard core predicate.
- $\mathcal{A}$ can calculate $\mathsf{PKE.Dec}(sk_f, ct_{m_b})$ and check its first bit to successfully recover b.

On computational indistinguishability and queries with trapdoors. However, DI-security is not enough, mainly because the usage of computational indistinguishability in its security definition. We point out that it is not easy to include computational indistinguishability *inside* a security game. Below we show the difficulties we discovered while attempting to define a new security notion through probabilistic queries, and that how we solved them. We first give an example, where distributional indistinguishability fails to handle.

Example 1.2. Let PKE be a public key encryption scheme. We explicitly write the randomness used in the encryption algorithm: $\mathsf{PKE.Enc}(pk, m; r)$, let $\mathcal{R}$ be the space of random seeds where $r \leftarrow \mathcal{R}$. We define function class $\mathcal{F}$ as follows:

$$f_{pk}(m, r) \in \mathcal{F} \Leftrightarrow \exists (pk, sk) \leftarrow \mathsf{PKE.KeyGen}, f_{pk}(m, r) = \mathsf{PKE.Enc}(pk, m; r).$$

Let FE be a functional encryption scheme for $\mathcal{F}$, and we consider the security notion which allows message distributions instead of messages.

We construct an adversary $\mathcal{A}$ which makes following queries:

- $\mathcal{A}$ runs $\mathsf{PKE.KeyGen}$ to get (pk, sk).
- Then, $\mathcal{A}$ submits f_{pk} as a key query.
- $\mathcal{A}$ chooses random m_0, m_1, and submits M_0, M_1 which are uniform distributions on $\{m_0\} \times \mathcal{R}$ and $\{m_1\} \times \mathcal{R}$.

Now we have that $f_{pk}(M_0) \leftarrow \mathsf{PKE.Enc}(pk, m_0)$ and $f_{pk}(M_1) \leftarrow \mathsf{PKE.Enc}(pk, m_1)$, hence the two distributions: $f(M_0)$ and $f(M_1)$ are computationally indistinguishable according to the IND-CPA security of PKE. However, the adversary $\mathcal{A}$ can easily distinguish between a ciphertext in $f(M_0)$ and $f(M_1)$ since it holds the secret key sk.

Although in [AM18], the authors constructed DI-secure FE for all polynomial-sized circuits, we show here that DI-security cannot be satisfied for a function family with trapdoors[1], which makes a contradictory. The main reason for this problem is that the notion of computational indistinguishability was not well-defined as in [AM18]: it must be made clear for which party it is to distinguish

[1] We also note that a trapdoor may not only be hidden in the function, but also in the messages. We slightly modify the function in Example 1.2, and let f be defined as: $f(pk, m, r) = \mathsf{PKE.Enc}(pk, m; r)$, and $M_b = \{pk\} \times \{m_b\} \times \mathcal{R}$, so $f(M_0)$ and $f(M_1)$ are distributions with trapdoor, and the trapdoor is hidden in the message distribution, not the function.

between the two distributions, and how much information it has. In Example 1.2, since an adversary may cheat, we cannot let $\mathcal{A}$ be the distinguisher. However, $\mathcal{A}$ is the only one who has the secret key sk, and for any other party, $f(M_0)$ and $f(M_1)$ are indistinguishable, which meets the same difficulties.

This is why we must extend computational indistinguishability into a stronger notion for such a security notion of FE to be well-defined. We informally state what it means by saying that two distributions are strictly computationally indistinguishable even considering trapdoors.

Definition 1.1. *(informal) Let $\mathcal{D}$ be a p.p.t. algorithm that outputs a pair of distributions D_0, D_1, we say that distributions from $\mathcal{D}$ are strictly computationally indistinguishable, if there is no auxiliary string aux corresponding with D_0, D_1 such that (D_0, aux) and (D_1, aux) are computational distinguishable.*

Note that the auxiliary string aux can be viewed as the trapdoor in distributions D_0, D_1.

Now, we revisit Example 1.2. We consider $\mathcal{A}$ as the algorithm which outputs $f_{pk}(M_0)$ and $f_{pk}(M_1)$ as a pair of distributions, and we let sk be the auxiliary string aux. So we can easily construct $\mathcal{B}$ that distinguish between $sk, f_{pk}(M_0)$ and $sk, f_{pk}(M_1)$, thus $f_{pk}(M_0)$ and $f_{pk}(M_1)$ cannot satisfy the condition of strict computational indistinguishability.

This additional auxiliary string has no affect on function families without trapdoors. Just consider PKE, for $(pk, sk) \leftarrow \mathsf{PKE.KeyGen}$, sk can be the auxiliary string if pk is fixed, but if we choose random pk, then there is no such aux as long as PKE has semantic security (we note that aux is not a variable, hence cannot be sk). Otherwise, aux becomes a "master trapdoor" which is unrelated to the randomness used in $\mathsf{PKE.KeyGen}$. Let $\mathcal{A}$ be an adversary which distinguishes between (D_0, aux) and (D_1, aux), then $\mathcal{A}^{aux}(.) = \mathcal{A}(., aux)$ (with aux hardwired in the adversary) can break the semantic security of PKE. We shall give a formal explanation for this case in Section 6.

The need for probabilistic function queries. It seems that everything is right with a new definition for computational indistinguishability. However, since we consider trapdoor functions, we extend Example 1.2 to construct another example just like Example 1.1:

Example 1.3. Let $\mathcal{F}$ be defined as in Example 1.2, and PKE' be a semantic secure public key encryption scheme, then the following FE construction for $\mathcal{F}$ is IND-secure even if we consider probabilistic ciphertext queries:

- $\mathsf{Setup}(1^\lambda)$: Let $(\mathsf{PKE}'.pk^*, \mathsf{PKE}'.sk^*) \leftarrow \mathsf{PKE}'.\mathsf{Setup}(1^\lambda)$, and returns $\mathsf{PK} = \mathsf{PKE}'.pk^*$, $\mathsf{MSK} = \mathsf{PKE}'.sk^*$.
- $\mathsf{Enc}(\mathsf{PK}, (m, r))$: Return $\mathsf{PKE}'.\mathsf{Enc}(\mathsf{PK}, m\|r)$.
- $\mathsf{KeyGen}(\mathsf{MSK}, f)$: Return (MSK, f).
- $\mathsf{Dec}(sk_f, ct_m)$: Parse $sk_f = (\mathsf{MSK}, f)$, decrypt $m\|r = \mathsf{PKE}'.\mathsf{Dec}(\mathsf{MSK}, ct_m)$, and return $f(m, r)$.

However, each sk_f totally leaks m, while $f(m, r)$ does not leak m (since f is the encryption of a semantic secure PKE).

The counter-example above holds, since if we allow the adversary to make even a single query, it can first use $\mathsf{PKE.Setup}$ to generate a pair pk, sk, and then query the function key for $f_{pk} = \mathsf{PKE.Enc}(pk, .; .) \in \mathcal{F}$, hence having the ability to trivially distinguish between $f_{pk}(M_0)$ and $f_{pk}(M_1)$. (To match Definition 1.1 above, we can trivially construct a distinguisher $\mathcal{B}$ with sk as the auxiliary string.) Since the adversary cannot make any queries, the same problem in Example 1.1 also occurs.

In order to avoid such counter-examples, we must allow probabilistic queries not only in the ciphertext query, but also in key queries. Each time the adversary makes a probabilistic key query F, the challenger first samples $f \leftarrow F$, then returns both f and sk_f to the adversary. We construct an adversary $\mathcal{A}$ which makes following queries (including probabilistic key queries):

- We let $\mathcal{A}$ submit two distributions: M_0 is uniform on all strings which first bit is 0, M_1 is uniform on all strings which first bit is 1.
- $\mathcal{A}$ makes a single key query by submitting a distribution F which is uniform on $\mathcal{F}$, and gets $f_{pk} \in \mathcal{F}$.
- $\mathcal{A}$ can calculate $\mathsf{PKE'.Dec}(\mathsf{MSK}, ct_{m_b})$ and check its first bit to successfully recover b.

Since f_{pk} is randomly chosen by the challenger, the adversary $\mathcal{A}$ cannot get the corresponding sk. Here, instead of $f(M_0)$ and $f(M_1)$, we only require that the distributions $F, F(M_0)$ and $F, F(M_1)$ be strictly computationally indistinguishable (sampling from $F, F(M_b)$ means sampling $f \leftarrow F$, $m \leftarrow M_b$ and returning $f, f(m)$.) By our definition, the auxiliary string aux is only related to the distribution $F, F(M_b)$ but independent from how the challenger chooses $f_{pk} \leftarrow F$ (thus independent with either pk or sk).

Now we finished the discussion of rationality for probabilistic queries. We can see that such a security notion can be well-defined, and also avoids the counter-intuitive examples in Example 1.1, 1.2 and 1.3. We call the new security notion *indistinguishability-based security against probabilistic queries* (pIND-security), and show that it is weaker than SIM-security but stronger than IND-security.

Construction of pIND*-secure FE for* P/poly. In this paper, we also give a construction of pIND-secure FE for P/poly, which allows unbounded number of both ciphertext and key queries. Concretely, we show that a fully pIND-secure FE scheme for P/poly can be constructed from a selective IND-secure FE scheme, while the latter can be constructed from both indistinguishability obfuscation [GGH$^+$13] and well-founded assumptions [JLS21,GP21,WW21]. We note that, although the existence of $i\mathcal{O}$ is a strong assumption, unbounded IND-secure FE for P/poly is more than sufficient in constructing $i\mathcal{O}$ [AJ15]. So the FE scheme we construct has stronger security without stronger assumptions.

1.2 Related Works

FE for randomized functionalities. Functional encryption for randomized functionalities (rFE) was introduced in [GJKS15]. The authors gave both SIM-based and IND-based security notions for rFE. We note that, since the authors also used computational indistinguishability to define IND-based security for rFE, the same problems occur as we pointed out in Example 1.2, so that the IND-security of rFE given by [GJKS15] cannot handle trapdoors or public-key encryption. By moving our definition (and construction, using the generic transformation of [AW17]) into the randomized case, these problems can be solved to get a well-defined pIND-based security for rFE.

Distributional Indistinguishability for FE. In [AM18], the authors gave the definition of distributional indistinguishability (DI) for FE, which is previously discussed on garbled circuit and randomized encodings [GHRW14,LPST16], and also gave a construction for DI-secure FE from standard IND-secure FE. Our security definition shares some similarities with theirs, such as allowing the adversary to submit two message distributions, rather than two messages, in the ciphertext query. However, since the DI definition does not allow probabilistic key queries, it still suffers from Example 1.2 and 1.3 which we pointed out above (see also the discussion in Section 6). Moreover, we also give a pIND-secure FE construction for P/poly with adaptive security, while the construction in [AM18] only satisfies selective security.

Function-private public key FE. Probabilistic key queries are also considered in the function-privacy of public key FE [BRS13,PMR19,BCJ+19] as in our work. However, we do not consider function-privacy: In our definition, the function chosen by the challenger is always known to the adversary. It is interesting that whether we can extend our security definition to handle function-privacy.

Other security definitions for FE. There are some other security definitions in early works of functional encryption. In [BO13], the authors gave some new security definitions compared with IND-security, but without a general construction. In [BF13], the authors considered the cases where a trapdoor is hidden in the function family $\mathcal{F}$ supported by FE (instead of function-key queries, which we consider in this paper), and made new security definitions that are even stronger than SIM-security. However, in this paper, we mainly focus on the case where the function family $\mathcal{F}$ is $\mathsf{P/poly}$, so we will not consider this problem.

2 Preliminaries

Notations. $x \leftarrow \chi$ for a distribution χ means that x is sampled from χ. $x \leftarrow X$ for a set X means that x is uniformly random chosen from X. $x \leftarrow \mathcal{X}$ for a p.p.t. algorithm $\mathcal{X}$ means that x is a random output of $\mathcal{X}$, where the abbreviation p.p.t. stands for probabilistic polynomial time. We say that ϵ is negligible in λ, if $\epsilon < 1/\Omega(\lambda^c)$ for any $c > 0$ with sufficiently large λ. $[n]$ for $n \in \mathbb{Z}_+$ is the set $\{1, ..., n\}$.

2.1 Functional Encryption and Security Definitions

Definition 2.1. *A functional encryption scheme* FE *for a function family* $\mathcal{F}$ *consists of the following four algorithms (let* $\mathcal{M}$ *be the message space):*

- $\mathsf{Setup}(1^\lambda)$*: output a pair* $(\mathsf{PK}, \mathsf{MSK})$*.*
- $\mathsf{KeyGen}(\mathsf{MSK}, f)$*: for* $f \in \mathcal{F}$*, output a function key* SK_f*.*
- $\mathsf{Enc}(\mathsf{PK}, m)$*: for* $m \in \mathcal{M}$*, output a ciphertext* CT_m*.*
- $\mathsf{Dec}(\mathsf{SK}_f, \mathsf{CT}_m)$*: output the function value* $f(m)$*.*

FE *is correct if for any* $(\mathsf{PK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)$*,* $\mathsf{SK}_f \leftarrow \mathsf{KeyGen}(\mathsf{MSK},f)$*,* $\mathsf{CT}_m \leftarrow \mathsf{Enc}(\mathsf{PK},m)$*, the probability that* $\mathsf{Dec}(\mathsf{SK}_f, \mathsf{CT}_m) \neq f(m)$ *is negligible.*

Now we give the definition for both IND-security and SIM-security of functional encryption.

