On the Relationship between FuncCPA and FuncCPA$^+$

Takumi Shinozaki[1], Keisuke Tanaka[1], Masayuki Tezuka[1], and Yusuke Yoshida[1]

Tokyo Institute of Technology shinozaki.t.ae@m.titech.ac.jp, keisuke@is.titech.ac.jp, tezuka.m.ac@m.titech.ac.jp, yoshida.yusuke@c.titech.ac.jp

Abstract. Akavia, Gentry, Halevi, and Vald introduced the security notion of function-chosen-plaintext-attack (FuncCPA security) for public-key encryption schemes. FuncCPA is defined by adding a functional re-encryption oracle to the IND-CPA game. This notion is crucial for secure computation applications where the server is allowed to delegate a part of the computation to the client.
Dodis, Halevi, and Wichs introduced a stronger variant called FuncCPA$^+$. They showed FuncCPA$^+$ implies FuncCPA and conjectured that FuncCPA$^+$ is strictly stronger than FuncCPA. They left an open problem to clarify the relationship between these variants.
Contrary to their conjecture, we show that FuncCPA is equivalent to FuncCPA$^+$. We show it by two proofs with a trade-off between the number of queries and the number of function inputs. Furthermore, we show these parameters determine the security levels of FuncCPA and FuncCPA$^+$.

1 Introduction

Akavia, Gentry, Halevi, and Vald [2] introduced the notion of function-chosen-plaintext-attack (FuncCPA security) for public-key encryption schemes in the context of homomorphic encryption. This notion extends IND-CPA security by adding a functional re-encryption oracle. This oracle takes a function f and a ciphertext ct as input and returns the encryption of the function applied to the ciphertext, $\mathrm{Enc}(pk, f(\mathrm{Dec}(sk, ct)))$, where pk and sk are the public and secret keys, respectively.

In practice, the oracle represents a situation where the client and server communicate during secure computation. In applications of secure computation using homomorphic encryption, the server may query the client for ciphertexts and receive the result as ciphertext to accelerate the process [9, 1, 3].

Intuitively, a functional re-encryption oracle may be unnecessary, as it simply outputs a new ciphertext based on a queried ciphertext and a function. The oracle also seems that any IND-CPA secure public-key encryption scheme should naturally be a FuncCPA secure scheme. However, Akavia et al. showed an IND-CPA secure scheme that can be attacked using a functional re-encryption oracle in [2].

Dodis, Halevi, and Wichs [5] extended the FuncCPA security from single-input to multi-input and studied its properties. Furthermore, they introduced a stronger security notion called FuncCPA$^+$ security. They showed how to construct a FuncCPA$^+$ secure scheme from any IND-CPA secure scheme. FuncCPA$^+$ security is defined by a different experiment compared to FuncCPA security, distinguishing between an oracle that provides functional re-encryption and one that always returns a ciphertext of 0.

Dodis et al. defined FuncCPA$^+$ security as stronger FuncCPA security, but no general difference between these notions has been found. In other words, no scheme has been discovered that is FuncCPA secure but not FuncCPA$^+$ secure. They also discussed the differences between these security notions and obtained two results by restricting the types of functions that can be queried to the functional re-encryption oracle. The first result is obtained when functions are restricted to identity functions only. In FuncCPA security, this restriction is referred to as ReEncCPA security, and in FuncCPA$^+$ security, it is referred to as ReEncCPA$^+$ security. They showed a scheme that is ReEncCPA secure but not ReEncCPA$^+$ secure. The second result is obtained when functions are restricted to single-input functions only. In FuncCPA$^+$ security, this restriction is referred to as 1-FuncCPA$^+$ security. In this case, they showed a scheme that is 1-FuncCPA$^+$ secure but not FuncCPA secure. Although FuncCPA$^+$ security appears to be a stronger notion than FuncCPA security, their equivalence remains an open problem.

1.1 Our Contribution

We show that a public-key encryption scheme that is FuncCPA secure is also FuncCPA$^+$ secure. We summarize the relationships among related security notions in Figure 1.

To investigate in detail, we define the (ℓ, q)-FuncCPA and (ℓ, q)-FuncCPA$^+$ security notions by introducing parameters for the number of function inputs ℓ and the number of queries q, as specified in Definitions 5 and 7. (ℓ, q)-FuncCPA security restricts FuncCPA security by allowing the adversary to query the functional re-encryption oracle up to q times for functions with ℓ inputs. Similarly, (ℓ, q)-FuncCPA$^+$ security restricts FuncCPA$^+$ security with the same parameters. We show the following theorems:

Theorem 1. $(\ell, 2q)$-FuncCPA *implies* (ℓ, q)-FuncCPA$^+$ *for any* $\ell \geq 2, q \geq 1$.

Theorem 2. $(\ell+1, q)$-FuncCPA *implies* (ℓ, q)-FuncCPA$^+$ *for any* $\ell \geq 1$, $q \geq 1$.

We also show that there exists a separation between ReEncCPA and FuncCPA notions by using our theorems.

Corollary 1. *There exists a public key encryption scheme that is* ReEncCPA *and not* FuncCPA.

Proof. If ReEncCPA implies FuncCPA, then from our results, ReEncCPA would imply ReEncCPA$^+$. This contradicts the separation between ReEncCPA and ReEncCPA$^+$ by Dodis et al. [5]. □

By focusing on the number of functional re-encryption oracle queries, we also discover a problem when the attacker performed two oracle queries in the discussion by Dodis et al. [5] regarding a scheme that is 1-FuncCPA$^+$ secure but not FuncCPA secure.

Theorem 3. *For any $\ell, \ell' \in \mathbb{N}$ such that $\ell < \ell'$, if there exists an* IND-CCA1 *secure public-key encryption scheme, there exists a scheme that is* $(\ell, 1)$-FuncCPA$^+$ *and not* $(\ell', 1)$-FuncCPA.

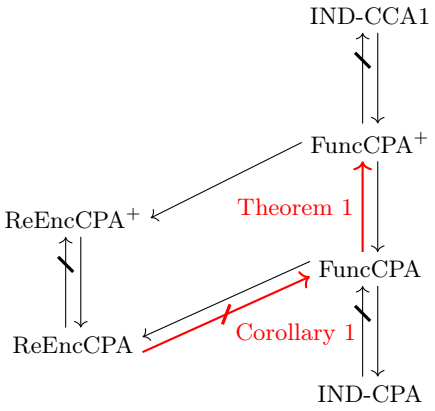

Fig. 1. FuncCPA and related security notions. $A \to B$ represents that a scheme satisfying A also satisfies B. $A \nrightarrow B$ represents that there exists a scheme that satisfy A but does not satisfy B. Black arrows indicate known results. Red arrows indicate our results.

1.2 Technical Overview

Proof of the FuncCPA implies FuncCPA$^+$ We show that when a public-key encryption scheme satisfies FuncCPA security, it also satisfies FuncCPA$^+$ security in Theorem 1 and 2.