Definition 2.2. *An 1-CT adaptive IND-CPA-security game for an FE scheme is defined as follows:*

- Setup*: The challenger runs* $\mathsf{Setup}(1^\lambda)$ *and returns* PK *to the adversary.*
- Phase 1*: The adversary chooses* $f \in \mathcal{F}$ *and gives it to the challenger. The challenger generates* $sk_f \leftarrow \mathsf{KeyGen}(\mathsf{MSK}, f)$ *and returns* sk_f *to the adversary. This can be repeated adaptively for any polynomial times.*
- Challenge*: The adversary chooses two messages of identical length* m_0, m_1 *and gives it to the challenger. The challenger randomly chooses* $b \leftarrow \{0, 1\}$*, generates* $ct \leftarrow \mathsf{Enc}(\mathsf{PK}, m_b)$ *and returns* ct *to the adversary.*
- Phase 2*: Same as Phase 1.*
- Output*: The adversary outputs a bit* b'*, and the winning advantage for the adversary is defined by* $\mathrm{Adv}^{\mathsf{IND}}(\mathcal{A}) = |\Pr(b' = b) - 1/2|$*.*

An adversary $\mathcal{A}$ *is said to be admissible, if for any query* f *in Phase 1 or Phase 2,* $f(m_0) = f(m_1)$*.* FE *is said to be* $\mathsf{ad\text{-}IND}$*-secure if for any p.p.t. admissible adversary* $\mathcal{A}$*,* $\mathrm{Adv}^{\mathsf{IND}}(\mathcal{A})$ *is negligible.*

For the selective IND*-security (*$\mathsf{sel\text{-}IND}$*-security), we require that* $\mathcal{A}$ *submits* m_0, m_1 *to the challenger at the beginning of the game.*

For the many-CT version of the game, we let the adversary submits any polynomial number of pairs of messages in the challenge phase, say $(m_0^1, m_1^1), ..., (m_0^q, m_1^q)$*, such that* $f(m_0^i) = f(m_1^i)$ *for any query* f *and* $i \in [q]$*. In the challenge phase, the challenger samples* $b \leftarrow \{0, 1\}$ *and returns* $(\mathsf{Enc}(\mathsf{PK}, m_b^i))_{i \in [q]}$*.*

It is not hard to show that 1-CT IND-security implies many-CT IND-security through hybrid arguments. In [ABSV15], the authors showed that any $\mathsf{sel\text{-}IND}$ secure FE scheme which is sufficiently expressive can be turned into an $\mathsf{ad\text{-}IND}$ secure FE scheme. Even if the FE scheme is not expressive enough, we can still use the standard complexity leverage method [BB04] to prove the $\mathsf{ad\text{-}IND}$-security, if we assume the sub-exponential hardness of the underlying hardness assumptions.

Next, we give the simulation-based security definition.

Definition 2.3. *Let FE be a functional encryption scheme for a function family* $\mathcal{F}$*. Consider a p.p.t. adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *and a stateful p.p.t. simulator* Sim*. Let* $U_m(.)$ *denote a universal oracle, such that* $U_m(f) = f(m)$*. Consider the following two experiments:*

$\mathsf{Exp}^{\mathsf{real}}_{\mathsf{FE},\mathcal{A}}(1^\lambda)$	$\mathsf{Exp}^{\mathsf{ideal}}_{\mathsf{FE},\mathcal{A}}(1^\lambda)$
1. $(\mathsf{PK},\mathsf{MSK}) \leftarrow \mathsf{FE.Setup}(1^\lambda)$;	1. $\mathsf{PK} \leftarrow \mathrm{Sim}(1^\lambda)$;
2. $(m, st) \leftarrow \mathcal{A}_1^{\mathsf{FE.KeyGen}(\mathsf{MSK},.)}(\mathsf{PK})$;	2. $(m, st) \leftarrow \mathcal{A}_1^{\mathrm{Sim}(.)}(\mathsf{PK})$;
3. $\mathsf{CT} \leftarrow \mathsf{FE.Enc}(\mathsf{PK}, m)$;	3. $\mathsf{CT} \leftarrow \mathrm{Sim}^{U_m(.)}(1^\lambda, 1^{\lvert m\rvert})$;
4. $\alpha \leftarrow \mathcal{A}_2^{\mathsf{FE.KeyGen}(\mathsf{MSK},.)}(\mathsf{PK},\mathsf{CT}, st)$;	4. $\alpha \leftarrow \mathcal{A}_2^{\mathrm{Sim}^{U_m(.)}(.)}(\mathsf{PK},\mathsf{CT}, st)$;
5. Output m, α.	5. Output m, α.

We call a stateful simulator algorithm Sim *admissible if, on each input f,* Sim *makes just a single query to its oracle $U_m(.)$ on f itself. The functional encryption scheme FE is then said to be adaptive simulation-based secure (*ad-SIM*-secure) if there is an admissible stateful p.p.t. simulator* Sim *such that for every p.p.t. adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, the two experiments are computationally indistinguishable.*

For the selective SIM*-security (*sel-SIM*-security), we require that $\mathcal{A}$ submits m to the challenger at the beginning of the game.*

3 Indistinguishability-based Security against Probabilistic Queries

3.1 Definition for pIND Security

First, we give a formal definition for the idea of strict computational indistinguishability introduced in Section 1. We say that a distribution F is efficiently samplable, if there exists a p.p.t. algorithm $\mathcal{F}$ which output follows F. Moreover, sampling from F means to run $\mathcal{F}$ with random seed and fetch its output, so we can use $\mathcal{F}$ to represent F if there is no confusion.

Definition 3.1. *Let $\mathcal{D}$ be a p.p.t. algorithm that outputs a pair of efficiently samplable distributions D_0, D_1. We say that distributions from $\mathcal{D}$ are strictly computationally indistinguishable, if for any p.p.t. algorithm $\mathcal{S}$ which outputs a pair of efficiently samplable distributions S_0, S_1 and an auxiliary string aux, either:*

(1) there exists a p.p.t. algorithm $\mathcal{P}$ which distinguishes between the output of $\mathcal{D}$ and $\mathcal{S}$ (without aux), which means that $\Pr(\mathcal{P}(D_0, D_1) = 1 \vert (D_0, D_1) \leftarrow \mathcal{D}) - \Pr(\mathcal{P}(S_0, S_1) = 1 \vert (S_0, S_1, aux) \leftarrow \mathcal{S})$ *is non-negligible;*

or

(2) there is no p.p.t. algorithm $\mathcal{B}$ which distinguishes between aux, S_0 and aux, S_1, which means that $\Pr(\mathcal{B}(aux, s_0) = 1 \vert s_0 \leftarrow S_0) - \Pr(\mathcal{B}(aux, s_1) = 1 \vert s_1 \leftarrow S_1)$ *must be negligible.*

Without loss of generality, we let the auxiliary string aux contains the two distributions S_0, S_1 (in the form of sampling algorithms), so the distinguisher $\mathcal{B}$ knows exactly the two distributions.

Since there must be no restriction on how $\mathcal{D}$ works, we cannot suppose that *aux* is output by $\mathcal{D}$, hence we introduce another algorithm $\mathcal{S}$ which outputs both the pair of distributions and the auxiliary string *aux*. In most cases, we can simply suppose that $\mathcal{S}$ acts similar as $\mathcal{D}$. However, since no p.p.t. algorithm can determine whether two distributions are equal or even statistical indistinguishable, in order to get a formal definition, we simply let the outputs of $\mathcal{D}$ and $\mathcal{S}$ be computationally indistinguishable.

Next, we use the idea of strict computational indistinguishability to define our new definition for functional encryption.

Definition 3.2. *Given message space $\mathcal{M}$ and function space $\mathcal{F}$, an 1-CT adaptive* pIND-*CPA-security game for an FE scheme is defined as the following:*

- Setup*: The challenger runs* $\mathsf{Setup}(1^\lambda)$ *and returns* PK *to the adversary.*
- Phase 1*: The adversary chooses an efficiently samplable distribution F on the function space $\mathcal{F}$, and gives the sampling algorithm to the challenger. The challenger samples $f \leftarrow F$, generates $sk_f = \mathsf{KeyGen}(\mathsf{MSK}, f)$ and returns f, sk_f to the adversary. This can be repeated adaptively for any polynomial times.*
- Challenge*: The adversary chooses two efficiently samplable distributions M_0, M_1 on the message space $\mathcal{M}$ which contain messages of same length, and gives the sampling algorithms to the challenger. The challenger randomly chooses $b \leftarrow \{0,1\}$, $m \leftarrow M_b$, generates $ct_m \leftarrow \mathsf{Enc}(\mathsf{PK}, m)$ and returns ct_m to the adversary.*
- Phase 2*: Same as Phase 1.*
- Output*: The adversary outputs b', and the winning advantage for the adversary is defined by* $\mathrm{Adv}^{\mathsf{pIND}}(\mathcal{A}) = |\Pr(b' = b) - 1/2|$.

An adversary $\mathcal{A}$ is said to be admissible, if the two distributions $(F_i, F_i(M_0))_{i\in[Q]}$ and $(F_i, F_i(M_1))_{i\in[Q]}$ are strictly computationally indistinguishable, Q is the number of KeyGen queries. FE *is said to be* ad-pIND-*secure if for any p.p.t. admissible adversary $\mathcal{A}$,* $\mathrm{Adv}^{\mathsf{pIND}}(\mathcal{A})$ *is negligible.*

For the selective pIND-*security (*sel-pIND-*security), we require that $\mathcal{A}$ submits M_0, M_1 to the challenger at the beginning of the game.*

For the many-CT version of the game, we let the adversary submits any polynomial number of pairs of messages in the challenge phase, say $(M_0^1, M_1^1), ..., (M_0^q, M_1^q)$, and the admissability is changed to: $(F_i, F_i(M_0^1), ..., F_i(M_0^q))_{i\in[Q]}$ and $(F_i, F_i(M_1^1), ..., F_i(M_1^q))_{i\in[Q]}$ are strictly computationally indistinguishable. In the challenge phase, the challenger samples $b \leftarrow \{0,1\}$ and returns $(\mathsf{Enc}(\mathsf{PK}, m_b^i))_{i\in[q]}$.

We note that when sampling from the distribution $(F_i, F_i(M_b))_{i\in[Q]}$, we only sample once from each F_i and M_b, so the elements from the distribution are in fact dependent with each other.

Now we present a lemma by applying the contrapositive of strict computational indistinguishability onto pIND-security definition. This lemma is useful in the following proofs.

Lemma 3.1. *For an adversary $\mathcal{A}$ in the* pIND*-CPA-security game, we define the trace of $\mathcal{A}$ as:*

$$tr_{\mathcal{A}} = (M_0, M_1, (F_i, f_i, f_i(m))_{i\in[Q]}).$$

Then FE *is* pIND*-secure, if and only if for every p.p.t. $\mathcal{A}$ such that* $\mathrm{Adv}^{\mathsf{pIND}}(\mathcal{A})$ *is non-negligible (not necessarily admissible), there exists a p.p.t. sampling algorithm $\mathcal{T}$ which outputs the distribution:*

$$(aux, \bar{b} \leftarrow \{0,1\}, \bar{m} \leftarrow \bar{M}_{\bar{b}}, \overline{tr} = (\bar{M}_0, \bar{M}_1, (\bar{F}_i, \bar{f}_i, \bar{f}_i(\bar{m}))_{i\in[Q]})),$$

and a p.p.t. algorithm $\mathcal{B}$ where:

- *(1) For any p.p.t. algorithm $\mathcal{P}$,* $\Pr(\mathcal{P}(tr_{\mathcal{A}}) = 1) - \Pr(\mathcal{P}(\overline{tr}) = 1)$ *is negligible;*
- *(2) aux is independent with the following conditional distributions: $\bar{m}|\bar{M}_0, \bar{M}_1$; $\bar{f}_1|\bar{F}_1$;...;$\bar{f}_Q|\bar{F}_Q$ (which can be considered as the randomness used in the choice of $\bar{m}, \bar{f}_1, ..., \bar{f}_Q$);*
- *(3)* $\Pr(\mathcal{B}(aux, \overline{tr}) = \bar{b}) - 1/2$ *is non-negligible.*

Proof. If FE is pIND-secure, then $\mathcal{A}$ with non-negligible advantage must be non-admissible, which means that $(F_i, F_i(M_0))_{i\in[Q]}$ and $(F_i, F_i(M_1))_{i\in[Q]}$ are not strictly computationally indistinguishable.

By the definition of strict computational indistinguishability, there is a sampling algorithm $\mathcal{S}$ which outputs $(\bar{F}_i, \bar{F}_i(\bar{M}_0))_{i\in[Q]}, (\bar{F}_i, \bar{F}_i(\bar{M}_1))_{i\in[Q]}, aux$, such that:

(1) The output of $\mathcal{S}$ except aux are computationally indistinguishable with $(F_i, F_i(M_0))_{i\in[Q]}$, $(F_i, F_i(M_1))_{i\in[Q]}$;

(2) There exists $\mathcal{B}$ which distinguishes between $aux, (\bar{F}_i, \bar{F}_i(\bar{M}_0))_{i\in[Q]}$ and $aux, (\bar{F}_i, \bar{F}_i(\bar{M}_1))_{i\in[Q]}$.

Let $\mathcal{T}$ do the following: first sample S_0, S_1, aux from $\mathcal{S}$, then sample $\bar{b} \leftarrow \{0,1\}$, $\bar{f}_i \leftarrow \bar{F}_i$, $\bar{m} \leftarrow \bar{M}_{\bar{b}}$, and return $(aux, \bar{b}, \bar{m}, \bar{tr} = (\bar{M}_0, \bar{M}_1, (\bar{F}_i, \bar{f}_i, \bar{f}_i(\bar{m}))_{i\in[Q]}))$. Since in the pIND-CPA-security game, the challenger samples from F_i and M_b honestly, we can see that $\bar{tr}$ is computationally indistinguishable with $tr_{\mathcal{A}}$, and aux is independent with the choice of $\bar{f}_i$ and $\bar{m}$, which means that aux is independent with $\bar{m}|\bar{M}_0, \bar{M}_1$; $\bar{f}_1|\bar{F}_1$;...;$\bar{f}_Q|\bar{F}_Q$, hence satisfies all three conditions.

Now, suppose that there exists $\mathcal{T}, \mathcal{B}$ satisfies all three conditions. Let $\mathcal{S}$ runs $\mathcal{T}$ and outputs aux and the two distributions $(\bar{F}_i, \bar{F}_i(\bar{M}_0))_{i\in[Q]}$, $(\bar{F}_i, \bar{F}_i(\bar{M}_1))_{i\in[Q]}$, which are computationally indistinguishable with $(F_i, F_i(M_0))_{i\in[Q]}$, $(F_i, F_i(M_1))_{i\in[Q]}$.

Then, we sample random $\bar{b} \leftarrow \{0,1\}$, $\bar{m} \leftarrow \bar{M}_{\bar{b}}$, $\bar{f}_i \leftarrow \bar{F}_i$, $i \in [Q]$, and let $(aux, (\bar{M}_0, \bar{M}_1, (\bar{F}_i, \bar{f}_i, \bar{f}_i(\bar{m}))_{i\in[Q]}))$ be the input of $\mathcal{B}$, then $\mathcal{B}$ distinguishes the two distributions $(\bar{F}_i, \bar{F}_i(\bar{M}_0))_{i\in[Q]}$, $(\bar{F}_i, \bar{F}_i(\bar{M}_1))_{i\in[Q]}$. By the definition of strict computational indistinguishability, $(F_i, F_i(M_0))_{i\in[Q]}$ and $(F_i, F_i(M_1))_{i\in[Q]}$ cannot be strictly computationally indistinguishable, which means that any adversary $\mathcal{A}$ with non-negligible advantage cannot be admissible. □

For the many-CT version of the game, we define the trace $tr_{\mathcal{A}}$ as:

$$((M_0^i, M_1^i)_{i\in[q]}, (F_k, f_k, f_k(m_1), ..., f_k(m_q))_{k\in[Q]}).$$

It is not hard to show that the result is the same as the 1-CT case.

In a general case, it seems to be hard to determine whether two distributions are strictly computationally indistinguishable, especially with the auxiliary string. But if the function class is a cryptographic primitive such as hash family or public key encryption, we can use its security definition to prove the indistinguishability. We give more details in Section 5 and Section 6.

3.2 Relationship between Different Security Definitions

In this section, we show that pIND-security satisfies the four properties we discussed in Section 1.1, which means that pIND-security can be used to avoid the drawbacks for both SIM-security and IND-security.

Theorem 3.1. *If FE is* SIM*-secure, then FE is* pIND*-secure.*

Proof. Let $\mathcal{A}$ be any pIND adversary, we can construct a SIM adversary $\mathcal{E}^{\mathcal{A}}$ as follows (for the real experiment):

- When $\mathcal{A}$ outputs a key query F, $\mathcal{E}$ chooses $f \leftarrow F$ and gives f to the challenger $\mathcal{C}$. When the challenger returns sk_f, $\mathcal{E}$ returns f, sk_f to $\mathcal{A}$.
- When $\mathcal{A}$ outputs the ciphertext query M_0, M_1, $\mathcal{E}$ first chooses $b \leftarrow \{0,1\}$ and gives $m \leftarrow M_b$ to the challenger $\mathcal{C}$.
- When $\mathcal{C}$ returns a ciphertext CT, CT is returned to $\mathcal{A}$ directly.
- When $\mathcal{A}$ outputs the guess b', $\mathcal{E}$ outputs b' along with the trace: $tr_{\mathcal{A}} = (M_0, M_1, (F_1, f_1, f_1(m)), ..., (F_Q, f_Q, f_Q(m)))$.

Since b' is the same as the output of $\mathcal{A}$ in the $\mathsf{sel\text{-}pIND}$ game, $\Pr(b' = b) - 1/2$ is non-negligible iff $\mathrm{Adv}^{\mathsf{pIND}}(\mathcal{A})$ is non-negligible.

Now consider the ideal experiment with simulator $\mathcal{S}$. Differ from the real experiment, we let the random bit and sampled message be $\tilde{b}, \tilde{m}$, output be $\tilde{b}'$, the trace by $\tilde{tr}_{\mathcal{A}}$, and $\tilde{tr}_{\mathcal{A}}$ is computationally indistinguishable with $tr_{\mathcal{A}}$ by the SIM-based security of the FE scheme. So $\Pr(\tilde{b}' = \tilde{b}) - 1/2$ is non-negligible iff $\mathrm{Adv}^{\mathsf{pIND}}(\mathcal{A})$ is non-negligible.

Using Lemma 3.1, we only need to construct the algorithm $\mathcal{B}$ and a sampling algorithm $\mathcal{T}$ which samples $(aux, \bar{b}, \bar{m}, \overline{tr})$.

Let $\mathcal{T}$ run the ideal experiment with adversary $\mathcal{E}^{\mathcal{A}}$ and simulator $\mathcal{S}^{U_{\tilde{m}}(\cdot)}$. When $\mathcal{S}$ queries $U_{\tilde{m}}(f)$, it directly returns $f(\tilde{m})$ to $\mathcal{S}$ (since $\tilde{m}$ is chosen by $\mathcal{E}^{\mathcal{A}}$), and let $\bar{b} = \tilde{b}, \bar{m} = \tilde{m}, \overline{tr} = \tilde{tr}_{\mathcal{A}}$. Finally, let $aux = (r_{\mathcal{A}}, r_{\mathcal{S}})$, where $r_{\mathcal{A}}, r_{\mathcal{S}}$ are the randomness used in $\mathcal{A}, \mathcal{S}$.

$\mathcal{B}(aux = (r_{\mathcal{A}}, r_{\mathcal{S}}), \overline{tr} = (\bar{M}_0, \bar{M}_1, (\bar{F}_i, \bar{f}_i, \bar{f}_i(\bar{m}))_{i \in [Q]}))$ is constructed from $\mathcal{A}, \mathcal{S}$ with $r_{\mathcal{A}}, r_{\mathcal{S}}$ as their randomness:

- $\mathcal{B}$ first runs $\mathcal{A}$ with randomness $r_{\mathcal{A}}$. When $\mathcal{A}$ outputs the ciphertext query $\tilde{M}_0, \tilde{M}_1$, first check $\tilde{M}_0 = \bar{M}_0, \tilde{M}_1 = \bar{M}_1$, otherwise abort. $\mathcal{S}$ is run with randomness $r_{\mathcal{S}}$.
- When $\mathcal{A}$ outputs the i-th key query $\tilde{F}_i$, first check $\tilde{F}_i = \bar{F}_i$, otherwise abort. Send $\bar{f}_i$ to $\mathcal{S}$, and when $\mathcal{S}$ queries $U_{\bar{m}}$, return $\bar{f}_i(\bar{m})$ to $\mathcal{S}$. Return $\bar{f}_i$ and $sk_{\bar{f}_i}$ generated by $\mathcal{S}$ to $\mathcal{A}$.

- When $\mathcal{S}$ returns a ciphertext CT, CT is returned to $\mathcal{A}$ directly.
- When $\mathcal{A}$ outputs the guess $\tilde{b}'$, return $\bar{b}' = \tilde{b}'$.

It is easy to see that if $\mathcal{B}$ never aborts, the output distribution is the same as $\mathcal{E}^{\mathcal{A}}$ in the ideal game, which means that $\Pr(\mathcal{B}(aux, \overline{tr}) = \bar{b}) - 1/2$ is non-negligible iff $\mathrm{Adv}^{\mathsf{pIND}}(\mathcal{A})$ is non-negligible, hence FE satisfies pIND-security. The non-abortness directly follows from the fact that the queries from $\mathcal{A}$ in both $\mathcal{B}$ and $\mathcal{T}$ are uniquely determined by the same randomness used by $\mathcal{A}, \mathcal{E}, \mathcal{S}$, so that $\tilde{M}_0, \tilde{M}_1, \tilde{F}_1, ..., \tilde{F}_Q$ in $\mathcal{B}$ are exactly the same as $\bar{M}_0, \bar{M}_1, \bar{F}_1, ..., \bar{F}_Q$ contained in $\overline{tr}$ generated from $\mathcal{T}$. Thus we finish the proof. □

Theorem 3.2. *If FE is* pIND*-secure, then FE is* IND*-secure.*

Proof. For any admissible IND adversary $\mathcal{A}$, we construct a pIND adversary $\mathcal{A}'$ as follows:

- When $\mathcal{A}$ submits m_0, m_1, $\mathcal{A}'$ submits M_0, M_1 such that $M_b(m_b) = 1$, $M_b(m') = 0$ for $m' \neq m_b$, $b \in \{0, 1\}$.
- When $\mathcal{A}$ submits f, $\mathcal{A}'$ submits F such that $F(f) = 1$, $F(f') = 0$ for $f' \neq f$, $f(m_0) = f(m_1)$.
- When $\mathcal{A}$ outputs a bit b', $\mathcal{A}'$ also outputs b'.

If $\mathrm{Adv}^{\mathsf{pIND}}(\mathcal{A}')$ is non-negligible, then there exists $\mathcal{B}$, aux and $\overline{tr} = (\bar{M}_0, \bar{M}_1, (\bar{F}_i, \bar{f}_i, \bar{f}_i(\bar{m}))_{i \in [Q]})$, such that $\Pr(\mathcal{B}(aux, \overline{tr}) = \bar{b}) - 1/2$ is non-negligible. Also, $\overline{tr}$ is indistinguishable from $tr_{\mathcal{A}}$, which means that sampling from $\bar{M}_0, \bar{M}_1$ and $\bar{F}_i$ always outputs fixed values $\bar{m}_0, \bar{m}_1, \bar{f}_i$, where $\bar{f}_i(\bar{m}_0) = \bar{f}_i(\bar{m}_1)$ for $i \in [Q]$ (otherwise $tr_{\mathcal{A}}$ and $\overline{tr}$ can easily be distinguished). So $\overline{tr}$ is independent from $\bar{b}$, also aux is independent from $\bar{b}$ by Lemma 3.1. Thus $\Pr(\mathcal{B}(aux, \overline{tr}) = \bar{b}) - 1/2 = 0$, which makes a contradiction.

So for every $\mathcal{A}'$ defined above, $\mathrm{Adv}^{\mathsf{pIND}}(\mathcal{A}')$ is negligible, which means that $\mathrm{Adv}^{\mathsf{IND}}(\mathcal{A})$ is negligible. □

Now we show that 1-CT pIND-security implies many-CT pIND-security, so that our new definition can really bypass the impossible result in [BSW11].

Theorem 3.3. *If FE is 1-CT* pIND*-secure, then FE is many-CT* pIND*-secure.*

Proof. We define a sequence of games:

G_i: the first i ciphertext queries always choose $m_i \leftarrow M_0$ despite whether b is. Suppose that $\mathcal{A}$ makes a total of q ciphertext queries, then G_0 is the original game, and the advantage for $\mathcal{A}$ in G_q is always 0.

If the advantage for $\mathcal{A}$ in G_0 is non-negligible, then there exists $i \in [q]$ such that the advantage of $\mathcal{A}$ to distinguish between G_{i-1} and G_i is non-negligible. Then we construct an 1-CT pIND adversary $\mathcal{A}_i$ as follows:

- For $(M_0^j, M_1^j)_{j \in [q]}$, we define $M_0(x), M_1(x)$ be two sampling algorithms with a single input $x \in [q]$, which sample from M_0^x and M_1^x. Thus $M_0(x), M_1(x)$ contains all information about $(M_0^j, M_1^j)_{j \in [q]}$.