For this proof, we construct an algorithm that reduces FuncCPA$^+$ security to FuncCPA security. To simulate the adversary for FuncCPA$^+$ security, we need to switch between the functional re-encryption oracle and the zero-encryption oracle based on a randomly chosen b by the challenger. First, we send messages $m_0 = 0$ and $m_1 = 1$ to the challenger and receive the challenge ciphertext $ct^* = \mathrm{Enc}(m_b)$. Next, we introduce a two-input function $g(x_1, x_2)$. This function g returns x_1 when x_2 is 0, and returns 0 when x_2 is 1. We appropriately respond to the adversary's oracle query using the challenge ciphertext ct^* and the function g through two oracle queries. As the first query, we forward the adversary's oracle query, consisting of a sequence of ciphertexts $ct_1, \ldots, ct_\ell$, and a function f, to the

challenger and receive the response ct'. We query ct', ct^*, g as the second. As a result, we receive $ct'' = \mathrm{Enc}(pk, g(\mathrm{Dec}(sk, ct'), b))$ and use this as the response for the adversary's oracle query. This approach allows us to correctly switch between the functional re-encryption oracle and the zero-encryption oracle based on the challenger's random b. Thus, with this constructed reduction algorithm, we can show our result.

Separations by the Number of Queries We reconsider the relationship between 1-FuncCPA$^+$ security and FuncCPA security as discussed in the paper by Dodis et al. [5] in Section 4. They claimed the following difference between FuncCPA security and FuncCPA$^+$ security:

> If CCA secure encryption schemes exist, then a 1-FuncCPA$^+$ secure encryption scheme is not FuncCPA secure.

However, upon examining their proof with a focus on the number of functional re-encryption queries, we identify a subtle issue.

To address this, we introduce a parameterized version of FuncCPA security, which clarifies the number of queries, and we attempt to correct Dodis et al.'s proof. Nevertheless, we cannot perfect the correction, so we slightly modified their claim to obtain a similar result. Additionally, they assumed IND-CCA2 security, but we show a similar result under the weaker assumption of IND-CCA1 security. We show the result below:

Theorem 3. *For any $\ell, \ell' \in \mathbb{N}$ such that $\ell < \ell'$, if there exists an* IND-CCA1 *secure public-key encryption scheme, there exists a scheme that is* $(\ell, 1)$-FuncCPA$^+$ *secure and not* $(\ell', 1)$-FuncCPA *secure.*

We consider their proof focusing on the number of functional re-encryption queries. They constructed a public-key encryption scheme with certain features from an IND-CCA2 secure scheme to show their claim. Here is a brief introduction to their constructed scheme. The key generation algorithm runs the IND-CCA2 secure key generation algorithm to obtain public key pk and secret key sk. It then selects two random strings $r, s \in \{0,1\}^\lambda$ and includes them as part of the secret key. Additionally, the public key includes the result $y = \mathcal{OWF}(r \oplus s)$, where the two random strings r and s are XORed and then applied to a one-way function $\mathcal{OWF}$. The encryption algorithm checks if the result of applying a one-way function to the message m matches part of the previously generated public key y. If it matches, return the message. Otherwise, encrypte the message as usual. The decryption algorithm takes the ciphertext (n, ct) as input. This algorithm changes behavior based on the number n. If $n = 1$, it returns r, which is part of the secret key. If $n = 2$, it returns s, which is also part of the secret key. Otherwise, it performs the decryption as usual.

They claimed that this public-key encryption scheme is 1-FuncCPA$^+$ by considering a sequence of q hybrids. For the i-th hybrid, the responses to the first $i - 1$ queries are a ciphertext of 0, and the responses from the i-th query onward are from the functional re-encryption oracle. Indeed, the 0-th hybrid corresponds to 1-FuncCPA$^+$ security with $b = 0$, and the q-th hybrid corresponds

to 1-FuncCPA$^+$ security with $b = 1$. They claimed that the i-th and $i + 1$-th hybrids are indistinguishable in two steps:

- The i-th query not being decrypted to $r \oplus s$ reduces to the one-wayness of $\mathcal{OWF}$.
- The indistinguishability of the i-th and $i+1$-th hybrids reduces to IND-CCA2 security.

Their encryption algorithm has the property of returning the message m as is when $\mathcal{OWF}(m)$ matches $\mathcal{OWF}(r \oplus s)$. Therefore, it must be shown that the i-th query being decrypted to $r \oplus s$ is negligible. However, we consider this reduction to the one-wayness of $\mathcal{OWF}$ to be insufficient. A 1-FuncCPA$^+$ security adversary can obtain the ciphertexts of r and s through two oracle queries due to the decryption algorithm's properties. These must satisfy the relationship $y = \mathcal{OWF}(r \oplus s)$. Simply reducing to one-wayness does not easily provide such r and s to the reduction algorithm.

We argue that a more detailed discussion is necessary on this reduction. Therefore, we fix the number of queries an adversary makes to 1 for this discussion. By fixing it to 1, the reduction algorithm returns either the ciphertext of r or the ciphertext of s, and it can be reduced to the one-wayness of $\mathcal{OWF}$. We present Theorem 3 and discuss the case where the number of queries is extended to an arbitrary q in Section 4.4.

1.3 Related Work

Nuida [8] discussed the case of querying invalid ciphertexts in the context of FuncCPA security. Depending on how to handle invalid ciphertexts, they propose funcCPA† security and funcCPA†† security.

As with FuncCPA, several security notions have been proposed in the context of homomorphic encryption.

Li and Micciancio [6] considered that for approximate homomorphic encryption schemes of CKKS [4], satisfying IND-CPA security alone is insufficient to prevent attacks. They propose IND-CPA^D security for such encryption schemes. Furthermore, they conduct a more detailed discussion by parameterizing the number of queries for IND-CPA^D security.

Mark Manulis and Jérôme Nguyen [7] discussed how fully homomorphic encryption (FHE) schemes achieve security beyond IND-CCA1. They proposed vCCA security and showed that this security is the strongest among several existing security achievable by FHE. Furthermore, they presented a general construction method that transforms any IND-CPA secure FHE scheme into a vCCA secure scheme.

2 Preliminaries

Notations. For a positive integer n, $[n]$ represents the set $\{1, \ldots, n\}$. For a finite set X, $x \xleftarrow{\$} X$ denotes sampling an element x uniformly at random from X. For

an algorithm $\mathcal{A}$, $y \leftarrow \mathcal{A}(x)$ indicates that $\mathcal{A}$ outputs y given input x. λ denotes the security parameter. Probabilistic polynomial time is denoted by PPT. A function $f(\lambda)$ is negligible with respect to λ if it converges faster than $\frac{1}{\lambda^c}$ for all constants $c > 0$, and is denoted as $f(\lambda) = \mathrm{negl}(\lambda)$.

2.1 Public-Key Encryption

Definition 1. *A public-key encryption scheme $\mathcal{E}$ over the message space $\mathcal{M}$ is a tuple of three* PPT *algorithms* $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ *as follows:*

- $(pk, sk) \leftarrow \mathrm{Gen}(1^\lambda)$*: The key generation algorithm takes a security parameter 1^λ as input, and outputs a pair of public and secret keys (pk, sk).*
- $ct \leftarrow \mathrm{Enc}(pk, m)$*: The encryption algorithm takes a public key pk and a message $m \in \mathcal{M}$ as input, and outputs a ciphertext ct.*
- $m' \leftarrow \mathrm{Dec}(sk, ct)$*: The decryption algorithm takes a secret key sk and a ciphertext ct as input, and outputs a message $m' \in \mathcal{M} \cup \{\perp\}$.*

Correctness: A public-key encryption scheme $\mathcal{E} = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ is correct if for any $m \in \mathcal{M}$,

$$\Pr\left[(pk, sk) \leftarrow \mathrm{Gen}(1^\lambda), ct \leftarrow \mathrm{Enc}(pk, m) : \mathrm{Dec}(sk, ct) = m\right] = 1$$

holds.