- When $\mathcal{A}$ submits $(M_0^j, M_1^j)_{j\in[q]}$, submit $M_0(i), M_1(i)$ to the challenger and get the ciphertext CT; sample $m_j \leftarrow M_0^j$ for $j < i$, $m_j \leftarrow M_1^j$ for $j > i$, let $CT_j \leftarrow \mathsf{Enc}(\mathsf{PK}, m_j)$ for $j \neq i$ and $CT_i = CT$, return $(CT_1, ..., CT_q)$ to $\mathcal{A}$.
- When $\mathcal{A}$ submits a key query F, directly pass it to the challenger and return (f, sk_f) to $\mathcal{A}$.
- When $\mathcal{A}$ outputs b', output b'.

So $\mathrm{Adv}^{\mathsf{pIND}}(\mathcal{A}_i)$ is non-negligible. By the 1-CT pIND security, there exist $\mathcal{T}_i$ and $\mathcal{B}_i$ satisfying Lemma 3.1, let $(aux, \overline{tr}_i)$ be sampled by $\mathcal{T}_i$, we write $\overline{tr}_i = (\bar{M}_0(i), \bar{M}_1(i), (\bar{F}_k, \bar{f}_k, \bar{f}_k(\bar{m}_i))_{k\in[Q]})$, and since $\bar{M}_0(i), \bar{M}_1(i)$ are indistinguishable from $M_0(i), M_1(i)$, we write the q pairs of distributions extracted from $\bar{M}_0(i), \bar{M}_1(i)$ as $(\bar{M}_0^j, \bar{M}_1^j)_{j\in[q]}$.

We first sample $(\bar{m}_j \leftarrow \bar{M}_b^j)_{j\neq i}$ and calculate $(\bar{f}_k(\bar{m}_j))_{j\neq i, k\in[Q]}$. Let $\mathcal{B}$ proceed the same as $\mathcal{B}_i$ except that we let the input $\overline{tr} = ((\bar{M}_0^j, \bar{M}_1^j)_{j\in[q]}, (\bar{F}_k, \bar{f}_k, \bar{f}_k(\bar{m}_1), ..., \bar{f}_k(\bar{m}_q))_{k\in[Q]})$. So $(aux, \bar{b}, (\bar{m}_i)_{i\in[q]}, \overline{tr})$ can be sampled by $\mathcal{T}_i$ with slight modification, aux is independent from the choices of $(\bar{m}_j)_{j\in[q]}$ and $(\bar{f}_k)_{k\in[Q]}$, and $\Pr(\mathcal{B}(aux, \overline{tr}) = \bar{b}) - 1/2$ is non-negligible, since the outputs of $\mathcal{B}$ and $\mathcal{B}_i$ are the same. Thus we finish the proof. □

4 Fully pIND-secure FE from IND-based FE schemes

We already show that pIND-secure FE can support unbounded ciphertext. The problem remaining is to show the existence of adaptive pIND-secure FE scheme for $\mathsf{P/poly}$ which supports unbounded key, so that our new security definition can avoid the [AGVW13] impossibility result. We show that the [ABSV15] generic transformation, which transforms selective IND-secure FE schemes into adaptive IND-secure ones, can be extended into pIND-security. In fact, we prove a result stronger than expected: we can transform any selective IND-secure FE scheme into an adaptive pIND-secure FE scheme.

Technical Overview. In [ABSV15], the authors constructed an adaptive IND-secure FE scheme for any function class $\mathcal{F}$ (even if $\mathcal{F} = \mathsf{P/poly}$) from an IND-secure private-key FE scheme for $\mathcal{F}$ with 1-CT query and unbounded key queries, and a "sufficiently expressive" selective IND-secure FE scheme, here private-key FE means that the encryption algorithm uses master secret key instead of master public key.

To prove the existence of IND-secure private-key FE with 1-CT and unbounded key queries, [ABSV15] relies on several results in the literature. First, in [GVW12], the authors constructed a 1-key, unbounded-CT SIM-secure private-key FE scheme for P/poly, which is also a 1-key, unbounded-CT IND-secure private-key FE scheme. In [BS15], the authors gave the generic transformation from private-key FE to function-private private-key FE, here function-private means that the function f is hidden from the adversary even given the function key sk_f. (A private-key FE without function-privacy is also called message-private.) Then, one can swap KeyGen and Enc in a function-private private-key

FE with 1-key and unbounded-CT, to obtain a private-key FE with unbounded-key and 1-CT.

The same method can easily be extended to pIND-security. Similar with [ABSV15], we can construct an adaptive pIND-secure FE scheme from pIND-secure private-key FE scheme and IND-secure (public-key) FE scheme, and by Theorem 3.1 in this paper (extended to private-key settings), we can show the existence of a 1-key, unbounded-CT message-private pIND-secure private-key FE scheme for $\mathsf{P/poly}$. What left for us is to transform a pIND-secure message-private private-key FE scheme into a pIND-secure function-private private-key FE scheme.

The idea of this construction is similar to the one in [BS15], but more complicated since we consider probabilistic queries. The [BS15] construction used two symmetric keys k, k' to hide the two functions f_0, f_1 correspondingly in both the message-private and function-private game. However, in our pIND-secure settings, in message-private game, the adversary learns an exact function f, while in function-private game, the adversary learns only two distributions F_0 and F_1 (see the formal definition below). So we need three keys k, k', k'' to encrypt f, F_0, F_1 correspondingly, and an additional game to switch between them, while the other parts of the proof is similar to [BS15].

Finally, combining all components together, we can construct an adaptive pIND-secure FE scheme for $\mathsf{P/poly}$.

Before further discussions, first we give formal definitions for both message-private and function-private private-key functional encryption with pIND-security.

Definition 4.1. *A private-key functional encryption scheme* skFE *for a function family* $\mathcal{F}$ *consists of the following four algorithms (let* $\mathcal{M}$ *be the message space):*

- $\mathsf{Setup}(1^\lambda)$*: output the master secret key* MSK.
- $\mathsf{KeyGen}(\mathsf{MSK}, f)$*: for* $f \in \mathcal{F}$*, output a function key* SK_f.
- $\mathsf{Enc}(\mathsf{MSK}, m)$*: for* $m \in \mathcal{M}$*, output a ciphertext* CT_m.
- $\mathsf{Dec}(\mathsf{SK}_f, \mathsf{CT}_m)$*: output the function value* $f(m)$.

FE *is correct if for any* $\mathsf{MSK} \leftarrow \mathsf{Setup}(1^\lambda)$*,* $\mathsf{SK}_f \leftarrow \mathsf{KeyGen}(\mathsf{MSK}, f)$*,* $\mathsf{CT}_m \leftarrow \mathsf{Enc}(\mathsf{MSK}, m)$*, the probability of* $\mathsf{Dec}(\mathsf{SK}_f, \mathsf{CT}_m) \neq f(m)$ *is negligible.*

Next, we define the (message-private) pIND-based security and function-private pIND-based security for private-key FE schemes.

Definition 4.2. *Given message space* $\mathcal{M}$ *and function space* $\mathcal{F}$*, a q-CT (or unbounded-CT), Q-key (or unbounded-key) adaptive (message-private) pIND-CPA-security game for a private-key FE scheme is defined as the following:*

- *Setup: The challenger runs* $\mathsf{Setup}(1^\lambda)$ *to get* MSK*, and randomly samples a bit* $b \leftarrow \{0, 1\}$.
- *Query Phase: The adversary can adaptively makes the following two types of queries:*

- *Key Query: The adversary chooses a p.p.t. sampling algorithm F which output is in $\mathcal{F}$, and gives it to the challenger. The challenger samples $f \leftarrow F$, generates $sk_f = \mathsf{KeyGen}(\mathsf{MSK}, f)$ and returns f, sk_f to the adversary. This can be repeated adaptively for any polynomial times.*
- *Ciphertext Query: The adversary chooses two p.p.t. sampling algorithms M_0, M_1 which outputs are in $\mathcal{M}$ and gives them to the challenger. The challenger randomly chooses $m \leftarrow M_b$, generates $ct_m \leftarrow \mathsf{Enc}(\mathsf{PK}, m)$ and returns ct_m to the adversary.*

The number of key queries is bounded by Q or unbounded; the number of ciphertext queries is bounded by q or unbounded.

– *Output: The adversary outputs b', and the winning advantage for the adversary is defined by $Adv^{\mathsf{pIND}}(\mathcal{A}) = |\Pr(b' = b) - 1/2|$.*

Let q, Q be the number of ciphertext queries and key queries, we write the i-th key query and the chosen function by F^i, f^i, the j-th ciphertext query and the chosen message by M_0^j, M_1^j, m^j.

An adversary $\mathcal{A}$ is said to be admissible, if the two distributions $(F^i, F^i(M_0^1), ..., F^i(M_0^q))_{i \in [Q]}$ and $(F^i, F^i(M_1^1), ..., F^i(M_1^q))_{i \in [Q]}$ are strictly computationally indistinguishable. We say that skFE is a (message-private) pIND-secure private-key FE if for any p.p.t. admissible adversary $\mathcal{A}$, $\mathrm{Adv}^{\mathsf{pIND}}(\mathcal{A})$ is negligible.

Lemma 4.1. *For a (message-private) pIND adversary $\mathcal{A}$ for a secret key FE scheme, define the trace of $\mathcal{A}$ as:*

$$tr_{\mathcal{A}} = ((M_0^j, M_1^j)_{j \in [q]}, (F^i, f^i, f^i(m^1), ..., f^i(m^q))_{i \in [Q]}).$$

Then skFE is a (message-private) pIND-secure private-key FE, if and only if for every p.p.t. $\mathcal{A}$ such that $Adv^{\mathsf{pIND}}(\mathcal{A})$ is non-negligible, there exists a p.p.t. algorithm $\mathcal{T}$ which outputs the following distribution:

$$(aux, \bar{b}, (\bar{m}_j)_{j \in [q]}, \overline{tr} = ((\bar{M}_0^j, \bar{M}_1^j,)_{j \in [q]}, (\bar{F}^i; \bar{f}^i, \bar{f}^i(\bar{m}^1), ..., \bar{f}^i(\bar{m}^q))_{i \in [Q]})),$$

where $\bar{b} \leftarrow \{0,1\}, \bar{m}^j \leftarrow \bar{M}_{\bar{b}}^j$ for $j \in [q]$, $\bar{f}^i \leftarrow \bar{F}^i$ for $i \in [Q]$, and a p.p.t. algorithm $\mathcal{B}$, which satisfies:

– *(1) For any p.p.t. algorithm $\mathcal{P}$, $\Pr(\mathcal{P}(tr_{\mathcal{A}}) = 1) - \Pr(\mathcal{S}(\overline{tr}) = 1)$ is negligible;*
– *(2) aux is independent with the following conditional distributions: $\bar{m}^j | \bar{M}_0^j, \bar{M}_1^j$, $j \in [q]$; $\bar{f}^i | \bar{F}^i$, $i \in [Q]$;*
– *(3) $\Pr(\mathcal{B}(aux, \overline{tr}) = \bar{b}) - 1/2$ is non-negligible.*

Proof. The proof is similar to Lemma 3.1 and we omit the details. □

Definition 4.3. *Given message space $\mathcal{M}$ and function space $\mathcal{F}$, a q-CT (or unbounded-CT), Q-key (or unbounded-key) adaptive function-private pIND-CPA-security game for a private-key FE scheme is defined as the following:*

– *Setup: The challenger runs $\mathsf{Setup}(1^\lambda)$ to get MSK, and randomly samples a bit $b \leftarrow \{0,1\}$.*

- *Query Phase: The adversary can adaptively makes the following two types of queries:*
    - *Key Query: The adversary chooses two p.p.t. sampling algorithms* F_0, F_1 *which output is in* $\mathcal{F}$*, and gives it to the challenger. The challenger samples* $f \leftarrow F_b$*, generates* $sk_f = \mathsf{KeyGen}(\mathsf{MSK}, f)$ *and returns* sk_f *to the adversary. This can be repeated adaptively for any polynomial times.*
    - *Ciphertext Query: The adversary chooses two p.p.t. sampling algorithms* M_0, M_1 *which outputs are in* $\mathcal{M}$ *and gives them to the challenger. The challenger randomly chooses* $m \leftarrow M_b$*, generates* $ct_m \leftarrow \mathsf{Enc}(\mathsf{PK}, m)$ *and returns* ct_m *to the adversary.*

  The number of key queries is bounded by Q *or unbounded; the number of ciphertext queries is bounded by* q *or unbounded.*
- *Output: The adversary outputs* b'*, and the winning advantage for the adversary is defined by* $Adv^{\mathsf{pIND}}(\mathcal{A}) = |\Pr(b' = b) - 1/2|$.

Let q, Q *be the number of ciphertext queries and key queries, we write the* i*-th key query and the chosen function by* F_0^i, F_1^i, f^i*, the* j*-th ciphertext query and the chosen message by* M_0^j, M_1^j, m^j.