Definition 2 (IND-CPA/IND-CCA1/IND-CCA2 Security). *Let $\mathcal{E}$ be a public-key encryption scheme over the message space $\mathcal{M}$. We say that $\mathcal{E}$ is* atk *secure for security notion* $\mathrm{atk} \in \{\text{IND-CPA}, \text{IND-CCA1}, \text{IND-CCA2}\}$ *if for any* PPT *adversary $\mathcal{A}$ has the success probability of the following experiment is negligible. The* atk *indistinguishability experiment* $\mathrm{EXP}^{atk}_{\mathcal{A},\mathcal{E}}(\lambda)$ *is defined as follows:*

1. $\mathrm{Gen}(1^\lambda)$ *is run to obtain a key-pair (pk, sk).*
2. *The adversary $\mathcal{A}$ is given the public key pk and access to the oracle $\mathcal{O}_1$.*
3. *$\mathcal{A}$ outputs a pair of messages $m_0, m_1 \in \mathcal{M}$, where $|m_0| = |m_1|$.*
4. *A random bit $b \in \{0, 1\}$ is chosen, and the ciphertext $ct^* \leftarrow \mathrm{Enc}(pk, m_b)$ is computed and given to $\mathcal{A}$. We call ct^* the challenge ciphertext.*
5. *$\mathcal{A}$ have access to the oracle $\mathcal{O}_2$.*
6. *$\mathcal{A}$ outputs a bit $b' \in \{0, 1\}$. The experiment outputs is defined as 1 if $b = b'$, and 0 otherwise.*

For IND-CPA security, the oracles $\mathcal{O}_1$ and $\mathcal{O}_2$ always return $\perp$. For IND-CCA1 security, $\mathcal{O}_1(ct) = \mathrm{Dec}(sk, ct)$ is decryption oracle and $\mathcal{O}_2$ always returns $\perp$. For IND-CCA2 security, $\mathcal{O}_1(ct) = \mathrm{Dec}(sk, ct)$ is decryption oracle and $\mathcal{O}_2$ is the same as $\mathcal{O}_1$, except it returns $\perp$ on the challenge cipher ct^* from the indistinguishability experiment.

2.2 Homomorphic Encryption

We use an additively homomorphic public-key encryption scheme in Section 4.4.

Definition 3. *A homomorphic public-key encryption scheme $\mathcal{E}$ over the message space $\mathcal{M}$ is a tuple of four PPT algorithms* $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec}, \mathrm{Eval})$ *as follows:*

- $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$*: Same as in Definition 1.*
- $\hat{c} \leftarrow \mathrm{Eval}(pk, C, ct_1, ct_2, \ldots, ct_\ell)$*: The evaluation algorithm takes a public key pk, a circuit $C : \mathcal{M}^\ell \to \mathcal{M}$, and a sequence of ciphertexts $ct_1, ct_2, \ldots, ct_\ell$ as input, and outputs a ciphertext $\hat{c}$.*

2.3 FuncCPA Security

Akavia, Gentry, Halevi, and Vald [2] introduced FuncCPA security as a notion for public-key encryption schemes. They formulated this security definition by adding a functional re-encryption oracle to the IND-CPA security game.

Definition 4 (FuncCPA Security). *Let $\mathcal{E}$ be a public-key encryption scheme with message space $\mathcal{M}$ and a family of functions $\mathcal{F} = \{f : (\mathcal{M} \cup \{\perp\})^\ell \to \mathcal{M} \mid \ell \in \mathbb{N}\}$. We say that a scheme $\mathcal{E}$ is* FuncCPA *secure with respect to $\mathcal{F}$ if for all* PPT *adversaries $\mathcal{A}$ there exists a negligible function* $\mathrm{negl}(\cdot)$ *such that:*

$$\Pr[\mathrm{EXP}^{\mathrm{FuncCPA}}_{\mathcal{A},\mathcal{E},\mathcal{F}}(\lambda) = 1] = \frac{1}{2} + \mathrm{negl}(\lambda)$$

where the FuncCPA *indistinguishability experiment* $\mathrm{EXP}^{\mathrm{FuncCPA}}_{\mathcal{A},\mathcal{E},\mathcal{F}}(\lambda)$ *is defined as follows:*

1. $\mathrm{Gen}(1^\lambda)$ *is run to obtain a key-pair (pk, sk)*
2. *The adversary $\mathcal{A}$ is given pk and access to a functional re-encryption oracle $\mathcal{O}$ defined as:*
   - *The oracle $\mathcal{O}$ is given a function $f \in \mathcal{F}$ and ciphertexts $ct_1, ct_2, \ldots, ct_\ell$, where ℓ is the number of the function inputs. Then the oracle computes $m_i \leftarrow \mathrm{Dec}(sk, ct_i)$ for $i = 1, \ldots, \ell$, $m' \leftarrow f(m_1, \ldots, m_\ell)$, and $c' \leftarrow \mathrm{Enc}(pk, m')$, and return c'.*
3. *$\mathcal{A}$ outputs a pair of messages $m_0, m_1 \in \mathcal{M}$ where $|m_0| = |m_1|$.*
4. *A random bit $b \in \{0, 1\}$ is chosen, and the challenge ciphertext $ct^* \leftarrow \mathrm{Dec}(pk, m_b)$ is computed and given to $\mathcal{A}$.*
5. *$\mathcal{A}$ can access the oracle $\mathcal{O}$.*
6. *The adversary $\mathcal{A}$ outputs a bit $b' \in \{0, 1\}$. The experiment's output is defined as 1 if $b = b'$, and 0 otherwise.*

When omitting the function family $\mathcal{F}$, all functions specified as a circuit are assumed to be included.

To conduct a more detailed analysis, we introduce parameters for the number of function inputs ℓ and the number of queries q and refer to (ℓ, q)-FuncCPA security. The number of function inputs ℓ specifies the number of inputs included in the function family. The number of queries q specifies the number of times the adversary accesses the functional re-encryption oracle during the experiment.

Definition 5 ((ℓ, q)-FuncCPA Security). *Let $\ell, q \in \mathbb{N}$ be two positive integers. We say that public-key encryption scheme $\mathcal{E}$ with message space $\mathcal{M}$ is* (ℓ, q)-FuncCPA *secure with respect to $\mathcal{F} = \{f : (\mathcal{M} \cup \{\perp\})^\ell \to \mathcal{M}\}$ if for all* PPT *adversaries $\mathcal{A}$ with up to q oracle queries during the experiment, there exists a negligible function* $\mathrm{negl}(\cdot)$ *such that:*

$$\Pr[\mathrm{EXP}^{(\ell,q)\text{-FuncCPA}}_{\mathcal{A},\mathcal{E},\mathcal{F}}(\lambda) = 1] = \frac{1}{2} + \mathrm{negl}(\lambda)$$

where the (ℓ, q)-FuncCPA *indistinguishability experiment* $\mathrm{EXP}^{(\ell,q)\text{-FuncCPA}}_{\mathcal{A},\mathcal{E},\mathcal{F}}(\lambda)$ *is defined as the same as the* FuncCPA *indistinguishability experiment* $\mathrm{EXP}^{\mathrm{FuncCPA}}_{\mathcal{A},\mathcal{E},\mathcal{F}}(\lambda)$*. Similar to FuncCPA security, when the function family $\mathcal{F}$ is omitted, it is assumed to include any ℓ-input function.*

2.4 FuncCPA$^+$ Security

Dodis, Halevi, and Wichs [5] introduced FuncCPA$^+$ security for public-key encryption schemes. They defined FuncCPA$^+$ security by an experiment that distinguishes between an oracle providing functional re-encryption and an oracle always returning a ciphertext of 0.

Definition 6 (FuncCPA$^+$ Security). *Let $\mathcal{E}$ be a public-key encryption scheme with message space $\mathcal{M}$ and a family of functions $\mathcal{F} = \{f : (\mathcal{M} \cup \{\perp\})^\ell \to \mathcal{M} \mid \ell \in \mathbb{N}\}$. We say that a scheme $\mathcal{E}$ is* FuncCPA$^+$ *secure with respect to $\mathcal{F}$ if for all* PPT *adversaries $\mathcal{A}$ there exists a negligible function* $\mathrm{negl}(\cdot)$ *such that:*

$$\Pr[\mathrm{EXP}^{\mathrm{FuncCPA}^+}_{\mathcal{A},\mathcal{E},\mathcal{F}}(\lambda) = 1] = \frac{1}{2} + \mathrm{negl}(\lambda)$$

where the FuncCPA$^+$ *indistinguishability experiment* $\mathrm{EXP}^{\mathrm{FuncCPA}^+}_{\mathcal{A},\mathcal{E},\mathcal{F}}(\lambda)$ *is defined as follows:*

1. $\mathrm{Gen}(1^\lambda)$ *is run to obtain a key-pair (pk, sk) and a random bit $b \in \{0, 1\}$ is chosen.*
2. *The adversary $\mathcal{A}$ is given pk and access to functional re-encryption oracle $\mathcal{O}_b$ defined as:*
   - *The oracle $\mathcal{O}_b$ is given a function $f \in \mathcal{F}$ and ciphertexts $ct_1, ct_2, \ldots, ct_\ell$, where ℓ is the number of the function f input. Then the oracle computes $m_i \leftarrow \mathrm{Dec}(sk, ct_i)$ for $i = 1, \ldots, \ell$, $m' \leftarrow f(m_1, \ldots, m_\ell)$, and responds with $c' \leftarrow \mathrm{Enc}(pk, m')$ if $b = 1$, and $c' \leftarrow \mathrm{Enc}(pk, 0)$ if $b = 0$.*
3. *$\mathcal{A}$ outputs a bit $b' \in \{0, 1\}$. The experiment's output is defined as 1 if $b = b'$, and 0 otherwise.*

When omitting the function family $\mathcal{F}$, all functions specified as a circuit are assumed to be included.

Similar to FuncCPA security, we introduced parameters for the number of function inputs and the number of queries to conduct a more detailed analysis of FuncCPA$^+$.

Definition 7 ((ℓ, q)-FuncCPA$^+$ Security). *Let $\ell, q \in \mathbb{N}$ be positive integers. We say that a public-key encryption scheme $\mathcal{E}$ with message space $\mathcal{M}$ is* (ℓ, q)-FuncCPA$^+$ *secure with respect to $\mathcal{F} = \{f : (\mathcal{M} \cup \{\perp\})^\ell \to \mathcal{M}\}$ if for all* PPT *adversaries $\mathcal{A}$ with up to q oracle queries during the experiment, there exists a negligible function* $\mathrm{negl}(\cdot)$ *such that:*

$$\Pr[\mathrm{EXP}^{(\ell,q)\text{-FuncCPA}^+}_{\mathcal{A},\mathcal{E},\mathcal{F}}(\lambda) = 1] = \frac{1}{2} + \mathrm{negl}(\lambda)$$

where the (ℓ, q)-FuncCPA$^+$ *indistinguishability experiment* $\mathrm{EXP}^{(\ell,q)\text{-FuncCPA}^+}_{\mathcal{A},\mathcal{E},\mathcal{F}}(\lambda)$ *is defined as the same as the* FuncCPA$^+$ *indistinguishability experiment* $\mathrm{EXP}^{\text{FuncCPA}^+}_{\mathcal{A},\mathcal{E},\mathcal{F}}(\lambda)$*. When we omit the function family $\mathcal{F}$, it is assumed to include any ℓ-input function.*

3 FuncCPA Implies FuncCPA$^+$

In this section, we prove that if a public-key encryption scheme is FuncCPA secure, it is also FuncCPA$^+$ secure. We perform a detailed analysis by parameterizing the number of queries and function inputs. As a result, we obtain two interesting theorems. In Theorem 1, it is necessary to make two queries from the FuncCPA secure adversary for each query from the FuncCPA$^+$ secure adversary. As another approach, Theorem 2 shows a case where the number of queries remains the same, although the number of function inputs increases by one. We infer a slight difference in the relationship between the FuncCPA security and FuncCPA$^+$ security from these theorems.

Theorem 1. $(\ell, 2q)$-FuncCPA *imples* (ℓ, q)-FuncCPA$^+$ *for any* $\ell \geq 2, q \geq 1$

Proof. Let $\mathcal{A}$ be an adversary against $\mathcal{E}$ for (ℓ, q)-FuncCPA$^+$ security and $\mathcal{CH}$ be an challenger for $(\ell, 2q)$-FuncCPA security. We construct a reduction algorithm $\mathcal{B}$ to attack $(\ell, 2q)$-FuncCPA security of $\mathcal{E}$ as bellow.

1. $\mathcal{CH}$ runs and $(pk, sk) \leftarrow \mathrm{Gen}(1^\lambda)$, then passes pk to $\mathcal{B}$.
2. $\mathcal{B}$ sends messages $m_0 = 0, m_1 = 1$ to $\mathcal{CH}$.
3. $\mathcal{CH}$ samples a random bit $b \xleftarrow{\$} \{0, 1\}$ and computes the challenge ciphertext $ct^* \leftarrow \mathrm{Enc}(pk, m_b)$, then passes it to $\mathcal{B}$.
4. $\mathcal{B}$ runs $\mathcal{A}$ with pk as input.
5. $\mathcal{B}$ receives an functional re-encryption oracle query $(ct_1, \ldots, ct_\ell, f)$ from $\mathcal{A}$.
   (a) $\mathcal{B}$ queries $\mathcal{CH}$ with $(ct_1, \ldots, ct_\ell, f)$.
   (b) $\mathcal{CH}$ sends $ct' \leftarrow \mathrm{Enc}(pk, f(\mathrm{Dec}(sk, ct_1), \ldots, \mathrm{Dec}(sk, ct_\ell)))$ back to $\mathcal{B}$.
   (c) $\mathcal{B}$ constructs a function g as

$$g(x_1, x_2) = \begin{cases} x_1 & \textit{if } x_2 = 1 \\ 0 & \textit{if } x_2 = 0 \end{cases}$$

   and queries $\mathcal{CH}$ with (ct', ct^*, g).

(d) $\mathcal{CH}$ sends back $ct \leftarrow \mathrm{Enc}(pk, g(\mathrm{Dec}(sk, ct'), \mathrm{Dec}(sk, ct^*)))$ to $\mathcal{B}$.
(e) $\mathcal{B}$ sends ct to $\mathcal{A}$.