An adversary $\mathcal{A}$ *is said to be admissible, if the two distributions* $(F_0^i(M_0^1), ..., F_0^i(M_0^q))_{i\in[Q]}$ *and* $(F_1^i(M_1^1), ..., F_1^i(M_1^q))_{i\in[Q]}$ *are strictly computationally indistinguishable. We say that* skFE *is a function-private pIND-secure private-key FE if for any p.p.t. admissible adversary* $\mathcal{A}$*,* $\mathrm{Adv}^{\mathsf{pIND}}(\mathcal{A})$ *is negligible.*

Lemma 4.2. *For a function-private pIND adversary* $\mathcal{A}$ *for a secret key FE scheme, define the trace of* $\mathcal{A}$ *as:*

$$tr_{\mathcal{A}} = ((M_0^j, M_1^j)_{j\in[q]}, (F_0^i, F_1^i, f^i(m^1), ..., f^i(m^q))_{i\in[Q]}).$$

Then skFE *is a function-private pIND-secure private-key FE, if and only if for every p.p.t.* $\mathcal{A}$ *such that* $Adv^{\mathsf{pIND}}(\mathcal{A})$ *is non-negligible, there exists a p.p.t. algorithm* $\mathcal{T}$ *which outputs the following distribution:*

$$(aux, \bar{b}, (\bar{m}^j)_{j\in[q]}, \overline{tr} = ((\bar{M}_0^j, \bar{M}_1^j)_{j\in[q]}, (\bar{F}_0^i, \bar{F}_1^i, \bar{f}^i(\bar{m}^1), ..., \bar{f}^i(\bar{m}^q))_{i\in[Q]})),$$

where $\bar{b} \leftarrow \{0,1\}, \bar{m}^j \leftarrow \bar{M}_{\bar{b}}^j$ *for* $j \in [q]$*,* $\bar{f}^i \leftarrow \bar{F}_{\bar{b}}^i$ *for* $i \in [Q]$*, and a p.p.t. algorithm* $\mathcal{B}$*, which satisfies:*

- *(1) For any p.p.t. algorithm* $\mathcal{S}$*,* $\Pr(\mathcal{S}(tr_{\mathcal{A}}) = 1) - \Pr(\mathcal{S}(\overline{tr}) = 1)$ *is negligible;*
- *(2) aux is independent with the following conditional distributions:* $\bar{m}^j | \bar{M}_0^j, \bar{M}_1^j$*,* $j \in [q]$*;* $\bar{f}^i | \bar{F}_0^i, \bar{F}_1^i$*,* $i \in [Q]$*;*
- *(3)* $\Pr(\mathcal{B}(aux, \overline{tr}) = \bar{b}) - 1/2$ *is non-negligible.*

Proof. The proof is similar to Lemma 3.1 and we omit the details. □

We give a lemma on the existence of private-key pIND-secure FE.

Lemma 4.3. *There exists a private-key pIND-secure FE with 1-CT and unbounded key for* $\mathsf{P/poly}$*, assuming the existence of one-way functions.*

Proof. By Theorem 3.1 (which can also be applied to private-key FE schemes), we can show that the SIM-secure private-key FE scheme for $\mathsf{P/poly}$ with 1-key and unbounded-CT queries in [GVW12] is also pIND-secure. If we can lift this scheme into a function-private pIND-secure private-key FE scheme, we can simply swap the KeyGen and Enc algorithms to obtain a private-key pIND-secure FE with unbounded-key and 1-CT for $\mathsf{P/poly}$.

The lifting is similar to the one in [BS15]. Let skFE be the pIND-secure message-private private-key FE scheme, Sym be a symmetric encryption scheme, PRF be a pseudo-random function family. We construct the pIND-secure function-private private-key FE as follows:

- $\mathsf{Setup}(1^\lambda)$: Generate three symmetric encryption keys $k, k', k'' \leftarrow \mathsf{Sym.KeyGen}(1^\lambda)$, let $\mathsf{skFE.MSK} \leftarrow \mathsf{skFE.Setup}(1^\lambda)$. Return $\mathsf{MSK} = (k, k', k'', \mathsf{skFE.MSK})$.
- $\mathsf{KeyGen}(\mathsf{MSK}, f)$: Let $\tilde{f}$ be defined as: $\tilde{f}(m, r) = f(m)$. Let $c = \mathsf{Sym.Enc}(k, \tilde{f})$, $c' = \mathsf{Sym.Enc}(k', \tilde{f})$, $c'' = \mathsf{Sym.Enc}(k'', \tilde{f})$. Return $\mathsf{skFE.KeyGen}(\mathsf{skFE.MSK}, g_{c,c',c''}$, where for any c_1, c_2, c_3, $g_{c_1,c_2,c_3}(m, k_1, k_2, k_3, r)$ is defined as follows:
  - If $k_1 \neq \perp$, let $f \leftarrow \mathsf{Sym.Dec}(k_1, c_1)$, return $f(m; r)$.
  - Else if $k_2 \neq \perp$, let $f \leftarrow \mathsf{Sym.Dec}(k_2, c_2)$, return $f(m; r)$.
  - Else if $k_3 \neq \perp$, let $f \leftarrow \mathsf{Sym.Dec}(k_3, c_3)$, return $f(m; r)$.
  - Else return $\perp$.
- $\mathsf{Enc}(\mathsf{MSK}, m)$: Sample a random seed r and return $ct \leftarrow \mathsf{skFE.Enc}(\mathsf{skFE.MSK}, (m, k, \perp, \perp, r))$.
- $\mathsf{Dec}(sk, ct)$: return $\mathsf{skFE.Dec}(sk, ct)$.

Now we prove the security of the construction above through a hybrid of games.

Game 0 is the original game.

In Game 1, the challenger first samples a uniform random seed r^*, and for each ciphertext query, returns $\mathsf{skFE.Enc}(\mathsf{MSK}, (m, k, \perp, \perp, r^*))$ instead of $\mathsf{skFE.Enc}(\mathsf{MSK}, (m, k, \perp, \perp, r))$ for a freshly sampled r. Game 0 and Game 1 are indistinguishable from the pIND-security of skFE. (Note that the distribution of messages in different ciphertext queries share the same r^*.)

In Game 2, when the adversary makes a key query, instead of sampling $f \leftarrow F_b$ using a random seed, the challenger samples two seeds s_0, s_1, and uses $\mathsf{PRF}(r^*, s_b)$ as the seed to sample $f \leftarrow F_b$. Game 1 and Game 2 are indistinguishable from the pseudorandomness of PRF.

In Game 3, for each key query, let $\tilde{c}' = \mathsf{Sym.Enc}(k', G_{F_0,s_0})$, $\tilde{c}'' = \mathsf{Sym.Enc}(k'', G_{F_1,s_1})$, returns $\mathsf{skFE.KeyGen}(\mathsf{MSK}, g_{c,\tilde{c}',\tilde{c}''})$ instead of $\mathsf{skFE.KeyGen}(\mathsf{MSK}, g_{c,c',c''})$, where $G_{F_b,s_b}(m, r)$ is defined as:

- Sample $f \leftarrow F_b$ using the seed $\mathsf{PRF}(r, s_b)$;
- Return $f(m)$.

Game 2 and Game 3 are indistinguishable from the security of Sym.

In Game 4, we change the ciphertext into $\mathsf{skFE.Enc}(\mathsf{MSK}, m, \perp, k', \perp, r^*)$ for $b = 0$ and $\mathsf{skFE.Enc}(\mathsf{MSK}, m, \perp, \perp, k'', r^*)$ for $b = 1$. Since $f_b(m) = G_{F_b}(m, r^*)$, we can see that the trace for skFE is the same in Game 3 and Game 4, so Game 3 and Game 4 are indistinguishable from the security of skFE.

In Game 5, for each key query, let $\tilde{c} = \mathsf{Sym.Enc}(k, \perp)$, returns $\mathsf{skFE.KeyGen}(\mathsf{MSK}, g_{\tilde{c},\tilde{c}',\tilde{c}''})$ instead of $\mathsf{skFE.KeyGen}(\mathsf{MSK}, g_{c,\tilde{c}',\tilde{c}''})$. Game 4 and Game 5 are indistinguishable from the security of Sym. Note that the function $g_{\tilde{c},\tilde{c}',\tilde{c}''}$ is the same for $b = 0$ and $b = 1$ in Game 5.

Now in Game 5, if $\mathcal{A}$ is an adversary for the function-private scheme with non-negligible advantage, there is an adversary $\mathcal{A}'$ which is an adversary for skFE with non-negligible advantage. By the pIND-based security of skFE, there exist a sampling algorithm $\mathcal{T}'$ and an algorithm $\mathcal{B}'$ with non-negligible advantage which satisfy Lemma 4.1. We write the output of $\mathcal{T}'$ as:

$$(aux, \bar{b}', (\bar{m'}^j)_{j\in[q]}, \overline{tr}' = ((\bar{M'}^j_0, \bar{M'}^j_1)_{j\in[q]}, (\bar{F'}^i, \bar{f'}^i, \bar{f'}^i(\bar{m'}^1), ..., \bar{f}^i(\bar{m'}^q))_{i\in[Q]})).$$

Since $\overline{tr}'$ is indistinguishable with $tr_{\mathcal{A}'}$, so elements in both $\overline{tr}'$ and $tr_{\mathcal{A}'}$ has the same structure, so we can write $\bar{m'}^j = (\bar{m}^j, \perp, \bar{k}', \perp, \bar{r}^*)$ for $\bar{b}' = 0$ and $\bar{m'}^j = (\bar{m}^j, \perp, \perp, \bar{k}'', \bar{r}^*)$ for $\bar{b}' = 1$, $\bar{f'}^i = g_{\bar{c},\bar{c}',\bar{c}''}$ where $\bar{c}, \bar{c}', \bar{c}''$ are Sym ciphertexts of $\perp, G_{\bar{F}_0,\bar{s}_0}, G_{\bar{F}_1,\bar{s}_1}$ defined as above.

Without loss of generalization, we suppose that $\bar{k}', \bar{k}''$ are contained in aux, since $\bar{k}', \bar{k}''$ are predetermined and independent with the choice of either queried message or function.

Now we construct $\mathcal{T}$ and $\mathcal{B}$ from $\mathcal{T}'$ and $\mathcal{B}'$.

$\mathcal{T}$ does the following:

- Call $\mathcal{T}'$ to get $\bar{h}', aux, \bar{b}', (\bar{m'}^j)_{j\in[q]}, \overline{tr}'$;
- Extract $\bar{M}^j_0, \bar{M}^j_1, \bar{m}^j$ from $\bar{M'}^j_0, \bar{M'}^j_1, \bar{m'}^j$, $\bar{F}^i_0, \bar{F}^i_1$ from $\bar{f'}^i$;
- Sample $\bar{f}^i \leftarrow \bar{F}^i_{\bar{b}'}$;
- Return $aux, \bar{b}', (\bar{m}^j)_{j\in[q]}, \overline{tr} = ((\bar{M}^j_0, \bar{M}^j_1)_{j\in[q]}, (\bar{F}^i_0, \bar{F}^i_1, \bar{f}^i(\bar{m}^1), ..., \bar{f}^i(\bar{m}^q))_{i\in[Q]}))$.

$\mathcal{B}(aux, \overline{tr})$ does the following:

- For each $\bar{M}^j_b$, $j \in [q]$, $b \in \{0, 1\}$, sampling from $\bar{M'}^j_b$ does the following:
  - Sample $\bar{m} \leftarrow \bar{M}^j_b$;
  - If $j = 1$, sample a random seed $\bar{r}^*$, otherwise use the same $\bar{r}^*$ as in $j' < j$[2];
  - If $b = 0$, return $\bar{m'}^j = (\bar{m}, \perp, \bar{k}', \perp, \bar{r}^*)$, otherwise return $\bar{m'}^j = (\bar{m}, \perp, \perp, \bar{k}'', \bar{r}^*)$.
- For each $\bar{F}^i_0$ and $\bar{F}^i_1$, sampling from $\bar{F'}^i$ does the following:
  - Sample two random seeds $\bar{s}_0, \bar{s}_1$;
  - Let $G_{\bar{F}_0,\bar{s}_0}$ and $G_{\bar{F}_1,\bar{s}_1}$ be defined as above, and $\bar{c}' = \mathsf{Sym.Enc}(\bar{k}', G_{\bar{F}_0,\bar{s}_0})$, $\bar{c}'' = \mathsf{Sym.Enc}(\bar{k}'', G_{\bar{F}_1,\bar{s}_1})$;

[2] Here we allows different distributions $\bar{M'}^j_b$ to include the same randomness $\bar{r}^*$, which means that there is a shared inner state between these sampling algorithms. We note that SIM-secure FE implies pIND-secure FE even considering stateful ciphertext queries like this, so it will not affect the validity of the proof.

  - Return $\bar{f'}^i = g_{\bar{c},\bar{c}',\bar{c}''}$.
- Call $\mathcal{B}'(aux, ((\bar{M'}^j_0, \bar{M'}^j_1)_{j\in[q]}, (\bar{F'}^i; \bar{f'}^i, \bar{f'}^i(\bar{m'}^1), ..., \bar{f'}^i(\bar{m'}^q))_{i\in[Q]}))$ to get the output.

It is not hard to see that $\mathcal{B}$ calls $\mathcal{B}'$ exactly with $(aux, \overline{tr}')$, where $\overline{tr}'$ is defined as above, and we already know that $\Pr(\mathcal{B}'(aux, \overline{tr}') = \bar{b}) - 1/2$ is non-negligible. So we successfully construct $\mathcal{T}$ and $\mathcal{B}$ satisfies Lemma 4.2. Thus the new scheme is a function-private pIND-secure private-key FE scheme. □

Theorem 4.1. *There exists a construction for* $\mathsf{ad\text{-}pIND}$*-secure FE for* $\mathsf{P/poly}$ *from* $\mathsf{sel\text{-}IND}$*-secure FE for* $\mathsf{P/poly}$ *assuming the existence of one-way functions.*

Proof. We simply write down the [ABSV15] construction here, and give a high level proof. The details are similar to the $\mathsf{ad\text{-}IND}$-security proof in [ABSV15]. Given the following primitives:

- A $\mathsf{sel\text{-}IND}$ secure public-key FE scheme for $\mathsf{P/poly}$ Sel;
- An $\mathsf{ad\text{-}pIND}$ secure 1-CT private-key FE scheme for $\mathsf{P/poly}$ OneCT;
- A symmetric encryption scheme with pseudorandom ciphertexts Sym;
- A pseudorandom function family PRF.

The adaptive scheme Ad is constructed as follows:

- $\mathsf{Setup}(1^\lambda)$: Sample $(\mathsf{Sel.PK}, \mathsf{Sel.MSK}) \leftarrow \mathsf{Sel.Setup}(1^\lambda)$, and return $\mathsf{PK} = \mathsf{Sel.PK}$, $\mathsf{MSK} = \mathsf{Sel.MSK}$.
- $\mathsf{KeyGen}(\mathsf{MSK}, f)$: Sample $C_E \leftarrow \{0,1\}^{l_1(\lambda)}$, $\tau \leftarrow \{0,1\}^{l_2(\lambda)}$, return $sk_f \leftarrow \mathsf{Sel.KeyGen}(\mathsf{Sel.MSK}, G_{f,C_E,r})$, $G_{f,C_E,r}(\mathsf{OneCT.MSK}, \mathsf{K}, \mathsf{Sym.K}, \beta)$ defined as follows:
  - If $\beta = 1$, output $\mathsf{Sym.Dec}(\mathsf{Sym.K}, C_E)$;
  - Otherwise, output $\mathsf{OneCT.KeyGen}(\mathsf{OneCT.MSK}, f; \mathsf{PRF_K}(\tau))$.
- $\mathsf{Enc}(\mathsf{PK}, m)$: Output $\mathsf{CT} = (\mathsf{CT}_0 \leftarrow \mathsf{OneCT.Enc}(\mathsf{OneCT.MSK}, m), \mathsf{CT}_1 \leftarrow \mathsf{Sel.Enc}(\mathsf{Sel.MPK}, (\mathsf{OneCT.MSK}, \mathsf{K}, 0^\lambda, 0))$.
- $\mathsf{Dec}(sk_f, \mathsf{CT})$: Output $\mathsf{OneCT.Dec}(\mathsf{Sel.Dec}(sk_f, \mathsf{CT}_1), \mathsf{CT}_0)$.

The $\mathsf{ad\text{-}pIND}$-security of this construction can be proved by a hybrid of games. Let Game 0 be the original $\mathsf{pIND\text{-}CPA}$ game.

In Game 1, C_E is replaced by $\mathsf{Sym.Enc}(\mathsf{Sym.K}^*, sk_f \leftarrow \mathsf{OneCT.KeyGen}(\mathsf{OneCT.MSK}, f; \mathsf{PRF_K}(\tau))$ for random $\mathsf{Sym.K}^*$. Game 0 and Game 1 are indistinguishable from the security of Sym.

In Game 2, CT_1 is replaced by $\mathsf{Sel.Enc}(\mathsf{Sel.MPK}, (0^\lambda, 0^\lambda, \mathsf{Sym.K}^*, 1))$. Since any adversary $\mathcal{A}$ distinguishing Game 1 and Game 2 makes only deterministic ciphertext queries to Sel, we can see that Game 1 and Game 2 are indistinguishable from the IND-security of Sel.

In Game 3, $\mathsf{PRF_K}(\tau)$ is replaced by a truly random R. Game 2 and Game 3 are indistinguishable by the pseudorandomness of PRF.

We see that any adversary $\mathcal{A}$ which has non-negligible advantage in Game 3 has also a non-negligible advantage in the $\mathsf{ad\text{-}pIND\text{-}CPA}$ game of OneCT. Then if OneCT is $\mathsf{ad\text{-}pIND}$-secure, we can construct $\mathcal{B}$ and the input distribution $(h', aux, tr_{\mathcal{B}})$ for $\mathcal{A}$ which satisfies Lemma 3.1, hence Ad is $\mathsf{ad\text{-}pIND}$-secure. □

5 Application of pIND-secure FE: Hashing a Secret Value

Next, we introduce a specific application scenario, which can be constructed from pIND-secure FE. This application is inspired by Example 1.1, the counterexample for IND-based security. We show that how we can use pIND-secure FE to output the hash of a secret value. Like blind signature [Cha82], we name this new primitive “blind hash”. We first give its syntax, which is similar to the syntax of functional encryption.

Definition 5.1. *A blind hash system consists of the following algorithms:*

- $\mathsf{Setup}(1^\lambda, 1^n, 1^k)$*: output the public key* pk *and the main secret key* msk*. We require that* $n \geq k$*.*
- $\mathsf{HashGen}(msk, h)$*: for a hash function* $h : \{0,1\}^n \to \{0,1\}^k$*, output its blinded version* H*.*
- $\mathsf{Enc}(pk, m)$*: output the encrypted message* c*.*

The blind hash system is called correct, if for $(pk, msk) \leftarrow \mathsf{Setup}(\lambda)$*,* $H \leftarrow \mathsf{HashGen}(msk, h)$*,* $c \leftarrow \mathsf{Enc}(pk, m)$*, the probability of* $H(c) \neq h(m)$ *is negligible.*

In this definition, we restrict the input length of the hash function to be n instead of arbitrary length, in order for Enc to be well-defined. We can choose large enough n, and pad any string with length $n' < n$ into a string of length n.

We require the one-wayness of a blind hash system.

Definition 5.2. *A blind hash system* $(\mathsf{Setup}, \mathsf{HashGen}, \mathsf{Enc})$ *is called one-way, if for any p.p.t. adversary* $\mathcal{A}$ *and a set* $\mathcal{S}$ *of (polynomial number of) universal one-way hash families, the winning advantage of the following game is negligible:*

- *Setup: The challenger runs* $\mathsf{Setup}(1^\lambda)$ *and returns* pk *to the adversary.*
- *Phase 1: Each time the adversary submits a universal one-way hash family* $\mathcal{H} \in \mathcal{S}$*, the challenger samples* $h \leftarrow \mathcal{H}$*, and returns* $(h, H \leftarrow \mathsf{HashGen}(msk, h))$ *to the adversary. This can be repeated for any polynomial numbers of times.*
- *Challenge: The challenger chooses* $m \leftarrow \mathcal{M}$*, and returns* $\mathsf{Enc}(pk, m)$ *to the adversary.*
- *Phase 2: Same as Phase 1.*
- *Guess: The adversary outputs* m'*. The winning advantage of* $\mathcal{A}$ *is defined by* $\Pr(m' = m)$*.*

In this definition, we give a set of universal one-way hash families outside the game instead of letting them to be chosen by the adversary, since both the adversary and the challenger are p.p.t., hence cannot have the ability to determine whether a hash family is universal one-way.

Before we give our construction for the blind hash system, we first introduce the Goldreich-Levin hardcore predicate for one-way functions.

Definition 5.3. *A polynomial time computable predicate* b *is a hardcore predicate of a function* $f : \{0,1\}^n \to \{0,1\}^k$*, if for any p.p.t. algorithm* $\mathcal{P}$*,* $|\Pr_{m \leftarrow \{0,1\}^n} (\mathcal{P}(f(m)) = b(m)) - 1/2|$ *is negligible.*

Lemma 5.1 (Goldreich-Levin Theorem). *If $f : \{0,1\}^n \to \{0,1\}^n$ is a one-way function, then $b(m,r) = \langle m,r \rangle$ is a predicate of the function $g : \{0,1\}^{2n} \to \{0,1\}^{2n}$, $g(m\|r) = f(m)\|r$.*

Now we are ready to construct our blind hash system. Given a functional encryption scheme FE, the blind hash system is constructed as follows:

- $\mathsf{Setup}(1^\lambda)$: Run $\mathsf{FE.Setup}(1^\lambda)$ and output the public key pk and the main secret key msk.
- $\mathsf{HashGen}(msk, h)$: Let $\bar{h}$ be the function which pads the output of h from k bits into n bits (by filling 0s). Let function g_h be defined as: $g_h(m\|r) = \bar{h}(m)\|r$. Calculate $sk_{g_h} \leftarrow \mathsf{FE.KeyGen}(msk, g_h)$. Let the blinded hash $H(c)$ be defined as:
  - Let $t \leftarrow \mathsf{FE.Dec}(sk_{g_h}, c)$;
  - Output the first k bits of t.
- $\mathsf{Enc}(pk, m)$: Let $r \leftarrow \{0,1\}^n$, output $\mathsf{FE.Enc}(pk, m\|r)$.

Theorem 5.1. *Let* FE *be pIND-based secure, then the construction above is a one-way blind hash system.*

Proof. Let $\mathcal{G}_\mathcal{H}$ be the p.p.t. algorithm that first samples $h \leftarrow \mathcal{H}$ and then outputs g_h (as define above), and M_0 (resp. M_1) be a p.p.t. algorithm that outputs a random string $m\|r \in \{0,1\}^{2n}$ where $\langle m,r \rangle = 0$ (resp. $\langle m,r \rangle = 1$). For each adversary $\mathcal{A}'$ attacks the one-wayness of the blind hash system, we consider any pIND adversary $\mathcal{A}$ for FE which makes specific queries as follows:

- When $\mathcal{A}'$ submits a query $\mathcal{H}_j$ in Phase 1 or Phase 2, $\mathcal{A}$ submits $\mathcal{G}_{\mathcal{H}_j}$, and gets sk_{g_h} from inside the blinded hash function H.
- At the challenge phase, $\mathcal{A}$ submits (M_0, M_1) to the challenger, and gets the challenge ciphertext of $\mathcal{A}'$.

We do not restrict the way that $\mathcal{A}$ gives its outputs b'.

By the definition of pIND-based security, if $Adv(\mathcal{A})$ is non-negligible, there exists a sampling algorithm $\mathcal{T}$ and an algorithm $\mathcal{B}$, where $(aux, \bar{b}, \bar{m}, \overline{tr}) \leftarrow \mathcal{T}$, $\overline{tr}$ is computationally indistinguishable from $tr_\mathcal{A}$, and $\Pr(\mathcal{B}(aux, \overline{tr}) = \bar{b}) - 1/2$ is non-negligible, where $\overline{tr}$ takes the form as:

$$\overline{tr} = (\bar{M}_0, \bar{M}_1, (\bar{F}_i, \bar{f}_i, \bar{f}_i(\bar{m}))_{i \in [Q]}).$$

Since M_0, M_1 are fixed and each F_i in $tr_\mathcal{A}$ is chosen only from a pre-determined polynomial size set $\{\mathcal{G}_\mathcal{H}\}_{\mathcal{H} \in \mathcal{S}}$, we can see that the computational indistinguishability between $tr_\mathcal{A}$ and $\overline{tr}$ implies that $\bar{M}_0 = M_0$, $\bar{M}_1 = M_1$, and $\bar{F}_i = \mathcal{G}_\mathcal{H}$ for some $\mathcal{H} \in \mathcal{S}$. We also write $\bar{f}_i$ as $g_{\bar{h}_i}$ where $\bar{h}_i \in \mathcal{H}$, thus $\bar{f}_i(\bar{m}) = g_{\bar{h}_i}(m\|r)$ for some m, r, and $\bar{b} = \langle m, r \rangle$.

Since aux is independent with the choice of $\bar{f}_i$ and $\bar{m}$, we define $\mathcal{B}_i(g_{\bar{h}_i}(m\|r)) := \mathcal{B}_i(aux, \overline{tr})$, so by a standard hybrid argument, $\Pr(\mathcal{B}(aux, \overline{tr}) = \bar{b}) - 1/2$ is non-negligible, only if there exists $\mathcal{B}_i$, such that $\Pr(\mathcal{B}_i(\{g_{\bar{h}_i}(m\|r)\}_{i \in [q]}) = \langle m, r \rangle) -$

1/2 is non-negligible. However, from Goldreich-Levin Theorem, $\langle m, r\rangle$ is a hardcore predicate for $g_{\bar{h}_i}(m\|r)$, and since each $\bar{h}_i$ is independently chosen from universal hash families, we have that $\{g_{\bar{h}_i}(m\|r)\}_{i\in[q]}$ are independent, so $\langle m, r\rangle$ is also a hardcore predicate for $g(m\|r) := g_{\bar{h}_1}(m\|r)\|...\|g_{\bar{h}_q}(m\|r)$, which means that $\Pr(\mathcal{B}'(\{g_{\bar{h}_i}(m\|r)\}_{i\in[q]}) = \langle m, r\rangle) - 1/2$ must be negligible. So we have that $\Pr(\mathcal{B}(aux, \overline{tr}) = \bar{b}) - 1/2$ is also negligible, hence $Adv(\mathcal{A})$ is negligible.