6. The output b' from $\mathcal{A}$ is used as $\mathcal{B}$'s output.

We confirm that the reduction algorithm $\mathcal{B}$ successfully emulates the challenger in the (ℓ, q)-FuncCPA$^+$ experiment. First, consider the case where the challenger $\mathcal{CH}$ samples $b = 0$. In this case, the challenge ciphertext ct^* is the encryption of 0. The reduction algorithm $\mathcal{B}$ sends the oracle query $(ct_1, \ldots, ct_\ell, f)$ from the adversay $\mathcal{A}$ directly to the challenger $\mathcal{CH}$. The challenger $\mathcal{CH}$ computes as follows:

$$\begin{aligned} m_i &= \mathrm{Dec}(sk, ct_i) \quad (i = 1, \ldots, \ell) \\ ct' &\leftarrow \mathrm{Enc}(pk, f(m_1, \ldots, m_\ell)). \end{aligned}$$

The reduction algorithm $\mathcal{B}$ receives ct'. Then, $\mathcal{B}$ sends the second oracle query (ct', ct^*, g). The challenger computes as follows:

$$\begin{aligned} pt_1 &= \mathrm{Dec}(sk, ct') = f(m_1, \ldots, m_\ell) \\ pt_2 &= \mathrm{Dec}(sk, ct^*) = 0 \\ ct &\leftarrow \mathrm{Enc}(pk, g(pt_1, pt_2)) = \mathrm{Enc}(pk, 0). \end{aligned}$$

Indeed, when $b = 0$, it outputs the encryption of 0. Finally, the reduction algorithm $\mathcal{B}$ sends ct to the adversary $\mathcal{A}$. This behavior is identical to that of the (ℓ, q)-FuncCPA$^+$ experiment challenger when $b = 0$.

Next, we consider the case where $b = 1$. In this case, the challenge ciphertext ct^* is the encryption of 1. Similarly, the reduction algorithm $\mathcal{B}$ performs two oracle queries. As a result, $ct \leftarrow \mathrm{Enc}(pk, f(m_1, \ldots, m_\ell))$ is sent to the adversay $\mathcal{A}$. Indeed, when $b = 1$, it sends the result of the functional re-encryption oracle to the adversary $\mathcal{A}$. This behavior is identical to that of the (ℓ, q)-FuncCPA$^+$ experiment challenger when $b = 1$. Therefore, the reduction algorithm $\mathcal{B}$ behaves in the same manner as the challenger for the adversary $\mathcal{A}$ in the (ℓ, q)-FuncCPA$^+$ experiment.

We evaluate the experiment probability of the adversary $\mathcal{A}$ as follows:

$$\Pr[\mathrm{EXP}^{(\ell,q)\text{-FuncCPA}^+}_{\mathcal{A},\mathcal{E},\mathcal{F}}(\lambda) = 1] = \frac{1}{2}\Pr[0 \leftarrow \mathcal{A} \mid b = 0] + \frac{1}{2}\Pr[1 \leftarrow \mathcal{A} \mid b = 1].$$

The reduction algorithm $\mathcal{B}$ simulates the challenger in the (ℓ, q)-FuncCPA$^+$ experiment for the adversary $\mathcal{A}$. Since the reduction algorithm $\mathcal{B}$'s output matches the adversary $\mathcal{A}$'s output,

$$\Pr[0 \leftarrow \mathcal{A} \mid b = 0] = \Pr[0 \leftarrow \mathcal{B} \mid b = 0], \quad \Pr[1 \leftarrow \mathcal{A} \mid b = 1] = \Pr[1 \leftarrow \mathcal{B} \mid b = 1].$$

Therefore, we obtain that:

$$\begin{aligned} \Pr[\mathrm{EXP}^{(\ell,q)\text{-FuncCPA}^+}_{\mathcal{A},\mathcal{E},\mathcal{F}}(\lambda) = 1] &= \frac{1}{2}\Pr[0 \leftarrow \mathcal{A} \mid b = 0] + \frac{1}{2}\Pr[1 \leftarrow \mathcal{A} \mid b = 1] \\ &= \frac{1}{2}\Pr[0 \leftarrow \mathcal{B} \mid b = 0] + \frac{1}{2}\Pr[1 \leftarrow \mathcal{B} \mid b = 1] \\ &= \Pr[\mathrm{EXP}^{(\ell,2q)\text{-FuncCPA}}_{\mathcal{B},\mathcal{E},\mathcal{F}}(\lambda) = 1]. \end{aligned}$$

Since we know the scheme $\mathcal{E}$ satisfies $(\ell, 2q)$-FuncCPA security. As a result, we obtain that:

$$\Pr[\mathrm{EXP}^{(\ell,q)\text{-FuncCPA}^+}_{\mathcal{A},\mathcal{E},\mathcal{F}}(\lambda) = 1] = \frac{1}{2} + \mathrm{negl}(\lambda).$$

□

Theorem 1 requires twice the number of queries to answer the functional re-encryption oracle query from the adversary for (ℓ, q)-FuncCPA$^+$ security. By merging these two queries into a single query, we obtain another reduction algorithm, which is stated in Theorem 2.

Theorem 2. $(\ell + 1, q)$-FuncCPA *imples* (ℓ, q)-FuncCPA$^+$ *for any* $\ell \geq 1$*,* $q \geq 1$*.*

Proof. Modify the oracle query in Theorem 1 as follows:

5. $\mathcal{B}$ receives an oracle query $(ct_1, \ldots, ct_\ell, f)$ from $\mathcal{A}$. For the function f, we construct the function g as

$$g(x_1, \ldots, x_{\ell+1}) = \begin{cases} f(x_1, \ldots, x_\ell) & \text{if } x_{\ell+1} = 1 \\ 0 & \text{if } x_{\ell+1} = 0. \end{cases}$$

The reduction algorithm $\mathcal{B}$ query the challenger $\mathcal{CH}$ with $(ct_1, \ldots, ct_\ell, ct^*, g)$. The challenger $\mathcal{CH}$ calculates $ct' \leftarrow \mathrm{Enc}(pk, g(\mathrm{Dec}(sk, ct_1), \ldots, \mathrm{Dec}(sk, ct_\ell), \mathrm{Dec}(sk, ct^*)))$ and sends ct' back to the reduction algorithm $\mathcal{B}$. The reduction algorithm $\mathcal{B}$ sends ct' to the adversary $\mathcal{A}$.

□

4 FuncCPA Security Separations

In this section, we discuss the separation between FuncCPA security and FuncCPA$^+$ security. Dodis et al. also discussed the separation between FuncCPA security and FuncCPA$^+$ [5] security. However, we notice some subtle points in their claim. Although we cannot fully correct their discussion, we can obtain similar results using parameters such as the number of queries and function inputs. Additionally, we can weaken the assumption in their claim. Their discussion required IND-CCA2 security, but our results show that IND-CCA1 security is sufficient.

In Section 4.1, we describe the discussion by Dodis et al. and the subtle points. We present our results and corrections in Sections 4.2 and 4.3.

4.1 1-FuncCPA$^+$ security and FuncCPA security

Dodis et al. discussed the relationship between FuncCPA security and 1-FuncCPA$^+$ security [5]. They claimed the following.

> "If CCA secure encryption schemes exist, then there exists a 1-FuncCPA$^+$ secure encryption scheme which is not FuncCPA secure."

"1-FuncCPA$^+$" includes any single-input function family and allows an arbitrary number of queries within FuncCPA$^+$. "CCA secure" refers to the IND-CCA2 security.

Carefully observing this proof, we notice the following facts within it.

- In their proof, there is an issue with queries made more than twice.
- We can construct a sequence of hybrid games in IND-CCA1, not IND-CCA2.

To explain their issue in detail, we present the public-key encryption scheme $\mathcal{E}_n$ using an IND-CCA2 secure scheme $\mathcal{E}$.

Construction of $\mathcal{E}_n$: Let $\mathcal{E}$ be a IND-CCA2 secure public-key encryption scheme with message space $\mathcal{M}$ and $OWF(\cdot)$ be a one-way function.

- Gen_n: Given 1^λ, output (pk^n, sk^n) computed as follows. Let $(pk, sk) \leftarrow \mathrm{Gen}(1^\lambda)$, sample n random strings $r_1, \ldots, r_n \leftarrow \{0,1\}^\lambda$, and $y \leftarrow OWF(\oplus_{i=1}^n r_i)$. Then, output $pk^n = (pk, y)$ and $sk^n = (sk, r_1, \ldots, r_n)$.
- Enc_n: Given $pk^n = (pk, y)$ and $m \in \mathcal{M}$, if $y = OWF(m)$ then output m, else output $(0, \mathrm{Enc}(pk, m))$.
- Dec_n: Given $sk^n = (sk, r_1, \ldots, r_n)$ and a ciphertext (tag, c), if $tag = 0$ then output $\mathrm{Dec}(sk, c)$, else output r_{tag}.

A key characteristic of this encryption scheme $\mathcal{E}_n$ is the r_i values generated by Gen_n. Dodis et al. use $\mathcal{E}_2$ (the case where $n = 2$) to discuss their claim. They implicitly proved their claim for a single query, assuming the adversary could not obtain all r_i values. However, multiple queries can be made in reality, allowing an adversary to obtain all r_i values with just two queries. Thus, their proof must be revised, and we attempt to revise the discussion. Although we could not achieve a perfect revision, by using parameterized FuncCPA security and FuncCPA$^+$ security, we can reach a similar theorem as follows:

Theorem 3. *For any $\ell, \ell' \in \mathbb{N}$ such that $\ell < \ell'$, if there exists an* IND-CCA1 *secure public-key encryption scheme, there exists a scheme that is* $(\ell, 1)$-FuncCPA$^+$ *secure and not* $(\ell', 1)$-FuncCPA *secure.*

If Dodis et al. implicitly proved it with one query, they discuss the separation of (1,1)-FuncCPA$^+$ and (2,1)-FuncCPA, and our theorem also indicates the same result as theirs.

Our main difference from their proof lies in the number of queries. Using the query count parameter, we resolve the issues in their proof. Additionally, we can show that the public-key encryption scheme is only IND-CCA1 secure, not IND-CCA2 secure. We prove Theorem 3 by Sections 4.2 and 4.3. In Section 4.4, we also consider the case where the parameter number of queries is two or more.

4.2 $\mathcal{E}_n$ is $(\ell, 1)$-FuncCPA$^+$ secure

For simplicity, we prove that a public-key encryption scheme $\mathcal{E}_n$ is (ℓ, q)-FuncCPA$^+$ secure instead of $(\ell, 1)$-FuncCPA$^+$ security using several lemmas.

Theorem 4. *For any $n, \ell, q \in \mathbb{N}$ such that $\ell q < n$, if an* IND-CCA1 *secure public-key encryption scheme $\mathcal{E}$ exists, there also exists a scheme $\mathcal{E}_n$ that is* (ℓ, q)-FuncCPA$^+$ *secure.*

Proof. When $\ell q < n$, we construct from an IND-CCA1 secure $\mathcal{E} = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ according to the method for constructing $\mathcal{E}_n = (\mathrm{Gen}_n, \mathrm{Enc}_n, \mathrm{Dec}_n)$ given in Section 4.1 and prove that this $\mathcal{E}_n$ is (ℓ, q)-FuncCPA$^+$ secure. To achieve this, we define a sequence of hybrid games: Game 0, Game 0', Game 1, Game 2.i for $i = 0, \ldots, q$ and Game 3.

Game 0. This is the (ℓ, q)-FuncCPA$^+$ secure game of $\mathcal{E}_n$ when $b = 1$. In concrete, the game proceeds as follows:

- $\mathcal{CH}$ generates $(pk, sk) \leftarrow \mathrm{Gen}(1^\lambda)$, samples $r_i \xleftarrow{\$} \{0,1\}^\lambda$ for $i = 1, \ldots, n$, set $x = \oplus_{i=1}^n r_i$ and computes $y \leftarrow \mathcal{OWF}(x)$. Then $\mathcal{CH}$ sends (pk, y) to the adversary $\mathcal{A}$ for (ℓ, q)-FuncCPA$^+$ security.
- When $\mathcal{CH}$ receives a query $((tag_1, ct_1), \ldots, (tag_\ell, ct_\ell), f)$, it decrypts all the ciphertexts to $(m_1, \ldots, m_\ell)$ according to Dec_n and computes $m = f(m_1, \ldots, m_\ell)$.
- If $y = \mathcal{OWF}(m)$ return m.
- Otherwise return $(0, \mathrm{Enc}(pk, m))$ to $\mathcal{A}$.
- Finally, $\mathcal{A}$ outputs b'.

Game 1. This game is almost the same as **Game 0**, except that the challenger $\mathcal{CH}$ does not check $y = \mathcal{OWF}(m)$ and never returns m to the adversary $\mathcal{A}$.

Lemma 1. *There exists a negligible function* $\mathrm{negl}(\cdot)$ *such that*

$$|\Pr[\mathcal{A} \textit{ outputs } 1 \textit{ in } \textbf{\textit{Game 0}}] - \Pr[\mathcal{A} \textit{ outputs } 1 \textit{ in } \textbf{\textit{Game 1}}]| = \mathrm{negl}(\lambda).$$

Proof. We define an event **Bad** as follows:

Bad. Let $\mathcal{A}$ be an adversary against (ℓ, q)-FuncCPA security with the query $(ct_1, \ldots, ct_\ell, f)$, where $f(\mathrm{Dec}_n(sk^n, ct_1), \ldots, \mathrm{Dec}_n(sk^n, ct_\ell))$ equals y, which is part of the public key of $\mathcal{E}_n$.

Then from the definition of **Bad**,

$$\Pr[\mathcal{A} \text{ outputs } 1 \wedge \neg\textbf{Bad} \text{ in } \textbf{Game 0}] = \Pr[\mathcal{A} \text{ outputs } 1 \wedge \neg\textbf{Bad} \text{ in } \textbf{Game 1}]$$

holds. Thus, it is sufficient to show that $\Pr[\textbf{Bad} \text{ in } \textbf{Game 0}] = \mathrm{negl}(\lambda)$. We analyze this probability of the **Bad** event in the following modified game.

Game 0' This game is almost the same as **Game 0** except that, $\mathcal{CH}$ samples $x \leftarrow \{0,1\}^\lambda$, instead of $x = \oplus_{i=1}^n r_i$. Since $\ell q < n$, there exists an i such that the attacker $\mathcal{A}$ does not receive the ciphertext of r_i. Because r_i is chosen uniformly at random, we consider x uniformly random from $\mathcal{B}$'s perspective due to $x = \oplus_{i=1}^n r_i$.

Thus, **Game 0** and **Game 0'** are perfectly indistinguishable. In particular, $\Pr[\textbf{Bad in Game 0}] = \Pr[\textbf{Bad in Game 0'}]$.