We know that if a function is one-one, then having a hardcore predicate implies one-wayness. Since the advantage of $\mathcal{A}$ is negligible, if we consider the function $f(m\|r) = sk_{h_1}\|...\|sk_{h_q}\|ct$, $ct \leftarrow \mathsf{FE.Enc}(pk, m\|r)$, which is a one-one function, we see that $\langle m, r\rangle$ is also its hardcore predicate, so $f(m\|r)$ is one-way. Since r can be directly generated from $\mathsf{FE.Dec}(sk_i, ct)$ given any sk_i, we can see that $f(m\|r)$ is one-way according to the input m, hence the advantage for $\mathcal{A}'$ is also negligible. Thus we finish the proof. □

The construction from pIND-secure FE to blind hash systems are quite straightforward. Since pIND-secure FE can be constructed from IND-secure FE schemes, blind hash systems can be constructed from IND-secure FE schemes.

However, we show that the same method in this section cannot be used to directly construct blind hash systems from IND-secure FE: let h be a collision-resistant hash function, and construct the hash family $\mathcal{H}$ be: $\{h_k : h_k(m) := h(k\|m)\}$. So if $\mathcal{A}$ make an admissible query, which means that $h_{k_1}(m) = h_{k_2}(m)$, it finds a collision for h, which contradicts the security of h, so $\mathcal{A}$ cannot make any queries. So if the construction above uses an IND-based FE scheme, the one-way property cannot be satisfied, like what we showed in Example 1.1.

Also, For SIM-based secure FE schemes, as it was proven in [AGVW13], there is no unbound-key SIM-based secure FE schemes supporting one-way functions, so SIM-based FE schemes for $\mathcal{H}$ cannot be constructed, hence it is impossible to directly construct blind hash systems from SIM-based FE schemes.

6 Application of pIND-secure FE: Semi-universal Proxy Re-encryption

Now we give another application scenario which can be constructed from pIND-secure FE but not other security definitions. We consider proxy re-encryption (PRE) schemes [BBS98], which can be used to transform a ciphertext encrypted under a delegator key into one encrypted under a delegatee key, without leaking the plaintext. However, in most existing PRE constructions, the delegator encryption scheme and the delegatee encryption scheme must be the same: they cannot re-encrypt a given ciphertext into another ciphertext under another public-key encryption scheme.

In [DN21], the authors introduced universal proxy re-encryption, and gave their construction from probabilistic $i\mathcal{O}$, where both the delegator and the delegatee can be arbitrary PKE schemes. Here, we discuss a weaker version of universal PRE, where only the delegatee ciphertext can be encrypted by arbitrary PKE schemes, and we call it semi-universal PRE. We now show that semi-universal PRE can be constructed by pIND-secure FE for P/poly. We note that pIND-secure FE for P/poly can be constructed from IND-secure FE for P/poly as we proved in Section 4, thus our construction of semi-universal PRE also has a weaker requirement than the existence of $pi\mathcal{O}$ in the construction of universal PRE [DN21] (we note that even constructing $i\mathcal{O}$ requires sub-exponential hardness IND-secure FE for P/poly).

We first give the syntax definition of semi-universal PRE.

Definition 6.1. *A semi-universal PRE consists of the following algorithms:*

- $\mathsf{KeyGen}(1^\lambda)$*: Output a public-key/secret-key pair* (pk, sk)*.*
- $\mathsf{Enc}(pk, m)$*: For a public key generated from* $\mathsf{KeyGen}(1^\lambda)$*, output a ciphertext* ct *for* m*.*
- $\mathsf{ReKeyGen}(sk_f, PKE, pk_t)$*: Let* sk_f *be generated from* $\mathsf{KeyGen}(1^\lambda)$ *and* pk_t *be a public key of the PKE scheme PKE. This algorithm outputs a re-encryption key* $rk_{f\to t}$*.*
- $\mathsf{ReEnc}(rk_{f\to t}, ct)$*: Let* ct *be a ciphertext encrypted by* pk_f *, output a new ciphertext encrypted by* pk_t*.*
- $\mathsf{Dec}(sk_f, ct)$*: For a ciphertext* $ct \leftarrow \mathsf{Enc}(pk_f, m)$*, output the corresponding message* m*.*

Let $\mathsf{PKE} = \mathsf{PKE.KeyGen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec}$ *be any public key encryption scheme. A semi-universal PRE scheme is correct, if for* $(pk_f, sk_f) \leftarrow \mathsf{KeyGen}(1^\lambda)$*,* $ct_f \leftarrow \mathsf{Enc}(pk_f, m)$*, both: (1)* $\mathsf{Dec}(sk_f, ct_f) = m$ *except for a negligible probability; (2)* $(pk_t, sk, t) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$*,* $rk_{f\to t} \leftarrow \mathsf{ReKeyGen}(sk_f, PKE, pk_t)$*, the ciphertext* $ct_t \leftarrow \mathsf{ReEnc}(rk_{f\to t}, ct_f)$ *satisfies:* $\mathsf{PKE.Dec}(sk_t, ct_t) = m$ *except for a negligible probability.*

We only define a weaker version of the single-hop security of PRE, where each delegator key must be generated at the setup phase, and allows only static corruption. For simplicity reason, we assume that the delegatee PKE scheme is always different from the delegator PKE scheme (which is a pIND-secure FE scheme as in our construction).

Definition 6.2. *For a semi-universal PRE* $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{ReKeyGen}, \mathsf{ReEnc})$*, let* $\mathcal{P}$ *be a set of semantic secure PKE scheme, and for any* $\mathsf{PKE} \in \mathcal{P}$*,* $\mathsf{Enc} \neq \mathsf{PKE.Enc}$*. The weak-CRA security of the semi-functional PRE is satisfied if for every adversary* $\mathcal{A}$*, the winning advantage of the following game is negligible:*

- *Setup: The adversary asks the challenger to run* $\mathsf{Setup}(1^\lambda)$ *for any polynomial numbers of times to get* $(\widehat{pk_i}, \widehat{sk_i})_{i\in[q]}$*. The challenger returns* $(\widehat{pk_i})_{i\in[q]}$ *to the adversary. Let* L *be an empty list.*
- *Phase 1: The adversary can make one of the following types of queries in arbitrary sequence:*
  - *Type 1: The adversary submits* $\mathsf{PKE} \in \mathcal{P}$*. The challenger generates* $(pk_{|L|+1}, sk_{|L|+1}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$*, adds the pair* $(\mathsf{PKE}; pk_{|L|+1})$ *into* L*, and returns* $pk_{|L|+1}$ *to the adversary.*

- *Type 2: The adversary submits* $\mathsf{PKE} \in \mathcal{P}$. *The challenger generates* $(pk_{|L|+1}, sk_{|L|+1}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$, *adds the pair* $(\mathsf{PKE}; pk_{|L|+1})$ *into* L, *and returns* $(pk_{|L|+1}, sk_{|L|+1})$ *to the adversary.*
- *Type 3: The adversary submits* $\widehat{pk_i}, i \in [q]$, *and* $(\mathsf{PKE}, pk_j) \in L$. *The challenger runs* $\mathsf{ReKeyGen}(\widehat{sk_i}, \mathsf{PKE}, pk_j)$ *and returns* $rk_{i\to j}$ *to the adversary if* $rk_{i\to j}$ *has not been generated before.*

These queries can be repeated adaptively.

– *Challenge: The adversary submits* $\widehat{pk_{i^*}}, i^* \in [q]$ *and a pair of messages* (m_0, m_1), *providing that for each Type 3 query which returns* $rk_{i^*\to j}$ *for some* j, pk_j *is generated from a Type 1 query. The challenger chooses* $b \leftarrow \{0, 1\}$, *and returns* $\mathsf{Enc}(\widehat{pk_{i^*}}, m_b)$ *to the adversary.*

– *Phase 2: Same as Phase 1, under the restriction that for all Type 3 queries* $(\widehat{pk_{i^*}}, \mathsf{PKE}, pk_j)$, pk_j *is generated from a Type 1 query.*

– *Guess: The adversary outputs* b'. *The winning advantage of* $\mathcal{A}$ *is defined by* $|Pr(b' = b) - 1/2|$.

Now we construct a weak-CRA secure semi-universal PRE from a pIND-secure functional encryption scheme FE. Let PRF be a pseudorandom function.

– $\mathsf{KeyGen}(1^\lambda)$: Output $(pk, sk) \leftarrow \mathsf{FE.Setup}(1^\lambda)$.

– $\mathsf{Enc}(pk, m)$: Sample a random seed r, and output $ct \leftarrow \mathsf{FE.Enc}(pk, m\|r)$.

– $\mathsf{ReKeyGen}(sk_f, \mathsf{PKE}, pk_t)$: Sample a random key K, and let $F(m\|r) := \mathsf{PKE.Enc}(pk_t, m; PRF(K, r))$. Return $\mathsf{FE.KeyGen}(sk_f, F)$.

– $\mathsf{ReEnc}(rk_{f\to t}, ct)$: Output $\mathsf{FE.Dec}(rk_{f\to t}, ct)$.

– $\mathsf{Dec}(sk, ct)$: Let $sk_{ID} \leftarrow \mathsf{FE.KeyGen}(sk, ID)$ where $ID(m) = m$, then output $\mathsf{FE.Dec}(sk_{ID}, ct)$.

Before we prove the security of the PRE scheme, we first give a lemma to show that the auxiliary string has no effect in distinguishing a PKE ciphertext with random key.

Lemma 6.1. *Let* PKE *be a public key encryption scheme with semantic security. For a pair of messages* m_0, m_1, *any p.p.t. algorithm* $\mathcal{B}$ *and auxiliary string* aux, *let* $(pk, sk) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$, $c_0 \leftarrow \mathsf{PKE.Enc}(pk, m_0)$, $c_1 \leftarrow \mathsf{PKE.Enc}(pk, m_1)$. *Then* $\Pr(\mathcal{B}(aux, pk, c_0) = 1) - \Pr(\mathcal{B}(aux, pk, c_1) = 1)$ *is negligible.*

Proof. Let $\mathcal{B}_{aux}(.,.)$ be the algorithm $\mathcal{B}(aux, ., .)$. We construct a IND-CPA adversary $\mathcal{A}$ for PKE, which submits m_0, m_1 as the challenge messages, and runs $\mathcal{B}_{aux}(pk, c)$ to get the output, then by the semantic security of PKE, the advantage of $\mathcal{A}$ is negligible, hence $\Pr(\mathcal{B}(aux, pk, c_0) = 1) - \Pr(\mathcal{B}(aux, pk, c_1) = 1)$ is negligible. □

We note that the adversary $\mathcal{A}$ in the proof above is non-uniform, so the scheme PKE must be secure against non-uniform adversaries, which is a rather standard assumption.

Theorem 6.1. *Let* FE *be pIND-based secure, then the construction above satisfies weak-CRA security.*

Proof. Given an adversary $\mathcal{A}'$ for the semi-universal PRE game. We construct a pIND adversary $\mathcal{A}$ for FE as follows:

In the setup phase, suppose that the adversary asks the challenger to run $\mathsf{Setup}(1^\lambda)$ for q times. $\mathcal{A}$ randomly choose $i' \leftarrow [q]$, and asks for the FE public key pk. Let $\widehat{pk_{i'}} := pk$. For $i \neq i'$, the challenger runs $\mathsf{FE.Setup}(1^\lambda)$ to get $(\widehat{pk_i}, \widehat{sk_i})$. $\mathcal{A}$ returns $(\widehat{pk_i})_{i\in[q]}$.

When $\mathcal{A}'$ generates a Type 1 query PKE, let F_{PKE} be the following algorithm:

– Run $(\mathsf{PKE}.pk, \mathsf{PKE}.sk) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$;
– Sample a random key K and return the function f where $f(m\|r) := \mathsf{PKE.Enc}$ $(\mathsf{PKE}.pk, m; \mathsf{PRF}(K, r))$.

$\mathcal{A}$ submits a KeyGen query F_{PKE}, and gets (f, sk_f), where f contains $\mathsf{PKE}.pk$. Let $pk_{|L|+1} = \mathsf{PKE}.pk$, store $sk_{f_{|L|+1}} := sk_f$. Return $pk_{|L|+1}$ and add $(\mathsf{PKE}, pk_{|L|+1})$ into L.

For a Type 2 query PKE, return $(pk_{|L|+1}, sk_{|L|+1}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$ directly while adding $(\mathsf{PKE}, pk_{|L|+1})$ into L.