Finally, we show $\Pr[\textbf{Bad in Game 0'}] = \mathrm{negl}(\lambda)$ by reduction to one-wayness of $\mathcal{OWF}$. Specifically, we construct a reduction algorithm B as follows:

1. $\mathcal{CH}$ performs $x \xleftarrow{\$} \{0,1\}^\lambda$ and computes $y = \mathcal{OWF}(x)$.
2. $\mathcal{CH}$ gives y to $\mathcal{B}$.
3. $\mathcal{B}$ executes $(pk, sk) \leftarrow \mathrm{Gen}(1^\lambda)$. Then, $\mathcal{B}$ performs $r_i \xleftarrow{\$} \{0,1\}^\lambda$ for $i = 1, \ldots, n$. It sets $pk^n = (pk, y)$ and $sk^n = (sk, r_1, \ldots, r_n)$.
4. $\mathcal{B}$ gives pk^n to $\mathcal{A}$.
5. $\mathcal{B}$ receives the oracle query $\big((tag_1, ct_1), \ldots, (tag_\ell, ct_\ell), f\big)$ from $\mathcal{A}$.
   (a) For $i = 1, \ldots, \ell$, if $tag_i = 0$, set $m_i = \mathrm{Dec}(sk, ct_i)$; else set $m_i = r_{tag_i}$.
   (b) Sets $x = f(m_1, \ldots, m_\ell)$.
   (c) If $y = \mathcal{OWF}(x)$, outputs x; else returns $\big(0, \mathrm{Enc}(x)\big)$ to $\mathcal{A}$.
6. When $\mathcal{B}$ receives the output from $\mathcal{A}$, it outputs $\perp$.

□

Next, we define hybrid games in **Game 2.i** for $i = 0, \ldots, q$. Note that **Game 2.0** is precisely the same as **Game 1**.

Game 2.i Up to the $(i-1)$-th time, respond with functional re-encryption oracle; from the i-th time onward, respond with zero encryption oracle.

Lemma 2. *Assume $\mathcal{E}$ satisfies* IND-CCA1 *secure, then for all $i = 0, \ldots, q-1$,*

$$|\Pr[\mathcal{A} \textit{ outputs } 1 \textit{ in } \textbf{\textit{Game 2.i}}] - \Pr[\mathcal{A} \textit{ outputs } 1 \textit{ in } \textbf{\textit{Game 2.(i+1)}}]| = \mathrm{negl}(\lambda)$$

holds.

Proof. We construct a reduction algorithm B, which plays the IND-CCA1 security experiment of $\mathcal{E}$. As an adversary $\mathcal{A}$ against (ℓ, q)-FuncCPA$^+$ security, we define the reduction algorithm $\mathcal{B}$ against IND-CCA1 security as follows:

1. $\mathcal{CH}$ executes $(pk, sk) \leftarrow \mathrm{Gen}(1^\lambda)$.
2. $\mathcal{CH}$ passes pk to $\mathcal{B}$.
3. $\mathcal{B}$ samples $r_i \xleftarrow{\$} \{0,1\}^\lambda$ for $i = 1, \ldots, n$. Let $x = \oplus_{i=1}^n r_i$, and calculate $y = \mathcal{OWF}(x)$. It sets $pk^n = (pk, y)$ and $sk^n = (\perp, r_1, \ldots, r_n)$.
4. $\mathcal{B}$ passes pk^n to $\mathcal{A}$.
5. $\mathcal{B}$ processes the functional re-encryption oracle queries from $\mathcal{A}$ as follows until the $(i-1)$-th query:
   (a) Let the oracle query be $\big((tag_1, ct_1), \ldots, (tag_\ell, ct_\ell), f\big)$.
   (b) For $j = 1, \ldots, \ell$, if $tag_j = 0$, query $\mathcal{CH}$ with ct_j.
      i. $\mathcal{CH}$ executes $m \leftarrow \mathrm{Dec}(sk, ct_j)$ for $\mathcal{B}$'s query ct_j and returns m.

ii. $\mathcal{B}$ sets the response m from $\mathcal{CH}$ as m_j.

(c) If $tag_j \neq 0$, set $m_j = r_{tag_j}$.

(d) $\mathcal{B}$ returns $\Big(0, \mathrm{Enc}\big(pk, f(m_1, \ldots, m_\ell)\big)\Big)$ to $\mathcal{A}$.

6. $\mathcal{B}$ receives the i-th oracle query $\big((tag_1, ct_1), \ldots, (tag_\ell, ct_\ell), f\big)$ from $\mathcal{A}$.
   (a) For $j = 1, \ldots, \ell$, if $tag_j = 0$, query $\mathcal{CH}$ with ct_j.
      i. $\mathcal{CH}$ executes $m \leftarrow \mathrm{Dec}(sk, ct_j)$ for $\mathcal{B}$'s query ct_j and returns m.
      ii. $\mathcal{B}$ sets the response m from $\mathcal{CH}$ as m_j.
   (b) If $tag_j \neq 0$, set $m_j = r_{tag_j}$.
   (c) Set $pt_0 = f(m_1, \ldots, m_\ell)$ and $pt_1 = 0$, then send them to $\mathcal{CH}$.
   (d) $\mathcal{CH}$ samples $b \xleftarrow{\$} \{0, 1\}$ and calculates $ct^* = \mathrm{Enc}(pk, pt_b)$.
   (e) $\mathcal{B}$ returns $(0, ct^*)$ to $\mathcal{A}$.
7. $\mathcal{B}$ responds to subsequent oracle queries from $\mathcal{A}$ with $\big(0, \mathrm{Enc}(pk, 0)\big)$.

□

Finally, we undo the change from **Game 0** to **Game 1**.

Game 3 This game is the same as **Game 2.q**, except that now the challenger checks $y = \mathcal{OWF}(m)$ and returns m to $\mathcal{A}$. Note that, this game is exactly the same as (ℓ, q)-FuncCPA$^+$ security game of $\mathcal{E}_n$ when $b = 0$.

Lemma 3. *There exists a negligible function* $\mathrm{negl}(\cdot)$ *such that*

$$|\Pr[\mathcal{A} \textit{ outputs } 1 \textit{ in } \textbf{\textit{Game 2.q}}] - \Pr[\mathcal{A} \textit{ outputs } 1 \textit{ in } \textbf{\textit{Game 3}}]| = \mathrm{negl}(\lambda).$$

Proof. The proof is almost identical to that of Lemma 1. □

To put the above lemma together, we obtain,

$$|\Pr[\mathcal{A} \text{ outputs } 1 \text{ in } \textbf{Game 0}] - \Pr[\mathcal{A} \text{ outputs } 1 \text{ in } \textbf{Game 3}]| = \mathrm{negl}(\lambda).$$

Therefore $\mathcal{E}_n$ satisfies (ℓ, q)-FuncCPA$^+$ security. □

4.3 $\mathcal{E}_n$ is not $(\ell, 1)$-FuncCPA secure

We prove that $\mathcal{E}_n$ is not $(\ell, 1)$-FuncCPA secure.

Theorem 5. *For any $n, \ell \in \mathbb{N}$ such that $n \leq \ell$. If an* IND-CCA1 *secure public-key encryption scheme $\mathcal{E}$ exists, then the scheme $\mathcal{E}_n$ is not* $(\ell, 1)$-FuncCPA *secure.*

Proof. We construct from an IND-CCA1 secure scheme $\mathcal{E} = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ according to the method for constructing $\mathcal{E}_n = (\mathrm{Gen}_n, \mathrm{Enc}_n, \mathrm{Dec}_n)$ given in Section 4.2. We also construct an adversary $\mathcal{A}$ against $(\ell, 1)$-FuncCPA for the public-key encryption scheme $\mathcal{E}_n$.

1. $\mathcal{CH}$ executes $(pk^n, sk^n) \leftarrow \mathrm{Gen}_n(1^\lambda)$.
2. $\mathcal{A}$ receives $pk^n = (pk, y)$ from $\mathcal{CH}$.
3. $\mathcal{A}$ makes an oracle query to $\mathcal{CH}$.
   (a) Define the function $f(x_1, \ldots, x_\ell) = \oplus_{i=1}^{\ell} x_i$.

   (b) Send $\big((1,0^\lambda),\ldots,(\ell,0^\lambda),f\big)$ to $\mathcal{CH}$.
   (c) Receive the response x from $\mathcal{CH}$.
4. $\mathcal{A}$ send $m_0 = 0$ and $m_1 = x$ to $\mathcal{CH}$.
5. $\mathcal{CH}$ randomly selects $b \xleftarrow{\$} \{0,1\}$ and sends $c^* \leftarrow \mathrm{Enc}_n(pk^n, m_b)$ to $\mathcal{A}$.
6. $\mathcal{A}$ receives c^*.
7. $\mathcal{A}$ outputs 1 if c^* matches x, and 0 otherwise.

The response to the oracle query is a ciphertext representing the XOR of $r_1,\ldots,r_\ell$. Since Enc_n returns the plaintext if it matches the image of y, x represents the XOR of $r_1,\ldots,r_\ell$. $\mathcal{CH}$ encrypts either x or 0 based on b. From Enc', c^* is either the plaintext x or a ciphertext of 0. Thus, if c^* matches x, b is 1; otherwise, b is 0. □

4.4 Regarding the query count $q \geq 2$.

We consider the case where the number of queries is $q \geq 2$. We can obtain a similar result to Theorem 5 by considering $\mathcal{E}$ as an IND-CCA1 homomorphic public-key encryption.

Theorem 6. *For any $n, \ell, q \in \mathbb{N}$ such that $q \geq 2, n \leq \ell q - 1$. If an* IND-CCA1 *secure public-key encryption scheme $\mathcal{E}$ exists, then the scheme $\mathcal{E}_n$ is not* (ℓ, q)-FuncCPA *secure.*

Proof. We construct from an IND-CCA1 secure scheme $\mathcal{E} = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ according to the method for constructing $\mathcal{E}_n = (\mathrm{Gen}_n, \mathrm{Enc}_n, \mathrm{Dec}_n)$ given in Section 4.2. We also construct an adversary $\mathcal{A}$ against (ℓ, q)-FuncCPA for the public-key encryption scheme $\mathcal{E}_n$.

1. $\mathcal{CH}$ executes $(pk^n, sk^n) \leftarrow \mathrm{Gen}_n(1^\lambda)$.
2. $\mathcal{A}$ receives $pk' = (pk, y)$ from $\mathcal{CH}$.
3. $\mathcal{A}$ makes a total of $q-1$ oracle queries to $\mathcal{CH}$.
   (a) Define the function $f(x_1,\ldots,x_\ell) = \oplus_{i=1}^{\ell} x_i$.
   (b) For the i-th oracle query, $i \in [q-1]$, send $\big(((i-1)\ell+1,0^\lambda),\ldots,(i\ell,0^\lambda),f\big)$ to $\mathcal{CH}$.
   (c) Receive the response $(0, ct_i)$ from $\mathcal{CH}$.
4. $\mathcal{A}$ perform XOR operations on $ct_1,\ldots,ct_{q-1}$ using homomorphic operations, and let the result be ct.
5. For the final oracle query, $\mathcal{A}$ send $\big((0,ct),((q-1)\ell+1,0^\lambda),\ldots,(q\ell-1,0^\lambda),f\big)$ and let the result be x.
6. $\mathcal{A}$ send $m_0 = 0$ and $m_1 = x$ to $\mathcal{CH}$.
7. $\mathcal{CH}$ samples $b \xleftarrow{\$} \{0,1\}$ and sends $c^* \leftarrow \mathrm{Enc}_n(pk^n, m_b)$ to $\mathcal{A}$.
8. $\mathcal{A}$ receives c^*.
9. $\mathcal{A}$ outputs 1 if c^* matches x, and 0 otherwise.

The adversary $\mathcal{A}$ obtains ciphertexts that represent the XOR of $r_1,\ldots,r_{\ell(q-1)}$ from the first $q-1$ oracle queries and homomorphic operations. The response to the final oracle query is a ciphertext representing the XOR of $r_1,\ldots,r_{\ell q-1}$. Since

Enc_n returns the plaintext if it matches the image of y, x represents the XOR of $r_1, \ldots, r_{\ell q-1}$. $\mathcal{CH}$ encrypts either x or 0 based on b. From Enc_n, c^* is either the plaintext x or a ciphertext of 0. Thus, if c^* matches x, b is 1; otherwise, b is 0. □

Using Theorem 4 from Section 4.2 and the previously mentioned Theorem 6, we can derive Theorem 3 in a form corresponding to a general number of queries q.

Theorem 7. *For any $\ell, \ell', q, q' \in \mathbb{N}$ such that $(q' = 1, \ell q < \ell')$ or $(q' \geq 2, \ell q < n \leq \ell' q' - 1)$. If an* IND-CCA1 *secure public-key encryption scheme exists, then there exists a scheme that is* (ℓ, q)-FuncCPA$^+$ *secure and not* (ℓ', q')-FuncCPA *secure.*

References

1. A. Akavia, D. Feldman, and H. Shaul. Secure search on encrypted data via multi-ring sketch. In D. Lie, M. Mannan, M. Backes, and X. Wang, editors, *ACM CCS 2018*, pages 985–1001. ACM Press, Oct. 2018.
2. A. Akavia, C. Gentry, S. Halevi, and M. Vald. Achievable CCA2 relaxation for homomorphic encryption. In E. Kiltz and V. Vaikuntanathan, editors, *TCC 2022, Part II*, volume 13748 of *LNCS*, pages 70–99. Springer, Cham, Nov. 2022.
3. R. Bost, R. A. Popa, S. Tu, and S. Goldwasser. Machine learning classification over encrypted data. In *NDSS 2015*. The Internet Society, Feb. 2015.
4. J. H. Cheon, A. Kim, M. Kim, and Y. S. Song. Homomorphic encryption for arithmetic of approximate numbers. In T. Takagi and T. Peyrin, editors, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 409–437. Springer, Cham, Dec. 2017.
5. Y. Dodis, S. Halevi, and D. Wichs. Security with functional re-encryption from CPA. In G. N. Rothblum and H. Wee, editors, *TCC 2023, Part II*, volume 14370 of *LNCS*, pages 279–305. Springer, Cham, Nov. / Dec. 2023.
6. B. Li and D. Micciancio. On the security of homomorphic encryption on approximate numbers. In A. Canteaut and F.-X. Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 648–677. Springer, Cham, Oct. 2021.
7. M. Manulis and J. Nguyen. Fully homomorphic encryption beyond IND-CCA1 security: Integrity through verifiability. In M. Joye and G. Leander, editors, *Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part II*, volume 14652 of *Lecture Notes in Computer Science*, pages 63–93. Springer, 2024.
8. K. Nuida. How to handle invalid queries for malicious-private protocols based on homomorphic encryption. In *Proceedings of the 9th ACM on ASIA Public-Key Cryptography Workshop*, pages 15–25, 2022.
9. W. Wang, Y. Jiang, Q. Shen, W. Huang, H. Chen, S. Wang, X. Wang, H. Tang, K. Chen, K. E. Lauter, and D. Lin. Toward scalable fully homomorphic encryption through light trusted computing assistance. *CoRR*, abs/1905.07766, 2019.