For a Type 3 query $(\widehat{pk_i}, \mathsf{PKE}, pk_j)$, if $i = i'$ and pk_j is generated from Type 2 queries, then return a random guess $b' \leftarrow \{0,1\}$ and abort. If $i = i'$ and pk_j is generated from Type 1 queries, return $rk_{i\to j} := sk_{f_j}$ (generated in Type 1 queries). If $i \neq i'$, return $rk_{i\to j} \leftarrow \mathsf{FE.KeyGen}(\widehat{sk_i}, f_j)$.

In the challenge phase, if $\mathcal{A}'$ queries for $i^* \neq i'$, then return a random guess $b' \leftarrow \{0,1\}$ and abort. Otherwise, let $M_b, b \in \{0,1\}$ be the algorithm that first randomly samples r and returns $m_b\|r$. Submit (M_0, M_1) and get the ciphertext ct, return ct to $\mathcal{A}'$.

Finally, return the guess b' from $\mathcal{A}'$.

We can see that $\mathcal{A}$ does not abort if and only if $i^* = i'$. Since q is polynomial, the non-aborting probability $1/q$ is non-negligible, so if the advantage of $\mathcal{A}'$ is non-negligible, the advantage of $\mathcal{A}$ is also non-negligible. By the definition of pIND-based security, there exists a sampling algorithm $\mathcal{T}$ and an algorithm $\mathcal{B}$ satisfies the definition. We write the output of $\mathcal{T}$ as $aux, \bar{b}, \bar{m}, \overline{tr} = (\bar{M}_0, \bar{M}_1, (\bar{F}_i, \bar{f}_i, \bar{f}_i(\bar{m}))_{i\in[Q]})$.

Since each key query of $\mathcal{A}$ is from a polynomial size set $\{F_{\mathsf{PKE}} : \mathsf{PKE} \in \mathcal{S}\}$ and a ciphertext query M_b samples $m_b\|r$, $b \leftarrow \{0,1\}$, we can see that as long as $\overline{tr}$ is computationally indistinguishable from $tr_{\mathcal{A}}$, $\bar{F}_i \in \{F_{\mathsf{PKE}} : \mathsf{PKE} \in \mathcal{S}\}$ and $\bar{M}_0, \bar{M}_1$ samples $\bar{m}_0\|r, \bar{m}_1\|r$ for fixed $\bar{m}_0, \bar{m}_1$ and random r. So we rewrite $\bar{f}_i(\bar{m})$ as $\bar{f}_i(\bar{m}_{\bar{b}}\|r) = \mathsf{PKE.Enc}(pk, \bar{m}_{\bar{b}}; \mathsf{PRF}(K, r))$, which is indistinguishable from $\mathsf{PKE.Enc}(pk, \bar{m}_{\bar{b}})$ by the pseudorandomness of PRF.

Since aux is independent from the choice of $\bar{f}_i$, it is also independent from the choice of pk, by Lemma 6.1, we have that $\Pr(\mathcal{B}(aux, (..., \bar{F}_i, \bar{f}_i, \bar{f}_i(\bar{m}_0\|r), ...)) = 1) - \Pr(\mathcal{B}(aux, (..., \bar{F}_i, \bar{f}_i, \bar{f}_i(\bar{m}_1\|r), ...)) = 1)$ is negligible. By a standard hybrid argument, we have that $\Pr(\mathcal{B}(aux, \overline{tr}) = 1|\bar{b} = 0) - \Pr(\mathcal{B}(aux, \overline{tr}) = 1|\bar{b} = 1)$ is negligible, hence $\Pr(\mathcal{B}(aux, \overline{tr}) = \bar{b}) - 1/2$ is negligible, which makes a contradiction. So the advantage of $\mathcal{A}'$ is negligible, thus we finish the proof. □

By a discussion similar to Section 5, we can see that SIM-based and IND-based FE schemes cannot be used to construct semi-universal PRE schemes directly. We also point out that why semi-universal PRE cannot be directly constructed from rFE [GJKS15]. The SIM-based secure rFE in [GJKS15] supports only selective security, hence cannot satisfy our security definition. (We note that adaptively SIM-based secure rFE also suffers from the impossible result of [AGVW13].) For IND-based secure rFE, the authors require that each post-challenge key query f, where f is a probabilistic function, satisfies that $f(m_0)$ and $f(m_1)$ are statically indistinguishable, rather than computationally indistinguishable, hence cannot be satisfied if $m_0 \neq m_1$ and f is $\mathsf{PKE.Enc}(pk, .)$ for a PKE scheme PKE. Even if we consider only pre-challenge key queries, where the authors only require that $f(m_0)$ and $f(m_1)$ are computationally indistinguishable, it still cannot handle the case where f is $\mathsf{PKE.Enc}(pk, .)$ since the adversary may hold the secret key sk corresponding to pk. The same thing happens for the distributional indistinguishability definition [AM18], which also requires $f(m_0)$ and $f(m_1)$ to be computationally indistinguishable.

7 Conclusion and Future Works

In this paper, we define a new security notion for FE: indistinguishability-based security against probabilistic queries (pIND-security). We justify our security notion from the following four points: (1) Our pIND-security is strictly between the classical SIM-security and IND-security; (2) Our pIND-security has both 1-CT to many-CT and selective to adaptive reductions; (3) We give a construction of fully secure FE for $\mathsf{P/poly}$ which satisfies pIND-security; (4) We give applications that can be directly constructed from pIND-secure FE schemes, but cannot be constructed from SIM-secure or IND-secure FE schemes in a same way.

We believe that our new definition has more potential applications than what we showed in this paper. We also hope that this new security notion can be used to simplify the construction from FE to $i\mathcal{O}$, hence pushing $i\mathcal{O}$ further into practical.

Acknowledgement. This work is partially supported by the National Key R&D Program of China (No. 2020YFA0712300) and the National Natural Science Foundation of China (No. 62202294, No. 62272294).

References

[ABSV15] Prabhanjan Ananth, Zvika Brakerski, Gil Segev, and Vinod Vaikuntanathan. From selective to adaptive security in functional encryption. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 657–677. Springer, 2015.

[AGVW13] Shweta Agrawal, Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption: New perspectives and lower bounds. In *Advances in Cryptology - CRYPTO 2013*, pages 500–518, 2013.

[AJ15] Prabhanjan Ananth and Abhishek Jain. Indistinguishability obfuscation from compact functional encryption. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 308–326. Springer, 2015.

[AKW18] Shashank Agrawal, Venkata Koppula, and Brent Waters. Impossibility of simulation secure functional encryption even with random oracles. In Amos Beimel and Stefan Dziembowski, editors, *Theory of Cryptography - 16th International Conference, TCC 2018, Panaji, India, November 11-14, 2018, Proceedings, Part I*, volume 11239 of *Lecture Notes in Computer Science*, pages 659–688. Springer, 2018.

[ALMT20] Shweta Agrawal, Benoît Libert, Monosij Maitra, and Radu Titiu. Adaptive simulation security for inner product functional encryption. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *Public-Key Cryptography - PKC 2020 - 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, May 4-7, 2020, Proceedings, Part I*, volume 12110 of *Lecture Notes in Computer Science*, pages 34–64. Springer, 2020.

[AM18] Shweta Agrawal and Monosij Maitra. FE and io for turing machines from minimal assumptions. In Amos Beimel and Stefan Dziembowski, editors, *Theory of Cryptography - 16th International Conference, TCC 2018, Panaji, India, November 11-14, 2018, Proceedings, Part II*, volume 11240 of *Lecture Notes in Computer Science*, pages 473–512. Springer, 2018.

[AW17] Shashank Agrawal and David J. Wu. Functional encryption: Deterministic to randomized functions from simple assumptions. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part II*, volume 10211 of *Lecture Notes in Computer Science*, pages 30–61, 2017.

[BB04] Dan Boneh and Xavier Boyen. Efficient selective-id secure identity-based encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 2004.

[BCJ$^+$19] James Bartusek, Brent Carmer, Abhishek Jain, Zhengzhong Jin, Tancrède Lepoint, Fermi Ma, Tal Malkin, Alex J. Malozemoff, and Mariana Raykova. Public-key function-private hidden vector encryption (and more). In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part III*, volume 11923 of *Lecture Notes in Computer Science*, pages 489–519. Springer, 2019.

[BF13] Manuel Barbosa and Pooya Farshim. On the semantic security of functional encryption schemes. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *Public-Key Cryptography - PKC 2013 - 16th International Conference on*

Practice and Theory in Public-Key Cryptography, Nara, Japan, February 26 - March 1, 2013. Proceedings, volume 7778 of *Lecture Notes in Computer Science*, pages 143–161. Springer, 2013.

[BO13] Mihir Bellare and Adam O'Neill. Semantically-secure functional encryption: Possibility results, impossibility results and the quest for a general definition. In Michel Abdalla, Cristina Nita-Rotaru, and Ricardo Dahab, editors, *Cryptology and Network Security - 12th International Conference, CANS 2013, Paraty, Brazil, November 20-22. 2013. Proceedings*, volume 8257 of *Lecture Notes in Computer Science*, pages 218–234. Springer, 2013.

[BRS13] Dan Boneh, Ananth Raghunathan, and Gil Segev. Function-private identity-based encryption: Hiding the function in functional encryption. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 461–478. Springer, 2013.

[BS15] Zvika Brakerski and Gil Segev. Function-private functional encryption in the private-key setting. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, volume 9015 of *Lecture Notes in Computer Science*, pages 306–324. Springer, 2015.

[BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In *Theory of Cryptography Conference*, pages 253–273. Springer, 2011.

[Cha82] David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology: Proceedings of CRYPTO '82, Santa Barbara, California, USA, August 23-25, 1982*, pages 199–203. Plenum Press, New York, 1982.

[DN21] Nico Döttling and Ryo Nishimaki. Universal proxy re-encryption. In Juan A. Garay, editor, *Public-Key Cryptography - PKC 2021 - 24th IACR International Conference on Practice and Theory of Public Key Cryptography, Virtual Event, May 10-13, 2021, Proceedings, Part I*, volume 12710 of *Lecture Notes in Computer Science*, pages 512–542. Springer, 2021.

[GGH$^+$13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 40–49. IEEE Computer Society, 2013.

[GHRW14] Craig Gentry, Shai Halevi, Mariana Raykova, and Daniel Wichs. Outsourcing private RAM computation. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 404–413. IEEE Computer Society, 2014.

[GJKS15] Vipul Goyal, Abhishek Jain, Venkata Koppula, and Amit Sahai. Functional encryption for randomized functionalities. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, volume 9015 of *Lecture Notes in Computer Science*, pages 325–351. Springer, 2015.

[GP21] Romain Gay and Rafael Pass. Indistinguishability obfuscation from circular security. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 736–749. ACM, 2021.

[GVW12] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption with bounded collusions via multi-party computation. In *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, volume 7417 of *Lecture Notes in Computer Science*, pages 162–179. Springer, 2012.

[JLS21] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 60–73. ACM, 2021.

[KLM$^+$18] Sam Kim, Kevin Lewi, Avradip Mandal, Hart Montgomery, Arnab Roy, and David J. Wu. Function-hiding inner product encryption is practical. In Dario Catalano and Roberto De Prisco, editors, *Security and Cryptography for Networks - 11th International Conference, SCN 2018, Amalfi, Italy, September 5-7, 2018, Proceedings*, volume 11035 of *Lecture Notes in Computer Science*, pages 544–562. Springer, 2018.

[LPST16] Huijia Lin, Rafael Pass, Karn Seth, and Sidharth Telang. Output-compressing randomized encodings and applications. In Eyal Kushilevitz and Tal Malkin, editors, *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, volume 9562 of *Lecture Notes in Computer Science*, pages 96–124. Springer, 2016.

[MSH$^+$19] Tilen Marc, Miha Stopar, Jan Hartman, Manca Bizjak, and Jolanda Modic. Privacy-enhanced machine learning with functional encryption. In Kazue Sako, Steve A. Schneider, and Peter Y. A. Ryan, editors, *Computer Security - ESORICS 2019 - 24th European Symposium on Research in Computer Security, Luxembourg, September 23-27, 2019, Proceedings, Part I*, volume 11735 of *Lecture Notes in Computer Science*, pages 3–21. Springer, 2019.

[O'N10] Adam O'Neill. Definitional issues in functional encryption. *IACR Cryptol. ePrint Arch.*, page 556, 2010.

[PMR19] Sikhar Patranabis, Debdeep Mukhopadhyay, and Somindu C. Ramanna. Function private predicate encryption for low min-entropy predicates. In Dongdai Lin and Kazue Sako, editors, *Public-Key Cryptography - PKC 2019 - 22nd IACR International Conference on Practice and Theory of Public-Key Cryptography, Beijing, China, April 14-17, 2019, Proceedings, Part II*, volume 11443 of *Lecture Notes in Computer Science*, pages 189–219. Springer, 2019.

[RSG$^+$19] Theo Ryffel, Edouard Dufour Sans, Romain Gay, Francis R. Bach, and David Pointcheval. Partially encrypted machine learning using functional encryption. *CoRR*, abs/1905.10214, 2019.

[WW21] Hoeteck Wee and Daniel Wichs. Candidate obfuscation via oblivious L-WE sampling. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part III*, volume 12698 of *Lecture Notes in Computer Science*, pages 127–156. Springer, 2021.